DİJİTAL KÜTÜPHANE KANALI

# Keyifli Okumalar Dileriz

Binlerce kategoride milyonlarca PDF e-kitap, ders kitapları, dergiler ve romanlara platformumuz üzerinden ücretsiz ulaşabilirsiniz.

RESMİ WEB SİTEMİZ

**https://rantey.org**

Yedek ve Duyuru Kanallarımız
Sitemize erişim engeli gelmesi veya adres değişikliği durumunda aşağıdaki kanallardan güncel giriş adresine erişebilirsiniz:

• Telegram: https://t.me/birkitapbir

• WhatsApp Kanalımız : https://whatsapp.com/channel/0029VbDHv8uE50Us4IvMoc0Y

# BİLGİSAYAR

ÇAĞDAŞ TEKNOLOJİ

# ANSİKLOPEDİSİ

7

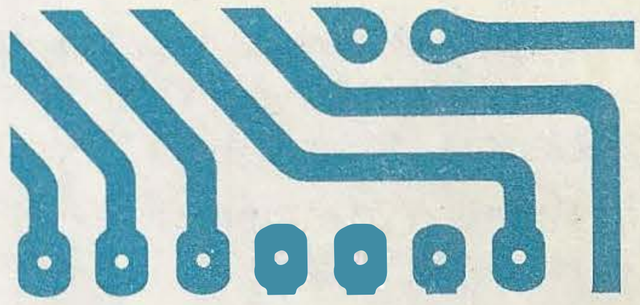
İletişim Yayınları

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Ücretsiz PDF Kitap Arşivi | https://arikey.org

# BİLGİSAYAR

## ÇAĞDAŞ TEKNOLOJİ ANSİKLOPEDİSİ



Cilt.7

Ücretsiz PDF Kitap Arsivi - https://rantey.org

# BİLGİSAYAR

## ÇAĞDAŞ TEKNOLOJİ ANSİKLOPEDİSİ

**İletişim Yayıncılık A.Ş. adına sahibi**
Vedat Çakmak

**Genel Yönetmen**
Murat Belge

**Yayın Yönetmeni**
İskender Savaşır

**Sorumlu Yazı İşleri Müdürü**
Ahmet Ziya Bayman

**Danışma Kurulu**
Kemal İnan, Ayşe Şenatalar, Taha Parla

**Yayın Kurulu**
Vedat Çakmak, Bilge Karasu, İskender Savaşır

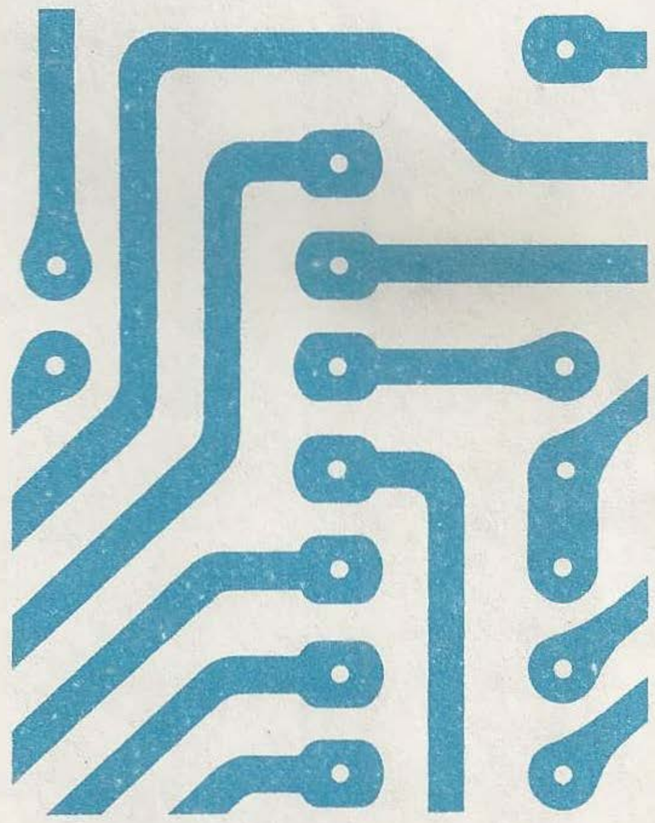
**Çeviri ve Yayına Hazırlayanlar**
Amet Ziya Bayman, Nilüfer Güngören,
Cengiz Onural, Ayşe Özberki

**Teknik Yönetmen**
H.Mirav Yüzgeç

**Grafik/Pikaj**
Süleyman Hakan

**Dizgi**
Hüsnü Abbas, Nuri Abbas,
Hamit Balaban, Zuhal Yolgun

**Baskı**
Çiftay Matbaacılık Ltd. Şti.



© *Armando Curcio Editore, 1984*

Özgün adı: **"Basic, Enciclopedia dell'Informatica, dei Mini e Personal Computer"**
Sante Senni yönetiminde, E.G.S. Şirketi tarafından Giulia Abriani, Maria Albergo, Giovanni Aloisi, Vittoria Antinori, Alberto Berni, Carla Bertini, Ricardo Cantani, Maria Rita Ciucci, Gabriella Costarelli, Patrizia Fazzio, Renato Lazzarini, Marcella Mareaccini, Gianni Mazzoleni, Rolando Mazzoni, Piergiorgio Palma, Rosella Pozza, Laura Salvini, Roberto Sed, Ugo Spezice ve Graziella Tassi'nin katkılarıyla hazırlanmıştır.

**İletişim Yayınları**
Klodfarer Caddesi, İletişim Han, Cağaloğlu, İstanbul
Tel: 520 14 53/54/55

© Bütün hakları İletişim Yayıncılık A.Ş.'ye Aittir.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Genel üretim kurallarına dayanan az sayıda çıkarım mekanizması kullanmanın yanısıra en ciddi uzman sistemleri tanıma bağlı olan bazı çıkarım mekanizmaları gerektirirler. Özelde bir çok uzman sistem genel akıl yürütme işlemine bağlı olmayan çıkarım mekanizmaları gereksinirler. Gerçekte değişik uzman sistemler, gerekli özel-amaçlı çıkarım mekanizmalarının sayısına göre nitelenebilir. Uzman sistemler neredeyse tümüyle özel amaçlı mekanizmalara dayananlardan çoğunlukla genel amaçlı algoritmalara dayananlara kadar uzanmaktadır. Satranç programları ilk tipten uzman sistemlere MYCIN ise ikinci tipe örnektir.

## Çıkarım Mekanizmaları

Çıkarım mekanizmaları, belli bazı akılyürütme tarzlarının formelleştirilmiş biçimleridir. Kuralların rastgele bir sıradansa, belirli bazı sıra ve düzenlere göre veya kısmi bazı hedefler gözetilerek uygulanması, birçok vakada olumlu sonuç vermiş olabilir. Bu durumda bir insan yeni bir problemle karşılaştığında, bütün bildiklerini seferber etmeyecek, ulaşabildiği bütün çıkarım kurallarını, rastgele bir sıra içerisinde problemi ifade eden tümcelere uygulamayacaktır. Aksine işe, daha önce başarılı olmuş bazı akılyürütme tarzlarını (çıkarım mekanizmalarını) deneyerek başlayacaktır.

Bu bakımdan çıkarım mekanizmaları, tıpkı karar yordamları gibi kurallar sistemine belli bir anlamda dışarıdan bakmayı, kuralların işleyişini kaydetmeyi ve bunlardan yeni genellemelere gitmeyi gerektirir. Ancak genellemelerden hiç değilse bir kısmı çoğu uzman sistemin akılyürütme bileşenine aksiyom ve kurallarla birlikte verilmiştir. Bunlardan önemli bazılarını aşağıda inceleyeceğiz.

### İkili Çözüm

İkili **(binary)** çözümün amacı uygun birer kök içeren iki tümceden hareketle yeni bir tümce türetmektir. Formel olarak ikili çözüm, iki tümce alır, her ikisinde kendisi ortak fakat işareti farklı (olumlanmış ve değillenmiş) iki tümce alır ve kökleri birleşiyorsa, yeni bir tümce türetir.

İkili çözümün gerçekleştirdiği çıkarımları önermeler mantığında aşağıdaki şekilde ifade edebiliriz:

((NOT P) OR R)
P OR Q

---

(Q OR R)

ya da

IF NOT P THEN R
IF P THEN Q

---

Q OR R

Klasik mantığın dilinde ''modus ponens'' diye anılan çıkarım, ikili çözümün bir halidir.

Bir örnek verelim:

NOT İŞİ(x,hemşirelik) OR ERKEK(x)
NOT ERKEK(x) OR KADIN(x)

---

NOT İŞİ(x, hemşirelik) OR KADIN(x)

### Birim Sonuç Çözümü

Birim sonuç (BS) çözümünün hedefi tek bir yüklemden oluşan bir birim tümce türetmektir. BS-çözümü ancak, birim olmayan bir tümceyle bir dizi birim tümceye uygulanabilir. Ayrıca, birim olmayan tümcenin, birim olan tümcelerin sayısından bir tane fazla yüklem içermesi gerekir (ancak bu yüklemin ille de farklı bir yüklem olması gerekmez).

Bir bakıma, BS-çözümü bir dizi ikili çözüm olarak düşünülebilir. Çünkü yapılan birim olmayan tümceyle, birim olanların her birini birleştirmekten ibarettir. Ama bütün birleştirmelerin aynı anda yapılması, yani arada ara sonuçların türemesine yol açmaması gerekir. Yine bir örnek verelim:

NOT EVLİ(x,y) OR NOT ANNESİ(x,z) OR
BABASI (y,z)
EVLİ(Ayşe, Ahmet)

---

NOT BABASI(Ahmet, Ali)

Yaklaşık yorumu: Eğer x, y'le evliyse ve x, z'nin annesi değilse, o zaman y, z'nin babası değildir.

Ayşe Ahmet'le evli. Ahmet Ali'nin babası değil. O halde Ayşe Ali'nin annesi değil.

### Aşırı-Çözümlülük

Aşırı çözümlülük **(hyper resolution)**, çekirdek (nucleus) adı verilen bir tümceyle uydu (satellite) adı verilen bir dizi tümceden yeni bir tümce üretir. Çekirdeğin köklerinden hiç değilse birinin negatif (değillenmiş) olması ve çekirdekteki yüklemlerin sayısının uyduların sayısına eşit olması gerekir. Hedef pozitif (olumlanmış) bir tümce türetmektir.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Örnek:

NOT EVLİ(x,y) OR NOT ANNESİ(x,z) OR
BABASI(y,z)
EVLİ(Ayşe, Ahmet) OR YAŞLI(Ayşe, Ahmet)
ANNESİ(Ayşe, Ali)

---

BABASI(Ahmet, Ali) OR YAŞLI(Ayşe, Ahmet)

Görüldüğü gibi birim sonuç çözümünün aksine, aşırı çözümlülük, birim olmayan sonuçlar türetebiliyor.

Kuşkusuz, örnek olarak verdiğimiz bu çıkarım mekanizmaları, uzman sistemlerde kullanılan mekanizmaların tamamı değildir.

## Uzman Sistemlerde Belirsizlik Sorunu

Daha önceki bölümlerde sergilenen yaklaşımın tersine birçok uzman sistemin karşılaştığı bir sorun da belirsizlik sorunudur. İddialar (assertion) kesin bir güvenle ileri sürülemez; bazıları yalnızca doğru olma olasılığı taşırlar. Örneğin biz de kendi pratisyen programımızda uğraşırken ''hastanın ateşi varsa grip olabilir'' türünden tümcelere ihtiyaç duymuştuk. Burada ''olabilir'' fiili sonucun kesin değil olası olduğunu ifade etmektedir. Ancak bir sonucun kesinliği diğer bazı yöntemlerin yanında standart tümce notasyonu ve standart otomatik akıl yürütme teknikleri kullanılarak da kolayca hesaplanabilir. Çeşitli derecelerde kesinliğin varlığıyla başa çıkmak için belirlenen yüklemlere cümlenin (önermenin) bilinen kesinliğini belirten ekstra bir argüman eklenir. Sonuçta buna tekabül eden argüman öncüllerin kesinliklerinin bir fonksiyonu olarak ortaya çıkmaktadır. Böylece çeşitli uzman sistemlerde gerekli olan belirsizlik kavramının otomatik akıl yürütme bağlamında halledilebileceğini görüyoruz.

Ne yapılabileceğini göstermek için daha önce verilen MYCIN kuralını tümcelerle ifade etme sorununu irdeleyelim. Önce belirsizlik sorunuyla başa çıkmalıyız. EMYCIN dayalı sistemlerde (belirlilik) kesinlik faktörü (KF) iki kavrama dayalıdır:

Bir hipoteze inanç ölçüsü (İÖ) ve bir hipoteze inamama ölçüsü (İMÖ). İO, O'dan -hiç inanç-, tümüyle inanmaya kadar uzanır. Benzer şekilde İMÖ: O'dan -hiç inançsızlık yok- 1'e tümüyle inanmamaya kadar uzanır. KF'yi tanımlayan formül,

KF = İÖ—İMÖ'dür.

Böylece KF—1'den 1'e kadar değişir. KF için, gerekli yüklemlerden birine ekstra bir argüman ekleriz.

EMYCIN'ın önemli kullanımlarından biri gündelik dilde ifade edilmiş kurallarla makinenin kendi dilindeki kuralları birbirlerine çevirmektir. Bunun için yüklemler belirlenebilir ve çeviri için otomatik mekanizmalar kullanılabilirdi. Biz bunu yerine basitçe uygun yüklemleri seçeceğiz. Çevireceğimiz kuralın dört öncül koşulu var:

**1** / Tedavi gerektiren enfeksiyon menenjit ve
**2** / Hastada ciddi deri ya da doku enfeksiyonu belirtileri var ve
**3** / Kültürde organizma görülmüyor ve
**4** / Enfeksiyon bakteriyel tipte

İlki şöyle temsil edilebilir:

EŞİT (şikayet menenjit)

Bu şikayetin menenjit olarak tanımlandığını belirtmektedir. İkncisi şöyle temsil edilebilir.

CİDDİENF (xkf)

ve $ GT(xkf,O)

Bunun anlamı xkf'nin ciddi deri ya da doku enfeksiyonunun kesinlik faktörü olduğu ve xkf>0'dır. Üçüncü öncül koşul şöyle temsil edilebilir:

GÖRÜLMEDİ (organizma, kültürde)

ve dördüncüsü de

ENFEKTİPİ (bakteriyel,1)

Bunun anlamı enfeksiyonun bakteriyel tipte olup kesinlik faktörünün 1 olduğudur.

Sonuç,

> Enfeksiyona neden olan organizmanın (kültürde ya da doku sıvısında görülen hariç) stophylococcus-coagpos (,75) streptocossus group-a (.5) olabileceğine ilişkin belirti vardır.

normal olarak iki satırla temsil edilecektir: NEDEN (scp,.75)

ve NEDEN (sga,.5)



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Verili notasyonu kullanarak şu iki madde ortaya çıkar:

EŞİT (şikayet menenjit)
CİDDİENF (xkf)
GT (xkf,0)
GÖRÜLMEDİ (organizma, kültür)
ENFEKTİPİ (bakteriyel, 1)
NEDEN (sga, .5)

Eşit derecede iş görecek başka temsiller de mümkündür.

## Basit Bir Uzman Sistem

Aşağıdaki uzman sistem öğün planlaması ve buna ilişkin diyet sorunlarına yardımcı olmaktadır. Sistemi tümceler biçiminde verecek uzman sistemlerin kolaylıkla tipik otomatik akıl yürütme programlarının anlayabileceği bir dilde temsil edildiklerini göstereceğiz. Uzman sistemin yalnızca küçük bir parçasını vereceğiz ama bu tartıştığımız şeylere somut örnek oluşturmaya yetecek.

Diyet danışmanı -burada sunduğumuz uzman sistem- evde ne gibi yiyeceklerin bulunduğunu nelerin alınması gerektiğini, bugün ne yenilmiş olduğunu, geçen hafta boyunca yenilenleri, yeterli vitamin alınıp alınmadığını ve doğru miktarda kalori alınıp alınmadığını izlemek için kullanılabilir. Üstelik öğün ve kahvaltılar için tavsiyelerde bulunur. Böylece sistem normal veri işleyiciliğini - gıda envanteri izlemek - öneri yapma yeteneğiyle birleştirmektir.

Bu örnekte sistemin akıl yürütme bileşenine yoğunlaşacağız. Bilgi edinme bileşeninin yaratılmış olduğunu varsayıyoruz ve diyet danışmanının kullandığı olgu ve kurallar doğru kabul edilmiş. Kullanıcı arabirimi olayları sisteme aktarıyor ve cevapları sergiliyor.

Diyet danışmanı karar vermek için gereken veri başlıklarından oluşan durumu elde etmek için otomatik bir akılyürütme programına bağlıdır. Bir olay olduğunda sözkonusu uzman sistem, sistemin durumundan aldığı bilgiyi kullanarak olayı işler ve uygun yanıtı hesaplar. Bir öğünün tüketilmesi bir olaydır (event), öğün için yemek listesi istenmesi ise bir diğer olay. Ara sıra olay, durumda bir değişiklik gerektirir.

Bu uzman sistemi yürütmek için otomatik akıl yürütme programı 5 tümce listesi işler:

1 / **Genel-aksiyom listesi:** Bilginin nasıl işleneceğine ilişkin kurallar da içerir.

2 / **Durum listesi:** Sistemin durumunu belirleyen olguları veren temel tümceleri (unit ground clauses) içerir. Örneğin
KALORIDURUM (şimdilik (1345), hedef (2200))
durum listesinde yer alarak bugün 1345 kalori alınmış olduğunu günlük hedefinse 2200 olduğunu bildirebilir.

3 / **Demodülator listesi:** Türetilmiş tümceleri basitleştirmek ya da normalleştirmek için kullanılabilen pozitif birimlerini içermektedir.

4 / **Destek-dizisi listesi:** Bir olay oluncaya kadar boş kalır. O zaman da olaya tekabül eden tümce destek-dizisi listesinde bir ya da iki tümce olarak gözükür.

5 / En son liste olan **verilmişler-listesi:** Bir zamanlar destek-dizisi listesinde olup görevlerini tamamlayarak buraya göçmüş olan tümceleri barındırır.

Bir olay olduğunda tümceler destek-dizisi listesinde gözükür. Bunlar ''verili tümceler'' olarak akılyürütme programındaki tümce ilgisini odakladıkça birer birer seçilirler. Her verili tümce için, çıkarım kuralı tarafından gerektirilen ve genel aksiyom listesi, durum listesi ve verilmişler listesi tarafından seçilen diğer tümcelere ilişkin olarak verili tümceden çıkarsanan diğer tümceler destek-dizisi listesinin sonuna yerleştirilir. Artı olarak eğer çıkarsanan tümcelerden herhangi biri ENVAN BUGÜNLÜK GÜNLÜK HEDEFLER GIDALAR ya da MENÜTAR öncüllerinden birine sahip temel bir birimse durum listesindeki tekabül eden birimin yerine geçer. Böylece eski birimin yerine yeni çıkarsanan birim yerleştirilir.

Kullanacağımız yaklaşımı az çok soyut bir tarzda tartışmak yerine genel-aksiyom ve durum listelerinde hangi tümcelere ihtiyacımız var, onun üzerinde duralım. Önce durum listesini oluşturan tümcelerle başlıyoruz. Sistemin olaylara tepki göstermesi ve iyi önerilerde bulunması için hangi veri başlıkları gereklidir? Kısaltılmış sistemde aşağıdakini kullanırız:

(a) Şu formdaki bir tümce;

ENVAN (kalem-numarası) (kalem-tan) (elde) (en-az-elde)

yağ yumurta un gibi temel gıda maddelerinin her biri için ne kadar bulunduğu ve ne kadar



Ücretsiz PDF Kitap Arsivi - https://rantey.org

olması gerektiğini takip edebilir. İlk argüman kaleme atıfta bulunmak için rasgele verilmiş bir sayı olan (kalem numarası)nı verir. İkinci argüman tanımı, üçüncü argüman depodaki kalem sayısını, son argümansa yeniden alım önerisini verir - eğer stok bu sayının altına düşerse daha al. Eğer sistem alışveriş listesi yapabilecek kadar genişletilseydi, malzemelerin hangi dükkandan alınacağını da bildirmesini bekleyebilirdiniz

(b) Şu formdaki maddeler;

MENÜKALEM (kalem-numarası) (kalem-tan) (istekler listesi)

herhangi bir menü kalemi için istekleri belirtmekte kullanılır. Bu nedenle

MENÜKALEM (52 ''peynirli makarna'' l(63,l,(46),(13,sıfır))))

peynirli makarnanın üç alt kalemden yapıldığını söylemekte kullanılır. Daha karmaşık bir sistemde tam miktarlarını ve başka tariflerini de verirdik.

Gördüğünüz gibi menü kalemleri ve temel gıda maddelerinden bahsediyoruz. Bazen tek bir gıda maddesi de menü kalemini oluşturabilir ama genellikle bir menü kalemi çeşitli gıda maddelerinden hazırlanır.

(c) şu formdaki maddeler;

GIDASAL ((kalem) (kalori) (vit-a) (vit-b)...)

bir servislik menü kalemindeki gıda içeriklerini listelemekte kullanılırlar. Burada yalnızca kalorilerin ve A vitamini birimlerinin sayısını izlediğimiz için şöyle tümceler

GIDASAL (162,kalori (375),a(16))

kullanılır. Bunun anlamı 162 no.lu kalemin servis başına 375 kalori ve 16 birimlik A vitamini barındırdığıdır.

(d) Şu formdaki tümceler:

BUGÜNLÜK ((kalori) (vit-a) (vit-b)...)

bugün şimdiye kadar tüketilmiş tümcelerin gıda değerlerini verir.

(e) Şu formdaki tümceler;

GÜNLÜKHEDEF ((kalori) (vit-a) (vit-b)...)

günlük hedefleri kaydetmede kullanılır.

(f) Gerek menü tekrarını önlemek gerekse kötü beslenme eğilimlerini saptamak için haftalık programı da kaydediyoruz. Bunu yapmak için iki ayrı tip tümce kullanıyoruz - biri gıda miktarını kaydetmek, öteki yenen gıdaların genel kategorilerini kaydetmek için. İlk tip sistemin beslenme bozukluklarını denetlemesini sağlar. İkincisi sistemin geçmişi tekrarlamayan menüler planlamasını sağlar. İlki şu biçimdedir:

GIDATAR (ort((kal),(vit-a),(vit-b)...))

((gün-7) 1((gün-6),l((gün-5),...))))))

Burada ort(...) kalori ve vitamin tüketiminin ortalama değerlerini verir. İkinci argüman yedi değerlik bir listedir; her değer gün ((kal),(vit-a)...) formundadır ve tek bir gün için beslenme değerlerini verir. Günün sonunda değerler karşılaştırılır, o günün değerleri listeye yazılır ve ortalamalar yeniden hesaplanır. İkinci tipten tarih şu biçimde bir tümcede saklanır:

MENÜTAR ((menü-kalem-liste),(et-listesi), (sebze-listesi))

Tümce geçen hafta boyunca kullanılan menü kalemlerinin, etlerin ve sebzelerin listesini içermektedir.

Bu noktaya kadar hep beraber sistemin durumunu oluşturan veri kalemlerini tartıştık. Şimdi her biri destek-dizisi listesinde bir tümcenin yerine geçen olası olay tiplerini gözden geçirelim. Bu sistemde aşağıdaki olay tiplerine izin verilmektedir.

(a) İlk tip olay bir yemek yendiğinde gerçekleşir. Tümce şu biçimdedir:

YENDİ ((kalem-numarası),(porsiyon boyutu))

Örneğin

YENDİ (92,.5)

92 numaralı menü kaleminden yarım porisyon yenmiş olduğuna işaret eder.

(b) Yemek alındığında şu biçimde tümceler



Ücretsiz PDF Kitap Arsivi - https://rantey.org

ALINDI ((kalem-numarası),(miktarı))

ENVAN tümcelerin eldeki-stok değerlerinin artırılması gerektiğine işaret eder.

(c) Bir yemek yendiği ya da atıldığında aşağıdaki biçimde tümceler yapılır:

KULLANILDI ((kalem-numarası),(miktarı))

(d) Sistemin kullanıcısı bilgi isteyebilir. Özel bir yemeğin gün boyunca tüketilenler göze alınarak kabul edilebilir olup olmadığını sormak için şu tümce eklenmiştir:

PEKİ YA ((kalem-numarası),(miktarı))

sisteme ne yeneceği üzerine tavsiye sormak için aşağıdaki biçimde bir tümce kullanılır:

ÖNER ((yemek-tipi),(et-tipi),(sebze-tipi))

Burada yemek tipi ''ara yemek'', ''kahvaltı'', ''öğle-yemeği'' ya da ''akşam-yemeği'' olabilir. (et-tipi) genel bir sınıf olarak belirtilebilir ya da herhangi bir tipin kabul edilebilir olduğunu göstermek için ''tanımlanmamış'' bırakılabilir. Aynı şekilde (sebze-tipi) istenirse sebzenin tipini belirmekte kullanılabilir. Kuşkusuz kapsamlı bir sistem çok daha fazla gıda kategorisine kısıtlama koyabilir, ancak bu kadar amacımız için yeterlidir.

(e) Son olarak şu şekilde bir tümce;

SON ((gün))

her günün sonunda yazılır.

Sistemin durumlarını oluşturan bilginin yazı düzeni (formatı), olayları temsil eden tümcelerin formatlarıyla birlikte sistemin belli bir kavramsal yapıyı benimsemesini gerektirir. Bu kavramsal yapıya rağmen her olayın nasıl işleneceğini belirleyecek kurallar vermek gereklidir. Önce basit bir olayla başlayalım. Özelde;

Varsayalım ki

ALINDI (92,12)

destek dizisinde göründü. Böyle bir olayı işlemek için hangi kural ya da kurallar gerekli? Bu olabilecek en basit olaylardan biri. Kuralın IF-THEN formu şöyledir:

IF

kalem-numarası i'den n birim aldıysanız

THEN

stoktaki birim sayısını n kadar arttır.

Bunu şu şekilde tümcelere çevirebiliriz:

ALINDI (xkalem,xmiktar)

ENVAN (xkalem,xelde,xenaz)

ENVAN (xkalem,xtanım,ğtop(xelde,
xmiktar),xenaz)

Bu işlem oldukça basittir. Kullanılmış bir kalemin işlenmesi biraz daha zordur. Sistemin stok değerlerinin istenilen minimumun altına düşmediğini kontrol etmesi gerekir. Bu iki tümcenin varlığını gerektirir.

KULLANILDI (xkalem,xmiktar)
ENVAN (xkalem,xtanım,sxelde,xenaz)
ENVAN (xkalem,xtanım,ğeksi(xelde,
xmiktar),xenaz)
ENVAN (xkalem,xtanım,xelde,xenaz)
X LT(xelde,xenaz)
DCT (PRINT I(''Daha-fazla-alma-zamanı'',
xtanım,sıfır)))

Tümcelerden ikincisi yalnızca stoğun çok azalma durumunda bir uyarının yazılmasına yol açmaktadır.

Diğer olayların işlenmesi için gereken mantık çok daha karmaşıktır. Burada yalnızca otomatik akıl yürütmeye dayalı diyet danışmanına verilebilecek çeşitli özellikleri göstermeyi hedeflediğimiz için bunun ayrıntılarına girmiyoruz. Ancak ÖNER olayını işlemek için gereken bazı kuralları incelemek yararlı olabilir.

ÖNER: İşlemek için gereken temel kural şudur:

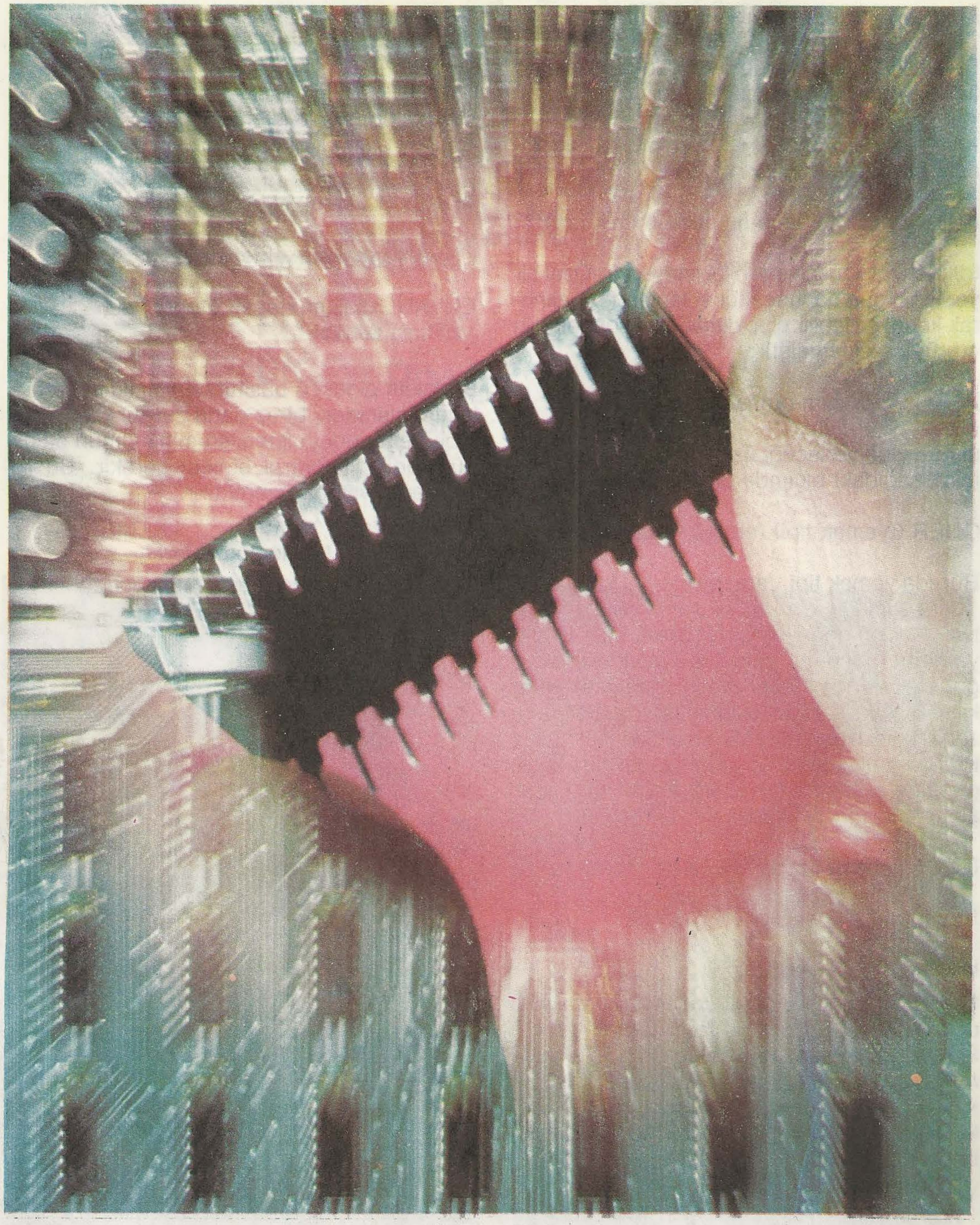
ÖNER (xtip,xet,xsebze)
KABUL EDİLDİ (xtip,xet,xsebze,xlist)
CEVABI AL (xlist)

IF-THEN formatında kural şöyledir:

IF
(istek, parametrelerce kısıtlanmış bir
öğün içinse)
xtip,xet,xseb,
ve



Ücretsiz PDF Kitap Arsivi - https://rantey.org



**Bugün uzman sistemlerin vazgeçilmez bir hale gelmiş olduğu alanlardan biri de tümleşik devre tasarımı ve üretimidir. Özellikle yüksek düzeyli tümleşme (LSI-large scale integration) ve çok yüksek düzeyli tümleşme (VLSI-very large scale integration) teknolojileri uzman sistemlersiz mümkün değildirler. Her ne kadar bu, kendi kendini üreten bir bilgisayar değilse de, kimilerine göre o yönde atılmış önemli adımlardan biridir.**

1736

Ücretsiz PDF Kitap Arsivi - https://rantey.org

olanaklılar listesinin hesabı xlist'te varsa

THEN

xlist'ten rasgele seçimi dene.

Bu kuşkusuz derinlikli bir kural değildir. Şimdi kabul edilebilir çözümlerin listesini hesaplamaya ilişkin altproblemi ele alalım. Olanaklı seçenekler dizisini hesaplamakta kullanılabilecek kural tiplerine bir örnek şudur:

```
KAHVALTI ETLERİ (xkahvaltı)
STOKTA (xkahlist,xstokta)
ŞURDANSEÇ(xstokta,xet,xlist)
KABULEDİLEBİLİR (kahvaltı, xet,
      belirlenmemiş, xlist)
```

Bu kuralın şöyle söylediği düşünülebilir: ''Olanaklı kahvaltı etleri dizisini seç, kahvaltı etleri dizisini (xkahlist) al ve stoktaki (stokta) altdiziyi (stokta) belirle; sonra (olanaklı kısıtlılıklara bağlı) bu diziden seç''. STOKTA (xkahlist, xstokta), stokta, stokta olan elemanların xkahlist'nin altlistesi olduğu zaman doğrudur. ŞURDANSEÇ (xstokta, xet, xlist) ise, xlist, xet'in kullanıcı tarafından ÖNER isteğiyle belirlenmiş kısıtlamalarına bağımlı xstokta bulunan kabuledilebilir aday etler dizisi olduğu zaman doğrudur. Genel-aksiyom listesinin şu biçimde bir tümce içerdiği varsayılır:

KAHVALTI(I(11,1 (18,I(63,I(82,sıfır)))))

Bu kahvaltı için kabuledilebilir etleri sıralamaktadır; ''sosis''in kalem numarası 11, ''jambon''un numarası 18, ''domuz pastırması''nınki 63 ve ''biftek''inki 82'dir.

Üstelik,

STOKTA (list1,list2)

list2, list1'den, list1 de stokta olan elemanların dizisi olarak hesaplandığı zaman tümüyle doğrudur. Bu hesaplamayı gerçekleştiren tümceler şunlardır:

```
STOKTA (sıfır,sıfır)
NOT STOKTA (x,y)
OR NOT ENVAN (xkalem, xtanım,o,xenaz)
OR STOKTA (I(xkalem,x),y)
NOT STOKTA (x,y)
OR NOT ENVAN (xkalem,xtanım,xmiktar,
      xenaz)
OR NOT ĞGT (xmiktar,0)
OR STOKTA (I(xkalem,x),I(xkalem,y))
```

Stokta bulunan et dizilerinden kabuledilebilir bir seçimin nasıl seçileceğini tanımlayan tümceler şöyledir:

```
EŞİT (xet,sıfır)
OR NOT LİSTEDE(xet,xstokta)
OR ŞURDANSEÇ (xstokta,xet,I(xet,sıfır))
LİSTEDE (x,I(x,y))
NOT LİSTEDE (x,y) OR LİSTEDE (x,I(z,y))
NOT MENÜTAR (xkalemler,xetliste,
      xsebzeliste)
OR NOT FARKSIZ  (xstokta,xetliste,xfark)
EŞİT (xfark,sıfır)
OR ŞURDANSEÇ (xstokta,belirlenmemiş,
      xfark)
NOT MENÜTAR (xkalem,xetliste,xsebzeliste)
OR NOT FARKSIZ (xstokta,xetliste,sıfır)
OR ŞURDANSEÇ (xstokta, belirlenmemiş,
      xstokta)
FARKSIZ  (sıfır,x,sıfır)
NOT FARKSIZ (x,y,z) OR NOT LİSTEDE
      (xc,y) OR FARKSIZ  (I(xe,x),y,z)
NOT FARKSIZ  (x,y,z) OR LİSTEDE (xe,y) OR
      FARKSIZ (I(xe,x)y,I(xe,z)
```

Burada FARKSIZ (x,y,z); z kesinlikle x listesinde bulunan, ama y listesinde bulunmayan elemanları içeren bir liste olduğu zaman doğrudur. Bu tümcelerin stokta bulunan ama önceki haftalarda yenmemiş etleri seçtiklerine dikkat edin. Eğer stoktaki bütün etler geçen hafta boyunca yenilmişse, o zaman stoktaki bütün etler kabuledilebilir olarak gösterilir.

Kabul edilebilir etler dizisinden nasıl seçileceğini tanımlayan tümceler bir etyemez tarafından istenildiğinde ''hiçbir et''e izin verilmemelidir. Çünkü aksi takdirde seçim kabul edilemez bir kahvaltıya yol açacaktır.

Tümcelerin kalori ve vitamin bağlamında en iyi seçenekleri kontrol etmediğine dikkat edin. Bu kontroller kolaylıkla eklenebilir ya da kontrollerin her türden gerekli uyarı mesajını verecek şekilde yapıldığı günün sonunae rtelenebilir.

Diyet danışmanı olarak çalışan uzman sistemin küçük bir kesimini tanıttıktan sonra sonuçlara varmak ve tümceleri işlemek için yeterli olan çıkarım kuralının aşırı çözümlülük olduğunu söyleyebiliriz. Elimizde bulunan, tipik otomatik bir akıl yürütme sisteminin terimleriyle tanımlanmış bir uzman sistemdir.

Uzman sistem tümce notasyonu kullanır ve bu daha önceki bölümlerde kapsadığımız çıkarım kuralları ve stratejileriyle çeşitli işlemlere dayanır.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Uygun bir gıda seçeneğini bulmak için gereken bütünlüklü kural dizisi oldukça geniştir. Bir çok uzman sistemde kural dizisi binlerce tümce kapsar. Olguları ve bu olguları yönlendirecek ve soruları cevaplayacak kuralları elde etmek, en azından zahmetli, kimi zaman da son derece güçtür. Uzman kişi bilgiyi verdikten sonra kullanıcı arabirimi geliştirilmelidir. Örneğin verili diyet sisteminin yararlı olabilmesi için, kullanıcının girdi harcamalarına kullanım ve isteklerine uygun olması gerekmektedir. Bu konu bir yana sizlere otomatik akıl yürütme programının günlük problemlerin çözümünde nasıl yardım getirdiğini göstermiş olduk.

## Uzman Sistemlerde Kullanılan Akıl Yürütmenin Derinliği

Geleceğin uzman sistemlerinde akıl yürütme bileşeni ne kadar güçlü olacak? Yaygın bir kanı bugün EMYCIN gibi sistemlerde kullanılan pek e güçlü olmayan akıl yürütme bileşenlerinin uygun olacağı. Bu kanıyı paylaşmadığımıza göre durumu bir inceleyelim.

Yıllar boynca çalışmalar değişik akılyürütme programları tipleri üzerinde ilerledi. Bu tip programlar giderek güçlenmekte birlikte otomatik akılyürütme programları ancak yakın zamanlarda açık sorular düzeyinde yararlı oldu. MYCIN DENORAL ve PROSPECTOR gibi programlarla elde edilen başarı akılyürütme programlarıyla daha fazla başarının elde edileceğinin göstergesi. İyi kullanıcı arabirimleri ve bilgi edinme bileşenlerine tasarlayarak incelikli olmayan akılyürütme bileşenlerine dayalı son derece kullanışlı sistemler yürütüldü. Kullanıcı arabirimleri ve bilgi edinme bileşenlerine ara sıra küçük eklemeler yaparak gelecek beş yıl içinde geniş bir uzman sistemi geliştirmek mümkün görünüyor.

Bu olgular şimdi varolan sınırlı akılyürütme bileşenlerinin öngörülebilir gelecekte de yeterli olacakları görüşünü haklı çıkartabilir mi? Bu soruyu cevaplayabilmek için aşağıdaki kaba akıl yürütme dereceleri sınıflandırmasını veriyoruz. Sınıflandırma uzman kişinin sorunları çözmeye nasıl giriştiğini inceleyerek yapılmıştır.

1 / İlk tipte planın tümü kavranır. Plan, her biri ulaşıldığında üzerinde çalışılan problemi çözen bir dizi hedeften oluşur. Bu akıl yürütme tipine **stratejik akılyürütme** adını veriyoruz.

2 / Bir tür stratejik aklı yürütmeyle temel ara hedefler seçildikten sonra hala her bir ara problem üzerinde çalışmak gereklidir. Genellikle buradaki en yüksek düzeyli işgörece odaksızdır. (Odaklı akılyürütme nosyonunu kısaca ele alacağız). Odaksız akılyürütmede uzmanın hangi olguların açık olduğuna ve hangi tip sonuçların daha çok bilgi sağlayabileceğine ilişkin fikri vardır ama doğrudan cevaba gidecek iyi-tanımlanmış bir işlem bilmemektedir. Bu nedenle bu tip akılyürütmeye **görece odaksız** adı veriyoruz.

3 / Bir ya da daha fazla olguyu neyin izlediğini araştırma sürecinde bazen temel araştırmayı kesip gereken bir değeri hesaplamak gerekebilir. Örneğin bir grafiğin çeşitli özelliklerini araştırırken iki düğüm arasındaki en kısa mesafeyi hesaplamak sık sık gerekebilir. Burada bilinen bir işlem kullanıldığı için akılyürütme tipi daha odaklıdır. İşlem geri izleme gerektirebilir ama temel hesaplama iyice denetim altındadır. Bu tip akılyürütme en iyi Prolog'un yaygın kullanımıyla sergilenebilir. Bu tip akılyürütmeye **görece odaklı** adını veriyoruz.

4 / Son olarak iki numarayı toplamak matriks tersine çevirmek ya da bir uydunun yörüngesini hesaplamak gibi şeylerin bilgisayarla yürütülmesi demek olan tamamen odaklanmış akılyürütmeden söz edebiliriz. Aslında ''akılyürütme'' terimini tam da hak etmeyen bu tip akılyürütme bilgi işlem biliminin kısa tarihi boyunca oldukça fazla incelenmiştir. İyi tanımlanmış hesaplamaları yürütecek çeşitli algoritmalar geliştirilmiştir ve giderek artan bir şekilde geliştirilmektedir.

Son üç tip akılyürütme cevap aramada ''odak'' derecelerinin farklılığıyla ayrılırlar. Kuşkusuz odaklı ve odaksız akılyürütmeler arasında kesin bir ayrım yoktur. Bu kitabın ele aldığı başlıkların çoğu görece odaksız akılyürütme adını verdiğimiz şeyi kullanırlar. Böyle durumlarda olanaklı sonuçlardan hangisinin en üretici olacağına ve sizi çözüme daha çok yaklaştıracağına karar veremezsiniz. Bu nedenle bu tip akılyürütmeye odaksız adını verdik. Öte yandan çözüme varmak için hangi adımların atılması gerektiğine ilişkin oldukça net bir fikir edinebilirsiniz. Bir sonraki adımı



Ücretsiz PDF Kitap Arsivi - https://rantey.org

doğru olarak atacağınıza ne kadar eminseniz akılyürütme o kadar odaklıdır. Matrisi çarpma durumunda gerekli her adım başlamanızdan önce tanımlanabilir, böyle bir faaliyete tümüyle odaklı akılyürütme deyiş nedenimiz de budur.

Görece odaklı ve görece odaksız akılyürütmeler arasındaki farklılıkları anlamak için odaksız akılyürütmeden yararlanan birçok otomatik akılyürütme programında kullanılan temel adımlara bir göz atalım. Böyle bir programın girdisi normal olarak üç ifade (statement) sınıfına ve böylece üç tümce (clause) dizisine ayrılabilir:

1 / Araştırma sistemin toplam özelliklerini tanımlayan genel aksiyomlar.
2 / Hem probleme özgü bilgiyi hem de ulaşılamaz hedef ya da yanlış sonucu varsaymaya yola açan ifadeleri içeren destek-dizisi ifadeleri
3 / Bilinen eşitliklere dayalı olarak tümcelerin yeniden ifade edilmesini sağlayan demodülatörler.

Akılyürütme programı biraz keşif kullanarak destek dizisinden ''ümid vaadeden'' bir tümce seçer. Bundan sonra program seçilen tümce genel aksiyomlar ve destek-dizisinden hareketle ve daha önce seçilen tümceleri izleyerek yeni tümceler dizisini çıkarsar. Her yeni tümce demodülatör ve istenilen bir özel-amaçlı basitleştirme algoritması kullanılarak basitleştirilir ve yeni bilginin daha genel bir bilgi parçası tarafından içerilmiş ya da çıkarsanmış olup olmadığı kontrol edilir. Bir tümce seçme yeni tümceler üretme yeni tümceleri basitleştirme ve tümcelerin korunmasının gerekip gerekmediğini anlamak için kontrol etme süreci görece odaksız bir süreçtir.

Öte yandan odaklı akılyürütmeyi niteleyen, sonucun nasıl elde edileceğine ilişkin net bir fikirdir. Ara sonuçlar pek az gereksinilir; genellikle önemli basitleştirme ve kontrol masrafları ortadan kaldırılır. Odaklı akılyürütmede geri izleme gerekebilir çünkü çözüme nasıl ulaşılacağına ilişkin net bir fikir her zaman gerekli akılyürütme yoluna işaret etmeyebilir. Odaklı akılyürütme yürütmesi hem daha hızlı hem daha kolay bir tiptir.

Bugünkü uzman sistemler yalnızca görece odaklı akılyürütme kullanırlar. Örneğin Prolog; verilen dört akılyürütme tiplerinden son ikisini yürütmek için çok çekici bir yol önerir; bununla birlikte tümüyle odaklı akılyürütmeye ilişkin devasa miktarda bilgi FORTRAN ve PASCAL gibi diğer dillerde kodlanmış durumdadır.

Uzman olarak sınıflandırılan insanların akılyürütmesi muhtemelen ağırlıklı olarak görece odaklı ya da tümüyle odaklıdır. Hatta genellikle verili bir alanda bir uzmanla bir çırak arasındaki tek fark uzmanın problem çözme yönteminin görece odaklı olması, çırağınsa daha genel, daha az yönlenmiş yaklaşım stillerine hapsolmasıdır.

Verili bir alandaki ilerlemenin bir göstergesi, standart problemlere yaklaşım biçimidir. Eski keşifler ağırlıklı olarak stratejik ve görece odaksız tipten akılyürütmelere dayalıdırlar. Sonunda doğrudan çözüm hesaplayan yöntemler bulunmuştur. Bu yöntemler bütün durumlarda ya da durumları ilgilendiren alt-dizilerde iş görebilirler. Ne olursa olsun odaklı ve iyi denetimli oldukları için gerekli çabayı büyük ölçüde azaltırlar. İyi odaklı teknikler bir kez kurulduktan sonra daha genel ve daha az yetkin yöntemler odaklı yaklaşımların istenilen çözüme ulaşmayı başaramadıkları durumlarda kullanılabilirler.

Yine geleceğin uzman sistemlerinde hangi tip akılyürütmenin kullanılacağı sorusuna dönelim. Kuşkusuz böyle sistemlerin temel akılyürütme bileşenleri uygun tarzda kodlanmış binlerce odaklı teknik içerecektir. Geriye yalnızca odaklı kısmın ne kadarının satranç programlarında kullanılan türden özel-amaçlı rutinlerden oluşacağını kestirmek kalıyor. Birkaç yıl içinde uzman sistemlerin akılyürütme bileşenlerindeki çalışmanın büyük bölümü odaklı tekniklerin geliştirilmesi olacak. İki ana görev var:

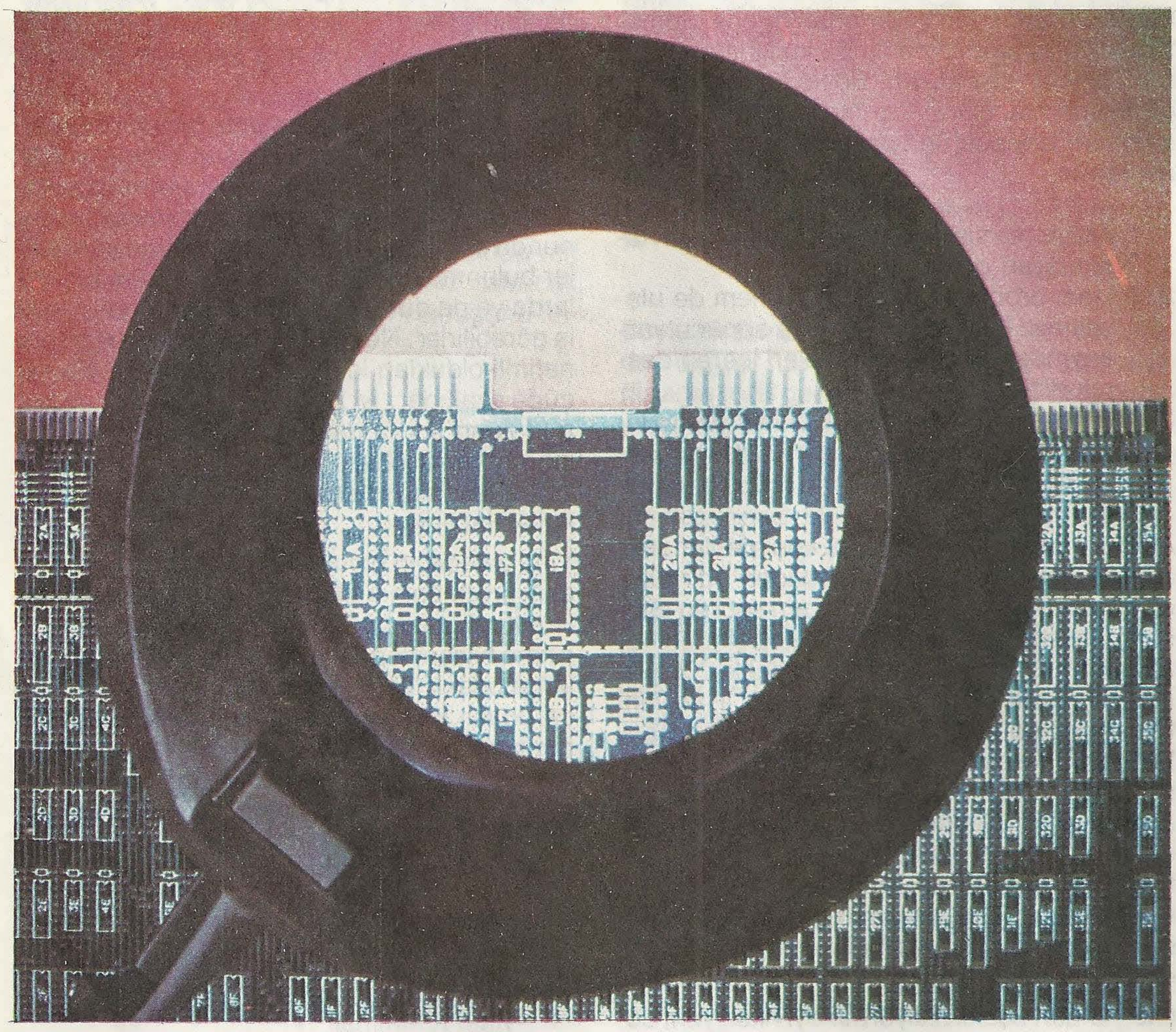
1 / İlkin bol miktarda bulunan algoritmalar, uzman sistemler tarafından kullanılabilir hale getirilmeli.
2 / İkinci görev gerçek uzmanlar tarafından kullanılmakta olan odaklı tekniklerin kodlanması. Bu göreve bilgi edinme adı veriliyor. Oldukça sorun çıkartabileceği için şimdilerde geliştirilmekte olan (ve eskiden geliştirilen) bilgi edinme araçları büyük önem taşıyacaktır.

Ancak odaklı tekniklerin yaygın uygulama alanlarının bulunduğu gerçeği, odaksız akılyürütmenin bir çok alanda önemli olduğu olgusunu gözardı ettirmemeli. Geleceğin uzman sistemlerinde stratejik ve odaksız akılyürütmenin kullanılamayacağı söylenemez. Birçok durumda odaksız akılyürütmenin son derece



Ücretsiz PDF Kitap Arsivi - https://rantey.org

üretici olduğu görülmüştür. Uzman sistemlerin yalnızca akılyürütmenin odaklı olduğu alanlarda kullanılması, birçok olanaklı kullanım alanını gözardı etmek demektir. Belki odaklı akılyürütmeye dayalı sistemler yaratmak daha kolaydır; hatta olası uygulamaların çoğu odaksız akılyürütme gerektirmeyebilir.



# Prolog: Bir Programlama Dili Olarak Mantık

Bundan önceki bölümlerde, çok değişik alanlardan otomatik akılyürütme programına problemleri nasıl aktarabileceğimizi gördünüz. Bütün durumlarda problem tanımı bir dizi tümceden oluşur. Bu bölümde tümcelerin daha farklı bir kullanımını inceleyeceğiz. İnceleyeceğimiz programlama dili PROLOG'dur. (Bu kelime -PROgramming for LOGic- ''Mantık İçin Programlama''nın ilk hecelerinden oluşmuştur.) Prolog'la yazılan programlar tümce biçiminde yazılmıştır. Notasyon biraz farklı olmakla birlikte, tümcelerin bu bağlamdaki anlamları diğer bölümlerdeki anlamlarıyla aynıdır.

Prolog programı tipik bir otomatik aklı yürütme programıyla temelde aynı notasyonu kullanmakla birlikte, iki programda tümcelerin kullanımı oldukça değişiktir. Bir prolog prog-



Ücretsiz PDF Kitap Arsivi - https://rantey.org

ramında tümceler, verili bir programı oldukça odaklanarak çözmekte kullanılan bir algoritmayı ifade etmek için kullanılırlar -bir çözümü pek az araştırmayla bulması istenen bir algoritmayı. Bunun tersine, tipik bir otomatik akıl yürütme programına verilen tümceler, çözüm için daha az odaklanmı bir arayışı başlatmakta kullanılırlar- daha az odaklanmış derken çözümü doğrudan hesaplayacak kesin bir algoritmanın kullanılmayışını kastediyoruz. Bu nedenle prolog bir algoritmanın ek bilgi toplamadan verili bir soruyu yanıtlamak üzere programlanabileceğini düşündüğümüz problemlerde kullanılabilir. Böyle bir algoritma bilinmediği zaman problem, gerekli kayıp bilginin araştırılması için bir akılyürütme programından yardım ister.

Bir Prolog programının etkililiği büyük ölçüde onu oluşturan tümcelerin düzenine bağlıdır, çünkü ilgi odağı bu düzene dayalıdır. Öte yandan genel bir akılyürütme programında, ilgi odağı kullanıcı tarafından sağlanan başka kriterlere, örneğin çeşitli kavramlara verilen ağırlıklar ve destek dizisine yerleştirilen tümceler gibi kriterlere dayalıdır. Her programlama dilinde olduğu gibi bir Prolog programındaki her adım tamı tamına dikte edilir; komutlar tümcelerin düzenine ve bu bölümde anlatılan yerleşik **(built in)** yapıya dayalıdır. Kuşkusuz otomatik akılyürütme programında atılan adımlar kesin komutlarla uyum içindedir; ama amaç, hedefe doğru son derece yönlenmiş bir yaklaşım uygulamak değildir. Prolog'da verili tümcelerle başlayan ve sorunun cevabı ile biten bir yol bulmayı umar hatta genellikle beklersiniz. Tipik bir akılyürütme programında da genellikle bir yol bulmayı ümid edersiniz; ancak soruyu yanıtlamadan önce bir çok yol araştırmayı beklersiniz. Bu nedenle size yardımcı olmak için bir Prolog programı seçmekle genel akılyürütme programı seçmek arasında karar vermek çeşitli faktörlere bağlıdır. Prolog'da çok daha hızlı bir uygulama potansiyeli vardır, bu potansiyel çeşitli problem sınıflarında kendini gösterir. Verili tümcelerle ifade edilen algoritma yaygın bir araştıtrmaya girmeye zorlandığında potansiyel gerçekleşmez.

Prolog'a verebileceğiniz problemlere bir örnek, iki tamsayının en büyük ortak bölenini hesaplama problemidir. Bu iş için FORTRAN programı yazabileceğiniz gibi Prolog programı da yazabilirsiniz. İstenen sonucu yeni bilgi için araştırma yapmaya gerek kalmaksızın hesaplayan bir algoritma mevcuttur. Kısaca belirttiğimiz gibi Prolog, diğer porgramlama dillerinin önermediği çeşitli özelliklere sahiptir; onu çekici kılan da budur zaten.

Prolog'un tek çıkarım mekanizmasının olumsuz aşırı çözüm (hyperresolution) kullanmak olduğunu düşünebilirsiniz. Ama Prolog'da olumsuz aşırı çözüme tekabül eden mekanizma (Prolog programı tarafından açıkca istenmediği takdirde) yeni tümceler eklemez.

Negatif aşırı çözümün amacı tüm yüklemlerinin önüne değil (NOT) sembolü gelen tümceler üretmektir. Böyle tümcelere negatif (olumsuz) tümceler adı verildiğini hatırlayınız. Olumlu bir tümce yalnızca olumlu yüklemler, karışık bir tümceyse hem olumlu hem olumsuz içerir. Olumsuz üst çözüm bir dizi olumsuz tümce uydu seçer ve bunlarla kullanılmak üzere olumlu ya da karışık bir tümce, bir çekirdek aramaya koyulur. çeşitli harfleri sonunda olumsuz bir tümce verecek şekilde birleştirecek değişken bir düzen çalışır.

Tipik bir akılyürütme programı ile Prolog arasında tümce dili ve çıkarım kuralı bağlantılarının yanısıra, strateji bağlantısı da mevcuttur. Bu programlama dilinin yaygın kullanımlarından birinde, bir soru sorulur. Soru olumsuz birim tümcei biçimini alır ve destek dizisine yerleştirilen tek tümcedir.

Yukarıda bahsettiğimiz bağlantılar yüzünden Prolog programını otomatik akılyürütmenin bir uygulaması olarak görebilirsiniz. Diğer bölümlerde bazı problem sınıflarının otomatik akılyürütme programıyla çözülüşüne odaklanmışken; burada hem kendisinin bir uygulaması hem de ''belirli'' nitelikte problemlerin çözümünde bir araç olan bir programlama diline odaklanacağız.

## Prolog'a Giriş

Prolog temsilin temel biçimi olarak tümceleri kullanan bir programlama dilidir. Diğer programalama dillerinde farkı mantıksl gücünün dilin içine ''yerleştirilmiş (built-in)'' olmasıdır. Özelde tümceler Prolog'a bilgi vermenin aracı oldukları için niceliği belirlenmiş değişkenleri içeren ifadeler kolayca yapılıp yönlendirilebilir.

Örneğin büyükanne-büyükbaba tanımı gerek tipik otomatik akılyürütme programına, gerekse Prolog'a kolayca iletilebilir; ancak notasyon farklıdır. İlki için şöyle yazarken

EBEVEYN(z,y) | EBEVEYN (x,z)
| EBEVEYN (x,Y)



Ücretsiz PDF Kitap Arsivi - https://rantey.org

ikincisini şöyle yazarsınız:

büyükebeveyn (X,Y):-ebeveyn (X,Z),
ebeveyn(Z,Y)

Daha önce bu iki tümceden ilkinin ''tüm z,y ve x için eğer z y'nin ebeveyni ve x z'nin ebeveyni ise o halde x y'nin büyükebeveynidir'' şeklinde okunacağını öğrendiniz. Prolog için ikinci tümce aynı şekilde okunabilir. Ya da bunu şöyle okuyabilirsiniz: ''x'in y'nin büyükebeveyni olduğunu göstermek için, öyle bir z bulun ki, x z'ni ebeveyni olsun, sonra z'nin y'nin ebeveyni olduğunu gösterin''. Şimdi eğer Tom'un Bob'un ebeveyni olduğunu biliyorsanız ve aynı zamanda Bob'un Linda'nın ebeveyni olduğunu da biliyorsanız çabucak Tom'un Linda'nı büyükebeveyni olduğunu hesaplayabilirsiniz.

Olgu:

EBEVEYN (Tom, Bob)

Olgu

EBEVEYN (Bob, Linda)

Sonuç

BÜYÜKEBEVEYN (Tom, Linda)

Eğer Prolog'a bu tanıdık tümceler ve büyükebeveyn tanımının ikinci versiyonu verilmiş ve büyükebeveynler konusunda uygun soru sorulmuş olsaydı, Prolog bunu hemen doğru olarak cevaplardı. ''Yerleşik'' **(built-in)** değişken işlenişi ve bir anlamda olumsuz aşırıçözüm çıkarsama kuralını anıştıran ''yerleşik'' tümce işleme yetisi sayesinde problem onun içn çok basit olacaktı.

Bu yerleşik özellikler diğer programlama dilleriyle önemli bir zıtlığa işaret ederler. Diğer dillerde, değişkenlerin ve ''büyükebeveyn'' türü üzerine genel bilginin işlenişlerini programlamak durumunda kalacaktınız. Bu diller size Prolog'un verdiği mantıksal gücü sağlamadıkları için, yukarıda verilen türden problemlerle karşılaşıldığında sizden çok daha fazla şey bekler. Garip olan Prolog'un size bu rahatlığı ve ek gücü sunmasına rağmen maliyetinin şaşırtıcı derecede ucuz olmasıdır. çeşitli yaygın kullanılan programlama dillerinin hızıyla neredeyse rekabet edebilecek bazı uygulamalar mevcuttur.

Prolog'a ilişkin bir kavrayışı, genelde otomatik akıl yürütmeye ilişkin kavrayışınızla bağlantılandırırken dikkat edeceğiniz ilk nokta, Prolog'la daha önceki bölümlerde verilen malzeme arasındaki notasyon farklılıkları olmalı. Prolog'da çeşitli temsil biçimleri kullanılmaktaysa da, biz en çok kabul görenini veriyoruz. Bunun değişkenleri, yüklemeleri ve fonksiyonları, hepsinden önemlisi de tümceleri temsil etmede bazı kolaylıkları vardır. Yukarıda büyük ebeveynlerin tanımı olarak verilen ilk tümcedense ikincisi daha yaygın olarak kullanılır.

büyükebeveyn (X,Y):-ebeveyn (X,Z), ebeveyn (Z,Y)

Böyle bir tümce bir yordamı tanımlar. Öte yandan olgular her zaman pozitif birim tümcelerdir ama tersi geçerli değildir. Yani olgu olmayan birim tümceler vardır ama hiç bir birim olmayan tümce olgu kabul edilmez. Bir Prolog programı olgu ve yordamlardan oluşur. Olguları değerlendirerek, yordamları da gerektiğinde yürüterek Prolog çeşitli soruları yanıtlayabilir.

Prolog'un nasıl çalıştığını görmek için, bir Prolog programını -insanlar arasındaki akrabalık ilişkilerine değin soruları yanıtlamakta kullanılabilecek bir programı- biçimlendiren olgu ve yordam dizilerini inceleyelim. Bu örnek için bizim verdiğimiz tüm olgular gerekli olmasa da, bu olguların oluşturduğu dizi Prolog'un kolayca başa çıktığı tipik durumu göstermektedir.

Çeşitli şekillerde akrabalık bağları olan 13 kişi var: Can, Sema, Leyla, Aysun, Ayşe, Zeynep ,Melike, Tarık, Bozkurt, Cevat, Deniz, Ethem ve Cemal. Armağan ve Tarık, Bozkurt, Cevat ve Aysun'un ebeveyni. Sema ve Bozkurt da Deniz ve Leyla'nın ebeveynleri. Ethem ve Melike, Cemal'in ebeveynleri. Aysun ve Cemal, Ayşe ve Zeynep'in ebeveynleri. Bu bilgi ya da olgular toplamı Prolog'da birim tümcelerle temsil edilir. Ayrıca, kimin kadın kimin erkek olduğunu belirtmemiz gerekiyor.

Yüklemler, fonksiyonlar ve sabitler küçük harf, değişkenler büyük harflerle başlarlar, tümceler noktayla sonlanırlar.

Şimdi Prolog'da programlanacak olan problemin büyük ebeveynlere ilişkin olduğunu varsayalım. Bu durumda büyükebeveyn olmayı tanımlamak gerekir ve bu tanım da aşağıda verilen tümcedir.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

(30) büyükebeveyn (X,Y): -ebeveyn (X,Z),
ebeveyn (Z,Y)

Söylediğimiz gibi 30 bir yordam tanımlar. Her yordamın kendisi ya bir tanımdır ya da bir değeri hesaplamak için gereken adımlardan oluşur. Bir yordamdaki her tümce tümcenin ''başı'' (**head**) adı verilen bir pozitif kökle başlar ve her yordam baş olarak aynı pozitif köke sahip yüklemleri içeren tümcelerden oluşur. Baş her zaman :- sembolünün solunda yer alır. :-'-nin sağındaki kökler her zaman negatif olarak düşünülür. Negatif köklere alt-hedef adı verilir. Pozitif kök ise üst hedeftir. Prolog'da amaç alt-hedefleri tatmin ederek hedefe ulaşmaktır.

EBEVEYNLİK İLİŞKİLERİ HAKKINDAKİ OLGULAR

```
1)  dişi(canan)                15)  ebeveyn(tarık,barış)
2)  dişi(sema)                 16)  ebeveyn(canan,cevat)
3)  dişi(leyla)                17)  ebeveyn(tarık,cevat)
4)  dişi(aysun)                18)  ebeveyn(canan,aysun)
5)  dişi(armağan)              19)  ebeveyn(tarık,aysun)
6)  dişi(zeynep)               20)  ebeveyn(sema,deniz)
7)  dişi(melike)               21)  ebeveyn(barış,deniz)
8)  erkek(tarık)               22)  ebeveyn(sema,leyla)
9)  erkek(tarık)               23)  ebeveyn(barış,leyla)
10) erkek(cevat)               24)  ebeveyn(melike,cemal)
11) erkek(deniz)               26)  ebeveyn(aysun,armağan)
12) erkek(ethem)               27)  ebeveyn(cemal,zeynep)
13) erkek(cemal)               28)  ebeveyn(aysun,zeynep)
14) ebeveyn(canan,barış)       29)  ebeveyn(cemal,zeynep)
```

Şimdi 30. tümcenin anlamını açıklayabiliriz:

X'in Y'nin ebeveyni olduğunu göstermek için önce öyle bir bul ki, X Z'nin ebeveyni olsun ve sonra Z'nin Y'nin ebeveyni olduğunu göster.

Bunu söylemenin daha mekanik bir yolu da şudur: X'in Y'nin büyükebeveyni olduğunu göstermek için, düzenlenmiş girdi tümcelerini tarayarak ilk alt hedefi tatmin edecek bir yol ara. Bizim ele aldığımız örnekte bul :-'nın sağ tarafındaki ilk kökle birleşen (olgular dizisinde yer alış sırasına göre) ilk olguyu seçmek anlamına gelir. Eğer varsa, bu birleşme Z'ye bir değer vermiş olur. Şimdi yine düzenli tümceler dizisinden giderek yarı değer verilmiş üçüncü kökle bir olguyu birleştirmeye çalışır.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Örnekte tümce 30 alttaki dört biçimden birine sahip olan soruyu yanıtlamakta kullanılabilir.

Tarık kimin ebeveynidir?
Leyla'nın büyükebeveyni kimdir?
Kim kimin büyükebeveynidir?
Armağan Leyla'nın büyükebeveyni midir?

Tümce 1'den 30'a kadar tümcelerden oluşan bir Prolog programını yukarıdaki sorulardan herhangi biriyle yürütebilirsiniz. Hangisini sorarsanız sorun program aynı şekilde yürüyecek ancak cevap herhalde farklı olacaktır. O halde bu sorulardan birini soralım ve bir Prolog programını yürütmek derken tam olarak ne kastedildiğini -Prolog'un tam olarak ne yaptığını- görelim.

Tarık kimin büyükebeveynidir?:büyükebeveyn (Tarık,Gc)

Bu soru sorulduğu zaman yürütülen Prolog programı önce olgularına bakarak cevabın verili olup olmadığını irdeler. Eğer programdaki olgulardan biri bu soruyu yanıtlıyorsa, program yanıtla beraber durur. Eğer yanıtlamıyorsa bir yordam başlar. Tümce 30 sahneye çıkar ve pozitif kök X yerine Tarık'ı koyarak kısmen değer verilmiş olur.

Söylediğimiz gibi soruyu yanıtlama hedefine ulaşmak için Prolog althedefleri soldan sağa çözmeye çalışır. İlk althedefi çözmek için Prolog programı olgulara kendisine verildikleri sırayla ilkinden başlayarak bakar. İkinci kökle, yani, :-'nın sağındaki ilk negatif kökle birleşen bir olgu aranmaktadır. Ancak olgu kısmen değer verilmiş ikinci kökle birleşmelidir çünkü X değişkeninin yerine Tarık konmuştur. (Değişkenlerin büyük harfle, sabit ve fonksiyonlarla yüklemlerinse küçük harfle başlamaları gerektiğini hatırlayın). Eğer böyle bir olgu bulunursa, Z değişkenine, bulunan olguyla birleşmenin ona dayattığı değer verilmiş olur. Bu olduğunda ilk althedef çözülmüş demektir. Sonra Prolog programı ikinci althedefi çözmeye, yani kısmî değer verilmiş olan ikinci negatif kökle birleşecek bir olgu aramaya girişir. İkinci kökteki Z argümanı (varsa), kendisine bir önceki althedef çözümünde dayatılmış olan değere sahiptir.

Eğer ikinci althedef verili kısıtlamalar içinde çözülemiyorsa, ilk althedef için başka bir çözüm aranır. Eğer yeni bir çözüm bulunursa, Prolog programı yine, bu kez yeni çözüm tarafından kısıtlanmış olan ikinci althedefi çözmeye girişir. İlk çözüm daha sonraki althedeflerin çözümüne otomatik olarak uygulanmadığı için, önceki althedefe ikinci bir çözüm arama yordamına **geriye izleme** adı verilir. Prolog althedefleri soldan sağa doğru sırayla çözmeye çalışmaya devam edecek, gerektikçe de tekrar tekrar geriye izleyecektir, ta ki althedefleri çözmek için tutarlı bir yol bulunamayıncaya kadar. (Pek görülmemekle birlikte bir programın sonsuza kadar geriye izlemesi mümkündür).

Örnek soru için Prolog ilk althedefi çözmek için tümce 15'i kullanacak, sonra ikinci althedefi çözmek için tümce 21'i kullanacaktır. Böylece bulunan cevap Tarık'ın Deniz'in büyükebeveyni olduğudur. Eğer sorulursa Prolog programı Tarık'un diğer torunlarını bulmak için devam edecektir.

Bu tanım size önceki bölümde incelediğimiz çıkarım mekanizmasını çağrıştırmıyor mu? Tümce 30, 15 ve 21 hep birlikte aşırıçözümlülük mekanizması kuralını uygluayarak şunu verirler:

BÜYÜKEBEVEYN (Tarık, Deniz)

Ancak bu noktada, Prolog'la programlamanın ne demek olduğunu merak etmeye başlamış olabilirsiniz.

Prolog ile programlamak için pozitif birim tümceleri olarak olguları tümce biçiminde temsil etmek erekir. Olguları kullanmak için -soruların yanıtlarını hesaplamak için- yordamlar uygun tümce dizilerini yazarak programlanır.

Spesifik bir yordam ya da alt-rutin için bütün tümceler özdeş yüklemli pozitif bir kökle başlamalıdır. Bir yordamdaki tümcelerin birim tümceler olması şart değildir. Son olarak tümceler sizin programın nasıl işlemesini arzu ettiğinize göre düzenlenebilir. Her programlama dilinde olduğu gibi düzenleme seçimi size kalmıştır. Tıpkı örneğin, FORTRAN'da kodu aynı sonucu üretecek şekilde birkaç yoldan düzenleyebildiğiniz gibi, bir Prolog programında da tümceleri düzenleyebilirsiniz. Düzenleme nè olacağını önbelirler, çünkü althedeflerin çözümünün ve özel bir althedefi çözmeye girişmenin düzeni önceden belirlenir.

Prolog'un içiçe yerleştirilmiş nitelikleriyle kısıtlandığınızı hissetseniz bile, tümcelerin yer-



Ücretsiz PDF Kitap Arsivi - https://rantey.org

leştiriliş düzenini seçme şansınız olduğunu hatırlayın. Daha önce de belirtildiği gibi bu programlama dili, değşkenleri ve tümceleri ele alış biçimiyle programcıya içe yerleşik mantıksal güç sağlar.

Diğer programlama dillerinde olduğu gibi Prolog da bir yordam ya da alt-rutinin diğerini çağırmasına izin verir. Bu, örneğin verili olgular dizisi bir alt-hedefi çözmekte yetersiz olduğunda gerçekleşir. Bu durumu göstermek için bazı ek yordamlara bir göz atalım:

**EBEVEYNLİK İLİŞKİLERİ HAKKINDAKİ OLGULAR (devam)**

```
(31) büyükanne (X,Y):-
       büyükebeveyn(X,Y), dişi(X),
(32) büyükbaba (X,Y):-
       büyükebeveyn(X,Y), erkek(X).
(33) baba(X,Y):-
       ebeveyn(X,Y),erkek(X),
(34) anne(X,Y):-
       ebeveyn(X,Y),dişi(X).
(35) kardeş(X,Y):-
       ne(X,Y),ebeveyn(E,Y),
       ebeveyn(E,Y).
(36) kuzen (X,Y):-
       dişi(X),ebeveyn(X,Z),
       kardeş(Z,Y).
(37) ne (X,Y):-X = Y,!, fail.
(38) ne (X,Y).
```

Tümce 31'den oluşan yordam, 1 ila 29 arasındaki tümcelerden olguları kullanarak kesinkes yürütülemez. İlk alt-hedef-tümce 31'in ilk negatif kökü-verili olgularla çözülemez. Bu althedefi çözmek için tümce 30 uygulanmalıdır. Diğer bir deyişle yordam 31, yordam 30'u harekete geçirir. Hangi şartlar altında tümce 31 yürütülecektir?

Tümce 31'den oluşan yordam, örneğin şu soruda devreye girecektir: Zeynep'in büyükannesi kimdir? Tümce 1-38'den oluşan bir Prolog programından bu soruyu sormak için

büyükanne (BA, Zeynep)

yazarsınız. Bu tümcede BA bir değişkendir ve Prolog'da gerektiği gibi bütün değişkenler büyük harfle yazılır ve büyük harfle yazılan tüm ifadeler değişken olmak durumundadır. Bu soru sorulduğu zaman tümce 31 araştırılacak tek yoldur çünkü başka hiçbir tümcede yüklem olarak büyükanne geçmemektedir. Prolog önce :-'nı sağ tarafındaki althedefi, tümce 31'in ilk negatif kökünü çözmeye girişir. Olgu 1-29 bu althedefi çözme niyetiyle sırayla incelenir. Bu olgu dizisindeki hiç bir olgu büyükebeveyn yüklemine sahip değildir bu nedenle Prolog bir diğer yordamı çağırarak althedefi çözmeye girişir.

Bir althedefi çözümeye girişirken yalnızca olgular değil, bir yordamdaki tüm tümceler de veriliş sıralarıyla ele alınırlar. Bu programda yalnızca tümce 30 değerlendirilecektir çünkü büyükebeveyn yüklemine sahip kökü bir tek o barındırmaktadır. Tümce 31'in ilk negatif köküne uyacak bir kök bulma çabasında tümce 30'u kullanmak, daha yaygın kullanılan programcılık dillerinde bir altrutin çağırmaya benzer. çözülecek spesifik althedef şudur:

büyükebeveyn (BA, Zeynep)

Çünkü,

büyükanne (BA, Zeynep)

sorusu tümce 31'in X ve Y değişkenlerine sırasıyla BA ve Zeynep değerlerini verir. Yordam 30 tümce 31'le çağrıldığında 30'un boş ya da ana hedefi olan X ve Y değişkenlerine sırasıyla BA ve Zeynep değerleri verir. Tümce 30, iki parametre alan bir altrutin olarak düşünülebilir. Bu çağrılışında altrutin 30, BA'ya ikinci parametre Zeynep'in ilk hesaplanan büyükebeveyninin adını verecektir.

Tümce 30 çağrıldığında çözülecek iki althedef sunar. Bunlar sırasıyla tümce 20 ve 18 tarafından çözülürler ve tümce 30'daki BA'ya bir sabit olan canan adı verilir. Prolog programının bu noktasında tümce 31'in ilk althedefi çözülür ve program bundan sonra ilkin çözümü tarafından sınırlanan ikinci althedefi çözmeye girişir. İkinci althedefte dişi (canan) tümce 1'le çözülür ve soru başarılı biçimde yanıtlanır.

Büyükebeveyn konumu, geri-izleme gerekliliğinin doğabileceği gözönünde bulundurularak, saklanır. Bu nedenle eğer Canan'ın büyükebeveyn olduğu olgusu programın tamamlanmasına yetmezse, program bu kez Tarık'ı deneyecektir. Zeynep'in büyükannesi sorulduğunda, Prolog programı geriye deneyecek başka seçenek kalmayıncaya kadar, bütün



Ücretsiz PDF Kitap Arsivi - https://rantey.org

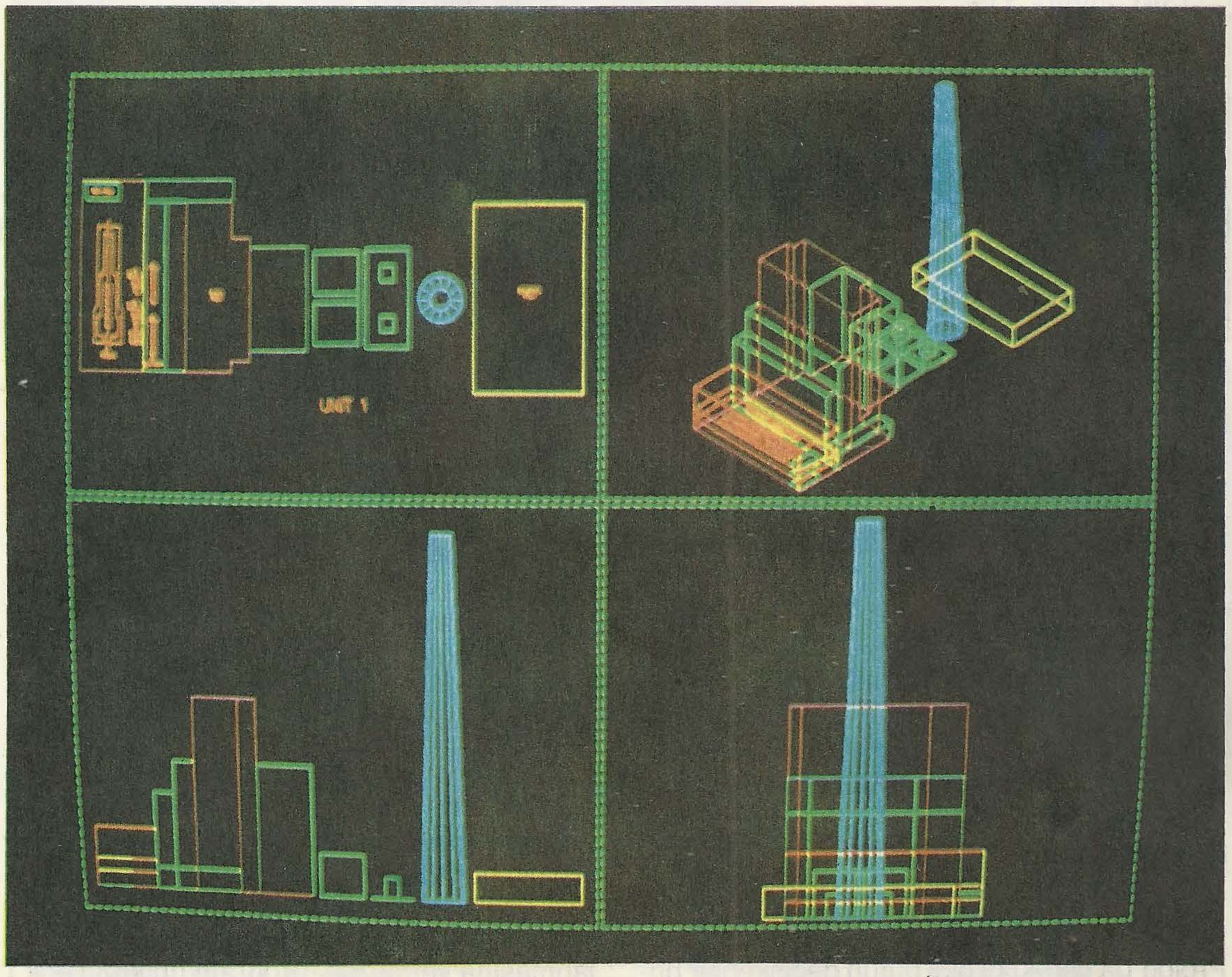
seçenekleri - yani, tümce 31'in iki alt hedefini birden aynı anda tatmin edebilecek bütün seçenekleri deneyecektir.

Bu, bir Prolog programının tipik işleyiş biçimidir: Bir yordam çağrılır ve yordamın alt hedeflerinin çözülebilirliği sınanır. Her biri için çözüm, verili olgular denenerek ve bunlar yetersiz kalırsa yeni yordamlar çağrılarak araştırılır. Eğer mevcut olgu ve yordamlar yetersiz bulunursa, program verili soruyu cevaplandıramadan durur.

## PROLOG VE DİĞER DİLLER

Şimdi Prolog'u diğer programlama dillerinden ayıran bazı ek farklılıklar üzerinde duracağız.

Eşitlik Prolog için yerleşik bir değerdir, yüklemler mantığı kullanılarak oluşturulması gerekmez.



**Prolog Japonların beşinci kuşak bilgisayarlar için tercih ettikleri dillerden biridir. Beşinci kuşak bilgisayarlar tasarısının önemli hedeflerinden biri bilgisayara komut ve verileri grafik imajlar şeklinde verebilmektir.**



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Bunun ne demek olduğunu tümce 37 temelinde görebiliriz. Tümce 37'nin ilk alt hedefi, yani

X = Y

X ve Y'nin verili değerleri birleştirilebildiğinde, otomatik olarak çözülmüş olur. Birleştirilemedikleri zaman ise, o althedef otomatik olarak hatalı sayılır. Tümce 37 ve 38, birlikte, iki terimin eşit olup olmadığını belirlemek için bir yordam oluştururlar. Önce tümce 37'nin ilk althedefinin çözümü olup olmadığı araştırılır. Eğer varsa, -yani X ve Y'nin değerleri aynıysa- o zaman tümce 37'nin ikinci althedefinin değerlendirilmesine geçilir.

İkinci althedefte gördüğümüz ! simgesi özel bir Prolog simgesidir ve tanımı gereği her zaman başarılıdır, yani her zaman çözülür. Burada Prolog'un normal geri izleme algoritmasının değişmesine yolaçan bir unsurla karşı karşıyayız.

! simgesine kesme (cut) simgesi adı verilir. Bununla karşılaştığında Prolog'un işlemleri hatırı sayılır derecede kısıtlanır. Örneğin, kesme simgesinden önce gelen althedeflere geri izleme yapılamaz. Daha sonra gelen althedefler için yapılacak geri izlemeler, kesme simgelerinin dayattığı sınırlamalar içerisinde yeralmak zorundadır. Ayrıca aynı yordamın bir parçası olan tümcelerin de hiçbiri incelenemez ki, bu tümce 37'de !'ye ulaşıldığı takdirde, tümce 38'in incelenemeyeceği anlamına gelir. Çünkü 37. ve 38. tümceler, her ikisinin de

# TEST 25

1 / Verilen olgularla, kuzen kavramını belirleyecek bir program yazın.
2 / Ebeveynler arasındaki ilişkilerle ilgili kurallarda 35. tümcenin ilk althedefi kaldırılacak olsa ne olurdu?
3 / Büyükanneyi belirleyecek alternatif bir yordam veriniz.
4 / Büyük-büyükebeveynleri hesaplayabilecek bir Prolog yordamı yazın.
5 / Birinin kardeşleri olup olmadığını belirleyebilecek bir Prolog yordamı verin. Özellikle

kardeş(p)

p'nin en az bir kardeşi olduğu durumda doğru olsun. Yani kardeş(p)'nin yalnızca hiçbir kardeşi olmadığı durumlarda çözümü olmasın/



Ücretsiz PDF Kitap Arsivi - https://rantey.org

baş kökü aynı yüklem olduğundan, aynı yordamın birer parçasıdırlar. (Bir yordamın, baş kökü aynı olan tümceler dizisi olarak tanımlandığını hatırlatırız.) Kısacası, bir !'ye ulaşıldığında, daha önceki kökler alternatif değerler verilemez hale gelir.

Bu kısıtlama, doğal olarak bu tümcelerin her zaman anlaşılabilir bir yorumu olup olmadığı, yordam içerisindeki işlevlerinin ne olduğu sorularını beraberinde getiriyor. Tümce 37'nin yalnızca yordamın işleyişi bağlamında bir anlamı vardır. Çünkü tekbaşına okunamaz, okunduğu takdirde hatalı çıkması kaçınılmazdır. Aslında 37. ve 38. tümceler birbirlerine bağlıdır ve bu bağlılık, programlanan algoritmanın bazı özellikleri bilindiğinde, oldukça yararlı olabilir.

Tümce 37'deki !'in bir özelliğinin otomatik olarak çözülebildiği (tatmin edilebildiği) olduğunu görmüştük. Onu izleyen üçüncü althedef olan ''fail'' (başarama!) ise yine özel bir Prolog simgesidir ve özelliği hiçbir zaman çözülememesidir. Bu nedenle X = Y althedefi başarılı olduğunda, yordam başarılı olmaz çünkü tümce 37(! simgesinden ötürü) incelenmeye devam edemez ve tümce çözülemezse, tümce 38 de incelenemez. Öte yandan, eğer tümce 37'nin ilk althedefi çözülemezse, tümce 38 değerlendirilecektir. Tümce 38'in althedefi olmadığı için otomatik olarak çözülür. Bu durumda sözkonusu terimler ''eşit değil'' olarak belirlenir.

Kesme (!) simgesi, özellikle bir koşul sınanmak istendiğinde ve özellikle bu koşulun ne zaman geçerli olmadığı bilindiğinde yararlı olur. Bu takdirde de şöyle bir yordam yazılabilir:

```
koşulgeçerli(PARAMETRELER):- koşuldeğil(PARAMETRELER),!,fail.
koşulgeçerli(PARAMETRELER).
```

## Prolog'un Sınırları ve Önemi:

Yapay zeka araştırmaları için Prolog'un en önemli özelliği kuşkusuz, olguları temsil ediş biçimi ile yüklemler mantığı arasındaki yakın ilişkidir. Yüklemler mantığı, bilgi işlem araştırmalarına destek olmak üzere değil, insan düşüncesinin geçerlilik koşullarını araştırmak ve temsil etmek üzere geliştirilmiş bir formel dil-



Ücretsiz PDF Kitap Arsivi - https://rantey.org

di. Bu bakımdan, amacı insan düşüncesinin benzetimini oluşturmak olan bir tasarı için, özel başka amaçlar için geliştirilmiş dillere kıyasla çok daha elverişli olacağı açıktır.

Ancak kuşkusuz Prolog da kusursuz bir dil değil. Teknik düzeydeki çeşitli sınırlılıklarından sözetmek mümkün. Bir örnek olarak, birleşme sırasında ''oluşum kontrolü'' (occurs check) yapamayışını verebiliriz. Bu, birleşme sırasında bir değişken için kullanılan terimin, o değişkeni gerçekten içerip içermediğinin kontrol edilememesi anlamına gelir. Bu yüzden, belirli koşullar altında sağlam olmayan bir çıkarım yapması, ilkece olanaksız değildir.

Yine de Prolog'un yakın gelecekte, teknik sınırlılıklarını aşmasını beklemek için çok neden var. Mantıkla olan yakın ilişkisinden olsa gerek, yapay zekanın gerçekleşmesinin ancak donanım teknolojisinde bir sıçrama ile mümkün olacağına inanan ve bu amaçla daha sonra anlatacağımız ''beşinci kuşak bilgisayarlar projesi''ni başlatan Japon endüstrisi, bu projede özellikle sistemlerinin akılyürütme bileşeni için Prolog'dan yararlanmaya karar verdiler. Üzerine bu kadar büyük bir yatırım yapılan bir programlama dilinin yakın gelecekte sınırlılıklarını aşacağına kesin gözüyle bakılabilir.

ÖZETLEYELİM:

**Bir Prolog programı bir dizi olgu ve yordamdan oluşur. Yordam, bir değer ya da değerler dizisinin nasıl hesaplanacağını belirleyen bir tümceler dizisidir. Bir yordamdaki bütün tümceleri baş (yani ilk) kökte aynı yükleme sahiptirler. Bir yordamdaki her tümce, yordamı yürütmek ve böylece değeri hesaplamak için soldan sağa doğru çözümlenmesi gereken bir dizi (bazı durumlarda, bu hiç olabilir) althedef sunar. Althedefler :-'in sağındaki köklerdir ve negatif olarak yorumlanırlar. Bütün althedefler çözüldüğünde, hesaplamanın başarılı olduğu söylenebilir.**

**Eğer althedeflerin tümünü aynı anda çözmeye ilişkin tüm çabalar başarısızlığa uğrarsa, yordam da başarısızlığa uğrar. Her althedefin çözümü, daha sonraki althedefin çözümleri üzerine bazı kısıtlamalar getirir. Daha sonraki bir althedef başarısızlığa uğradığında, Prolog geri izler - daha önceki althedef için başka bir çözüm dener.**

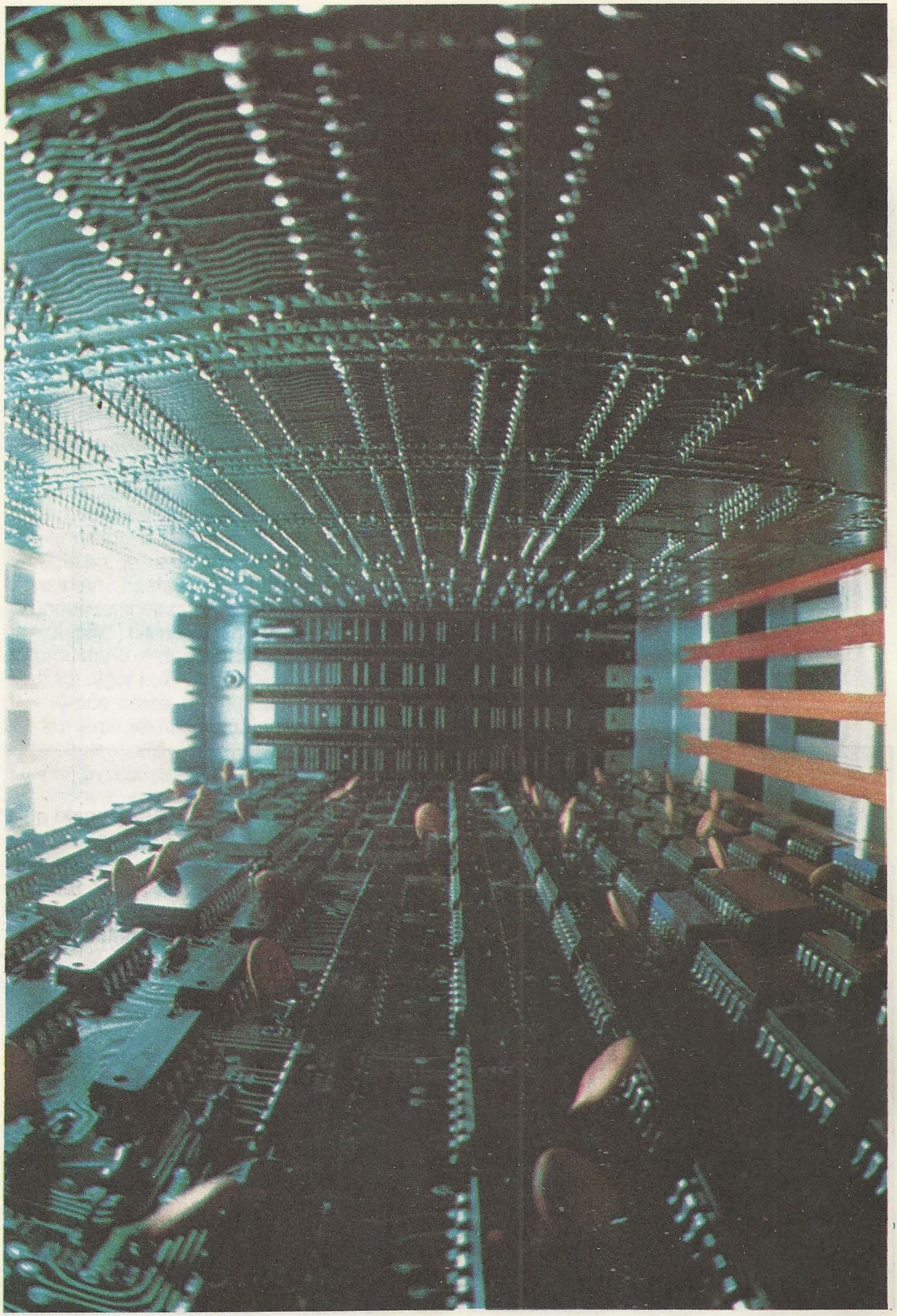
**Bir althedefin çözümünde mevcut olgular ya da başka yordamlar kullanılabilir. Bir althedefi çözmek için çağrılan olgu ve yordamlar, Prolog programında yeraldıkları sırayla kullanılmak zorundadırlar.**

**Prolog'daki bütün tümceler, programa sorulan sorular hariç, tam tamına bir pozitif kök içermek zorundadırlar (ne fazla ne eksik). Sorular negatif birim tümceler biçimini alırlar. Özel olarak istenmediği sürece bir Prolog programının yürütülmesi sırasında yeni tümceler yaratılmaz.**

**Bir Prolog programının mantıksal gücü hakkında bir fikir edinebilmek için, burada verilen örneği -bir insanın büyükebeveynlerinin kim olduğunu belirleme problemini- bir başka dilde programlamaya çalışmanızı öneririz.**



Ücretsiz PDF Kitap Arsivi - https://rantey.org



1750

Ücretsiz PDF Kitap Arsivi - https://rantey.org

# YAPAY ZEKANIN SINIRLARI

İnsani ve makanik bilgi işlem yöntemlerinin büyük ölçüde aynı temel işlem safhalarından oluştuğu iddiasının yapay zekanın temel iddialarından biri olduğunu görmüştük. Bu iddia, zeki olduğu iddia edilen ilk önemli sistemlerden biri olan GPS'nin (General Problem Solver-Genel problem çözücü) mucidi Newell ve Simon tarafından şu şekilde dile getirilmiştir:

''Yapay zeka yaklaşımı, gerek insan düşüncesinin gerekse sayısal bilgi işlemin ''genel amaçlı simge işleyen aletler'' oldukları ve bilgisayarların temel bilgi işlemi süreçlerini, beynin çalışmasını benzetleyebilecek şekilde programlanabilecekleri varsayımlarının ötesinde, bilgisayarların donanımıyla beynin benzer olduğu yolunda bir varsayım içermez''.

Fakat, bu kadarıyla da bu tartışma götürür bir varsayımdır. ''Genel amaçlı simge işleyen alet'' terimi ile ne kastedildiğini, ''temel bilgi işlemler''in neler olduğunu anlayabilmek için yine bilgisayarlara başvurmak zorundayız. Çünkü bütün yapay zeka iddiaları sayısal bilgisayarlarla temellendirildiği gibi, bugün için tasarlayabildiğimiz tek ''genel amaçlı simge işleyen alet''de onlardır.

Artık bildiğimiz gibi, bilgisayarlarda bilgi, ikili sistemde, yani bir dizi ''evet'' ve ''hayır'', elektrik anahtarlarının açık ya da kapalı olmasıyla temsil edilir. Makine, birbirlerine belirli birtakım kurallarla bağlı olan elemanlarının oluşturduğu bir kümeye göre çalışır. Bunları gözönünde bulundurarark, insanların da genel amaçlı simge işleyen aletler gibi davrandıkları iddiasının beraberinde şu varsayımları getirdiğini görürüz:

1/ Merkezi sinir sisteminin bilgiyi açık/kapalı anahtarlar sisteminin biyolojik karşılığı olan bir dizi soyut işlemlerle edindiği yolunda biyolojik bir varsayım.

2/ Beynin, formel kurallara göre bilgi elemanları üzerinde çalışan bir alet olarak görülebileceğini öne süren psikolojik bir varsayım.

3/ Her bilginin formelleştirilebileceği, yani anlaşılabilir her şeyin mantıksal ilişkiler, daha kesin olarak, Boole cebiri terimleriyle



Ücretsiz PDF Kitap Arsivi - https://rantey.org

elemanların ilişkilerini kurallara göre düzenleyen mantıksal işlemlerle açıklanabilirliğini iddia eden epistemolojik bir varsayım. Bu psikoloji ve felsefede tarihi oldukça eskilere giden ve oldukça yaygın kabul gören bir varsayımdır. Örneğin, bilgi elemanlarının izlenimler olduğunu iddia eden Hume gibi ampiristler de, bilgi elemanlarının bir kurallar sistemi olarak düşünülmesi gerektiğini iddia eden Kant gibi rasyonalistler de, bilginin bu şekilde (bilen) özneden bağımsız olarak formelleştirilebileceği varsayımını kabul etmişlerdir.

4/ Son olarak, sayısal bilgisayarlara verilen her türlü bilginin birbirinden bağımsız olarak tanımlanan elemanlar halinde olması gerektiğinden, insan aklını bilgisayarı temel alarak açıklayan model, akılcı davranış için gerekli olan her şeyin, dünyayla ilgili her türlü bilginin, ilkece içinde bulunduğu durumdan bağımsız olarak incelenebileceğini iddia eder. Gerçeklerin bilgisinin mantıksal olarak birbirlerinden bağımsız olarak temsil edilebileceği hakkında bu görüş, aslında onotolojik bir varsayımdır.

## BİYOLOJİK VARSAYIM

Beynin temel bilgi işlem sürecinin sayısal bir modelle anlaşılıp anlaşılamayacağı, ampirik bir sorudur. İlkece beyin, bilgiyi sayısal bir bilgisayardan çok farklı yöntemlerle de işleyebilir.

Örnek olarak bir olasılık verelim: Bilginin, bir direnç analogunun bir şebeke içerisindeki en kısa yolun bulunması problemini çözmesini andırır bir biçimde işlenmesi de ilkece olanaksız değildir. Dahası, merkezi sinir sisteminin, birtakım anahtarların açılıp kapanmasıyla temsil edilebilecek sayısal ilkelere göre çalışmadığını düşündüren gelişmeler de mevcuttur.

Modern bilgisayarların yaratıcılarından John von Neumann, bu konudaki kuşkularını, daha 1956'da şöyle dile getirmişti:

''İnsan sinir sistemi, insan eliyle yapılmış, bilinen her türlü aletten en azından 10*6 kez daha kapsamlı ve etkinlikleri de buna bağlı olarak çok daha çeşitli olan görkemli bir mekanizmadır.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Dış duyulardan gelen uyarıların, fiziksel, kimyasal hareketliliklerin yorumlanması, motor hareketliliklerin ve içsel kimyasal/hormonal düzeylerin denetimi, bilginin araştırılma ve dönüştürülmesinin son derece karmaşık süreçleriyle bellek ve tabii ki, kodlanan uyarıların ya da az çok nicel olan mesajların yerlerine ulaştırılması... bunların hepsi sinir sisteminin görevleridir. Elimizdeki sınırlı ve yetersiz bilgiler, insanın sinir sisteminin bu görevleri sayısal ilkelerden çok farklı ilke ve yöntemlerle yerine getirdiğini düşündürmektedir. Sinir sisteminde mesajlar, sayısal sinyallerle değil, sinyallerin birbirine eklenmesiyle oluşan zincirler aracılığıyla analojik olarak iletilir. Bu bağlamda şu tür unsurlardan sözedebiliriz: Bir sinir hücreleri hattındaki uyarıların zamansal yoğunluğu, bir kümedeki hatlarla uyarı zaman dizileri arasındaki karşılıklı ilişkiler vd.''.

# TEST 24'ün Çözümleri

**1** / Eğer ''Ahmet evde'' önermesini P, ''Ayşe mutlu'' önermesini Q, ''Ayşe çalışıyor'' önermesini de R ile simgelersek:

a. NOT P OR Q
Bunu gündelik dile, yaklaşık olarak çevirdiğimizde, ''Ahmet evde değil veya Ayşe mutlu'' cümlesini elde ederiz.

b. NOT (NOT P) OR R = P OR R
Bunun gündelik dildeki karşılığı ise, ''Ahmet evde veya Ayşe çalışıyor''.

---

**2** / Gündelik dilin cümlelerini yüklemler mantığına çevirirken ilk yapmanız gereken, kullanacağınız yüklemlerin ne olduğuna karar vermek. Hangi yüklemlerin kullanılacağı bir ölçüde tercihe bağlı olduğundan, birden fazla doğru cevap olabilir.

a. YAŞLI (Ahmet, Ali) OR ZENGİN (Ahmet, Ali)

b. SEVİYOR (Ahmet, Ayşe) OR SEVİYOR (Ahmet, Zeynep)



Ücretsiz PDF Kitap Arsivi - https://rantey.org

c. SEVİYOR (Ahmet, Ayşe) AND SEVİYOR (Ahmet, Zeynep)
d. MUTLU (Ahmet) AND BABA (Ahmet)

---

**3** / a. Cümleyi önce IF-THEN işlemcisini kullanarak yazıyoruz:

(x) (IF HAKEM(x) THEN İNATÇI(x))

Yorumu: Bütün x'ler için, eğer x hakemse, x inatçıdır.
IF-THEN'i OR'a çeviren kural gereğince,

(x) (NOT HAKEM(x) OR İNATÇI(x))

Yorumu: Bütün x'ler için, x hakem değildir ya da x inatçıdır.

Bizim kullandığımız notasyonda, tek başına kullanılan bir değişkenin tümel niceleyeninin kapsamı içindeymişçesine yorumlandığını gözönünde bulundurarak, tümceyi şöyle yalınlaştırabiliriz:

NOT HAKEM(x) OR İNATÇI(x)

b. NOT TAMSAYI(x) OR TEK(x) OR ÇİFT(x)

c. Bu soruyu cevaplamak için önce ''dar veya'' (XOR) bağlantısının, AND ve OR cinsinden nasıl ifade edileceğini görelim:

P XOR Q = ((P OR Q) AND (NOT(P AND Q)))

Bu eşdeğerliğin doğruluğunu doğruluk değerleri tablosu ile sınayalım:

| P | Q | P XOR Q | P OR Q | P AND Q | NOT (P AND Q) | (P OR Q) AND (NOT (P AND Q))) |
|---|---|---|---|---|---|---|
| D | D | Y | D | D | Y | Y |
| D | Y | D | D | Y | D | D |
| Y | D | D | D | Y | D | D |
| Y | Y | Y | Y | Y | Y | Y |

Demek ki, yukarıdaki tabloda son sütun cümlemizin önermeler mantığında ifadesidir. Bunu yüklemler mantığına çevirmek için önce tek tek önermeleri çevirelim.

''Bütün tamsayılar tektir'': NOT TAMSAYI(x) OR TEK(x)
''Bütün tamsayılar çifttir'': NOT TAMSAYI(x) OR ÇİFT(x)

(NOT TAMSAYI(x) OR TEK(X)) OR (NOT TAMSAYI(x) OR ÇİFT(x))
NOT((NOT TAMSAYI(x) OR TEK(X)) AND (NOT TAMSAYI(x) OR ÇİFT(x))

Dilerseniz, ''AND'in OR üzerinde dağıtılabilirliği kuralını kullanarak, bu kuralı daha da yalınlaştırabilirsiniz.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

c. NOT YAŞLI(x) OR HASTA(x)

d. NOT HASTA(x) OR YAŞLI(x)

e. NOT KURTULDU(x) OR NOT ERKEK(x)

---

**4** / Bu soruları cevaplandırmak için varoluşsal niceleyeni kullanabileceğimiz gibi, fonksiyonlara da başvurabiliriz. Biz aslen uzman sistemlerde yaygın olan fonksiyon kullanımını tercih edeceğiz, ama öğretici amaçlarla birinci sorunun cevabı için varoluşsal niceleyenle yazılmış bir tümce de önereceğiz.

a. NOT İNSAN(x) OR TORUNU(x,dedesi(x))

Yorumu: Bütün x'ler için, x insan değildir ya da x kendi dedesi olan birinin (fonksiyonun ifade etmekte olduğu budur) torunudur.

Bunu varoluşsal niceleyen kullanarak şöyle yazabilirdik:

(x) NOT İNSAN OR (∃y) (DEDESİ(y,x)

Yorumu: Bütün x'ler için, x insan değildir ya da öyle bir y vardır ki, o x'in dedesidir.

b. NOT TAMSAYI(x) OR NOT TAMSAYI(y) OR TAMSAYI(toplam(x,y)

Yorumu: Bütün x'ler ve y'ler için, ya x tamsayı değildir ya y tamsayı değildir ya da ikisinin toplamı olan bir sayı vardır ki, o da tamsayıdır.

---

**5** / Bu soruların yanıtları giderek güçleşiyor. Bu yüzden nihai sonuca varıncaya kadar geçilmesi gereken aşamaları, adım adım göstereceğiz.

a. Cümlenin mümkün çevirilerinden biri şudur:

IF(İŞEGİDER(Pazartesi, Ahmet) AND İŞEGİDER(Salı, Ayşe))
THEN İŞEGİDER(Çarşamba, ben)

Bu, şu şekle çevrilebilir:

NOT(İŞEGİDER(Pazartesi, Ahmet) AND İŞEGİDER(Salı, Ayşe))
OR İŞEGİDER(Çarşamba, ben)

Cümlenin son şekli ise şu olacaktır:

NOT İŞEGİDER(Pazartesi, Ahmet) OR NOT İŞEGİDER(Salı, Ayşe)
OR İŞEGİDER(Çarşamba, ben)



Ücretsiz PDF Kitap Arsivi - https://rantey.org

b. Cümlenin şu şekli,

(i) (x) IF(FUTBOLCU(x) AND GÜÇLÜ(x) AND HIZLI(xx) THEN DEĞERLİ(x)

aşağıdaki gibi yeniden yazılabilir:

(ii) NOT(FUTBOLCU(x) AND GÜÇLÜ(x) AND HIZLI(x)) OR DEĞERLİ(x)

Yukarıdaki tümcenin öncülünü önermeler mantığında şu şekilde yazabiliriz:

(iii) NOT(P AND Q AND R)

Bu tümceyi parantezlerle istediğimiz gibi böleriz, çünkü

(iv) P AND Q AND R = P AND(Q AND R) = (P AND Q) AND R.

O halde (iii)'ü şu şekilde bölelim:

(v) NOT(P AND(Q AND R))

Q AND R'ı tek bir önerme şeklinde düşünebiliriz.

(vi) Q AND R = S

(v)'te Q AND R'ın yerine S'i yerleştirerek (vii)'yi elde ederiz.

(vii) NOT(Q AND S)

Bu ise de Morgan kuralları gereğince (viii)'i verir:

(viii) NOT P OR NOT S

S'i açınca,

(ix) NOT P OR NOT(Q AND R)

Artık önermeler mantığı notasyonundan yüklemler mantığı notasyonuna geri dönebiliriz. Tümcemiz sonuç olarak şu şekli alacaktır:

(xi) NOT FUTBOLCU(x) OR NOT GÜÇLÜ(x) OR NOT HIZLI(x)
OR DEĞERLİ(x)

c. Şu tümce,

(i) IF İŞEGİDER(Pazartesi, Ahmet)
THEN (İŞEGİDER(Salı, ben) AND İŞEGİDER(Çarşamba, ben)

a şağıdaki tümce çiftine çevrilebilir:

(ii) NOT İŞEGİDER(Pazartesi, Ahmet) OR İŞEGİDER(Salı, ben)



Ücretsiz PDF Kitap Arsivi - https://rantey.org

(iii) NOT İŞEGİDER(Pazartesi, Ahmet) OR İŞEGİDER (Çarşamba, ben)

Alt alta yazılmış tümcelerin AND bağlantısı ile bağlanmış olduklarının farz-edildiğini hatırlatalım.

d. Şu tümce,

(i) IF (İŞEGİDER(Pazartesi, Ahmet) AND İŞEGİDER(Salı, Ayşe))
THEN (İŞEGİDER(Çarşamba, ben) AND İŞEGİDER(Perşembe, ben))

IF'İ OR cinsinden ifade eden kural gereğince şu şekli alır:

(ii) NOT (İŞEGİDER(Pazartesi, Ahmet) AND İŞEGİDER(Salı, Ayşe))
OR (İŞEGİDER(Çarşamba, ben) AND İŞEGİDER(Perşembe, ben)

(ii) ise yine bir önceki örnekte olduğu gibi bir çift tümceye dönüştürülebilir:

(iii) NOT İŞEGİDER(Pazartesi, Ahmet) OR NOT İŞEGİDER(Salı, Ayşe)
OR İŞEGİDER(Çarşamba, ben)

(iv) NOT İŞEGİDER(Pazartesi, Ahmet) OR NOT İŞEGİDER(Salı, Ayşe)
OR İŞEGİDER(Perşembe, ben)

e. Bu oldukça zor bir soru. Cümlenin şu iki önermeyi içerdiğini düşünebilirsiniz:

(i) Eğer bir güreşçi yorgunsa, karşılaşmayı kaybedecektir.

Ve

(ii) Eğer bir güreşçi güçsüzse, karşılaşmayı kaybedecektir.

Bunların her birini formelleştirmek kolay. Toplam tümcenin ise şu şekli alması gerekiyor:

(iii) IF(GÜREŞÇİ(x) AND (YORGUN(x) OR GÜÇSÜZ(x)))
THEN KAYBEDECEK(x,yarışma)

Ara adımların ve nihai biçimin çözümünü size bırakıyoruz.

Von Neumann ''canlı organizmaların karmaşık karakteri''nden ne anladığını ise şu şekilde açıklıyor:

''Nöron bir uyarıyı iletir... Sinir uyarısı temelde, ikili sistemdeki sayılar gibi, ya hep ya hiç olan bir durumu gösterir. Yani, gerçekten de bilgiye karşılık gelen sayısal bir unsur vardır. Fakat sayısal bir unsurun varlığı, her şeyin sayısal ilkelere göre açıklanabileceğini göstermez.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

... Organizmalarda bazıları sinirsel, dolayısıyla sayısal, bazılarıysa hormonal, dolayısıyla analog olan, yani ilk uyarıdan son tepkiye kadar çeşitli aşamalardan geçen, bileşik işlevsel dizilerin varlığı bilinen bir gerçektir.''

Fakat von Neumann'ın söyledikleri bile sayısal modelin önemini fazla abartıyor olabilir. Sinir uyarısının ya hep ya hiç türünden bir hareketlilik olduğu gerçeği, ille de sayısal bir işleme karşılık geldiğini göstermez. Sayısal ve analog (benzeşik) bilgi-işlem arasındaki ayrım mantıksal bir ayrımdır; donanım ya da sistem içerisindeki elektrik sinyallerinin tür ve niteliğinden bağımsızdır. Aradaki temel fark, sayısal bilgi-işlemde sistemin kullandığı enerji türünün her biriminin tanımlayıcı bir dil aracılığıyla bir bilgi elemanına tekabül ettirilmesi, analog bilgi-işlemde ise bilgiyi, sürekli fiziksel değişkenlerin temsil etmesidir. Beyin, ancak bilgi-işlem dizisindeki her sinyal bir simge ile ilişkiliyse, sayısal bir bilgisayar olarak görülebilir. Fakat eğer von Neumann'ın da iddiasında olduğu gibi, sinir sisteminin etkinliklerinde temel birim uyarıların iletilme hızı ise, o zaman beyin olsa olsa analog bir bilgisayar olarak çalışıyor olabilir.

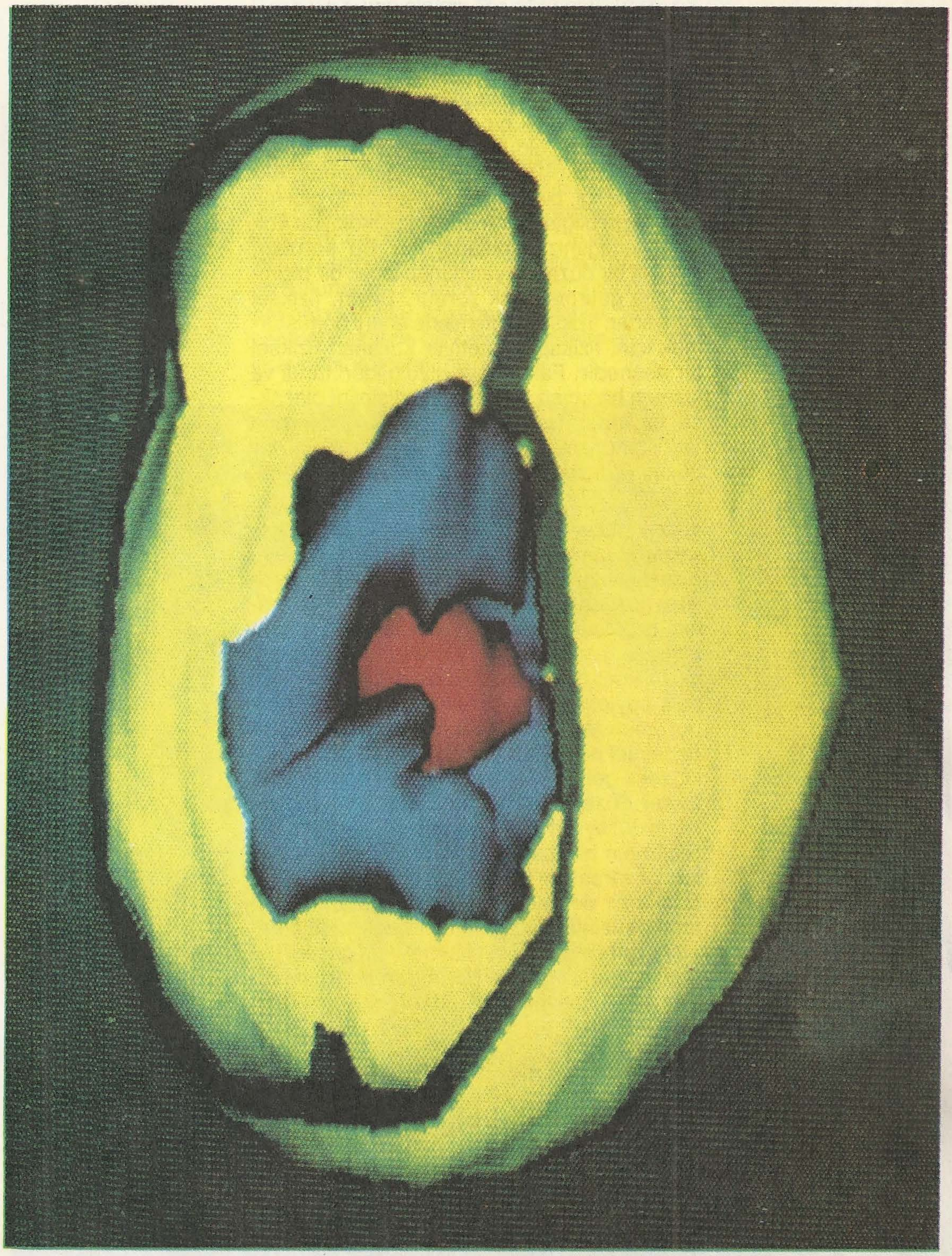
Nöropsikolojide bilgisayar kullanımının öncülerinden biri olan Walter Rosenblith durumu şöyle özetliyor:

''Sinirsel uyarıların ya hep ya hiç özelliği veri alınarak, bilgisayar komutlarının yerine getirilmesinin, nöronlar için bir model oluşturduğu yolunda yaygın görüşe katılmak mümkün değil. Bugün artık ilgimizi giderek, nöronlar arasındaki ilişkilere yöneltiyoruz: Bazı durumlarda bir dizi sinirsel uyarı binlerce nöronun hareketini yansıtıyor olabilir. Çok sayıda nöronun çok yakın etkileşimlerle birarada bulduğu bir sistemin işleyişinin, hepsi bağımsız birimler olarak kabul edilen nöronlar arasındaki ilişkiler temelinde açıklanabilmesi pek olası gözükmüyor. Bu yüzden bilgisayar sistemleri ile insan beyninin işleyiş yapıları arasında çok yakın bir paralellik kurma çabası, büyük bir olasılıkla düş kırıklığıyla sonuçlanacaktır.''

Bu durumda beynin genel amaçlı simge işleyen bir alet olarak sayısal bir bilgisayar gibi çalıştığı görüşü, ampirik bir hipotez olarak gününü doldurmuş gibi gözükmektedir. Gerçekten de beynin etkileşimli doğasıyla, mevcut bilgisayarların birbirinden daha bağımsız parçalardan oluşan düzeni arasındaki farklılık, yapay zeka iddialarını doğrulanmasının, hiç değilse biyolojik bakımdan pek olası olmadığını düşündürmektedir.



Ücretsiz PDF Kitap Arsivi - https://rantey.org



**Yapay zekâ iddialarına inandırıcılık kazandıran önemli faktörlerden biri, sinir sisteminin de sayısal ilkelere göre çalışıyor olabileceği görüşü idi. Ama sinir sisteminde bile, sinirsel sinyallerin iletilme hızının önemi gibi sayısal ilkelere göre davranmayan bir unsurun varlığı bir yana, bu görüş, insan davranışından belirleyici bir yere sahip olan hormonal sistemi tamamen gözardı etmiş oluyordu. Aşağıda resmi görülen pitüiter bezesinin de bir parçası olduğu hormonal sistemin salgıladığı hormanların miktarı ile orantılı olarak bazı davranışsal değişiklikler gözlemlenebilmektedir.**



Ücretsiz PDF Kitap Arsivi - https://rantey.org

## Psikolojik Varsayım

Beynin bir sayısal bilgisayar gibi çalışıp çalışmadığı, nörofizyoloji tarafından cevaplanacak tamamıyla ampirik bir sorudur. Ancak aşağıdaki ilişkili ama oldukça farklı sorulara, ampirik düzeyde kolay bir cevap bulunamaz: Aklın işlevselliği, bir sayısal bilgisayarınki gibi midir? Ya da bilgisayar modelini psikolojide kullanabileceğimiz kanıtlandı mı? Bu soruları, ampirik bir düzeyde yanıtlanabilecek bir biçimde dile getirmek çok zordur. Beyin, fiziksel dünyadan fiziksel yöntemlerle enerji dönüştürmek için, fiziksel yöntemler kullanan fiziksel bir nesnedir. Fakat eğer biyolojiden farklı ve kısmen bağımsız bir psikoloji disiplini olacaksa, beyindeki fiziksel-kimyasal tepkimelerden farklı bir düzey, davranışsal ya da zihinsel bir düzey ve bu düzeyin işlevselliği tanımlanabilmelidir.

Yapay zeka araştırmalarını yönlendiren kuramlara göre psikolojinin bu özgül düzeyi, bilgi-işlem düzeyidir ve bu düzeydeki işlevsellik aklın yaptığı kıyaslama, sınıflama, depolanmış bilgi listelerinde veri arama türünden işlemlerle tanımlanabilir. Ancak aklın bu düzeyi, fiziksel duzeyin tersine, farklı yorumlara açık söylemsel bir düzey olarak tanımlamak durumundadır. Bu nedenle buradaki sorun deneysel olmaktan öte, felsefidir. Göreceğiz ki, zeka gerektiren davranışların berisinde fiziksel düzeyden kısmen bağımsız bir bilgi-işlem düzeyinin yattığı varsayımı, düşünce süreçlerinin bilgisayar tarafından taklit edilebileceği varsayımı kadar açık bir iddia değildir. Başka bir deyişle, zeka gerektiren davranışların berisinde süregiden bir bilgi-işlem sürecinin yattığından kuşkulanmak için geçerli nedenler vardır. Bu ise doğrudan doğruya yapay zeka iddialarını ilgilendiren bir kuşkudur.

## Bir Örnek: Algılama ve Tanıma

Zihinsel herhangi bir olguyu inceleyebilmek için öncelikle o olguyu diğerlerinden ayırdedebilmek ve daha da önemlisi, o zihinsel olgunun ilgili olduğu nesne ve olayların tanımını keşfedebilmek gerekir. Yapay zeka iddialarına göre bu tanımların, sözkonusu olay ve nesnelere ulaşılmasını mümkün kılacak bir dizi komut biçimini alması gerekiyor. Ancak somut bir örnek sözkonusu olduğunda, ''tam bir tanımın bir dizi komut biçimini alması gerektiği'' iddiasının ne anlama geldiği çok açık olmayabilir.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Basit bir örnek alalım: Bir sürü geometrik şekil ortasından, verilen belirli bir komuta göre, kırmızı bir kare seçme davranışını düşünelim. Yapay zeka iddialarına göre bu davranışın tam bir tanımı, bir komutlar dizisi, bu görevi yerine getirirken izlenecek bir plan olacaktır.

Ancak bu şekilde davranmakta olan bir insanı gözümüzün önüne getirmeye çalışalım. Bu işte başarılı olması için ona vermemiz gereken komutlar ne olabilir? Herhalde ''komutları dinle!'', ''nesnelere bak!'', ''şekillerini gözönünde bulundur!'', ''seçimini yap!'' gibi çok genel bazı kurallar sözkonusu olabilir. Ama bir çemberi değil de bir kareyi ayırdetmeye yönelik ayrıntılı komutlar ne olabilir?



**Yapay zekâ iddialarına göre, zekâ bir bilgi-işlem programı olarak düşünülebilir. Bu bakımdan yapısal ve işlevsel özellikleri bakımından donanımdan, yani insan sözkonusu olduğundan gövdeden tamamen bağımsızdır. Oysa alternatif bir görüşe göre, insanın ana rahminden başlayarak çevresiyle kurduğu nedensel ilişkilerin tamamı, zekâ için belirleyicidir.**



Ücretsiz PDF Kitap Arsivi - https://rantey.org

## Bilgisayarlar Düşünebilir mi?

*Bugün beynin nasıl çalıştığını ayrıntılı olarak bilmemekle birlikte, beyin süreçleriyle zihinsel süreçler arasındaki genel ilişkiler üzerine bir fikrimiz olacak kadar bilgi sahibiyiz. Zihinsel süreçlere, beynin davranış öğeleri yol açar. Aynı zamanda bu öğelerin oluşturduğu yapı içinde gerçeklenirler. Sanırım bu yanıt, biyolojik fenomenlere getirilen standart biyolojik yaklaşımlarla uyum içindedir. Aslında bu, dünyanın nasıl çalıştığı hakkındaki bilgilerimiz gözönünde tutulursa, soruna getirilen sağduyuya uygun bir yanıttır. Yine de azınlığın görüşüdür. Felsefe, psikoloji ve yapay zeka alanında hüküm süren görüş ise, insan beyniyle sayısal bilgisayarların işleyişi arasında analojiler kurmaktır. Bu görüşün en aşırı versiyonuna göre insan beyni yalnızca sayısal bir bilgisayar, zihinse yalnızca bilgisayar programıdır. Benim ''aşırı yapay zeka'' ya da ''aşırı YZ'' diye adlandırdığım bu görüş, zihnin beyin için anlamının, programın bilgisayar donanımı için anlamıyla aynı olduğu söylenerek özetlenebilir.*

*Bu görüş, insan zihninin özde biyolojik bir yönü olmadığı neticesini verir. Beyin, insan zekasını oluşturan programları taşıyabilecek sonsuz sayıdaki olası bilgisayar donanımlarından yalnızca biridir. Görüşe göre, doğru girdi ve çıktıları içeren doğru programa sahip herhangi bir fiziksel sistem, tıpkı sizinki ve benimki gibi bir zihne sahip olacaktır. Böylece örneğin rüzgar değirmeniyle çalıştırılan eski bira kutularından bir bilgisayar yaptığınızda, doğru programı varsa, bir zihni olacaktır. Üstelik de önemli olan, bildiğimiz kadarıyla düşünce ve duygularının da olabileceği değil, duygu ve düşüncelerinin olması gerektiğidir; çünkü duygu ve düşünce sahibi olmanın tek koşulu budur: doğru programın yürütülmesi.*

*Bu görüşü savunanların çoğu, henüz zihne tekabül eden programlar tasarımlamamış olduğumuzu düşünürler. Ama genelde, bilgisayar bilimcileri ve yapay zeka üzerine çalışanların, insan beyin ve zihnine eşit bilgisayar donanımı ve program tasarımlamalarının yalnızca zaman sorunu olduğu konusunda fikir birliğine varmış gibidirler. Bu yapay zekalar ve zihinler insan beyin ve zekasına her açıdan özdeş olacaktır.*

*Yapay zeka alanının dışındaki birçok kişi, böyle bir görüşe inanılabilmesini hayretle karşılıyorlar. Bu nedenle görüşün eleştirisine geçmeden önce, bu alandaki insanların gerçekten söylemiş olduklarına birçok örnek vereyim. Carnegie-Mellon Üniversitesi'nden Herbert Simon, kelimenin tam anlamıyla düşünebilen makinelere zaten sahip olduğumuzu söylüyor. Gelecekteki bir makineyi bekleme sorunu yok; çünkü varolan sayısal bilgisayarlar, tıpkı sizinkine ve benimkine benzeyen düşüncelere sahip. Buna ne dersiniz! Filozoflar yüzyıllardır makinelerin düşünüp düşünemeyecekleri konusunu dert edip durdular, şimdiyse böyle makinelerin Carnegie-Melon'da zaten olduğunu öğrenmiş bulunuyoruz. Simon'un iş arkadaşı Alan Newell, zekanın yalnızca bir simge işleme sorunu olduğunu; herhangi spesifik türden biyolojik ya da fiziksel donanımla özsel bir bağlantısı olmadığını keşfettiğimizi (dikkat edin, Newel ''keşfettiğimiz'' diyor; ''farzediyoruz'' ya da ''olanaklılığını değerlendiriyoruz'' değil **keşfediyoruz)** öne sürüyor. Yani, fiziksel simgeleri doğru şekilde işleyebilen herhangi bir sistem, insanlarınkiyle aynı anlamda zeka yetisine sahiptir. Simon da, Newel de, kendilerine göre bu iddialarında mecazî bir yan olmadığını vurguluyorlar; bunları gerçekten kasdediyorlar. Freeman Dyson'un, evrim açısından bakıldığında bilgisayarların bize oranla avantajlı olduklarını söylediği belirtilmiştir. Bilinçlilik yalnızca formel süreçlere ilişkin bir sorun olduğuna göre, bilgisayarlarda bu süreçler, soğumakta olan bir evrende bizim gibi ıslak ve karmakarışık malzemeler yerine daha iyi dayanabilecek bir maddenin içinde bulunacaklarından, süregideceklerdir. MIT'den Marvin Minsky gelecek kuşak bilgisayarların zekalarının o kadar yüksek olacağını ki, ''eğer bizi evcil hayvan olarak evde tutmak isterlerse kendimizi şanslı addetmemiz gerektiğini'' söylüyor. Sayısal bilgisayarlar üzerine abartılmış iddialardan en çok bayıldığım da ''yapay zeka'' teriminin mucidi John McCarthy'ninki. McCarthy, ''termostatlar kadar basit makinelerin bile inançları olduğu söylenebilir'' diyor. Hatta ona göre problem çözme yetisine sahip her makinenin anançları olduğundan söz edilebilir. McCarthy'nin cesaretine hayranım. Bir keresinde ona şunu sordum: ''Termostatının ne gibi inançları var?'' Şöyle cevap verdi: ''Termos-*



Ücretsiz PDF Kitap Arsivi - https://rantey.org

tatımın üç inancı var: burası çok sıcak, burası çok soğuk ve burası çok iyi''. Bir filozof olarak bu iddiaları basit bir nedenden ötürü seviyorum. Birçok felsefi tezin aksine, oldukça netler ve basit ve kesin bir çürütmeye izin veriyorlar. İşte bu bölümde deneyeceğim de böyle bir çürütme.

## Bilgisayar Simgelerinin Formelliği-Zihinsel Süreçlerin Anlamlılığı

Çürütmenin niteliğinin bilgisayar teknolojisinin herhangi bir aşamasıyla hiçbir ilgisi yok. Bu noktayı vurgulamak önemli; çünkü genel eğilim problemlerimizin çözümünün henüz yaratılmamış bir teknolojik harikayı beklemek durumunda olduğu yolunda. Ama gerçekte çürütmenin niteliği teknolojinin herhangi bir aşamasından tümüyle bağımsız. Sayısal bilgisayarın tanımının kendisine; sayısal bilgisayarın ne olduğuna ilişkin.

Bizim sayısal bilgisayar kavrayışımıza göre işlemlerin tümüyle formel olarak belirlenebilmesi çanalıcı bir nokta; yani biz bilgisayarın işlemlerini oluşturan adımları soyut simgelerle belirliyoruz -örneğin manyetik bir banda yazılmış sıfır ve birler dizisi. Tipik bir bilgisayar ''kuralı'', makine özel bir durumda olduğunda ve bandında özel bir simge bulunduğunda, simgeyi silmek ya da başka bir simge yazmak ve bandı bir kare sola kaydırmak türünden başka bir duruma girmek gibi belli bir işlem yapacağını belirler. Ama simgelerin hiçbir anlamı, semantik içeriği yoktur; birşeye ilişkin değildir. Salt kendi formel ya da sentaktik yapılarının terimleriyle belirtilmeleri gerekir. Örneğin sıfırlar ve birler yalnızca rakamlardır, numara yerine bile geçmez, birşeyi saymazlar. Hatta sayısal bilgisayarları güçlü kılan onların tam da bu özelliğidir. Tek ve aynı tip donanım, uygun şekilde tasarımlandığı sürece, belirsiz bir ranjda farklı programların işletilmesinde kullanılabilir. Ve tek ve aynı program, belirsiz sayıda farklı donanımda çalıştırılabilir.

Ancak programların salt formel ya da sentaktik olarak tanımlanıyor olma nitelikleri, zihinsel süreçlerle program süreçlerinin aynı olduğu görüşüne bir darbe indirir. Bunun nedeni basitçe belirtilebilir. Zihnin formel ya da sentaktik süreç olmanın ötesinde özellikleri de vardır. İçsel zihinsel durumlarımızın, tanımı gereği belli tür içerikleri vardır. Eğer Kansas City hakkında düşünüyor ya da soğuk bir bira istiyorsam veya faiz oranlarında bir düşüş olup olmayacağını merak ediyorsam, her durumda zihinsel durumum, olası formel özelliklerin yanısıra belli zihinsel içeriklere de sahiptir. Yani düşüncelerim bana simge şeritleri halinde gelse de, düşüncede soyut şeritlerden öte bir yan vardır; çünkü şeritlerin kendilerinin hiçbir anlamı olamaz. Eğer düşüncelerim herhangi birşey **hakkında** olacaksa, o halde şeritlerin düşünceleri o şey hakkında kılacak bir **anlamlarının** bulunması gerekir. Tek kelimeyle, zihnin sentakstan öte bir yanı, semantiği vardır. Hiçbir bilgisayar programının zihin olamamasının nedeni, bilgisayar programlarının yalnızca sentaktik, zihinlerinse sentaktikten öte birşeyler olmalarıdır. Zihinler semantiktir, formel bir yapının ötesine geçmeleri anlamında bir içerikleri vardır.

## Çin Odası Deneyi

Bu noktayı açıklamak için belli bir düşünce deneyimi tasarımladım. Bir grup bilgisayar programcısının bilgisayarın Çince anlamayı benzetlemesini sağlayacak bir program yazdıklarını düşünün. Böylece, örneğin bilgisayara Çince bir soru verildiğinde soruyu belleğiyle ya da veri tabanıyla karşılaştıracak ve sorulara Çince uygun karşılıklar verecektir. Şimdi, bu işlem temelinde, bilgisayar Çince anlıyor mu; kelimenin tam anlamıyla, Çince konuşanların Çince anladıkları anlamda anlıyor mu? Eh, bir odaya kilitlenmiş olduğunuzu ve odada Çince simgelerle dolu sepetler olduğunu tahayyül edin. Kendinizin de (benim gibi) tek kelime Çince anlamadığınızı, ama elinize bu Çince simgeleri nasıl işleyeceğinizi gösteren Türkçe bir kural kitabı verildiğini düşünün. Kurallar simgelerin işlenişini salt formel olarak, semantikleri değil, sentaksları açısından tanımlıyorlar. Bu nedenle kural şöyle diyebilir:

''Bir numaralı sepetten o eğri-büğrü simgeyi alın ve onu iki numaralı sepetteki büğrü-eğri simgenin yanına koyun.'' Şimdi başka Çince simgelerin odaya sokulduğunu ve bu simgeleri odadan çıkartmak için başka kurallar verildiğini farzedin. Sizin bilginiz dışında, odaya giren simgelerin odanın dışındakiler tarafından



Ücretsiz PDF Kitap Arsivi - https://rantey.org

''sorular'' olarak adlandırıldığını, sizin dışarı çıkarttıklarınıza ise ''soruların yanıtları'' adı verildiğini farzedin. Üstelik farzedin ki programcılar programları tasarımlamakta o kadar usta, siz de simgeleri işlemekte o kadar beceriklisiniz ki, çok kısa sürede cevaplarınız ana dili Çince olan birisininkinden ayırt edilemez bir hale geliyor. Orada odanızda kilitli olarak Çince simgeler karıştırıyor ve içeriye giren Çince simgelere karşılık dışarıya Çince simgeler yolluyorsunuz. Tanımladığım şekliyle bu durum temelinde, yalnızca formel simgeleri işleyerek Çince öğrenmenize imkan yoktur.

Öykümün ana fikri şu: bir dış gözlemcinin bakış açısına göre formel bir bilgisayar programını yürütmek yoluyla, tıpkı Çince anlıyormuş gibi davranıyorsunuz ama aynı zamanda tek kelime Çince anlamıyorsunuz. Ancak Çince anlamak için uygun bir bilgisayar programını taramak **sizin** Çince'yi anlamanıza yetmiyorsa, **hiçbir başka sayısal bilgisayar**ın anlamasına da yetmez. Ve yine bunun nedeni oldukça basit ifade edilebilir. Eğer siz Çince anlamıyorsanız başka hiçbir bilgisayar Çince anlayamaz çünkü hiçbir sayısal bilgisayarın programı yürütürken sizden öte bir özelliği yoktur. Bilgisayarın bütün sahip olduğu da, tıpkı sizin gibi, yorumlanmamış Çince simgeleri işlemek için bir programdır. Tekrarlarsak, bilgisayarın sentaksı vardır; semantiği değil.

Çince odasına ilişkin meselin tüm amacı, bize zaten bildiğimiz bir gerçeği hatırlatmaktır. Bir dili anlamak, hatta zihinsel bir duruma sahip olmak demek yalnızca bir formel simgeye sahip olmaktan öte birşeydir. Bir yorum, ya da bu simgelere ilişkin bir anlama sahip olmayı içerir. Ve tanımlandığı kadarıyla sayısal bir bilgisayar salt formel simgelerden başkasına sahip değildir; çünkü daha önce de söylediğim gibi bir bilgisayarın işlemleri, program yürütmesine ilişkin yetisi temelinde tanımlanır. Bu programlarsa salt formel olarak tayin edilebilir -yani semantik bir içerikleri yoktur.

Bu argümanın gücünü görmek için, Türkçe sorular sorulup yanıtlar vermekle, sözcüklerin anlamı üzerine hiçbir bilgimiz olmayan bir dilde sorular sorulup yanıtlamak arasındaki farkı düşünmemiz yeterlidir. Çince odasında Türkçe olarak yaşınız veya yaşam öykünüz gibi şeyler üzerine sorular sorulduğunu ve bunları yanıtladığınızı tahayyül edin. Çince vakasıyla Türkçe vakası arasındaki fark nedir? Eh, eğer siz de benim gibi hiç Çince anlamıyor ve Türkçe'yi ise anlıyorsunuz; çünkü sizin için bilinen anlamları olan simgelerle ifade ediliyor. Aynı şekilde Türkçe cevaplar verirken sizin için anlamlı olan simgeler üretiyorsunuz. Ama Çince vakasında bunlardan hiçbiri yok. Yalnızca bir bilgisayar programına göre formel simgeleri idare ediyor ve öğelerin herhangi birine anlam atfetmiyorsunuz.

Bu argümana yapay zeka ve psikoloji alanlarının yanısıra felsefeden de çeşitli yanıtlar gelmiştir. Hepsinin tek bir ortak özelliği var: hepsi de yetersiz. Neden yetersiz olmak durumunda kaldıklarının da apaçık bir nedeni var; çünkü argüman yalnızca sentaksın semantik için yeterli olmadığı basit mantıksal gerçeğine dayalıdır ve sayısal bilgisayarların, bilgisayar oldukları sürece, tanımları gereği yalnızca sentaksları vardır.

## Nasıl Bir Makine?

Bana karşı sık sık getirilen argümanları değerlendirerek bu konuya açıklık getirmek istiyorum.

Bazı insanlar Çince odası örneğine, sistemin tümünün Çince anladığını söyleyerek karşı çıkıyorlar. Buradaki fikir benim, simgeleri idare eden odadakı insanın Çince anlamadığı ama bu insanın yalnızca bilgisayarın merkezi işlem birimi olduğum. Onlara göre toplam olarak oda, simgelerle dolu sepet, programları içeren defterler ve belki de başka kalemlerden oluşan sistemin kendisi Çince anlamaktadır. Ama bu da benim yapmış olduğum itirazla karşı karşıya kalır yine. Sistemin sentakstan semantiğe geçmesi için bir yol yoktur. Merkezi işlem birimi olarak benim bu simgelerin herhangi birinin ne anlama geldiğini anlama imkanım yoktur; keza sistemin de.

Bir diğer yaygın karşı çıkış, bir robotun içine Çince anlayan bir program konulduğunu tahayyül etmektedir. Eğer robot etrafta dolaşıp dünyayla nedensel ilişkiler kuruyorsa, bu onun Çince anladığını göstermeye yetmez mi? Bir kez daha semantik-sentaks ayrımının amansızlığı bu manevrayı atlatır. Robotun beyin yerine yalnızca bir bilgisayara sahip olduğunu varsaydığımız sürece, tıpkı Çince anlı-



Ücretsiz PDF Kitap Arsivi - https://rantey.org

*yormuş gibi davranabilmekle birlikte, sentakstan Çince semantiğine ulaşabilmesi için hiçbir yol yoktur. Eğer benim bilgisayar olduğumu tahayyül ederseniz bunu görebilirsiniz.*

*Robotun kafasının içinde durup, bazılarının bana robotun başına bağlı televizyon kameralarından geldiğini ve diğerlerinin de robotun ellerini ve ayaklarını hareket ettirmek üzere dışarı çıktığını bilmeden sembolleri karıştırıp dağıtıyorum. Elimdeki formel bir bilgisayar programından ibaret olduğu sürece, bu sembollere herhangi bir anlam atfetmemin imkanı yok. Ve robotun dış dünyayla nedensel etkileşimlere girmekte olduğu gerçeği, eğer bu gerçeği bulgulamak için bir yolum yoksa, benim simgelere bir anlam atfetmemi sağlamayacaktır. Varsayalım ki robot bir hamburgeri eline aldı ve bu, hamburger simgesinin odaya girmesine yolaçtı. Nedenleri ya da oraya nasıl girdiği hakkında bir fikrim olmadan yalnızca simgeye sahip olduğumda, anlamını bilebilmem için elimde hiçbir yol yoktur. Robotla dünyanın geri kalanı arasındaki nedensel etkileşimler, şu ya da bu zihinde temsil edilmedikleri sürece sözkonusu değildir. Ancak zihin adı verilen şey yalnızca salt formel, sentaktik işlemlerden oluştuğu sürece de sözkonusu olabilmesi mümkün değildir.*

*Benim argümanımda neyin iddia edilip neyin edilmediğini tam olarak anlamak oldukça önemli. Benim yukarıda sözünü ettiğim soruyu sorduğumuzu varsayalım: ''Bir makine düşünebilir mi?'' Eh, kuşkusuz bir anlamda hepimiz makineyiz. Kafamızın içindeki nesneyi etten makine olarak yorumlayabiliriz. Ve kuşkusuz, hepimiz de düşünebiliyoruz. Bu nedenle, ''makine''nin bir anlamında, yani makinenin belli işlem türlerini yürütme yetisine sahip fiziksel bir sistem olduğu anlamda, hepimiz makineyiz ve hepimiz düşünebiliriz. Bu nedenle düşünebilir makineler vardır. Ama bizi rahatsız eden soru bu değildi. O halde başka bir formülasyon deneyelim. İnsan eliyle yapılmış bir şey (artefact) düşünebilir mi? İnsan yapımı bir makine düşünebilir mi? Burada da yanıt yine yapıtın türüne bağlıdır. Moleküllerine kadar insandan ayırt edilemez bir makine tasarımladığımızı varsayalım. O durumda nedenleri tekrarlayabildiğiniz zaman, muhtemelen etkileri de tekrarlayabilirsiniz. Bu kez de sorunun yanıtı, en azından ilkesel olarak, önemsiz olarak evet'tir. Eğer insanoğluyla aynı yapıda bir makine inşa edebilseydiniz, muhtemelen o makine düşünebilirdi. Hatta o bir insan taslağı olurdu. Eh, bir daha deneyelim.*

*Soru şu değil: ''Bir makine düşünebilir mi?'' ya da ''İnsan yapımı bir şey düşünebilir mi?'' Soru şu: ''Sayısal bir bilgisayar düşünebilir mi?''. Ama bir kez daha soruyu nasıl yorumladığımıza dikkat etmemiz gerekiyor. Kastedilen, matematiksel bir bakış açısına göre,* ***sanki*** *bir sayısal bilgisayarmış gibi tanımlanabilen herşey. Bunun nedeni de bir bilgisayar programını gerçekleyen ya da yürüten olarak tanımlanabilmesi. Tamamen önemsiz bir anlamda, önümdeki masanın üzerindeki kalem bir sayısal bilgisayar olarak tanımlanabilir. Yalnızca çok sıkıcı bir bilgisayar programına sahiptir. Programı şöyle der: ''Orada dur''. Şimdi bu anlamda herhangi bir şey sayısal bir bilgisayardır, çünkü herşey bir bilgisayar programının yürütücüsü olarak tanımlanabilir; böylece bir kez daha sorumuz önemsiz bir yanıt alır. Kuşkusuz beyinlerimiz de sayısal bilgisayarlardır, çünkü çok sayıda bilgisayar programını yürütürler. Ve kuşkusuz beyinlerimiz düşünebilirler. Bu nedenle bir kez daha soruya önemsiz bir yanıt alıyoruz. Ama bu bizim gerçekten sormaya çalıştığımız soru değildi.*

*Sormak istediğimiz şuydu: ''Tanımlandığı şekliyle bir sayısal bilgisayar düşünülebilir mi?'' Yani şunu demek istiyoruz: ''Doğru girdi ve çıktılarla doğru bilgisayar programını gerçeklemek ya da yürütmek, düşünmek için yeterli midir ya da düşünceyi oluşturur mu?'' Ve bu sorunun yanıtı, öncekilerin tersine, açıkça ''hayır''dır. ''Hayır'' olmasının nedeni, söylediğimiz şeydir, yani bir bilgisayar programı salt sentaktik olarak tanımlanmıştır. Ama düşünmek, anlamsız simgelerin idare edilmesinden öte bir şeydir; anlamlı semantik içerikleri de içerir. Bu semantik içerikler, ''anlam'' derken kastettiğimiz şeydir.*

*Burada bilgisayar teknolojisinin belli bir aşamasından söz etmediğimizi tekrar vurgulayalım. Argümanın bilgisayar biliminde gelecek olan, şaşırtıcı gelişmelerle hiçbir ilgisi yoktur. Seri ve paralel süreçler arasındaki ayrımla ya da bilgisayar işlemlerinin hızıyla ya da çevreleriyle nedensel etkileşime girebilen bilgisayarlarla, hatta robotların icadıyla hiçbir ilgisi yoktur. Teknolojik ilerleme her zaman aşırı derecede abartılır ama bu abartmayı çıkarttığı-*



Ücretsiz PDF Kitap Arsivi - https://rantey.org

nız zaman bile bilgisayarların gelişimi oldukça hayret verici olmuştur; gelecekte daha da hayret verici gelişmelerin gerçekleştirileceğini mantıklı olarak bekleyebiliriz. Hiç kuşkusuz bilgisayarlarda insan davranışını şimdi olduğundan ya da geçmişte başarıldığından çok daha iyi benzetleyebileceğiz. Benim anlatmaya çalıştığım, zihinsel durumlardan söz ederken, bir zihne sahip olmaktan sözederken, tüm bu benzetlemelerin önemsizleştiği; teknolojinin ne kadar iyi olduğu ya da bilgisayarda hesaplamaların ne kadar hızlı yapıldığı hiç farketmez. Eğer sözü edilen gerçekten bir bilgisayarsa, işlemleri sentaktik olarak tanımlanmak durumundadır; oysa bilinç, düşünce, duygu, duyu ve diğer şeyler sentakstan ötesini gerektirirler. Tanımı gereği bilgisayar bu özellikleri ne kadar güçlü **benzetleme** yetisine sahip clursa olsun, **aynen yeniden üretmekten** acizdir. Buradaki ana ayrım, benzetlemekle yeniden üretmek arasındadır. Ve hiçbir benzetleme sendi başına aynısını yeniden üretme anlamına gelmez.

## Beyinler ve Zihinler: Biyolojinin Belirleyiciliği

Şimdiye kadar yaptığım, bu konuşmaya başlarken verdiğim örneklerin gerçekten göründükleri kadar mantığa aykırı oldukları sezgisine bir temel oluşturmaktı. Yine de bu tartışmada şaşırtıcı bir soru var: ''Niye insanlar bilgisayarın düşünebildiğini ya da duyu ve duygulara vb. sahip olduğunu düşünmüş olsunlar?'' Formel bir tanımı verilebilen her sürecin bilgisayar benzetlemesini yapabiliriz. İngiliz iktisadında para akışının ya da İşçi Partisi'ndeki iktidar dağılımı örüntüsünün bilgisayar benzetlemesini yapabiliriz. Yağmur fırtınalarının ya da Doğu Londra'daki ambar yangınlarının bilgisayar benzetlemesini yapabiliriz. Bu durumlardan hiçbirinde kimse bilgisayar benzetlemesinin aslında gerçek vaka olduğunu düşünmez; kimse bir fırtınanın bilgisayar benzetlemesinin bizi ıslatacağını, bir yangının bilgisayar benzetlemesinin evi yakacağını düşünmez. O halde zekası yerinde bir insan, niçin zihinsel süreçlerin bilgisayar benzetlemesinin zihinsel sürecin kendisi olduğunu düşünsün? Bunun cevabını gerçekten bilmiyorum, çünkü aslını söylemek gerekirse bu fikir bana ta başından beri delice geliyor. Ama birkaç spekülasyon yapabilirim.

Bir kere, zihin işin içine girdiğinde, birçok insan bir tür davranışçılığa eğilim duyar. Eğer bir sistem Çince biliyormuş gibi davranıyorsa, gerçekten Çince bilmesi gerektiğini düşünür. Ama bu tür bir davranışçılığı, Çince odası argümanıyla zaten saf dışı bıraktık. Pek çok kişinin paylaştığı bir başka varsayım, zihnin biyolojik dünyanın bir parçası olmadığı, doğa dünyasının bir parçası olmadığıdır. Aşırı yapay zeka iddiası zihnin tümüyle formel kavranışına dayalıdır; zihnin, şu ya da bu şekilde diğer biyolojik ürünler gibi biyolojik süreçlerin somut ürünü olarak ele alınamayacağı inancına. Kısacası bu tartışmalarda bir tür kalıntı halinde bir düalizm vardır. YZ partizanları zihnin doğal biyolojik dünyanın bir parçası olmaktan öte bir şey olduğuna, onun yalnızca salt formel olarak tanımlanabileceğine inanırlar. Bunda yatan paradoks, YZ literatürünün, ''düalizm'' denilen bir görüşe karşı lanetlerle dolu olmasıdır; oysa gerçekte aşırı YZ'nın tüm tezi bir tür düalizme dayalıdır. Zihnin, dünyadaki diğerleri gibi doğal biyolojik bir fenomenden ibaret olduğu fikrinin yadsınmasına dayalıdır.

Bu bölümü zihin ve beyin arasındaki ilişkiler hakkında düalizme karşıt bir görüş geliştirerek bitirmek istiyorum. Argümanın oldukça basit bir mantıksal yapısı vardır; bu nedenle geçerli olup olmadığını anlayabilirsiniz. İlköncül şudur:

***1. Beyinler zihinlere neden olur***

Şimdi bu gerçekten de çok kaba. Bununla kastettiğimiz, bir zihni oluşturduğunu düşündüğümüz zihinsel süreçlerin beynin içindeki süreçler nedeniyle oluştuğu, tamamen onlar nedeniyle. Ama kabalaşmaktan çekinmeyelim ve bunu dört sözcüğe indirelim: beyinler zihinlere neden olur. Ve bu, dünyanın nasıl işlediğine ilişkin bir olgudur. Şimdi iki numaralı önermeyi yazalım:

***2. Sentaks semantik için yeterli değildir.***

Bu önerme kavramsal bir gerçektir. Neyin salt formel, neyin içeriğe haiz olduğuna ilişkin nosyonumuzun ayrımını belirtir. Şimdi -beyinlerin zihinlere neden olduğu ve sentaksın semantik için yeterli olmadığına dair- iki önermemize bir üçüncü, bir de dördüncü önerme ekleyelim:

***3. Bilgisayar programları tümüyle formel, ya da sentaktik yapılarla tanımlanırlar.***

Bu önerme bence tanımı gereği doğrudur, bilgisayar programı nosyonuyla kastettiğimiz şe-



Ücretsiz PDF Kitap Arsivi - https://rantey.org

yin bir parçasıdır.

**4. Zihinlerin zihinsel içerikleri vardır; spesifik olarak, semantik içerikleri vardır.**
Ve bence bu, zihnimizin nasıl çalıştığına ilişkin apaçık bir olgudur. Benim düşüncelerim, inançlarım ve arzularım bir şeye ilişkindir, ya da bir şeye atıfta bulunurlar ya da dünyadaki durum ve olayları ilgilendirirler: bunu yapmalarının nedeni de içeriklerinin onları dünyadaki bu olayların durumlarına yöneltmesidir. Şimdi bu dört öncülden ilk sonucumuzu çıkartabiliriz ve bu da açıkça 2., 3. ve 4. öncüllerden türer:

**SONUÇ: 1. Hiçbir bilgisayar programı kendi başına bir sisteme zihin vermeye yeterli değildir.**

**Kısacası programlar** zihin değildirler ve kendi başlarına zihin sahibi olmak için de yeterli değildirler.

Şimdi bu çok güçlü bir sonuçtur çünkü yalnızca programlar tasarımlayarak zihin yaratma tasarısının baştan iflas ettiği anlamına gelir. Ve bu sonucu teknolojinin herhangi bir özgül aşamasıyla ya da programın karmaşıklık durumuyla hiçbir alakası olmadığını burada bir kez daha vurgulamak gerekir. Bu tüm ilgili tartışmacıların (ya da hemen hemen hepsinin) üzerinde fikir birliğine vardığı bir dizi aksiyomdan türetilmiş salt formel ya da mantıksal bir sonuçtur. Yani en ateşli yapay zeka savunucuları bile gerçekte bir biyoloji sorunu olarak beynin zihinsel süreçlere neden olduğunu ve programların salt formel olarak tanımlandığını kabul ederler. Ancak bu sonuçları bildiğimiz başka şeylerle birleştirdiğiniz zaman, buradan derhal aşırı YZ tasarısının gerçekleşemeyeceği ortaya çıkar.

Ancak bir kez bu aksiyomları elde ettikten sonra, başka ne türetebileceğimize de bakalım: İşte ikinci sonucumuz:

**SONUÇ 2. Beyin fonksiyonlarının zihne neden olma tarzı, yalnızca bilgisayar programı yürütme yoluyla olamaz.**

Ve bu sonuç, ilk öncülü ilk sonucumuzla birleştirerek ortaya çıkar. Yani, beyinlerin zihne neden olduğu ve programların bu işi yapmakta yetersiz olduğu gerçeğinden, beyinlerin zihne neden olma yollarının yalnızca bilgisayar programı yürütmek olmadığı ortaya çıkar. Ben bunun da önemli bir sonuç olduğunu düşünüyorum, çünkü bence beynin bir sayısal bilgisayar olmadığı, ya da en azından salt bir sayısal bilgisayar olmadığı neticesini veriyor. Daha önce herşeyin önemsiz olarak sayısal bilgisayar olarak tanımlanabileceğini gördük ve beyinler de buna istisna teşkil etmez. Ancak bu sonucun önemi, beynin hesaplayıcı özelliklerinin, zihinsel durumlar üretecek şekilde işleyişini açıklamakta yetersiz oluşu. Gerçekten de bu hepimiz için sağduyuya dayalı bilimsel bir sonuç olmalı çünkü tüm yaptığı bize beyinlerin biyolojik motorlar olduğu, biyolojilerinin önemli olduğu olgusunu hatırlatmak. Yapay zeka alanındaki bazılarının öne sürdüğü gibi zihinlerin yalnızca insan beyinlerinde bulunması, zihin hakkında önemsiz bir olgu değildir.

Şimdi, ilk öncülümüzden üçüncü bir sonuç çıkartabiliriz:

**SONUÇ 3. Zihne neden olan başka herşey, en azından beyninkine eşit nedensel güçlere sahip olmalıdır.**

Bu üçüncü sonuç da ilk öncülümüzün önemsiz bir sonucudur. Bu biraz şunu demeye benzer: ''Eğer benzin motorum arabamı saatte 75 mil hızla sürüyorsa, bunu yapma yetisine sahip herhangi bir dizel motoru benim motorumunkine eşit bir güç çıktısına sahip olmak durumunda olacaktır.'' Kuşkusuz başka bir sistem de, beynin gerçekte kullandığından tümüyle farklı kimyasal ya da biyokimyasal özellikler kullanarak zihinsel süreçlere yol açabilir. Başka gezegenlerde ya da başka güneş sistemlerinde zihinsel durumları olan ve bizimkinden tümüyle farklı biyokimyalar kullanan varlıklar olduğu ortaya çıkabilir. Marslıların dünyaya geldiklerini ve onların zihinsel durumları bulunduğu sonucuna vardığımızı varsayalım. Ama beyinlerini açtığımızda içinde yalnızca yeşil balçık görmüş olalım. Yine de bu yeşil balçık, eğer bilinçlilik ve Marslıların zihinsel hayatının diğer yönlerini üretme işlevi gördüyse, insan beynine eşit nedensel güçlerinin bulunması gerekir.

Bu tartışmanın neticesi, zaten bilmekte olduğumuz şeyi bize bir kez daha hatırlatmaktır sanırım: zihinsel durumlar biyolojik fenomenlerdir. Bilinçlilik, niyetlilik, öznellik ve zihinsel neden-sonuç ilişkisi; büyüme, yeniden üreme, öd salgılama ve sindirimin yanısıra, biyolojik yaşam tarihimizin birer parçasıdır.

**John Searle**
**1984 Reith Konferansları'ndan**



Ücretsiz PDF Kitap Arsivi - https://rantey.org

''Kenarları say, eğer dört kenar varsa, bu bir karedir'' denebilir. Ancak bu sorunu bir aşama geri götürüyor. Bir kenarı ayırdetmekle ilgili komutlar ne olacaktır? ''Rastgele noktalar al ve bunların uç noktaları arasındaki en kısa uzaklığa denk düşen bir çizgi üzerine düşüp düşmediklerine bak!'' vb. İnsan bu noktaları nasıl bulur? Her şeyden önce, geometrik şekillerle karşı karşıya olan birinin deneyim alanında, hiç değilse fenomenolojik olarak noktalar yoktur. Belki burada komutlar bitiyor ve bu noktada artık yalnızca şu söylenebilir. ''Noktaların kendisi, bilinçsizce ve otomatik olarak görülüp, aralarındaki en kısa uzaklık otomatik olarak kestiriliyor.'' Ancak komutların nerede duracağının ve nerede böylesi bir doğrudan görüye başvuracağımızın ölçütü nedir? Neden komutlar daha önce ya da daha sonra değil de, tam bu düzeyde duruyor? Benzer bir sorular ve güçlükler dizisini, kareyi çemberden değil de, kırmızıyı maviden ayırdetmekle ilgili olarak türetebiliriz. Görüldüğü gibi, psikolojide tam bir tanımın bir dizi komut biçimini alması gerektiği varsayımı hiç de ampirik bir düzeyde sınanabilecek bir açıklıkta değildir.

## Psikolojik Varsayımın Tarihi

Ancak yorumlanma güçlüğü bir yana, bu varsayım hatırı sayılır bir düşünce geleneğinden kaynaklanmaktadır. Ünlü Alman filozofu Kant, tüm insan düşüncesini ve hatta algılama ve yaşantının kendisini bile, açık seçik dile getirilebilecek kurallarla çözümlemeye çalışmıştı. Bilginin açık seçik bir dizi komuta çözümlenebileceği varsayımı ise daha da eskidir. Bu varsayımın temelinde yatan inanışı bir cümleyle şu şekilde özetleyebiliriz: ''Herhangi bir olguyu bir dizi komuta çözümleyen tam bir tanım mümkün olabilir -olmalıdır-, çünkü ancak böyle bir çözümleme, ne olup bittiğini anlayabilmemizi sağlayabilir.'' Bu inanışın kökeni felsefenin başlangıcına, yani anlamak ve akla ilişkin kavramlarımızın ilk oluştuğu döneme dayanır. *Euthyphro*'da anlamanın çözümlemesini bu şekilde yapan Platon, *Meno*'da ''davranışı felsefeci için anlaşılır kılan kurallar, bu şekilde davranmakta olan kişi tarafından da izleniyor mu?'' diye sorar. Başka bir deyişle, kurallar yalnızca davranışa dışardan bakmakta olan felsefeci ne olup bitmekte olduğunu anlamak istiyorsa mı gereklidir? Yoksa zekice davranmakta olan bir insan da, zekice davranabilmek için bu kuralları izlemek zorunda mıdır? Kurallar zeka gerektiren davranışı mümkün kılan ön-koşullar mıdır?

Platon çoğu becerinin bir tür pragmatik oyalanma olduğunu düşündüğü için, kuralların beceri ürünü olan davranışları ne anlamak ne de üretmekle bir ilgisi olduğunu düşünüyordu. Ama teorem kanıtlama ya da erdemli bir biçimde davranma sözkonusu olduğunda ''insanlar tam anlamıyla kuralların bilincinde olmaksızın eylemde bulunsalar da, eylemlerinin filozof tarafından açıklanabilecek rasyonel bir yapısı vardır'' görüşünü savundu. Onun için soru, matematikçi ve erdemle davranan kişinin, örtük bir biçimde de olsa, bu kurallardan oluşan programı izleyip izlemediğiydi.

Anlamak ve açıklamakla ilgili kavramlarımızın tarihi açısından bu, belirleyici bir konudur. Platon'un yanıtı kuşkuya yer bırakmayacak bir biçimde olumludur. Ona göre, anlamlı olan, diğer bir deyişle keyfi olmayan her eylemin, bir kuramla ifade edilebilecek rasyonel bir yapısı vardır ve sözkonusu eylemde bulunan kişi, bir dizi kural biçiminde dile getirilmiş olan bu kuramı, hiç değilse örtük olarak izliyordur.

Platon'a göre bu kurallar zihinde doğuştan programlanmış bir biçimde durmaktadırlar ve uygun soruların yöneltilmesiyle açıklığa kavuşturulabilirler. Bu yüzden, belli bir davranışın neyi sağladığını **anlamamızı** mümkün kılan bir kuram, aynı zamanda bu davranışın nasıl üretildiğinin de **açıklamasıdır.** Anlama kavramı ve anlamayla açıklama arasındaki ilişki bu şekilde düşünüldüğünde, yapay zeka iddialarını kabul etmek kolaylaşır. Bildiğiniz gibi bu iddialara göre, tam bir davranış tanımı kesin bir kurallar dizisidir ve bu kurallar, sözkonusu davranışı üretmek üzere sayısal bir bilgisayarı programlamakta komutlar olarak kullanılabilir.

## Doğal Bilimler ve İnsan Davranışı

Düşünmenin kurallarla betimlenebilecek bir hesaplama olduğu varsayımını araştırdık. Bu varsayımın cazibesi, ahlaklı hayatın çok daha dayanılabilir ve bilginin çok daha kesin olduğu sonuçlarını beraberinde getiriyor olmasından ileri geliyor olsa gerek. Ama cazibesi bir yana, bu varsayımın olabilirliğinin tek dayanağı, modern fizik bilimin ardında yatan mekanistik varsayımlara, davranış bilimlerinde formalistik bir varsayımın tekabül etmesi gerektiği yolundaki inançtır.

Bu inancın anlamlı olduğu bir düzey, kuşkusuz vardır. İnsan bir nesnedir. Modern fizik bilimi bize, fiziksel bir nesnenin davranışlarının tam bir tanımını kesin kanunlara dile getirme



Ücretsiz PDF Kitap Arsivi - https://rantey.org

olanağı sağlamıştır. Bu kanunlar bir bilgisayara verilen komutlar olarak kullanılabilir ve bilgisayar böylelikle, ilkece de olsa, sözkonusu nesnenin davranışını benzetleyebilir (**simulate**). Buradan hareket ederek, insan davranışının nöropsikolojik bir tanımının enerji girdileri, beyindeki fiziksel-kimyasal olaylar ve fiziksel gövdenin hareketi gibi veriler cinsinden verilebileceği inancına varılır. Bütün bu saydığımız veri ve olay türlerinin ortak özelliğiyse, bir bilgisayarda benzetlenebilmeleridir.

Fiziksel bir nesne olduğu ölçüde insan gövdesinin de bu şekilde tanımlanıp incelenebileceği düşüncesi, modern çağın başından beri bilim tasarısının bir parçası olmuştur. Beyin, açıkça bir enerji dönüştürüm organıdır.

Fakat bir psikolog için bu fiziksel tanımlama, tüm psikolojik terimleri dışladığı için, hiçbir şekilde psikolojik bir açıklama olamaz. Psikoloji bağlamında açıklamak istediğimiz olayların hiçbirini -eğilimleri, tutumları, algıları, anıları vb.- bu düzeyde tanımlayamayız. Enerji alınmakta ve dönüştürülmektedir -nöropsikoloji düzeyinde kaldığımız sürece anlatabileceğimiz hikâye bundan ibarettir.

## Fizyoloji, Fenomenoloji ve Psikoloji

Ancak, davranışlar sözkonusu olduğundan hepimizin yakinen tanıdığı bir düzey daha vardır. Gündelik hayatımız içersinde ve davranışlarımızı gündelik dille açıklarken başvurduğumuz düzey. Bu düzeyi ''fenomenoloji'' düzey diye adlandırabiliriz. Ancak bu düzeyde yaptığımız betim, tanım ve açıklamalar kuşkusuz bilimsel bir amaçla yapılmamaktadır. Bu düzeyde gördüklerimizi masalar, sandalyeler, diğer insanlar -dostlar, yabancılar-, işittiklerimizi sesler, bazen kelimeler, yapıp ettiklerimizi de amaçlı davranışlar ya da alışkanlıklar şeklinde, ama her zaman anlamlı kategorilere başvurarak tarif ederiz. Fakat kendisi için bağımsız bir bilgi objesi tanımlama durumunda olan bir psikolog için bu düzey, komutlar ve kurallar içermediği için, fizyolojik düzeyden daha tatmin edici değildir. Dolayısıyla bu düzeyde kaldığımız sürece, davranışların bilgisayar tarafından benzetlenmesini mümkün kılacak açıklamalara ulaşmamız sözkonusu olamaz.

Bu kavramsal açmaz karşısında psikologlar, sürekli olarak kendi disiplinlerini meşrulaştıracak, fizyolojik ve fenomenolojik düzeylerin dışında, bir davranış açıklamasını mümkün kılacak bir üçüncü düzey tanımlama çabalarını sürdürmek zorunda kalmışlardır.

Eğer psikoloji insan davranışının bilimi olacaksa, insanı bir nesne olarak tasarlayarak çalışmak durumundadır. Ancak bu nesneyi fiziksel enerji girdilerine tepkime gösteren bir nesne olarak tasarlayamaz, çünkü bu, fiziğin, fizyolojinin ve nöropsikolojinin özgül görevidir. Seçenek, insan davranışını soyut da olsa bir tür nesnenin, soyut da olsa bir tür girdiye tepkimesi olarak incelemektir. Bu soyut nesne ve enerjinin ne olduğu tanımlanamıyor da olsa, eğer istenen bir açıklamaysa, insan, birbirinden bağımsız olarak tanımlanabilen elemanlara, kanunlara bağlı olarak tepki gösterebilen bir aygıt olarak ele alınmalıdır. Sözü geçen kanunlar, organizmadaki sabit eğilimlerin, nasıl olup da çevreden gelen girdilere karşı, gözlenen davranış çeşitliliğini ürettiğini açıklayan nedensel kanunlar olmak zorundadır.

İnsanın modeli olarak düşünülen aygıt bir refleks makinesi, girdilerle davranışlar arasındaki ilişkileri açıklayan kanunlar da çağrışım kanunları olarak düşünüldüğünde, kaynağını İskoç filozofu David Hume'un ampirisizminde bulan U-D (Uyarıcı-Davranım) psikolojisine varılır. Ya da alternatif olarak, insanın modeli olarak düşünülen aygıt bir bilgi-işlem makinesi, kanunlar da (felsefedeki Kantiyen modele uygun olarak) gerekçeler olarak düşünülebilir. Bu modelde, gerekçeler zihindeki kurallar olarak düşünülür. Geleneksel olarak idealist, entelektüalist ya da akılcı olarak anılan bu görüş bugün bilişsel (**cognitive**) psikoloji ve yapay zeka adları altında savunulmaktadır.

Bilgisayarların yaygınlaşmasına kadar daha başat olmuş olan okul, ampirisist okul oldu. Çünkü akılcı okul, şu soruya bir yanıt bulamıyordu: Eğer insanın tamamı, bir dizi hesaplama ve formel olarak betimlenebilir işlemden ibaretse, bu hesapları yapan, işlemleri uygulayan özne kimdir? Bu soruyu yanıtlamaya çalışan filozoflar zihni, eninde sonunda, insanın aklına ve bütün davranışlarına kuralları uygulayarak, hesapları yaparak rehberlik eden bir küçük adam, bir 'homunculus' olarak düşünmek zorunda kalıyorlardı. Bu ise, elbette, açıklanması istenen nesneye, daha üst bir düzeyde açıklayıcı olarak başvurmak anlamına gelir.

Bilgisayarın bu gelenek için çekiciliği büyük ölçüde, bilgisayarların bir özneye gerek duymadan kurallara göre çalışmasından kaynaklanmaktadır. Dahası bir dili konuşmak gibi U-D psikolojisi çerçevesi içerisinde açıklanamayacak kadar karmaşık bazı davranışlar da, ilk bakışta formel kurallar aracılığıyla çözümlenebilir



Ücretsiz PDF Kitap Arsivi - https://rantey.org

gözükmektedir.

Bilgisayar fiziksel bir nesnedir. Ama onu betimlerken, transistörlerdeki elektronlardan değil, açık ya da kapalı durumda olan elektronik devrelerinin oluşturduğu düzenden, yani işlevselliğinden söz ederiz. Eğer psikolojik kavramlar da bu yüksek düzeyli düzenler şeklinde yorumlanabilirse, aranan, fizyoloji ve fenomenolojiden bağımsız düzey bulunmuş olacaktır.

Ancak böylesi bir düzeyin varlığı psikoloji için öylesine hayati bir önem taşımaktadır ki, bilgisayarlar aracılığıyla keşfedilen bilgi-işlem düzeyinden söz etmek için kullanılan dilin anlaşılır ve tutarlı bir dil olup olmadığı sorusu sorulmamaktadır bile. Örneğin şöyle bir cümleyi ele alalım: ''Organizma bir planı yürüttüğünde, adım adım ilerler ve bir sonraki adıma ancak bir öncekini tamamladıktan sonra geçer.''

Böyle bir cümlede üç ayrı kavramsal düzey birbirine karışmış durumdadır. Organizma biyolojiye ait bir terimken, ''yürütme'' insani eylemlere ait bir terimin makinelere uyarlanmış bir kullanımıdır. '''Adım adım ilerler'' cümleciğinin öznesi ise olsa olsa, günelik dilden tanıdığımız, niyetleri, alışkanlıkları vb. olan bir 'fail' (**agent**) olabilir ancak. Kısacası böyle bir söyleyişte, organizmanın ve giderek makinenin kişiselleştirilmesi ile karşı karşıyayız. Tersi de sık sık rastlanan bir durum. ''Zihinsel işlemler'' ve ''zihinsel süreçler'' gibi terimlerde rastladığmız ise aklın mekanikleştirilmesi.

Bu kavramsal karmaşayı çözmenin bir yolu, bilgi-işlemcilerin özellikle yapay zeka ile ilgili iddialarını, özelliklerini nispeten daha iyi bildiğimiz fiziksel ve fenomenolojik (gündelik dil) düzeylerini akıldan çıkarmadan değerlendirmek olabilir. Görsel sistem üzerinde çalışan ve bilgisayarların örüntüleri (**pattern**) tanıma olanaklarını araştıran Neisser şöyle yazmaktadır: ''Gerçekten de ağaçlardan, insanlardan, arabalardan ve hatta kitaplardan oluşan bir gerçek dünya vardır (...) Ancak bu gerçek dünyaya ve özelliklerine doğrudan erişemeyiz''(...) ''Bizim için duyumsal girdi bir kitap sayfasının kendisi değil, bir ışık huzmeleri örüntüsüdür.'' Buraya kadar söylenenlere katılınabilir. Ancak bir sonraki cümlede fenomenolojik ve fizik düzeylerin karıştırılmakta olduğunu görüyoruz. ''Gözdeki mercek tarafından odaklanan bu huzmeler başlattığı sinirsel süreç, sonunda görmeye, okumaya ve anımsamaya yolaçar.''

Burada ''yolaçar'' teriminin çift anlamlılığından yararlanılmaktadır. ''Görme'' kimyasal bir süreçin adı değildir; gündelik dile ait bir terim olarak hepimizin yakinen tanıdığı bir yeteneğimize işaret eder. Bu yüzden ''yolaçma'' terimi ile aynı dizi üzerinde yeralan olayların birbirini izlemesi kasdediliyorsa, kimyasal ve sinirsel süreçler, kendisi kimyasal olmayan bir olaya yolaçamaz. Diğer yandan ''yolaçma'' terimiyle ''yeterli ve gerekli koşul olma'' kasdediliyorsa, o zaman bu sinirel süreçler zincirinin tamamı (yalnızca son aşaması değil), ''görme''nin yeterli ve gerekli koşuludur.

Burada sorun aslında başta kullanılan ''duyumsal girdi'' (**sensory input**) kavramının muğlaklığında yatmaktadır. Eğer duyumsal girdi ile kasdedilen bir enerji türüyse, o zaman ancak başka bir enerji türüne dönüştürülebilir-sinir sisteminde yeralan olayların tümü, enerji dönüştüren fiziksel-kimyasal olaylardır. Enerji/madde dönüştürülebilir, saklanabilir, yeniden kullanılabilir vb. ama hiçbir zaman enerji/madde cinsinden tarif edilmeyen bir şeye dönüştürülemez. Dolayısıyla, enerji dönüştürülmesi olaylarından oluşan bir dizi içersinde, bu cinsten tanımlanamamış bir olay olan ''görme'' yeralamaz, (Kuşkusuz bu, sinirsel olayların görmeyle bir ilgisi olmadığı anlamına gelmiyor. Görsel sistemimiz ve orada gerçekleşen olayların tamamı, ''görme'' diye adlandırdığımız fenomenolojik olayın ön-koşuludur. Başka bir deyişle, biz görmenin mümkün olabilmesi için hangi sinirsel yapı ve olayların gerekli olduğunu araştırabiliriz. Ama aynı nedensel olaylar zincirini fiziksel ve kimyasal olaylarla başlatıp, görme ile bitiremeyiz).

Diğer yandan ''duyumsal girdi''nin bir enerji birimi olarak değil de, algının ilkel bir biçimi olarak kullanılmakta olduğu, yani kendisinin fenomenolojik bir terim olarak kullanıldığı düşünülebilir. Ama bu da kendi yaşantımızla çelişki halindedir. Biz ''ışık huzme örüntüleri'' değil, doğrudan doğruya eşyanın kendisini algılarız -bu anlamda algımız kademeli bir süreç değildir.

Ancak bilgi-işlemin sıralı bir süreç olması gerektiği yolundaki kabul, yapay zeka iddialarını savunanları yalnızca algılamayı çeşitli evrelere bölmeye değil, algılamayla, algılanan nesnenin tanınmasını da birbirlerinden ayırmaya yöneltiyor. Bu görüşlere göre, duyumsal girdiler dönüştürüldükten sonra, kendisi formel bir yapı olan kavramın içerdiği bilgi tarafından düzenlenerek bütünleştiriliyorlar. Yani gördüğümüz aslında bir ağaç ya da insan değil, ''ağaç'' ya da ''insan'' kavramının içerdiği bil-



Ücretsiz PDF Kitap Arsivi - https://rantey.org

gi tarafından düzenlenmiş ''duyumsal girdi''lerdir.
Kuşkusuz kanıtlanmasını istediğimiz iddiayı, yani beynin sayısal bir bilgisayar olduğunu, baştan kabul edersek, kavramsal karmaşayı aşar, ''kavramı'' verileri düzenleyen formel bir yapı olarak tanımlayabiliriz. Bu takdirde, ''duyumsal girdi'' de ne bir enerji birimi ne de ilkel bir algı olarak, aksine bilgi ''bit''leri olarak tanımlanacaktır. Kavram da bu bitlerin nasıl düzenleneceğini belirleyen bir dizi komut olacaktır. Bu ise insanların dünyasının kendisinin çeşitli şekillerde düzenlenebilecek bilgi bitleri şeklinde örgütlenmiş bilgi bitlerinden oluştuğu yolundaki ontolojik varsayımı gerektirmektedir. Ontolojik varsayımı daha sonraki bir bölümde inceleyeceğiz. Ancak bu noktada sözünü ettiğimiz varsayımların birbirlerinden tamamen bağımsız olmadığına, psikolojik varsayımın, ancak ontolojik varsayım kabul edildiği takdirde tutarlı ve anlaşılır bir biçimde dile getirilebileceğine dikkati çekelim.

## Epistemolojik Varsayım

Yapay zeka iddialarının temelinde yatan psikolojik varsayım reddedilse bile, bu bir iyimserlik kapısını hâlâ açık bırakır. İnsan davranışları anlaşılabilir kurallara göre yürüyen bir dizi bilinç dışı işlemle açıklanamıyor bile olsa, bu aynı davranışları bu tür kurallarla bizim taklit edemeyeceğimiz anlamına gelmez.

Başka bir deyişle, psikolojik varsayımın reddi beraberinde felsefi (ya da aşırı) diye adlandırdığımız yapay zeka iddialarının da reddini getirir. Ama teknik (ya da ılımlı) diye adlandırdığımız yapay zeka iddiaları, yani başka ilkelere göre de olsa bilgisayarların insan davranışlarını taklit edebileceği iddiası, psikolojik varsayımın reddine rağmen savunulabilir. Teknik yapay zeka iddialarının temelinde yatan varsayım, insan aklının formelleştirilebilir ve dolayısıyla bilgisayar tarafından taklit edilebilir bir nesne olduğudur. Bu epistemolojik bir varsayımdır.

Gezegenleri düşünelim. Gezegenler güneşin çevresinde dönerken diferansiyel denklemler çözmüyorlar. Hiçbir kuralı izlemedikleri halde, biz onların davranışlarını bir dizi kural aracılığıyla betimleyebiliyoruz. Ya da farklı bir örnek verelim: Bisiklet üzerindeki bir adam, ağırlığını o yana ya da bu yana vererek, düşme olasılığını azaltabilir. Biz, gözlemci olarak,sürücünün ne yapmakta olduğunu, bir kural aracılığıyla açıklayabiliriz: Eğimi hızın karesiyle ters orantılı olan eğrilerin dönme hareketi. Bisiklet sürücüsü bu kurala, elbette ki, bilinçli olarak uymuyor. Fakat bu formelleştirme bize onun yeterliliğini, yani neyi başarıp neyi başaramayacağını anlama olanağını verir. Ama bu kural, hiçbir şekilde onun eyleminin açıklaması değildir. Kural bize bir bisikletin nasıl olup da sürülebildiğini söyler ama bisiklete binme eylemi sırasında, sürücünün beyninde ya da zihninde yeralmakta olan süreçler hakkında hiçbir bilgi vermez.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Bu noktada psikolojik ve epistemolojik varsayımlar arasındaki önemli bir farka geliyoruz: Gerçi her iki varsayım da, anlamakla formelleştirmeyi eşanlamlı olarak gören Batı düşüncesinin temel kabulünü benimser ama psikolojik varsayımı benimseyenler davranışı üreten kurallarla, davranışın anlaşılması amacıyla formelleştirilmesinde kullanılan kuralların bir ve aynı kurallar olduğunu öne sürerler. Oysa epistemolojik varsayımın benimsenmesi için, davranışın aslında nasıl üretildiği hakkında hiçbir görüş bildirmeden, her davranışın birtakım kurallara göre formelleştirilebileceğini iddia etmek yeterlidir. Bu formelleştirme sonunda elde edilen kurallar da, sözkonusu davranışının bilgisayarda taklit edilmesi için kullanılabilir.

Görüldüğü gibi epistemolojik varsayım daha zayıftır. Bu yüzden de eleştirilerden daha az zedelenir. Çünkü yalnızca epistemolojik varsayımı benimseyenler, formellik gereğini, yalnızca bir yeterlik kuramı (**theory of competence**) olarak ele almakta, onu, insan eylemlerine dair bir kuram olarak yorumlamamaktadırlar.

Bu durumda epsitemolojik varsayımın iki temel iddia içerdiğini söyleyebiliriz:

**1** / Her davranış formelleştirilebilir.
**2** / Bu formelleştirme sözkonusu davranışın yeniden üretilmesini sağlayabilir.

Bu bölümde biz birinci iddia ile ilgili olarak, bu iddianın doğal bilimlerden devralınmış ve insan bilimleri alanında geçerliliği sınanmamış bir iddia olma olasılığı üzerinde duracağız. İkinci iddia ile ilgili olarak ise, her yeterlik kuramının aynı zamanda bir eylem kuramı olmayabileceği olasılığı üzerinde duracağız. Başka bir deyişle, fizik bilimi ile teknolojik uygulamaları arasındaki ilişkiden hareketle, insan bilimlerinin olası uygulamaları için sonuçlar çıkaramayabiliriz. Özgül bir zaman ve mekandan bağımsız bir yeterlik kuramı, insan davranışlarının yeniden üretilmesi için yeterli olmayabilir. Hatta daha da ileri giderek, ''kuram'' kelimesine yüklenen geleneksel anlamla, insan faaliyetlerinin (**performance**) kuramı olamayacağı olasılığı üzerinde de durulabilir. Böyle bir sonuç kabul edildiği takdirde, o zaman yapay zeka iddiaları da inandırıcılıklarını yitirir.

Bugüne kadar, özellikle bilgi-işlem ve yapay zeka alanlarında epistemolojik varsayımı desteklemek için en çok başvurulan alanlar, fizik ve dilbilim oldu. Bu yüzden bu iki alan üzerinde ayrıntılı olarak durmamız gerekecek. Fizikte gerçekten de hareketin genel kanunlarını betimleyen formel bir sistem buluruz. Ama bu tür bir formel sistemin, yapay zekâ iddiasına ilişkin olmadığı olasılığı, gözden geçirilmek zorunda olan bir olasılıktır.

Dilbilim, insan davranışlarını daha doğrudan ilgilendiren formel sistemler üzerinde çalıştığından, ilk bakışta yapay zekâ iddiaları ile çok daha yakından ilintili görünmektedir. Fakat konuşmakta olan bir insanın yeterliğini açıklayan bu formel sistemin ille de aynı davranışın yeniden üretilmesini, yani konuşmacının başardığını bilgisayarın da başarması mümkün kılmayabilir.

## Fizik ve Yapay Zeka

Yapay zeka iddialarına inanırlık kazandıran şeylerden birinin, fiziğin doğal nesnelerin hareket kanunlarını formel bir biçimde dile getirmiş olması olduğunu görmüştük. Buradan hareketle şu sorulmaktadır. Neden aynı şekilde ''insan davranışının kanunları da keşfedilebilip formelleştirilmesin?''

Burada sorunu iyice anlaşılır kılmak için şu noktaya değinelim: ''Davranışın kanunları olması''ndan kasdedilen sadece davranışların belli bir düzenliliğe sahip olması değil, bir bilgisayar programına verilebilecek ya da başka bir formel sistemde dile getirilebilecek bir türden düzenliliğe sahip olmasıdır.

Burada şöyle bir adım atabiliriz. Eğer ''davranış'' terimiyle anlamlı insan eylemlerini değil de, biyolojik bir nesne olarak insan organizmasının fiziksel, kimyasal hareketlerini kasdedersek, kuşkusuz bunlarla ilgili kanunları, fiziği ve kimyanın bir parçası olarak keşfedebiliriz. Başka bir deyişle, insanın sinir sisteminin işleyişinin kanunlarını keşfedebiliriz. Bu denklemler, bir önceki bölümde gördüğümüz türden ''duyumsal girdi'' gibi terimler değil, sinir sisteminin kimyasal bileşenlerine ilişkin değerler içerecektir. Bir kez bunu başardıktan sonra, bu kanunları dile getiren denklemleri sayısal bir bilgisayara yükleyerek çözdürmemizin önünde hiçbir engel kalmaz.

Ancak burada yapılmakta olanın bir bilgi-işlem sürecinin benzetlenmesi olduğunu ve dolayısıyla insan beyninin de aslında sayısal bir bilgisayar olduğunu söyleyecek olursak, gemilerin ve uçakların da aslında birer sayısal bilgisayar olduğunu kabullenmek durumundayız demektir. Çünkü onların hareketleri de, sayı-



Ücretsiz PDF Kitap Arsivi - https://rantey.org

sal bir bilgisayar tarafından çözülebilecek matematiksel denklemler tarafından ifade edilebilir.

Sorun neyin bir bilgi işlem sürecinin benzetlenmesi sayılması gerektiğinin kesin olarak belirlenebilmesidir. Herhangi bir sayısal bilgisayar, başka türden, örneğin analog, bir bilgi işlem sürecini sürdürmekte olan bir aygıtın yapısını ve işleyişini, bu yapı genel fiziksel yasalara uygun olduğu sürece, benzetleyebilir. Ancak bu aygıtın fiziksel süreçlere göre işleyişinin benzetlenmekte olduğu, aygıtın bilgi işlem sürecinin de benzetlenebildiği anlamına gelmez. Çünkü bu durumda, sayısal bilgisayarın işlemekte olduğu bilgi, diğer aygıtın fiziksel ve kimyasal yapısıyla ilgili bir bilgidir. Dolayısıyla her tür bilgi işlem sürecinin sayısal bir bilgisayarda benzetlenebileceği iddiası, yanıltıcı bir iddiadır. Bütün söylenebilecek şey, her tür bilgi için, sayısal bir biygisayarın, o tür biygiyi işleyebilecek bir aygıtın yapısını benzetleyebileceğidir.

## Dilbilim ve Yapay Zeka

Yapay zeka iddialarının inandırıcılığı bakımından fizik biliminin başarısı fazla ikna edici olmadığının görüldüğü bir durumda, bu alanda çalışanların, doğrudan bir insan davranışını formelleştirmeye çalışan dilbilime dönmesi çok doğaldır.
Galileo, nesnelerin Aristoteles fiziğinin ilginç bulduğu bir sürü özelliğini gözardı edip geri kalan matematiksel ilişkilerin nesnenin hareketini betimlemeye yettiğini keşfederek modern fiziğin temellerini attı. 1960'larda, benzer bir soyutlamanın dilbilimde de gerçekleştirildiği iddia edilmeye başlandı. Chomsky insanın konuşma faaliyeti **(performance)** -yani belli durumlarda o duruma uygun cümleleri kullanabilme yeteneği- gözardı edildiğinde, geriye kalanın -yani bir cümlenin gramer kurallarına uygun olup olmadığını tayin edebilme yeteneğinin formelleştirilebileceğini iddia ettiler. Başka bir deyişle, bu iddiaya göre, dili kullanma becerisi değilse de, dilsel yeterliliğin **(linguistic competence)** önemli bir bölümü formelleştirilebilir.
Bu iddianın yapay zeka alanında çalışanları heyecanlandırması kaçınılmazdı. Çünkü onların da benimsediği epistemolojik varsayıma göre, insan davranışları, bu davranışların fiziksel, biyolojik betimlemesine başvurmaya gerek kalmadan formelleştirilebilir. Dilbilim de, düzenli olan bazı davranışların sayısal bir bilgisayar tarafından işlenebilecek şekilde formelleştirilebileceğini göstermiş oluyordu.
Ancak dilbilimin başardığı formelleştirme, epistemolojik varsayımın sadece yarısını doğrulamaktadır. Çünkü yapay zeka alanında çalışanların formelleştirmeye çalıştıkları şey, dilsel yeterlilik türünden bir şey değildir. Eğer makinelerin insanlarla doğal dil kullanarak iletişim kullanması isteniyorsa, bunlara gramer kurallarının ve dönüştürümlü dilbilimcilerin keşfettikleri türden kuralların yüklenmesi yetmez; aynı zamanda dilsel faaliyetle, yani somut durumlarda konuşabilme becerisi ile ilgili kurallara da ihtiyaç vardır. Başka bir deyişle, dönüştürümlü dilbilim kuramına varmak için gözardı edilen şey -yani, insanların bildikleri dili kullanabildikleri- yapay zeka alanında çalışanların tam da ihtiyacı oldukları şeydir.
Epistemolojik varsayımın doğrulanabilip doğrulanamayacağı şu kilit soruda düğümlenmektedir: Dilbilimcilerin pragmatik diye adlandırdıkları şey -yani insanların somut durumlarda bu somut duruma uygun cümleler kullanabilme becerileri- formelleştirilebilir mi? Dönüştürümlü dilbilim başarısının bu alana da genellenemeyeceğini düşünmek için iki önemli gerekçe vardır:
**1/** Birincisi ilke düzeyinde bir gerekçedir: Formel bir pragmatik kuramının, insan bilgisinin tamamı hakkında formel bir kuramın olabilmesi gerekir. Bu bir sonraki bölümde göreceğimiz gibi, olanaksız görünüyor. (Bu noktada, psikolojik varsayımın da doğruluğunun böyle bir varsayıma dayandığını hatırlatalım.)
**2/** İkincisi ise daha çok gözleme dayalı bir gerekçe: Bütün dilsel davranışlarımız kurallara uyuyormuş gibi gözükmemektedir. Kimi cümlelerin kuralları ihlal ettiğini görebildiğimiz halde, bu cümleleri anlamakta güçlük çekmiyoruz. O kadar ki dillerin değişmesinde çok önemli bir rol oynayan, mecazi dil kullanımlarının hepsini bu türden olaylara örnek olarak verebiliriz. Örneğin ''Kalemi umut dolu'' bir yazarın üslubunu tartışırken kullanabilecğimiz bir cümle olabilir. Ancak bu durumda, bilgisayar kendisine verdiğimiz kurallar gereğince hareket edeceğinden, ''umut'' kelimesini, kaleme doldurulabilecek şeyler (mürekkep, su vs.) listesinde arayacaktır.
Oysa böyle bir durumla karşı karşıya kalan insan, dilbilgisi kurallarını dahi gözden geçirmek zorunda kalmayacaktır. Çünkü bizim kullandığımız kuralların uygulanabilirlik alanlarının sınırları kesin olarak tesbit edilmiş değil gibi gözüküyor. Oysa bir makine için karşılaştığı her şeye ya kurallara uygun ya da tamamen keyfidir.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Bu mülahazalar dilbilimcilerin dilsel yeterlilik alanındaki başarılarının dil kullanımına genellenemeyeceğini düşündürüyor. Çünkü dilsel yeterliliği açıklamak için geliştirilmiş olan formel sistem zamandan tamamen bağımsız bir sistem olarak, belli somut durumlarda insanların cümleleri anlama ve kullanma süreçlerini açıklama iddiasında değildir. Halbuki yapay zeka alanında çalışanlar, belli nesnelerin yerel bir öneme haiz olduğu somut bazı durumlarda makinelerinin insanlarla iletişim kurabilmesini istiyorlar. Nesnelerin yerel bir öneme sahip olması ise ancak bu durumla ilgilenmekle, bu durumun bir tasarının parçası haline gelmesiyle mümkün olur.

Kuşkusuz eğer sözkonusu durum yanlızca, genel bazı özellikleriyle tarif ediliyorsa, bilgisayar bu duruma uygun bir cümle üretebilir. Başka bir deyişle, yapay bir zeka ile donanmış bir bilgisayar, somut bir durum hakkında kuramsal bilgiye sahip olabilir. Bilgisayarın benzetleyemeyeceği pratik zeka diye adlandırabileceğimiz bir şeydir. Her durum soyut genel özelliklerinin yanısıra, o durumda olan bir insanın kavrayışı ve hedeflerine bağlı olarak yerel bir anlam kazanır.

Dilbilimin başarılarının yapay zekaya genellenebileceğini düşünenlerin kuramsal bilgi ile pratik zeka arasında çok düz bir ilişki görmekte oldukları söylenebilir. İddialara göre, pratik zeka ve genel olarak insan eylemliliğinin, bilimin dünyasını tarif etmekte kullanılan nesnel terimlerle tarif edilebileceğine inanıyorlar. Oysa kuramsal bilgi ile pratik arasındaki bu ilişki, bir fizik kuramı ile teknolojisi arasındaki ilişkiyle eşit düzeyde görülemez. Örneğin füzeleri yönetmede fizik kanunları kullanıldığında, belli bir füzenin hareketi, o genel kanunların örneklenmesinden ibarettir. Oysa, somut bir durumda konuşan insanlar, konuşma faaliyetine, dilsel yeterlilik kuramından özellikle soyutlanmış olan bazı genel ve yerel kabulleri ve hedefleri getirirler. Başka bir deyişle, genel bir yeterlilik kuramı geliştirebilmenin önkoşulu, insanların somut durumlarda konuşabilmesini mümkün kılan özelliklerini gözardı etmektir. Dolayısıyla, insanların somut durumlarda konuşması, füze örneğinde olduğu gibi genel bir yeterlik kuramının örneklenmesi sayılamaz.

## Ontolojik Varsayım

Şimdi sıra, yapay zeka iddialarının en temel varsayımı addedebileceğimiz sonucu varsayıma geldi: Eğer bilgisayar, algılayacak, konuşacak ve genel olarak zekice davranacaksa, işleyeceği verinin ayrık (discrete), acık ve kesin olması gerekir. Aksi takdirde, sözkonusu veri bilgisayara verilebilecek ve dolayısıyla da kurallı olarak işlenebilecek bir veri olmayacaktır. Ancak bu noktada sormamız gereken bir soru daha vardır: İnsanların dünyasını doğru olarak betimleyecek veriler, bu türden veriler olabilir mi?

Yapay zeka iddialarını savunmakta olanlar bu sorunun ancak kısmen farkındadır. İnsanların ''işledikleri'' verilerin, bilgisayarın gerektirdiği veriler gibi, ikili sayı sistemiyle kodlanmaya elvirişli olmadığına işaret edildiğinde, yapay zeka iddialarını savunanların verdikeleri yanıtlardan biri, bilgi-işlem uzmanlarının zekayı benzetleyebilmek için yüksek düzeyli dillerde kurallar içerisinde kodlanamayacağı oluyor. Başka bir deyişle, bu savunuya göre, zeka temsil etmek üzere geliştirilen yazılım düzeyine, bütün akıl yürütme süreçlerinin birtakım elektrik devrelerinin açık/kapalı olması şeklined temsil edildiği olgusu yansımıyor. Bu tamamen donanıma ilişkin bir olgudur ve



Ücretsiz PDF Kitap Arsivi - https://rantey.org

açık/kapalı mantığı hiçbir zaman akış şemalarında görülmez.
Ancak kullanılmakta olan bilgi işlem modeli, makineye verilebilecek bilgi türünü kısıtlamaktadır.
Sayısal bir bilgisayara verilecek bütün bilgiler ayrık (discrete) bir biçimde olmak zorundadırlar.
Bu ise özel bir sorunun ortaya çıkmasına yolaçar; daha doğrusu, bilgisayara sorulacak tüm soruların, verilecek tüm bilgilerin ayrık bir biçimde kodlanmak zorunda olması kendi başına özel bir sorun yaratır. Sorunu en tarafsız biçimiyle şöyle özetleyebiliriz. Somut bir durumda kullanılan bir cümleyi, bir problemin yapısını anlayabilmek ya da bir örüntüyü tanıyabilmek için bilgisayarın da, bir insan gibi, verileri yorumlamasını sağlayacak somut bir bağlama **(context)** ihtiyacı vardır. Bu bağlamın kendisini bilgisayara nasıl vereceğiz? Bu sorunun -gene tarafsız terimlerle- en çarpıcı ifadesi Eden'ın bilgisayarlara elyazılarını tanıtma çalışmaları ile ilgili söylediklerinde bulunabilir:
''Bir insan, okunaksız bir yazıyla yazılmış bir mektubu okurken, dilin gramerinden, metnin okuyamadığının kadarıyla anlamından, konunun niteliğinden ve belki de yazarın ruh halinden hareketle okuyamadığı da anlamını yeniden kurabilir. Ama maalesef yazı karakterleri bilgisini aşmakla birlikte bu karakterlerin tanınmasına katkıda bulunan bu dünyevi bilginin, nasıl düzenlenip nasıl bilgisayara yüklenebileceğine dair elimizde hiçbir ipucu yoktur.''
Burada Eden'ın ''dünyevi bilgi'' ile tam olarak neyi kasdettiğini açıklamaması boşuna değil. Oysa ontolojik varsayımdan ötürü, yapay zeka iddialarına kaynaklık eden sayısal bilgi işlem modeli bu soruya, artık tarafsız olmak yerine sayısal bilgisayarların gerekleriyle tutarlı bir cevap dikte eder. Başka bir deyişle, yapay zeka iddialarını doğru olabilmesi için, dünyevi bilgimizin, dünya hakkındaki bilgimizin tamamının büyük bir ayrık olgular kütlesi olması gerekir.
Bu yüzden, çoğu zaman yapay zeka sorunu bir bellek hacmi sorunu olarak ortaya konabilmektedir. Örneğin, yapay zeka alanında çalışanların en önde gelenlerinden biri olan Marvin Minsky, kendisi için çok önem taşıyan ''İnsansı bir zeka için gereken bilgi kütlesinin hacmi nedir?'' sorusunu şu şekilde cevaplamaktadır:
''Eğer bir uzmanın bilgisinden değil de sıradan, gündelik bilgiden, herkesin gündelik hayatını sürdürebilmek için sahip olmak zorunda olduğu bilgiden sözediyorsak, ilk önce bazı vazgeçilmez ve herbiri oldukça karmaşık olan bir kategoriler dizisi ile karşılaşıyoruz: Eşyanın ve mekânın geometrik ve mekanik özellikleri; birkaç bin nesnenin özellikleri ve kullanılışları, ne işe yaradıkları; yüzlerce insan hakkında yüzlerce bilgi ve onlarca insan hakkında binlerce bilgi; yüzlerce kuruluş, örgüt ve kurum hakkında yüzlerce bilgi. Bunlar sınıflandırılmaya başlanınca önce kategoriler hızla artıyor fakat bir süre sonra sınıflandırma giderek güçleşiyor. Şimdilik çok sağlam olmamakla birlikte benim izlenimim, birbirleri arasında ''bağ''lar **(links)** olan ondan fazla alan olmadığı yolunda. İnsanın hakkında binlerce şey bildiği yüzlerce şey olamaz: ya da herbiri yüz ''bağ''a sahip binlerce şey. Dolayısıyla sıradan durumlarda kabul edilebilir sağduyu sınırları içerisinde davranacak bir makinenin yüzbinler dolayında bilgi edinmesi gerekecektir. Gerçekten hatırı sayılı bir zeka içinse bir milyon bilgi gerekecektir. Eğer söylediklerimi ikna edici bulmadıysanız, verdiğim bütün sayıları onla çarpın.''

Son cümlesinden de anlaşılabileceği gibi, Minsky için yapay zeka ile ilgili sorun ancak bir bellek ve zeka için gereken veri tabanının hacmi sorunu olabilir. Çünkü sayısal bilgi işlem modeli içerisinde, hakkında bilgi sahibi olduğumuz olgular, ortamdan edindiğimiz verilerle, bu verileri yorumlamamızı sağlayacak çerçeveyi oluşturan bağlam arasında kavramsal bir ayırım yapmaya olanak yoktur. Birtakım verileri yorumlamak için başka birtakım şeylere ihtiyacımız varsa, bu başka şeyler de, sayısal bilgi işlem modeli içerisinde ancak yine sayısal (ayrık) olarak kodlanmış veriler olabilir. Ayrık veriler halinde kodlanmamış, insan ihtiyaçlarından, hedeflerinden, tasarılarından, sezgilerinden oluşan ve herhangi bir veri ya da nesnenin yorumlanmasını mümkün kılan bir faaliyetler bağlamı kavramı, sayısal bilgi işlem modeli içerisine anlamlandırılabilecek bir kavram değildir.

Oysa genel olarak, belirgin bazı şeyleri, belli bazı nesneler hakkındaki bilgimizi açık seçik bir şekilde dile getirmemizi ve hatta formelleştirmemizi mümkün kılan, insanlık durumu hakkında örtük bir anlayış ve anlayışlılık olabilir. Herhangi bir durum hakkında açık seçik olarak dile getirebileceğimiz ya da formelleştirebileceğimiz her şeyin, biz onu dile getirmeden de, bilinçsiz bir biçimde formel bir ''model yapısı'' içerisinde yeraldıklarını söylemek için,



Ücretsiz PDF Kitap Arsivi - https://rantey.org

ontolojik varsayımın ötesinde hiçbir neden yoktur.

## Ontolojik Varsayımın Tarihi

Bu ontolojik varsayım yapay zeka alanında çalışanlara niçin bu kadar apaçık ve kabul edilebilir gözüküyor? Bu görüşün oldukça geleneksel olduğunu ve kaynaklarının felsefi düşüncenin doğuşuna gittiğini görmüştük.

Daha önce de söylediğimiz gibi felsefi Batı düşüncesine egemen olan tasa, ahlaki olsun, kurumsal olsun, pratik olsun her alanda belirsizliği aşmak, kesinliğe ulaşmaktır. Bilginin, yorumlamak gerektirmeyecek açıklıkta dile getirilmiş kurallar ya da tanımlarla ifade edilmesi gerektiği yolundaki inanç ta Platon'a kadar gitmektedir. Aynı şekilde bu kuralların üzerine uygulanacağı unsurların da yalın, birbirlerinde bağımsız nesneler olması gerektiği yolundaki inanç da.

17. yüzyıla gelindiğinde Leibniz'de bu geleneksel bilgi anlayışıyla, dünyanın da bu şekildeki ayrık nesnelere çözümlenebilmesi gerektiği yolundaki görüş açık olarak birbirlerine bağlanır.



**Yapay zekâ iddialarının doğru olabilmesi için dünya hakkındaki bilgimizin tamamının birbirinden bağımsız olarak tanımlanmış nesneler hakkında, yine birbirinden mantıken bağımsız olan önermelerle ifade edilebilecek özellikler şeklinde olması gerekir. Bu görüşün dışarda bıraktığı, somut bir durumun iç tutarlılığının bir nesneler ve olgular listesinden farklılığını ifade eden ''bağlam'' (context) kavramıdır.**

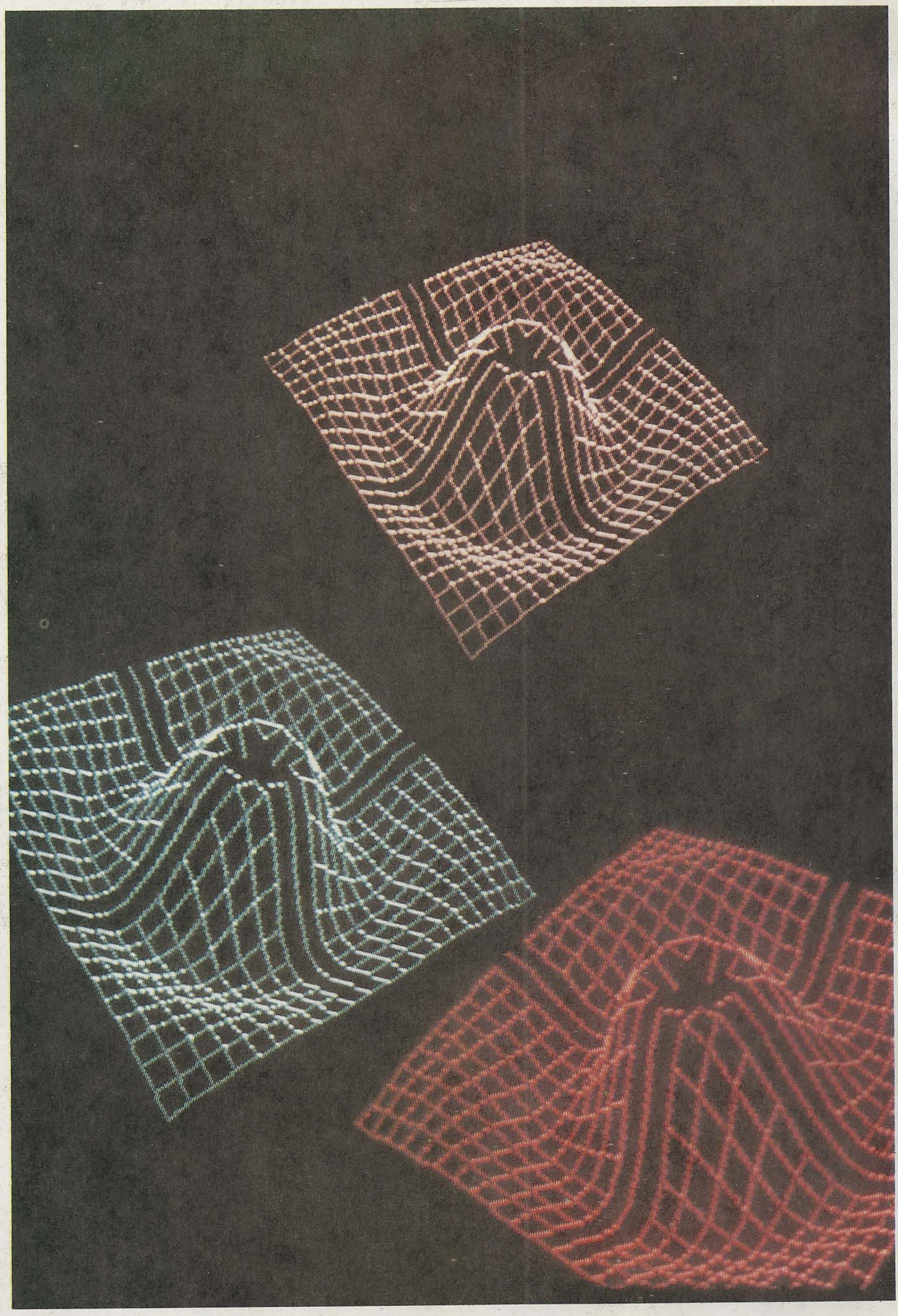
1776

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Leibniz'e göre bir şeyi anlamak demek, belli bir kavramı yalın unsurlarına ayrıştırmak demektir. Giderek yalınlaşan unsurlardan oluşan sonsuz bir diziden kurtulmak için de, her kavramın ayrıştırılabileceği bir nihai en yalın unsurlar düzeyi bulunmalıdır. Dahası, bu kavramların dünyadan sözetmesi mümkün olacaksa, dünyada, kavramların bu en yalın unsurlarına tekabül eden yalın nesneler olması gerekir. Bilgisayar fikrinin de atalarından olan Leibniz ''insan düşüncesi için bir evrensel alfabe'' tasarlıyordu, öyle ki, bu alfabenin harflerinin kullanılış tarzı, dünyadaki nesnelerin sınıflandırılması, ilişkileri ve düzenleri hakkında bir şey gösterebilsin.

Daha önce değindiğimiz ampirisist gelenekte de bilgi hakkında benzer görüşlere rastlıyoruz. Hume için mümkün her yaşantı izlenimlerden oluşur: Birbirinden ayrılabilir, yalın ve belirgin yaşantı atomları. Akılcı ve ampirisist gelenekler 20. yüzyıl İngiliz filozofu Bertran Russel'da mantıksal atomizm görüşünde içiçe geçti ve genç Wittgenstein'ın *Tractatus Logico Philosophicus* adlı eserinde en gelişmiş ifadesini buldu. Burada dünyanın tamamı, birbirinden bağımsız mantıki önermelerle ifade edilen atomik olgulardan oluşuyormuşcasına değerlendirilir. Bu ontolojik varsayımın en açık ifadesi ve bütün yapay zeka çalışmalarının önkoşuludur. Böylelikle yapay zekada felsefe ve teknoloji, Platon'dan beri insanların filozofların aramakta olduğu şeyi bulmuş olduklarını ilan ettiler: Açık seçikliğin, kesinliğin ve kontrol imkanının garanti altına alındığı, veri tabanlarından, karar kurallarından ve otomasyondan oluşan bir dünya.





1778

# Ontolojik Varsayımın Ötesinde

Ontolojik varsayımın en etkin ifadesini bulmuş olduğu bu yıllar (1920'ler ve sonrası), aynı zamanda bu varsayımın en köklü bir biçimde sorgulanmaya ve insanların dünyasının klasik ontolojik varsayımın öngördüğünün dışında bir biçimde tarif edilmesi gerektiği olasılığının en ciddi biçimde araştırılmaya başladığı yıllar oldu.

Gerçi ontolojik varsayımın egemenliği altında sürmüş olan felsefî Batı düşüncesi tarihi boyunca, bu varsayımı kabullenmemiş tek tük düşünürler olmuştu. Örneğin 17. yüzyılda, bir yandan Newton fiziğinin başarısı, insanların ontolojik varsayıma olan inançlarını, insan davranışları da dahil olmak üzere dünyanın tamamının formel bir sistem çerçevesi içinde betimlenebilmesinin artık gün meselesi olduğuna duydukları güveni pekiştirirken, diğer yandan kendisi de bir hesaplama aygıtı tasarlamış olan Fransız filozof ve matematikçisi Blaise Pascal, doğal bilimlerin kullandığı formel sistemlerin ancak sınırlı gerçeklik alanlarını betimlemekte kullanılabileceğini iddia etti.

Ama ancak 20. yüzyılda, dünyanın tamamının ayrık bir nesneler, özellikler ve olgular kümesi olması gerektiği yolundaki ontolojik varsayımın yerine, alternatif bir ontolojik varsayım geliştirebilme olanakları sistematik bir biçimde araştırılmaya başlandı.

Bu araştırma çabalarını uyaran, birbiriyle neredeyse zıt yönde iki gelişme olmuştur. Bunlardan Gödel teoremi olarak anılan birincisi, matematik, ya da daha özgül olarak sayı kuramı gibi, en formel gözüken sistemlerin bile tamamen formelleştirilemeyeceği iddiasına inanılırlık kazandırdı.

İkincisi ise önce yeni gelişmekte olan fenomenoloji akımının, daha sonra ise, yapay zeka çalışmalarının kendisinin, formelleştirme çabalarına en çok güçlük çıkaran davranışların, teorem kanıtlamak, gibi nispeten uzmanlaşmış ve ''yüksek düzeyli'' davranışlar değil, bir bira ısmarlamak gibi en sıradan davranışlar olduğunu göstermesiydi.

Bunları, şimdi, sırayla özetleyelim:

## Gödel Teoremi

Gödel teoremi, Avusturyalı matematikçi Kurt Gödel'in 1931'de yayınlanan *Principia Mathematica'da ve Benzer Sistemlerdeki Formel Olarak Karar Verilemeyecek Önermeler Üzerine* adlı yazısında yayınlandı. Teoremin en genel ifadesi şu şekilde verilebilir:

> **Sayı kuramının tutarlı bir aksiyomatizasyonu veya bu türden herhangi bir sistem, tamamlanamaz; yani sisteme ait olup olmadığına karar verilemeyecek önermeler içerir.**

Bu teoremin ne anlama geldiğini ve niçin bu kadar önemli olduğunu görmek için, dönemin entelektüel atmosferine ve teoremin ne tür kaygularla dile getirildiğine bakmakta yarar var.

## Gödel Teoreminin Tarih-Öncesi

*Principia Mathematica*, Bertrand Russel ve Alfred North Whitehead'ın 1910-1913 yılları arasında yayınladıkları ve matematiğin mantıksal temelini formelleştirmeye çalıştıkları kitabın adıydı. Bu kitap, daha önceki bölümlerde gözden geçirdiğimiz ve epistemoloji ve ontolojik varsayımların doğru olduğunu gösterme çabasının merkezi bir yer işgal ettiği düşünce geleneğinin içersinde yer alıyordu. Daha açık bir deyişle, yapay zeka çalışmalarından önce yapılmış ve insanın akılyürütme sürecinin mekanik bir betimlemesini verme çabalarından biriydi.

Yüzyıllar boyunca bu tür çabaların en önemlileri aynı zamanda ilkeleri olarak kaldı; Aristoteles'in tümdengelimsel **(deductive)** kıyası formelleştirmesi ve Euclid'in geometriyi formelleştirmesi.

Ancak 19. yüzyılda, Euclid'in formelleştirdiği geometrinin dışında bazı geometrilerin de mümkün olduğunun gösterilmesiyle çok önemli bir adım atılmış oldu. (Burada ''bir geometri'' ile kasdettiğimiz, soyut nokta ve çizgilerin özellikleri hakkında formel bir kuramdır.)

Yüzyıllar boyunca geometri ile Euclid'in formelleştirmesinin eşanlamlı şeyler olduğu, Euclid'in sisteminde ufak tefak pürüzler olsa da, geometrideki herhangi bir ilerlemenin Euclid'in sisteminin kapsamının genişletilmesiyle olacağına inanıldı. Bu inançlar, 19. yüzyılda birbirinden bağımsız olarak birkaç insanın, Euclid'in formelleştirmesine uymayan geo-

metriler keşfetmesiyle köklü bir biçimde sarsıldı ve neredeyse entelektüel bir kriz yaşanmaya başladı. Çünkü farklı geometrilerin varlığı, matematiğin gerçek dünyanın özelliklerini araştırdığı yolundaki yerleşik inanca gölge düşürüyordu. Nasıl olurdu da aynı gerçeklik içersinde farklı türden noktalar ve çizgiler yeralabilirdi.

Yine 19. yüzyıl içersinde kıyasın ve diğer akılyürütme biçimlerinin formelleştirilmesi konusunda da önemi adımlar atıldı. Bu alanda İngiliz mantıkçıları George Boole ve Augustus de Morgan'ın katkılarına zaten değinmiştik. Yüzyılın sonuna doğru ise Jena'da Gottlob Frege ile Turin'deki Giuseppe Peano, mantık ve matematik alanlarındaki, o zamana kadar bağımsız seyretmiş gelişmeleri birleştirdiler. Her ikisi de tümdengelimsel kıyas ve diğer formel akılyürütme türleri üzerinde yapılan çalışmaları, sayılar ve kümeler **(sets)** üzerinde yapılan çalışmalara ışığında değerlendirmeye çalıştı. Aynı yıllarda Almanya'da Göttingen'de David Hilbert geometrinin formelleştirilmesinde Euclid'inkine kıyasla daha kesin yollar araştırıyordu. Bütün bu çalışmaların ortak bir yönü vardı; çeşitli alanlarda, bir önermenin kanıtlandığı söylendiği, yani doğruluğu bilinen başka önermelerden hareketle sözkonusu önermenin de doğruluğu tartışma götürmez bir biçimde karara bağlandığı zaman yapılan işin ne olduğu, bu işin herhangi bir kişi ya da makine tarafından da yapılabilmesini sağlayacak şekilde nasıl betimlenebileceği hakkında açık seçik ve formel cevaplar arıyorlardı.

Diğer yandan, bu dönemde klasik matematik alanında da ilginç gelişmeler oluyordu. Sonsuzluk kavramı, büyük bir olasılıkla Hintlilerden beri matematikçilerin zihnini işgal etmiş kavramlardan biriydi. 1880'lerde George Cantor farklı farklı sonsuzluk türlerinin anlamlarının ne olabileceği ve matematikte nasıl kullanılabilecekleri hakkında yeni, güçlü ve yalın bir kuram geliştirdi. ''Kümeler kuramı'' **(set theory)** aslında sınıflandırma işlemini veri kabul ediyor ve sınıflandırmalar sonucunda elde ettiğimiz kümeler arasındaki formel ilişkileri araştırıyordu. Cantor'un kuramının en önemli yanı bu bağlam içerisinde sonsuzlukla ilgili çeşitli sorunları da dile getirebilmesi ve inceleyebilmesiydi. Ancak kuram sezgisel değildi - bazı sonuçları sağduyuya aykırı gözüküyordu. Çok geçmeden kümeler kuramı çevresinde bir dizi paradoks keşfedildi. Bu paradokslar, matematik türü sistemlerin tamamen formelleştirilemeyeceğini kanıtlayan Gödel teoremi ile yakından ilgili olduğundan, birkaç örnek üzerinde durmakta yarar var.

## Paradokslar

Bu paradoksların en ünlüsü, daha önce değindiğimiz İngiliz filozofu Bertrand Russel'ın adıyla anılanıdır. Bu paradoksun nasıl ortaya çıktığını anlamak için ''sıradan'' ve ''kendi kendini içeren'' kümeler arasındaki farkı gözden geçirmemiz lazım:

Sıradan kümelerin temel özelliği kendi kendilerini içermemeleridir. Başka bir deyişle, küme kendi kendisinin elemanı değildir. Örneğin ''mevcut bütün köstebekler'' diye bir kümeden sözettiğimizi düşünelim. Bu terimle (''mevcut bütün köstebeklerin kümesi'' terimiyle) işaret ettiğimiz şeyin kendisi, bir köstebek değildir. Aynı şekilde yalnızca Halit Ziya Uşaklıgil'i içeren küme de sıradandır - çünkü Halit Ziya'nın kendisini içeren küme, Halit Ziya'nın kümesini içermez (bir küme bir insan değildir).

Oysa, ''kendi kendini içeren kümeler'', bu addan da anlaşılabileceği gibi, kendi kendilerini içerirler. Başka bir deyişle, kendi kendini içeren bir kümenin içerdiği elemanların tüketici bir dökümü, bu kümenin kendisini de içermek durumundadır. Örneğin bütün kümelerden oluşan kümeyi düşünelim. Gerçekten mevcut ve mümkün bütün kümeleri kastediyorsak, o zaman bu kümenin elemanlarından biri de kendisi olmak zorundadır. Aynı şekilde Halit Ziya'nın dışında kalan her şey, dediğimiz zaman da kendi kendini içeren bir kümeden sözediyoruz: Dünyada Halit Ziya olmayan şeylerden bazıları, kümelerdir; daha da özgül olarak Halit Ziya olmayan şeylerden bir tanesi de, Halit Ziya olmayan şeyleri içeren kümedir.

Buraya kadar bu ayrım, formelleştirilebilen ve dolayısıyla sorun yaratmayan bir ayrımdır. Ancak bu ayrımı veri kabul etmek yeterli olmuyor, çünkü öyle bazı kümeler düşünmek mümkün ki, bazılarını ne sıradan kümeler ne de kendi kendini içeren kümeler arasında saymak mümkün değildir. Örneğin K diye bir kümeyi, ''bütün sıradan kümelerin oluşturduğu küme'' diye tanımlayabiliriz. K'nın kendisi sıradan bir küme midir, yoksa kendi kendini içeren bir küme mi? Bu soruya nasıl cevap verirseniz verin, bir paradoksla karşılaşacaksınız. Eğer K sıradan bir kümeyse, o zaman ''bütün sıradan kümelerin oluşturduğu küme''nin elemanlarından birisinin de K olması gerekir. Bu ise K'nın sıradan (yani kendi kendini içerme-

yen bir küme) tanımıyla çelişecektir. Yok K, kendi kendini içeren bir küme olarak tanımlanırsa, o zaman da ''bütün sıradan kümelerin oluşturduğu küme''nin elemanlarından biri, sıradan olmayan bir küme (yani K'nın kendisi) olacaktır.

Yapılabilecek tek şey K'yı ''marazî'' bir küme olarak tanımlamak. Ama formel bir sistem için, bu tür mecazi tanımlamalar yeterli olmaktan çok uzaktır.

Merkezî sorun küme ve sınıf hakkındaki, gündelik düşünüşümüzü yönlendiren sezgisel fikir ve inanışlarla formellik gereklerinin uyuşmazlığıdır. Sorunu kısaca şöyle ifade edebiliriz: ''Gündelik sezgilerimizle uyuşan, ama aynı zamanda da paradokslara düşmeyen bir kümeler kuramı geliştirebilir miyiz?'' Sorun sadece sınıflandırma ve kümeler alanına özgü değildir; sezgilerle formellik gereklerinin uyuşturulması sorunu, sayı kuramı ve geometri gibi, ilk bakışta formel görünen, diğer alanlarda da karşımıza çıkmaktadır.

Russel paradoksunun şaşırtıcı bir versiyonu da Grelling paradoksu adıyla bilinir. Bu paradoksta kümelerdense tamlamaları ve sıfatlar kullanılır. Mümkün bütün tamlama ve sıfatları ikiye ayırdığımızı düşünelim; öyle ki, birinci grup kendi kendisini betimleyen tamlama ve sıfatlardan, ikinci grupsa diğerlerinden oluşsun. Birinci gruba örnek olarak ''dört heceli'' ya da ''uzatılmış'' gibi tamlama ve sıfatları verebiliriz; ikinci grubaysa ''siyah'', ''iki heceli'' ya ''tutarsız'' gibi tamlama ve sıfatları. Paradoks ''kendi kendini betimlemeyen'' gibi tamlamalarla ortaya çıkmaktadır.

Ama bu tür paradoksların en eskisi ve ünlüsünün tarihi, kümeler kuramından çok daha eskilere uzanmaktadır: Giritli yalancı ya da Epimenides paradoksu. Epimenides tarihe tek bir ünlü cümle bırakan bir Giritlidir. Cümlesi şu:

Bütün Giritliler yalancıdır.

**O halde sorunu bir kez daha şöyle dile getirebiliriz.**

**İnsan düşüncesini bildik mekanik süreçlerden ayıran en önemli özelliklerden birisi, insan düşüncesinin kendi üzerine dönüşlü olma özelliği. Ama diğer yandan formel sistemlerde, kendi üzerine dönüşlü olma özelliği, paradoksların doğmasına yol açan başlıca etken. O kadar ki, herhangi bir sınıflandırma sistemini ya da sayı kuramı ya da geometri gibi akıl yürütme tarzını formelleştirmenin ön koşulu, bu sistemi kendi üzerine dönüşlülük özelliğinden arındırmakmış gibi gözüküyor.**

Aslında burada sözkonusu olan paradoksu şu cümle daha da çarpıcı bir biçimde ortaya koymaktadır:

Bu cümle yanlıştır.

Bu cümleye doğru veya yanlış herhangi bir doğruluk değeri vermek paradoksa yol açacaktır.

Buraya kadar gözden geçirdiğimiz bütün paradoksların ortak bir özelliği var: Hepsi kendi kendine gönderme yapan ya da kendi üzerine dönüşlü olan niteleme, betimleme ya da kümelerle ilgili olarak ortaya çıkıyor. Ancak kendi üzerine dönüşlü olma özelliği ile daha önce de, buradakinden çok farklı olmakla birlikte çok önemli bir bağlamda karşılaşmıştık. Karar verme yordamlarını inceler ve karar verme yordamları ile aksiyom ve teoremlerin farkını açıklarken, insan düşüncesini mekanik süreçlerden ayıran en önemli özelliğin, kendi üzerine dönüşlü olmak olduğuna değinmiş, bir dizi kuralı görünüşte mekanik bir biçimde uygularken bile, insan düşüncesinin bu kuralları uygulamakta olduğunun bilincinde olduğunu, bu süreci kaydettiğini ve gerektiğinde bu süreçlerden yeni genellemelere ulaştığını görmüştük. Şimdi bu özellik üzerinde kısaca biraz daha duralım.

Öyle görünüyor ki, en basit olgusal cümlenin bile insan düşüncesinde oynadığı rolü ifade edebilmek için, aslında iki önermeye ihtiyacımız var. Biri cümlenin içeriğini ifade eden bir önerme, diğeriyse cümleyi söyleyen kişinin cümlenin içeriği hakkındaki farkındalığını ifade eden bir önerme. ''Hava sıcak'' ya da ''su yüz derecede kaynar'' gibi olgusal bir cümle ele alalım. Bu cümlelerin insanlar tarafından işlenmesini doğru olarak yansıtabilmek için, hem cümlenin kendisine hem de cümleyi söyleyen kişinin cümlenin doğru olduğuna inandığını (cümlenin doğruluğunun farkında olduğunu) ifade eden ikinci bir cümleye ihtiyacımız var:

i) Hava sıcak.
ii) Ben şu anda ''hava sıcak'' cümlesinin doğru olduğuna inanıyorum.

Burada dikkat edilmesi gereken nokta (ii)'nin anlamını açımlamaya çalıştığımız cümlenin kendisine gönderme yapması, yani cümlenin kendi üzerine dönüşlü olması.

## Tipler Kuramı

Gerçekten de Russel ve Whitehead bu görüşü veri kabul ettiler. Bir bakıma *Principia Mathematica*'nın tamamı, mantığı, sayı kuramını ve kümeler kuramını kendine üzerine dönüşlülük özelliğinden arındırma çabasıdır.

Bunu başarmak için geliştirdikleri kuram **''Tipler Kuramı''** adıyla bilinir. Bu kuramda her kümenin belli bir tipi olduğu ya da belli bir düzeye ait olduğu varsayılır. Öyle ki, her düzeydeki küme, eleman olarak yalnızca kendinden bir alt düzeydeki elemanları içerebilir. Buna göre en düşük düzeydeki kümeler, eleman olarak başka kümeleri içeremez, yalnızca ''nesne''leri **(objects)** içerebilirler. Bir üst düzeydeki kümeler ise eleman olarak ancak en düşük düzeydeki kümeleri içerirler.

Görüldüğü gibi bu kuram kümelere bir hiyarşi empoze etmektedir. Bu hiyerarşi aynı zamanda ''kendi kendini içeren kümeler''i olanaksız kılmaktadır. Her küme ancak bir alt düzeyden elemanlar içerebileceğinden kendisini (kendi kendisiyle aynı düzeyden olduğundan) içeremez.

Tipler Kuramı, Russel paradoksunun ortaya çıkmasını önlemekle birlikte, yalnızca kümelerle ilgili olduğundan, Grelling ve Giritli Yalancı paradoksları ile ilgili hiçbir çözüm öneremez. İlgileri kümeler kuramıyla sınırlı olan biri için, kuşkusuz, bu kadarı yeterlidir. Ancak genel olarak formelleştirilmiş düşünce süreçlerindeki bütün paradokslardan kurtulmak için, mümkün bütün dillerde kendi üzerine dönüşlülüğü engelleyen benzer bir hiyerarşik yapıya gerek vardır. Böyle bir hiyeyarşinin en alt düzeyde bir nesne dilinin bulunması gerekiyor, öyle ki bu düzeyde ancak sınırlı bir gerçeklik alanından bahseden cümleler türetmek mümkün olsun. Bu düzeyde cümlelerin kendisini ya da çeşitli özelliklerini -gramer kurallarını ya da belli cümleleri- konu edinen cümleler olanaksız olacaktır. Bu amaç için üst-dil **(metalanguage)** diye adlandırabileceğimiz ikinci bir düzey olacaktır.

Bu şekilde tanımlanmış iki ayrı dil düzeyi, yabancı dil öğrenmiş olan herkesin yakînen tanıdığı bir şeydir. Ancak bu, yine de dil düzeylerinin bu şekilde tanımlanmasının bir sonsuz döngü tehlikesi içerdiği gerçeğini değiştirmez. Çünkü üst-dilin kendisi de hakkında konuşabilecek birşey olduğuna göre, bu sefer de bir üst-üst-dile gerek duyulacaktır. Bu formelleştirme tarzına göre her cümlenin açıkça tanımlanmış belli bir düzeye ait olması ge-

rekiyor ve bu şekilde tanımlanmış belli bir düzeye ait olduğu gösterilemeyen her cümlenin ise anlamsız olduğuna karar veriliyor.

Gündelik hayatımızda sürekli olarak kullanmadığımız soyutlamalar üzerine inşa edilmiş olan kümeler kuramında bu tür bir hiyerarşi -biraz tuhaf kaçsa da- kabul edilebilir. Ama sözkonusu olan dil gibi hayatımızın her alanını kuşatan bir olgu olduğunda, böyle hiyerarşiler açıkça yetersiz ve hatta gülünç olmaktadır. Gündelik konuşmalarımız sırasında tek bir dil kullanırız; bir diller düzeyi içerisinde bir aşağı bir yukarı düzeye sıçrayıp durmayız. Bir örnek verelim: Böyle bir diller hiyerarşisi ''Bu metinde ben tipler kuramını eleştiriyorum'' gibi sıradan bir cümleyi iki gerekçeden ötürü yasaklayacaktır. Birincisi ''bu metin'' teriminden ötürü - çünkü ''bu metin'', kendisi metin olan bir şeyin içersinde kullanılamaz; ''bu metin''den sözedebilmek için bir ''üst-metin''e sahip olmamız gerekir. Aynı şekilde ''ben'' teriminin bütün kullanımları, zorunlu olarak konuşan kişinin kendi kendine gönderme yaptığından Tipler Kuramı gibi bir hiyerarşi tarafından anlamsız sayılacaktır.

Bu örnek, hem kuramın sıradan ve tanıdık bir bağlama aktarıldığında ne kadar sağduyuya aykırı sonuçlar verdiğini, hem de kendi kendine gönderme yapmanın, kendi üzerine dönüşlülüğün gündelik hayatımızda ne kadar merkezî bir yer tuttuğunu gösteriyor. Dolayısıyla, her tür kendi kendine gönderme yapma olgusunu, sistemi paradokslardan arındırmak uğruna anlamsız saymak, istenenden çok daha fazlasını gerçekleştiren bir çözümdür.

Kuşkusuz Tipler Kuramı'nın gündelik dille ilgisi olmadığı, yalnızca formel dillere ilişkin bazı sorunları çözmek için geliştirildiği söylenebilir. Ancak formel sistemlerin geliştirilmesinin temelinde yatan güdü de, son çözümde, sıradan, gündelik insan düşüncesini daha anlaşılabilir kılmak ve benzetleyebilmekti.

## Matematik, Mantık ve Üst-Matematik

1920'lerde bu tür sorunların farkına varılması, matematikçi ve filozofları sayılar kuramı gibi, matematiğin o güne kadar en somut ve en sağlam alanı olarak sayılagelmiş alanlarının dahi anlaşılabilirliğinden ve temellerinin sağlamlığından kuşku duymaya yöneltti. Kümeler Kuramı'nın temelinde yatan ''küme'' kavramı, çok açık, sağduyuya çok uygun bir kavram gibi görünüyordu. Buna rağmen, kümeler kuramında bu tür paradokslarla karşılaşılabiliyorsa, zaten daha sorunlu olan başka alanlar nasıl formelleştirilebilirdi? Benzer bir kaygı da mantıkta karşılaşılan Giritli Yalancı türünden paradokslarla ilgili olarak duyuluyordu. Eğer bu tür paradoksların mantığa da içkin olduğu keşfedilirse, bu hem mantığın hem matematiğin sağlamlığına, kesinliğine büyük gölge düşürecekti. Bu özellikle, matematik ve mantığın, her ikisinin de birer formel sistem olması gerekçesiyle birbirlerine çevrilebileceğine, hatta bir ve aynı şey olduklarına inananlar için özellikle proplematik bir konuydu ki, bu inancı paylaşanlar da o yıllarda çoğunluktaydı. Dolayısıyla, bu yıllarda tartışmalar matematik ve mantığın bir mi ayrı mı olduğu sorusu çevresinde yoğunlaştı.

Bu alanda yapılan çalışmalar üst-matematik (metamathematics) -ya da kimi zaman, mantıkla matematik o kadar iç içe geçmiş olduğundan, üst-mantık **(metalogic)-** diye adlandırıldı. Üst-matematikçiler için en acil sorun matematiksel akılyürütmenin doğasının ne olduğuydu. O güne kadar matematiksel akılyürütme hep doğal dillerden (Latince, Arapça, Türkçe vs.) biri içersinde yapılmış olduğundan, muğlaklık payı hep yüksek oluyordu. Sözcükler farklı insanlar için farklı anlamlar ifade ediyor, farklı imgeler çağrıştırıyorlardı. Dolayısıyla mümkün bütün matematiksel işlem ve akılyürütmelerde kullanılabilecek tek bir formel dil herkes tarafından çok önemseniyordu. Bu ise mümkün bütün düşünce ve akılyürütme tarzlarının, özellikle de matematikte kullanılanların formelleştirilmesini gerektiriyordu.

*Principia Mathematica*'nın bu kadar önemsenmesinin ve hayranlık uyandırmasının temelinde yatan nedenler de işte bunlardı. Ancak kimse

a) Russel ve Whitehead'in açıkladığı yöntemlerin matematiğin tamamını kapsayıp kapsamadığından,
b) Ve hatta verilen yöntemlerin kendi içlerinde bir tutarlılığa sahip olup olmadıklarından emin değildi.

Russel ve Whitehead'in açıkladığı yöntemler izlendiğinde, birbirleriyle çelişmeyen sonuçların asla türetilemeyeceği gerçekten kesin miydi?

Bu soru özellikle Alman matematikçisi (ve üst-matematikçisi) David Hilbert'i rahatsız ediyordu. Bu yüzden Hilbert dünyadaki bütün matematikçi ve üst-matematikçileri şu soru üzerinde çalışmaya çağırdı: *Principia Mathematica*'daki sistemin hem tutarlı, hem de tamam-

lanmış olduğunu göstermek. İşte Gödel'in teoreminin yerel sayılabilecek önemi bu tasarıya son vermesiydi. Çünkü Gödel,1 931'de yayınladığı yazıyla, değil Principia'daki sistemin, benzer hiçbir sistemin tamamlanmış olamayacağını gösteriyordu. Ancak Gödel'in bunu nasıl yaptığını görmeden önce, ''tutarlılık'' **(consistency)** ve ''tamamlanmışlık'' **(completeness)** terimleriyle ne kasdedildiği üzerinde durmamız gerekiyor.

## Tutarlılık

Formel sistemlerle ilgili olarak iki tür tutarlılıktan sözedebiliriz:

a) Dış tutarlılık
b) İç tutarlılık

Dış tutarlılık formel sistemlerin yorumlanmasıyla **(interpretation)** ilgili olarak ortaya çıkan (ya da çıkmayan) bir olgudur. Formel sistemlerin ve kuralların kendi başlarına bir anlamları ya da yorumları yoktur. Örneğin IMU sistemi ile ilgili olarak alıştırmalarımızı yaparken, I, M ve U simgelerine herhangi bir anlam yüklemedik. Gerçi daha sonra bu simgelerin, Türkçe'ye benzer, ancak daha ilkel bir dilin, sözcük türlerine işaret ediyormuşcasına yorumlanabileceğini gördük, ama sistem içersinde yeni diziler türetirken, bu yoruma başvurmamız gerekmiyordu.

Aslında herhangi bir dizi nesneyi sayarken de yaptığımız, belli bir formel sistemi, tamsayılar sistemini, yorumlamaktan ibarettir. Burada her simgeye (yani sayının) saymakta olduğumuz nesnenin belli bir niceliğine karşı gelecek şekilde yorumlarız.

Bir formel sistemi her zaman farklı farklı biçimlerde yorumlamak mümkündür. Bu olanağı bir örnek üzerinde görelim: Farzedelim ki elimizde tek bir yüklemden (X) ve üç sabitten (a, b

**Kısaca özetleyelim:**
**Tutarlılığın iki farklı şekilde kavranabileceğini söyledik: Birinci kavrayışta, bir formel-sistem-artı-bir-yorumun eğer yorumlanmış her teoremin doğru olduğu keşfediliyorsa, dış dünya ile tutarlı olduğu söylenir. İkinci kavrayışa göre ise, bir formel-sistem-artı-bir-yorumun, ancak yorumlanmış bütün teoremleri birbirleriyle uyarlı olduğu durumlarda da iç tutarlılığa sahip olduğu söylenebilir.**

ve c) luşan bir formel sistem var. Sistemin teoremleri şunlar:

i) X(a,b)
ii) X(b,c)
iii) X(c,a)

Şimdi bu sistemde X'i ''döver'' yüklemi, a, b ve c'yi de üç insanın ismi (Ahmet, Ali, Mustafa) olarak yorumlayalım. Bu takdirde, formel sistemimizin dış tutarlılığı, gerçekten dış dünyada Ahmet, Ali ve Mustafa'nın arasındaki ilişkilerin, bu şekilde yorumlanmış formel sistemimizin öngördüğü biçimde olup olmamasına bağlıdır. Yani, gerçekten Ahmet Ali'yi, Ali Mustafa'yı, Mustafa da Ahmet'i dövüyorsa, sistemimiz dış tutarlılığa sahip demektir.

Görüldüğü gibi dış tutarlılık kavramının ''doğruluk'' kavramıyla yakın bir ilgisi vardır. Eğer belli bir yorumun sonucunda, bir formel sistemin bütün önermelerinin doğru olduğu görülüyorsa, o formel sistemin dış tutarlılığa sahip olduğu söylenir.

Şimdi farzedelim ki dış dünyada olaylar bizim yorumlanmış formel sistemimizin öngördüğü gibi değil. Yani, Ahmet'in Ali'yi, Ali'nin Mustafa'yı, Mustafa'nın da Ahmet'i dövdüğü doğru değil. Bu takdirde, dış tutarlılığa sahip olmayan sistemimiz, yine de iç tutarlılığa sahip olacaktır. Bu iki kavramın, -dış tutarlılık ve iç tutarlılık- arasındaki farkı daha iyi gösterebilmek için, şimdi formel sistemimizi farklı bir şekilde yorumlayalım: Öyle ki, X ''döver'' değil, ''büyüktür'' anlamına gelsin. Bu takdirde, daha a, b ve c sabitlerini yorumlamadan, her üç teoremin birden doğru olamayacağını farkederiz. İşte böyle bir yorumun sistemin iç tutarlılığını zedelediği söylenir. Başka bir deyişle, bu yorum altında sistemimizin iç tutarlılığa sahip olması olanaksızdır, çünkü teoremler birbirleriyle uyarlı **(compatible)** değildir.

Görüldüğü gibi herhangi bir formel sistemdeki bütün simgeleri yorumlamadan da, sistemin iç tutarlılığa sahip olmadığını söylemek mümkün olabilir (bizim örneğimizde tek bir simgeyi yorumlamak yetti). İç tutarlılığa sahip olmayan bir sistemde, yetirli sayıda simge yorumlandıktan sonra, bütün teoremlerin birden doğru olamayacağını görmek mümkün olur. Ancak burada asıl sorun, doğruluk-yanlışlık sorunu değil, olanaklılık sorunudur. Bu yorum altında teoremler birbirleriyle uyarlı olmadıklarından, üçünün birden doğru olması olanaklı değildir.

Bu iki tutarlılık türü arasında aslında yakın bir ilişki vardır. Bir kere iç tutarlılık, dış tutarlılığın mümkün olabilmesi için gerekli ama yeterli olmayan bir koşuldur.

Ama bizi şu anda ilgilendiren, başka bir ilişki. İç tutarlılığın doğrulukla, ama gerçek dünyada doğru olmaktan çok, tasarlanabilen bir dünyayla dış tutarlılığa sahip olması ile ilgili. Bir dizi önermenin birbirleriyle uyarlı olup olmadığını keşfetmek için yapılması gereken şey, içinde her birinin birden aynı zamanda doğru olacağı bir dünya tasarlamaktır. Dolayısıyla aslında iç tutarlılığın da dış dünyayla ilgisi vardır -ancak bu durumda, dış dünya artık içinde yaşamakta olduğumuz dünya olarak değil, tasarlanabilen herhangi bir dünya olarak yorumlanır.

''Tasarlanabilen bir dünya'' **(hypothetriaal world)** ile ne kasdediyoruz? Bir an için şu sorular üzerinde düşünün:

> a'nın b'den, b'nin c'den, c'ninse a'dan büyük olduğu bir dünya tasarlamak mümkün mü?
>
> İçinde kare şeklinde dairelerin yeraldığı bir dünya tasarlamak mümkün mü?
>
> Bir insanın aynı zamanda hem erkek hem de kadın olduğu bir dünya tasarlamak mümkün mü?
>
> Yerçekimi ve görelilik yasalarının geçerli olmadığı bir dünya tasarlamak mümkün mü?
>
> Halit Ziya Uşaklıgil'in Türkiye Cumhuriyeti'nin ilk cumhurbaşkanı olduğu bir dünya tasarlamak mümkün mü?

İlk bakışta bu dünyalardan bir kısmı, diğerlerine kıyasla daha tasarlanabilir gözüküyor. Bu soruları, sorunun tasarlanmasını istediği dünyanın, gerçek dünyaya ait hangi bilgi alanına ait bir bilgi ile çelişmekte olduğuna göre sınıflandırabiliriz. Buna bağlı olarak da, farklı türden tutarlılıklardan sözedebiliriz. Örneğin, bir sistemin yorumu sonucunda, sistemin teoremlerin ikisi birbirini çelmiyorsa, sistemin *mantıkî* olarak tutarlı olduğunu, yorumlanmış teoremler matematiksel kuralları ihlal etmediği sürece *matematiksel* olarak, fizik yasalarını ihlal etmediği sürece *fiziksel* olarak tutarlı vs. olduğunu söyleyebiliriz. Bu ilkeye göre, örneğin biyolojik tutarlılığa sahip olmasını istediğimiz bir sistem, ''Halit Ziya Türkiye Cumhuriyeti'nin ilk cumhurbaşkanıdır'' şeklinde yorumlanan bir teoreme izin verirken,

# TEST 25'in Çözümleri

**1 /** Kuzenlerin kim olduğunu hesaplayacak, dolayısıyla da ''kuzen'' kavramının işlemsel bir tanımını verecek yordam şudur:

```
kuzen(X,Y):—ebeveyn(EX,X),
   ebeveyn(EY,Y),
   kardeş(EX,EY)
```

**2 /** Eğer 35. tümcenin ilk althedefi kaldırılacak olursa, bir insan kendi kendini kardeşi sayılabilirdi. Çünkü ne*(X,Y) 73. fasikülde açıklandığı gibi bir eşitlik koşuludur. Bir althedef olarak kullanılması. 35. tümcede olumsuz olarak yorumlanmasını sağlamaktadır. Böylelikle, daha sonraki althedeflere ancak X ve Y aynı kişi olmadığı sürece ulaşılabilmektedir.

**3 /** Birinin büyükannesinin kim olduğunu belirlemede kullanılabilecek diğer bir yordam şu olabilirdi:

```
büyükanne(X,Y):—anne(X,Z), ebeveyn(Z,Y)
```

**4 /** Büyük-büyükebeveynleri belirlemede kullanılabilecek bir yordam şu olabilir:

```
büyük-büyükebeveyn:—ebeveyn(X,Z), büyükebeveyn(Z,Y)
```

**5 /** Birinin kardeşi olup olmadığını belirleyebilmek için gereken tümceler şunlardır:

```
kardeşivar(X):—kardeş(X,Y)
kardeşiyok(X):—kardeşivar(X),!,fail.
kardeşiyok(X).
```

''Hücresiz hayvanlar vardır'' şeklinde yorumlanan bir teoreme izin vermeyecektir.
Ancak formel sistemler üzerinde yapılan araştırmalar bakımından, bu tutarlılık türleri aynı derecede ilginç değildir. Daha çok üzerinde durulan, bizi özellikle ilgilendiren matematiksel ve mantıki tutarlılıktır. Dolayısıyla, bu bağlamda ''tasarlanabilen dünya''lar üzerine konan kısıtlılıklar, sadece mantık ve matematik kuralları ile çelişmemektir.

## Tamamlanmışlık

İç tutarlılığın dış tutarlılığın da ön koşulu olduğunu ve dış tutarlılığın da şu şekilde tanımlanabileceğini gördük: ''Belli bir formel sistem tarafından türetilen her teorem doğruysa, o formel sistem dış tutarlılığa sahiptir.'' Bu tanımı tam tersine çevirdiğimiz takdirde ise ta-

mamlanmışlığın tanımını elde ederiz. Yani: ''İlgili olduğu alandaki her doğru bilgiyi, yorumlanmış bir formel sistem türetebiliyorsa, o formel sistem tamamlanmıştır.'' O halde tutarlılıkla tamamlanmışlığı kısaca şöyle tanımlayabiliriz:

Tutarlılık: Yorumlanan her teoremin tasarlanabilen bir dünyada doğru çıkması hali.

Tamamlanmışlık: Tasarlanabilen bir dünya hakkındaki bütün doğru önermelerin hepsinin sözkonusu formel sistemin teoremi olması hali.

## Sayı Kuramının Tamamlanmamışlığı

Gödel'in sorunun *Principia Mathematica*'da anlatılan sistemin tamamlanmış olup olmadığını göstermek olduğunu söylemiştik. Bunun için yine büyük ölçüde, bütün matematiksel akılyürütme tarzlarının formelleştirilmesini içerdiği iddia edilen *Principia*'daki yöntemlerin kullanılması gerekecekti.

Bu bağlamda Gödel'in karşılaştığı sorunun bütün üst-metamatikçilerin karşılaştığı sorun olduğunu belirtelim. Üst-matematikçilerin kullanabilecekleri en sağlam ve güvenilir yöntem mantık ve matematiğinkilerdi. Ancak matematiksel önermeler -ve dolayısıyla sayı kuramına ait önermeler de- tamsayıların özellikleri hakkındadır. Tamsayılar ve özellikleri ise önermeler değlilerdir. Burada yine diller ve üstüdiler hakkındaki sorunla karşılaşıyoruz. Sayı turamının tamamlanmışlığı hangi dilde ifade edilecek?

Gödel'in buluşu sayıların önermelere işaret ediyormuşcasına da yorumlanabileceği ile ilgiliydi. Bu amaçla Gödel, o zamandan beri Gödel sayılaması diye bilinen bir kodlama sistemi geliştirdi. Bu kodlamada sayı kuramına ait her önermeye, içerdiği simge dizilerine göre, bir Gödel sayısı verilir. Böylelikle sayı kuramına ait her önermeyi iki ayrı düzeyde birden yorumlamak mümkün olur: Sayı kuramına ait bir önerme olarak ve sayı kuramına ait bir önerme hakkında önermeler içeren bir üst-dilin önermesi olarak.

Gödel sayılamasının en önemli sonucu Giritli Yalancı paradoksunun *Principia Mathematica*'nın sistemi içerisinde ifade edilebilmesiydi. Daha önce Tipler Kuramı'nın *Principia Mathematica*'da Russel paradoksunun ortaya çıkmasını önlediğini fakat Giritli Yalanca paradoksu ile ilgili olarak bir çözüm getirmediğini görmüştük. Dolayısıyla, *Principia Mathematica* sistemi içerisinde Giritli Yalancı paradoksu ile ilgili olarak bir çözüm getirmediğini görmüştük. Dolayısıyla, *Principia Mathematica* sistemi içersinde Giritli Yalancı paradoksunun ifade edilebilmesi, yani *Principia Mathematica*'nın aksiyomlarından hareketle ve *Principia Mathematica*'nın kurallarını kullanarak, ''Bu teorem yanlıştır'' ya da ''Bu teoremin bir kanıtı yoktur'' diye bir teoremin türetilebilmesi, bu teoremin doğruluğuna ya da yanlışlığına aynı sistem içerisinde kalınarak karar verilemeyeceği anlamına gelir.

## GÖDEL TEOREMİNİN BİR YORUMU

Gödel teoremi, verili hiçbir formel sistemin, ilgili olduğu alanda bilinen, (daha doğrusu ifade edilebilecek) her şeyi türetemiyeceğini, daha doğrusu temellendiremiyeceğini gösterdi. Bunu söylemenin başka bir yolu, insan zekası ile ilgili hiçbir alanın tamamen yapısallaştırılamayacağı ya da programlanamayacağını... Hatta giderek, Gödel teoreminden hareketle, insan zekasının matematik, mantık, bilim gibi yüksek düzeyli ürünlerinin hiçbirinin, matematiğin ve mantığın ölçütlerine uyan tam bir bilimsel betimlemesi yapılamayacağı da iddia edilebilir.

Ancak kuşkusuz, Gödel teoremi araştırmacı ve bilim adamlarının insan zekası ve faaliyetleri hakkında bilimsel betimleme ve açıklamalara ulaşma çabasını engellemedi. Mekanik yapay zeka çalışmalarının kendisi Gödel sonrası bir fenomendir.

Aşağıda, görüşleri daha sonra tanıtacığımız Logo dilinin gelişmesinde etkili olmuş, ünlü psikolog ve filozof Jean Piaget'nin, Gödel teoremi hakkındaki bir yorumunu okuyacaksınız. Piaget'ye göre Gödel'in gösterdiği zekanın ulaşmış olduğu hiçbir aşamanın tam olarak yapısallaştırılamayacağı, yani formel bir sistem içersinde ifade edilemiyeceği. Ancak bu, zekanın **gelişmesinin** formel olarak tarif edilebilir bir seyir izlemediği anlamına gelmiyor.

**Gödel'den beri matematiğin temellerini araştıran düşünürler, daha ''güçlü'' ve daha ''zayıf'' yapılar arasında bir ayrım yapmaktadırlar. Bu ayrıma göre, daha güçlü yapıların işlenmesi ancak daha zayıf olanların kurulmuş olmasına bağlıdır. Ancak daha zayıf olan yapıların kuruluşu ise kendi başlarına olanaklı değildir. Daha güçlü yapıların varlığını gerektirir. Böylelik-**

Böyle bir teoremin türetilmesi, sayı kuramına ait önermeler, aynı zamanda bir üst-dile ait önermelermişcesine yorumlanamadıkça, olanaksızdı, Çünkü ancak böyle bir çifte yorum olanağı ''bu teorem'' gibi kendi kendine gönderme yapan bir terimin kullanılmasına olanak tanır.

Gödel teoremi ne anlama geliyor? Bunu teknik olarak zaten açıkladık: *Principia Mathematica* ayarındaki her sistem tamamlanmamıştır; yani doğru ya da yanlış olduğuna karar verilmeyecek önermeler içerir. Ancak buradan hareketle daha geniş bir yoruma varabiliriz. *Principia Mathematica* ayarındaki herhangi bir formel sistemi kullanarak öyle önermeler türetebiliriz ki, bunu formel sistemin kendisi yapamaz -yani formel sistem sınırları içerisinde kamaz -yani formel sistem sınırları içerisinde kalarak yapılan (önermeler) temellendirilemez.

**le soyut yapılardan oluşan bir formel sistem tasarısı hiçbir zaman tamamlanamayacak. bir ''bütün''ün oluşturulma tasarısına dönüşür. Tamamlanamamanın gerekçesi formellikle ilgili [Gödel'in gösterdiği] kaçınılmaz sınırlılıklardır. Ya da şöyle de söylenebilir: [belli bir aşamadaki formelleştiriemeyen] bir içerik bir alt aşamadaki içeriğin formudur, [formelleştirilmesidir] ve her form da daha üst düzeydeki bir form için bir içerik oluşturur.(..) Zeka psikolojisi gibi gelişmenin, gündelik gözlemlerin tâ içine sokulduğu bir alanda, yapı ve gelişmenin (genesis) birbirlerine zorunlu bağımlılığının farkında olmamak olanaksızdır. Gelişme dediğimiz şey, bir yapıdan başka bir yapıya geçişten başka bir şey değildir; ancak bu geçiş daima daha ''zayıf'' bir yapıdan daha ''güçlü'' bir yapıya doğrudur; başka bir deyişle ''biçimlendirici'' (formative) bir geçiştir. Yapı ise sadece bir dönüştürmeler (transformations) sistemi olarak tanımlanabilir; ancak bu sistemin kökleri yine formel işlemlerdir. Dolayısıyla [bir d önüştürümler sistemi olarak tanımlanan yapı] daha önceden dönüştürmeyi yapacak aletlerin oluşmuş olmasına bağlıdır -bunlar da dönüştürme kuralları ya da yasalarıdır.**

**Gelişme sorunu yalnızca psikolojiye özgü bir sorun da değildir. Her alanda ana epistemolojik seçenekler ya her şeyin önceden belirlenmiş olduğu yolunda bir inançtır (predestination) ya da kuruluşçuluktur (constructivism). Matematikçi için kuşkusuz, idealara, negatif ve sanal sayıların ezelden beri Tanrı'nın zihninde varolduğuna inanmak çekicidir. (...) Ama ''soyut'' yapılardan ''gerçek'' ya da ''doğal'' yapılara geçtiğimizde, gelişme sorunu büsbütün önem kazanır. Sözünü ettiğimiz epistemolojik seçenekler organik yapılar sözkonusu olduğunda bile geçerlidir. Bu yapıları ya evrilmekte olan bir kuruluş sürecinin ürünü olarak tasarlayacağız ya da ilk DNA molekülleri tarafından tâ baştan beri belirlenmiş olduklarını düşüneceğiz.**

# Gündelik Hayatın Fenomenolojisi: Alternatif Bir Ontolojiye Doğru

Gözden geçirmiş olduğumuz psikolojik, epistemolojik ve ontolojik varsayımların ortak bir yönleri var: İnsanın ayrık ve atomik olgular biçimini alan veriler üzerinde, kurallara göre işlem yapan bir aygıt olduğunu varsayıyorlar. Bu görüş çok güçlü iki akımın buluşma noktasında oluşmuş ve inandırıcılığını büyük ölçüde bu akımların köklülüğünden alan bir araştırma programı doğurdu: Yapay zekâ. Bu akımlardan birincisi kaynağını Platon'da bulan ve bütün akılyürütme biçimlerini açık seçik kurallara, dünyayı da atomik ve ayrık olgulara indirgeyen, düşünceyi yorumlama faaliyetinin risk ve belirsizliklerinden arındırmaya çalışan felsefi gelenektir. İkincisi ise birbirinden mantıkî olarak bağımsız olan ayrık verileri açık seçik, formel kurallara göre işleyen bir genel amaçlı bilgi işlem aygıtı olan elektronik sayısal bilgisayarların icadıyla en yetkin ürününü vermiş olan teknolojik gelenek. Ancak sadece bu felsefi geleneğin varlığı yüzündendir ki, sayısal bilgisayarların zekâyı benzetleyebileceği düşünülmüştür. Böyle bir geleneğe sahip olmayan başka bir kültürde, bilgisayarlar insan zekâsı için hiç de iyi bir model olarak görülmeyebilirdi. Ancak kaynağını Yunan akılcılığında bulan bizim geleneğimizde, sayısal bilgisayarların, insan zekâsına ulaşabilmek için sadece gerekli programı beklediklerini düşünmek için çok sayıda hazırlayıcı etken vardı.

Bu yüzden, ikibin yıllık felsefi gelenekle en yetkin teknoloji ürününün buluşmasından doğan yapay zekâ araştırma programının, kendisine yöneltilen çeşitli eleştirileri başlangıçta fazla dikkate almadan sürmesi, kaçınılmazdı. Ancak bir lokantada yemek ısmarlama, sohbet etme gibi en sıradan insan davranışlarının benzetlenmesinin yüksek düzeyli davranışların benzetlenmesinden çok daha farklı türden sorunlar yarattığı görülmeye başlandıkça, bu alanlarda çalışanlar, farklı bir geleneğe kulak verme gereğini duydular.

Bu gelenek, bugün yapay zekâ alanında yaygın olarak kullanılmakta olan yüklemler mantığı, kiplikler **(modal)** mantığı vb. gibi notasyon ve sistemlerin geliştirildiği yıllarda, kısmen bu gelişmelere ve bu gelişmelerin berisinde yatan

Batı düşünce geleneğine tepki olarak doğmuş olan fenomenolojiydi.

Fenomenoloji, ''İnsan, zekâ gerektiren davranışları nasıl üretmektedir?'' sorusuna cevap ararken, önceden cevabın açık seçik kurallar biçimini alması gerektiğini varsaymamaya çalışma çabasından doğmuştur, denebilir. Hatta fenomenolojinin ''insan, zekâ gerektiren davranışları nasıl üretmektedir?'' sorusundaki, ''üretmek'' terimine dahi itiraz ettiği söylenebilir. Çünkü ''üretmek'' terimi zekâ gerektiren davranışları ister istemez bir ürün olarak düşündürüyor. Ürünlerinse, formel olarak betimlenebilir bir üretilme tarzı olması gerekir.

İşte bu yüzden fenomenolojinin kalkış noktası, açıklanması istenen davranış hakkında herhangi bir soru sormadan önce, davranışın mümkün olduğunca ayrıntılı bir betimidir. Bu aşamada, bir ağacı görmek, bir şey istemek gibi yaşantıları, o yaşantı kılan belli bir genel özelliğin olup olmadığı, varsa bunun ne olduğu sorulur. Eğer buradan hareketle, genel olarak algılamanın, dil kullanmanın, problem çözümünün genel özelliklerinin ne olduğuna dair bir cevaba varılabilirse, bir açıklamaya varılmış olunur.

Bu tür bir yaklaşımın kaynağı bir anlamda Aristoteles'in yöntemine götürülebilir; Aristoteles'in yöntemine, ancak onun betimleme ve açıklamalarına değil. Platon, zekâ gerektiren davranışlar için kuralları andırır ölçütler ararken, Aristoteles, algı va yargı gibi insan faaliyetlerinin yapısını araştırmayı tercih ediyordu. Ancak eylemlerin berisinde, ''pratik bir kıyas''ın **(practical syllogism)** yattığını iddia ettiği gözönünde bulundurulduğunda onun da hiç değilse kısmen, zekâyı insanın biyolojik yapısından, gövdesinden ayıran ve kendi içinde zekâyı formelleştirmeye çalışan geleneğin içerisinde yeraldığına karar vermek zorundayız.

Ancak 20. yüzyılda, kısmen de bilgisayarlar sayesinde insanın bir nesne ya da aygıt olduğu iddiasının tam olarak neyi içerdiği anlaşıldığında, iddia felsefi olarak sınanabilir ve aşılabilir bir hale geldi. Bu sınama ve aşma çabasının öncüleri, 20. yüzyılın başında Matin Heidegger ve Ludwig Wittgenstein gibi düşünürlerdi. Daha sonra Maurice Merleau-Ponty ve Michael Polanyi gibi düşünürler bu öncülerin görüşlerini geliştirirken, çeşitli bilim alanları da bu görüşlerden etkilenmeye başladı. Son olarak da Hubert Dreyfus bu görüşlerin ışığında yapay zekâ çalışmalarını eleştirirken, daha önce klasik yaklaşım içersinde önemli çalışmalar yapmış olan Terry Winograd gibi bir araştırmacı, son yıllardaki çalışmalarını daha fenomenolojik bir çerçeveye yerleştirmeye çalıştı.

Şimdi bu çeşitli düşünürlerin katkılarıyla geliştirilmekte olan bu görüşler ailesinin anahatlarını vermeye çalışalım.

## Zekâ Gerektiren Davranışlarda Gövdenin Rolü

İnsan davranışlarının açık seçik kurallarla formelleştirilebileceğini ve bilgisayarda benzetlenebileceğini düşünenler, insanın bir gövdeye sahip olduğunu hesaba katmamak durumundalar, çünkü hiç değilse yapay zekâ çalışmalarının bu aşamasında, insanın bir gövdeye sahip olduğu anlamda bilgisayarlar bir gövdeye sahip değildirler. Gövdeyi hesaba katmama tercihinin berisinde de uzun bir felsefi gelenek yatmaktadır. Platon'dan Descartes'a kadar filozoflar gövdeyi zekâ için vazgeçilmez bir unsur olarak değil, onu sınırlandıran bir unsur olarak gördüler.

Aşağıda Descartes'dan vereceğimiz örnek özellikle robot tasarılarından birini içermesi bakımından ilginç. Ama bu örneğin bir ilginçliği daha var. Hernekadar, Decscartes bir robotun bir insanın zeki olduğu gibi zeki olamayacağını iddia etmişse de, aradaki farkı robotun bir gövdeye hapsolmuş olmasına, insanın ise gövdesine rağmen maddi olmayan bir ruhu olmasına bağlamıştır.

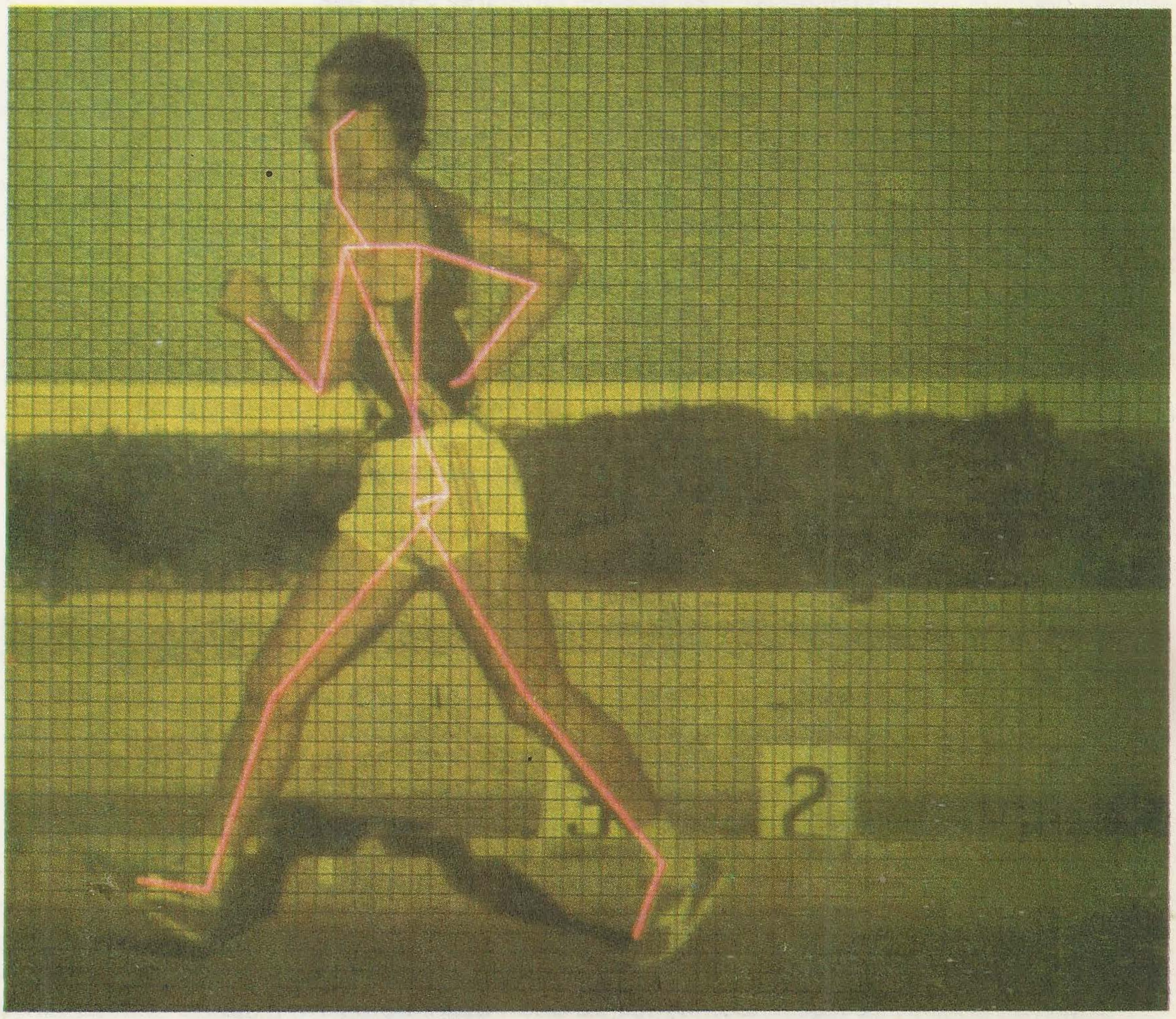
''(...) Hernekadar, bu tür makineler bazı işleri insanlar kadar, hatta onlardan daha bile iyi yapabilirlerse de, başka bazı işlerde kaçınılmaz olarak başarısızlığa uğrayacaklardır(...) Çünkü akıl her tür durumda kullanılabilecek evrensel bir alettir oysa bir makinenin organlarının her yeni durum için yeniden düzenlenmesi gerekmektedir. Bir makinenin içinde, hayatın bütün farklı durumlarında, bizim aklımız sayesinde davranabildiğimiz kadar çeşitli biçimlerde davranmasını mümkün kılacak kadar çok sayıda farklı aygıt bulunması pratik olarak olanaksızdır.''

Bu ilginç bir argüman olmakla birlikte, sayısal bilgisayarların bugün ulaşmış oldukları düzey gözönünde bulundurulduğunda, kanıtlamak istediği şeyi kanıtlayabilmekten uzaktır. Argüman, bir makinenin sınırlı sayıda durumda bulunabileceği varsayımına dayanmaktadır. Oysa çağdaş bir sayısal bilgisayarın $10^{10^{10}}$ durum alabileceği düşünülürse, farklı tepkiler gösterebileceği durumların neredeyse sınırsız olduğu görülür. Bu bakımdan, Descartes'ın argümanına göre sayısal biygisayarın davra-

nışlarının zekice davranmakta olan bir insanın davranışlarından ayırt edilmez olması gerekiyor.

Ancak modern fenomenoloji, zekayı ancak maddi olmayan bir öznenin sahip olabileceği bir yetenek olarak değil, tam tersine ancak dünyayla nedensel ilişkiler kurabileceği pratik faaliyetlerle uğraşabilecek maddi bir gövdenin sahip olabileceği bir yetenek olarak tanımlama eğilimindedir. Başka bir deyişle, bu görüşe göre zeka gayrı-maddi bir ruh ya da donanımdan bağımsız olarak tanımlanan bir program değil, insanın uğraşabildiği maddi faaliyetlerin toplamının bir türevidir.

Sorun merkezi sinir sisteminin çalışmasının benzetlenebilmesinden de bağımsızdır ve yapay zeka alanında çalışanlar da bunun farkındadırlar. Örneğin Herb Simon ''Esnek bir merkezi sinir sisteminin otomasyonu, aynı esneklikte ve duyumsama, alet kullanabilme ve ha-



reket edebilme yeteneklerine sahip bir sistemin otomasyonunun gerçekleşmesinden çok daha kolaydır'' demektir. Ama bu noktada fenemonologlar şöyle bir soru sormaktadır: ''Ya merkezi sinir sisteminin esnekliği ve çalışması harekî sistemin kendisine bağlıysa?'' Ya da daha da fenomenolojik olarak şöyle sorabiliriz: ''Ya daha yüksek düzeyli, daha belirgin ve kurallı gözüken zeki davranışlar, daha bütüncül (global) ''düşük düzeyli'' davranışların bir türevi ise ve onlar tarafından yönlendiriliyorlarsa?''

Yapay zeka çalışmalarında, ''düşük düzeyli'' diye adlandırılan işlevlerin benzetlenmesinde çekilen güçlükler, bu soruların cevaplarının olumlu yönde olması gerektiğini düşündürtüyorlar. Yüksek düzeyli davranışlar değil, örüntüleri tanıyabilme **(pattern recognition)** gibi hayvanlarla paylaştığımız işlevler bilgisayarlar tarafından benzetlenememektedir.

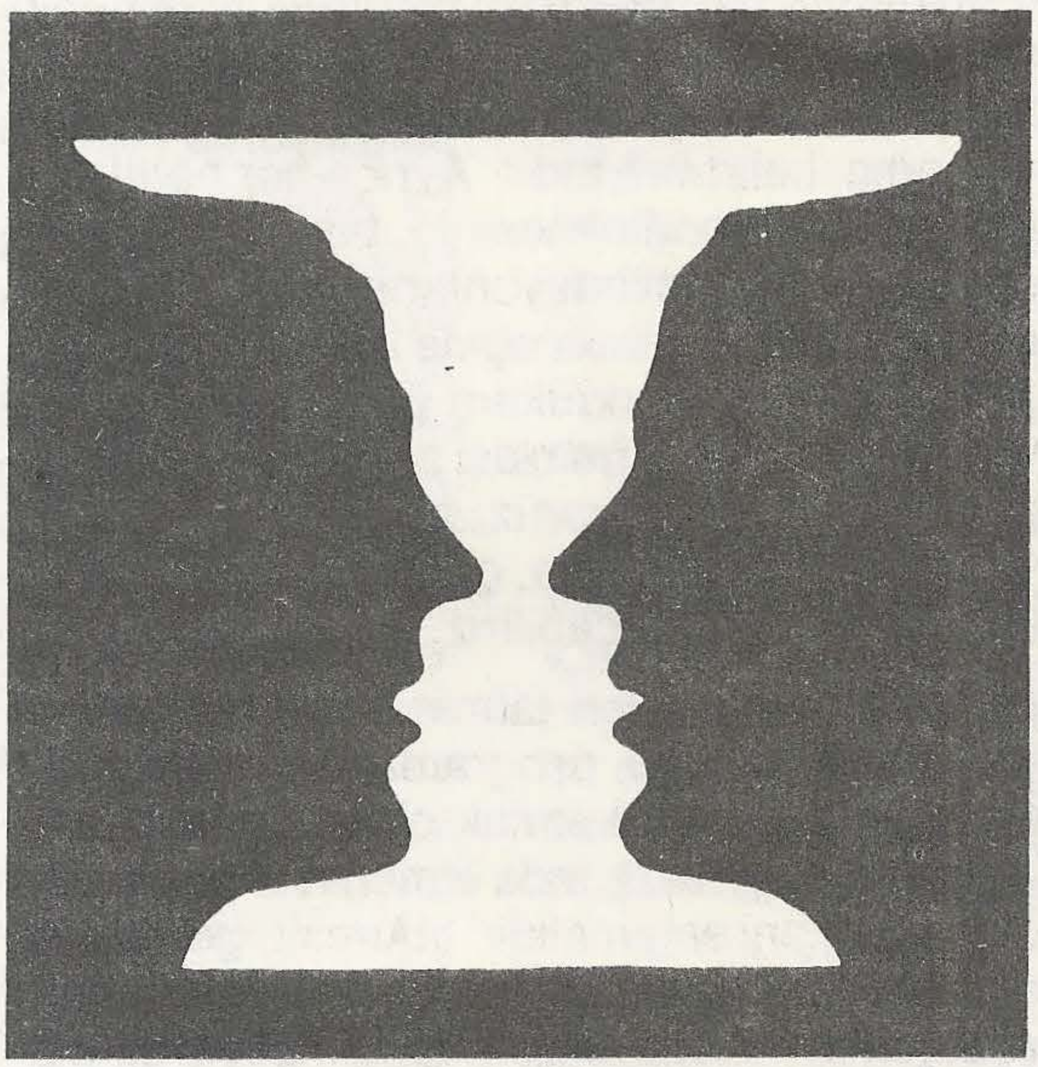
Çünkü örüntü tanıma hakkında yapılan deneyler, tanıdık biçim ve örüntülerin ne olduğuna, belli bir ölçütler listesini tarayarak karar vermediğimizi gösteriyor. Özellikle Gestalt psikologların çalışmaları algalama sürecini yönlendiren ve bu sürecin işlediği verilerin birbirinden bağımsız olarak tanımlanabilecek özgül bir ölçütler listesi olduğu yolundaki tezleri çürütmede etkili olmuştur. Örneğin bir melodiyi, bağımsız olarak verilmiş notaları ''toplayarak'' algılamayız; aksine notalar, nota olarak değerlerini melodi içerisindeki konumlarından kazanırlar. Aynı şekilde nesnelerin görsel algısında da, nesnenin ne olduğundan bağımsız, nötr ölçütler yoktur. Örneğin aynı puslu tabakayı, algılamakta olduğum nesnenin balmumundan yapılmış bir elma olduğunu düşünüyorsam bir toz tabakası, yok gerçek bir elma olduğunu düşünüyorsam, bir nem tabakası olarak algılarım. Bütünün algısı bu tür ölçüt ve özelliklerin toplanmasıyla elde edilmez; aksine bu tür ayrıntıların ne olduğu ve bütünün algısı için ne denli önemli olduklarına, ancak bütün algılandıktan sonra karar verilebilir.

Fenomenologlar algalamanın bu bütüncül niteliğinin berisindeki iki tür belirginleştirilemeyen farkındalığın yattığını düşünüyorlar. Bu farkındalıklardan birincisi ve psikologlar tarafından daha çok incelenmiş olanı, biçim-zemin **(figure-ground)** fenomeni diye bilinen fenomen. Herhangi bir algının mümkün olabilmesi için, biçim ile zeminin birbirinden ayrışabilmesi gerekir. Herhangi bir yaşantı içersinde karşılaştığımız ve dikkatimizi çeken her şey, bir zeminden ayrışır ve onun üzerinde belir-

ginleşir. Zeminin kendisi az ya da çok belirsiz kalarak biçimin belirgin, birleşik ve sınırlandırılmış olarak algılanmasına katkıda bulunur.

Bu fenomeni daha bariz kılan sayısız örnekten biri bu sayfada görmekte olduğunuz çizimdir. Bu çizimde eğer dikkat birbirine bakmakta olan iki yüzün üzerine yoğunlaşırsa, beyaz alan belirsizleşir. Yok eğer vazo biçimine yoğunlaşırsa, o zaman siyah alan belirsizleşir. bir fon, bir zemin olarak algalanmaya başlar.

Zeminin bu belirsizliğinin insanın algılama süreçlerinde merkezi bir rol oynamaktadır.



Merleau-Ponty'ye göre yaşantımız içerisinde olan şeylerin çoğu böyle belirsiz bir şekilde arka planda, zeminde kalmak zorundadır ki, bazı şeyler ön planda, belirgin biçimler olarak algılanabilsin:

''En basit ''duyumsal veri''lerin bir zeminin üzerindeki bir biçim olması (...) insan algasını tanımlayan bir olgudur (...) Algıladığımız bir şey, daima başka bir sürü şeyin ortasındadır; algılanan nesne daima bir 'alan'ın parçasıdır''.

Ama biçim-zemin ayrımı yalnızca algılamanın değil, birçok düşünce sürecinin de ayrılmaz bir özelliğidir. Bir satranç programı yazma amacıyla usta satranç oyuncularından, yüksek sesle düşünmelerini, düşünce süreçlerini yansıtmalarını isteyen araştırmacılar, bu bölümde anlattığımız biçim-zemin olgusuna çok benzer bir olgunun bu alanda da geçerli olduğunu görmüşlerdir.

## Satranç ve Biçim-Zemin Olgusu

Satranç, ilkece, mümkün her hamlenin sonuçlarının hesaplanarak -belli bir kurallar sistemi içerisinde değerlendirilerek- oynanabilecek bir oyun. Ancak bilindiği gibi, burada gözden geçirilmesi gereken seçenekler her hamleyle üstel **(exponential)** bir biçimde artmaktadır. Bu yüzden alternatifler o kadar hızla artmaktadır ki, çok kısa bir zamanda normal bir oyuncu için çeşitli stratejiler birbirleriyle karşılaştırılmayacak bir hale gelmektedir. Bilgisayarların satranç oynamasını mümkün kılacak yazılımlar üzerine çalışan Newell, bu yüzden her bir hamleyi gözden geçirdiği takdirde, makinenin belli bir hamle üzerinde karar vermesinin tahammül edilmeyecek kadar uzun bir zaman alacağını belirtmektedir. Ayrıca her hamlenin tek tek değerlendirilmesinin, bazen önemli ve özgün bazı kombinasyonların da gözden kaçırılmasına yolaçabileceğine işaret etmektedir: ''Makinenin, bulundukları yerde belli bir işlevi olan bütün taşların olası hamlelerini gözden geçirmesini istemiyoruz; ancak bunu da hiç yapmadığımız takdirde, gerçek bazı fırsatların kaçırılmasına yolaçabiliriz.''

Bu çifte gereklilikleri tatmin etmek üzere Newell'ın ilk buluşu programa bir ''tesadüf'' **(random)** unsuru katmak oldu: ''Arada sırada makine, bir vezir feda etmenin elverişli olup olmayacağını sınamalıdır''. Ancak, gerçek bir insanın satranç oynamasını benzetleyecekse, makinenin ''arada sırada'' değil, böyle bir olasılığın satranç tahtası üzerindeki pozisyona ilişkin olduğu durumlarda, vezir feda etmenin elverişli olup olmayacağını sınamalıdır. İyi bir keşif taktiğinden **(heuristics)** beklediğimiz de budur: Mevcut seçeneklerin sayısını sınırlandırarak, daha çok ümit vaadeden seçenekler üzerinde daha fazla zaman harcanmasını sağlamak.

Ancak henüz böylesi bir genel keşif taktiği geliştirilemedi. Bugün kullanılmakta olan taktiklerin hepsi ya usta bazı oyuncuların bulacağı bazı hamleleri bulamıyor ya da değerlendirilecek hamle sayısının üstel artışı ile doğru oranlı bir biçimde çok zaman harcıyor.

Bu noktada gerçek bir satranç oyuncusunun kendi düşünce sürecini nasıl betimlediğine bir göz atmakta yarar olabilir:

''Taşlarından birinin, Kale'nin, korunmasız olduğunu farkediyorum; bundan yararlanabilmenin yolları olmalı. Şimdi farzedelim ki şupı-yonu Fil dörde ittim, eğer Fili kaçarsa Vezirle

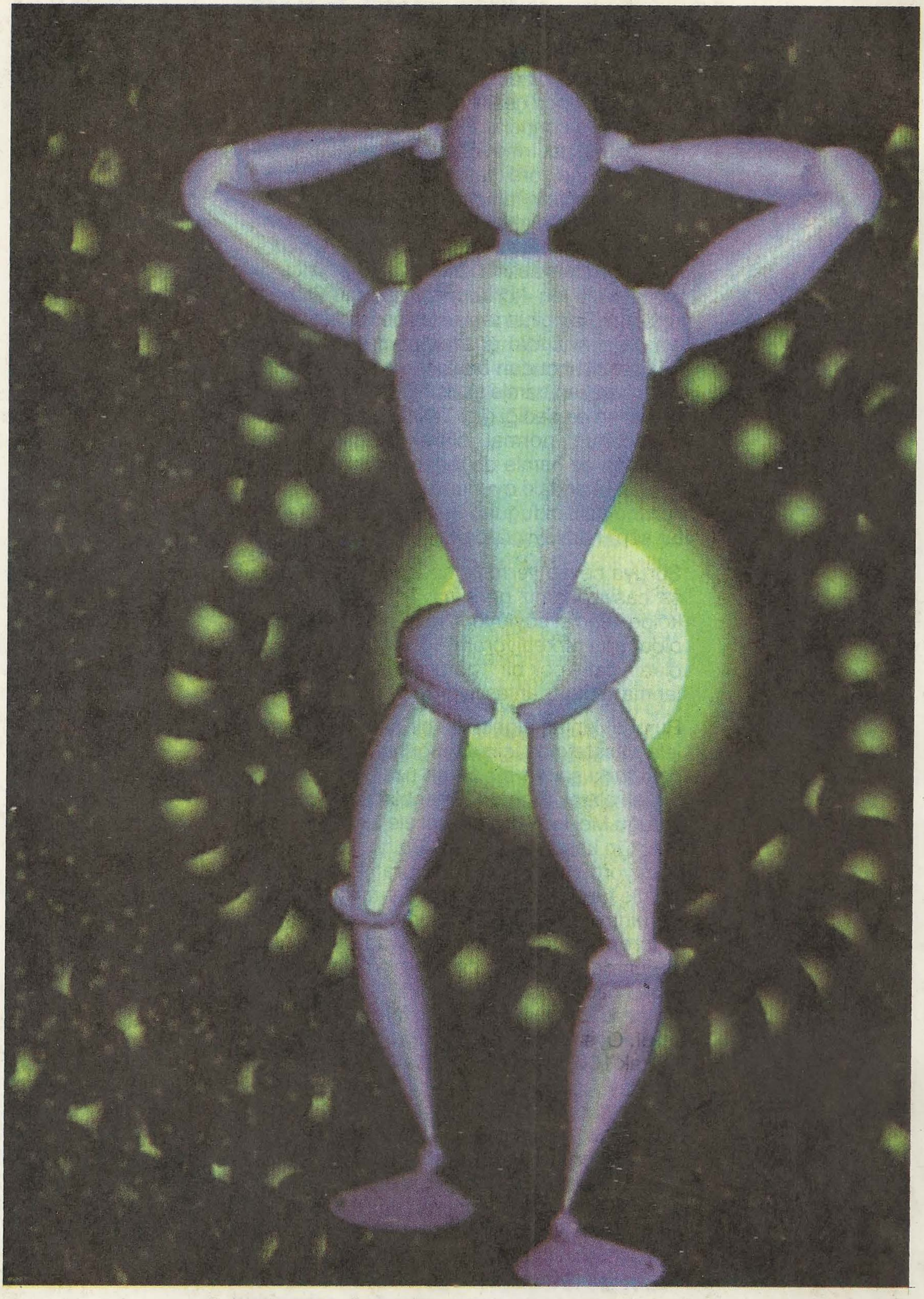
Şah derim sonra da Kale'yi alırım. Eğer... vs.''

Bu protokolün son cümlelerinde, ''sayma'' diyebileceğimiz bir düşünce süreci ile karşı karşıyayız. Oyuncu çeşitli hamleleri, her bir hamlenin bütün sonuçlarını tek tek değerlendirerek gözden geçirmeye başlamıştır. Yapay zeka iddialarına göre, satranç oyuncusunun düşünce sürecinin tamamı, belli birtakım kurallara bağlı keşif taktikleri ile bu tür saymaların toplamından ibaret olmalı. Ama kendi kendimize şunu sorabiliriz: Yukarıdaki oyuncu ilk başta Kale'nin korunmasız olduğunu nasıl keşfetti? Rakibinin taşlarının her birini sırayla (ya da aynı anda) gözden geçirerek mi? Bu belli bir oyuncudan beklenen gözden geçirilmesi gereken hamle sayısını fazlasıyla arttırır. Newell'ın da dediği gibi ''elimizdeki veriler bir oyuncunun normal olarak hamle başına 100'den az hamle değerlendirdiğini'' gösteriyor. Yukarıdaki oyuncunun bir kez Kale'nin korunmasız olduğunu keşfettikten sonra da, bir sürü hamleyi gözden geçirmesi gerekiyor.

Soruya cevap vermek için çok fazla içebakışa başvurmamız gerekmiyor; oyuncu protokolde kendisi söylüyor ''Kale'nin savunmasız olduğunu *farkediyorum*''. Burada sözkonusu olanın daha çok bir ''görüverme'' ya da odaklanma olduğu anlaşılıyor.

Bu odaklanma olgusu ile bir makinenin hamleleri keşfetmek için yaptığı hesaplar arasındaki farkı somut bir örnek üzerindeg örelim. Örnek olarak Richard Greenblatt'in geliştirmiş olduğu McHack programını ele alacağız. MacHack'in kendisi bile mümkün her hamleyi sayamıyordu. Bu yüzden programa, bir ''olası hamle türeticisi'' eklenmişti ve makine ancak bu birimin türettiği hamlelerin sonuçlarını sayıyordu. Ancak belli bir turnuvada sıkıştığı bir durumda, MacHack bir sonraki hamlesini hesaplamadan önce 15 dakika ''düşündü'' ve bu sırada toplam 26.000 hamleyi gözden geçirdi. Oysa usta bir insan oyuncunun normal olarak 100, sıkıştığı durumlarda ise belki 200 hamleyi gözden geçirdiğini biliyoruz. 200 ile 26.000 arasındaki bu fark insanların satranç oynarken, hamleleri saymaktan, her birini tek tek gözden geçirmekten farklı bir iş yaptığını düşündürüyor. Burada ilginç olan soru şu: İnsanlar ne yapıyor olmalılar ki, sadece 200 hamle temelinde, bilgisayarın 26.000 hamleyi gözden geçirdikten sonra bulduğundan daha iyi bir hamle bulabilsinler?

Düşünce sürecini örnek olarak verdiğimiz oyuncu, ''sayma'' sürecine satranç tahtası-



**Gövdenin insan zekâsını belirlemedeki merkezi rolünün anlaşılması, kimi araştırmacıları yapay zekâ çalışmaları ile robot araştırmaları arasında daha yakın bir ilişki hedeflemeye yöneltmiştir.**

1798

nın belli bir bölgesinden başlıyor (''Korunmasız Kale''). Ama bu bölgeye ulaşmak için diğer bazı seçenekleri gözden geçirmiş olduğunu söylemiyor. İşte fenomenologlara göre oyuncunun satranç tahtasının belli bir bölgesi üzerinde bu şekilde odaklanması, algılama sürecinde gördüğümüz biçim-zemin ayrımı ilkesine benzer bir ilke sayesinde gerçekleşmektedir. ''Korunmasız Kale'' birden bire bir figür dikkate değer bir biçim olarak oyuncunun dikkatini çekmiştir; ancak bu bölgenin bu şekilde öne çıkması, kısmen belirsiz kalan zeminin, satranç tahtasının tamamının sayesindedir.

William James bu tür fenomenleri anlamak için ''bilinçliliğin dış kenarları'' **(the fringes of consciousness)** şeklinde bir mecazi ifade kullanmıştı. Bilinçliliğin dış kenarları ile ne kastetmiş olabileceğinin en basit örneğini, ancak durduğu zaman işitmekte olduğumuzu farkettiğimiz bir saatın çalışmasını algılama tarzımızda görebiliriz.

Kendisi de insan zihninin mekanik modelleri üzerinde çalışan bir araştırmacı, bilinçliliğin dış kenarları diye ifade edilmeye çalışılan olgunun yapay zeka çalışmaları için önemini şöyle dile getiriyor:

''Bazı insan işlevleri bazen, açık olarak dikkat edilmeyen ya da üzerinde odaklanmayan bilgiden hareketle yerine getiriliyor. Bu ise bu tür işlevlerle onları benzetleyen otomatik süreçler arasındaki temele bir farka işaret ediyor. Bu farkın nedeni, sayısal bilgisayarların işlemlerinin doğaları gereği ikili mantığa uymak zorunda olmasıdır. Bu işlemler her aşamada ya hep ya hiç durumunda olduklarından, her aşamada ''evet'' mi ''hayır'' mı sorusunu yanıtlayacak kadar özgül ve açık seçik olmak durumundadırlar''.

Bu tür örtük bilgilere en sıradan algılama süreçlerinde bile başvurmaktayız: Tanıdığımız, içinde yaşadığımız bir evin ön yüzü, bize, rasgele bir cepheden daha kalın gözükür; çünkü bilinçliliğimizin dış kenarlarında o evin içi ile ilgili bilgiler bulunmaktadır. Benzer bir biçimde satranç oyununda da, satranç tahtasının her bir yanında aldığımız ipuçları, bilinçliliğimizin dış kenarlarında kalmakla birlikte, dikkatimizi bazı bölgelere yöneltir, o bilgeleri tehlikeli, ümet vaadeden, ya da incelenmesi gereken olarak görmemizi sağlar.

Newell ve Simon'ın da farkında olduğu gibi, satrancın dili, bu düzeydeki olgulara işaret

eden terimlerle doludur.

''İnsanların satranç oynamasına ilişkin kavramların bazıları, bilgisayarlarla ilgili olarak kullandıklarımızdan çok daha bütüncüldür; örneğin ''merkezin kontrolü'', ''kazanmakta olan bir pozisyon'', ''zayıf şah tarafı'', ''tıkanmış bir pozisyon'' vs. (...) Kimi zaman oyuncular ''bu, beyazı kazandıracak bir pozisyondur'' gibi teşhislerde bulunuyorlardı, fakat pozisyonun yapısının hangi yönü ya da özelliğinin bu teşhise yolaçtığını ayırdedebilmek mümkün olmuyordu''.

Öyle görünüyor ki, satranç oyuncuları satranç tahtasının belli bir bölgesine yoğunlaştıktan sonra gerçekten de bir bilgisayarın benzetleyebileceği işlemlerle o bölgeyi çözümlüyorlar.. Ama hangi bölgenin bu şekilde çözümleneceğine karar verilmesi, bu işlemler açıklanabilecek bir olgu değil. Bu bölgenin, satranç tahtasının genel ve belirsiz kalan zemininden ayrışarak üzerinde durulmaya değer bir biçim olarak görülmesi, daha çok Gestalt psikologlarının incelediği türden algılama ilkelerine bağlı olarak gerçekleşen bir olay. Kuşkusuz hangi bölgenin biçim olarak belireceği hem satranç oyuncusunun ustalığına, hem de oynanmakta olan partideki daha önceki hamlelere bağlıdır.

Oysa satranç programları, bir partinin hamlelerinin oluşturduğu ''tarih''i bile izleyemez; her hamleyi bir kitapta bulunmuş bir satranç problemini çözer gibi çözmek durumundadır. Bu kaçınılmazıdır çünkü her taşın geçmiş pozisyonlarını da biriktiren bir programın, biriken veri ağırlığı altında çalışamaz hale gelmesi mukadderdir.

**O halde satranç gibi oyunlardaki problem çözümü ve genel olarak düşünce süreçleriyle ilgili olarak şunu söyleyebiliriz: Burada ihtiyacımız olan şey,hem oyuncunun geçmiş deneyimlerinden hem de oynanmakta olan partinin daha önceki hamlelerinden oluşan zeminin, nasıl olup da satranç tahtasının belli bir bölgesinin oyuncunun dikkatini çeken bir biçim olarak belirmesine yolaçtığının açıklamasıdır.**

## Bütünsellik

Bilinçliliğin dış kenarlarındaki bilginin, satranç oynamak, bir konuşma sırasında sarfedilmiş sözcükleri tanıyabilmek, tanıdık herhangi bir nesneyi algılayabilmek için vazgeçilmez olduğunu gördük. Fenomenolojinin kurucusu olan Edmund Husserl, açıkça hesaba katılmadığı halde yoksayılmayan bu bilgiler bütününü (zemini) ''dış ufuk'' diye adlandırmıştı. Ancak satranç örneğinde olsun, diğer örneklerde olsun, somut bir örnekte bu dış ufuktaki bilgilerin üzerinde yoğunlaştığımız bir problemin çözümüne, bir şeklin tanınmasına nasıl katkıda bulunduğu üzerinde daha durmadık. Bunun için fenomenologların ve Gestalt psikologlarının üzerinde çalıştığı ikinci bir belirsizlik türü üzerinde durmamız gerekiyor: Edmund Husserl'in ''iç ufuk'' diye adlandırmış olduğu ve yine açık seçik dile getirilmemiş bilgi ve tasarımlardan oluşan bir zemin.

''İç ufuk''la kasdedilen şeyin ne olduğunun en kaba örneğini, herhangi bir üç boyutlu nesnenin algılanması ile ilgili olarak görebiliriz.

Herhangi bir nesneyi algıladığımızı zaman, sadece o an gözden geçirmekte olduğumuz yönlerinden daha fazlasının farkında oluruz. Örneğin normal olarak, bir nesnenin yalnızca bir yönünü değil tamamını görmekte olduğumuzu söyleriz. Daha önce de söylediğimiz gibi bir evi, yalnızca önden görmekte olduğumuz zaman bile bir cepheden daha fazla bir şey olarak algılarız.

Husseri'in ''iç ufuk'' terimiyle işaret ettiği, algılama sürecine katkıda bulunan, bilgiler, beklentiler, tasarılar bütününün etkisi, en kolay bir aksamanın varolduğu durumlarda görülebilir. Örneğin bir bardak suya uzandığınız ama bardağın su yerine süt içerdiği bir durumu tasarlamaya çalışın. Bu takdirde, ilk yudumu içtiğinizde göstereceğiniz tepki, olsa olsa ''karmaşa'' ya da ''dağılma'' **(disorientation)** gibi terimlerle adlandırılabilecek bir tepki olacaktır. Ağzınızda hissedeceğiniz tad suyun tadı olmayacaktır ama sütün tadı da olmayacaktır. Herhangi bir ön hazırlık olmaksızın duyumsanan bu tada Husseri, yine mecazi bir ifade kullanarak, ''saf maddenin duyumu'' diyor.

Doğru bütünsel anlamı keşfettiğinizde ilk önce algılayacağınız, ağzınızdakinin ''süt'' olduğu olacaktır. Bunun taze mi bayat mı olduğunu, ekşi, pastörize ya da başka bir tadda olup olmadığını ancak ağzınızdakinin süt olduğunu algıladıktan sonra ayırdedebilirsiniz.

Başka bir deyişle, ''taze'', ''bayat'', ''pastörize'', ''ekşi'' gibi ölçütlere, ancak bu ölçütler için bir zemin işlevi görecek ''süt'' gibi genel bir nitelemeye ulaştıktan sonra ulaşabilirsiniz. Bu tadlar -''taze'', ''bayat'', ''ekşi'' vb.- ancak *sütün* tadları olarak algılanabilirler, dolayısıyla sütü tanımakta kullanılan ayırdedici ölçütler işlevini göremezler.

Bu noktada doğal olarak akla, ''süt'' nitelemesine nasıl ulaşıldığı sorusu geliyor. Örneğin ağızdakinin ''benzin'' değil de ''süt'' olduğu nasıl algılanıyor? İşte Husseri'in iç ufuk terimi ile işaret ettiği, pratik alışkanlıklardan, beklentilerden oluşan ön hazırlık bu noktada devreye giriyor. Yemek yeme faaliyetleri ile ilgili alışkanlıklarımız, gövdesel tanışıklığımız, öğünler hakkında bildiklerimiz, ağzımızdakinin ne olabileceği hakkında seçenekleri, herhangi bir karşılaştırma, sınama ya da arama işlemine gerek kalmadan, önceden sınırlandırır.

Kuşkusuz bu ön-hazırlık şaşmaz ya da yanılmaz değildir; kimi zaman bizi yanlış cevaplar vermeye, yanlış tanılarda bulunmaya yöneltebilir.

Şimdi bir an için bir bilgisayarın böyle bir fenomeni nasıl benzetleyebileceğini düşünelim: Belirgin verileri açıkça tanımlanmış kurallara göre işlemek zorunda olan bilgisayar, böyle bir olayı ancak bir dizi hipotezi sınayarak benzetleyebilir.

Gerçekten de, bütün insan davranışlarını ''çıplak'' ve belirgin veriler üzerinde yapılan kurallı işlemlerden türeten felsefî geleneğin de etkisiyle, yapay zeka iddiasını savunanlar, bu bölümlerde gözden geçirmekte olduğumuz, bütünsellik özelliğiyle yüklü Gestalt fenomenlerini, ''sınama/yanılma'' ya da ''hipotez test etme'' modelleri çerçevesinde kavramsallaştırmaya çalışmışlardır. Bu onlara, sözünü ettiğimiz bütünselliği bir önermeler kümesi ya da hiyerarşik bir yapı şeklinde tasarlamak olanağını vermiştir. Ancak bu şekilde tasarlanan bütünsellik, fenomenologların incelediği ve yapay zeka iddialarını sınırlandırmak için öne sürdükleri bütünsellikle aynı şey değildir.

Genellikle bütünsellik şu şekilde nitelenir: Her bir parçanın yorumunun içinde bulunduğu bütüne bağımlı olduğu yapı ve süreçler, bütünsel yapılardır. Ancak bütünselliğin bu tanımı, hem hiyerarşik olarak tasarlanan hem Gestalt türü yapıları kapsayacak kadar geneldir. Bir örnek verelim:

Geliştirilen anaojik problem çözümü makinelerinden biri, üst üste yerleştirilmiş iki figürü birbirinden ayırdedebilmektedir ve makinenin bunu yaparken, durumun ''bütünsel'' özelliklerinden hareket ettiği iddia edilmektedir. Oysa burada ''bütünsel'' özellikle kasdedilen belli bir hiyerarşik yapı içerisinde daha üst düzeyde yeralan özelliklerdir sadece. Böyle bir yapı içerisinde, her özellik kendinden daha alt düzeydekiler için bir ''bütün'' işlevini görebilir. Makinenin ''algılama'' yoluyla edindiği ve girdi işlevini gören ilk ''ölçüt''' ya da bilgi biti bir arama sürecini başlatır. Bu süreç algılanan ölçütü içeren daha üst düzeydeki bir ölçüt ya da yüklemi buluncaya kadar sürer. Aynı ölçüt birden fazla yüklem ya da ölçüt tarafından içeriliyorsa, arama daha üst düzeylerde de sürer.

Bu şekilde hiyerarşik yapılarla temsil edilen bütünsel süreçlerle fenomenologların ilgilendiği Gestalt fenomenler arasındaki farkı kavrayabilmek için, bütünselliğin daha açık bir tanımını vermek ve gövdeyle ilişkisini göstermek zorundayız. Kendisi zekanın mekanize edilebileceğine inanan bir psikolog olmakla birlikte Neisser, zamansal bir Gestalt fenomenin-

den, müzikteki ritm olgusundan, hareketle böyle bir tanım vermektedir.

''Müzikal ritm dediğimiz olgunun kendisi, hiçbir zaman noktasında mevcut olmadığı halde, parçalarının her biri, yani her tek vuruş, anlamını (göreli konumunu) ritmik bütünsellikten alır. Bu bütünsellik yalnızca dinlemekte olan öznenin zihinsel bir yönelişi, bir kasdı **(intent)** olarak, yani bir Gestalt olarak varolur''.

Bu örneği daha önce gözden geçirmiş olduğumuz hiyerarşik ölçütler örneğinden ayıran en önemli özellik, burada neyin bir parça sayılabileceğinin bile bütünsel yapının kendisi tarafından belirlenmesi. Oysa hiyerarşik bir yapı içerisinde, belli bir durumdaki yorumlanış tarzı daha üst düzeydeki ölçütler tarafından belirleniyor olsa bile, her ölçütün diğerlerinden bağımsız olarak tanımlanması gerekir. Dolayısıyla, fenomenologların bütün dediği şey böyle bir çözümleme içersinde ancak birbirinden bağımsız olarak tanımlanmış ölçütlerin mantıki, kesişimi, kombinasyonu olarak kavranabilir. Oysa Neisser'in ''zihinsel yöneliş'' dediği şey, bizi zeka ile gövde arasındaki ilişkinin düğüm noktasına getiriyor.

Genellikle herhangi bir beceriyi edinme sürecimiz içerisinde -bu beceri ister araba kullanmak olsun, ister dansetmek, ister yabancı bir dilde konuşmak- önce bilinçli olarak birtakım kurallara göre davrandığımız bir aşama vardır. Bu aşamada genellikle yavaş, hantal ve beceriksiz bir biçimde davranırız. Sonra öyle bir nokta gelir ki, bu kurallar unutulur ve sözkonusu davranış kendiliğinden ve otomatik bir biçimde yapılmaya başlanır. Bu noktada artık algı ve hareketlerimizi düzenleyen, kaslar düzeyinde yeni bir Gestalt edinmişizdir ve bu sayede davranışımız yeni bir kıvraklık ve esneklik kazanır.

Genel olarak beceriler için verilen bu çözümleme algılama sürecinin kendisi için de önerilebilir. Merleau-Ponty dokunma duyumu ile ilgili çarpıcı bir örnek vermiştir. İpeğin dokusunu algılamak ve başka duyumlardan ayırdetmek, ancak elin kumaş üzerinde belli bir şekilde, hareket ettirilmesinin öğrenilmesi ve bu hareketlerle, edinilen ve beklenilen duyumların koordine edilebilmesiyle mümkün. Bu gerekli becerileri edinmeden önce hissedeceğimiz sadece karışık birtakım duyumlardır.

Dokunma, işitme ve tadalma gibi duyumlarla ilgili olarak gövdenin rolünü görmek nisbeten kolaydır ama görme de, öğrenilmesi gereken bir beceridir. Odaklama, doğru perspektifi bulabilme, birtakım ayrıntıları seçebilme gibi yeteneklerin hepsi birtakım eylem ve beklentilerin koordine edilmesini gerektirir. Görsel dünyamızın karmaşık duyumlardan değilde sabit nesne ve ufuklardan oluşmasını mümkün kılan algısal sabitler **(perceptual constancies)** Piaget'ye göre ''gerçek eylemlerin ürünüdürler. Bu eylemler bakışın ya da diğer duyum organlarının fiili ya da gücül (potansiyel) hareketleridir''

Algılama sürecine içkin olan bu gövdesel beceriler, nesneleri yalnızca tek bir duyum kanalıyla tanımamızı sağlamazlar. Aynı zamanda çevreyi araştırabilme ve yoklayabilme becerilerimiz, çeşitli duyum kanallarımız arasında, yine his düzeyinde bir eşdeğerliğin kurulmasını sağlar. Bu eşdeğerlik görülen bir nesnenin aynı zamanda dokunulabilen bir nesne olarak algılanmasını mümkün kılar. Bu beceriyi benzetleyebilecek bir bilgisayarınsa, görsel olarak çözümlenen bir nesnenin özelliklerinin bir listesini çıkarması ve bu listeyle, dokunma duyargaları ile çözümlemiş olduğu nesneden edindiği listeyi karşılaştırmak üzere programlanması gerekir. Bu ise her nesne için, farklı her duyum kanalında farklı bir model olacağı ve bir nesnenin ancak bütün bu modellerle paylaştığı ortak özellikleri temelinde tanınabileceği anlamına gelir.

Oysa insan gövdesi böylesi bir formel çözümlemeyi gereksiz kılar. Algılama becerisinin edinilmiş olması demek, tek tek uyarıları karşılaştırma ve birleştirme yeteneğinin edinilmiş olması demek değil; belli tür durumlara belli bir türden davranımla karşılık verebilme yeteneğinin edinilmiş olması demektir. Bu durumlar çok değişken olabilir, ayrıca davranımlar da kimi zaman bir organ kimi zaman başka bir organ tarafından yerine getirilebilir. Ancak bu durumlar ve davranımlar değişkenliği içerisinde ortak olan, özellikleri ya da ayırdedici ölçütleri arasında kısmi bir benzerlik ya da ayniyet değil, ortak bir anlam ve önemdir. İşte bu tür nedenlerden ötürüdür ki, bir yüzeyin sertliğini, dirençliliğini, ellerimle de, ayaklarımla da ve hatta bakışımla da görebilirim. Yine Merleau-Ponty'nin sözleriyle söyleyecek olursak, ''herhangi bir duyumu uyaran bir nesne, aynı zamanda bütün diğer duyumların işlemlerini harekete geçirir.''

Tıpkı makineler gibi, algılamakta olan bir insanın da geri-bildirime **(feedback)** ihtiyacı vardır. Ama burada da sözkonusu geri-bildirimin türleri arasında önemli bir fark vardır. Bilgisayar ancak belli hipotezler geliştirmek üzere

programlanabilir, dolayısıyla onun kabul edebileceği geri-bildirim yanlızca geliştirilmiş olan hipotezin doğrulanıp doğrulanmadığı hakkında bilgi içerebilir. Oysa insanlar için geri-bildirim çok daha esnek bir ölçüt hakkında, duruma uygun davranılıp davranılmadığı ya da duruma hakim olunup olunmadığı hakkında bilgi verir. Bir sonraki bölümde insanlar için ''durum''ların ne anlama geldiği ve neden duruma uygun davranmak, duruma hakim olmak gibi tanımlanan eylemlerin formel bir tanımı verilemeyeceği üzerinde duracağız.

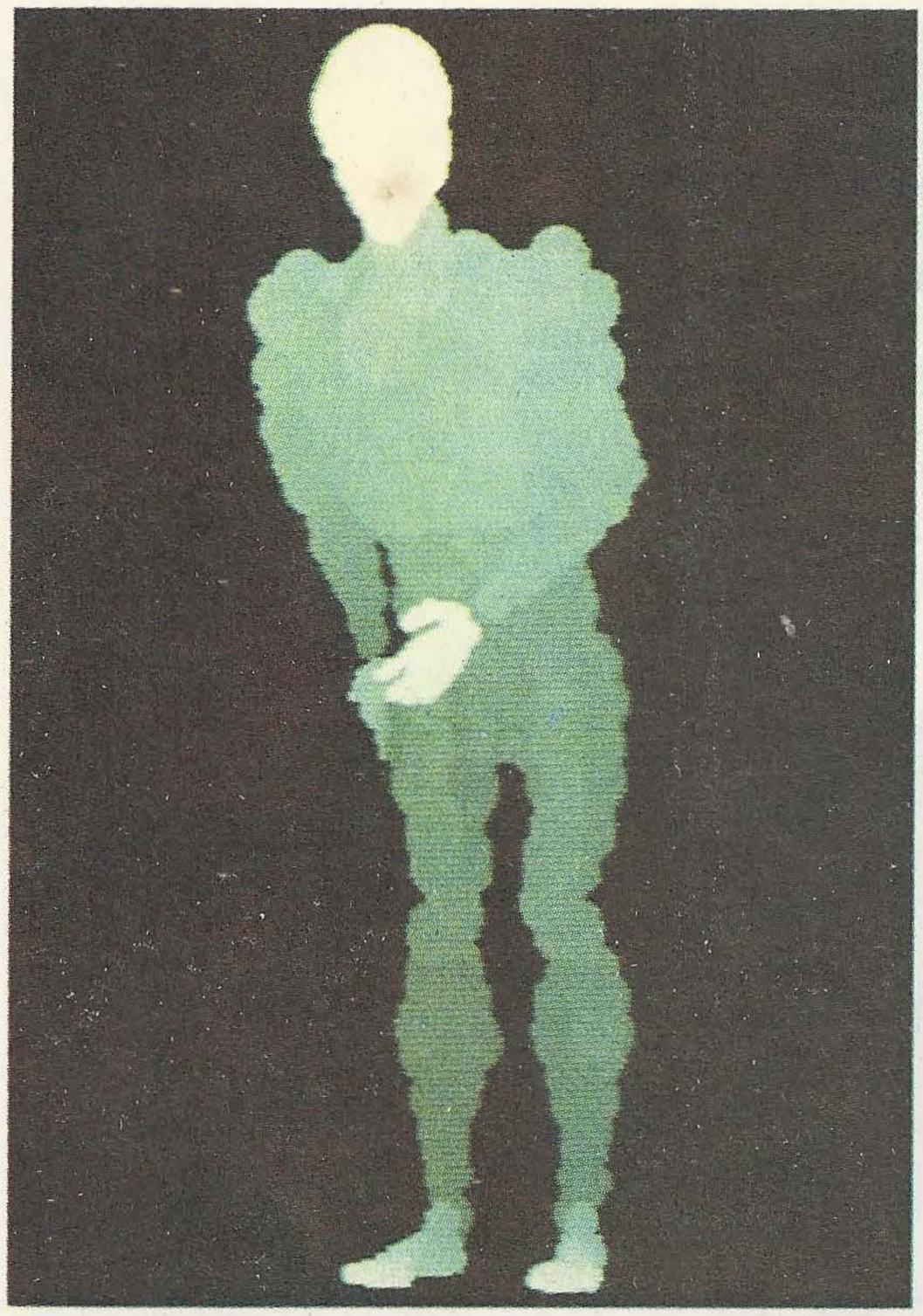
**Sonuç olarak:**
**Bilgisayarların, insanların algısal süreçlerini, algılanacak örüntünün** (pattern) **sınırlı sayıda bir ölçütler kümesi temelinde mümkün diğer nesnelerden ayırdedilebileceği durumlarda benzetleyebileceğini söyleyebiliriz. Ancak bu türden bir örüntünün algılandığı bir algılama deneyimi, yapay bir süreçtir. Gündelik hayatımızda algıladığımız karmaşık örüntülerin algılanması, mekanik olarak benzetlenemeyecek bazı başka ilkelere dayanmaktadır. Bu ilkeler, fenomenologlara göre, kaynağını insanın içinde bulunduğu duruma, kendi kendisini, ihtiyaç, hedef ve tasarılarına göre uyarlayabilen bir gövdesi olmasına bağlıdır.**

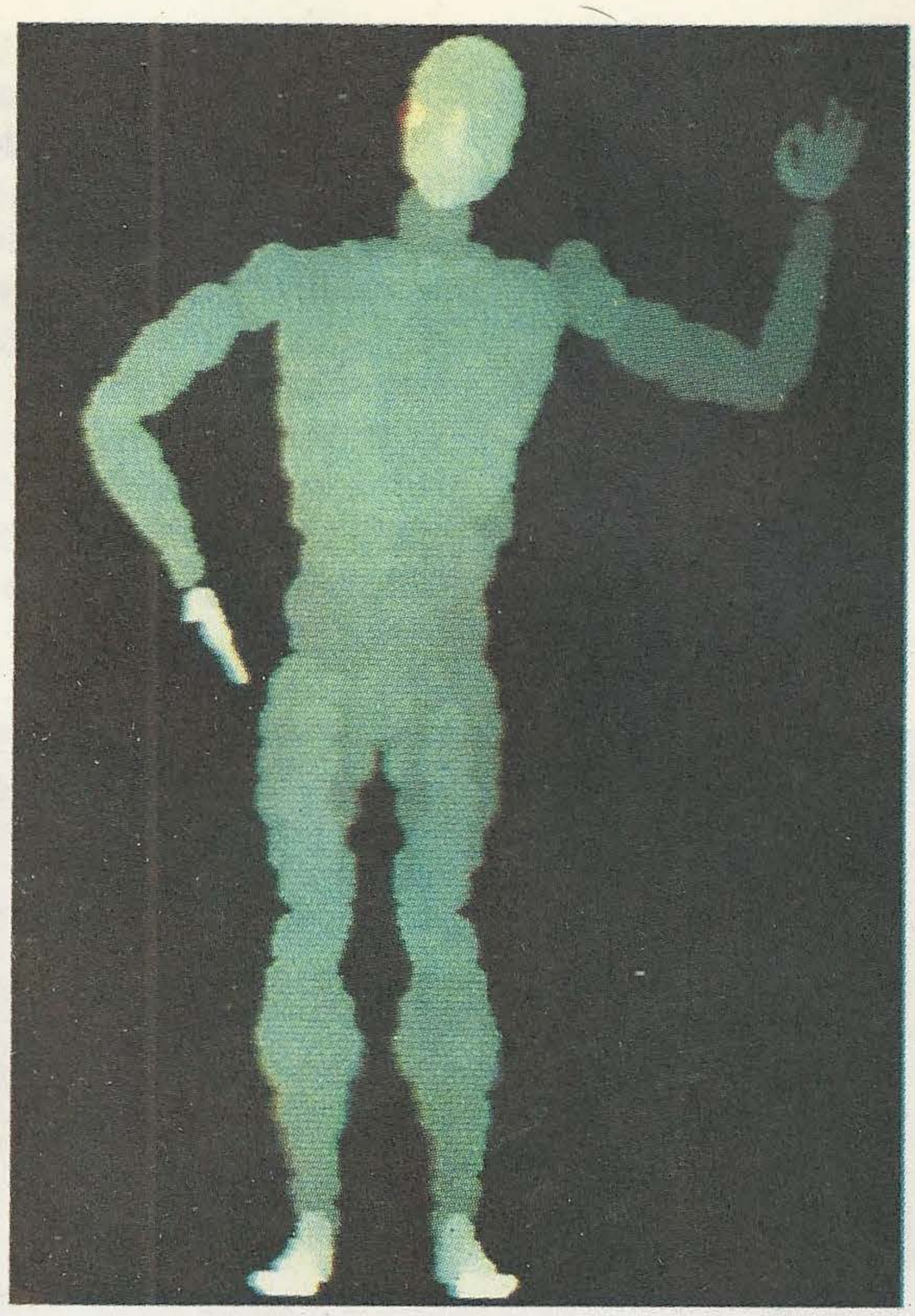
## Fiziksel Durumlar ve İnsanlık Durumları

Bir önceki bölümde insanların benzer davranımlarla karşılık verdiği durumlar arasındaki ortak özelliğin, sınırlı sayıda ve açık seçik dile getirilmiş belirgin ölçütlerle ifade edilemeyeceği, farklı durumların algılanma ve sınıflandırılmasında, bu durumların anlam ve önemlerinin çok daha merkezi bir rol oynayabileceği olasılığı üzerinde durmuştuk. Eğer bu doğru ise, insanların belli bir durumda o durumun, kendi o andaki kayguları bakımından önemli olan özellikleri üzerine yoğunlaşabilme, neyin kendi yerel tasarılarına ilişkin olup olmadığını kestirebilme yetenekleri var, demektir.

Bu yeteneğin bilgisayar tarafından benzetlenebilmesi için, önce mevcut ölçütlerin gözden geçirilmesi ve bir önem hiyerarşisi üzerinde değerlendirilmesi gerekir. Ancak bu yeteneğin bu şekilde çözümlenmesi iki tür bilgi, yani bir şey hakkındaki bilgi **(knowing that...)** ile bir şeyi yapabilmenin bilgisi **(know how...)** arasındaki farkı ihmal ediyor.

İngiliz Filozofu Charles Taylor bu şöyle dile getiriyor:





**Fenomonologların iddialarına göre zekâ, donanımdan bağımsız olarak tasarlanabilecek bir program değil, bütün insan gövdesinin bir işlevidir. Zekâ ile gövde arasındaki bu süreklilik, insan davranışlarının kıvraklığını ve esnekliğini açıklar. Bilgisayarın benzetleyebileceği, ancak acemiyken ve dolayısıyla bilinçli olarak kurallara göre davranırken yaptığımız kesintili ve ayrık davranışlardır.**

''İnsanların durumlar ve bu durumlara içkin olanaklar hakkındaki bilgisinin hatırı sayılır bir bölümü bir şeyi yapabilmeyi bilme **(know how...)** türündendir. Dolayısıyla, bu bilgi olgusal cümlelerle ya da özgül komutlarla tüketici bir biçimde betimlenemez ama yine de belli bir durum hakkında olgusal cümleler ya da özgül komutlar üretebilme yeteneğinin ardında yatan da beceri türünden bu bilgidir. Bu beceri türünden bilginin genellikle sadece birtakım eylemleri yapabilmekle ilgili olduğunu düşünürüz. Ama aslında sadece yapıp ettiklerimizin değil, başımıza gelenlerin ve tutkularımızın berisinde de benzer türden bir becerikllik yatar. Nasıl ellerimi kullanmanın, yürüyüşe çıkmanın, bir araba kullanmanın, ya da (bir yargıç isem eğer) mahkemede bir davayı yürütmenin ne demek olduğu hakkında genel bir kavrayışım varsa, tehdit edilmenin, sevgilim tarafından terkedilmenin, aptal yerine konmanın ne demek olduğu hakkında buna benzer bir genel kavrayışım vardır.

Şimdi somut bir örnek üzerinde duralım: Bir seyisin bazı genel insani eylem ve tutkular

hakkında böyle bir genel kavrayışı vardır. At yarışının tehlikeli bir girişim olduğunu, kazanılması için jokeyin ve atın bütün yetenek ve çabalarını seferber etmeleri gerektiğini bilir. Ama aynı kavrayış bu irade ve çabanın yetmeyeceğini, boşa çıkacağı sayısız durumu tasarlayabilme ve tanıyabilme yeteneğini de içerir. Bu sayısız durum zihninde veya beynindeki, yarış hakkındaki genel kavrayıştan bağımsız olgular halinde depolanmamıştır. Kuşkusuz bu genel kavrayışın kapsam ve kesinliği farklı insanlarda farklı farklı derecelerde olabilir. Örneğin sözkonusu kişinin kendisi daha önce ata binmişse, bu faaliyeti daha yakından ve içerden tanıyacaktır. Ama hep şehirlerde yaşamış biri bile, sırf kazanmav e kaybetmenin, bu uğurda yarışmanın vb. ne demek olduğunu bildiği için, at yarışlarına belli bir ön hazırlık ve aşinalıkla yaklaşacaktır.

Bütün bunlara karşın yapay zekayı savunan biri, yine de söylediklerimizin yalnızca alternatif bir ''depolama'' yöntemi ile ilgili olduğunu iddia edebilir ve bu depolama yöntemin bilgisayarlarda uygulanamayacağını kabul etse bile, insanlarla ilgili olarak erişim probleminin nasıl çözüldüğünü sorabilir? Bizim örneğimizdeki seyis, belli bir durumda depolamış olduğu bütün bilgilerden hangilerinin o duruma ilişkin olduğunu nasıl kestiriyor? Şöyle cevap verilebilir: Eğer dünyayı kavrayışımızın çeşitli yeteneklerimiz sayesinde onunla kurduğumuz ilişkiler ve çeşitli tasalarımızın ondan etkilenmişliğinden kaynaklandığını düşünürsek, o zaman nasıl olup da belli bir tasa ya da hedefin, bir durumun sözkonusu tasa ya da hedefe ilişkin olan özelliklerini seçtiği sorunu ortaya çıkmaz bile. Çünkü belli bir tasası, bir hedefi olmak ya da belli bir durumla belli bir tür ilişki kurmak, o durumun farkında oluşumuza sonradan eklenen, o farkındalıktan ayrı bir şey değildir.''

Yani bu durumun önce farkında olup, sonra tasalarımızla, hedeflerimizle ilgisini kurmayız; her farkında oluş zaten ancak belli bir tasanın, tasarının, hedefin ışığında gerçekleşir.

Başka bir deyişle, insanların dünyası zaten daima hedeflerle, tasalarla yapılandırılmıştır. Öyle ki, sadece neyin önemli, neyin önemsiz sayılabileceği değil, neyin bir nesne olduğu da daima bir insanî tasaya gönderme yapar, onu içerir. Bu ise, sadece evrensel olarak, yani durumdan bağımsız **(context free)** olarak tanımlanmış nesne ve ölçütler hakkında bilgileri işleyebilecek bir bilgisayar tarafından algılanabilip değerlendirilemez.

## Durumlar ve İlgi Alanları

Heidegger insanların dünyasının bu şekilde insani tasa ve hedeflerle yapılandırılmışlığını ve iç-tutarlılığını, bir işlik örneğinden hareketle açıklamaya çalışmıştır. Bir işlikteki aletler yanlızca anlamlarını ve kullanışlıklarını değil, ne olduklarını bile ancak birbirleriyle kurdukları karşılıklı ilişkiler ve nihaî olarak da insan ihtiyaçları ile olan ilgileri temelinde edinirler: Çivilerin, tahtaların vs. tanınmadığı bir işlikte, çekiç diye bir aletten de sözetmek anlamsız olurdu.

Aynı şekilde, bir otomobilin sinyali gibi, tamamen fiziki düzeyde ve ayrık **(discrete)** olarak betimlenebilir görünen bir olgu bile, anlamını ancak bütün bir pratik bağlam ile kurduğu ilişkiden ötürü edinir.

Nesne ve olguların bizim pratik tasa ve tasarılarımızla içiçeliği kuşkusuz yanlız bilgisayar uzmanlarının değil çoğumuzun gözünden kaçar. Çünkü Wittgenstein'ın da dediği gibi ''şeylerin bizim için en önemli yönü, basitliklerinden ve onları çok yakından tanıdığımızdan ötürü, gözümüzden kaçar. (İnsan bir şeyi hep burnunun dibinde durduğu için de farketmeyebilir).''

Bütün bunlardan hareketle, insan zekâsının yöneldiği dünya ve durumların, bir ayrık olgular ve nesneler kümesi olarak değil, içiçe geçmiş ilgi alanları olarak varolduğunu söyleyebiliriz. Bir nesne daima ancak bir ilgi alanının bir parçası olarak varolur. Nesnenin ne olduğunu belirleyen, kendisine içkin olan ve içinde bulunduğu ortamdan bağımsız olarak betimlenebilecek özellikleri değil, içinde bulunduğu bütünsel alandır. Bu alan ise insan ilgi, tasa ve hedefleri ile yapılandırılmıştır.

Herhangi bir durumda (bir satranç oyununda, bir at yarışında vs.) o durumun bizim o anki ilgilerimiz, tasalarımız için önemli, onlara ilişkin yönüne odaklanmamızı mümkün kılan da, durumların bu özelliğidir. Nesneler zaten, baştan beri bizim ilgi ve tasalarımızın türevi olan alanların bir parçası olarak varoldukları için, daima şu ya da bu tasarıya ilişkin olarak algılanırlar.

Kuşkusuz bütün bunlar belli durumların hiçbir zaman, formel kurallar ve olgular şeklinde çözümlenemeyeceği anlamına gelmiyor. Ancak yapay zeka iddialarını savunanlarla fenomenologlar arasındaki tartışma formelliğin yeri konusundaki düğümleniyor. Yapay zeka iddialarını savunanlara göre, formel kurallar ve

her durumdan bağımsız olarak tanımlanmış olgular, mümkün her zeki davranışın zeminini oluşturuyorlar. Oysa fenomenologlara göre formelleştirme çabasının kendisi de, belli bir amaçla ve belli bir bağlamda girişilen ve ancak bu amaç ve bağlam gözönünde bulunduğunda anlaşılacak bir faaliyettir. Formelleştirme girişiminin bağlam ve amaçlarının, bu girişimin nihai ürünü olan formel yapı üzerindeki etkisini, bir örnek üzerinde görelim.

## Programcının Sezgileri

Örnek olarak inceleyeceğimiz sistem Edward Feigenbaum tarafından geliştirilmiş **EPAM** sistemi. Öğrenme yeteneği de olan bu sistemin amacı belli bir ilgi alanındaki nesnelerin bütün özelliklerini, dallanmış bir ağaç yapısı **(tree structure)** şeklinde temsil etmek. Özellikler, herhangi somut bir bağlamdan bağımsız olarak tanımlanmış yüklemler, yapının kendisiyle hiyerarşik bir yapı... Dolayısıyla en üst düzeydeki bir yüklemden başlanıp belli bir dallanmalar sırası izlenerek en uçtaki düğümlerden **(nodes)** birine varıldığında, belli bir nesnenin tüketici bir betiminin elde edilebilmesi gerekiyor.

Ancak EPAM sisteminde gözden kaçırılan çok önemli bir nokta var. Bu hiyerarşik dallanmaların yapısının (düğümlerin birbirlerine göre sırasının), bilginin düzeni üzerinde belirleyici bir etkisi olduğu gözden kaçırılmış oluyor. EMAM'da bilgisayarın nesneleri ayırdetme ve öğrenme sırasına göre, nesnenin özelliklerini temsil eden yüklemler de hiyerarşide farklı bir düzeye yerleştiriliyor. Bunun yarattığı sorunları ise, kendisi de bir bilgi işlemci olan William Wynn şöyle dile getiriyor:

''EPAM'ın sınıflandırma sürecinin oluşturduğu yapı, sınıflandırma sürecinin kendi tarihi tarafından fazlasıyla belirleniyor. Bu ise yapının uyarlanamaz olması sonucunu doğuruyor. Çünkü bu sistem ancak aşağıdan yukarı oluşturulabiliyor ve bir kez oluşturulduktan sonra yukardan aşağı yeniden düzenlenme olanakları içermiyor. Dolayısıyla ayırdetme süreci içersinde kullanılan, fakat daha sonra belli bir uyarıcı kümesi için fazla ayırdedici olamadığı keşfedilen testlerden kurtulunamıyor; ne de hiyerarşinin daha üst düzeylerine yeni genel amaçlı ayırdetme testleri eklenebiliyor. Yani bir kez oluşturulduktan sonra, EPAM'ın nesneleri ayırdetmeye yönelik hiyerarşik yapısı, daha hızlı erişim sağlamak uğruna yeniden düzenlenemiyor. Erişimi hızlandırmak için testlerin herhangi bir şekilde yeniden düzenlenmesi, bellekte depolanmış bazı maddelere erişimi olanaksızlaştırıyor.''

Görüldüğü gibi, ayırdetmelerin sırası ve dolayısıyla yüklemlerin hiyerarşideki düzeyi belirleyici bir önem taşıyor. Oysa fiziki dünyada bu yüklemlere tekabül eden özellikler arasında böyle bir düzey farkı, bir öncelik sonralık sırası yoktur. Bu düzey farkını yaratan, ayırdetmelerin sırasını belirleyen, programcının kendisidir. Bu sunuş sırası sayesinde aynı ilgi alanına ait nesnelerin ayırdedilme sırası belirlenerek, bu ilgi alanını temsil eden bir yüklemin hiyerarşinin üst düzeyinde yeralması ve hatta düğümlerin birbirlerine göre sırasının içiçe geçmiş ilgi alanlarını yansıtması sağlanabilir. Bu takdirde, programcının bilgisayarın öğrendiği nesnelerin içinde yeraldıkları somut bağlamlar hakkındaki pratik sezgisi, gerçekten de program tarafından içerilmiş olacaktır. Ancak bu ''bilgi''yi hiyerarşik yapının hiçbir düğümü yansıtmayacak, aksine bu bilgi ayırdetmelerin sırası tarafından belirlenmiş hiyerarşik yapının tamamı tarafından içerilecektir. Ancak bu hiyerarşik yapı da, belli ve tek bir amaç gözününde bulundurularak oluşturulmuş olduğundan, programcının pratik bilgisinin aksine, Wynn'in de işaret ettiği gibi, katı ve uyarlanamaz olacaktır.

Diğer yandan esnekliği sağlama uğruna, her türlü sıralamadan ve dolayısıyla hiyerarşik bir yapıdan vazgeçildiği takdirde, nesneler hakkındaki bütün bilgilerin yapılandırılmamış bir yüklemler listesi halinde saklanması, listenin tamamının da her işlemde teker teker gözden geçirilmesi gerekecektir. Böyle bir liste yapılandırılmamış olduğundan her tek nesnenin rengini, biçimini, dokusunu, olası kullanılışlarını vb. ayrı ayrı içermek zorunda olacağından sonsuz olacaktır.

## Formel Kural ve Olguların Bağlamı

Bütün bunlar, insanların bazen başka her şeyden yalıtılmış çıplak verileri alıp, eskiden depolanmış veriler ışığında işlemeye çalışmadığı anlamına gelmiyor. Sherlock Holmes'un ve polisiye edebiyatın diğer bütün ünlü detektiflerinin ve kendisini tanımadığı bir durumun içerisinde bulan herkesin yaptığı da budur. Ama bu durumlarda bile, veri işleme sürecini yönlendiren bir genel bağlamın bulunması şarttır.

Örneğin, dünyayı ziyaret etmek zorunda kalan bir Merih'linin durumu buna çok benzeye-

## Mecazi İfadelerin Önemi

*Yapay zeka iddialarına fenomenologların getirdiği eleştirileri gözden geçirirken, sık sık fenomenologların kullandığı mecazi ifadelere başvurmak zorunda kaldık.*

*Kuşkusuz, bütün kavramların açık seçik ve formel bir biçimde tanımlanması gerektiğini veri kabul eden yapay zeka paradigması açısından bakıldığında, bu bir kusur.*

*Ama diğer yandan fenomenologların kalkış noktasının, verili somut davranışın mümkün olduğunca zengin ve ayrıntılı bir betimlemesi olduğunu söylemiştik. Böyle bir amaç için mecazi ifadeler vazgeçilmez bir araç rolünü oynuyorlar. Bu bakımdan, mecazi ifadelerin fenomenoloji için yöntemsel bir önemi var.*

*Ancak bu çerçeve yazıda George Lakoff ve Mark Johnson, mecazi ifadelerin fenomenoloji için, yöntemseli aşan bir öneme sahip olabileceğini düşündürüyorlar. Eğer insanların dünyası, gerçekten fenomenologların dediği gibi, insani ilgiler, tasalar, kaygular vb. ile yapılandırılmışsa, o zaman çevremezdeki eşyaya insani yüklemler atfetmek, yalnızca şiirsel değil, onun yapılandırılmasındaki insani kayguların yerini göstermesi bakımından doğru da olabilir.*

*Ancak Lakoff ve Johnson'un iddiaları bundan da ilerig idiyor. Onlar yalnızca mecazi dil kullanımının doğru olabileceğini değil, kavramsal sistemimizin tamamının mecazi olduğunu iddia ediyorlar. Benzer bir iddia filozof Wittgenstein tarafından da, algılama ile ilgili olarak öne sürülmüştü. O da her görmenin, gördüğümüz şeyi ''bir şey gibi görmek'' (seeing as...) olduğunu söylüyordu.*

*Bu iddialar doğruysa, benzetmelerin üzerine kurulacağı temel bilgi bitlerine ulaşılamaz demek. Eğer bir şeyi daima başka şeylere benzeterek, başka şeylerden hareketle algılıyorsak ve bütün bu benzetmelerin kaynağı kültürel olarak yorumlanmış gövdeyse, bu bilgi eninde sonunda kendisi bilgi olmayan bazı yaşantılara dayanıyor demektir. Bu yaşantıların kendisi bile, bu yazarlara göre tutarlı bir bütün oluşturmuyor. Kimi belli bir tür gövdeye (örneğin ön-arka ve aşağı-yukarı eksenlerinde esimetrik bir yapısı olan bir gövdeye) sahip olmakla ilgili, kimiyse belli bir kültürün üyesi olmakla. Ancak her halükârda, sözkonusu olan (bilgi ya da kavram değil) yaşantının kendisi olduğundan, bunun bir bilgisayar tarafından temsil edilmesi sözkonusu bile olamaz.*

## Yaşamımıza Giren Kavramlar

*Birçok kişi için mecaz, şiirsel imgelemin ve retorik akışın aracıdır; olağan değil, olağanüstü dile ait bir meseledir. Bunun yanısıra mecaz, yalnızca dilin bir niteliği olarak -düşünce veya eylemin değil- sözcüklerin meselesi olarak görülür. Bu nedenle de birçok kişi yaşamlarını mecazsız da pekala sürdürebileceklerine inanırlar. Oysa biz, tam tersine, mecazın gündelik hayata yayılmış olduğunu, yalnız dile değil, düşünce ve eyleme de girmiş olduğunu düşünüyoruz. Terimleriyle düşünüp eylediğimiz olağan kavramsal sistemimiz, doğası gereği mecazi'dir.*

*Düşüncelerimize hükmeden kavramlar yalnızca akla ilişkin değillerdir. En ince ayrıntılarına kadar gündelik işlevlerimize de hükmederler. Kavramlarımız algılayışımızı, dünyada yaşayış tarzımızı ve başka insanlarla ilişkimizi yapılandırırlar. Bu nedenle kavramsal sistemimiz gündelik gerçekliğimizi tanımlamada canalıcı bir rol oynar. Eğer kavramsal sistemimizin büyük ölçüde mecazi olduğunu öne sürmekte haklıysak, o zaman düşünüş tarzımızın, yaşantımızın ve her gün yaptıklarımızın da büyük ölçüde mecazi olduğunu söyleyebiliriz.*

*Ancak kavramsal sistemimiz, bizim normal olarak bilincinde olduğumuz bir şey değildir. Gün boyu yaptığımız küçük küçük hareketlerde, genellikle belli doğrultularda hemen hemen otomatik olarak düşünür ya da eyleriz. Bu doğrultuların ne olduğu hiç de açık değildir. Bunu bulmanın bir yolu, dile bakmaktır. İletişim, düşünme ve eylemede kullandığımız kavramsal sisteme dayalı olduğundan, dil bu sistemin neye benzediği konusunda güçlü bir veri kaynağıdır.*

*Gündelik dile baktığımızda, kavramsal sistemimizin mecazi nitelikte olduğunu görüyoruz. Algılayışımızı, düşünüşümüzü ve yaptıklarımızı şekillendiren mecazların hangileri olduklarını ayrıntılı olarak belirlememize yardımcı olacak kaynakları da gündelik dilde bulabiliyoruz.*

*Bir kavramın mecazı olmasının ve gündelik etkinliği biçimlendirmesinin ne anlama geldiğini sergilemek için, TARTIŞMAK kavramıyla ve TARTIŞMAK SAVAŞTIR kavramsal mecazıy-*

*la başlayalım. Bu mecaz gündelik dilimize çeşitli ifadelerle geçmiştir:*

*TARTIŞMAK SAVAŞTIR*

*İddiaların* savunulamaz.

*Argümanım* bütün zayıf noktalara saldırdı.

*Eleştirileri* tam hedefe yönelmişti.

*Argümanını* yerle bir ettim.

*Ona karşı hiçbir tartışmayı* kazanamadım

*Karşı mı çıkıyorsun?* Vur *öyleyse.*

*Eğer o* stratejiyi *kullanırsan, seni* ezip geçer.

*Bütün argümanlarımı* darma duman etti.

*Tartışmalardan savaş terimleriyle yalnızca* bahsetmekle *kalmadığımızı görmek önemli. Tartışmaları gerçekten de kazanıp kaybederiz. Tartıştığımız kişiyi hasım gibi görürüz. Onun görüşlerine saldırır, kendi görüşlerimizi savunuruz. Stratejiler planlar ve kullanırız. Eğer bulunduğumuz noktanın savunulamaz olduğunu görürsek, onu terkedip başka savlar ararız. Tartışırken* yaptığımız *şeylerden çoğu, kısmen savaş kavramı tarafından biçimlendirilmiştir. Fiziksel bir çarpışma yoksa da, sözel bir çarpışma mevcuttur ve tartışmanın yapısı -saldırma, savunma, karşı -saldırı vs.- bunu yansıtır. İşte bu anlamda TARTIŞMA SAVAŞTIR mecazını bu kültürde yaşarız; bu, tartışmada yaptığımız eylemleri yapılandırır.*

*Tartışmaların savaş terimleriyle ele alınmadığı, kimsenin kazanıp kaybetmediği, saldırma ve savunma olmadığı bir kültür hayal etmeye çalışın. Tartışmanın bir dans olarak, katılımcıların danscılar olarak görüldüğü; hedefin ise dengeli ve estetik bir gösteri sergilemek olduğu bir kültür hayal edin. Böyle bir kültürde insanlar tartışmalara başka türlü bakacaklar, baş-ka türlü yaşayacaklar, başka türlü yürütüp onlardan başka türlü bahsedecekler. Ama biz onların tartışmakta olduklarını düşünmeyeceğiz: bambaşka bir şey yapıyor olacaklar. Yapmakta oldukları şeye ''tartışma'' adını vermek bile bize garip gelecek. Belki de onların kültürüyle bizimki arasındaki farkı tanımlamanın en iyi yolu, bizim söylem biçimimizin savaş tarafından yapılandırıldığını, onlarınkininse dans terimleriyle yapılandırıldığını söylemektir.*

*Bu, bir mecazi kavramın, TARTIŞMAK SAVAŞTIR'ın, yapıp ettiklerimizi ve tartışırken yaptıklarımızı, nasıl gördüğümüzü nasıl yapılandırdığının bir örneğidir.* ***Mecazın özü, bir tür şeyi, bir diğerinin terimleriyle anlamak ya da yaşamaktır.*** *Tartışma, savaş kavramının bir alt sınıfı değildir. Tartışma ve savaşlar başka başka şeylerdir - sözel söylem ve silahlı çelişki. Ve sergilenen eylemler farklı türden eylemlerdir. Ama TARTIŞMA kısmen SAVAŞ terimleriyle yapılanır, anlaşılır, sergilenir ve bahsedilir. Kavram mecazi olarak yapılanmıştır, etkinlik mecazi olarak yapılanmıştır ve sonuç olarak dil mecazi olarak yapılanmıştır.*

*Üstelik bu, tartışma yapmanın ve tartışmadan bahsetmenin olağan yoludur. Bir görüşe saldırmaktan bahsederken normal olarak kullanacağımız sözcükler ''bir görüşe saldırmak'' olacaktır. Tartışmalardan geleneksel bahsediş tarzımız, farkında bile olmadığımız bir mecaz barındırır. Mecaz yalnızca kullandığımız sözcüklerde değildir - tartışma kavramımızın ta kendisindedir. Tartışmanın dili şiirsel, hayali, retorik değildir; gerçeğe uygundur. Tartışmalardan bu şekilde bahsetmemizin nedeni, onları bu şekilde kavramamızdır - şeyleri kavrama tarzımıza uygun davranırız.*

*Şu ana kadar öne sürdüğümüz en önemli iddia, mecazın yalnızca bir dil, yani sözcük meselesi olmadığı. İnsanın düşünce sürecinin büyük ölçüde mecazi olduğunu ileri süreceğiz. İnsanın kavramsal sisteminin mecazi olarak yapılandığını ve tanımlandığını söylerken kasdettiğimiz buydu. Dilsel ifadeler olarak mecazların mümkün olabilmelerinin nedeni, insanın kavramsal sisteminde mecazların bulunmasıdır.*

*Bu nedenle burada içinde mecazlardan söz ettiğimiz her defasında, örneğin TARTIŞMA SAVAŞTIR derken, mecazın,* ***mecazi kavram*** *anlamına geldiği anlaşılmalıdır.*

## Mecazi Kavramların Sistematikliği

*Tartışmalar genellikle bir düzen izler, yani tartışırken tipik olarak yaptığımız ve yapmadığımız belli şeyler vardır. Tartışmaları kısmen savaş terimleriyle kavramlaştırdığımız olgusu, sistematik olarak tartışmaların aldığı biçimi ve tartışırken ne yaptığımız üzerine konuşma tarzımızı etkiler. Mecazi kavram sistematik oldu-*

ğu için, kavramın bu yönünden bahsederken kullandığımız dil de sistematiktir.
TARTIŞMA SAVAŞTIR mecazından, savaş sözlüğünden aldığımız ifadelerin, örneğin bir görüşe saldırmak, savunulamaz, strateji, yeni saldırı hattı, kazanmak, saf kazanmak vb. gibi terimlerin, tartışmanın savaşıma benzeyen yönlerine ilişkin sistematik bir konuşma tarzı biçimlendirdiğini gördük. Bu ifadeleri tartışmalardan bahsetmekte kullanırken, o anlamlar içermeleri tesadü değildir. Savaşın kavramsal ağının bir bölümü, kısmen tartışma kavramını niteler, dil de buna uyar. Dilimizdeki mecazı ifadeler sistematik bir biçimde mecazı kavramlara bağlı olduğu için, mecazı kavramların niteliğini incelemek ve etkinliklerimizin mecazı niteliklerini anlamak için mecazı dilsel ifadeleri kullanabiliriz.
Gündülk dildeki mecazı ifadelerin bize gündelik etkinliklerimizi yapılandıran kavramların mecazı niteliklerine ilişkin nasıl öngörü sağladığını anlamak için, çağdaş Türkçe'de yansıdığı şekliyle VAKİT NAKİTTİR mecazı kavramına bir bakalım.

VAKİT NAKİTTİR

Vaktimi harcıyorsun..

*Bu makine saatler* kazanmanı *sağlayacak*.

*Sana* verecek *zamanım* yok.

*Bu günlerde vaktini nasıl* harcıyorsun?

*Lastiğin patlaması bir saatime* maloldu.

*Onun üzerine çok zaman* yatırımında *bulundur*.

*Ona* ayıracak yeterli zamanım *yok*.

*Vaktin* tükeniyor.

*Zamanını iyi* hesaplaman *lazım*.

*Pinpon oynamak için biraz vakit* ayır.

*Bu zamanını vermeye* değer mi?

*Fazla* zamanın kaldı mı?

Çalınmış *zamanları kullanarak yapıyor*.

*Zamanını* kârlı *kullanmıyorsun*.

*Hastalanınca çok zaman* kaybettim.



*Bana vakit ayırdığın için teşekkürler.*

*Kültürümüzde zaman değerli bir maldır. Hedeflerimize ulaşmak için kullandığımız kısıtlı bir kaynaktır. İş kavramının modern Batı kültüründeki gelişimi nedeniyle, iş aldığı zamanla tipik olarak bağlantılandırıldığı, zaman da kesin birimlere ayrıldığı için, insanlara saat, hafta ya da yıl hesabıyla para ödemek gelenekselleşmiştir. Bizim kültürümüzde birçok anlamda VAKİT NAKİTTİR: telefon mesajı birimleri, saat başı ücretler, otel odası fiyatları, topluma borcunuzu ''süre doldurarak'' ödemeniz... Bu uygulamalar insan türünün tarihinde görece yenidir ve her kültürde yer almazlar. Modern endüstrileşmiş toplumlarda ortaya çıkmışlardır ve incelikli bir biçimde temel gündelik etkinliklerimizi yapılandırırlar. Zaman değerli bir malmış -kısıtlı bir kaynak, hatta paraymış- gibi* hareket ettiğimiz *olgusuna tekabül eden bir biçimde, zamanı böyle* kavrarız. *Bu nedenle zamanı harcanabilecek, iyi ya da kötü yatırımlarda bulunulabilecek, tasarruf edilecek ya da çarçur edilecek bir şey olarak anlar ve yaşarız.*

*VAKİT NAKİTTİR, VAKİT KISITLI BİR KAYNAKTIR ve VAKİT DEĞERLİ BİR MALDIR hep mecazi kavramlardır. Biz para, kısıtlı kaynaklar ve değerli mallara ilişkin günlük deneyimimizi zamanı kavramlaştırmak için kullanmaya başladığımızdan beri mecazidirler. Bu insanların zamanı kavramlaştırmaya zorunlu oldukları tek biçim değildir; kültürümüze bağlıdır. Zamanın hiç de bu anlamlara gelmediği kültürler vardır.*

*VAKİT NAKİTTİR, VAKİT KAYNAKTIR ve VAKİT DEĞERLİ BİR MALDIR mecazi kavramları, altkategorileştirmeye dayalı tek bir sistem kurarlar çünkü bizim toplumumuzda para kısıtlı bir kaynaktır. Kısıtlı kaynaklarsa değerli mallardır. Bu altkategorileştirme ilişkisi mecazlar arasındaki intikal ilişkisini niteler. VAKİT NAKİTTİR, VAKİTİN KISITLI BİR KAYNAK olduğunu ifade eder.*

*En spesifik mecazi kavramı -bu durumda VAKİT NAKİTTİR- kullanarak tüm sistemi nitelemek yolunu kullanacağız. VAKİT NAKİTTİR mecazının altında sıralanan ifadeler arasından bazıları (harcamak, yatırmak, hesaplamak, kârlılık, maliyet) spesifik olarak paraya, diğerleri kısıtıl kaynaklara (kullanmak, tüketmek, elinde yeterince olmak, bitirmek), diğerleri de değerli mallara (sahip olmak, vermek, kaybetmek, müteşekkir olmak) atıfta bulunmaktadır.*

***Sürecek...***

bilir ama eğer sözkonusu Merih'li insanların tasa ve tasarılarından hiçbirini paylaşmıyorsa, belli bir duruma ilişkin olan verileri ilişkin olmayanlardan, asli olanı tali olandan ayırmaktaki durumu, yapay zeka ile donatılmış bir bilgisayarın ki kadar umutsuz olacaktır.

Yalnız tanımadığımız durumlarda değil, başka bazı sınırlı bağlamlarda da formel kurallara göre davranırız. Örneğin daha önce briç, satranç gibi oyunların bazı aşamalarında oyuncuların belli birtakım formel kurallara göre hesaplamalar yaptıklarını, mümkün hamleleri ''saydıklarını'' görmüştük. Ancak bu, yapay zeka iddialarını destekleyen değil aksine çürüten bir veridir. Formel kurallara göre davranmanın ne demek olduğunu bildiğimiz, bu şekilde davrandığımızda bunun bilincinde olduğumuz içindir ki, başka bir sürü durumdaki davranışlarımızın da formel kurallar tarafından türetilmediğini söyleyebiliyoruz.

## Düalizm ve Yapay Zeka

Bu noktada, fenomenolojinin yapay zekanın iddialarına değil, yaygın olarak paylaşılan, hatta sağduyusal kabul edilen bazı iddilara karşı da öne sürdüğü görüşlere geliyoruz.

Fenomenolojiye göre biz zihnimizde, içinde yaşamakta olduğumuz dünyanın bir modelini barındırmaz, dünyayla ilişkilerimizi bu modelden hareketle kurmayız. İlişki kurduğumuz, anlamlı şeyler, nesneler hakkındaki zihinsel tasarımlarımız değil, doğrudan doğruya şeylerin, eşyanın kendisidir.

Trafik işaretlerinden, işliklerden vb. oluşan kamusal dünya düşünüldüğünde, bu görüş sağduyuya uygun görünüyor. Fakat ya kişinin kendisinin olan kişisel tasarılar, öznel olgular...? Bunların da zihinsel olmadığı, bir biçimde zihinlerde temsil edilmemekte olduğunu iddia etmek sağduyunun köklü kabullerinden birini çelmiş olmuyor mu?

Ancak fenomenolojiye göre, sağduyusal diye nitelendirdiğimiz bu görüşün temelinde yatan, insanların dünyası ile fiziğin evreninin birbirlerine karıştırılması yatıyor. Oysa fenomenolojinin alternatif görüşüne göre, nasıl genel olarak insanların kamusal hedefleri gündelik eşyanın tanımlayıcı bir parçasıysa, onlara sonradan eklenen bir zihinsel yük değilse, kişisel anılar, tasarılar da, kişiyi çevreleyen eşyaya ve bu eşyayla kurulan etkileşim tarzına içkin olan özelliklerdir. Örneğin anı dediğimiz şey eski bir iskemlenin tanıdık bir görünüşü, bir kez yaralanılmış olan bir sokak köşesinin

ürpertici havasıdır. Daha açık bir şekilde söyleyecek olursak, anıların birikmesi, formel olarak betimlenebilecek birtakım zihinsel tasarımların depolanması değil, merkezi sinir sisteminden başlayarak insanın bütün varlığının belli bir biçimde dönüşmesidir.

Böyle bir görüş kabul edildiğinde, yapay zeka iddialarının inandırıcılığını neden yitirdiğini şimdi görebiliriz. Hernekadar sinir sistemi dış dünya ile fiziksel enerji alışverişinde bulunan fiziksel ve sistemse de, her fiziksel enerji birimine tekabül eden bir bilgi biriminden bahseden bir bilgi işlem dilinin mümkün olduğu anlamına gelmez. Aksine beynin ve genel olarak gövdenin fonksiyonları, beynin ve gövdenin yapısından bağımsız olarak betimlenmeyebilir.

Bunların birbirlerinden bağımsız olarak betimlenebileceği görüşünün çekiciliği, daha önce de söylediğimiz gibi, kaynağını Dascartes'de bulan ve artık sağduyuya da içkin bir hale gelmiş olan, düalizmdir.

Descartes'a göre, dünya ile tanışıklığımız, saf fiziksel hareket olarak varolan dış dünya ile, kendisi maddi ve fiziksel olarak varolmayan zihnin karşılaşmasından doğar. Bu ikisi arasındaki ilişkinin niteliğinin ne olduğu tam olarak açık değilse de, zihin, bir şekilde, bu saf fiziksel hareket olarak varolan uyarılardan kalkarak tanıdık eşyalar ve anlamlar dünyasını kurar.

Ancak yapay zeka programının devraldığı biçimiyle bu görüş daha da sorumlu bir hale gelir. Çünkü Descartes, nasıl'ını açıklayamazsa da, bir şekilde tanıdık eşyalar, anlamlar, tasalar, hedefler dünyasının zihinsel olarak kurulduğunu veri kabul eder. Oysa, yapay zeka iddiası tam olarak da bu nasıl'ın, formel bir bilgisayar programı tarafından betimlendiği iddasıdır. Ancak bu betimlemenin ürününün bize henüz insan amaçları ve tasarılarını vermediğini, sadece her durumdan bağımsız olarak tanımlanmış bazı nesneler ve ölçütler ve bunların nasıl işleneceğini belirleyen açık seçik kurallardan bir model oluşturduğunu gördük.

Bu modelin, insanların zeka gerektiren davranışlarının modeli olduğu iddia ediliyor. Ama zeka gerektiren davranışlar daima belli bir nedenden, bir tasadan ötürü ya da bir şey uğruna yapılır. Yapay zeka modellerinin değerler, hedefler, tasalar gibi zeka sahibi olmaktan ayırdedilemeyen olgularla nasıl bağdaştırılmaya çalışıldığını bundan sonraki bölümde inceleyeceğiz.

## Değerler ve Sınırları

Pratik dünyanın, bizim çabalarımızla örülmüş, bizim ilgilerimizle yapılandırılmış bir dünya olmasından ötürü, içerdiği nesnelerin anlamının hesaplamayı andırır işlemler sonucu keşfedilmesinin gerekmediğini, nesnelerin zaten bizim bir ilgi alanımızın parçası olarak verildiğini görmüştük. Fenomenolojik görüşe göre, verili olanın zaterı daima bir ilgi alanı olarak verilmiş olması, belli bir tasarıya ya da hedefe ilişkin olan öğelerin keşfedilmesi sorununu ortadan kaldırıyor.

Yapay zeka programı bellekte depolanmış olgular ve kurallarla, bu bilgilerin aracı olacağı hedefler arasındaki ilişkiyi nasıl kuruyor?

Bu sorunu çözümlemek için araştırmacılar olgular ve kuralların yanında, ek olarak bir de yararlılık fonksiyonu **(utility function)** ya da değerler tablosu programlamayı öngörüyorlar.

Böyle bir tablo ya da fonksiyon, bir hedef verildiğinde, erişilebilen bütün nesne ve olgulara, bu hedefe ulaşabilmeye katkıları açısından somut ve özgül bir değer veriyor.

Böyle bir çözümün sorunlarını, kendisi de makinelerin örüntülerin tanımasını mümkün kılacak programlar üzerinde çalışan Satosi Watanabe şöyle dile getiriyor:

''İnsanların değerlendirmelerinde kullandıkları değerler, hem genel hem de yarı- duygusaldır. Oysa, bir makineye programlanabilecek olan hedefler, çok özgül olarak belirlenmiş bir değerler tablosu ya da çok açıkça formelleştirilmiş bir ölçüt olmak zorundadır... Aradaki fark önemsiz gibi görülebilirse de aslında çok köklüdür. Bu farkı çok kabaca şöyle ifade edebiliriz: Bir makinenin ancak hedefleri olabilir, oysa bir insanın *değerleri* vardır. Gerçi kuşkusuz insanların da hedefleri vardır ama bu hedefler birtakım değerlerden türemiştir ve bir makinenin eylemlerinde olduğu gibi, insanın eylemlerinin nihai belirleyicisi rolünü oynamazlar.''

Hedef bir kez belirlendikten sonra makine de bu hedefe ulaşmak için, bir insanın davrandığı gibi davranabilir. Aynı şekilde insanların yararcı davranışları da bilgisayar tarafından benzetlenebilir. Ama ancak bütün seçenekler sabitse ve bilgisayara verilmişse ve makine her seçeneğin yararını niceliksel olarak ifade edebiliyorsa bu mümkündür. Ama makine hiçbir zaman niceliksel olarak ifade edilmiş bu yararın kaynağı olan değere ulaşamaz.''

Watanabe'ye göre makinenin ulaşamadığı bu değerler, zekanın asli bir ögesini oluştururlar. Çünkü, bir kere, her deneyim birbirinden farklı binlerce hipotez ve seçeneği birden doğrulayabilir. Bu seçenek ve hipotezlerin, herhangi bir yarar fonksiyonuna göre değerlendirilmeden önce, sayılarının sınırlandırılması, ancak çok belirsiz ve muğlak bir şekilde tasarlanan, yalınlık gibi bazı estetik ölçütlere göre gerçekleştirilir. Daha da önemlisi, Watanabe'ye göre herhangi iki nesneyi ele aldığımızda, bunların daima aynı sayıda ortak özelliğe (yükleme) sahip olacağı kanıtlanabilir. Pratikte bunun bizim için bir sorun yaratmaması, bu ortak yüklemlerden bazılarının daha önemli bazılarınınsa daha önemsiz olduğunu kestirebilmemizdendir. Neyin önemli neyin önemsiz olduğu hakkında verdiğimiz kararın nihai dayanağı ise değerlerimizdir.

Ancak bu değerleri, bir hedefler listesinden ayırdeden nedir? Buraya kadar sözünü etmiş olduğumuz tek farklılık, değerlerin daha belirsiz ve muğlak olması. Ancak bu ikisini birbi-

rinden ayıran tek fark eğer, değerlerin daha muğlak ve belirsiz olması ise, bu onların da matematiksel bir biçimde, örneğin niceliksel bir ölçek üzerinde tek bir nokta olarak değil de, bir aralık olarak temsil edilemeyeceği anlamına gelmez.

Çünkü aslında değerlerden sözeden felsefi gelenek de, insanların dünyasını matematize edilebilir, nesnelleştirilebilir olgular halinde tasarlamaya çalışan gelenekle aynı gelenektir. İnsanın pratik sezgileri ve becerileri, değerler şeklinde anlaşıldığı takdirde, değerin kendisi de, nesnelere eklenecek, diğer özelliklerinin yanında bir özellik olarak görülebilir.

Watanabe'nin farkında olduğu ancak doğru bir biçimde dile getiremediği sorunu anlamak için, nesnelerle, içinde yeraldıkları ilgi alanı arasındaki köklü ontolojik farklı kavramamız gerekir.

## İlgi ve Seyir

Nesnelerin daima belli bir ilgi alanının parçası olarak varolmaları, ilgi alanlarının daima bizim tasa ve tasarılarımız tarafından yapılandırılmış olması, nesnelerle ancak somut bir faaliyetin, belli bir işin içerisinde karşılaştığımız anlamına gelir. Bu faaliyeteki alışkanlığımız, işteki ustalık derecemiz ise sözkonusu ilgi alanını nasıl algıladığımızı belirler. Örneğin, usta bir satranç oyuncusu için, satranç tahtasındaki bir pozisyon hiçbir zaman sadece duruk bazı taşlardan oluşmaz; o, tahtanın üzerinde bazı çekim güçleri, tehditler, tıkanık alanlar ve gücül hareketlilikler görür.



**Yapay zekâ iddialarının, fiziğin doğal dünya ile ilgili olarak gerçekleştirdiği soyutlamayı, insanların dünyası ile ilgili olarak gerçekleştirme çabasından kaynaklandığını görmüştük. Ancak fizik doğal dünyayı ilgilenilen değil, seyredilen bir dünya olarak tasarlar. Bu bakımdan modern fiziğin ilk başarılarının astronomi alanında olması rastlantısal değildir.**

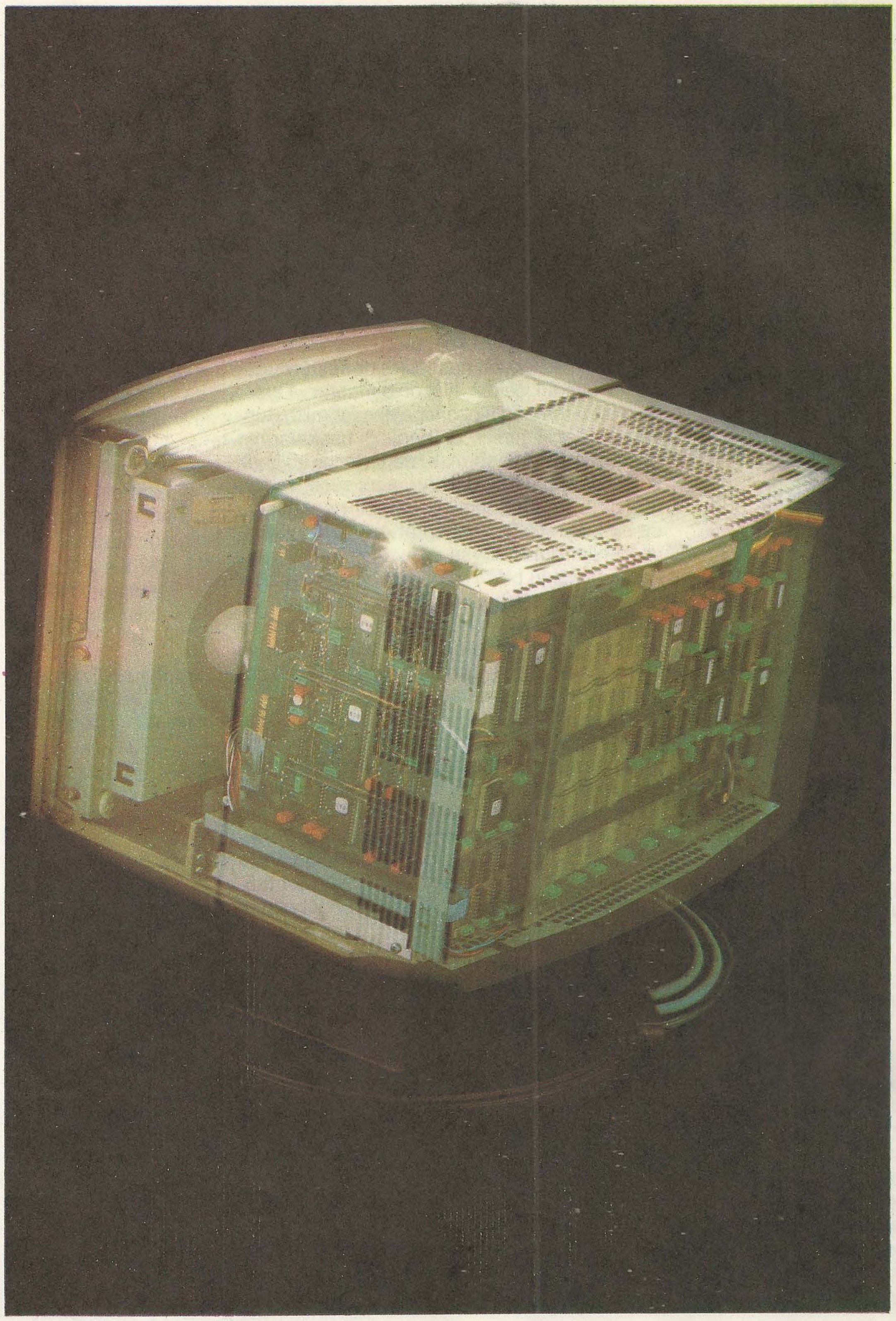
Sıradan algısal süreçler sözkonusu olduğunda ise hepimiz birer usta sayılabiliriz. İlgilerimiz, tasalarımız, tasarılarımız görüş alanımızdaki çeşitli nesnelerden biri, nesnelerin ek bir özelliği olmadığı için, bu alan üzerinde böyle düzenleyici bir etki yaparlar.

Oysa, dünyayı formel tasarımlarını çıkarmanın mümkün olduğu ayrık bir olgular kümesi halinde tasarlamayı hedefleyen felsefi gelenek, aynı zamanda, dünyayı ilgilenilen bir alanlar bütünü olarak değil, seyredilen bir nesneler yığını olarak tasarlamıştır. Heidegger geleneksel felsefenin ve çağdaş teknolojinin, bizim hayatımızı yönlendiren kayguları, seyredilebilen ve kontrol edilebilen nesnelere dönüştürdüğüne ilk dikkati çeken filozof olmuştur. Oysa, Sokrates'ten beri filozoflar eylemlerimizi, dünyamıza bağlılığımızı, bu bağlılığı başkalarının bağlılığıyla karşılaştırmak, değerlendirebilmek ve yargılayabilmek uğruna, açık seçik ve formel bir biçimde betimlenebilir bir nesneye ya da bir tasarıma **(representation)** dönüştürmeye çalışmışlardı. Oysa dünyamız bağlılığımız tuhaf ve köklü bir özelliği var; bu şekilde açık seçik bir nesne haline getirilir getirilmez, bizim üzerimizdeki kavrayışını yitiriyor, bizim dışımızdaki seçeneklerden biri haline geliyor ve böylelikle de pratik dünyamızı düzenleyen unsurlardan biri olmaktan çıkıyor.

Bu noktada bu bölümün başında sormuş olduğumuz soruyu tekrar sorabiliriz? Bazı davranışların bir yararlılık fonksiyonu temelinde yapılması ile nihai bir kaygu ya da tasadan haraketle yapılması arasındaki fark nedir? Watanabe'nin bu ikisi arasındaki farklardan birine dikkat çektiği görmüştük. Bir değerler tablosu ya da yararlılık fonksiyonu özgül olmak zorundadır.

Oysa insan davranışlarının esnekliği, genel olanı belli amaçlar ışığından kavramaktan ve dilini yeni amaçlar doğrultusunda, kurallı olmayan bir biçimde genelleştirebilmesinden kaynaklanmaktadır.

Daha da önemlisi, bir bilgisayara programlanan yararlılık fonksiyonunun, bilgisayarın depolamış olduğu olgular ve verilerle ancak dışsal bir ilişki kurmasıdır. Oysa insanın ''nihai kaygu ya da tasaları'' terimleriyle işaret etmeye çalıştığımız gerçeklik, bir hedefler dizisinde yeralan son halka değildir. Aksine bu kaygularla verili her durum ve olgunun bir parçası olarak karşılaşılır ve kaygu verili durumu ya-



pılandıran unsurlardan biri olarak işlev görür. Oysa makinenin yararlılık fonksiyonu ya da değerler tablosunun olgularla ilişkisi dışsal olduğundan, veri işleme sürecine belli fasılarla ara verip, son durumu değerler tablosu ışığında gözden geçirmek gerekir. Başka bir deyişle, kavrama ve değerlendirme bilgisayar için iki ayrı işlem olmak durumundadır. Oysa insanlar için bu, ancak çok özgül bazı durumlarda ortaya çıkan yapay bir ayrımdır.

Değerlendirmenin, veri işleme sürecine müdahale eden bir olaymışcasına kavranmasına, birçok somut yapay zeka uygulamasında rastlıyoruz. Örneğin duygu ve güdülerin zeki davranışlar için önemli olduğuna karar veren Herbert Simon ve Walter Reitman, yine de kendi uygulamalarında, duygu ve güdüleri bir müdühale **(interrupt)** olarak benzetlemişlerdir. Onlara göre bilgisayarın 'duygulanması' belli bir veri işlem süreci sürerken, o veri eşlem sürecine ilişkin olmayan başka bir süreç ya da sorunla ilgili mülahaza ve bilgilerin araya girmesinden ibarettir. Oysa fenomenologlara göre, duygu, ilgi ve tasalar insanların bilişsel süreçlerine dışardan müdahale etmez, onlara eşlik eder ve onları yönlendirirler.

## Zekâ ve Sanatsal Yaratım

Buraya kadar nesnelerin daima belli bir ilgi alanının, bir işin parçası olarak verildiğinden ve bunların kendi dışlarında bir hedefe göre değerlendirilmediğinden sözettik. Bu görüşe göre bir nesnenin önce farkına varıp sonra onunla ilgilenilmez. Her nesne önce ve daima zaten bir ilginin objesi olarak vardır.

Bu açıklamayı somutlayabilmek için Heidegger sanatsal yaratım benzetmesine başvurmuştur. Hatta o kadar ki, zekayı bir bilim adamının faaliyetlerinden hareketle kavramsallaştırmaya çalışan yapay zeka programının aksine, Heidegger bütün insan davranışlarının temelde sanatsal yaratımla önemli ortak özellikler paylaştığını öne sürmüştür. Şöyle ki, tıpkı sanatsal yaratımda olduğu gibi, çoğu insan faaliyetlerinde de, faaliyetin amacını, bu amaç gerçekleşmeden önce açık seçik olarak dile getiremeyiz. Dolayısıyla, herhangi bir davranışı da önceden formüle edilmiş bir problemin çözümü ya da bir hedefin aracı olarak betimleyemeyiz.

Bu benzetmenin alışkanlıklarımızın çoğunun önemli birçok özelliğini açıklayıcı bir yönü varsa da, genel bir açıklama olarak yeterli değildir. Çünkü örneğin ihtiyaçlarımızın çoğu ol-

dukça belirgindir ve bu yüzden bizim için, çok özgül bazı hedefler tarif ederler. İhtiyaçların bu belirginliği ile zeka arasındaki ilişkiyi anlamak için, ihtiyaç kavramıyla ne kasdettiğimiz hakkında daha aydınlık bir fikre sahip olmamız gerekir.

## Zeka ve İhtiyaçlar

Yapay zeka çalışmalarına kaynaklık eden felsefi ve psikolojik gelenekler, ihtiyaç kavramı üzerinde fazla durmamış oldukları için, ihtiyaç olgusunu benzetlemeye çalışan uygulamalara da pek rastlanmamaktadır. O kadar ki, *''Makineler ve İnsanlar''* adlı birmakalede, N.S. Sutherland şöyle yazabilmektedir.

''Hayatta kalma ve kendi kendini sürdürebilme, insanda beyne bir dizi dürtü ve hedefin, genetik olarak verilmesiyle sağlanmıştır. Bu genetik dürtü ve hedeflerin en açık olanları açlık, susuzluk, cinsellik ve acıdan kaçınmadır. Aslında bu dürtülerin hepsi, bir anlamda, fazlalıktır. Çünkü onları ikame edebilecek karmaşık bilgi işlem süreçleriyle donanmış zeki sistemler tasarlamak pekala mümkündür.''

Bu gözlemin doğruluğunu sınamak için bir kez daha Heidegger'in kullandığı sanatsal yaratım ve insan faaliyetleri benzetmesine dönmekte yarar var. Gerçekten de harhangi bir şeye ihtiyaç duyduğumuzda, önce tam olarak neye ihtiyacımız olduğunu kestiremeyebiliriz. Neye ihtiyacımız olduğunun keşfedilmesi de, çeşitli nesne ve faaliyetlerin nesnel olarak tanımlanmış ve belirgin bir ölçütle karşılaştırılmasıyla olmaz.Bunu sağlayan, daha çok ''tatmin duygusu'' denebilecek bir duygudur. Bu duygu, insana, ta başından beri neye ihtiyacı olmuş olduğunu da bildirir. Ancak bu geriye dönüşlü bir bilgidir ve başlangıçta ihtiyacımızı belirginleştiremediğimizi bizden gizler. Başka bir deyişle tatmin, daha önceden ancak genel bir rahatsızlık olarak hissedilmiş ihtiyacı, özgül bir nesneye duyulmuş bir ihtiyaç olarak yeniden tanımlar. Dolayısıyla herhangi bir ihtiyacın ilk tatmini, yaratıcı bir keşfe benzetlilebilir.

**Sören Kierkegaard, insanların tatmin deneyimleri sonucu, eski ihtiyaçlarını yeniden ve daha özgül biçimlerde yeniden tanımlamasının üzerinde önemle durmuş bir filozoftur. Böyle bir dönüşüm insanın bütün ilgi ve tasalarını ve bu ilgi ve tasaların belirledikleri ve düzenleri ilgi alanlarını ve bu ilgi alanlarından oluşan dünyayı dönüştürdüğü için, Kierkegaard bu tür dönüşümlere varoluşsal dönüşüm adını vermiştir.**

**Varoluşsal dönüşüm, bundan sonraki sayfalarda inceleyeceğimiz kavramsal ve kültürel dönüşümlerle birlikte fenomenologların en çok üzerinde durdukları olaylardır.**

## Alternatif Bir İnsan Tasarımı

Bu bölümü, yapay zeka alanında çalışanların, giderek daha çok ilgilenmek ve buluşlarından yararlanmak zorunda kaldıkları fenomenolojinin alternatif insan tasarımını çok kısaca özetleyerek bitireceğiz, Bugün ulaştıkları düzeyde fenomenolojinin insan tasarımı ile yapay zeka araştırmalarının veri kabul ettikleri insan tasarımı, birbirleriyle bağdaşmaz ve hatta neredeyse karşıt bir durumda. Ancak sıradan sağduysal bilginin benzetlenmesi çalışmaları geliştikçe, hiç değilse kimi araştırmacılar, bu karşıtlığın aşılabileceğini umuyorlar. Yine de biz burada karşıtlıkları vurgulayarak her iki tasarımın da daha net olarak görülebilmesini sağlamaya çalışacağız.

Yapay zeka araştırmacıları, kısmen de kullanmakta oldukları donanımın zorlamasıyla, mümkün olduğunca açıkça tanımlanabilir unsurlardan oluşan yalın davranışlardan hareketle, karmaşık davranışları anlamaya çalışıyorlar.

Oysa, daha önce de söylemiş olduğumuz gibi fenomenologlar için, önemli olan verili, karmaşık davranışların mümkün olduğunca ayrıntılı betimlenmesi. Yukarıda özetlediğimiz ihtiyaç çözümlemesi bile, ''açlık'', ''susuzluk'' gibi nisbeten yalın ihtiyaçlar değil ''cinsel arzu'' gibi içgüdüsel temeli zayıf olmakla birlikte daha karmaşık olan ihtiyaçlar temelinde anlaşılabilecek bir çözümleme.

Örnekleyelim: İnsanın açlığının bir besine olduğunu anlamasının, ancak bir kez besin aldıktan sonra mümkün olması, çok ikna edici bir açıklama olmayabilir. Ancak aynı açıklamayı cinsel arzuya uyguladığımızda, durum çok farklı oluyor. Belli bir insana cinsel arzu duyan biri, o ilişkiden tatmin olduğu takdirde, daha önceki cinsel arzusunu genel bir açlık olarak değil, o insana duyulan bir arzu olarak tarif etmeye başlayabilir. Böyle bir yeniden tanımlama, önceden verili olan bir hedefin gerçekleştirilmesi değil, belli bir süreç içerisinde sözkonusu insanın yapısal bir dönüşüme uğramasıdır.

## Kavramsal Dönüşümler

Ancak buraya kadar söylediklerimizden, fenomenolojik açıklamaların yalnızca cinsel arzu, sanatsal yaratım gibi geleneksel olarak zeki davranışların, rasyonalitenin kıyısında yeraldığı düşünülen olguları açıkladığı izlenimini ya-

ratmamalı. Aksine, son yıllarda kamuoyunun ve özellikle de bilgi-işlem ve yapay zeka ile ilgilenen kamuoyunun dikkatini fenomenoloji üzerine çeken önemli olaylardan biri, bilimsel devrimleri ve bilimsel kuramların yapısını inceleyen Thomas Kuhn un eserleri ve bu eserlerdeki iddialarla, fenomenolojinin görüşleri arasındaki yakınlıklar olmuştur.

Kuhn'a göre bilim adamları dünyaya kavramsal bir çerçeve, bir paradigma içerisinden bakarlar; belli birtakım olayları görebilmeleri, birtakım verilere sahip olabilmeleri ancak bu kavramsal çerçeveler sayesinde mümkün olur. Tıpkı fenomenologların sözünü ettikleri pratik tasa, kaygu ve tasarılar gibi, kavramsal çerçeveler de bilim adamı için dünyayı düzenler, neyin onlar için bir olgu olabileceğini belirlerler.

Bilimsel devrimler döneminde ise bu kavramsal çerçevenin kendisi değiştiğinden, bilim adamları hangi iddiaların doğru olduğunu sınayabilmek için, çatışan iddiaların herbirinin de gönderme yaptıkları ortak olgular bulamazlar: ''Verilerin kendisi değişmiştir. Bu yüzden bir anlamda, bilimsel bir devrimden sonra artık farklı bir dünyada çalışmakta oldukları söylenebilir.. Bir bilimsel devrim sırasında cereyan eden şey, sabit ve istikrarlı bir veriler kümesinin yeniden yorumlanması şeklinde anlaşılamaz.''

Ancak bilgi, yapay zeka araştırmalarında olduğu gibi nötr bazı veriler arasında kurulan sistematik ilişkiler şeklinde kavranırsa, bu kavramsal dönüşüm dönemleri anlaşılamaz. Çünkü Kuhn'a göre bilimsel ya da ampirik olarak nötr olan diller ya da kavramlar yoktur. Bizim amaçlarımız bakımından daha önemlisi, bu kavramsal çerçevelerin kendisinin bilim adamları için bir olgu olamamasıdır:

''Bilim adamları çalışmalarını yönlendiren paradimalarını belirlemede anlaşabilirler. Ama bu onların, bu paradigmanın yorumunda değil anlaşabilmek, tam bir yorumu ya da akılcı bir betimini **(rationalization)** verebilecekleri anlamına gelmez... Standart bir yorumunun olmayışı ya da pradigmanın herkesin üzerinde anlaşabileceği şekilde bir dizi kurala indirgenemeyişi, paradigmanın araştırmaları yönlendirmedeki etkililiğini zedelemez. Hatta, paradigmanın varlığı, ille de böyle tam bir kurallar dizisinin olduğu anlamına da gelmez... Paradigmalar, kendilerinden soyutlanabilecek bir araştırma kuralları dizisinden hem daha önce gelirler, hem daha bağlayacıdırlar hem de daha

tamamdırlar.''
Bu noktada bir kez daha yapay zeka araştırmacılarının problem çözümleri hakkındaki söylediklerine geri dönebiliriz. Verilerin depolanması ve belli bir hedef gözetilerek taranması modeli, bu kavramsal dönüşümleri benzetlemek şöyle dursun, hakkını da veremez. Oysa bu kavramsal dönüşümler problemlerin dile getirilmesini mümkün kılan ve hangi verilerin problemlerin çözümüne ilişkin ve önemli olacağını belirleyen çerçeveleri yaratır. Nasıl algısal bir ilgi alanı algılama sürecini yönlendiriyorsa, egemen kavramsal çerçeve de bilimsel araştırmayı yönlendirir.

## Kültürel Dönüşümler

Son olarak fenomenologların yapay zeka iddiaları karşısında yanılmış olabileceklerini de belirttiklerine değinelim. Ancak bu kabul ilk bakışta göründüğü gibi anlaşılmamalıdır. Şöyle ki, insanın yapısının, egemen kültürel çerçeveler, insanların kendi davranış ve faaliyet tarzları tarafından belirlediğini vurgulayan bilgi-işlem ve yapay zeka çalışmalarının yaygınlaşmasının bu iddiaları *doğru kılabileceği* olasılığı üzerinde duruyorlar. Bunu biraz daha açalım:

Kuhn'un sözünü ettiği kavramsal dönüşümlerden daha da temel bir düzeyde, kültürel dönüşümlerden sözedebiliriz. Bu bölüm boyunca sık sık, örtük olarak da olsa böyle bir kültürel dönüşüm çağına gönderme yaptık: M.Ö. 6-5. yüzyıllarda felsefenin ilk doğuş çağı. Bu dönüşüm, kendisinden sonraki bütün kavramsal dönüşümlere bir zemin oluşturacak bir doğa, insan ve akıl anlayışını yerleştirdi.

Daha sonra Batı'da Hıristiyanlığın doğuşu yeni bir kültürel dönüşümü birlikte getirdi. Örneğin: Hıristiyanlıkla birlikte, daha önce mümkün olmayan bir sevgi türü mümkün oldu, kahramanlık, bir erdem olmaktan çıkıp, kibirle akrabalığı yüzünden, kuşku duyulan bir şey oldu vb.

Bütün bu kültürel dönüşümlerin işaret ettiği, doğa ile kültür arasındaki sınırın sanıldığı kadar katı olmayabileceği, içgüdüsel ihtiyaçların bile dönüşebileceği, insanın sabit bir doğası olmadığı...

Fenomenologlara göre insanın yapısı öylesine değişkendir ki, bir kez daha değişmenin eşiğinde olabilir. Eğer bilgi işlem ve yapay zeka paradigmaları belli bir yaygınlığa ulaşırsa, buraya kadar saydığımız gerekçelerden ötürü makinelerin insana benzeme olasılığı olmadığına göre, insanlar giderek daha çok makinelere benzemeye başlayabilirler. Son iki bin yıldır Batı kültüründe, nesnelliğin önemi, eylemlerimizin sabit ve nicel olarak ifade edilebilir değerlerle belirlendiği, becerilerimizin formelleştirilebileceği ve genel olarak pratik faa-

liyet hakkında kuramsal bilgi sahibi olabileceğimiz inançları yalnızca felsefe, psikoloji ve toplumsal bilimler gibi düşünsel faaliyetleri değil, gündelik hayatı da etkiledi. Bu pratik etki, sayısal bilgisayarların birer alet değil, bir insan modeli olarak yaygınlaşmasıyla derinleşebilir. Bu ise insanların kendilerini giderek daha çok soyut makinelerin, katı hesaplamalar aracılığıyla benzetleyebileceği nesneler olarak görmelerine yolaçabilir. İnsanların ne oldukları ise kendilerini nasıl gördüklerinden bağımsız değildir.



# Beşinci Kuşak Bilgisayarlar

Yapay zeka iddilarına yönelik eleştirileri gözden geçirirken özellikle iki eleştirinin sık sık değişik biçimlerde yinelendiğini gördük. Bunlardan birincisi, insan gövdesinin ve özellikle sinir sisteminin sayısal ilkelere göre çalıştığı iddiasına yöneltilen eleştirilerdi. Bu bağlamda özellikle, bilgisayarların bütün komutları belli ve sabit bir sıra içerisinde yerine getirmeleri ve işlemleri de yine belli bir sıra içersinde yapmalarıyla, sinir sisteminde bir nöronun uyarısının aynı anda binlerce nöronu birden uyarması arasındaki fark üzerinde duruluyordu.

Daha sık karşımıza çıkan ikinci bir eleştiri ise, insan düşüncesinin kendi üzerinde dönüşlü olma özelliğini, mevcut bilgisayarların benzetlemesine olanak olmadığı yolunda idi.

Bugün bilgisayar dünyasının en çok merak uyandıran bilinmezi olan Beşinci Kuşak Bilgisayarlar'ın, her iki tür eleştirinin de işaret ettiği kusurları aşacağı ve gerçek yapay zeka sorununu donanım düzeyinde çözeceği iddia ediliyor.

Kuşkusuz, beşinci kuşak bilgisayarlar tasarısı gibi dev ve pahalı bir tasarıya (bu tasarıyı gerçekleştirme amacıyla sadece Japon hükümetinin, araştırmanın ilk on yılı için tahsis ettiği bütçe, 450 milyon dolar) yapay zeka iddialarına yöneltilmiş felsefi eleştirileri karşılamak üzere girişilmedi. 1990'ların bilgisayarı olarak sunulan beşinci kuşak bilgisayarları destekleyen Japon hükümetine verilen ilk raporda, bu projeye yolaçan gerekçeler ve projenin gerçekleştirmesi beklenen hedefler şöyle özetleniyordu:

**1 /** Üretkenliğin düşük olduğu alanlarda üretkenliği arttırmak.

Sözkonusu belgede bu alanların başında belge işlem, büro yönetimi, işletmelerde karar alma gibi genel olarak hizmetler adıyla anılanve bürolarda yürütülen işler geliyordu. Daha sonra da göreceğimiz gibi, beşinci kuşak bilgisayarlardan bu alanlar da umulan yararlar, yapay zeka iddiaları ile yakından ilgili.

**2 /** Uluslararası rekabette öne geçmek ve uluslararası dayanışmaya katkıda bulunmak.

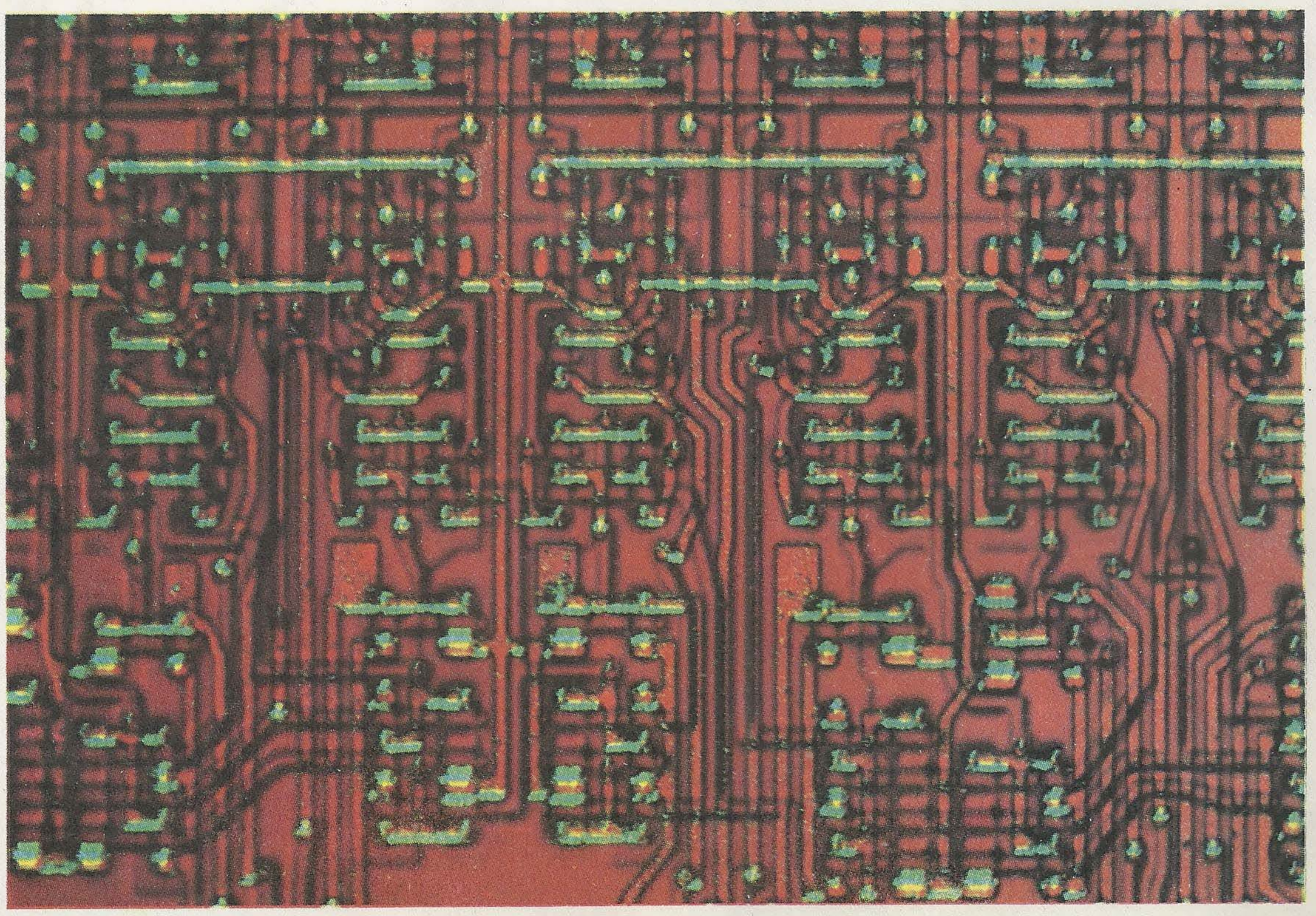
Bu bağlamda Japonya'nın besin ve ener-

ji kaynakları alanında kendine yeterli olamadığı, dolayısıyla öne geçmek için insan malzemesinden yararlanmak durumunda olduğu, bu insan malzemesinin ise 1990'ların dünyasında besin ve enerji ile kıyaslanabilecek bir meta haline dönüşecek olan bilgiyi üretmede kullanılabileceği vurgulanıyordu. Yine aynı kalem altında, bugüne kadar bilgisayar teknolojisi alanında başka ülkeleri (A.B.D.'yi) izlemiş olan Japonya'nın bundan böyle teknolojik gelişmenin seyirini belirlemesi gerekliliği üzerinde duruluyordu.

**3 /** Enerji ve kaynak tasarrufuna katkıda bulunmak.

**4 /** Yaşlanmakta olan bir toplumun sorunlarına çare bulmak.

Tıbbi gelişmeye bağlı olarak çağdaş toplumlarda yaş ortalamasının hızla yükselmekte olduğuna dikkati çeken rapor (1990'da Japonya'nın toplam nüfusunun %12'sinin 65 yaş yada üzerinde olacağı vurgulanıyor), buna bağlı olarak sosyal refah harcamalarının da arttığını iddia ediyor. Yaşlılara verilecek tıbbi, eğitsel ve diğer hizmetlerin bilgisayar tarafından devralınmasının bu harcamaları önemli ölçüde azaltacağı savunuluyor.



# Beşinci Kuşak Bilgisayarların Ticari ve Teknolojik Arka Planı

Beşinci kuşak bilgisayar tasarısının özgül tarihini incelemeye geçmeden önce bilgisayar teknolojisi tarihine ve bu tarihin uluslararası ve ulusal ticaretler arası rekabetlerle ilişkisine kısaca bir göz atmakta yarar var. Çünkü göreceğimiz gibi, beşinci kuşak bilgisayarlar tasarısı, aynı tarihsel ve ticari sürecin bir uzantısı.

## Bilgisayar Teknolojisinin Tarihine Kısa Bir Bakış

Bilgisayar teknolojisinin bir bilimsel araştırma objesi olmaktan çıkıp, ilk kez sanayii ve ticari bir meta halini alması, büyük ölçüde vakum tüpleri yerine transistörlerin kullanılmaya başlamasına bağlı oldu. 1956 yılında Amerikan telefon tröstü ATT, telefon şebekesi üzerindeki tekel hakları uğruna, transistörler üzerindeki patent hakkından vazgeçti. Bu, vakum tüp temelli bilgisayar üreticilerinin sonu oldu.

Transistörlerin, vakum tüplerine kıyasla başlıca avantajları arasında, çok daha az yer kaplamaları, daha az elektrik sarfetmeleri, daha az ısı üretmeleri ve çok daha dayanıklı olmaları sayılabilir. 1956-1962 yılları arasında Shockley, Fairchild ve Texas Instruments gibi firmalar birçok transistörü tek bir yonga üzerinde birleştirerek tümleşik devreler imal etmek için iktisadi yollar geliştirdiler. Aynı yıllarda, IBM, Sperry Univac, Honeywell, Control Data gibi büyük sistem firmaları ise, bilgisayar imalatında çok sayıda tümleşik devreden yararlanmaya başladı. Tümleşik devrelerin ucuzluğu, bilgisayarların çeşitli askeri ve ticari alanlarda yaygın olarak kullanılabilmelerini mümkün kılıyordu.

İlk bakışta askeri siparişlerin sağladığı destekle yarı-iletkenler teknolojisi hızla gelişmeye başladı. (Askeri sipariş ve fonların bilgisayar teknolojisindeki gelişmeler üzerindeki belirleyici etkisi daha sonraki yıllarda da sürdü.) İlk yıllarda gerçekleştirilen ve yaklaşık 10 transistörü bir yonga üzerinde birleştirmeyi mümkün kılan küçük ölçekli tümleştirme (**small scale integration-SSI**) yerini neredeyse 100 transistörü tek bir yonga üzerinde birleştirmeyi mümkün kılan orta ölçekli tümleştirmeye (**medium scale integration-MSI**) bıraktı.

1970'lerde yaşanmaya başlayan kişisel bilgi-

sayar devriminin temelinde ise büyük ölçekli tümleştirme (**large scale integration-LSI**) var. Ancak 1970'lere gelindiğinde, IBM gibi ''büyük sistem'' şirketleri fonlarını, insan kaynaklarını zaten oldukça kârlı olan tanıdık biçimlerin üretimine ayırmış, girişkenlik açısından bakıldığında hantallaşmış durumdaydılar. Bu yüzden LSI'ın olanaklarından ilk önce, en ünlüleri Apple olan bir dizi küçük ve yeni kurulmuş şirket yararlandı. Bu firmaların başarısı ve kişisel bilgisayar üretiminin büyük yatırımlar gerektirmeyişi, büyük sermayenin küçük şirketler karşısında geleneksel olarak alışmış oldukları avantajlara sahip olamayacağı izlenimini yaratıyordu.

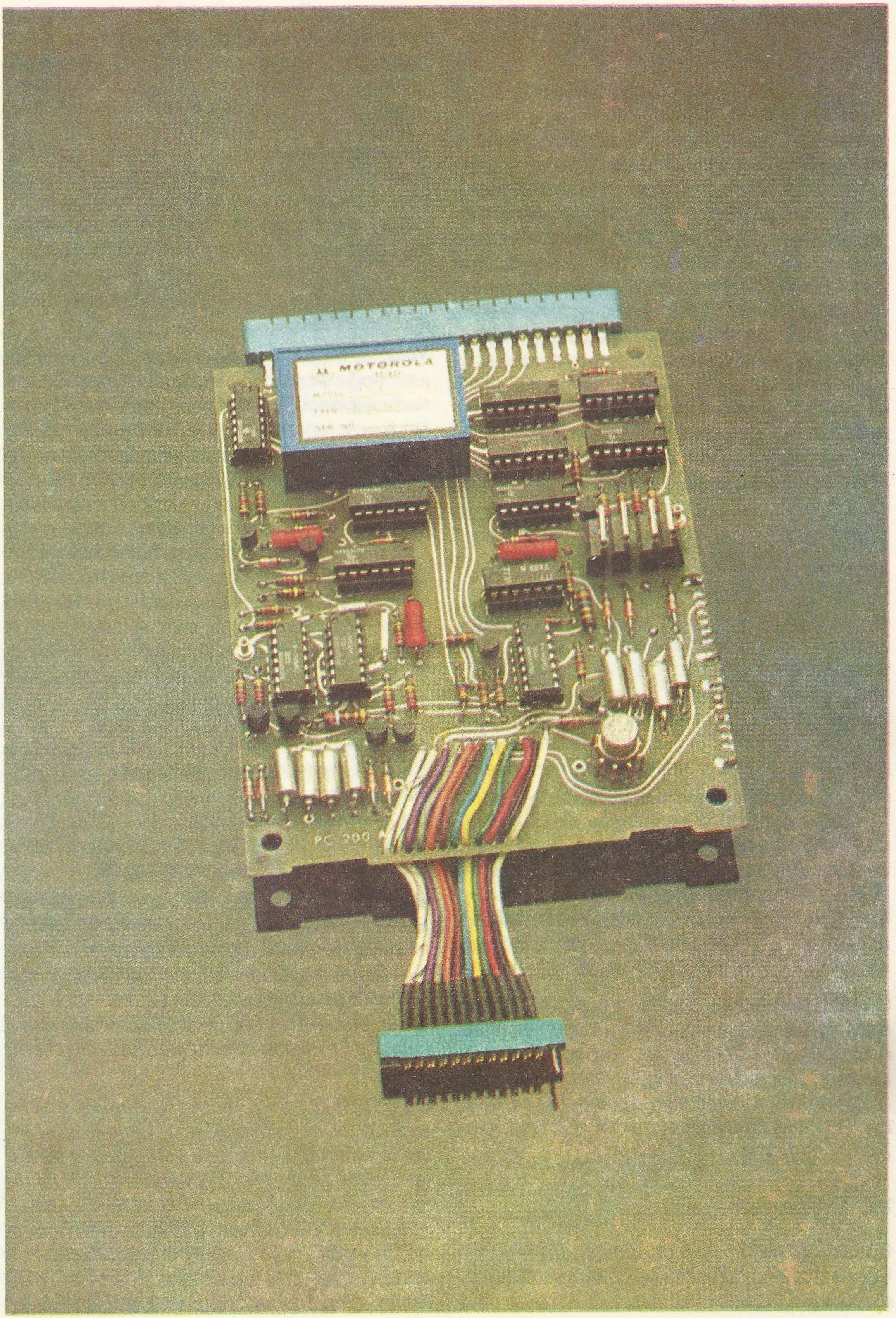
## Destek Sanayiiler

Ancak bilgisayar üretiminde görülen bu şirketler çoğalması ve yaygınlaşması, bilgisayarın yapı taşı olan yarı-iletken yongaları üreten firmalarda görülmüyordu. Aksine 1970'lerin, bu endüstrinin 'istikrar' kazandığı dönem olduğu söylenebilir. Pazarın neredeyse tamamı hepsi de Amerikan olan bir dizi dev firmanın elindeydi. Bu firmaların en önemli müşterileri elbette bilgisayar üreticileri oluyordu. Bunların yanında robotların kontrol birimleri, telekomünikasyon aygıtları ve diğer elektronik gereçler üreten firmalar da müşterilerinin geri kalan kısmını oluşturuyordu.

LSI teknolojisi, bu firmalara siparişe göre üretilmiş, özel amaçlı yongalar üretme olanağını da tanıyordu. Ancak bu firmaların kitlesel üretime (mass production) yönelik yapısı bu olanakları onlar için çekici kılmıyordu. Birkaç tanesi sınırlı sayıda özel amaçlı yonga üretmek için gerekli teknolojik eklemeleri yaptıysa da, kârları yeterli bulmadıklarından, onlar da kısa süre içinde bu uygulamadan vazgeçtiler.

Oysa özel amaçlı ve özel olarak tasarlanmış yongalar, büyük sistem firmaları için çok gerekliydi. Bu firmaların gizli tutmaları istedikleri bilgiler, ancak standart olmayan, özel olarak tasarlanmış bellek yongalarında saklanabilirdi. Bu yüzden büyük sistem firmaları kendi yonga üretim birimlerini oluşturdular ve bu işbölümü 1970'lerin sonuna kadar sürdü: Bir yanda standart bellek yorganları üreten dev firmalar, bir yanda onların ürünlerinden kişisel bilgisayar üretiminde yararlanan bir sürü küçük firma ve kendi özel amaçlı yorganlarını üreten dev ''büyük sistem'' **(mainframe)** firmaları.

Ama 1970'lerin sonundan başlayarak, bir dizi ticari ve teknolojik gelişme, bu işbölümünü bir dar boğaza soktu. Bu gelişmelerin başın-


MOTOROLA
PC 200

# Mecazi İfadelerin Önemi (2)

## Ontolojik Mecazlar

*Yukarı-aşağı, ön-arka, açık-kapalı, merkez-çevre, gibi mekan yönelimleri, kavramları yönelim terimleriyle anlamak için son derece zengin bir temel sağlarlar. Ancak yönelimle yapabileceğimiz yalnızca bu kadardır. Fiziksel nesne ve tözlerle deneyimimiz kavrayış için daha zengin bir temel sağlar -salt yönelimin ötesine geçen bir temel. Nesne ve tözlerle ilişkimiz açısından yaşantımızı kavramak, yaşantımızın belli kısımlarını seçip onları tek tek tek varlık ya da tözler olarak ele alabilmemizi sağlar. Yaşantılarımızı bir kez varlık ya da töz olarak tanımladıktan sonra artık onlara atıfta bulunabilir, kategorileştirebilir, gruplayabilir ve ölçebiliriz- ve bu yollarla onlar üstüne akıl yürütebiliriz.*

*Şeyler açık bir şekilde ayrımlanmış ya da sınırlanmış olmadıkları zaman, onları yine de öyleymişlercesine kategorileştiririz; örneğin dağlar, sokak köşeleri, çitler vs. gibi. Belli amaçlarımız için fiziksel fenomenlere bu şekilde betimlemek kaçınılmazdır: Dağları konumlandırmak, sokak köşelerinde buluşmak, çitleri budamak gibi amaçlarımız için. İnsanın amaçları, fiziksel fenomenleri de tıpkı kendimiz gibi ayrımlandıran yapay sınırlar çizmemizi gerektirir.*

*İnsanın mekansal yönelimlerine ilişkin temel deneyimlerin yönelim mecazlarına yol açışı gibi, fiziksel nesnelerle deneyimlerimiz de (özellikle kendi gövdemizle) son derece çeşitli ontolojik mecazlar için, yani olayları, etkinlikleri, duyguları, fikirleri vs.'yi varlık ya da töz olarak görmek için, bir temel sağlar.*

Ontolojik mecazlar *çeşitili amaçlara hizmet eder ve mevcut mecazların çeşitleri hizmet edilecek amaçları yansıtır. Örneğin mecazi olarak* enflasyon *ismi yoluyla varlık olarak görülebilen yükselen fiyatlar deneyimini ele alalım. Bu, bize deneyime atıfta bulunabilmek için bir yol sağlar:*

### *Varlık Olarak Enflasyon*

Enflasyon *hayat standardımızı düşürüyor. Eğer* enflasyon büyürse, *yaşamımızı sürdüremeyeceğiz.*

Enflasyonla savaşmamız *lazım.*
Enflasyon *bizi köşeye sıkıştırıyor.*
Enflasyonla başa çıkmanın *en iyi yolu mülk edinmek.*

Eflasyon beni hasta ediyor.

*Bu durumlarda enflasyona bir varlık olarak bakmak, ona atıfta bulunmamızı, onu ölçmemizi, belli bir yönünü tanımlamamızı, onu bir neden olarak görmemizi, ona göre davranmamızı hatta onu anladığımızı sanmamızı getirir. Bu türden ontolojik mecazlar, kendi yaşantımızla rasyonel olarak uğraşmaya yeltenmek için gereklidir.*

*Yönelim mecazlarında olduğu gibi, bu ifadelerin çoğu da mecaz olarak gözükmez. Bunun nedenlerinden biri, ontolojik mecazların yönelim mecazları gibi çok kısıtlı bir amaç ranjına hizmet etmesidir - atıfta bulunmak, ölçmek vs. Fiziksel olmayan bir şeye varlık ya da töz olarak bakmak, ona ilişkin yeterince ayrıntılı bir kavrayış geliştirmemizi sağlamaz. Ama ontolojik mecazlar daha da ayrıntılandırılabilirler. Bir ontolojik mecaz olan ZİHİN VARLIKTIR'ın kültürümüzde nasıl inceltildiğine ilişkin iki örnek verebiliriz:*

### *Zihin Bir Makinedir*

*Bu denklemin çözümünü hâlâ üretmeye çalışıyoruz.*

*Bugün* zihnim *resmen* işlemiyor
*Haa, şimdi* şarj ettim
*Bugün zihnim* biraz paslı.
*Sabahtan beri bu problem üzerinde çalışıyoruz ve* istimimiz tükenmeye başladı.

*Bu mecazlar farklı nesne türlerini belirliyorlar. Bize zihnin ne olduğuna ilişkin farklı mecazi modeller vererek zihinsel deneyimin farklı yönleri üzerine odaklanabilmemizi sağlıyorlar. MAKİNE mecazı bize zihnim bir açık-kapalı durumu, etkililik devresi, üretim kapasitesi, enerji kaynağı ve işleme koşulu olduğuna ilişkin bir kavrayış sağlıyor.*

*Bu türden ontolojik mecazlar düşüncemizde öyle doğal ve yaygındırlar ki, genellikle zihinsel fenomenin kendinden verili, doğru tanımları olarak görülürler. Bunların mecazi oldukları olgusu çoğumuzun hiç aklına gelmez. ''Baskı altında çatladı'' gibi cümleleri hemen doğru ya da yanlış olarak görürüz. Aslında bu ifade, çeşitli gazeteciler tarafından Dan White'ın neden silahını alıp San Francisco Bele-*

diye Binası'na giderek Belediye Başkanı George Moscone'u öldürdüğünü açıklamak için kullanılmıştı. Bu türden açıklamalar çoğumuza son derece doğal gelir. Bunun nedeni kültürümüzde **zihin kırılgan bir nesnedir** gibi mecazların zihne ilişkin modellerin bir parçası olmasıdır; bu çoğumuzun düşünmekte ve işlemekte kullandığımız modeldir.

## Kap Mecazları

Bizler, dünyanın geri kalan kısmından derimizin yüzeyiyle ayrılmış ve sınırlanmış fiziksel varlıklarız ve dünyanın geri kalanını dışımızda olarak görürüz. Her birimiz sınırlayıcı birer yüzeyi ve iç-dış yönelimi olan kaplarız. Kendi iç-dış yönelimimizi yüzeylerle sınırlanmış diğer fiziksel nesnelere de yansıtırız. Böylece onları da içi ve dışı olan kaplar olarak görürüz. Odalar ve evler apaçık kaplardır. Bir odadan diğerine geçmek, bir kaptan diğerine geçmektir, yani bir odanın dışına, öteki odanın içine geçmektir. Katı nesnelere bile bu yönelimi veririz, bir kabuğu içinde ne olduğunu görmek üzere kırdığımız zaman yaptığımız gibi. Bu yönelimi doğal çevremize de dayatırız. Ormandaki bir açıklık, sınırlayıcı bir yüzeye sahip gibi gözükür; kendimizi de açıklığın içinde ya da dışında, ormanın içinde ya da ormanın dışında olarak düşünürüz. Ormandaki açıklığın doğal sınır olarak algılayabileceğimiz bir yönü vardır -ağaçların az çok seyreldiği ve açıklığın az çok başladığı çalılık bölge örneğin. Ama bir kabı tanımlamak için bizce yeterli bir doğal fiziksel sınır bulunmadığı zamanlarda bile, bir duvarla, çitle ya da soyut bir çizgi ya da düzlemle -bölgeyi bir içi ve sınırlayıcı yüzeyi olacak şekilde işaretleyen- sınırlar dayatma eğilimindeyizdir. Bölgesellikten daha temel olan pek az insan içgüdüsü vardır. Ve bölgenin böyle tanımlanması, etrafına bir sınır konuşu, bir ölçümlendirme eylemidir. İster insan, isterse kaya, ya da toprak parçası olsun, sınırlandırılmış nesnelerin boyutları vardır. Bu onların içerdikleri töz miktarına göre ölçümlendirilebilmelerini sağlar. Örneğin Kansas sınırlı bir bölge -bir KAP- olduğu için ''Kansas'ta bir sürü toprak var'' diyebiliyoruz.

Tözlerin kendilerine de kap olarak bakılabilir. Örneğin bir küvet dolusu suyu ele alalım. Küvete girince suya girmiş olursunuz. Hem küvet, hem de su kap olarak görülürler, ama farklı türden kaplar. Küvet KAP NESNE, su ise KAP TÖZ'dür.

## Olaylar, Eylemler, Etkinlikler ve Durumlar

Ontolojik mecazları; olayları, eylemleri, etkinlikleri ve durumları anlamada kullanırız. Olay ve eylemler mecazen nesne olarak, etkinlikler töz olarak, durumlar da kap olarak kavramlaştırılırlar. Örneğin yarış, ayrı bir varlık olarak görülebilen bir olaydır. Yarış zaman ve mekanda varolur ve iyi tanımlanmış sınırları vardır. Bu nedenle ona, içinde (nesne olan) katılımcılar, (mecazi nesneler olan) başlangıç ve son gibi olaylar ve (mecazi töz olan) koşma etkinliği içeren bir KAP NESNE olarak görürüz. Bu nedenle yarış hakkında şunları söyleyebiliriz:

Pazar günkü yarışa giriyor musun? (KAP NESNE olarak yarış)
Yarışa gidiyor musun? (NESNE olarak yarış)
Yarışı gördün mü? (NESNE olarak yarış)
Yarışın bitişi çok heyecanlıydı. (Bitiş, KAP NESNE'nin içindeki OLAY NESNE)
Yarışta epey koşu vardı (koşma, KAP'ın içindeki TÖZ).
Sona yaklaşıncaya kadar depar atamadım. (TÖZ olarak depar atmak)
Yarışın yarısında enerjim tükendi. (KAP NESNE olarak yarış)
Şimdi yarış dışı kaldı. (KAP NESNE olarak yarış)

Etkinlikler genellikle mecazen TÖZ, dolayısıyla da KAP olarak görülürler:
Camı silmekteyken, heryere su döktüm.
Jerry cam silme işinin içinden nasıl sıyrıldı?
Cam silmenin dışında, ne iş yaptın?
Ne kadar cam silme işi yaptın?
Cam silme işine nasıl girdin?
Şu anda cam silmeye dalmış durumda.

Böylece etkinlikler, eylemleri ve kendilerini oluşturan diğer etkinlikleri içeren kaplar olarak görülür. Ayrıca enerji, bunlar için gereken malzeme ve onların içinde ya da onlardan çıkan olarak algılanan yan ürünleri için kap olarak görülebilir.

Cam silmeye çok enerji harcadım.
Cam silmekten çok zevk alıyorum.
Cam silme işinde çok zevkli bir yan var.

Çeşitli durum türleri de kap olarak kavramlaştırılabilir. Bu nedenle şöyle örneklere de rastlarız:

*Sıkıntılı durumdan çıktık.*
*Komadan çıkıyor.*
*Gittikçe forma giriyorum.*

*Bir zindelik durumuna* geçti.
*Bunalıma* girdi.
*Final haftasının sonundan beri* girmiş olduğu *katatonik durumdan nihayet çıktı.*

## Mecazi Yapılanmanın Kısmi Niteliği

*Bu ana kadar mecazi tanımlanmış kavramların sistematik özelliğini tarif ettik. Böyle kavramlar bir dizi farklı mecazın terimleriyle anlaşılmaktadır (ör.* ***vakit nakittir, zaman akan bir nesnedir*** *vs.). Kavramların mecazi yapılanması zorunlu olarak kısmidir ve bu kısmi yön dilin söz dağarcığına -''temelsiz olmak'' gibi sabit-biçimi ifadeleri içeren deyimler sözlüğü dahil- yansır. Kavramlar, örneğin* ***teoriler binadır*** *gibi sistematik bir biçimde mecazi olarak yapılandıklarından, başka bir alandaki (BİNA) ifadeleri (inşa etmek, temel) mecazi olarak tanımlanmış alanlardaki (TEORİLER) tekabül eden kavramlardan bahsetmek için kullanabiliriz. Temelin mecazi olarak tanımlanmış alanda (TEORİ) kazandığı anlam, mecazi kavram* ***teoriler binadır****'ın, TEORİ kavramını nasıl yapılandırmakla kullanıldığının ayrıntılarına bağlıdır.*

*TEORİ kavramını yapılandırmakta kullanılan BİNA kavramının kısımları, temel ve dış cephedir. Çatı, iç odalar, merdivenler ve holler, binanın TEORİ kavramının bir parçası olarak kullanılmayan kısımlarıdır. Bu yüzden* ***teoriler binadır*** *mecazının ''kullanılan'' bir kısmı (temel ve dış cephe) ve ''kullanılmayan'' bir kısmı (odalar, merdivenler vs.) vardır.* İnşa etmek *ve* temel *gibi ifadeler böyle bir mecazi kavramın kullanılan kısmının örnekleridir ve teoriler üzerine olağan günlük dilimizi oluştururlar.*
*Peki* ***teoriler binadır*** *gibi bir mecazın ''kullanılmış'' kısmını yansıtan dilsel ifadeler nelerdir? İşte üç örnek:*

*Teorisinin binlerce küçük odası ve uzun, dolambaçlı koridorları var.*
*Oluk gibi ağızlarla çevrelenmiş masif Gotik teorileri yeğler.*
*Karmaşık teorilerin genellikle tesisat sorunu bulunmaktadır.*
*Bu cümleler normal günlük dilin alanının dışına, ''figüratif'' ya da ''imgesel'' dediğimiz kısmına denk düşerler. Bu nedenle gündelik ifadeler (''Bir teori inaş etti'') ve imgesel ifadeler (''Teorisi su oluklarıyla dolu'') aynı genel mecazın* ***(teoriler binadır)*** *örnekleri olabilir.*

*Burada imgesel (ya da gündelik olmayan) mecazın üç farklı alttürünü sayabiliriz:*

***1*** */ Mecazın kullanılan kısmının uzantıları, ör. ''Bu olgular benim teorimin kumu ve harcıdır.'' Burada binanın dış cephesine atıfta bulunulmaktadır, oysa* ***teoriler binadır*** *mecazı kullanılan malzemeleri saymaya dek uzanmaz.*

***2*** */ Gündelik mecazın kullanılmayan kısmından örnekler, ör. ''Teorisinin binlerce küçük odası ve uzun, dolambaçlı koridorları var''.*

***3*** */ Yeni mecazdan, yani normal kavramsal sistemimizin bir kısmını yapılandırmak için değil, bir şey hakkında yeni bir düşünce tarzı getirmek için kullanılan mecazdan örnekler, ör. ''Klasik teoriler, çoğu sürekli birbirleriyle kavga eden birçok çocuğa babalık eden atalardır.''*

*Bu alttürlerin herbiri, normal kavramsal sistemimizi yapılandıran mecazi kavramın* kullanılan *kısmının dışında kalmaktadır.*

*Bu arada genel mecazi kavramları nitelemek için verdiğimiz bütün dilsel ifadelerin figüratif olduğunu da belirtmek isteriz. Örnekler* ***vakit nakittir, teoriler binadır*** *vs.'dir. Bunların hiçbiri gündelik değildir. Bu, bu mecazların yalnızca bir kısmının normal kavramlarımızı yapılandırmakta kullanıldığı olgusuna dayalıdır.*

*Normal kavramlarımızda kullanılmayan parçaları zorunlu olarak içerdikleri için, gündeliğin beldesinin dışına çıkarlar.*

*Şimdiye kadar söz ettiğimiz mecazi ifadelerin her biri (ör. zamanı* gelecek, *teoriye* saldırıyoruz*) tüm bir mecazi kavramlar sisteminin içinde kullanılır -yaşarken ve düşünürken sürekli kullandığımız kavramlar sistemi. Bu ifadeler, tıpkı dildeki diğer deyimler tarafından ve söz dağarcığı gibi gelenekler tarafından sabitlenmiştir. Toplam mecazi sistemlerin parçası olan bu örneklere ek olarak tek başlarına duran ve dil ya da düşüncemizde sistematik olarak kullanılmayan özel durumlara has mecazi ifadeler vardır. Bunlar dağın* etekleri, *soğanın* başı, *masanın* ayağı *gibi iyi bilinen mecazlardır.*

*Bunlar mecazi kavramların ayrık örnekleridir,*

her birinde bir (ya da iki veya üç) kullanılan kısım bulunur. Bu yüzden dağın eteği, **dağ bir kadındır** mecazının tek kullanılan kısmıdır. Normal söylemde dağın bluzundan, çorabından, pabucundan söz etmeyiz; ancak özel bağlamlarda bu kullanılmayan kısımlara dayalı yeni mecazi ifadeler üretmek mümkün olabilir. Aslında **dağ bir kadındır** mecazında öyle bir yan vardır ki, dağcılar dağın sırtlarından ya da dağla baş etmekten, dağı fethetmekten veya dağ tarafından öldürülmekten söz ederler. Dağların canlı olarak gösterilip tepelerinin baş şeklini aldığı çizgi filmler vardır.

Burada önemli olan nokta, kültürümüzde ve dilimizde marijinal olan **dağ bir kadındır** gibi mecazların bulunmasıdır; kullanılan kısımları dilin yalnızca geleneksel olarak sabitlenmiş tek bir ifadesinden oluşabilir ve bu tip mecaz çok az olduğu için hiçbir zaman diğer mecazi kavramlarla sistematik olarak etkileşmez.

Bu onları bizim amaçlarımız için görece önemsiz kılar, yine de tümüyle önemsiz değillerdir çünkü kullanılmamış kısımlarından yeni mecazi ifadeler üretmek, şakalar yapmak vs. için üretilmeye açıktırlar. Ve bizlerin onların kullanılmayan kısımlarını yayma becerimiz, ne kadar marjinal olsalar da, varlıklarını sürdürdüklerinin bir belirtisidir.

Dağın eteği gibi örnekler duruma has, sistematik olmayan ve tecrit edilmiş örneklerdir. Diğer mecazlarla etkileşmezler, kavramsal sistemimizde hiçbir ilginç rol oynamazlar ve bu nedenle yaşamımızın temelini oluşturan mecazlar değillerdir. Sahip oldukları tek yaşam işareti, altkültürlerde yaygınlaştırılabilmeleri ve kullanılmamış kısımlarının (görece önemsiz) yeni mecazlar için temel oluşturmasıdır. Eğer ''ölü'' diye adlandırılmayı hakkeden mecazi ifadeler varsa, işte bu tür mecazlardır; yine de küçük bir yaşam ışığı taşırlar çünkü kısmen **dağ bir kadındır** gibi marijinal mecazi kavramlar açısından anlaşılabilirler.

Bu tecrit edilmiş ve sistematik olmayan örnekleri, tartışmakta olduğumuz sistematik mecazi ifadelerden ayırmak önemlidir. Zaman harcamak, görüşlere saldırmak, yolların ayrılması vs. gibi ifadeler, eylem ve düşüncelerimizi yapılandıran sistematik mecazi kavramların yansılarıdır. En temel anlamda ''canlı''dırlar: kendilerini yaşadığımız mecazlardır. Bunların Türkçe'nin söz dağarcığında geleneksel olarak sabitlenmiş bulunmaları, canlılıklarından hiçbir şey kaybettirmez.

## Kavramsal Sistemimiz Nasıl Temellenmiştir?

Normal kavramsal sistemimizin büyük bölümünün mecazi olarak yapılandığını iddia ediyoruz; yani kavramların çoğu kısmen diğer kavramların terimleriyle anlaşılmaktadır. Bu, kavramsal sistemimizin temellenişi üzerine önemli bir soru doğurmaktadır. Mecazsız olarak, doğrudan anlaşılan hiç kavram var mıdır? Eğer yoksa, bir şeyi anlayabilmemizin yolu nedir?

Doğrudan olarak anlaşılan kavramlara en yakın olanları, YUKARI gibi basit mekansal kavramlardır. Mekansal YUKARI kavramımız, mekansal yaşantımızdan kaynaklanır. Gövdelerimiz vardır ve dik dururuz. Yaptığımız hemen hemen her hareket, yukarı-aşağı yönelimimizi ya değiştiren, ya koruyan, önvarsayan ya da bir şekilde kaale alan bir motor program içerir. Dünyadaki sürekli fiziksel etkinliğimiz, -uyurken bile- fiziksel etkinliğimizle yalnızca ilintili değil, merkezi olarak ilintili bir yukarı-aşağı yönelimi içerir. Motor programlarımızdaki ve günlük fonksiyonlarımızdaki yukarı-aşağı yöneliminin merkeziliği, insana bu yönelim kavramının hiçbir alternatifi olamazmış düşüncesini getirmektedir. Ancak nesnel konuşacak olursak, kendi içlerinde yukarı-aşağı yönelimi olmayan Kartezyen koordinatlar dahil, mekansal yönelim için birçok olası çerçeve mevcuttur. Ancak insanın mekansal kavramları, YUKARI-AŞAĞI, ÖN-ARKA, İÇERİ-DIŞARI, YAKIN-UZAK vs.yi içermektedir. Günlük sürekli gövde fonksiyonlarımızla ilintili olanlar bunlardır ve bu nedenle bizim için diğer tüm olası mekan yapılanmaları karşısında öncelikleri vardır. Diğer bir deyişle mekansal kavramlarımızın yapısı, sürekli mekansal yaşantımızla, yani fiziksel çevremizle etkileşimimizle temellenmektedir. Bu şekilde ortaya çıkan kavramlar en temel şekilde kendilerine dayalı yaşadığımız kavramlardır.

Bu nedenle YUKARI salt kendi terimleriyle anlaşılmaz; içinde yaşadığımız yer çekimi alanıyla bağlantılı olan dik pozisyonumuzla ilintili sürekli yapılan motor fonksiyonlarımızın toplanmasından kaynaklanır. Her tür yerçekimi alanının dışında yaşayan, başka her tür deneyimden habersiz olan küresel bir varlık tahayyül edin. Böyle bir varlık için YUKARI'nın ne anlamı olabilir? Bu sorunun cevabı yalnızca bu

*küresel varlığın fizyolojisine değil, kültürüne de bağlı olacaktır.*

*Diğer bir deyişle, ''doğrudan fiziksel yaşantı'' diye adlandırdığımız şey hiçbir zaman sadece belli türden bir gövdeye sahip olma sorunu değildir; daha ziyade kültürel önvarsayımlar temelinde yer alan her türden* yaşantıya *dayanır. Bu nedenle doğrudan fiziksel yaşantıdan, daha sonra kendi kavramsal sistemimizin terimleriyle ''yorumladığımız'' anlık bir yaşantı çekirdeğiymişcesine söz etmek yanlış yerlere götürebilir bizi. Kültürel varsayımlar, değerler ve tavırlar, üzerine canımız istedikçe deneyleri yerleştirip yerleştirmemeyi seçebileceğimiz kavramsal bir örtü değildir. Tüm yaşantının tepeden tırnağa kültürel olduğunu yinelemek daha doğru olur; ''dünya''mızı öyle yaşarız ki kültürümüz yaşantının ta kendisinde zaten mevcuttur.*

*Ancak her yaşantının kültürel önvarsayımlar içerdiğini teslim etsek bile; ayakta durmak gibi ''daha'' fiziksel yaşantılarla, düğün törenine katılmak gibi ''daha'' kültürel yaşantılar arasında önemli bir ayırım yapabiliriz. ''Fiziksel''e karşı ''kültürel'' yaşantıdan söz ederken terimleri bu anlamda kullanıyoruz.*

*Gövdemizin fonksiyonuyla ilgili bazı merkezi kavramlar -YUKARI-AŞAĞI, İÇERİ-DIŞARI, ÖN-ARKA, AYDINLIK-KARANLIK, SICAK-SOĞUK, ERKEK-DİŞİ vs. diğerlerinden daha keskin olarak resmedilmişlerdir. Duygusal yaşantımız mekansal ya da algısal yaşantımız kadar temel olmakla birlikte, duygusal yaşantılarımız gövdemizle ne yaptığımız açısından çok daha az keskin olarak resmedilmiş bir kavramsal yapıya kaynaklık eder. Başka bir deyişle, yalnızca duygusal işlevlerimizden hareketle keskin tanımlı bir kavramsal yapı doğmaz. Duygularımızla (örneğin mutluluk) algısal-motor yaşantımız (dik durmak) arasında* sistematik korelasyonlar *bulunduğu için, bunlar mecazi mekan kavramlarının* ***(mutluluk yükselir)*** *temilini oluştururlar. Böyle mecazlar bizim duygularımızı daha keskin tanımlanmış terimlerle kavramlaştırmamızı ve genel durumlara ilişkin kavramlarla (ör. SAĞLIK, YAŞAM, DENETİM vs.) ilintilendirmemizi sağlar. Bu anlamda zuhur eden mecazlar ve zuhur eden kavramlardan dem vurabiliriz.*

*Örneğin NESNE, TÖZ ve KAP gibi kavramlar doğrudan zuhur ederler. Kendimizi dünyanın geri kalanından ayrı varlıklar olarak -içi ve dışı olan kaplar halinde- algılarız. Kendi dışımızdaki şeyleri de varlıklar olarak görürüz -genellikle içi ve dışı- bulunan kaplar olarak. Kendimizi tözlerden -örn. et ve kemik-, dış nesneleri de çeşitli tözlerden -tahta, taş, metal vs.- oluşmuş olarak algılarız. Görme ve dokunma yoluyla birçok şeyin kesin sınırları olduğunu algılarız; bunların kesin sınırları bulunmadığında da onlara sınırlar dayatırız -onları varlıklar ve genellikle kaplar olarak (örn. ormanlar, açıklıklar, bulutlar vs.) kavramlaştırırız.*

*Mekan mecazları gibi temel ontolojik mecazlar da* yaşantımızla sistematik korelasyonlar *yoluyla temellenirler. Örneğin görmüş olduğumuz gibi* ***görsel alan bir kaptır*** *mecazı, gördüğümüz şeyle, sınırlı fiziksel mekan arasındaki korelasyona dayalıdır.* ***Zaman hareket eden nesnedir*** *mecazı, bize doğru hareket eden bir nesneyle, onun bize ulaşması için geçen zaman arasındaki korelasyona dayanır. Aynı korelasyon,* ***zaman bir kaptır*** *mecazının da (''Beş dakika içinde yaptı''da olduğu gibi) temeldir; nesnenin katettiği sınırlı mekanla, bunu katetmesi için gereken zaman arasında korelasyon yapılmaktadır. Olaylar ve eylemler sınırlı zaman dilimleriyle bağlantılandırılırlar, bu da onları KAP NESNE'ler haline getirir.*

*Belki temellendirmeye ilişkin vurgulanması gereken en önemli nokta, yaşantıyla bizim onu kavramlaştırma biçimimiz arasındaki ayrımdır.*

*Fiziksel yaşantının, duygusal, zihinsel, kültürel vs. ne türden olursa olsun, diğer yaşantılardan daha önemli olduğunu öne sürmüyoruz. Tüm bu yaşantıların hepsi de fiziksel yaşantı kadar temel olabilir. Ancak bizim temellendirmeye ilişkin öne sürdüğümüz, tipik olarak fiziksel olmayanı fizikselin terimleriyle kavramlaştırdığımızdır -yani daha az keskin resmedilmiş olanı, daha net resmedilenin terimleriyle kavramlaştırıyoruz. Şu örneğe bakalım:*

*Ahmet mutfakta.*
*Ahmet YeniGündem'de.*
*Ahmet sevdada düşmüş.*

*Cümleler yaşantının üç farklı alanına atıfta bulunuyor: Mekansal, toplumsal, duygusal.*

*Bunlardan hiçbirinin diğeri üzerinde yaşantısal öncelliği yoktur; hepsi de yaşantının eşit derecede temel türleridir.*

*Ancak kavramsal yapılanma açısından bir fark*

*vardır. İlk cümledeki ismin -de hali ekinin anlamı, doğrudan açıkça resmedilmiş, bir tarzda mekansal yaşantıdan kaynaklanır. Bu bir mecazi kavram örneği değildir. Oysa diğer iki cümle mecazi kavramlara örnektir. İkincisi, toplumsal grup kavramının terimleriyle yapılandığı* ***toplumsal gruplar birer kaptır*** *mecazının örneğidir. Bu mecaz, bizim mekanlandırma yoluyla toplumsal grup kavramına ''hakim olabilmemizi'' sağlar. ''İçinde'' sözcüğü ve ismin aldığı -de hali ile İÇİNDE kavramı, her üç örnekte de aynıdır; bunlar için üç ayrı kavram ya da eşsesli sözcük mevcut değildir. Zuhur etmiş bir içinde (-de) kavramımız, bunun için çeşitli sözcüklerimiz ve toplumsal gruplarla duygusal durumları kısmen tanımlayan iki mecazi kavramımız vardır. Bu vakaların gösterdiği, eşit temellikte yaşantıların her zaman eşit derecede temel olmayan kavramlaştırmalara yol açmalarının mümkün olduğudur.*

## Yapısal Mecazların Temellenmesi

*Kavramsal sistemimizdeki herhangi bir şey kadar temel olan ve onlar olmaksızın dünyada hiçbir işlevi yürütemeyeceğimiz basit fiziksel kavramlara dayalı mecazlar, kendi içlerinde fazla zengin değildirler. Bir şeyin içi ve dışı olan bir KAP NESNE olarak görüldüğünü söylemek, onun hakkında fazla bir şey söylemek değildir. Ancak* ***zihin bir makinedir*** *mecazında ve çeşitli kişileştirme mecazlarında gördüğümüz gibi, mekan mecazlarını çok daha spesifik anlamda ayrıntılandırabiliriz. Bu yalnızca bizim ZİHİN gibi, bir kavramı önemli ayrıntılarına kadar inceltmemizi sağlamakla kalmaz, bu kavramın bazı yönlerini ortaya çıkartıp, bazı yönlerini de gizlememize olanak verir.* ***Rasyonel tartışma savaştır*** *gibi yapısal mecazlar böyle bir ayrıntılandırma için en zengin kaynaklardır. Yapısal mecazlar, basit yönetimsel ve ontolojik mecazlar gibi kavramları yönlendirmemizi, onlara atıfta bulunmamızı, ölçmemizi ve sağlamanın yanısıra, son derece yapılanmış ve açık olarak resmedilmiş bir kavramı bir diğerini yapılandırmakta kullanmamızı da mümkün kılar.*

*Yönelimsel ve ontolojik mecazlar gibi yapısal mecazlar da yaşantımız içindeki sistematik korelasyonlardan temellenirler. Bunun ayrıntılı olarak ne anlama geldiğini görmek için* ***rasyonel tartışma savaştır*** *mecazının rasyonel tartışmayı daha kolay anladığımız bir şeyin, yani fiziksel çatışmanın terimleriyle kavramlaştırabilmemizi sağlar. Hayvanlar aleminde kavga her yerde görülür; en çok da insanlarda. Hayvanlar istedikleri şeyi elde etmek için kavga ederler -yemek, cinsellik, yaşama alanı, denetim, vs.- çünkü aynı şeyleri isteyen ya da onların isteklerini elde etmelerini engellemek isteyen başka hayvanlar vardır. Aynı şey insanlar için de geçerlidir; ancak bizler bunun için daha incelmiş teknikler geliştirmişizdir. ''Rasyonel hayvanlar'' olarak kavgamızı çeşitli biçimlerde kurumsallaştırdık, bu biçimlerden biri de savaştır. Çağlar boyunca fiziksel çatışmayı kurumsallaştırmış olmamıza ve en iyi beyinlerimizden bazılarını çatışmayı yürütmek için etkin yollar bulmada kullanmamıza rağmen, temel yapısı özde değişmemiştir.*

*İki vahşi hayvan arasındaki kavgada bilim adamları, karşısındakinin gözünü korkutmak için meydan okuma, yaşama alanı oluşturma ve savunma, karşı saldırı, geri çekilme ve teslim olma davranışlarını gözlemlemişlerdir. İnsan kavgalarında da aynı davranışlar içerilmektedir.*

*Ancak rasyonel bir hayvan olmanın bir anlamı da, kendini gerçek fiziksel çatışma tehlikesine sokmadan istediğini elde etmektir. Sonuç olarak biz insanlar, sözel tartışma denen toplumsal kurumugeliştirmişizdir. İstediklerimizi elde etmek için tartışır dururuz ve bunlar bazen fiziksel şiddet düzeyine ''iner''. Böyle sözel çarpışmalar hemen hemen fiziksel çarpışmalarla aynı terimlerle kavranır. Örneğin bir karı-koca kavgasını ele alalım. Karıyla kocanın her biri, kendi istediğini elde etmeye çalışıyor; örneğin karşısındakini bir bakış açısına ikna etmek, ya da en azından bir bakış açısına göre davranmaya ikna etmek gibi. Her ikisi de yitirecek ve kazanacak şeyleri, oluşturacak ve savunacak yaşama alanları bulunduğunu düşünüyor. Engelsiz bir tartışmada elinizde bulunan tüm sözel yolları -gözünü korkutmak, tehdit etmek, otorite kurmak, hakaret etmek, aşağılamak, otoritesine meydan okumak, meselelerden kaçmak, pazarlık etmek, pohpohlamak, hatta ''rasyonel nedenler'' vermek- kullanarak saldırır, savunur, karşı saldırıya geçersiniz. Ama bu taktiklerin çoğu gerekçeler olarak sunulabilir ve genellikle sunulur da; örneğin:*

... Çünkü senden daha büyüyüğüm (gözünü korkutmak)
... Çünkü yapmazsan, ben de... (tehdit)
... Çünkü patron benim (otorite)
... Çünkü aptalsın (hakaret).
... Çünkü genellikle yanlış yapıyorsun (aşağılama)
... Çünkü benim de en az senin kadar hakkım var (otoriteye meydan okuma).
... Çünkü seni seviyorum (meseleden kaçınma).
... Çünkü yaparsan... Ben de... (pazarlık)
... Çünkü sen çok daha iyi beceriyorsun (pohpohlama)

Bu türden taktikler kullanan tartışmalar, kültürümüzde en yaygın olanlarıdır; gündelik hayatımızın çok fazla parçası oldukları için de bazen onları farketmeyiz bile. Ancak kültürümüzde böyle taktiklerin en azından ilkesel olarak ''mantık dışı'' ve ''haksız'' bulunarak aşağılandığı önemli ve güçlü kesimler vardır. Akademik dünya, hukuk dünyası, diplomasi dünyası, din dünyası ve basın dünyası, tüm bu taktiklerin yasak olduğu ideal ya da ''daha yüksek'' bir RASYONEL TARTIŞMA biçimi kullandıklarını iddia ederler. Bu RASYONEL TARTIŞMA'da izin verilen tek taktik, sözde öncüllerin konulması, destekleyici kanıtların anılması, mantıksal sonuçlar çıkarsanmasıdır. Ama tüm bu koşulların geçerli olduğu en ideal durumlarda bile RASYONEL TARTIŞMA yine savaş terimleriyle kavranır ve yürütülür. Yine kurulması ve savunulması gereken bir görüş alanı vardır, kazanıp kaybedersiniz, görüş alanına saldırdığınız, yıkmaya çalıştığınız, tartışmasını darma duman etmeye çalıştığınız bir karşıtınız mevcuttur. Eğer tümüyle başarıya ulaşırsanız, onu ezip geçersiniz.

Burada önemli olan nokta yalnızca bizim tartışma kavramımız değil, onu yürütme biçimimizin de fiziksel çarpışmaya ilişkin bilgi ve deneyimimizden temellenmesidir. Hayatınız boyunca savaş bir yana hiç yumruk dövüşü bile yapmamış da olsanız, eğer konuşmaya başladığınız zamandan beni tartışmaktaysanız, yine tartışmaları algılayışınız ve yürütüşünüz TARTIŞMA SAVAŞTIR mecazına göredir, çünkü mecaz içinde yaşadığınız kültürün kavramsal sisteminin içine inşa edilmiştir. RASYONEL TARTIŞMA idealini yerine getirdiği varsayılan tüm ''rasyonel'' tartışmaların SAVAŞ terimleriyle algılanması bir yana, hemen hepsi de ideal biçimdeki rasyonel tartışmaların aştığı varsayılan ''mantık dışı'' ve ''haksız'' taktikleri gizli biçimde içerirler. İşte birkaç tipik örnek:

... Şunu kabul etmek akla uygundur (gözünü korkutma)
Açıkça,
Kuşkusuz...
... Bilimsel olmazdı... (tehdit)
Bunu söylemek... Safsatadır...
... Descartes'ın gösterdiği gibi (otorite)
Hume gözlemlemiştir ki...
Dipnot, 374: Alıntı Verschlugenheimer, 1954.

Bu çalışma gerekli ihtisası ne yazık ki... Böyle bir teoriye ''Dar'' Rasyonalizm adını verebiliriz. ''Akademik nesnellik'' kisvesiyle... (hakaret)

Bu çalışma şekillenmiş bir teori üretemeyecek sonuçları ölçülebilir değil. (aşağılama)

Günümüzde pek az insan bu görüşü ciddi ciddi savunmaktadır.

Pozitivist yaklaşımların hatasına düşmediğimiz sürece (otoriteye meydan okuma)

Davranışçılık şuna yol açmıştır...

Alternatif bir teori sunmuyor (Meseleden kaçınmak)
Ama bu şey sorunu...
Yazar bazı ilginç olgular sunuyor, ancak...

Görüşün şu noktaya kadar doğru (pazarlık)...
Eğer gerçekçi bir bakış açısından ele alırsak, şu iddiayı kabul edebiliriz...

Heyecan verici makalesinde (poppohlamak)
Makalesi bazı ilginç konulara parmak basıyor...

Böyle örnekler, rasyonel tartışmadan ''akılcı olmayan'' tartışmaya (gündelik tartışma) oradan da fiziksel çarpışmadaki köklerine varmamızı sağlar. Göz korkutma, tehdit, otoriteye başvurma vs. taktikler, rasyonel tartışmalarda da belki daha rafine cümlelere saklanmış da olsalar, tıpkı gündelik tartışma ve savaştaki gibi mevcuttur. İster bilimsel, akademik, hukuksal bir ortamda rasyonel tartışmanın idealine ulaşıyor olalım, ister evimizde didişerek istediğimizi yaptırmaya çalışalım, tartışmalarımızı algılama, yürütme ve tanımlama biçiminiz TARTIŞMA SAVAŞTIR mecazında temellenmiştir.

Şimdi hayatımızda önem taşıyan diğer yapısal mecazlara bakalım: EMEK KAYNAKTIR ve ZAMAN KAYNAKTIR. Bu mecazların her ikisi de kültürel olarak maddi kaynaklarla ilintili yaşantımızdan temellenmiştir. Maddi kaynaklar tipik olarak hammadde ya da yakıt kaynaklarıdır. Her ikisinin de yararlı amaçlara hizmet ettiği düşünülür. Yakıt ısınma, taşıma ya da bitmiş bir ürünü üretmekte kullanılan enerji olarak değerlendirilir. Hammaddeler tipik olarak doğrudan ürünlere giderler. Her iki durumda da maddi kaynaklar ölçülebilir ve bunlara bir değer verilebilir. Her iki durumda da amaca ulaşmak için önemli olan malzemenin özel bir parçası ya da miktarı değil, türüdür.

Örneğin, uygun kömür türünü kullandığınız sürece, evinizi ısıtmak için hangi kömür parçasını kullandığınız önemli değildir. Her iki durumda da, amaca hizmet edildikçe malzemeler yavaş yavaş tüketilir. Özetlersek:

Maddi kaynak bir tür tözdür. Hemen hemen kesin olarak ölçülebilir. Birim miktarına bir değer verilebilir. Yararlı bir amaca hizmet eder, amaca hizmet ettikçe yavaş yavaş tüketi'lir.

Hammaddeden bir mamul ürettiğiniz basit bir durumu ele alalım. Bu, belli miktarda emek gerektirir. Genelde, ne kadar emek verirseniz, o kadar üretirsiniz. Bunun doğru olduğunu -emekle ürün miktarının aynı oranda arttığını- varsayarak emeğe, bir ürün birimi üretmek için gereken zamanın terimleriyle bir değer verebiliriz. Bunun en mükemmel modeli, hammaddedin bir uçtan geldiği, emeğin adım adım gelişen süreçler halinde uygulandığı, süresinin bant tarafından belirlendiği ve mamülün öbür uçtan çıktığı seri üretimdir. Bu, **emek KAYNAKTIR** mecazı için şu şekilde temel sağlar.

EMEK bir tür etkinliktir (hatırlayın: ETKİNLİK TÖZDÜR). Yaklaşık bir kesinlikle ölçülebilir (zaman açısından). Birim başı değer verilebilir. Yararlı bir amaca hizmet eder, amaca hizmet ettikçe yavaş yavaş tüketilir.

Emek zaman açısından ölçülebildiği ve sanayi toplumunda genellikle de ölçüldüğü için, **zaman kaynaktır** mecazının temelini bulunuz.

ZAMAN bir tür (soyut) TÖZDÜR. Yaklaşık bir kesinlikle ölçülebilir. Birim başı değer verilebilir. Yararlı bir amaca hizmet eder, amaca hizmet ettikçe yavaş yavaş tüketilir.

Kültürümüzde olduğu gibi **emek kaynaktır** ve **zaman kaynaktır** mecazlarına uygun yaşandığında, bunları hiç de mecaz olarak görme eğiliminde olmayız. Ancak bunların yaşantıdaki temellendirilmesinin de gösterdiği gibi, her ikisi de Batı sanayi toplumlarında temel bulan yapısal mecazlardır.

Her iki karmaşık yapısal mecaz da basit ontolojik mecazlar kullanılır. **Emek kaynaktır etkinlik tözdür**'ü kullanır. **Zaman kaynaktır, zaman tözdür**'ü kullanır. Bu iki TÖZ mecazı emek ve zamana nicelik verilebilmesini sağlar -ölçülmelerini, yavaş yavaş ''tüketilmekte'' olarak algılanmalarını ve parasal değerler atfedilmesini. Ayrıca zaman ve emeği çeşitli amaçlar için ''kullanılabilen'' şeyler olarak görmemizi sağlarlar.

**Emek kaynaktır** ve **zaman kaynaktır** hiçbir alanda evrensel değillerdir. Kültürümüzde doğal olarak zuhur etmelerinin nedeni bizim işe bakış tarzımız, nicelik atfetme tutkumuz ve yararlı amaçlara olan saplantımızdır. Bu mecazlar emek ve zamanın kültürümüzde merkezi önem taşıyan yönlerini vurgularlar. Bunu yaparken aynı zamanda emek ve zamanının bazı yönlerini de hafifletir ya da gizlerler.

Her iki mecazın da neyi gizlediğini anlamak için neye odaklandıklarına bakabiliriz.

Emeği bir tür etkinlik olarak değerlendirirken, mecaz emeğin açıkça tanımlanabileceği ve emek olmayan şeylerden ayrılabileceğini varsayar. İşi oyundan, üretici etkinliği üretici olmayandan ayırt edebileceğimizi varsayar. Kuşkusuz bu varsayımlar çoğu zaman gerçeğe uymazlar; belki seri üretim vs. durumları hariç. Emeği, kimin ürettiğinden, nasıl yaşadığından, yaşamında ne anlamı olduğundan bağımsız olarak bir tür etkinlik olarak görmek; işin kişisel olarak anlamlı, doyurucu ve insani olup olmadığı meselelerini gizlemektedir.

Emeğin zaman açısından ölçülmesi, zamanın yararlı bir amaca hizmet ediyor olarak görülmesiyle birlikte, İŞ ZAMANI kavramına paralel bir BOŞ ZAMAN nosyonu doğrur. Bizimki gibi etkinsizliğin yararlı bir amaç sayılmadığı toplumlarda, boş zaman etkinliklerine adanmış koca bir endüstri gelişmiştir. Bunun sonucunda BOŞ ZAMAN da bir KAYNAK olur -üretici olarak kullanılacak, akıllı kullanılacak, tasarruf edilecek, hesaplanacak, harcanacak,

*kaybedilecek vs. Emek ve zamanın KAYNAK mecazlarıyla gizlenen, EMEK ve ZAMAN kavramlarının BOŞ ZAMAN kavramımızı etkileyerek onu hayret verici ölçüde EMEK'e benzer bir şeye döndürme tarzıdır.*

*Emek ve zamanın KAYNAK mecazları diğer kültürlerde ve kendi kültürümüzün bazı altkültürlerinde varolan her türden farklı emek ve zaman kavramlarını gizler: İşin oyun olabileceği, hareketsizliğin üretici olabileceği, EMEK olarak sınıflandırdıklarımızın çoğunun yararı hiç de açık olmayan ya da değersiz amaçlara hizmet ettiği fikirlerini gizler.*

*Bu bölümde ele aldığımız üç yapısal mecazın -RASYONEL TARTIŞMA SAVAŞTIR, EMEK KAYNAKTIR ve ZAMAN KAYNAKTIR- hepsinin de güçlü kültürel temelleri vardır. Bizimki gibi bir kültürde doğal olarak doğarlar, çünkü vurguladıkları bizim toplu olarak yaşadıklarımıza çok iyi tekabül ederken, gizledikleri pek az şeye tekabül eder. Ancak bunlar fiziksel ve kültürel yaşantılarımızda temellenmekle kalmaz, yaşantı ve eylemimizi de etkilerler.*

## Öznellik ve Nesnellik Mitleri

*Doğrunun kavrayışa dayalı oluşu üzerine bir değerlendirmede bulunduk. Doğrunun her zaman bir kavramsal sistemle bağlantılı olduğunu, insanın kavramsal sisteminin ağırlıklı olarak mecazi nitelikte olduğunu, bu nedenle tümüyle nesnel, koşulsuz ya da mutlak hiçbir doğru bulunmadığını öne sürdük. Bilim kültüründe ya da mutlak doğrunun verili kabul edildiği diğer altkültürlerde yetişen insanlar için bir öznelliğe ve gelişigüzelliğe -bir şeyin ''ne istiyorsam o olması, ne eksik, ne fazla'' anlamına geldiğine ilişkin nosyona- teslim olmak gibi gelecektir. Aynı nedenle romantik gelenekle özdeşleşenler de nesnellik üzerine herhangi bir zaferi imgelemin bilim üzerine zaferi olarak görebilirler - her bireyin kendi gerçekliğini her türlü kısıttan uzak olarak yarattığı görüşünün zaferi olarak.*

*Bu görüşlerin her ikisi de, nesnelliğin tek alternatifinin radikal öznellik olduğuna -yani ya mutlak doğruya inanmanız gerektiğine* ya da *dünyanızı kendi imgeleminizle yaratabileceğinize ilişkin yanlış kültürel varsayıma dayalı birer yanlış anlama olacaklardır. Eğer* nesnel *değilseniz,* öznel *olursunuz; üçüncü bir seçenek yoktur. Biz, nesnellik ve öznellik mitlerine üçüncü bir seçenek getirdiğimizi düşünüyoruz.*

*''Mit'' terimini küçültücü bir anlamda kullanmıyoruz. Mitler yaşantıyı kavramak için yol sağlarlar, yaşamımıza düzen verirler. Mecazlar gibi mitler de etrafımızda ne olup bittiğini anlamlandırmak için gereklidirler. Bütün kültürlerin mitleri vardır ve insanlar tıpkı mecazsız yapamadıkları gibi mitsiz de işlev gösteremezler. Ve tıpkı kendi kültürümüzdeki* mecazları *doğru olarak gördüğümüz gibi, kendi kültürümüzdeki* mitleri *de doğru olarak görürüz. Nesnellik miti de bu anlamda kısmen sinsidir. Yalnızca mit değilmiş gibi gözükmekle kalmaz, gerek mitleri, gerekse mecazları aşağılanacak ve küçümsenecek nesneler haline getirir: nesnelci mite göre, mitler ve mecazlar ciddiye alınamazlar çünkü nesnel olarak doğru değildirler. Göreceğimiz gibi nesnellik mitinin kendisi nesnel olarak doğru değildir. Ama bu onu aşağılanacak ya da küçümsenecek bir şey kılmaz. Nesnellik miti bu kültürdeki her bireyin günlük işlevlerinin bir parçasıdır. İncelenmesi ve anlaşılması gerekmektedir. Biz ayrıca tamamlanmasının da gerekli olduğuna inanıyoruz - karşıtı olan öznellik miti tarafından değil, bizim yaşantımızın gerçeklerine daha iyi uyduğuna inandığımız yeni bir deneysel mit tarafından tamamlanmalı. Deneysel bir seçeneğin nasıl olacağını netleştirmek için önce öznellik ve nesnellik mitlerini ayrıntılı olarak incelemeliyiz.*

## Nesnellik Miti

*Nesnellik miti şöyle der:*

1. *Dünya nesnelerden oluşmuştur. Nesneler, kendileri ile etkileşen insan ve diğer varlıklardan bağımsız özelliklere sahiptir. Örneğin bir kayayı ele alalım. Ayrı, bir nesnedir ve serttir. Evrende hiçbir insan ya da varlık varolmasa da, yine ayrı bir nesne olacak ve yine sert olacaktır.*

2. *Dünyaya ilişkin bilgimizi içindeki nesnelerle ilişkiye girerek ve bu nesnelerin ne olduğunu anlamaya çalışarak ediniriz. Örneğin bir kayanın ayrı bir nesne olduğunu ona bakarak, dokunarak, etrafında dolaşarak vs. anlarız. Sert olduğun elleyerek, sıkmaya çalışarak, tekmeleyerek, daha yumuşak bir şeye çarparak vs. anlarız.*

*3. Dünyamızdaki nesneleri kategori ve kavramların terimleriyle anlarız. Bu kategori ve kavramlar, nesnelerin kendi içlerinde (içsel olarak) sahip oldukları niteliklere ve nesneler arasındaki ilişkilere tekabül ederler. Böylece KAYA kavramına tekabül eden bir ''kaya'' sözcüğümüz vardır. Bir kaya verili olduğunda, onun KAYA kategorisinde olduğunu, piyano, ağaç ya da kaplanın ve bu kategoriye girmeyeceğini biliriz. Kayaların her tür varlıktan bağımsız içsel özellikleri vardır kat, sert, yoğundurlar; doğada oluşurlar vs. Bir ''kaya''nın ne olduğunu bu özellikler yardımıyla anlarız.*

*4. Nesnel bir gerçeklik vardır ve onlar hakkındaki şeylerin nesnel, mutlak ve koşulsuz olarak doğru ya da yanlış olduklarını söyleyebiliriz. Ancak insan olarak bizler insanî hatalar yaparız, yani yanılsamalar, algı hataları, yargı hataları, duygular ve kişisel ve kültürel önyargılara sahibizdir. Tek tek insanların öznel yargılarına güvenemeyiz. bilim bize öznel sınırlarımızın üzerine yükselmemizi ve evrensel olarak geçerli ve önyargısız bir bakış açısından kavrayış kazanmamızı sağlayan bir metodoloji verir. Bilim nihai olarak gerçekliğin doğru tanımsal ve genel bir değerlerdirmesini verir ve metodolojisi yoluyla sürekli o amaca doğru ilerler.*

*5. Sözcüklerin sabit anlamları vardır. Yani dilimiz terimleriyle düşündüğümüz kavram ve kategorileri ifade eder. Gerçekliği doğru olarak tarif edebilmek için, anlamları kesin ve net olan, gerçekliğe uyan sözcüklere ihtiyacımız vardır. Bunlar doğal olarak gelişen sözcükler ya da bilimsel teorideki teknik terimler olabilir.*

*6. İnsanlar nesnel olabilir ve nesnel konuşabilirler ancak bu açık ve kesin olarak tanımlanmış, doğrudan ve düz olan, gerçekliğe uyan bir dil kullanırlarsa mümkündür. İnsanlar ancak bu şekilde konuşarak dış dünya üzerine kesin olarak iletişebilirler ve nesnel olarak doğru ya da yanlış diye yargılanabilen cümleler kurarlar.*

*7. Mecaz ve diğer şiirsel, hayali, retorik ya da figüratif dil türlerinden nesnel olarak konuşurken kaçınılabilir ve kaçınılmalıdır da; çünkü anlamları açık ve kesin değildir ve gerçekliğe herhangi bir apaçıklıkla uymazlar.*

*8. Nesnel olmak genelde iyi bir şeydir. Yalnızca nesnel bilgi gerçek bilgidir. Yalnızca nesnel, koşullanmamış bir bakış açısından bakarsak kendimizi, başkaların ve dış dünyayı anlayabiliriz. Nesnellik bizim kişisel önyargıları aşmamızı, hakkani olmamızı ve dünyaya dengeli bir açıdan bakmamızı sağlar.*

## Öznellik Miti

*Öznellik miti şunları söyler:*

*1. Gündelik pratik etkinliklerimizin pek çoğunda duygularımıza güveniriz ve güvenebileceğimiz sezgiler geliştiririz. Önemli konular çıktığında, başkaları ne derse desin kendi duyu ve sezgilerimize güveniriz.*

*2. Yaşamımızdaki en önemli unsurlar duygularımız, estetik duyarlılıklarımız, ahlaki yaşantımız ve manevi uyanıklığımızdır. Bunlar tümüyle özneldir. Hiçbiri salt rasyonel ya da nesnel değildir.*

*3. Sanat ve şiir, rasyonaliteyi ve nesnelliği aşarlar ve bizi kendi duygu ve sezgilerimizin daha önemli gerçekliğine ulaştırırlar. Bu bilinçli akıl yürütmeden çok imgeleme dayalıdır.*

*4. İmgelemin dili, özellikle mecaz, yaşantımızın en biricik ve kişisel olarak en önemli yönlerini ifade etmek için gereklidir. Kişisel kavrayış meselelerinde sözcüklerin doğal, üzerinde anlaşılmış anlamlar işe yaramazlar.*

*5. Nesnellik tehlikeli olabilir, çünkü bireysel insanlar için en önemli ve anlamlı olan şeyleri kaçırır. Nesnellik haksız olabilir çünkü yaşantımızın en anlamlı beldelerini soyut, evrensel ve kişisel olmayan adına ihmal eder. Aynı nedenle nesnellik insanlık dışı olabilir. Duygularımıza, estetik duyarlılıklarımıza vs. ulaşmanın nesnel ve rasyonel bir yolu yoktur. İç yaşamımızdaki en önemli şeylere gelince, bilimin hiçbir faydası yoktur.*

**(sürecek)**

da çok yüksek ölçekli tümleştirme VLSI devrimi sayılabilir. Ancak bu dar boğaza yolaçan gelişmeleri gözden geçirmeden önce, bir şeye daha, bu işbölümünün dışında kalan süper bilgisayarlara, göz atmamız gerekiyor.

## Süper Bilgisayarlar

Süper bilgisayarlar yeni bir ürün olmadığı gibi, diğer büyük sistemlerden de nitel olarak değil ancak nicel olarak, içerdikleri yonga sayısı bakımından ayrılan ürünler. Bu nicel farkın büyüklüğü ve önemi hakkında bir fikir verebilmek için şöyle bir karşılaştırma yapabiliriz: Bugün bir Apple II'e kişisel bilgisayarında 31 adet yonga varken, Cray X-MP süper bilgisayarında 240.000 yonga vardır. Apple IIe 31 yongası sayesinde saniyede 500.000 işlem yapabiliyor; Cray X-MP ise saniyede 400 milyon işlem...

İlk süper bilgisayar 1964 yılında bir **DARPA** (Defense Advanced Research Projects Agency-İleri Savunma Araştırma Tasarıları Bürosu) tasarısı olarak üretilmişti. Bundan sonra da, süper bilgisayar üretimi ticari amaçlı kitlesel üretimi taşıyacak kadar yaygın bir kullanım alanı bulamadı. Gerçi son yıllarda uçak yapımı, maden ve petrol arama ve özellikle bilgisayar devre tasarımları alanında süper bilgisayarlar kullanılmaya başlandı ama daha 1983 yılında dünyada topu topu 74 adet (hepsi de Amerikan yapımı) süper bilgisayar vardı. Bu yüzden süper bilgisayarlar ticari firmalardan çok hükümet desteğiyle ya da Üniversiteler tarafından üretiliyordu. Ticari firmalar da, ürettikleri zaman, daha çok kendi araştırma geliştirme amaçlarını karşılamak için kullanıyorlardı.

**Mevcut teknoloji ile süper bilgisayarların yaygınlaşamaması gerekçesiz değil. Son 35 yıl içerisinde gerçekleşen bütün gelişmelere karşın, bilgisayarların temel işlem şeması -bilgisayar mühendislerinin bilgisayarın "mimarisi" dedikleri şey- değişmedi; sıralı ya da ardışık (sequential) işlem tarzına bağlı kaldı. Dünün vakum tüplü bilgisayarlarında olduğu gibi günümüzün transistörlü bilgisayarlarında da, bilgisayarın 'beyni' konumunda olan merkezi işlem birimi önce bir işlemi yapıyor, sonuçlarını depoluyor ancak ondan sonra bir ikinci işleme geçebiliyor. Bu ilke, 350.000 yonga içeren Cray-1 için de geçerli.**

**Bu mimari düzen içerisinde, bilgisayarın toplam hızını arttırmanın yolu y a veri aktarım ve depolama hızını arttır-**

## Japonların Katkısı

Buraya kadar anlattıklarımız Amerikan bilgisayar endüstrisinin gelişmesine ve bu gelişmenin yarattığı teknolojik yeniliklere ilişkindi. Ancak son yıllarda, bilgisayar piyasasasında etkilerini ilk önce 1970'lerde duyurmaya başlayan Japonların etkisi giderek daha fazla hissedilmeye başlandı. Öyle ki, beşinci kuşak tasarısı dahil olmak üzere, bilgisayar teknoloji ve piyasasındaki son yeniliklerin çoğu ya doğrudan doğruya, Japon kaynaklı ya da Japon Amerikan firmalarının rekabetinin bir türevi. Bu bölümde Japonların robot çalışmaları ve uygulamaları üzerinde durmayacak, daha doğrudan doğruya kişisel bilgisayar üretimine yönelik katkılarını gözden geçireceğiz. 1970'lerin yüksek ölçekli tümleştirme devriminin en önemli ürünlerinden biri, bellek yongaları, daha da özgül olarak 16K'lık RAM belleklerdi. Bu yongalar, tasarlanması ve üretilmesi en kolay yongalar olmakla birlikte, talebin de en yoğun olduğu ürünlerdi. Japonlar bu alanda önemli bir sıçrama yaptıkları ve 1978 yılında teknik olarak Amerikan Firmaları ile eşdeğer bir düzeye vardıkları halde, 16K RAM alanında, bu yılda dünya piyasasının ancak %40'ını ele geçirebildiler.

Bunun çeşitli nedenleri arasında, kuşkusuz Amerikan firmalarının piyasayı daha önceden tutmuş olmaları sayılabilir. Ama en önde gelen neden, Amerika'nın tüm dünya işgücü piyasası üzerinde kurmuş olduğu siyasal ve ekonomik egemenlikti. Gerçekten de ABD firmalarının satmakta olduğu devrelerin %80'i üçüncü dünya ülkelerinde imal edilmekte ve sınanmaktadır. Bu ise, ücretlerin düşüklüğünden ötürü, maliyeti düşürebilme konusunda ABD firmalarına hatırı sayılır bir avantaj sağlamaktadır.

**mak üzere kullanılan yongaların sayısını arttırmak ya da merkezi işlem birimi için daha hızlı devreler üretmeye çalışmaktır. Yongaların sayısını arttırmak bilgisayarın maliyetini arttırıyor; sözünü ettiğimiz 240.000 yongalı Cray X-MP'nin maliyeti 11 milyon dolar.**

**Daha hızlı elektronik devreler üretmek konusunda ise belli süper bilgisayarlarda bile bir sınıra ulaşılmış durumda. Bugün ulaşılmış olan yonga sayısını arttırmak, bilgisayarın hızını arttırmayacak; çünkü veri aktarım ve depolama hızı artık merkezi işlem biriminin aktarılan verileri işleyebilme ve depolama hızına ulaşmış durumda. Bundan sonraki arttırmalar olsa olsa bir bekleyen veriler kuyruğuna, bir 'trafik sıkışıklığı' yolaçabilir.**

Ancak çok yüksek ölçekli tümleştirme **(very large scale intergration-VLSI)**, bir yandan ABD'nin bu avantajını yitirmesine yol açarken, diğer yandan da 1990'ların bilgisayarları ve bilgisayar piyasası için yepyeni ufuklar açtı.

## Çok Yüksek Ölçekli Tümleştirme-VLSI

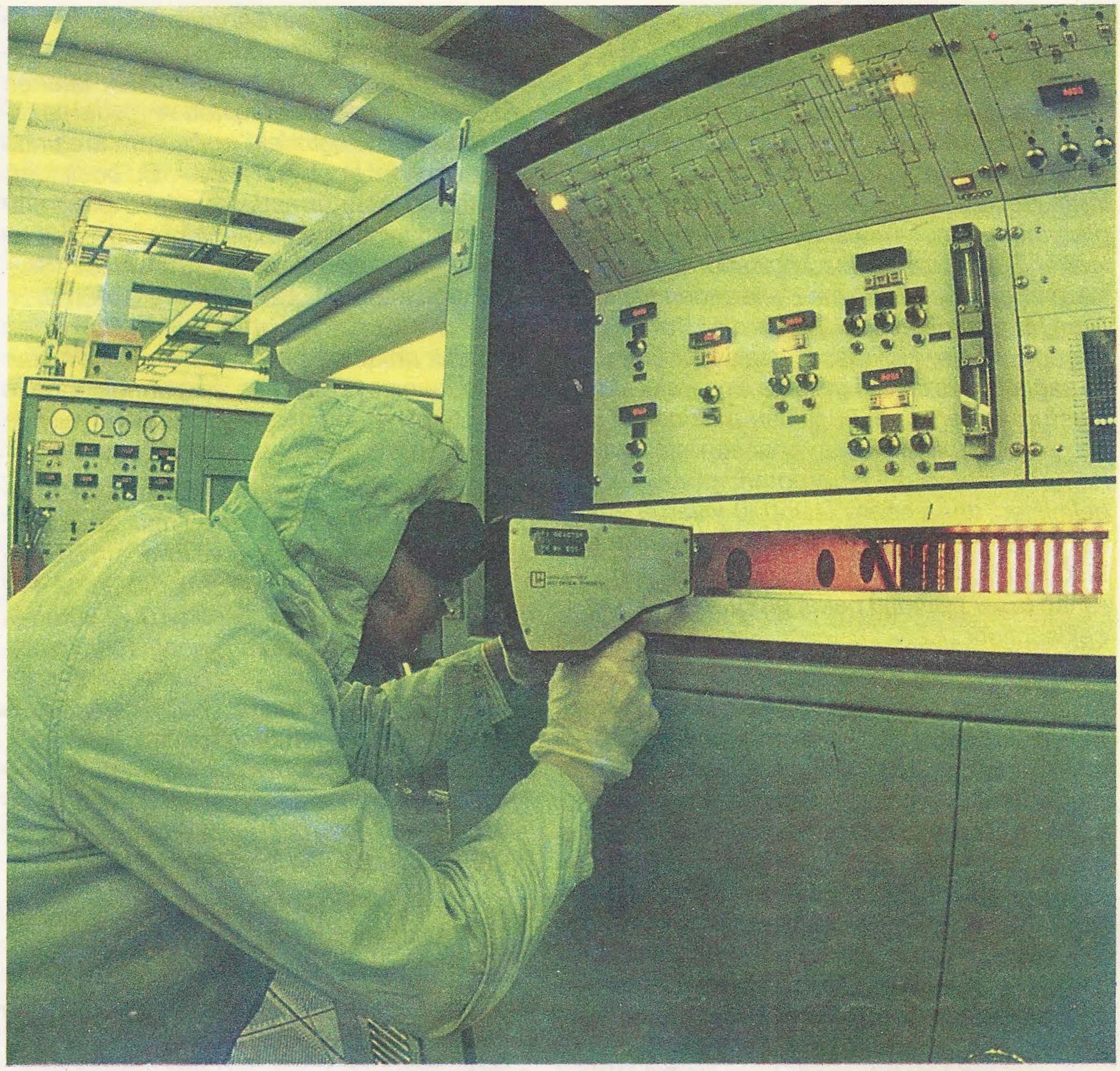
Çok yüksek ölçekli tümleştirme, adının düşündürebileceği gibi, LSI teknolojisine göre sadece nicel bir ilerleme değildir. Ya da daha doğrusu, VLSI teknolojisinin, bir yonganın ancak sığdırılabilecek transistör sayısında sağladığı nicel artış, bugün bilgisayar üreticileri ve ürünleri piyasasında yerleşmiş olan işbölümü ve ayrımları köklü bir şekile dönüştürecek kadar önemli bir artıştır.

Örneğin, en ileri VLSI teknolojisinden yararlanarak tasarlanmış olan Hewlett-Packard 9000'in merkezi işlem birimini oluşturan yonga 450.000 adet transtör içermektedir. (LSI teknolojisinin bir yongada 'binlerce' transistörü mümkün kıldığını hatırlayalım.) Her ne kadar bir masa bilgisayarı olan HP 9000'in fiyatı, bugün 40.000 dolar civarında ise de, gücü ve yetenekleri bakımından bu bilgisayar kişisel bilgisayarlarla değil, genellikle bir buzdolabı kadar yer kaplayan, minikişlemcileri düzinelerce yonga içeren ve yüzbinlerce dolara satılan büyükçe ''küçük bilgisayarlar'la **(minicomputers)** karşılaştırılabilir. Örneğin HP 9000 saniyede 1 milyon komut işleyebilmektedir.

Bu henüz bir başlangıç. Yakın bir gelecekte VLSI teknolojisine göre üretilen kişisel (micro) bilgisayarların saniyede 3 milyon işlem yapabileceği düşünülüyor. Bu ise tek bir yongadan oluşan bir merkezi işlem birimi içeren bilgisayarların yeteneklerinin, büyük bilgisayarlarınkine **(mainframe)** ulaşacağı anlamına geliyor. Oysa bugünkü büyük bilgisayarların merkezi işlem birimi yüzlerce ve hatta bazen binlerce yonga içeriyor.

Görüldüğü gibi bu gelişmeler, bugün bilgisayar türlerinden sözederken kullanmaya alışkın olduğumuz ayrımları (ev bilgisayarı, kişisel bilgisayar, küçük bilgisayar, büyük bilgisayar) anlamsızlaştırıyor.

Ürünler düzeyindeki eski sınıflandırmaların anlamsızlaşması, üretici firmalar için de yeni sorunlar doğuruyor.



## 5. KUŞAK BİLGİSAYAR SİSTEMLERİNE GENEL BAKIŞ

*Günümüzde gelene dek bilgisayarlar çeşitli aşamalardan geçmiştir. Bu gelişim sürecinde 1. kuşak olarak adlandırılan bilgisayarlar, radyo lambalarından (triyot, vb.) kullanılarak yapılıyordu. 1946-1956 yıllarında geliştirilen 1. kuşak bilgisayarlar (Eniac, Univac, vb.) büyük hacimli ve çok yavaştılar. 1950'lerde bulunan transistör ile 2. kuşak bilgisayarlar (IBM 7094, CDC, 6600, vb.) yapıldı (157-1963). Tümleşik devreleri kullanımı ile ise 3. kuşak bilgisayarlar (IBM 360, IBM 370, Vax 11/780, vb.) yapıldı. (1964-1981). Daha sonraları ise çok büyük ölçekli tümleşik devrelerin (VLSI) kullanımı ile 4. kuşak bilgisayarlar (Cray XMP, IBM 3080, Amdahl 580, vb.) yapılmıştır.*

*Yukarıda sözü edilen gelişim içinde bilgisayarın kullanım alanları içinde bilgisayarın kullanım alanları genişlemiştir. 1950'lerin bilgisayarları sadece bilimsel uygulamalar için kullanılır iken, 1960 ve 1970'lerde buna ticari ve endüstriyel uygulamalar katılmıştır. 1980'li yıllarda ise sıradan kişiler de (tüketiciler de) bilgisayarları kolayca kullanabilir duruma gelmişlerdir. Bugün, rahatlıkla bilgisayar kullanan, fakat bilgisayarın çalışma ilkelerinden hiç anlamayan ''bilgisayarcı''lar vardır. Bu kayda değer gelişme, kuşkusuz, bilgisayar kullanımı sağlayan yazılıdaki gelişme ile paralel oluşmuştur. Örneğin, ilk kuşakları makine dili ile programlanabilen bilgisayarlar sonraları çevirici (assembler) dili ile programlanabilir hale gelmişler, daha sonra (1950'li) yıllarda yüksek düzeyli diller geliştirilmiştir. Günümüzdeki kullanım kolaylığı, kullanıcıya dostça davranan (user friendly) paket programlar ile gerçekleşmiştir.*

*Çok sayıdaki bilgi üzerinde hızlı çalışmaları için tasarımlanan bilgisayarlar, gerçekleştirimde kullanılan teknolojinin değişmiş olmasına karşın tasarım ilkesi açısından bir aşama kaydetmemiştir. Günümüz bilgisayarlarının yapısı ilk kuşakta da olduğu gibi bugün de von Neumann'ın tanımlamış olduğu yapıya uymaktadır. Bu yapıda bilgi ve komutlar bellekte tutulmakta ve makine komutları bilgileri tek tek işlemektedir. Ancak sözkonusu bilgisayar mimarisi, günümüz bilgisayar kullanım alanlarının ve biçimlerinin büyük farklılıklar göstermesi üzerine kimi sorunlara yol açmaktadır. Bu sorunlardan bazıları aşağıda verilmiştir:*

*1 / Günümüz bilgisayarlarının kullanıcıya sağlamış olduğu programlama dilleri alt düzeyde veri soyutlamasına (data abstraction) izin vermektedir. Bu durum birtakım uygulamalarda (doğal dil işleme gibi) önemli bir engelleyici (gerçekleştirimi zorlaştırıcı) olmaktadır.*

*2 / Günümüz bilgisayarları bir problemin göze alınan yönleri için çözüm getirebilirler. Ancak, kimi alanlarda (örüntü tanımapattern recognition, yapay us -artificial intelligence- doğal dil işleme, vb. gibi) tek çıkar yol sınama-yanılma olmaktadır. Günümüz bilgisayarlarında uygulanabilen çözüm yaklaşımları ise yapıları gereği bu alanlarda yetersiz kalmaktadır.*

*3 / Günümüzde, bir sorunun bilgisayar ile çözümü için, bir modelin geliştirilmesi gerekmektedir. Ancak, çözülmek istenen sorun ile modeli arasında ve model ile bilgisayarın ''içindeki'' gösterim arasında anlamsal boşluklar vardır. Bu boşluklar, model oluşturmada, modelin programlanmasında ve geliştirilen programın işletilmesinde sorunlar yaratmaktadır. Sözkonusu durum 1970'lerin ve günümüzün yazılım krizine yol açmıştır. Bunun en açık anlatımı, bir uygulama için yapılan donanım-yazılım harcamalarında, yazılım payının günümüzde yüzde 80 civarında olduğunu belirtmek olacak/1, s.11/.*

*Günümüz bilgisayarlarının yukarıda kısaca değinilen sorunlarına çözüm bulmak amacı ile, Japon Uluslararası Ticaret ve Endüstri Bakanlığı (MITI: Ministry of International Trade and Inrdustry), 1981 Ekim ayında 10 yıl sürecek ve 850 milyon dolara mal olacağı tahmin edilen beşinci kuşak bilgisayar sistemleri projesini (BKBSP) duyurmuştu. BKBSP, günümüzdeki bilgisayarlardan binlerce kez daha hızlı, öğrenebilen, mantık yürütebilen, konuşma dilini anlayabilen, kısacası, zekası gerektiren işleri başarabilecek sistemleri önermektedir. BKBSP'nin amaçları şöyle özetlenebilir.*

*1 / Öğrenme ve akıl yürütmenin gerektirdiği temel mekanizmaları donanım düzeyinde gerçekleştirmek;*

*2 / Yukarıda söz edilen donanımı sonuna dek kullanabilecek yazılımı hazırlamak;*

*3 / Bir bilgi tabanını (knowledge base) donanım ve yazılım düzeyinde gerçekleştirmek (Bilgi tabanı bir veri tabanının -data base- akıl yürütme işlemleri yapabilecek biçime geliştirilmiş hali olarak düşünülebilir);*

*4 / Örüntü tanıma ve yapay us konusunda geliştirilmiş çalışmalardan, insan-makine ilişkisini kullanıcıya büyük kolaylıklar sağlayacak biçimde yararlanmak;*

*5 / Yazılım geliştirmede karşılaşılan zorlukları çözebilecek programlama olanakları yaratmak.*

*Projenin yürütülmesi için kurulan enstitü, Yeni Kuşak Bilgisayar Teknolojisi Enstitüsü (Japonca kısaltımı ICOT) sekiz Japon şirketinin (Fujitsu, Hitachi, Nippon Electric Corporation, Mitsubishi, Matsushita, Oki, Sharp, Toshiba)*

ve iki ulusal araştırma laboratuvarının (Nippon Telephone and Telegraph'ın Musashino Laboratory ve MITI'nin Electrotechnical Laboratory) katkıları ile oluşturulmuştur. ICOT'ta yukarıda sözü geçen kuruluşlardan izinli 40 araştırmacı çalışmaktadır. Bu araştırmacılar, üniversitede görevli olan bilim adamlarınca da desteklenmektedir. ICAT'ta yapılan araştırmaya koşut olarak, projeye katılan firmalara da kontratla iş verilmektedir. Kaynakları kullanmada tüm yetki ICOT'a aittir, ancak bütün mali kaynak MITI'den gelmektedir.

Resmen 1982'de başlatılan ve on yıllık bir süreyi kapsayan BKBSP üç aşamadan oluşmaktadır. Üç yıllık ilk aşamada ilk araştırmalar yapılacak, gerekli araçların ilk taslakları oluşturulacaktır. (Bu aşama sonuçlanmıştır.) Dört yıllık ikinci aşamada paralel çalışması amaçları makinelerin prototipleri oluşturulacak ve üç yıllık son aşamada değerlendirme yapılarak gerekli değişiklikler oluşturulacaktır. Sonuçta oluşacak makinelerin, 1982 yılının bilgisayarlarına göre 10.000-100.000 arasında daha hızlı olması amaçlanmaktadır. Bu hız artışının, teknolojinin beklenen gelişiminden ötürü sağlayacağı doğal hızlanmadan çok daha fazla olduğu açıktır.

## Beşinci Kuşak Bilgisayarlarının Yapısı

Yapısı, bu bölümde, kısaca tanıtılacak olan beşinci kuşak bilgisayarların gerçekleştirimi için gerekli olan bilimsel ve teknolojik birikiminin sağlandığı söylenebilir. BKBSP için gereken bilimsel ve teknolojik gelişmeler dört ana başlıkta toplanabilir.

Yapay Us: Bu konuda üretilen ve insan-makine arasında kolay iletişimi sağlayacak olan doğal dil işleme, konuşma ve görüntüyü anlama aşamasına ulaşmıştır.

Yazılım Mühendisliği: Yeni yüksek düzeyli programlama dilleri, hesaplama modelleri ve programlama çevresi sağlanmıştır.

Bilgisayar Mimarisi: Bu konuda, bilgisayar ağlarını destekleyen dağıtık (distributed) mimariler, çok hızlı çalışan paralel bilgisayar yapıları tanımlanmış ve gerçekleştirilmiştir.

VLSI Teknolojisi: VLSI teknolojisinde büyük ilerlemeler kaydedilmiştir. Devre yapımında galyum arasenür (GaAs) kullanılması VLSI ile yapılmış devrelerin çok hızlı çalışmasını sağlamaktadır. Ayrıca VLSI'lı devrelerin tasarımında bilgisayar destekli tasarım (CAD) gerçekleştirilmiştir.

Gerçekleştirimi, özellikle yukarıda belirtilen konular ile ilintili olan BKBSP, günümüz bilgisayarlarından farklı olarak veri (data) değil bilgi (knowledge) işlemeye yönelik olacaktır. Geliştirilecek olan bilgisayarlar, bilgi bankalarının yardımı ile, kullanıcıların akıllı yardımcıları olacaklardır. Bu bilgisayarların hastalık tanımı, danışma, dil çevirici, ofis otomasyonu, bilgisayar destekli tasarım-üretim ve benzeri konularda çok etkin bir biçimde kullanılacağı beklenmektedir. Beşinci kuşak bilgisayarlarına, yoğun bir biçimde bilgi kullanıp ürettikleri için Bilgili Bilişim Sistemi (Knowledge Information Processing System: KIPS) adı da verilmiştir.

Beşinci kuşak bilgisayarları üç makinenin bütünleşmesinden oluşacaktır:

1 / Problem çözüme ve karar verme makinesi (problem solving and inference machine),
2 / Bilgi tabanı makinesi (knowledge hase machine),
3 / Akıllı arabirim makinesi (intelligent interface machine).

1 / Problem çözme ve karar verme makinesi

Günümüz bilgisayarının merkezi işlem birimine (CPU: Central Processing Unit) karşılık gelen problem çözme ve karar verme makinesinin oluşmasında, projenin öteki makinelerinde de olduğu gibi, aşamalı bir gerçekleştirme planlamıştır. Bu aşamaların başında yer alan ve kararları peş peşe veren PSI (Personel Sequential Inference) makinesi gerçekleştirilmiş durumdadır. PSI, yaklaşık 20-30 bin mantıksal çıkarımı bir saniyede gerçekleştirmektedir. Başka bir deyiş ile PSI'nın hızı 20-30 bin LIPS'dir.

(LIPS: Logical Inferences Per Second. Beşinci kuşak bilgisayarlarının hızı LIPS ile ölçülmektedir. Günümüz bilgisayarlarında 1 LIPS'in 100-1000 arasında komut ile gerçekleştirileceği hesaplanmıştır. Dolayısı ile, 1 LIPS günümüz bilgisa-

yarları için 100-1000 komut/saniye-IPS-demektir.) Daha sonra yapılacak olan süper PSI'lerin hızı $10^5$ - $10^6$ LIPS düzeyinde olacaktır. Süper PSI'deki deneylerden yararlanarak oluşturulacak olan SIM (Sequential Inference Machine) daha büyük bir hıza sahip olacaktır. SIM, iki altbirimden oluşmaktadır: Sembol işleme makinesi (symbol manipulation machine) ve sayısal hesap makinesi (numerical computation machine).

Japonlar, araştırmanın ileriki aşamalarında, 1000 kadar SIM'i paralel olarak çalıştırmayı planlamışlardır. Bu aşamada, paralel çalışan makinelerin hızının, $10^8$ - $10^9$ LIPS dolayında olacağı umulmaktadır. 1 LIPS günümüz bilgisayarları için 100-1000 IPS olduğuna göre, $10^{11}$ - $10^{12}$ IPS gerektirecektir. Örneğin, yaygın mikroişleyicilerden Motorola 68000'in hızı $10^6$ IPS (1 MIPS)'dir; VAX 8600'ün hızı ise $4x10^6$ IPS'dir. Dolayısı ile bir beşinci kuşak bilgisayarı, Motorola 68000 den $10^5$ - $10^6$ ve VAX 8600'den ise $2,5 \times 10^4$ - $2,5x10^5$ dolayında hızla olacaktır.

**2** / Bilgi tabanı makinesi

Bir bilgi tabanı, veri tabanının genişletilmiş biçimi olarak düşünülebilir. Eğer veri tabanını bir gerçekler ve gerçekler arasındaki ilişkiler biçiminde düşünürsek, bir bilgi tabanını veri tabanı ve kurallar topluluğu olarak düşünebiliriz. Bu kurallar yardımı ile, veri tabanında yer almayan yeni ilişkiler ve gerçekler üretilebilir. Bilgi tabanı yöntemi ile bilginin gösterimi, saklanması, ulaşılması, değiştirilmesi ve üretilmesi biçiminde tanımlanabilir. Bir bilgi tabanı makinesinin gerçekleştirilmesinde en önemli payın bir veri tabanı makinesi olacağı sonucuna varılmıştır. Sonuç olarak, paralel çalışan bir veri tabanı makinesi (Delta) ile bilgi tabanı makinesinin yapılmasına başlanmıştır. Sonuçta oluşturulacak olan bilgi tabanı makinesinin $10^{11}$ - $10^{12}$ byte büyüklüğünde bir belleğe sahip olması düşünülmektedir.

**3** / Akıllı arabirim makinesi

Yukarıda adı geçen iki makinenin ve insan-makine arasındaki iletişimi doğal diller ve görüntüden anlama düzeyine çıkartan akıllı arabirim makinesinin bütünleştirilmesi ile beşinci kuşak bilgisayarı oluşturulmuş olacaktır. Bu amaçla, konuşmanın ve görüntünün anlanması, üretilmesi, saklanmasına yönelik donanım ve yazılım çalışmalarına başlanmıştır.

## Icot Tarafından Geliştirilen Bilgisayarlar

Yazının bu bölümünde, ICOT'un BKBSP'nin birinci aşamasında geliştirilmiş olduğu PSI ve Delta makineleri kısaca tanıtılacaktır. Sözkonusu makinelerin ilk örnekleri 1984 Kasım'ında Tokyo'da ICOT'ca düzenlenen konferansta sergilenmiştir. ICOT'da çalışan araştırmacılar, sözkonusu makinelerin işlevsel tanımı yapmışlar, ICOT'a katkı veren ticari firmalar ise gerekli ayrıntılı tasarımı yapmış ve makineleri gerçekleştirmişlerdir.

Projenin amacı, paralel çalışan (çok sayıdaki işlemi aynı anda yapan) karar (inference) makinesi ve bilgi tabanı makinesi olmasına rağmen, ilk aşama sonucunda yapılan çalışma istasyonu (work station) sıralı (sequential) çalışma ilkesini amaçlamaktır.

İlk aşama sonucunda PSI (Personel Sequential Inference) adı verilen bir iş istasyonu ve Delta adı verilen bir veri tabanı makinesi geliştirilmiştir. PSI'ı destekleyen dil Kernel Larguage O (KLO) olarak adlandırılmış ve KLO kullanılarak ESP (Extended Self-contained Prolog) adlı bir işletim sistemi gerçekleştirilmiştir. Daha gelişmiş bir yazılım için yapılan çalışmalar, bir paralel bilgi işleme dili olan KL1'i oluşturmuştur. KL1'in yanı sıra bilgi tabanlarında kullanılmak üzere Mandala dili de yaratılmıştır. Geliştirilen yazılımı denemek amacı ile birtakım küçük uygulamalar yapılmışsa da, ICOT henüz kayda değer bir uygulamanın yapıldığını belirtmemektedir.

PSI'ın amacı, mantık programlamasına (logic programming) uygun bir mimariyi gerçekleştirmek için gerekli deney ortamını yaratmaktır. Bu amaçla, PSI mikro-programlanabilir bir biçimde hazırlanmıştır. Bu özellik, araştırmacıların değişik komut kümelerini aynı makinede deneyerek en iyi komut kümesini bulmalarını sağlayacaktır.

PSI'ın sıradan çalışma istasyonlarına göre en önemli farkı belleğinin 80M byte ($M = mega = 10^6$) olmasıdır. Bu büyüklükteki bir bellek, henüz günümüz bilgisayarların da bile, pahalılığından ötürü, sözkonusu olamamaktadır. Ancak teknolojinin gidişinden, bu büyüklükteki belleklerin yakın gelecekte ticari bilgisayarlarda da olası olacağını göstermektedir. Böylece, ICOT, geleceğin çok büyük bellekli uygulamalarını şimdiden gerçekleştirerek bu türden büyük bellekli makinelerin sahip olması gereken özellikleri ortaya çıkartacaktır. PSI'da denenen büyük bellek uygulaması, şimdiden kimi yeniliklere yol açmıştır. Örneğin, ICOT üyesi olan Nippon Electric Corporation, bu günlerde 320 M byte büyüklükte bir belleği olan iş istasyonunu piyasaya sürmek üzeredir.

PSI, gelişmiş iş istasyonlarında rastlanan gelişmiş insan-makine ilişkisini sağlamaktadır. 1200-900 nokta (pixel) ile tanımlanmış olan göstergesinde her nokta adreslenebilmektedir. Grafik görüntülerini denetlemek için ''mouse'' olanağı da vardır. ICOT, bu amaçla gerekli olan donanımı Japonya dışından sağlamış, ancak yazılımı kendi geliştirmiştir.

PSI'nın belleğinde sözcük boyu 40 bit olup, bu 40 bit'den 32'si veri için, 8'i ise işaret (tag) amacı ile kullanılmaktadır. İşaret bitleri, üst düzeyli dillerde kullanılacak bilgilerin tümünü belli etmek gibi amaçlar için kullanılmaktadır. İşlemleri, hızlandırmak amacıyla, sistem 40 bit x 4K'lık (1K = 1024) iki cache bellek içermektedir. Ayrıca, bir adres çevirme birimi, 32-bit'lik görüntü bellek adreslerini 24 bit'lik fiziksel adreslere dönüştürmektedir.

PSI'deki mikroprogramlama KLO ile yazılmış programları yorumlama (interpret) edebilmektedir. KLO, Prolog (Programming in Logic) diline çok benzemektedir.

İlk aşamada geliştirilen Delta, bir ilişkisel veri tabanı makinesidir. Delta, yapay us uygulamalarında bilgi tabanı işlemlerini gerçekleştirmek amacı ile kullanılacaktır. ICOT genel sistem yapısını, çok sayıdaki PSI'nın çok hızlı çalışan bir ağ ile az sayıda Delta makinesine bağlanması biçiminde görmektedir. Böylece, bir bilgi tabanı iş istasyonları tarafından paylaşılacaktır. Delta makinesi, bir ağ arabirim işlemcisi, üç ilişkisel makine ve bir denetim işlemcisinden oluşmaktadır. Sistemde 128 M byte'lık bir ana bellek ve 20 G byte ($1G = 10^9$ yan bellek bulunmaktadır. Ağ arabirimi, Delta'nın dış dünya ile ilişki kurmasını sağlamaktadır.

Delta, bilgisayar ağından gelen ilişkisel sorgulara karşılık üretmek amacını gütmektedir. Gelen sorgular denetim birimine iletilmektedir. Bu sorgular ilişkisel makinelerce işlenebilecek biçimlere dönüştürülmektedir. Bunlar, SIRALA (sort), BİRLEŞTİR (join) ve kısıtla (RESTRICT) gibi işlemlerdir. Bir, ya da daha fazla ilişkisel makine bu işlemleri yapmak için görevlendirilmektedir. İşlemler, eğer uygun ise, makinelerce koşut bir biçimde yerine getirilir.

PSI ve Delta, ilke olarak, sıralı (sequential) çalışan makinelerdir. PSI'daki paralellik, geleneksel giriş/çıkış ve işlemi aynı yapmak ilkesidir. Delta'daki paralellik ise, yapılan işlemleri küçük bir çarpan oranında hızlandırabilmektedir.

Altı PSI makinesi Nisan 1984'ten bu yana kullanılmaktadır. Şu sıralarda bu sayının yirmiye ulaşması beklenmektedir. PSI ve Delta'nın alışılmışın dışındaki büyük belleği, mantık programlaması ve ilişkisel işlemlere yönelik yapıları ile günümüz bilgisayarlarından farklılık göstermektedirler. BKBSP'nin sonunda gerçekleştirilen makineler belki PSI ve Delta ile çok fazla bir benzerlik göstermeyeceklerdir; ancak, sözkonusu iki makinenin kendilerinden sonraki gelişmeleri önemli oranda etkileyecekleri beklenmektedir.

ICOT, projenin bundan sonraki yedi yılında üretilen ve üretilecek olan prototipleri en iyi biçime dönüştürülmeye çalışılacaktır. Bu yedi yıl içinde, yapılabilecek ya da yapılmış olan yanlış tasarım biçimlerinden dönmek için pek vakit kalmayacağı sanılmaktadır. Yapılan seçimlerdeki riskler, amaçlarının uygulamaların gerektirdiği donanımın ve yazılımın tam olarak bilinememesinden kaynaklanmaktadır.

## Beşinci Kuşak Bilgisayarlarının Programlama Dili: Prolog

Japonların beşinci kuşak bilgisayarlarının çekirdek dili olarak Prolog'u (programming in Logic) seçmeleri ile Prolog dili büyük önem kazanmıştır. Prolog yaklaşık on yıl önce, öncülüğünü A. Colmerauer'in yaptığı Yapay Üs Grubu (GIA: Groupe d'Intelligence Artificielle) tarafından Marsilya'da geliştirilmiştir.

Prolog'un kullanmış olduğu mantıksal çıkarım-

lar ile programlama yaklaşımı şöyle anlatılabilir.

— Bilgisayara bilgileri mantık kullanarak aktarmak;

— Bilgisayara problemleri mantık kullanarak aktarmak;

— Mantıksal çıkarım kullanarak bu problemleri bilgisayara çözdürmek.

Bu yaklaşımı izleyen GIA, 1973 yılında Fransa'da FORTRAN ile ilk Prolog yorumlayıcısını gerçekleştirmiştir. Bunu dünyanın çeşitli yerlerinde yazılan Prolog yorumlayıcıları ve derleyicileri izlemiştir. Ancak 1980 yılına dek bu çalışmalar akademik çevre içinde kısıtlı kalmıştır. Prolog'un beşinci kuşak bilgisayarlarının çekirdek dili olarak seçilmesinden sonra durum değişmiş, Prolog endüstride de çok yaygın olarak araştırma konusu haline gelmiş ve Prolog'un kullanılabileceği değişik alanlar araştırılmaya başlanmıştır.

Prolog ile program yazmak için

— Nesneler ve nesnelerin arasındaki ilişkileri anlatan gerçeklerin (facts) tanımlanması;

— Nesneler ile ilgili ve nesnelerin arasındaki ilişkileri düzenleyen kuralların tanımlanması gerekmektedir. Gerçeklerin ve kuralların tanımlanmasından sonra, nesneler ve ilişkiler, hakkında sorular sorulabilir.

Prolog dili için basit bir örnek aşağıda verilmiştir. Bu örnekte nesneler kişiler (kişiler adları ile tanımlanacaktır); ilişkiler ise kişiler arasındaki akrabalık ilişkileridir. Öncelikle, birtakım gerçeklerin Prolog cümleleri ile tanımlanması gerekir. Aşağıda bu amaçla iki örnek verilmiştir:

```
baba (ali, ahmet).
baba (ahmet, aydın).
```

Yukarıdaki ilk cümlede Ali'nin Ahmet'in babası olduğu belirtilmekte, ikincisinde ise benzer biçimde Ahmet'in de Aydın'ın babası olduğu gerçeği belirtilmektedir.

Nesneler arasındaki gerçekler tanımlandıktan sonra, nesneler arasındaki ilişkileri düzenleyen kurallar tanımlanır:

```
büyükbaba (x,y):-baba (x,y),
                 anne (y,z).
büyükbaba (x,z):-baba (x,y),
                 baba (y,z).
```

Yukarıdaki Prolog cümleleri x'in z'in büyükbabası olabilmesi için gerekli kuralları belirtmektedir. Bu kurallara göre x'in y'in büyükbabası olabilmesi için x'in y'in babası ve y'nin de z'nin annesi olması, ya da x'in y'nin babası ve y'nin de z'nin babası olması gerekmektedir.

Gerçeklerin ve kuralların tanımlanmasından sonra nesneler arasındaki ilişkiler üzerine sorular sorulabilir. Örneğin ''Ahmet, Aydın'ın babası mıdır?'' sorusu Prolog'da

```
[baba (Ahmet, Aydın)?]
```

biçiminde yazılır. Bu sorunun karşılığı olan ''evet'' ya da ''hayır'' tanımlanmış gerçekler incelenerek bulunur. Prolog'da [büyükbaba (ALİ,R)?] biçiminde yazılacak olan ''Ali'nin torunu kimdir?'' sorusu ise kurallar aracılığı ile cevaplanır. Bu amaçla sorgu bütün ''büyükbaba'' tanımları ile çakıştırılır:

```
büyükbaba(x = ali,z = R):
        -baba(ali,y),anne(y,R).
büyükbaba(x = ali,z = R):
        -baba(ali,Y),baba(y,R).
```

Yukarıdaki kurallara göre, Ali Y'nin babası ile ve Y, R'nin annesi (ya da babası) ile Ali'nin torunu Y'dir. Daha önceden tanımlanmış olan gerçekler gereği Aydın Ali'nin torunu olarak bulunur.

Prolog kullanımının uygun olacağı ve kullanılmakta olduğu alanlar şunlardır:

— Bilgisayar yardımı ile tasarım;

— Mantıksal (çıkarımlar yapmak;

— Veri tabanı yönetim sistemleri;

— Doğal dillerin bilgisayar tarafından anlaşılmasını sağlamak üzere yapılan çalışmalarla yaygın olarak kullanılmaktadır;

— Yüksek düzeyde benzetim yapmakta (bu uygulamada benzetim modelinin özellikleri Prolog cümleleri ile verilebilmektedir).

— Prolog ayrıca sembolik matematik, matematiksem mantık, derleyici ve uzman sistemleri tasarım ve uygulanmasında kullanılması da çeşitli yararlar sağlamaktadır.

Görüldüğü gibi, Prolog'un kullanılabileceği ve kullanılmakta olduğu alanlar beşinci kuşak bilgisayarlarının kullanılmasının amaç olarak alındığı alanlar ile çakışmaktadır.

## 5. BKBSP'nin Öteki Ülkeler Üstündeki Etkileri

Japonların BKBSP öteki ülkeler üzerinde bilgisayar konusundaki yatırımları kamçılayıcı bir etkisi olmuştur. Bu bölümde, Batı ülkelerinde başlatılan çalışmalar kısaca tanıtılacaktır.

ABD'de BKBSP'ne tepki olarak MCC (Microelectronics and Computer Technology Corporation) Austin Texas'da 1984 yılında kurulmuştur. İlk çalışma yılında 50 milyon dolarlık bir bütçesi olan MCC'ye Advanced Micro Devices, Allied, Control Data, Digital Equipment Corporation (DEC), Harris, Honneywell, Martin-Marietta, Mostek, Motorola, NCR, National Semiconductor, RCA ve SPerry Computer Seystems katılmıştır. MCC çalışmalarını dört ana konuda yoğunlaştırılmıştır. (1) Mikroelektronik paketleme, (2) Yazılım teknolojisi, (3) Bilgisayar yardımı ile tasarım ve üretim (CAD/CAM), (4) İleri bilgisayar yapıları. MCC'nin çalışmaları ilk ürünlerini CAD/CAM konusunda vermiştir. Çok yakın bir gelecekte 10 milyon transistörlü bir yonganın ilk düşünülüşü ile yapılmış olması arasındaki sürenin bir aya düşeceği sanılmaktadır. Yazılım teknolojisi ile ilgili çalışmalar yazılım geliştirme süreci ve yazılım ürününün kalitesi konularında yoğunlaşmıştır.

Yakın bir geçmişte ABD Kuzey Carolina eyaletinde kurulan SRC (Semiconductor Research Cooperative), yarı-iletken endüstrisinin önde gelen 23 şirketinin katkısı ile kurulmuştur. SRC, adından da anlaşılacağı gibi araştırmalarını yarı-iletkenler üzerine yoğunlaştırmıştır.

Yine Kuzey Carolina eyaletinde bulunan MCNC (Microelectronics Center of North Carolina), MCC ve SRC'nin araştırmalarını tamamlayıcı yönde faaliyet göstermektedir.

MCNC'nin çalışmaları beş üniversite tarafından desteklenmektedir. (Duke University, North Carolina A& T State University, North Carolina State University, University of North Carolina at Chapel Hill, University of North Carolina at Charlotte).

ABD'de, yukarıda kısaca tanıtılan özel girişimce desteklenmiş kurumların yanısıra, devletçe de desteklenen çeşitli araştırma kurumları vardır. Bunlardan biri Carnegie-Mellon Üniversitesi'nde, Savunma Bakanlığı'nın (DOD) desteği ile kurulan SEI'dir (Software Engineering Institute). Bu kuruluş, çalışmalarını özellikle kritik işlerde (Savaş araçları, vb. gibi) kullanılacak olan yazılımın güvenilir olarak geliştirilmesi üzerine yoğunlaştırmıştır. SEI'nin bulguları SPC (Software Productivity Consortium) aracılığı ile MCC tarafından da kullanılabilecektir.

ABD'de devletçe desteklenen kuruluşlardan biri de Pentagon'a bağlı olarak çalışan DARPA (Defense Advanced Research Projects Agency)'dir. Bu kuruluş 1890'a dek 1 milyar dolara yakın bir harcama ile süper-zeki bilgisayarlar yapmayı amaçlamıştır. Bu bilgisayarlar özellikle savaş araçlarının otomasyonuna yöneliktir.

AET tarafından 1984 yılında başlatılan ESPRIT (European Strategic Program for Research and Information Technology) çeşitli devlet kuruluşları, üniversiteler ve firmalarca desteklenmektedir. ESPRIT'e katkı veren 12 firma şunlardır; İngiltere'den General Electric Co. (GEC), International Computers Ltd, Presley; Fransa'dan CGE, Machines Bull, Thomson Brandt; Almanya'dan AEG-Telefunken, Nixdorf Computer, Siemens; Hollanda'dan Philips; İtalya'dan Olivetti ve STET'dir. 1984 yılında kurulan ESPRIT, ilk beş yıllık süre için 1,5 milyar dolarlık bir yatarım öngörmektedir.

ESPRIT beş dalda araştırmalarını yoğunlaştırmıştır: İleri mikroelektronik, yazılım teknolojisi, ileri bilgi-işlem, büro otomasyonu, bilgisayar tümleşimli esnek üretim. Mikroelektronik dalında, ESPRIT, bir mikrondan daha küçük devreler yapmayı amaçlamıştır. Yazılım teknolojisinde ise özdevimli (otomatik) yazılım geliştirilmesi amaçlanmıştır. İnsan-makine iletişimi, ileri bilgi-işlem sistemleri ile, konuşma ve görüntü boyutlarına çıkartılacaktır. Büro otomasyonu çalışmaları, bir iletişim ağı içinde çalışan uçları (terminalleri) ses ve görüntü işlemeyi ofise getirmeyi amaçlamaktadır.

Bilgisayar-tümleşimli esnek üretim sistemleri, bilgisayar yardımı ile tasarım ve üretim, bilgisayar yardımı ile sınama ve tamir ve montaj konularını içermektedir.

İngiltere'de BKBSP'ne tepki olarak doğan çalışmalar yukarıda da belirtildiği gibi AET çerçevesinde ve ona ek olarak 1983 yılında başlatılan Alvey projesine bağlı olarak sürdürül-

*mektedir. 5 yıllık olan ve 500 milyon dolarlık bir bütçesi olan Alvey Projesi hükümet ve endüstri tarafından desteklenmektedir. Proje, ICOT ve DARPA'dan esinlenerek hazırlanmış olup, ESPRIT projesini tamamlayıcı özellikte-dir.*

*BKBSP'nin ABD, İngiltere ve AET devletlerinde yeni çalışmaların başlatılmasında etkili olduğu görülmektedir. Benzer biçimde, SSCB ve Doğu-bloku ülkelerinde de yetersiz bütçeli olarak değerlendirilen birtakım araştırmalar başlatılmıştır.*

## BKBSP ve Türkiye

*BKBSP ve bu projenin çeşitli ülkelerde başlatmış olduğu çalışmaları da inceledikten sonra düşünülmesi gereken durum, Türkiye'nin bu konuda ne yapabileceğidir. Açıktır ki, Türkiye şu andaki mali gücü, az sayıdaki yetişmiş insangücü, bilim ve teknoloji potansiyeli ile böylesine bir projeyi başlatamaz. Ancak yapılabilecek çalışmalar vardır:*

*(1) Beşinci kuşak bilgisayarlarının neyi amaçladığı bilinmektedir. Ülkemiz üniversitelerin bilgisayar mühendisliği ve ilişkin konularda eğitim veren bölümleri, beşinci kuşak bilgisayarlarınca hedef alınan ve araç olarak kullanılan konularda araştırma ve öğretim faaliyetlerini yoğunlaştırabilirler. Böylece beşinci kuşak atılımının ilk ürünleri yurdumuza girdiğinde hemen etkin bir biçimde kullanılabileceklerdir.*

*Bu sayede, yalnız kendi uzmanlarımızca geliştirilebilecek olan ülkemizin sorunlarını çözmeye yönelik, bilgi bilişim sistemleri, uzman sistemlerle, vb. gerçekleşecektir.*

*(2) Türkiye'nin AET çerçevesinde yürütelen ESPRIT araştırmalarına katkı vermesi sağlanabilir. Böylesine bir ilişki, yurda bilgi akışını hızlandıracak ve Türkiye'nin yazılım gücünün artmasını sağlayacaktır.*

*(3) BKBSP ve Türkiye'nin ne yapabileceği, çok daha geniş ve derinlemesine inceleme yapılmasını gerektiren bir çalışmadır. Bu amaçla üniversiteler, TÜBİTAK ve öteki araştırma kuruluşları işbirliği yapabilirler.*

*Sevindirici olan şudur ki, kimi üniversitelerimizde BKBSP'ne ilişkin konularda çalışmalar başlamıştır. Örneğin, beşinci kuşak bilgisayarlarının dili olarak kabul edilen Prolog dili öğretilmekte, PROLOG derleyicileri yazılmaktadır. Üniversiteler, bu konuda öncülüklerini kazandırarak sürdüreceklerdir.*

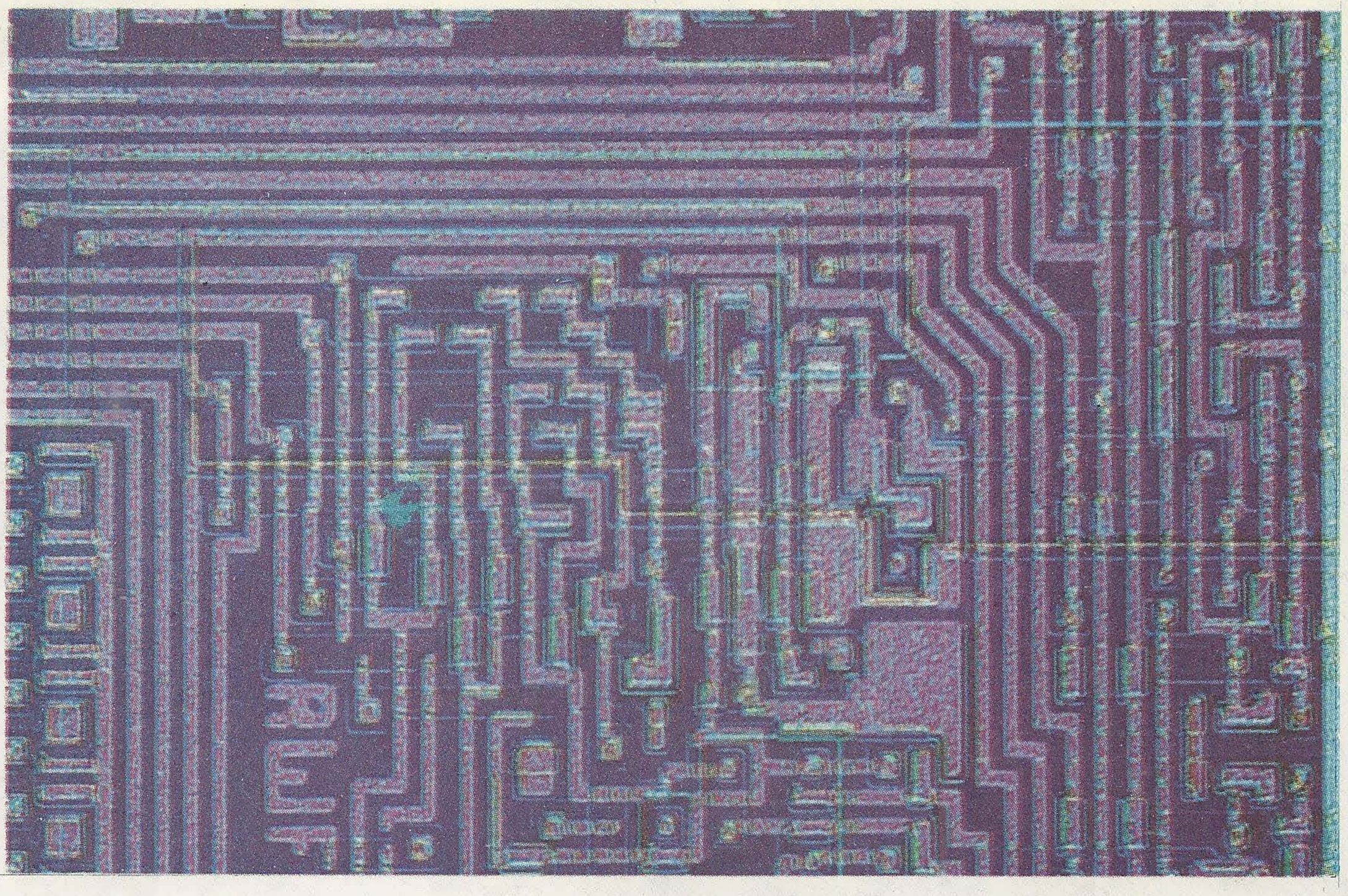
## Sonuç

*Bu yazıda, 1982 mali yılı içinde Japonya'da resmen başlatılan BKBSP ana hatları ile tanıtılmıştır. Günümüz bilgisayarlarından farklı olarak mantıksal çıkarımlar yapmaya yatkın olan bu sistemler, makine-insan ilişkisini doğal diller ve görüntü düzeyine çıkartma amacındadır. Yazıda, sözkonusu sistemlerin mimarisi, BKBSP'nin gerçekleştirme aşamalarının amaçları, sonuçlanan ile aşamanın ürünleri ve beşinci kuşak bilgisayarlarında türevleri kullanılacak olan Prolog kısaca anlatılmıştır. Yazıda ayrıca, Japonya'da başlatılan projenin, Batı ülkeleri üzerindeki etkisi ve bu ülkelerde başlatılan çalışmalar tanıtılmıştır.*

*BKBSP belki tüm amaçlarına ulaşamayacaktır, ancak gerek bu projenin, gerekse tepki olarak öteki ülkelerde sürdürülen araştırmaların, bilgisayar bilimine ve teknolojisine büyük katkısı olacağı kesindir.*

**Prof.Dr. Mehmet BARAY**
**Doç.Dr. Asuman DOĞAÇ**
**Dr.Fazlı CAN**

(BİLGİSAYAR DERGİSİ
3. Türkiye Bilgisayar Kongresi'ne sunulan bildiri)



**Beşinci kuşak tasarısını mümkün kılan en önemli etkenlerden biri, tek bir tümleşik devreye yüzbinlerce yongayı sığdıran VLSI teknolojisi olmuştur.**

Daha önce, bugün standart yarı-iletken yongalar üreten firmalarla, bu yongaları birleştirerek bilgisayar üreten firmalar arasında bir işbölümü olduğunu söylemiştik. Ancak tek bir yonganın, bu denli güç kazanması, hem bu işbölümünü, hem de yongalar düzeyindeki standartlaşmayı zorlayacaktır. Bu gelişmelerden en olumsuz bir biçimde etkilenecek olan, büyük bir olasılıkla bilgisayar üreten firmalar olacaktır. Tek bir yonganın, bir merkezî işlem biriminin bütün işlevlerini yerine getirmeye başlayabildiği durumlarda, başka firmalar tarafından üretilen parçaları birleştirerek bilgisayar üreten firmaların işlevleri, doğrudan yonga üreten firmalara kıyasla bir hayli daralacaktır.

## Japonlar ve VLSI

VLSI teknolojisinin gelişmesi, yalnızca farklı iş kolları arasındaki denge ve işbölümünü değiştirmekle kalmayacak, uluslararası işbölümü ve rekabeti de etkileyecektir. Ve zaten özellikle 1980'lerden beri etkilemektedir de.

VLSI teknolojisinin ilk yaygınlaşan ürünü, bugün ev bilgisayarları alanında bir standart haline gelmiş olan 64K'lık RAM bellek yongala-

rıdır. 1982 yılına varıldığında Japonlar bellek yongası üretimi teknolojisi alanında Amerikalı firmaları aşmışlar ve dünya piyasasasının % 70'ini ele geçirmişlerdi.

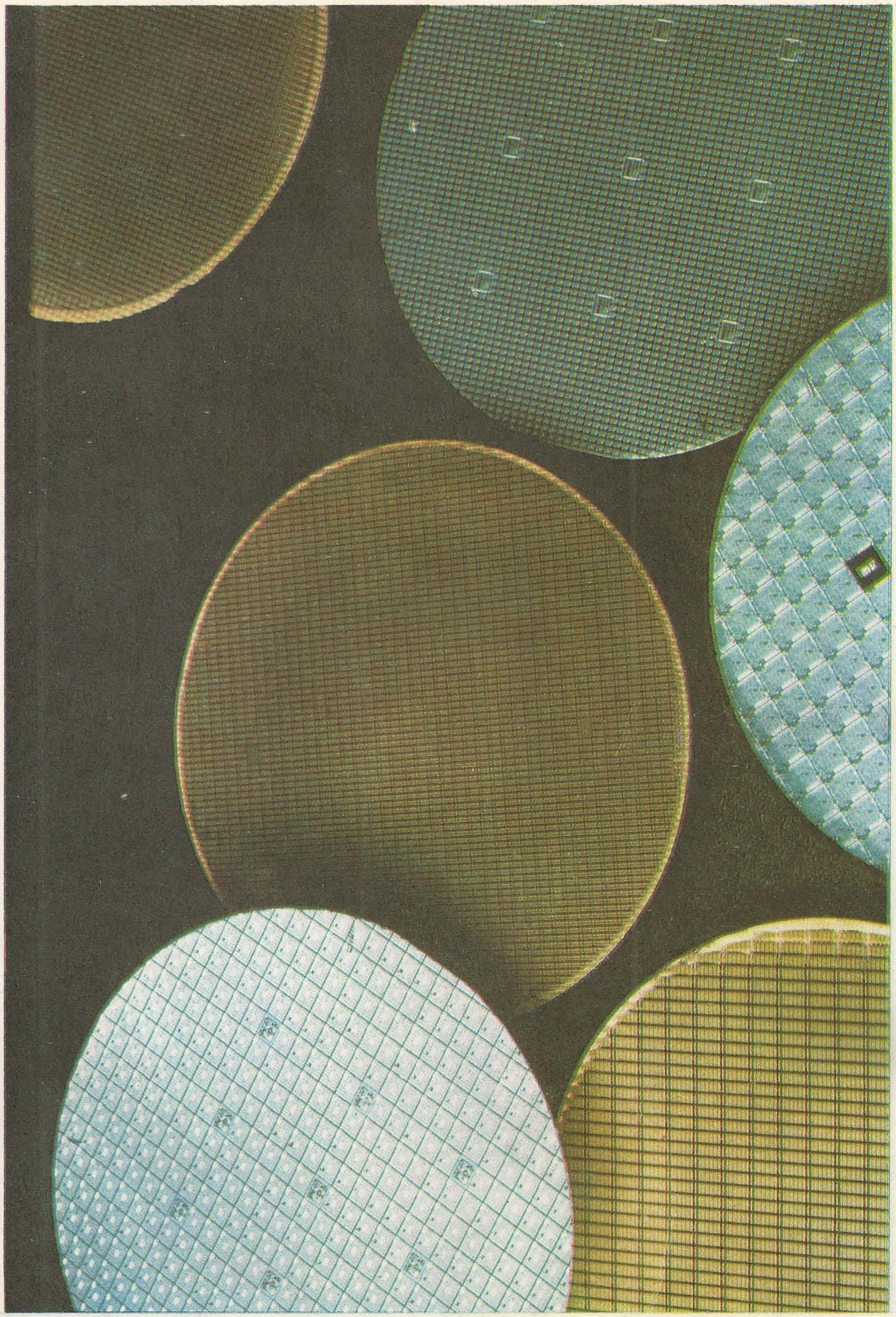
Bunun nedenleri 64K'lık RAM belleklerinin ve genel olarak VLSI teknolojisi ürünlerinin üretim gerekleri ile ilgili. Daha önceki 16K'lık bellekleriin aksine 64K'lık bellekleri hatasız olarak üretmek neredeyse olanaksız. Bu yüzden çoğunlukla üreticiler her bir yonga için birtakım fazladan bellek hücreleri üretirler. Sonunda, üretilen her bir bellek hücresinin sınanması ve hatalı olan hücrelerin devreden çıkarılması gerekir. VLSI teknolojisinin bir yongaya yerleştirilmesini mümkün kıldığı hücre sayısı düşünülürse, tek bir yongada bile, her bir hücrenin insan emeği tarafından sınanamayacağı anlaşılır.

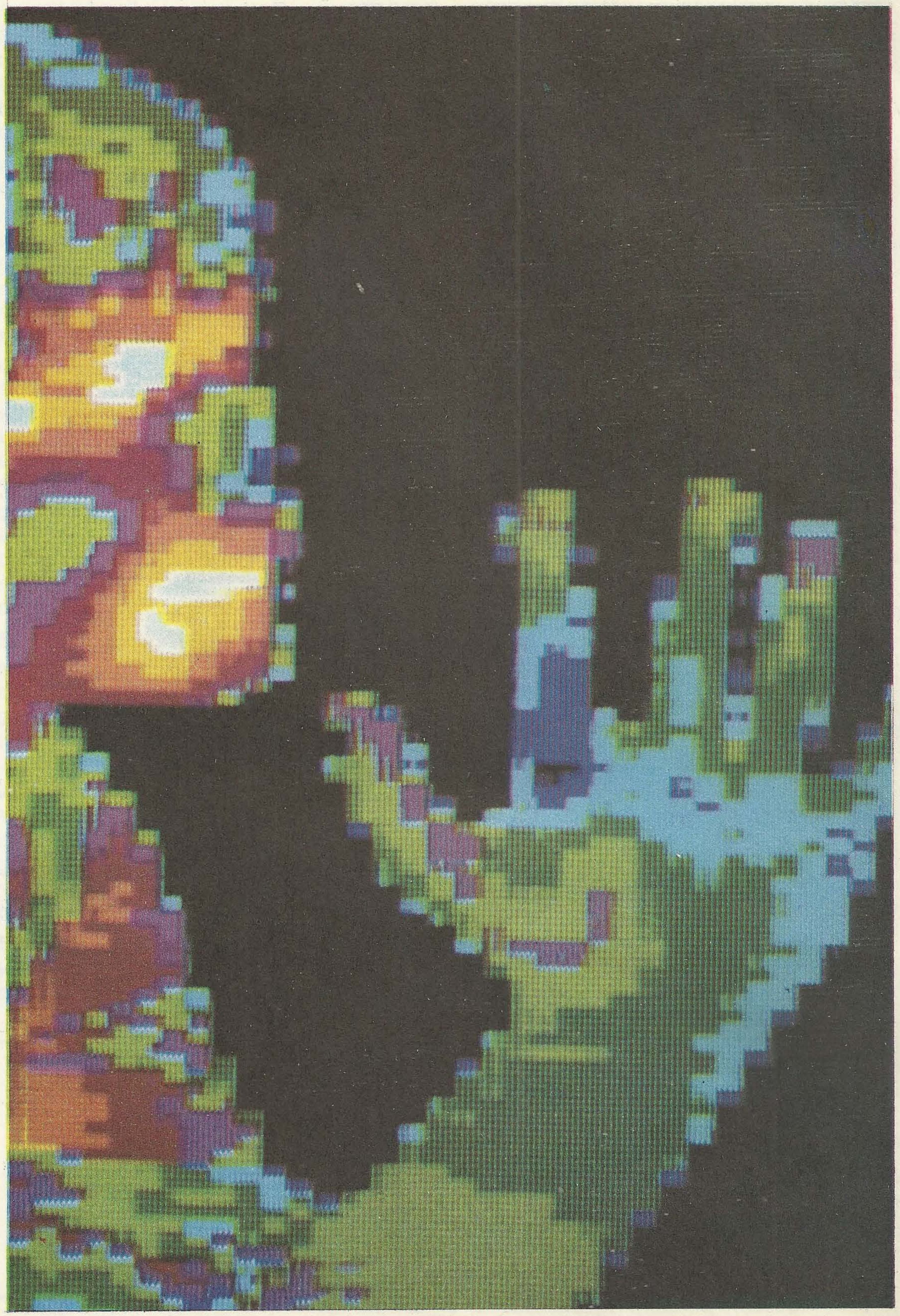
Bu teknolojik gelişmenin çeşitli iktisadi sonuçları arasında, Amerikan firmalarının üçüncü dünya ülkelerindeki ucuz işgücüne ulaşmaktaki kolaylıklarının kendileri için artık bir avantaj teşkil etmemeye başlaması sayılabilir. (Bu gelişmelerin üçüncü dünya ülkeleri üzerindeki daha doğrudan etkilerine daha sonra değineceğiz.) Ayrıca Japonların robot uygulamaları alanındaki üstünlüğü, otomatik yonga üretiminde kendilerine ek bir avantaj da sağlıyor.

## Firmalar arası İşbölümü ve Beşinci Kuşak Tasarısı

Bilgi işlem alanındaki Amerikan firmalarının belli alanlarda (yonga ya da bilgisayar üretimi) uzmanlaşarak, farklı ve bağımsız firmalar tarafından üretilen destek sanayii ürünlerini girdi olarak kullandıklarını ve kendi alanlarında uzmanlaşmış diğer firmalarla rekabet ettiklerini görmüştük. Oysa Hitachi, Fujitsu ve Nippon Elektrik gibi Japon bilgisayar üreticileri, aynı zamanda robot, yonga, tümleşik devre üreticileri de barındıran, **kieratsu** adlı geniş birer şirketler ailesine mensupturlar. Böylelikle, firmalar arasındaki bu yakın iletişim ve işbölümü sayesinde herhangi bir alandaki yenilikten (örneğin tümleşik devre üretimindeki VLSI teknolojisinden) diğer alanlarda da (örneğin süper bilgisayar üretimin de) yararlanılması çok daha kolay ve hızlı gerçekleşebilmektedir.

Gerçekten de Beşinci Kuşak Tasarısı da bu tür bir firmalar arası işbirliği tarafından yürütülmekte. 1982 Nisanı'nda MITI'nin (Ministry of International Trade and Industry-Uluslararası Ticaret ve Sanayii Bakanlığı) desteğiyle sekiz Japon firması (Fujitsu, Hitachi, Nippon Elektrik -NEC-, Mitsubishi, Oki, Sharp ve Toshiba) araştırmacı ve fonlarının bir kısmını ayırarak **ICOT** (Institute for New Generation Technology-Yeni Kuşak Bilgisayar Teknolojisi Enstitüsü) adlı bir enstitü kurdular. Bu enstitünün kadrosu genel müdür hariç hepsi de 35 yaşın altında olan 40 araştırmacıdan oluşuyordu. Önümüzdeki yıllarda bu kadronun 200 kişiye ulaşacağı bekleniyor.





1852

# Mecazi İfadelerin Önemi (3)

## *Mecaz Korkusu*

*Varolabilmek için nesnellik ve öznelliğin birbirlerine ihtiyaçları vardır. Herbiri kendini diğerinin karşısına alarak tanımlar ve diğerini düşman olarak görür. Öznellik müttefiklerini duygular, sezgisel öngörü, imgelem, insancıllık, sanat ve 'daha yüksek'' doğrular olarak seçer. Nesnellik mütefik olarak bilimsel gerçeği, rasyonelliği, kesinliği, hakkaniyeti ve yansızlığı seçer. Her ikisi de kendi alanının hakimidir ve kendi alanını daha iyi olarak görür. Birlikte, ama ayrı alanlarda varolurlar. Hepimizin yaşamında nesnel olmanın uygun düştüğü bir alanla öznel olmanın uygun düştüğü bir alan vardır. Yaşamlarımızın nesnellik ve öznellik tarafından hükmedilen kısımları insandan insana ve kültürden kültüre büyük farklılık gösterir. O kadar ki, kimi insanlar yaşamlarının tümünün bu mitlerden birinin ya da diğerinin egemenliğine girmesine izin verebilirler.*

*Yine de genel olarak Batı kültürü içersinde, bilimi, hukuğu, siyaseti, basın yayını, ahlağı, işletmeleri ve genel olarak iktisadı ve her türlü akademik araştırmayı yönlendirme iddiasında olan nesnelcilik, bu mitlerden daha önde gelenidir. Ama bütün bunlara karşın, bu yazı boyunca bizim iddiamız, nesnelciliğin bir mit olduğu oldu.*

*Yunanlıların zamanından beri Batı kültürü içersinde doğruluk ve hakikatle, sanat arasında gergin bir ilişki olmuştur. Bir yanılsama olarak görülen sanat, tiyatro ve şiir ile ilişkisinden ötürü hitabete yakın görülmüştür. Platon şiir ve hitabete (retoriğe) kuşkuyla baktı ve her ikisini de, kendi başlarına bir hakikatı iletmedikleri, duyguları ayaklandırdıkları ve bu yoldan insanları gerçek hakikate karşı körleştirdikleri gerekçesiyle kendi ütopik Devlet'inde yasakladı. Fakat Platon'un kendisi de her hatip gibi- hakikatın mutlak olduğu, sanatınsa bir yanılsamadan ibaret olduğu yolundaki görüşlerini güçlü bir mecaz aracılığıyla, Mağara Meseli ile ifade etti. O günden bugüne, hakikatın mutlak olduğu yolundaki görüşünü, usta ve incelikle bir biçimde dile getiren mecazlar Batı felsefesi geleneğinden ifade edildi durdu. Diğer yandan Aristoteles ise şiirde olumlu bir değer gördü: ''Şiirsel biçimleri ustaca kullana-*

*bilmek gerçekten de büyük bir şeydir... Ama en büyüğü mecazın ustası olabilmektir'' (Poetika, 1459'a). ''...sıradan sözler sadece, zaten bilmekte olduklarımızı iletirler: yeni bir şeyse ancak mecaz aracılığıyla kavrayabiliriz'' (Retorik, 1410 b).*

*Her ne kadar Aristoteles'in mecaz hakkındaki görüşleri, eleştirmekte olduğumuz klasik görüşlerin bir versiyonuysa da, mecazı yeni içgörü ve sezgiler uyandırabildiği için övmesi, modern felsefi düşünce içersinde yankısını bulmadı. Ampirik bilimlerin gelişerek hakikate ulaşmanın örnek tarzı olarak kavranmasıyla, bir kez daha mecaz ve diğer söz sanatları küçümsenmeye başlandı. Örneğin Hobbeş, mecazları saçma ve yanıltıcı bir biçimde duygusal olmakla suçlamaktadır: ''...onlar üzerinde düşünmek sayısız saçmalıklar arasında gezinmektir ve hepsinin hedefi, çatışma, boş iddialar ve nefrettir'' (Leviathan, I. bölüm, 5. kısım). Hobbes ''doğru kelimelerdense mecazı ve diğer söz sanatlarını kullanmayı saçma'' buluyor. ''hernekadar konuşurken, 'yol oraya ya da şuraya gidiyor' demek ya da 'atasözü şöyle der' demek meşruysa da, yollar bir yere gitmez, atasözleri bir şey söylemez. Dolayısıyla, hesap yaparken ya da hakikatı ararken bu konuşma tarzlarına başvurmak gerekir'' (a.g.e.).*

*Aynı ampirisist geleneği sürdüren Locke da, boş sözlerin aleti ve hakikatın düşmanı olarak gördüğü söz sanatlarına karşı Hobbes'in duyduğu nefreti paylaşıyor:*

*''...her şeyi adlı adıyla söyleyecek olursak, retoriğin düzen ve açık seçiklik dışındaki bütün ilkelerinin, belagatin icat etmiş olduğu bütün yapay ve süslü söz söyleme tarzlarının tek işlevlerinin yanlış fikirler uyandırmak, tutkuları harekete geçirmek ve böylelikle insanın yargı yeteneğini yanlış yöne yöneltmek olduğuna karar vermek zorundayız. Bütün sahtekarların da yaptığı budur. Dolayısıyla, belagat bunlara münazaralarda ve popüler söylevlerde ne kadar övülesi ve hayran olunası bir kılık verirse versin, eğitmek ya da öğretmek amacını güden bütün konuşmalarda bunlardan kaçınmak gerekir.*

*Ampirisist gelenekte mecaz ve retorikten korkulması, öznellikten -duygu ve imgelemden- korkulmasıdır. Sözcüklerin, doğruları ifade eden ''esas anlamları''nın olduğu düşünülür. Sözcükleri mecazi olarak kullanmak, esas anlamdan uzaklaşmak, imgelemi ve böylece duyguları karıştırmak, böylelikle de gerçekten uzaklaşıp yanılsamaya doğru yönelmek anlamına gelir. Ampirisistin mecaza ilişkin güvensizlik ve korkusu Samuel Parker tarafından mükemmelen özetlenmiştir.*

*''Felsefedeki bütün o salt mecazi terimlerle ifade edilmiş teoriler gerçek doğrular değil, yalnızca (oyuncak bebekler gibi) birkaç boş sözcükle süslenmiş imgelem ürünleridir... Bu nedenle Aklın yatağına tırmanan şehvet ve lüks düşkünü hayalleri, onu iffetsiz ve yasadışı kucaklamalarla kirletmekle kalmaz, zihni şeylere ilişkin gerçek kavram ve bilgiler yerine rüzgarlı fantazilere gebe bırakırlar.* **[Free and Impartial Censure of the Platonick Philosophy** *(1666)]*

*Bilim teknoloji yoluyla güç kazandıkça ve Endüstri Devrimi insanı insanlıktan çıkartan bir gerçeklik oldukça şairler, sanatçılar ve bazı filozoflar arasında tepki gelişti: Romantik gelenek.. Romantik gelenek. Wordsworth ve Coleridge seve seve akıl, bilim ve nesnelliği insanlıktan çıkmış ampirisistlere bırakarak imgelemi, daha yüksek doğrulara ulaşmanın insani yolu, duyguyu da kendini idrakin doğal kılavuzu olarak yücelttiler. Bilim, akıl ve teknoloji insanı kendisinden ve doğal çevresinden yalıtmıştı ya da Romantikler buna inanıyorlardı, şiiri, sanatı ve doğaya dönüşü insanın kaybolmuş insanlığını yeniden kazanmasının yolu olarak görüyorlardı. Sanat ve şiir, aklın ürünü olarak değil, ''güçlü duyguların ani taşması'' olarak görülüyordu. Bu Romantik görüşün sonucu, sanatçı ve şairin toplumun genel akışına yabancılaşmalarıydı.*

*Romantik gelenek öznelliğe sarılarak bir yanda doğru ve akıl, öte yanda sanat ve imgelem bulunan çatallaşmayı güçlendirdiler. Rasyonellikten vazgeçerek gücü artmış olan nesnellik mitinin ekmeğine yağ sürdüler. Ancak Romantikler kendilerine öznelliğin hüküm sürdüğü bir alan yarattılar. Bu, nesnelliğin alanıyla karşılaştırıldığında yoksullaştırılmış bir alandı.*

*Toplumumuzdaki gerçek güç açısından bakıldığında -bilim, hukuk, hükümet, iş dünyası ve gündelik ortamda- nesnellik miti hüküm sürmektedir. Öznellik kendisine sanat ve belki dinde bir yer kazanmıştır. Bu kültürdeki insanların çoğuysa özelliği nesnellik alanının olsa olsa bir eki olarak ve duygu ve imgeleme bir geri çekiliş olarak görürler.*

## Üçüncü Seçenek: Yaşantıcı Bir Sentez

*İdrak ve doğrunun yaşantıcı* ***(experientialist)*** *değerlendirilmesinde önerdiğimiz, öznellik ve nesnelliğin bizim tek seçeneklerimiz olduklarını yadsıyan bir alternatif. Mutlak ve koşulsuz bir doğru bulunduğuna ilişkin nesnelci görüşü, doğrunun yalnızca imgelem yoluyla, dış koşullarca kısıtlanmadan elde edilebileceğine ilişkin öznelci görüşü benimsemeksizin reddediyoruz. Mecaz üzerinde bu kadar durmamızın nedeni, akıl ve imgelemi birleştirmesidir. Akıl en azından kategorileştirme, gerektirme ve çıkarsama içerir. İmgelem, birçok yönünden birinde, bir tür şeyi başka tür bir şeyin terimleriyle görmeyi içerir - mecazi düşünce adını verdiğimiz şey. Bu nedenle mecaz* ***imgesel rasyonellik****'tir. Gündelik düşüncemizin kategorileri büyük ölçüde mecazi olduğundan ve gündelik akıl yürütmelerimiz mecazi gerektirme ve çıkarsamalar içerdiğinden ötürü olağan rasyonellik de kullanmaktadırlar.*

*Yaşantıcı yaklaşım bizim, tarafsızlık ve adil ve nesnel olma imkânına ilişkin nesnellik ve öznellik mitleri arasındaki uçurumu da kapatabilmemizi sağlar. Mitlerin önerdiği iki seçenek bir yanda mutlak nesnellik, öte yanda salt öznel sezgidir. Doğrunun anlamayla bağlantılı olduğunu gördük, bu nedenle dünya hakkında mutlak nesnel doğrular elde edebileceğimiz mutlak bir bakış açısı yoktur. Bu, doğruların bulunmadığı anlamına gelmez; yalnızca doğrunun bizim kavramsal sistemimizle bağlantılı olduğunu belirtir, kavramsal sistemimiz ise fiziksel ve kültürel çevremiz ve diğer insanlarla günlük etkileşimimizde, bizim ve diğerlerinin yaşadığı yaşantılarda temellenmiştir ve burada sürekli sınanır.*

*Mutlak bir nesnellik bulunmamakla birlikte, bir kültürün kavramsal sistemine bağlı bir tür nesnellik olabilir. Toplumsal meselelerde tarafsızlık ve adalet noktası, ilgili* ***bireysel*** *temayüllerin üzerine çıkmaktır. Bilimsel deneylerde nesnellik noktası,* ***bireysel*** *yanılsama ve hataların etkisini hesaba katmaktır. Bu, bireysel temayüllerin ötesine geçip bir kavramsal sistem ya da kültürel değerler dizisine bağlı tam bir nesnellik elde etmenin her zaman ya da herhangi bir zaman mümkün olabileceği demek değildir. Yalnızca salt öznel sezginin tek başvuracağımız kaynak olmadığını söylemektir. Belli bir kültürdeki kavram ve değerlerin o kültürdeki nihai adalet hakemi olduğu*

anlamına da gelmez. Her zaman o kültürdeki adalet standardında çok farklı kültürler üstü kavram ve değerler olabilir, olur da. Örneğin Nazi Almanyası'nda adil olan, dünya kamuoyunun gözünde adil değildi. Bu ülkede de, çelişen değerlere ait altkültürler arasında adalet meselesini içeren çatışmalara rastlanır sık sık. Burada kültünün çoğunluğu adaleti **kendi** değerlerine ilişkin tanımlar, ancak bu genel kültürel değerler akışı zaman içinde değişir ve genellikle başka kültürlerin eleştirisine maruz kalırlar.

Nesnellik ve öznellik mitlerinin atladıkları nokta, dünyayı onunla **etkileşimimiz** yoluyla **anladığımız**'dır. Nesnelliğin atladığı, anlamanın, dolayısıyla doğrunun, zorunlu olarak kültürel kavram sistemlerimize bağımlı olduğu, dolayısıyla mutlak ya da yansız bir kavramsal sistemle çerçevelenemeyeceğidir. Öznelliğin atladığı ise, anlayışımızın, en imgeci anlayışımız dahil, bizim fiziksel ve kültürel ortamımızda başarıyla işlev görmemizde temellenmiş bir kavramsal sistem tarafından sağlandığıdır. Ayrıca mecazi anlamanın mecazi gerektirirliği içerdiğini, bunun da rasyonelliğin imgesel biçimi olduğunu gözden kaçırır.

## Batı Felsefesi ve Dilbiliminde Nesnelcilik Miti

Nesnelcilik miti, Batı kültüründe, özellikle de Batı felsefesinde, Sokrates öncesi filozoflardan bugüne kadar, egemen olmuştur. Mutlak ve koşulsuz doğruluğa ulaşılabileceği görüşü, Batı felsefesi geleneğinin temel taşıdır. Nesnelcilik miti, hem akılcı, hem de ampirisist geleneklerde başköşeyi almıştır. Bu bakımdan, iki geleneğin ayrıldığı tek nokta, o mutlak doğruluklara nasıl erişileceği konusundadır. Akılcılara göre, sadece doğuştan gelen akıl yetimiz bize nesnelerin gerçekte nasıl oldukları hakkında bilgi verebilir. Ampirisistlere göre de, dünya üzerine bütün bilgimiz, (doğrudan ya da dolaylı olarak) duyu algılarımızdan doğar ve duyum (sensation) öğeleriyle kurulur. Kant'ın akılcılık ile ampirisizmi birleştiren sentezi de, kendilerinde olduğu gibi, herhangi bir nesnenin bilgisi olamayacağı iddiasına karşın, nesnelci gelenek içinde yer alır. Kant'ı nesnelci kılan şey, bütün insanların duyularıyla nesnelerin türlerini yaşantılayabileceğini (deneysel miras); ayrıca, evrensel aklını, kullanarak, evrensel geçerliliği olan bilgi ve ahlâk yasalarına sahip olabileceğini (akılcı miras) ileri sürmesidir. Nesnelci gelenek Batı felsefesinde, mantıkî pozitivistleri izleyenlerde, Frege'ci gelenekte, dilbilimde, Chomsky geleneğinden türeyen yeni-akılcılıkta, bugüne kadar süregelmiştir.

Bizim mecaz yaklaşımımız, bu geleneğe ters düşer. Mecazı, insan kavrayışı için vazgeçilmez bir öğe, hayatımızda yeni anlamlar ve yeni gerçeklikler yaratmak için bir mekanizma olarak görüyoruz. Bu da, mecazı öznelciliğin bir kolu ve mutlak doğruluk arayışını engelleyici bir güç olarak gören, Batı felsefe geleneğinin

*büyük bir bölümünden bizi ayırıyor. Ayrıca, geleneksel mecazın, kavramsal sistemimizi tamamıyla kapladığı ve ilk sırada gelen bir anlama mekanizması olduğu yolundaki görüşlerimiz de, yakın zamanlarda İngiliz-Amerikan analitik felsefesinde egemen olan ve modern dilbilimin büyük bir bölümüyle, diğer disiplinlerde, sorgulanmadan üstlenilen, dil, anlam, doğruluk ve anlama üzerine çağdaş görüşlerden de ayırıyor bizi. Dil, anlam, doğruluk ve anlama üzerine bu varsayımların, örnek oluşturacak bir listesini çıkarttık. Bütün nesnelci filozof ve dilbilimciler, bu varsayımların hepsini kabul etmeseler de, en azından önde gelenleri, bunlardan çoğunu kabul ediyor.*

- *Doğruluk, kelimelerin dünyaya denk düşürülmesidir.*
- *Doğal dil için üretilen anlam teorisi, insanların dili nasıl anladıklarından ve kullandıklarından bağımsız, bir doğruluk teorisine dayandırılır.*
- *Anlam, insanın kavrayışından bağımsız ve nesneldir; bir duruma bağlı değildir.*
- *Cümleler, içkin yapıları olan, soyut nesnelerdir.*
- *Bir cümlenin anlamı, parçalarının anlamından ve cümlenin yapısından çıkarılabilir.*
- *İletişim, konuşanın, sabit bir anlamı olan bir mesajı dinleyene aktarmasıdır.*
- *Bir kişinin bir cümleyi nasıl anladığı, cümlenin **onun için** ne anlama geldiği, cümlenin nesnel anlamı ile kişinin dünya ve cümlenin dile getirildiği bağlam üzerine düşündüklerinin bir işlevidir.*

*Bizim, geleneksel mecaz yaklaşımımız, bütün bu varsayımlarla çelişir. Bir cümlenin anlamı, kavramsal bir yapıya göre, belirlenir. Gördüğümüz gibi, doğal bir dilin kavramsal yapısının büyük bir kısmı, doğası gereği, mecazidir.*

*Kavramsal yapı da, geleneksel mecazlar gibi, fizikî ve kültürel yaşantıya dayanır. Bu yüzden, anlamın hiçbir zaman bir duruma bağlı olmadığı ya da nesnel olduğu söylenemez; her zaman kavramsal bir sistemin kurulmasına ve kullanımına dayanır. Ayrıca, doğruluk da hep kavramsal bir sisteme ve onu yapılayan mecazlara göre, belirlenir. Bu yüzden, doğruluk mutlak ya da nesnel değildir; anlamaya dayanır. Dolayısıyla cümlelerin de, içkin, nesnel olarak verili anlamları yoktur. İletişim, sade-*

ce bu tür anlamların aktarımı olamaz.

Bizim bu konulardaki yaklaşımımızın, standart felsefi ve dilbilimsel konumlardan neden bu kadar farklı olduğu, o kadar da açık değil. Göründüğü kadarıyla bunun temel nedeni, bütün standart konumlar nesnelcilik mitine dayanırken, bizim mecaz yaklaşımımızın, bu mitle uyuşmaz oluşudur. Bu kadar temel konularda hakim olan teorilerden, bu kadar radikal bir biçimde sapma, tabiî ki açıklama gerektirir. Bir mecaz yaklaşımı için, doğruluk anlam ve anlama üzerine, Batı felsefe geleneği içindeki hakim eğilimlerden doğan temel varsayımları sorgulamak nasıl mümkün olabilir? Bu soruya verilecek cevap, dil, doğruluk ve anlamı üzerine, nesnelci varsayımların burada şimdiye kadar anlattıklarımızdan daha ayrıntılı bir sunumunu gerektirmektedir. Daha ayrıntılı olarak, (a) nesnelci varsayımların neler olduğunu, (b) nelere dayandıklarını ve (c) dil, doğruluk ve anlam konusunda genel bir yaklaşım için, ne gibi sonuçlar doğurduklarını ele almayı gerektirmektedir.

Bu araştırmanın amacı, sadece bizim dil üzerine görüşlerimizin, standart görüşlerden farkını ortaya koymak değil; aynı zamanda, nesnelcilik mitinin, Batı kültüründe, genellikle farkında olmadığımız biçimlerde, nasıl etkili olduğunu örnekleriyle göstermektir. Daha da önemlisi, kültürümüzün pek çok sorunlu alanının, nesnelcilik mitini körükörüne kabul etmekten kaynaklanabileceğini ve radikal öznelciliğe mahkûm olmamızı gerektirmeyecek başka bir alternatif olduğunu söylemek istiyoruz.

## Standart Anlam Teorileri, Nesnelcilik Mitine Nasıl Bağlıdır

Nesnelci geleneğin temeli olan nesnelcilik mitinin, bir anlam teorisi için oldukça özgül sonuçları vardır. Bu sonuçların neler olduğunu, nesnelcilik mitinden nasıl çıktıklarını ve yaşantıya dayalı **(experiantialist)** bir görüşten yola çıkılınca, neden kabul edilemez hale geldiklerini göstermek istiyoruz. Bütün nesnelcilerin, -aşağıda anlatacağımız konumların hepsini birden kabul etmeseler de- şu ya da bu biçimde, bunlardan çoğunu üstlendiklerini söylemiştik.

### Anlam Nesneldir

Nesnelci birisi, anlamı yalnızca nesnel doğruluk ya da yanlışlık koşullarına göre nitelendirir. Nesnelci görüşe göre, dildeki uzlaşımlar, her cümleye, nesnel doğruluk koşullarını belirleyen bir **nesnel anlam** atfeder: Bunun için de, ''indeks öğeleri'' denilen, bağlamın belli öğelerinin verilmesi gerekir: Konuşan kimdir, dinleyen kimdir, ne zaman, nerede söylenmiştir, ''bu'', ''şu'' gibi kelimelerle atıfta bulunulan nesneler nelerdir vs. Yani, bir cümlenin nesnel anlamı, herhangi bir verili kişinin cümleyi nasıl anladığına ya da hepten anlayıp anlamadığına bağlı değildir. Örneğin bir papağan, söylediklerinin Türkçe'de ne anlama geldiğini hiç anlamadan, ''yağmur yağıyor'' demesi için eğitilebilir. Ama ister bir papağan, isterse bir insan, ister bir bilgisayar söylesin, cümlenin değişmeyen nesnel bir anlamı vardır ve eğer yağmur yağıyorsa doğru, yağmı-

*yorsa yanlış bir cümle olacaktır. Anlam sorununa nesnelci yaklaşımı esas aldığımızda, bir kişi ancak hangi koşullar altında bir cümlenin doğru ya da yanlış olduğunu anlarsa, cümlenin nesnel anlamını da anlayabilecektir.*

*Nesnelci birisi, sadece nesnel doğruluk ve yanlışlığın koşullarının varolduğunu değil, aynı zamanda insanların bunları elde edebilecek-lerini de varsayar. Bunun gayet açık olduğunu düşünür. Çevrenize bir bakın. Eğer yerde bir kurşun kalem varsa, o zaman ''yerde bir kurşun kalem var'' cümlesi doğrudur. Eğer bu dili konuşuyorsanız ve yerdeki kalemi algılayabiliyorsanız, cümleyi de gayet tabii doğru olarak kabul edeceksiniz. Böylesi cümlelerin nesnel olarak doğru ya da yanlış oldukları, bu türden pek çok doğruya erişebileceğiniz varsayılmaktadır. Bir cümlenin nesnel olarak hangi koşullar altında doğru olabileceğini, insanlar anlayabileceklerine göre, bir dilin, cümlelere nesnel anlamlar atfedebilmesi için, uzlaşımları olabilir. Yani, nesnelci görüşe göre, bir dilin, cümlelerle nesnel anlamları eşlemek için sahip olduğu uzlaşımlar, o dili konuşan, belli bir nesnel anlamı olan bir cümleyi anlayabilecek kişilerin elindedir. Nesnelci birisi, bir cümlenin (düz) anlamını anlamaktan söz ettiğinde, bir cümleyi neyin nesnel olarak doğru ya da yanlış kıldığını anlamaktan söz etmektedir aslında. Genel olarak nesnelci anlama, hangi koşullarda doğruluğun ya da yanlışlığın sözkonusu olduğunu anlamakla sınırlıdır.*

*Bizim ''anlama''dan kasdettiğimiz ise bu* ***değil****'dir. Nesnelci birisinin, anlamı, anlamadan bağımsız olarak ele aldığını söylersek, ''anlama''yı onun değil, bizim kullandığımız anlamda almış oluruz.*

## Anlam Bir Duruma Bağlı Değildir

*Nesnelci görüşe göre, nesnel anlam birisine* ***yönelik*** *anlam değildir. Doğal bir dildeki ifadelerin, (ancak anlamları, insanların konuşarak ya da bir eylemde bulunarak, yaptıklarından bağımsız olduğunda), nesnel bir anlam taşıdıkları söylenebilir. Yani, anlam bir duruma bağlı olmamalıdır. Örneğin Frege, bir işaretin nesnel işlemi (meaning) olarak ''anlam''ı (Alm.* ***Sinn,*** *İng.* ***Sense****), ''fikir''* ***(idea)****den ayırır. Fikir,*

> *Sahip olduğum anılardan, duyu izlenimlerinden, yaptığım içsel ve dışsal edimlerden [doğar]... Fikir, özneldir... Bunun ışığında,*

> ***belirli*** *bir anlamdan söz etmek için, tereddüt etmeye gerek yok. Ama fikir sözkonusu olunca, kesinlikle kime ve hangi zamana ait olduğunu da eklemek gerekir. [Frege, 1966, s.59-60]*

*Frege'nin ''anlam''ı, nesnel ve bir duruma bağlı olmayan bir işlemdir. Bir dildeki her ifade, bir duruma bağlı olmayan, kendisine eşlik eden bir işleme sahiptir.*

*Frege'ci gelenek Richard Montague'yi izleyenlerle diğer başkalarının çalışmaları sayesinde bugüne kadar geldi. Anlambilim (semantics) üzerine yapılan bu çalışmalardan hiçbirinde, bir cümlenin anlamı, bir insanın o cümleyi nasıl anlayacağına bağlı kılınmaz. Montague şöyle diyor: ''Donald Davidson gibi, bir doğruluk teorisi -keyfi bir yoruma dayanarak, daha genel bir doğruluk kavramı oluşturmayı, ciddi bir sözdizimi ve anlambilimin ana hedefi olarak görüyorum'' (1974, s.188) ''Keyfi yorum'', buradaki önemli kelimelerdir. Montague, anlam ve doğruluk teorilerinin sadece matematik düzenlemeler olduğunu varsayıyordu. Amacı, insanların bulaşmadığı, özellikle insan psikolojisiyle ya da insan kavrayışıyla lekelenmemiş bir ''keyfi yorum''a ulaşmaktı. Çalışmasının, evrendeki her türlü canlıya uygulanabilir olmasını ve herhangi bir canlı tarafından getirilecek herhangi bir sınırlamadan da bağımsız olmasını istiyordu.*

## *İnsanları ya da İnsan Kavrayışını Katmadan, Kelimeleri Dünya ile Denkleştirme*

*Nesnelci gelenek, anlambilimi, insan kavrayışının müdahalesi olmadan, dilsel ifadelerin doğrudan dünyaya nasıl denk düştüğünün incelenmesi olarak görür. Belki de, bu konumu en açık biçimde, David Lewis dile getirmiştir:*

> *''Önerilerim, bir anlamı çözümlerken, hemen dili kullananın psikolojisine, sosyolojisine (niyetlere, duyu-deneyine, zihinsel fikirlere ya da toplumsal kurallara, uzlaşımlara, kurallara) başvuranların beklentilerini karşılamayacaktır. İki konuyu birbirinden ayırıyorum: Birincisi, soyut anlam sistemleri olarak, bütün mümkün dillerin ya da gramerlerin betimlenmesi. Burada, sembollerin dünyanın görünümleriyle örtüşmesi önemlidir. İkincisi, psikolojik ve sosyolojik olgulann betimlenmesi. Burada, soyut anlam sistemleri içinde, bir kişi ya da halk tarafından kullanılanı önem taşır. Bu iki konuyu birbirine karıştırmak, sadece karmaşa doğurur.''*

*Lewis burada dilin dünyaya nasıl denk düşebileceğini göstermeye çalışırken, Montague'nin tecrübesinden yararlanır. ''Semboller, dünyanın görünümleriyle nasıl bağdaşırlar?'' sorusu, insanların dili nasıl kullandıkları ve nasıl anladıkları üzerine, akla gelebilecek her psikolojik ya da sosyolojik olguyu eleyecek kadar genel ve keyfî tutulmuştur.*

## Doğruluk Kuramına Dayanan Bir Anlam Teorisi

*Herhangi bir insan kavrayışından bağımsız, nesnel bir doğruluk olasılığı, nesnel bir anlam teorisini de mümkün kılar. Nesnelci doğruluk görüşüne göre, kendi başına bir cümle dünyaya denk düşebilir ya da düşmeyebilir. Eğer denk düşerse doğrudur; düşmezse, yanlış.*

*Bu da, doğruluğa dayalı, nesnelci bir anlam yaklaşımını doğurur. bunu en açık biçimde ifade eden, yine David Lewis'dir: ''Bir cümle için anlam, cümlenin hangi koşullar altında doğru ya da yanlış olduğunu belirleyen bir şeydir.''*

*Bu görüş sonradan, emir ve vaadler gibi, edimsel (performative) cümlelere de anlam yüklemek için, genelleştirilmiştir. Bu teknik, ''dünyaya denk düşme''ye göre, doğruluğun tanımını yapar. ''Dünyaya denk düşme'' de, teknik olarak, bir model içinde tatmin edici koşulların sağlanmasıyla tanımlanır. Konuşma edimlerinin* ***(speech-act)*** *uygunluk koşulları da, aynı biçimde, tatmin edici koşullar ya da ''dünyaya denk düşme''ye göre tanımlanır. Aşağıda ''doğruluk'' ve ''yanlışlık''tan söz ettiğimizde, tatmin edici koşulları kasdettiğimiz, önermelere, konuşma edimlerini de dahil ettiğimiz, anlaşılmalıdır.*

## Anlam, Kullanımdan Bağımsızdır

*Nesnelci doğruluk yaklaşımı, anlamın da nesnel olmasını ister. Eğer anlam nesnel olacaksa, bütün öznel öğeleri, yani belli bir bağlam'a, kültüre ya da anlama biçimine özgü her şeyi dışta bırakmalıdır. Donald Davidson'un dediği gibi: ''Kelimeler ve cümlelere, özel kullanım bağlamlarından ayrı, düz anlam ve doğruluk koşulları atfedilebilir.''*

## Anlam Düzenlemeye Bağlıdır- Blok Teorisi

*Nesnelcilik mitine göre, dünya nesnelerden oluşur; nesnelerse onları yaşantılayan herhangi bir canlıdan bağımsız olarak, kesin tanımlanmış birtakım içkin nitelikler taşırlar. Zamanın belli bir noktasında, kendi aralarında sabit ilişkiler kurarlar. Nesnelcilik miti, bu yönleriyle, anlam konusunda bir blok teorisi doğurmuştur. Eğer dünya kesin-tanımlanmış*

nesnelerden oluşuyorsa, onlara bir dil içinde adlar verebiliriz. Eğer nesnelerin, kesin-tanımlanmış, içkin nitelikleri varsa, bu niteliklerden her birine tekabül edecek tek gözenekli yüklemleri olan bir dil olabilir. Ve eğer nesneler (en azından verili bir zamanda), birbirleriyle sabit ilişkiler içindelerse, bu ilişkilerden her birine tekabül edecek çok-gözenekli yüklemleri olan bir dil olabilir.

Dünyanın bu biçimde olduğunu, bizim de böyle bir dile sahip olduğumuzu varsayarsak, bu dilin sözdizimini kullanarak, dünyadaki herhangi bir duruma doğrudan tekabül edecek cümleler kurabiliriz. Cümlenin doğruluk koşulları yani cümlenin hangi koşullar altında, belli bir duruma denk düşeceği, bütün cümlenin anlamı olacaktır. Bu da, tamamen parçalarının anlamlarına ve birbirlerine nasıl uyduklarına bağlı olacaktır. Parçaların anlamları, hangi adların hangi nesneleri, hangi yüklemlerin hangi nitelikleri ya da ilişkileri seçebileceğini gösterecektir.

Nesnelci anlam teorilerinin hepsi, doğaları gereği, düzenlemeye bağlıdır; yani hepsi blok teorileridir, böyle olmak zorundadırlar. Bunun nedeni de, dünyanın zaten nesnelciler için bloklardan, tanımlanabilir nesnelerden ve açıkça görülen içkin niteliklerle ilişkilerden oluşmasıdır. Dahası, dildeki her cümle, bütün gerekli blokları içermelidir ki, sözdizimi ile birlikte, cümlenin doğruluk koşullarını sağlamak için, başka bir şeye ihtiyaç duyulmasın. Bu dışta bırakılan ''başka bir şey'' de, her türlü insan kavrayışıdır.

## Nesnelcilik İnsan Kavrayışını Dışta Bırakarak Ontolojik Göreceliğe İzin Verir

Mantıki pozitivistler (örneğin Carnap) yukarda sözü edilen blok niteliklerine ve buraya kadar tartıştığımız bütün diğer özelliklere sahip, evrensel olarak uygulanabilir, formel bir dil kurma yolunda, nesnelci bir program uygulamaya çalıştılar. Richard Montaque, doğal dilleri, evrensel olarak uygulanabilir formel bir dil içine yerleştirecek ''evrensel bir gramer'' oluşturduğunu ileri sürdü.

Bu evrenselci iddialara karşı çıkan Quine, her dilin kendi içinde kurulu bir ontolojisi olduğunu ve nesne, nitelik ya da ilişki sayılan şeyin, dilden dile değişebileceğini ortaya attı. Bu konum, ''ontolojik görecelik'' tezi olarak bilinir.

Nesnelci programın sınırları içinde kalarak, insan kavrayışına ya da kültürel farklılığa başvurmadan, ontolojik görecelik tezini savunmak mümkündür. Böyle bir görecelik konumu, bütün doğal dillerin aynı biçimde aktarılabileceği, tek bir evrensel uygulanabilir mantıki dil kurma olasılığını bir yana bırakır. Onun yerine, her doğal dilin, dünyada çeşitli biçimlerde bulunan şeyleri ortaya çıkardığını ileri sürer. Diller, hep gerçekten orada olan nesneleri, gerçekten bulunan nitelikleri ve ilişkileri seçer. Ama değişik diller, içlerinde değişik ontolojiler taşıyabileceklerinden, herhangi iki dilin, genelde birbirlerine benzer olacaklarını kimse garanti edemez.

Nesnelci anlam yaklaşımının, görecelikçi bi-

*çimi, anlam ve doğruluk koşullarının, evrensel değil, ancak verili bir dile göre nesnel olarak varolduğunu ileri sürer. Doğruluğun nesnel olduğunu ve dünyada, içkin nitelikler taşıyan nesneler olduğunu savunduğu için, nesnelcilik mitine bağlı kalır. Ancak görecelikçi nesnelciliğe göre, her dil dünyayı farklı bir biçimde sunar; bu yüzden bir dilde söylenen doğrular, başka bir dile çevrilemeyebilir. Ama dilin bir varlık olarak seçip çıkardıkları, nesnel olarak, varlık biçiminde dünyada bulunurlar. Bu yaklaşımda (verili bir dile göre de olsa) doğruluk ve anlam, hâlâ nesneldir; insan kavrayışı, hâlâ anlam ve doğruluk için gereksiz sayılmaktadır.*

## Nesnelci Dilbilimin Öncülü: Dilsel İfadeler Birer Nesnedir

*Nesnelcilik mitine göre, nesneler kendi içlerinde, kendilerine özgü niteliklere sahiptirler; onları anlayan birilerinden bağımsız olarak, birbirleriyle ilişki içindedirler. Kelimeler ve cümleler kağıda geçirildiklerinde, gerçekten birer nesne olarak görülebilirler. Antik çağdaki başlangıcından bugüne kadar, nesnelci dilbilimin öncülü budur: Dilsel ifadeler, içlerinde, kendilerine özgü nitelikler taşıyan nesnelerdir.*

*Birbirleriyle, kendilerinden konuşan ya da anlayan herhangi bir kişiden bağımsız olarak, sabit ilişkiler içindedirler. Birer nesne olarak, parçalara sahiptirler; bloklardan yapılmışlardır: Kelimeler, kök, önek, sonek ve içekten oluşur; cümleler, kelime ve deyişlerden; konuşmalar da cümlelerden oluşur. Bir dil içinde parçalar, blok yapılarına ve içkin niteliklerine göre, birbirleriyle değişik ilişkiler içinde olabilirler. Blok yapısının, parçaların içkin niteliklerinin ve birbirleriyle ilişkilerinin incelenmesi, geleneksel olarak* ***gramer*** *olarak adlandırılagelmiştir.*

*Nesnelci dilbilim kendini, dil alanındaki tek* ***bilimsel*** *yaklaşım olarak görür. Nesneler, insanların onları anlamalarından ya da bağlamlardan bağımsız ve kendileri olarak incelenebilmelidirler. Nesnelci felsefede olduğu gibi, dilbilimde de ampirisist ve akılcı gelenekler vardır. Bloomfield, Harris ve onları izleyenlerin temsil ettiği son dönem Amerikan yapısalcılığının ampirisist geleneği, metinleri, bilimsel incelemenin tek nesnesi olarak görür. Jakobson*

gibi Avrupa yapısalcıların, Sapir, Whorf ve Chomsky gibi Amerikalı düşünürlerin temsil ettikleri akılcı gelenek ise, dilsel ifadeleri zihinsel gerçek nesneler olarak gördüler; dilin de, zihinsel bir gerçekliğe sahip olduğunu ileri sürdüler.

## Gramer, Anlam ve Kavrayıştan Bağımsızdır

Nesnelcilik mitinin nasıl, dilsel ifadeleri içkin niteliklere, bir blok yapısına ve diğer nesnelerle sabit ilişkilere sahip nesneler olarak gören bir dil görüşünü doğurduğunu yukarda gördük. Nesnelcilik mitine göre, varolan dilsel nesneler -ve onların blok yapıları, nitelikleri ve ilişkileri- insanlann onları anlamalarından bağımsızdır. Dilsel ifadelerin nesne olduğu görüşünden, gramerin de anlam ya da insan kavrayışından bağımsız olarak incelenebileceği sonucu çıkmaktadır.

Bu gelenek, gramerin anlamdan ya da insan kavrayışından bağımsız, saf bir biçim olduğunu ısrarla savunan Noam Chomsky'nin dilbiliminde özetlenir. Bu bakımdan, dilin insan kavrayışını da içeren herhangi bir yönü, Chomsky için daha başından, gramer incelemesinin dışında bırakılır. Chomsky'nin ''edim''e **(performance)** karşılık, ''yeti'' **(competence)** terimini kullanışı, dilin belli yanlarını, bilimsel dilbilimin tek meşru nesnesi olarak tanımlama çabasıdır. Chomsky'nin bilimsel dilbilimi, nesnelci dilbilimin akılcı kolu dediğimiz, yalnız saf biçimle ilgili şeyleri içeren, insan kavrayışı ve dil kullanımıyla ilgili her şeyi dışarda bırakan bir dilbilimdir. Her ne kadar dilbilimi, psikolojinin bir dalı olarak görürse de, aslında Chomsky'ye göre dilbilim **bağımsız** bir daldır; yani insanların aslında dili nasıl anladıklarına kesinlikle bağlı değildir.

## Nesnelci İletişim Teorisi: ALIŞVERİŞ Mecazı

Nesnelci dilbilim ve felsefe içinde, anlamlar ve dilsel ifadeler bağımsız olarak varolan nesnelerdir. Böyle bir görüş, **iletişim bir alışveriştir** mecazına çok benzeyen bir iletişim teorisi doğurur:

Anlamlar nesnedir.
Dilsel ifadeler nesnedir.
Dilsel ifadelerin (kendi içlerinde) anlamları vardır.
İletişimde bir konuşan bir dinleyene, anlamla örtüşen dilsel bir ifade aracılığıyla, kesin bir anlam aktarır.

*Bu açıdan bakıldığında, kastedileni nesnel olarak söylemek mümkün olmaktadır. İletişimdeki başarısızlıklar ise, ancak öznel hatalar sonucu doğar: Anlamlar, kelimeler içinde nesnel olarak doğru yerde bulunduklarından, ya siz kastettiğinizi söylemek için doğru kelimeleri kullanmadınız ya da yanlış anlaşıldınız demektir.*

## Anlamaya Nesnelci Bir Yaklaşım Nasıl Bir Şey Olurdu?

*Bir cümlenin düz, nesnel anlamını anlamaktan, yani bir cümlenin nesnel olarak hangi koşullar altında doğru ya da yanlış olacağını anlamaktan, nesnelci birinin ne kastettiğini gösterdik. Ama nesnelciler bir kişinin, verili bir bağlamda bir cümleyi, düz, nesnel anlamından başka bir anlamda anlayabileceğini de farkettiler.*

*Bu öteki anlam için, genel olarak ''konuşanın verdiği anlam'' yada ''söyleyenin verdiği anlam'' ya da ''kasdedilen'' denir. Nesneciler, tipik bir biçimde, bütün anlamayı kapsayacak bir yaklaşımın, bu durumları da içermesi gerektiği üzerinde dururlar.*

*Örneğin, alaycı olduğu açıkça belli edilen bir bağlamda söylenmiş ''o gerçek bir dahidir'' cümlesini alalım. Nesnelci yaklaşıma göre, cümlenin **nesnel** bir **anlamı** vardır. ''O gerçek bir dahidir'', onun zihinsel gücü oldukça yüksek, demektir. Ancak konuşan, cümleyi alaycı bir biçimde söyleyerek, tam karşıt anlamı, yani onun tam bir aptal olduğunu söylemek istemektedir. Burada **konuşanın verdiği anlam**, cümlenin nesnel anlamının tersidir.*

*Konuşanın verdiği anlam meselesini, alaycı bir bağlam içinde şöyle sergileyebiliriz:*

> *Konuşan, nesnel M anlamını (M = Onun zihinsel gücü oldukça yüksek) taşıyan, S cümlesini (S = ''O gerçek bir dahildir'') söyleyerek, dinleyenin mecazi M' anlamını (M' = O gerçek bir aptal) anlamasını ister.*

*Bu, birisine **yönelik** anlamının, nesnelci bir çerçevede nasıl açıklanacağını gösteriyor. (A) cümlesi, verili bir bağlamda nesnel olarak doğru ya da yanlış olabilir. Eğer (A) doğruysa, S cümlesi (''O gerçek bir dahidir''), hem konuşan, hem de dinleyen için (eğer dinleyen konuşanın niyetini sezmişse) **o gerçek bir aptaldır** anlamına gelir.*

*Konuşma-edimi teoristlerinin ortaya attığı bu teknik, birisine **yönelik** anlamı, cümlenin nesnel anlamından, yani nesnel doğruluk ya da yanlışlık koşullarından dışarda tutmanın bir yolu olarak, nesnelci geleneğe adapte edildi. Buradaki teknik oyun, nesnel M ve M' anlamlarını, kendisi de nesnel bir anlama sahip olan (S) cümlesiyle birlikte kullanmaktadır. Böylece, konuşanın verdiği anlam ile dinleyenin aldığı anlamı, yani birisine **yönelik** anlamı, birleştirmektedir. Bu yine de, konuşanın niyetinin, nesnel olarak gerçek olduğunu kabul etmeyi gerektirir ki, bunu da bazı nesnelciler pekala geri çevirebilirler.*

*Verdiğimiz örnek, M ile M''nin karşıt anlamlar, yani karşıt doğruluk koşulları taşıdıkları bir alaycılık örneğidir. Düz anlamda konuşulduğunda, M = M' olur. Nesnelci program, bunu bütün, birisine **yönelik** anlam olayları için, özellikle konuşanın bir şey söyleyip, başka bir şeyi kastettiği hallerde -abartma, küçümseme, dokundurma, ironi ve en önemlisi mecazi dilde- genel bir anlaşılır kılma tekniği olarak görür. Bu programı uygulamak, aşağıdaki soruyu cevaplayacak, birtakım genel ilkelerin saptanmasını da içerebilir:*

> *S cümlesi **düz** ve nesnel M anlamı ve bağlama ilişkin gerekli bilgi sunulsa **konuşanın verdiği M' anlamı**'nın bu bağlamda ne olacağını tahmin edebilmemizi, hangi özgül ilkeler sağlayabilir?*

*Özellikle, mecaz konusunda bu karşımıza çıkıyor. Örneğin ''Bu teori ucuz sıvadan yapılmış''ın, nesnelci bakışa göre, düz bir nesnel anlamı (M) vardır; o da yanlıştır, yani **bu teori pahalı olmayan harçla yapılmıştır** demektir.*

*Düz, nesnel anlam yanlıştır, çünkü teoriler harçla yapılan şeyler değildir. Ancak ''Bu teori ucuz sıvadan yapılmış''ın, açıklanmamış, konuşanın verdiği bir anlamı (M') olabilir; bu da doğru olabilir, yani **bu teori zayıftır**, demek olabilir. Bu durumda sorun, dinleyenin S cümlesinden (''Bu teori ucuz sıvadan yapılmıştır''), nesnel M anlamı aracılığıyla (**bu teori pahalı olmayan harçtan yapılmıştır**), söyleyenin verdiği, açıklanmayan M' anlamına **(bu teori zayıftır)** varabilmesi için, kendine nasıl genel yorum ilkeleri verilebileceği sorunudur.*

*Nesnelci biri, bütün mecazları, M ≠ M' olarak, yani dolaylı anlam durumlarıymış gibi görür.*

*Mecaz içeren bütün cümlelerin, tipik durumlarda ya göze çarpacak kadar yanlış (örneğin ''Bu teori ucuz sıvadan yapılmış'') ya da göze çarpacak kadar doğru (örneğin, ''Musolini bir hayvandı''), nesnel bir anlamları vardır. Mecazi bir cümleyi (örneğin, ''Bu teori ucuz sıvadan yapılmıştır'') anlamak her zaman, düz ve nesnel M anlamından (**teori, pahalı olmayan harçtan yapılmıştır**) farklı olan, nesnel M' anlamı (**teori zayıftır**) üzerinden, o cümleyi dolaylı olarak anlamayı kapsar.*

*Anlamaya nesnelci yaklaşım, bu yüzden hep nesnelci doğruluk yaklaşımına dayanır. Bu yaklaşım, iki türlü, -doğrudan ve dolaylı- anlamayı içerir. Doğrudan anlama, nesnel olarak doğru olabileceği koşullar altında bir cümlenin düz, nesnel anlamını anlamaktır. Dolaylı anlama, konuşan dolaylı bir anlamı aktarmak için, başka bir cümle kullanırken, nesnelci doğruluk koşullarına uygun olarak, aktarılan anlamın nerede **doğrudan anlaşılabileceğini** kestirmeyi içerir.*

*Nesnelci mecaz yaklaşımından, otomatik olarak şu dört sonuç çıkar:*

***Tanımsal** olarak, **mecazi bir kavram ya da mecazi anlam diye bir şey olamaz.** Anlamlar nesneldir ve nesnel doğruluğun koşullarını gösterir. Tanımları gereği, anlamlar dünyayı olduğu gibi ya da olabileceği gibi nitelendirme yollarıdır. Nesnel doğruluğun koşulları, bir şeye başka bir şey dolayısıyla bakma yollarını göstermez. Dolayısıyla, nesnel anlamlar, mecazi olamazlar.*

***Mecaz, bir anlam meselesi olamayacağına göre, sadece bir dil meselesi olabilir.** Nesnelci görüşe göre, bir mecaz en iyi halde, nesnel M' anlamı üzerine **konuşmak** için, bize dolaylı bir yol gösterebilir. Bunu yaparken, başka bir nesnel M anlamı üzerine, ki çoğunlukla yanlış olduğu bellidir, konuşmak için düz olarak kullanılabilen bir dili kullanır.*

***Yine tanım gereği, düz (uzlaşmaya bağlı) mecaz diye bir şey olamaz.** Bir cümle M' = M ise, yani söyleyenin verdiği anlam, nesnel anlamsa, düz olarak kullanılır. Mecazlar ancak, M' ≠ M ise doğar. Bu yüzden, nesnelci tanıma göre, düz bir mecaz, kendi içinde bir çelişkidir. Düz dil de, mecazi olamaz.*

***Mecazlar ancak, bizim nesnel benzerlikleri, nesnel M ve M' anlamlar arasındaki benzerlikleri, görmemizi sağlayarak, anlamaya katkıda bulunurlar.** Bu benzerlikler, nesnelerin ortak, paylaştıkları, gerçekten kendilerinde bulunan **içkin nitelikleri**'ne dayanır.*

*Dolayısıyla, nesnelci anlam yaklaşımı, bu yazıda savunduğumuz her şeye tamamen ters düşüyor. Anlam ve mecaza bu tür bir bakış, eski Yunanlılardan beri bizimle beraber.*

## Mecaz, Nesnellik Mitinin Sınırlılıklarını Nasıl Ortaya Çıkarır?

*Felsefedeki nesnelci geleneğin özü doğrudan nesnellik mitinden gelmektedir: Dünya, bağımsız nesnelerden oluşmuştur, bu nesnelerin içsel nitelikleri ve birbirleriyle her an sabit ilişkileri vardır. Biz, dilsel kanıtlar (özellikle mecaz) temelinde, nesnelci felsefenin yaşantımızı, düşüncemizi ve dilimizi anlama tarzımızı değerlendirmekte başarısız olduğunu öne sürüyoruz. Bizce uygun bir değerlendirme şunları içermelidir:*

- *Nesneleri yalnızca bizim dünyayla etkileşimimiz ve onu yansıtmalarımıza bağlı varlıklar olarak görmek.*
- *Özellikleri içsel değil, etkileşimsel görmek.*
- *Kategorileri dizi teorisi yoluyla sıkıca sabitlenmiş ve tanımlanmış olarak değil, prototip yoluyla tanımlanmış yaşantısal gestaltler olarak görmek.*

*Doğal dilin anlamı ve insanların gerek dillerini, gerekse yaşantılarını anlamalarına ilişkin meseleleri; a priori felsefi varsayım ve savlar meselesinden ziyade, salt yaşantısal meseleler olarak görüyoruz. Mecaz ve onu anlayış biçimimizi, bu konulara kanıt sağlayabilecek birçok olası kanıt alanının içinden seçtik. Mecaz üzerine odaklanmamızın dört nedeni şunlardır:*

*Nesnelci gelenekte mecaz, en iyisinden marjinal bir ilgi alanıdır ve semantik (nesnel anlam) alanından tümüyle dışlanmıştır. Doğrunun değerlendirilmesiyle ancak marjinal bir ilgisi olduğu düşünülür.*

*Yine de mecazın yalnızca dilimizde değil, kavramsal sistemimizde de hüküm sürdüğünü gördük. Kavramsal sistemimizde bu kadar temel yer tutan bir olgunun, doğruluğun ve anlamın değerlendirmesinde merkezî olmaması bize inanılmaz geliyor.*

*Mecazın yaşantımızı anlamak için en temel mekanizmalardan biri olduğunu gözlemledik. Bu da mecazın anlamın ve doğruluğun değerlendirmesinde çevresel önem taşıdığı ve anlamada olsa olsa marjinal bir rol oynadığına ilişkin nesnelci görüşle uyuşmadı.*

*Mecazın yeni bir anlam yaratabileceğini, benzerlikler yaratabileceğini, böylece yeni bir gerçeklik tanımlayabileceğini gördük. Dünyanın standart nesnelci resmedilişinde böyle bir görüşe yer yoktur.*

## Geleneksel Mecazın Nesnelci Değerlendirilmesi

*Burada tartışmakta olduğumuz olguların çoğu uzun süredir nesnelci gelenekte bilinmektedir, ancak bunlar bizimkinden çok daha farklı biçimde yorumlanmışlardır.*

*Bizim gündelik kavramsal sistemimizi yapılandırdığını düşündüğümüz geleneksel mecazi kavramları nesnelciler yoksayarlar. Onlar için mecazlar salt dile ilişkin meselelerdir; mecazi kavram diye bir şey yoktur.*

*Bizim mecazi kavramların örneği olarak gördüğümüz sözcük ve ifadeler (örn. ''Tüm bu gerçekleri hazmedemiyorum''daki **hazmetmek**) nesnelciler tarafından hiç de canlı mecazın örnekleri olarak algılanmayacaktır. Onlara göre **hazmetmek** sözcüğünün iki ayrı ve bağımsız gerçek (nesnel) anlamı vardır - yemek için **hazmetmek**$_1$, fikirler için **hazmetmek**$_2$. Bu değerlendirmede eşsesli iki **hazmetmek** sözcüğü vardır, tıpkı iki **kara** sözcüğü olduğu gibi (**renk kara** ve **toprak parçası olan kara**).*

*Nesnelci belki **fikri hazmetmenin, bir zamanlar** mecaz olmuş olduğunu teslim edebilir; ama artık mecazi olmadığını iddia edecektir. Onun için bu ''ölü mceaz'', gelenekselleşmiş ve kendi gerçek anlamını kazanmış bir mecazdır. Bu, eşsesli iki adet **hazmetmek** sözcüğü olduğu anlamına gelir.*

*Nesnelci muhtemelen **hazmetmek**$_1$ ve **hazmetmek**$_2$'nin benzer anlamları olduğunu ve bunun özgün mecazın temeli olduğunu teslim edecektir. Bu, diyecektir, iki ayrı anlamı ifade etmekte aynı sözcüğün kullanılmasının nedenidir; bir zamanlar mecazdı, dilin gelenekseI bir parçası oldu; öldü ve dondu, eski mecazi anlamı yeni bir gerçek anlam aldı.*

*Nesnelci ölü mecazın dayalı olduğu benzerliklerin bugün de birçok vakada algılanabileceğini gözlemleyecektir.*

*Nesnelci mecaz değerlendirmesine göre, özgün mecaz, bir kullanım ve konuşmacının kasdettikleri meselesiydi, gerçek nesnel anlam değil.*

*Gerçek nesnel anlam A'yı (A = Fikilerini, hazım borumdaki kimyasal ve kas hareketiyle, gövdemin emebileceği bir biçime dönüştüremiyorum) taşıyan bir cümle C (C = ''Fikirlerini hazmedemedim'') söylerken konuşmacı dinleyiciye konuşmacının anlamı A' (A' = fikirlerini, zihinsel hareket yoluyla, zihnimin emebileceği bir biçime sokamadım) iletmeye niyetlenmektedir.*

*Nesnelcinin değerlendirmesinin geçerli olabilmesi için iki şeyin doğru olması gerekir. İlkin, fikirlere gönderme yapmaya niyetli konuşmacının anlamı ' nesnel olarak verili nesnel doğruluk değerlerine sahip olması gerekir. Diğer bir deyişle, aşağıdakiler, içkin özellikleri nedeniyle zihin ve fikirler için nesnel olarak doğru olmalıdır:*

- *Fikirler, içkin özellikleri nedeniyle biçim alabilen, dönüşebilen ve zihinde masedilebilen bir şey olmalıdırlar.*
- *Zihin, içkin özellikleri nedeniyle, zihinsel etkinlikte bulunabilen, fikirleri dönüştürebilen ve onları kendinde masedebilen bir şey olmalıdır.*

*İkinci olarak mecaz, önceden varolan benzerliklere dayalı olmalıdır. Yani zihinle hazım borusu arasında da, tıpkı fikir ile yemek arasındaki gibi ortak **içkin** özellikler bulunmalıdır.*

*Özetlersek: **hazım**'a ilişkin ölü mecaz değerlendirmesi şu aşağıdakileri öne sürecektir:*

- ***Hazım** sözcüğü özgün olarak yemek kavramına göndermede bulunmuştur.*
- *''Canlı'' bir mecazla **hazım** sözcüğü, yemek ve fikirler arasında önceden varolan nesnel benzerlikler temelinde fikirler alanında önceden varolan bir nesnel anlama aktarılmıştır.*
- *Sonunda mecaz ''öldü'' ve **fikri hazmetme**'nin mecazi anlamı gelenekselleşti. **Hazım** böylece ikinci bir gerçek nesnel anlam kazandı. Nesnelci değerlendirmede bu,*

kendilerini ifade edecek sözcüklerden yoksun, önceden varolan anlamlara sözcük sağlamanın tipik bir yolu olarak görülür. Bütün bu vakalar eşseslilik örnekleri olarak görüleceklerdir.

## Nesnelci Değerlendirmenin Yanlışlıkları

Görmüş olduğumuz gibi geleneksel mecaz nesnelciler tarafından eşseslilikle açıklanmaktadır. Tüm bu görüşlere karşı ayrıntılı tartışmalar yürütmüş durumdayız. Bu tartışmalar şimdi özel bir önem kazanıyor. **Yalnızca nesnelcilerin mecaz görüşünün uygun olmadığını göstermekle kalmıyor, tüm nesnelci programın yanlış varsayımlara dayalı olduğunu da gösteriyor.** Mecazın nesnelci değerlendirmesinin tam nerede uygun olmadığını görmek için, soyutlama, eşseslilik ve benzerliğe ilişkin görüşlere karşı tartışmalarımızı, nesnelci **geleneksel** mecaz değerlendirmesiyle ilintili oldukları kadarıyla hatırlayalım.

## Benzerlik Durumu

FİKİRLER GIDADIR mecazıyla ilgili olarak, mecazın benzerliklere dayalı olmakla birlikte aralarındaki benzerliğin içkin olmadığını, başlıca mecazlara -özelde ZİHİN KAPTIR, FİKİRLER NESNEDİR ve ALIŞVERİŞ mecazlarına- dayalı bulunduğu söylenebilir. FİKİRLER NESNEDİR görüşü, ontolojik bir mecaz yoluyla zihinsel fenomene varlık statüsünün yansıtılmasıdır. **ZİHİN KAPTIR** görüşü, idrak melekelerimize iç-dış yönelimi ile varlık statüsünün yansıtılmasıdır. Bunlar fikir ve zihnin **içkin nesnel özellikleri** değildirler. Bunlar **içetkileşimsel özellikleridir** ve mecaz yoluyla zihinsel fenomenleri nasıl **algıladığımızı** yansıtmaktadırlar.

Aynı şey ZAMAN ve SEVGİ kavramlarımız için de geçerlidir. ''Hareket zamanı geldi'' ve ''Zamanımızı hesaplamamız gerek'' gibi cümleleri sırasıyla ZAMAN HAREKET EDEN BİR NESNEDİR ve VAKİT NAKİTTİR mecazlarının terimleriyle anlarız. Ancak nesnelci değerlendirmede böyle mecazlar olamaz. Bu cümlelerdeki GELİNCE ve HESAPLAMAK ölü mecazlar, yani bir zamanlar yaşamış mecazlardan tarihsel olarak türemiş eşsesliler olacaktır. Bu bir zamanlar yaşamış mecazlar bir yandan zaman ve hareket eden nesneler, öte yandan da zaman ve para arasındaki içkin benzerliklere dayalı olacaktır. Ama görmüş olduğumuz gibi böyle benzerlikler içkin değildir; ontolojik mecazlar yoluyla yaratılmışlardır.

SEVGİ kavramını içeren ifadeler için içkin-benzerlik analizi yapmak daha da zordur, örneğin ''Bu ilişki bir yere çıkmıyor'', ''Aramızda bir çekim var'' ve ''Bu ilişki ölüyor'' cümlelerinde. SEVGİ kavramı içkin olarak iyi tanımlanmış değildir. Kültürümüz bize sevgi deneyimlerini AŞK BİR YOLCULUKTUR, AŞK FİZİKSEL GÜÇTÜR vb. geleneksel mecazlarla bakmaya ilişkin geleneksel yollar sunar; ve dilimiz de bunları yansıtır. Ancak nesnelci değerlendirmeye göre SEVGİ kavramı, yolculuk, çekim ve hasta insanlarla içkin benzerlikler içerebilmek için içsel özellikleri açısından çok iyi tanımlanmış olmalıdır. Burada nesnelci yalnızca sevginin yolculuk, çekim ve hasta insanlarla benzer içkin özellikleri olduğunu öne sürmenin kamburunu taşımakla kalmıyor; bu benzerliklerin varolabilmesi için sevginin bu içkin özellikler bakımından yeterince net tanımlandığını da öne sürmek durumunda.

Özet olarak, bu fenomenlerin alışılmış nesnelci değerlendirmeleri içkin özelliklere ve önceden varolan benzerliklere dayalıdır. Genelde benzerlikler vardır, ancak bunlar **içkin** özelliklere dayandırılamazlar. Benzerlikler kavramsal mecazların **sonucu olarak** ortaya çıkar ve içkin olmaktan ziyade **etkileşimsel** benzerlikler olarak görülmelidirler. Ancak etkileşimsel özelliklerin kabulü, nesnelci felsefenin temel öncülüyle tutarsızlık taşır. Bu, nesnelcilik mitinden vazgeçmek anlamına gelir.

## Nesnelci İhmal: ''Bizim İşimiz Değil''

Nesnelciye kalan tek alternatif, YEMEK ve FİKİR'e ilişkin **hazım** anlamı arasındaki ilişkiyi değerlendirmekten vazgeçip (mecazın herhangi bir zaman varolmuş olduğunun yadsınmasıyla birlikte) güçlü eşseslilik görüşüne dönmektir. Bu görüşe göre tamamen farklı ve bağlantısız anlamlara sahip bir **hazım** sözcüğü vardır -**kara** sözcüğünün iki anlamı kadar farklı. Ancak güçlü eşseslilik görüşü şunları açıklayamaz:

— İç sistematiklik
— Dış sistematiklik
— Mecazın kullanılan kısmının yaygınlaşması

— *Soyut yaşantıyı yapılandırmak için somut yaşantının kullanılması*
— *Fikirlerin yemek terimleriyle mecazi kavramlaştırılmasına dayalı olarak* ***hazım****ın iki anlamı arasında gerçekte gördüğümüz benzerlikler.*

*Kuşkusuz nesnelci filozof ya da dilci, böyle sistematiklikleri, benzerlikleri ve daha az somutu daha çok somut olanın terimleriyle anlamayı açıklayamadığını teslim edecektir. Bu onu hiç de rahatsız etmeyebilir. Böyle bir değerlendirmenin kendi görevi olmadığını öne sürebilir. Böyle şeyler psikolog, nöropsikolog, filolog ya da bir başkasının meselesidir. Bu, Frege'nin ''anlam''ı ''fikir''den ayırma ya da Lewis'in ''soyut semantik sistemleri'' ''psikolojik ve sosyolojik olgular''dan ayırma geleneğine dahildir. Eşsesliliik görüşünün, nesnelciler olarak amaçlarına uygun, yani dilsel ifadeler için nesnel doğruluk değerleri sağlamaya ve bunların terimleriyle gerçek nesnel anlam değerlendirmesi yapmaya yeterli olduğunu öne süreceklerdir. Bunun, sistematiklik, benzerlik, anlama vs. değerlendirmeler yapılmadan da* ***hazım****ın iki anlamı için bağımsız olarak yapılabileceğini varsayarlar. İşlerini böyle algılamalarına bağıl olarak* ***hazım****ın geleneksel mecazi kullanımları salt eşsesliler ve ölü ya da canlı olsun mecaz olmayan şeyler içerir. Yalnızca geleneksel olmayan mecazları tanırlar (örn. ''Fikirlerin ucuz sıvadan yapılma'' ya da ''Sevgi ortak bir sanat yapıtıdır''). Bunlar da konuşmacının kasdettiklerine ait meseleler olup cümlenin gerçek nesnel anlamı olmadıkları için, bunlardan doğan doğruluk ve anlam sorunları konuşmacının yukarıda verilen anlamının değerlendirilmesiyle halledilecektir.*

*Özet olarak, geleneksel mecaza ilişkin tek iç tutarlılığı bulunan nesnelci görüş, bizim ilgilendiğimiz konuların -geleneksel mecazların özellikleri ve bunları kavramada kullanışınız- onlar için konu dışı olduğudur. Böyle meselelerden sorumlu olmadıklarına ve geleneksel mecaza ilişkin bu türden olguların nesnelci programla ya da nesnelcilerin inandığı şeyle bir ilgisi olmadığında ısrar edeceklerdir.*

*Böyle nesnelciler* ***etkileşimsel özellikler*** *ve* ***yaşantı gestaltları****nın gerçekten de insanların yaşantılarını mecaz yoluyla nasıl anladıklarının değerlendirilmesinde gerekli olduklarını da teslim edebilirler. Ancak bunu kabullenirken bile şu temellerde yaptığımız herşeyi görmezlikten gelmeyi sürdüreceklerdir; yaşantıcıların, tüm kısıtlılıkları verildiğinde, insanların gerçekliği nasıl anladıklarıyla ilgilendiklerini; nesnelcinin ise insanın birşeyin doğru olduğunu nasıl anladığıyla değil, birşeyin* ***gerçekten doğru olmasının*** *ne anlama geldiğiyle ilgilendiklerini söyleyeceklerdir.*

*Nesnelci yanıt, nesnelci ile yaşantıcı arasındaki temel farkı mükemmelen aydınlatmaktadır. Böyle bir nesnelci yanıt, onların insan fonksiyonları ve idrakinden tümüyle bağımsız bir ''mutlak doğru'' ve ''nesnel anlam''a ilişkin temel kaygılarının ifadesi demektir. Buna karşı biz mutlak doğru ya da nesnel anlam bulunduğuna inanmanın hiçbir nedeni olmadığını savunmaktayız. Bunun yerine, doğruluk ve anlamın yalnızca insanın dünyada işlev göstermesi ve onu anlamasına bağlantılı olarak değerlendirilebileceğini savunuyoruz. Böyle nesnelcilerden basitçe ayrı bir felsefi evrendeyiz.*

## Nesnelci Felsefenin İnsan Kaygılarına Bağlantısızlığı

*İnsan idrakine, kavramsal sistemimize ve doğal dilimize ilişkin uygun bir nesnelci değerlendirme olabileceğini düşünen nesnelcilerle ise aynı felsefi evrendeyiz ve ciddi görüş ayrılıklarımız var. Geleneksel mecazın insan dili ve kavramsal sisteminde hakim olduğunu ve anlamak için temel bir araç olduğunu ayrıntılarıyla tartıştık. İdrakin uygun bir değerlendirmesinin etkileşimsel özellikler ve deneysel gestaltler gerektirdiğini savunduk. Bütün nesnelci değerlendirmeler içkin özellikler gerektirdiği, çoğu da kavramsal sınıflandırmaları değerlendirmesi kümeler kuramından hareket ettiği için, insanların dünyayı nasıl kavramlaştırdıklarını uygun olarak değerlendiremezler.*

## Nesnelci Felsefenin Dışında Nesnelci Model

*Klasik matematik nesnelci bir evrenle uzlaşır. Birbirinden açıkça ayrık varlıkları, örn. sayıları vardır. Matematik varlıkların içkin özellikleri bulunur; örn.* ***üç*** *tek sayıdır. Ve bu varlıklar arasında sabit ilişkiler vardır, örn.* ***dokuz üç****'ün karesidir. Matematiksel mantık, klasik matematiğe temel sağlama işleminin bir parçası olarak geliştirilmişti. Formel semantik de*

buradan geliştirilmişti. Formel semantik kullanan modeller, ''nesnelci model'' diye adlandırdığımız şeyin örnekleridir - içkin özellikleri bulunan varlıklar ve bu varlıklar arasında sabit ilişkiler olan söylem evrenlerine uygun modellerdir bunlar.

Ancak gerçek dünya nesnelci evren değildir; özellikle de gerçek dünyanın insana ilişkin yönü: insan yaşantısı, insan kurumları, insan dili, insanın kavramsal sistemi. Sıkı bir nesnelci olmanın gereği, dünyaya gerçekten olduğu haliyle uyan bir nesnelci model olduğunu öne sürmektir. Nesnelci felsefenin, dil, doğruluk, anlama ve insanın kavramsal sistemi hakkında yanlış öngörülerde bulunduğu için deneysel olarak yanlış olduğunu tartışmış bulunuyoruz. Bu temelde nesnelci felsefenin insan bilimleri için yetersiz bir temel sağladığını iddia ettik. Yine de son derece öngörülü birçok matematikçi, mantıkçı, dilci, psikolog ve bilgisayar bilimcisi insan bilimlerinde kullanılmak için nesnelci modeller tasarımlamışlardır. Bütün bu çalışmanın yararsız olduğunu ve nesnelci modellerin insan bilimlerinde hiçbir yeri olmadığını mı iddia ediyoruz?

Böyle birşey iddia etmiyoruz. Matematik varlıklar olarak nesnelci modellerin zorunlu olarak nesnelci felsefeyle bağlantılandırılmaları gerektiğine inanmıyoruz. İnsan, nesnelci modellerin insan bilimlerinde bir işlevi olduğuna -hatta önemli bir işlevi olduğuna- inanabilir, ama dünyaya gerçekte olduğu haliyle tümüyle ve kesinkes uyan bir nesnelci model bulunduğuna ilişkin nesnelci öncülü benimsemesi gerekmez. Ancak bu öncülü reddedince, nesnelci modeller için geriye hangi rol kalmaktadır.

Bu soruyu yanıtlayabilmemiz için ontolojik ve yapısal mecazların bazı özelliklerine bakmamız gerekir:

— Ontolojik mecazlar yaşantımızı kavramak için elimizde olan en temel araçlar arasındadır. Her yapısal mecazın altbölüm olarak tutarlı bir dizi ontolojik mecazı vardır. Verili bir durumu anlamak için bir ontolojik mecazlar dizisi kullanmak, bu duruma bir varlık yapısı dayatmak anlamına gelir. Örneğin SEVGİ YOLCULUKTUR, SEVGİ'ye bir başlangıç, hedef, yol, yolda katedilen mesafe vs. gibi bir varlık yapısı dayatır.

— Tek tek her yapısal mecaz iç tutarlılık taşır ve yapılandırdığı kavrama tutarlı bir yapı dayatır. Örneğin TARTIŞMAK SAVAŞTIR mecazı iç tutarlı SAVAŞ yapısını TARTIŞMA kavramına dayatır. Sevgiyi yalnızca SEVGİ YOLCULUKTUR mecazının terimleriyle anladığımız zaman, iç tutarlılığı olan YOLCULUK yapısını SEVGİ kavramına dayatırız.

— Aynı kavram için farklı mecazlar birbirleri ile her zaman tutarlı değillerse de, birbirleri ile tutarlı olan mecaz dizileri bulmak mümkündür. Bunlara **tutarlı mecaz dizileri** adını verelim. Tek tek her mecazın iç tutarlılığı olduğundan, her tutarlı mecaz dizisi bizim bir durumu varlıklar arasında tutarlı ilişkiler bulunan iyi tanımlanmış bir varlık yapısı içinde kavramamızı sağlar.

— Tutarlı bir mecaz dizisinin varlıklar arasında bir dizi ilişkiyle varlık yapısını dayatması, nesnelci bir model tarafından temsil edilebilir. Modelde, varlıklar ontolojik mecazların dayattıklarıdır ve varlıklar arasındaki ilişkiler yapısal mecazların iç yapıları tarafından verilenlerdir.

***Özetlersek: Bir durumu böyle tutarlı bir mecazlar dizisinin terimleriyle yapılandırmaya çalışmak, kısmen o durumu nesnelci modelin terimleriyle yapılandırmaya çalışmak gibidir. Bunun dışında kalan mecazların yaşantısal temelleri ve mecazların gizledikleridir.***

O halde sorulacak doğal soru, insanların gerçekten de tutarlı mecaz dizileriyle davranıp konuşmakta olup olmadıklarıdır. Böyle yaptıkları özel bir durum, bilimsel teorilerde, örneğin biyoloji, psikoloji ya da dilbilimde formülleştirmelerdir. Formel bilimsel teoriler bir dizi ontolojik ve yapısal mecazı tutarlı olarak yaygınlaştırma girişimleridir. Ancak bilimsel teorileştirmenin yanısıra, biz insanların çok çeşitli durumlarda tutarlı mecaz dizileriyle düşünüp hareket etmeye çalıştıklarını hissediyoruz. Bunlar; insanların yaşantılarına nesnelci modeller uygulamakta oldukların düşünülebileceği durumlardır.

İnsanların bir yaşam durumunu nesnelci model, yani tutarlı mecazlar dizisinin terimleriyle görmeye çalışmaları için mükemmel bir neden vardır. Bunun nedeni, bu görüş gerçekleştiğinde durumdan birbiriyle çelişmeyen çıkarsamalar yapılabilmesidir. Yani davranış için çelişmeyen beklenti ve öneriler çıkarsayabiliriz. Ve dünyaya ilişkin tutarlı bir görüşe, ke-

sin bir beklenti dizisine ve ne yapmanız gerektiğine ilişkin çelişkisiz buyruklara sahip olmak son derece rahatlatıcıdır. Nesnelci modellerin gerçek bir çekiciliği vardır -ve en insani nedenlerle.

Bu çekiciliği aşağılamak istemiyoruz. Bu, yaşantıda ya da yaşam deneyiminin bir kısmında tutarlılık bulmanın çekiciliğine benzer. Beklenti ve eylem için temele sahip olmak yaşamak için önemlidir. Ancak tek bir nesnelci modeli bazı kısıtlı durumlara dayatıp o modelin terimleriyle -belki de başarılı olarak- işlev yürütmek bir şeydir; o modelin tüm gerçekliğin uygun yansısı olduğuna karar vermek başka birşeydir. Kavramsal sistemlerimizin tek bir kavram için tutarsız mecazlara sahip olmasının iyi bir nedeni vardır. Bunun nedeni tek bir mecazın işi görememesidir. Her biri kavramın bir yönüne ilişkin bir kavrayış sağlayıp diğerlerini gizlemektedir. Yalnızca bir dizi tutarlı mecazın terimleriyle çalışmak gerçekliğin birçok yönünü gizlemek olur. Gündelik yaşantımızda başarılı işlev yürütmek mecazların sürekli kaymasını gerektirmek gibidir. Eğer gündelik varlığımızın ayrıntılarını kavrayacaksak, birbirleriyle tutarsız birçok mecazın kullanımı bize gerekli görünmektedir.

İnsan bilimlerinde formel-nesnelci modellerin incelenmesinde bir bariz yarar bizim **kısmen** tutarlı mecaz dizileriyle akıl ve işlev yürütme yetisini anlamamızı sağlamalarıdır. Bu ortak bir etkinliktir ve anlaşılması önemlidir. Aynı zamanda bizim tutarlılığı gerekli kılmanın neden yanlış olduğunu, herhangi bir tutarlı mecaz dizisinin muhtemelen gerçekliğin çok çeşitli yönlerini -yalnızca o mecazla tutarsız mecazların vurgulayabileceği yönlerini- gizleyeceğini görmemizi sağlar.

Formel modellerin bir bariz kısıtlılığı, tahayyül edebildiğimiz kadarıyla mecazın yaşantısal temelini de içermek için bir yol sunmamaları, dolayısıyla mecazi kavramların yaşantımızı kavramamızı nasıl sağladıklarını değerlendirmemize imkan tanımamalarıdır. Bunun, bir bilgisayarın olguları insanlar gibi kavrayıp kavrayamayacağı meselesiyle ilişkili bir sonucu da vardır. Bizim yanıtımız hayırdır -çünkü anlamak yaşantı gerektirir ve bilgisayarların gövdesi, insani deneyimleri yoktur.

Ancak bilgisayar modellerinin incelenmesi bize insanın zihinsel kapasitesi üzerine çok şey öğretir, özellikle insanların kısmen nesnelci modellerin terimleriyle işlev yürüttüğü alanlarda. Dahası, bilgisayar bilimindeki bugünkü formel teknikler **tutarsız** mecaz dizilerini temsil etme ihtimalini de taşımaktadır. Bu insanların ahenkli ama tutarsız mecazi kavramların terimleriyle akıl ve işlev yürütme biçimleri üzerine inandırıcı öngörülere yol açabilir. Formel çalışmanın sınırlarının, kavramsal sistemimizin yaşantısal temeli alanında olduğu görülmektedir.

## Özet

Genel sonucumuz nesnelci programın insan idraki ve böyle bir değerlendirme gerektiren diğer konularda doyurucu bir değerlendirme yapmakta yetersiz kaldığıdır. Bu konular arasında şunlar vardır:

- insanın kavramsal sistemi ve insan rasyonelliğinin doğası
- insan dil ve iletişimi
- beşeri bilimler, özellikle psikoloji, antropoloji, sosyoloji, dilbilim
- ahlaki ve estetik değerler
- insanın kavramsal sistemi yoluyla bilimsel idrak
- matematiğin temellerinin insan idrakindeki temeli olması için herhangi bir yol

Bu insani meselelerin uygun olarak ele alınması için, anlamaya ilişkin yaşantıcı değerlendirmenin temel öğeleri -etkileşimsel özellikler, deneysel gestaltler, mecazi kavramlar- gerekli gözükmektedir.

**George Lukoff**
**Mark Johnson**

**The Metaphors We Live By**

1871

Nasıl kişisel bilgisayar devriminin öncüsü olan küçük firmalar güçlü ve büyük şirket imajının ve piyasa ve işbölümünün yapısının önemli bir biçimde değişmesine yolaçtıysa, 1980'lerin ikinci yarısında ICOT'un da yine böyle bir etkisi olacağa benziyor. Öyle ki ICOT'un kurulmasından hemen sonra, Amerikan bilgisayar firması ve süper bilgisayar üreticisi, Control Data şirketinin kurucusu ve genel müdürü William Norris, piyasadaki bütün yarı-iletken ve bilgisayar firmalarının yöneticilerini Florida'da bir toplantıya çağırdı. Bu toplantının sonucunda aralarında Honeywell, Motorola, RCA ve Control Data'nın kendisinin de bulunduğu 12 büyük firma ortak bir araştırma kooperatifi kurmaya karar verdiler: **MCC** (Microelectroelectronics and Computer Technology-Mikroelektronik ve Bilgisayar Teknolojisi).

Büyük ölçüde Japon modelinde örgütlenmiş olan MCC bugüne kadar Amerikan şirketlerinin alışmış olduğu çalışma tarzından cüretli bir kopuşu temsil ediyordu. MCC'yi kuran şirketler, geleneksel olarak birbirleriyle rekabet eden, bu yüzden de geliştirdikleri teknolojik yenilikleri bir ticari sır olarak büyük bir gizlilikle saklayan şirketlerdi.

Yeni araştırma kooperatifinin hedefi hernekadar beşinci kuşak tasarısı diye adlandırılmıyorsa da, gerçekleştirilmek istenen amaçlar Japonların beşinci kuşak tasarısı ile gerçekleştirmek istedikleri ile aynı: Yeni teknolojiler arasında bağlantılar kurmak, özellikle bilgisayar, tümleşik devre ve yonga üretimi ve tasarımında bilgisayar destekli tasarım ve üretimi (CAD/CAM) geliştirmek, sıralı bilgi işlemin dayattığı darboğazları aşan yeni bilgisayar mimarileri keşfetmek ve yapay zeka...

## Beşinci Kuşak Tasarısının Teknolojik Hedefi

Hem Japon ICOT'un beşinci kuşak tasarısının hem de Amerikan MCC'in hedeflerini kısaca şu şekilde özetleyebiliriz: ''Bilgi mühendisliği'' **(knowledge engineering)** diye adlandırılabilecek yeni bir bilgi işlem anlayışının uygulama alanlarını alabildiğince genişletmek. Bu yeni anlayışın ilk akla gelen uygulama alanları arasında uzman sistemleri, doğal dillerin bilgisayarlar tarafından anlaşılabilmesini ve robot uygulamalarını sayabiliriz. Bu yazılım ve yeni bilgisayar dilleri alanında hatırı sayılır yenilikler gerektiriyor gerçi, ama yaygın kanıya göre bugün asıl sorun donanım teknolojisinde düğümleniyor.

Kendisinden beklenenleri gerçekleştirebilmek için, yeni bir donanım teknolojisi aşamasını temsil eden beşinci kuşak bilgisayarların geleneksel veri tabanlarından farklı ve onlardan çok daha geniş ''bilgi tabanları''na **(knowledge base)** sahip olması gerekiyor. Ayrıca bu bilgisayarlar yüksek hızlarda ''çağrışımsal erişimler'' **(associative retrieval)** gerçekleştirebilmeli, bugünkü bilgisayarların aritmetik işlemler yaptığı hızda mantıksal çıkarımlar yapabilmeli, paralel yapılar içeren program dilleri işletebilmeli, donanımda yüksek hızda paralel süreçlere izin verebilmeli ve doğal diller ve görsel malzemelerle komut verilmesini mümkün kılan kullanıcı arabirimleri ile donatılabilmeliler.

Bugüne kadar bilgi mühendisleri tarafından geliştirilen bütün uzman sistemler başlıca üç ana bileşenden oluşuyor. Birincisi problem çözümü ve genel olarak ''anlayış'' için gereken bilgi tabanını, işleten bilgi tabanı işletim bileşeni. İkincisi problem çözüm ve çıkarım bileşeni. Bu bileşenin işlevleri hangi bilgilerin eldeki probleme ilişkin olduğunu keşfetmek ve bu bilgilerden hareketle, adım adım bilgisayarı çözüme, en akla yakın yoruma ya da uygun bir hipoteze ulaştıracak bir akıl-yürütme tarzı oluşturmak. Üçüncüsü ise kullanıcı ile makinenin ''doğal'' ve rahat bir biçimde etkileşmesini sağlayacak bileşen.

Bilgi tabanının işletimi, problem çözümü -ve çıkarım ve insani etkileşim- bugüne kadar her üç alanda da hatırı sayılır gelişmeler kaydedildi. Ama bu gelişmeler tamamıyla yazılım alanındaki ilerleme ve yeniliklerden kaynaklanıyordu. Bu gelişmeler ise eninde sonunda, daha önce von Neumann mimarisi diye anmış olduğumuz, sıralı işlem yapma zorunluluğunun yarattığı darboğazda tıkandılar.

Beşinci kuşak tasarısı da bu üç bileşeni veri kabul ederek, bilgisayarı bu üç birim çevresinde düzenliyor ama çok önemli bir farkla.

Her bileşenin kendine özgü bir yazılım, bir de donanım düzeyinin olması öngörülüyor. Bu ise üç farklı bileşenlerin yazılım ve donanım düzeylerinin birbirleriyle etkileşmesini mümkün kılacak bir ''dil''in tasarlanması sorununu gündeme getiriyor.

Sorunu kısaca da olsa şöyle açabiliriz: Bilgi tabanındaki bilgilerin, problem çözüm ve çıkarım bileşeninin işleyebileceği bir simgesel kodda ve bellek yapısında saklanmış olması gerekir. Bu kod ve yapı çeşitli şekillerde temsil edilebilir.

En sık kullanılan temsil biçimlerinden biri **obje** olarak anılanıdır. Bu temsil biçiminde bütün bilgiler bir obje'yi tanımlayan nitelikler olarak temsil edilir. Her obje ise diğerleriyle simgesel göndermeler **(links)** aracılığıyla ilişkilendirilir. Böylelikle ''çağrışımsal şebeke'' **(associative network)** diye bilinen bir yapı oluşturulur. Bunun tipik bir örneği hiyerarşik taksonomilerdir. Burada her obje şu şekilde temsil edilir: ''... bir .... dir'':

Kırlangıç bir kuş'dır
Kuş bir uçan hayvan'dır

Bu tür iki objeden hareketle doğrudan doğruya bilgi tabanı işletim sistemi kırlangıcın uçabileceği çıkarımını yapar.

Sık kullanılan bir diğer temsil biçimi ise **kural**'dır **(rule)**. (Ancak ''kural''ın bu anlamının, daha önce yapay zeka bahsinde kullandığımız anlamından daha dar olduğunu vurgulayalım.) Her kural genellikle bir IF'le ifade edilen bir öncülle, bu öncüle bağlı bir sonuç ya da öncül doğrulandığında yapılması gereken bir işlemden oluşur. Örneğin ''IF sis 200 m.'den yakınsa ve resmi hava tahminine göre bir saate kadar dağılmayacaksa,THEN iniş tehlikelidir VE uçuş kurallarına aykırıdır VE yakın bir havaalanı ara!''

Belli bir sorunla karşılaşıldığında, problem çözüm ve çıkarım bileşeninin bilgi tabanındaki kuralları gözden geçirerek hangilerinin eldeki probleme ilişkin olduğuna karar vermesi gerekir. Bugün beşinci kuşak için öngörülen hacimde bir bilgi tabanında gerekli tarama muazzam olacaktır. Dolayısıyla, tarama için gerekli işlemi asgariye indirmek için bilgi tabanının nasıl düzenlenmesi gerektiği yine bilgi tabanı işletim sistemleri tarafından belirlenir.

Geliştirilmekte olan beşinci kuşak prototipinin, binlerce kural ve binlerce objeden oluşan oldukça mütevazi bir bilgi tabanı bileşeni içermesi öngörülüyor. Bu, bugünkü uzman sistemlerin veri taban hacmine denktir.

Ancak tasarının 10 yıllık ilk mühleti bittiğinde ulaşılması istenen hedef, onbinlerce kural ve yüz milyon obje. Bu, örneğin *Encyclopedia Brittanica*'nın tamanını içerebilecek bir hacim.

Böylesine büyük bir hacmi mümkün kılan daha önce sözünü etmiş olduğumuz VLSI teknolojisi. Ancak böylesine büyük bir bilgi tabanına erişim nasıl ve buradaki tarama nasıl mümkün olacaktır. Japonlar 10 yıllık sürenin sonunda 1 milyon LIPS'lik bir güce erişimi olmayı umuyorlar. **(LIPS-Logical Inferences per Second** -Saniyedeki mantıki çıkarım- kelimelerinin baş harflerinden türetilmiş bir birimdir. Kullanımı özellikle süper bilgisayarlar ve beşinci kuşak tasarısıyla birlikte yaygınlaşmaya başladı). Bu hız nasıl mümkün olacak? Bu soruyu cevaplamak için ''paralel işlem'' **(paralel processing)** ve ''veri akışı'' **(data flow)** kavramlarını gözden geçirmemiz gerekiyor.

## Paralel İşlem ve Veri Akışı Sistemleri

Geleneksel, von Neuman mimarisi diye adlandırılan yapıya uygun olarak inşa edilmiş bilgisayarların hepsinin temelde sıralı işlem **(sequential processing)** ilkesine göre çalıştıklarını söylemiştik. Bu işleyiş tarzında bir işlem tamamlanıp sonuçları elde edilmeden önce, ikinci bir işleme geçilemez.

Paralel işlem ilkesinde ise bir dizi merkezi işlem birimi birden çalışabilir ve herhangi bir işlemin yapılabilmesi için gerekli veriler elde edildiği anda, o işlem yerine getirilir.

Paralel işlemi mümkün kılan en önemli kavramlardan biri veri akışı **(data flow)** sistemleri kavramıdır. Şimdiye kadar incelediğimiz bilgisayar dillerinden temel birimin daima bir bellek yerleşimi adresi olduğunu gördük. Başka bir deyişle, bu dillerde bilgisayara (sabitler dışında) hiçbir değer doğrudan doğruya verilmiyor, bir bellek yerleşiminin adresi veriliyor ve bilgisayardan değeri gidip o adresden alması bekleniyordu.

Oysa veri akışı sistemlerinden belli değişken adları, değeri içeren bellek yerleşimine değil doğrudan doğruya belli değerlerin kendisine bağlanır. Bu yüzden bir veri akışı sistemi dilinde kullanılan X: = B + C gibi bir komut ancak yüzeysel (sözdizimsel) olarak diğer dillerdeki değer atama **(assignment)** komutlarına benzer. Veri akışı sistemi dilindeki X: = B + C komutunun anlamı, B + C'nin değerinin hesaplanması ve sonucun X adı ile ilintilendirilmesidir. Böylelikle X kesin bir matematiksel anlam kazanır ve X'in bu anlamı içinde yeraldığı program bölümü boyunca aynı kalır. Bu yüzden de diğer aritmetik ya da mantıksal işlemciler de X ile ilintilendirilmiş değeri, yalnızca X adına atıf yaparak kullanabilirler.

Bu sistemlerin ikinci bir özelliği de, her işlemin bazı fonksiyonların belli değerlere uygulanarak yeni değerler elde edilmesi şeklinde yapılmasıdır. Bu özelliğe sahir dillere, bu yüzden uygulayımsal' **(applicative)** denir. Değerlerin kendisine yönelik, uygulayımsal diller işlemler üzerine, kullanılmakta olan algoritmaya içkin olan veri bağımlılıklarının ötesinde herhangi bir sıralılık zorunluluğu empoze etmezler. Her fonksiyon kendisi için gereken bütün girdilerin hesaplanmasını beklemek durumundadır ama fonsiyonların hesaplanma sırası nihai sonucu etkilemez. Bu sistemlerde her tek

komutu sıra ile harekete geçiren bir merkezi kontrol birimine yer yoktur. Bu model dillere olduğu kadar mimarilere de uygulanabilir.

**Veri Bağımlılık Grafikleri:** Bir veri akışı sisteminde yapılacak işlemler bir veri bağımlılıkları grafiği şeklinde temsil edilir. Böyle bir grafikte her düğüm bir fonksiyonu, düğümleri birbirine bağlayan çizgilerse değerleri temsil eder. Bir veri akışı sistemi bilgisayarının makine dili de geleneksel bir komutlar dizisi değil, bir bağımlılıklar grafiğidir. Böyle bir bilgisayarda bir program sayacı da yoktur. Bunun yerine herhangi bir fonksiyonun ne zaman mümkün olduğunu, yani bütün girdilerinin ne zaman hazır olduğunu saptayan bir mekanizma vardır. Mümkünleşen **(enabled)** komutlar girdi ve varış noktaları ile birlikte işlemcilere yollanır. Sonuçlar varış noktalarına yollanır ve burada yeni komutları mümkün kılabilir. Bu işlem tarzına veri itimli **(data driven)** denir.

**Avantajları ve Sınırlılıkları:** Geleneksel bilgisayar mimarileri ve program yapıları kullanıldığında, belli bir problem paralel yapılar içeriyor bile olsa, bu yapıların bilgisayar tarafından işlenebilmesi için sıralı bir mantığa çevrilmesi gerekir. Oysa işlemler problemin kendi mantığında çok daha hızlı yapılabilir. Ayrıca bu süreç esnasında bir dizi düşük hızlı işlemci birarada kullanılabileceğinden, VLSI teknolojisinden de yaralanmak mümkün olmaktadır.

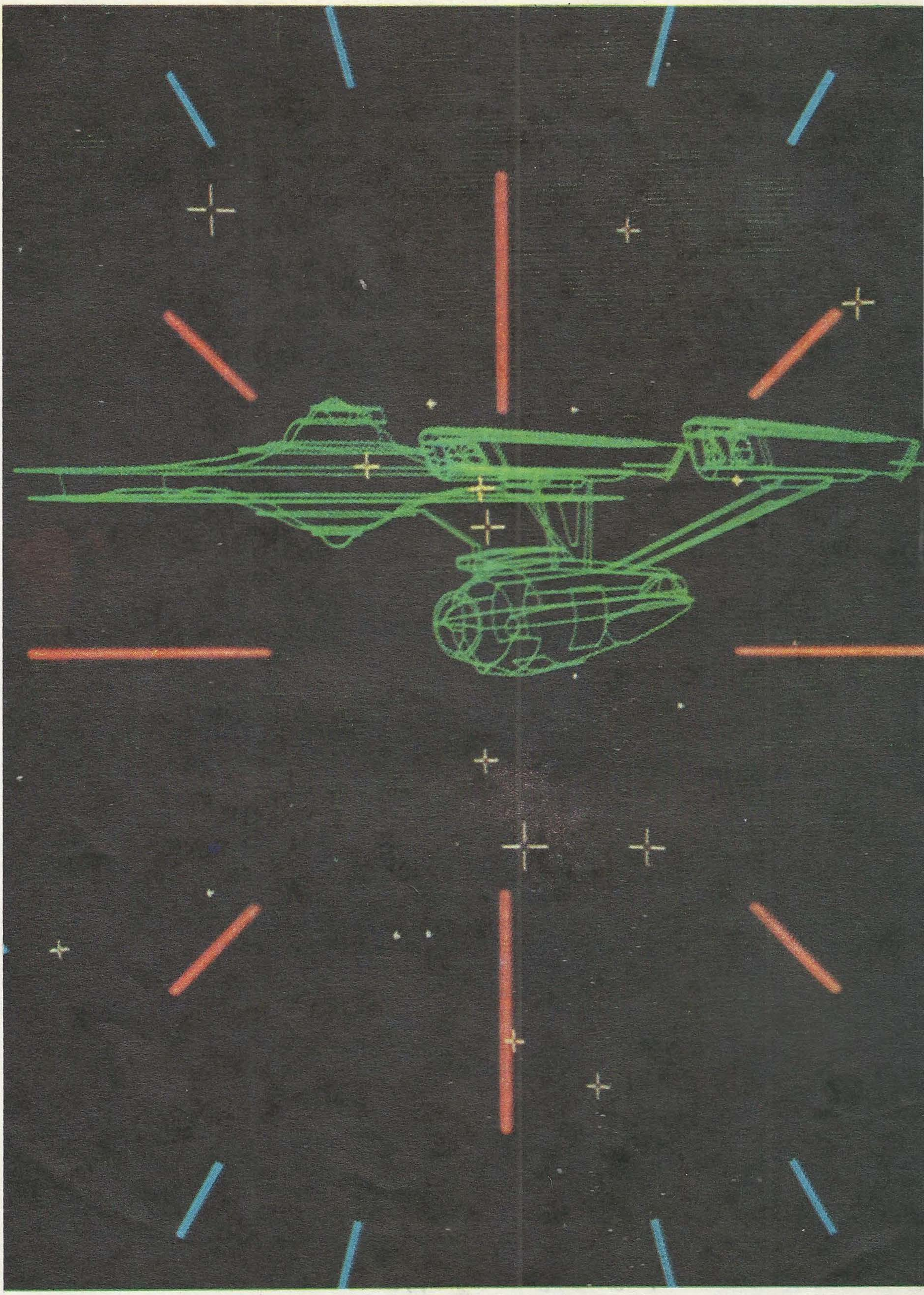
Ancak henüz çözülmemiş bazı sorunlar da mevcuttur. Bunların başında karmaşık veri yapılarının nasıl tek bir değer halinde ifade edilebileceği sorunu gelmektedir. Bu sorun kısmen paralel işlem tarafından çözülmektedir ama paralel işlem gerektirmeyen problemlerde, karmaşık veri yapıları aşırı uzunlukta işlem kuyruklarının **(pipeline)** doğmasına yolaçmaktadır. Ayrıca programcının belleğini doğrudan doğruya kontrol etmiyor oluşu da özel bellek temizleme mekanizmaları geliştirilmesini gerekli kılmaktadır.

# Bilgisayarlı Kuşaklar ve LOGO

New-York bilgisayar piyasasından bir yönetici ''1965'ten önce doğduysan bu işin dışında kalacaksın evlat'' diyor. Gerçekten de, ''bilgisayar çağı''nın ne zaman başladığı kesin olarak söylenemese de, bu tarih 1960'lardan geriye gitmez. Yetişkinler bilgisayarı anlamaya ya da açık/kapalı, G/Ç mantık devrelerini öğrenmeye çalışırken, çocuklar bilgisayarı ''yaşıyorlar''. Onbinlerce çocuk program kütüphaneleri kuruyor, aldığı dergilerde sık sık kendi yaşıtlarının yazı ve programlarıyla karşılaşıyor, ''PEEK'lemek yerine POKE'layacaktın'' gibi bir jargonla konuşuyor, yeni modelleri ve programları kimi zaman satıcılardan bile önce izliyorlar.

Bu yaygınlığın koşullarını öncelikle kuşkusuz teknoloji sağladı: Bilgisayarların hacimleri ve fiyatları küçüldükçe her eve girebilir aletler durumuna geldiler.

Bilgisayar oyunlarının popülerliği de bir ölçüde bilgisayarın kendisine yönelerek ilgiyi besliyor. Çok sayıda oyuncu için bir sonraki adım bir başka oyuna geçmek olsa da, bilgisayar oyunları kendi içinde kapanan bir döngü değil ve makinelerin oyun amacıyla kullanımı programcılığa da kapı açıyor.



1876

''Gelecek kaygusu'' bir başka neden; öğrenci velileri eve bilgisayar almak için bütçelerini dengelerken ya da çocuklarının okuluna bilgisayar alınması için bağış yaparken, bilgisayar kullanmanın gelecekte iş bulmak için öncelik taşıdığını düşünüyorlar. Ama bu kaygu-yu aynı şiddetle hissetmeyen çocukların bilgisayara yönelmelerini açıklamada bu da yetersiz kalıyor.

Bilgisayarın çocuklar için çekiciliği büyük ölçüde onunla kurdukları ilişkide yatıyor: Makineyi kontrol edebilmeleri, düşünüp bir şeyi istemeleri ve onu gerçekleştirmeleri, pek çok çocuğun yalnızca bilgisayar sayesinde tatmin edebildiği şeyler. 14 yaşında bir çocuk Time muhabirine durumu şöyle açıklıyor: ''Bu makineleri seviyorum. Tüm bu gücü parmak uçlarımla elde edebiliyorum. Bilgisayarlar olmazsa ne olacağım belirsiz. Oysa onların karşısındayken ben de birisi oluyorum.''

Çocukların bilgisayar üzerinde kurdukları hakimiyeti gözleyenlerden biri de, son dönemin en gözde diller ailesi LOGO'nun yaratıcısı Seymour Papert: ''Bugün pek çok okuldaki bilgisayar-destekli-eğitim'e, bilgisayarın çocuğa öğrettiği bir süreç olarak bakılıyor. Bundan hareketle, bilgisayarın çocuğu programladığı öne sürülebilir.''

Ama hem matematikçi, hem de eğitimci olan Papert'in eğitsel ilgileri daha ağır basıyor ve onun asıl üzerinde durduğu, çocukların öğrenme mekanizmaları. Piaget'nin izleyicisi olan Papert, bu mekanizmayı ''Piagetci öğrenme'' ya da ''öğretilmeden öğrenme'' olarak adlandırıyor.

Çocuk öğrenirken, **doğal çevre**'sindeki malzemeleri kullanarak bir dil ve bilgi yapısı oluşturur. Bu doğal öğrenme süreci, Fransızcayı Fransa'da öğrenmek gibi örneklenebilir. Bilgisayarlar da çocuk için bir doğal öğrenme çevresi durumuna getirilirse, çocuk onlara hakim olmayı ve öğrenmek için kullanmayı kolaylıkla becerebilir. Oysa geleneksel eğitim, özellikle dershane eğitimi böylesi bir doğal çevre sunmaz, öğrenmek için çok yapay ve verimsiz bir çevredir. LOGO'nun yaratılması bu temel düşünceye dayanır: Bilgisayar, öğrenmeyi doğal bir süreç durumuna getirebilir ve çocuğa öğrenme için gereken araçları sağlayabilir.

Bu araçlar, çocuğun soyut bilimsel ifadeleri somutlayabileceği doğallıkta, bunları kullanarak düşünebileceği nesneler olmalıdır. LOGO'daki **''Turtle''** (tosbağa), bu tür bir araç-

tır. Turtle, bilgisayar kontrollu sibernetik bir hayvandır ve LOGO'nun oluşturduğu ''çevre''-de, programlama ve düşünmeyi kolaylaştıran bir araç olarak iş görür. LOGO komutları uyarınca (bu komutlar çocuğun konuşma diline yakın ve çok basittir), ekranda hareket eder.

Turtle kullanarak ekranda çeşitli şekiller oluşturan çocuk, bu şekillere bir ad vererek daha sonra yeni bir komut olarak kullanabilir. Örneğin bir üçgen ve bir kare kullanarak çizdiği basit şekle EV adı verirse, EV'i tek komutla ekrana çağırıp hareket ettirebilir, yeni parçalar ekleyebilir, renklendirebilir, vb.

Bunun çok eğlenceli oluşu tartışma götürmez, ama bu süreç aynı zamanda çok güçlü öğrenme biçimleri de içeriyor. Biçim, hareket, hız, işlemler, yardımlar üzerine doğal bir süreçte bilgilenen çocuk, matematik konuşmaya başlıyor.

LOGO bugün, çocukların kendilerine bir ''bilgisayar çevresi'' oluşturmaları, bilgisayarı etkin ve üstün olarak kullanabilmeleri için en elverişli dil. Bu arada, çocukların da bilgisayarı yetkin olarak kullanabilecekleri savını destekleyen en geçerli kanıt. Çocukların bilgisayar kullanabilecekleri artık tartışma konusu bile değil; dahası, bilgisayarı yetişkinlerden daha iyi kullanmaları sözkonusu.

## Eleştiriler

Bu noktada yeni bir başlık beliriyor: Bilgisayarla kurulan ilişki, insanların genel düşünme ve öğrenme biçimlerini de etkilemez mi? Bilgisayar kullanmayı öğrenen çocuğun, diğer öğrenme biçimleri de değişiyor mu?

Programcılıkta kullanılan mantığın kesin, kurallı adımları, yeni kuşağın düşünme sürecini da şekillendiriyor. Analitik ve rasyonel düşüncenin egemenliği daha derin ve erken beliriyor; daha doğrusu çocukları gözleyen yetişkinler durumu böyle değerlendiriyorlar.

Ama bunun ''iyi'' mi, ''kötü'' mü olduğunu cevaplamak çok zor. Çocukların makinalarla bir üstünlük-kurma, denetim altına alma ve yönetme ilişkisi kurduğu düşünülürse; bilgisayarların, bu ilişki biçiminin egemen olduğu bir dünyanın koşullarını oluşturduğu söylenebiliyor. Başka bir söyleyişle, ''uyum kurmayı öğrenmek'' yerine ''yönetmek için gerekli bilgi ve beceriyi edinmek'' ön plana çıkıyor. Bilgisayar dilinin özellikleri nedeniyle, bu dilin dünyasında yaratıcılık ve insan duygulanımına yer olmadığını öne sürenler de var.

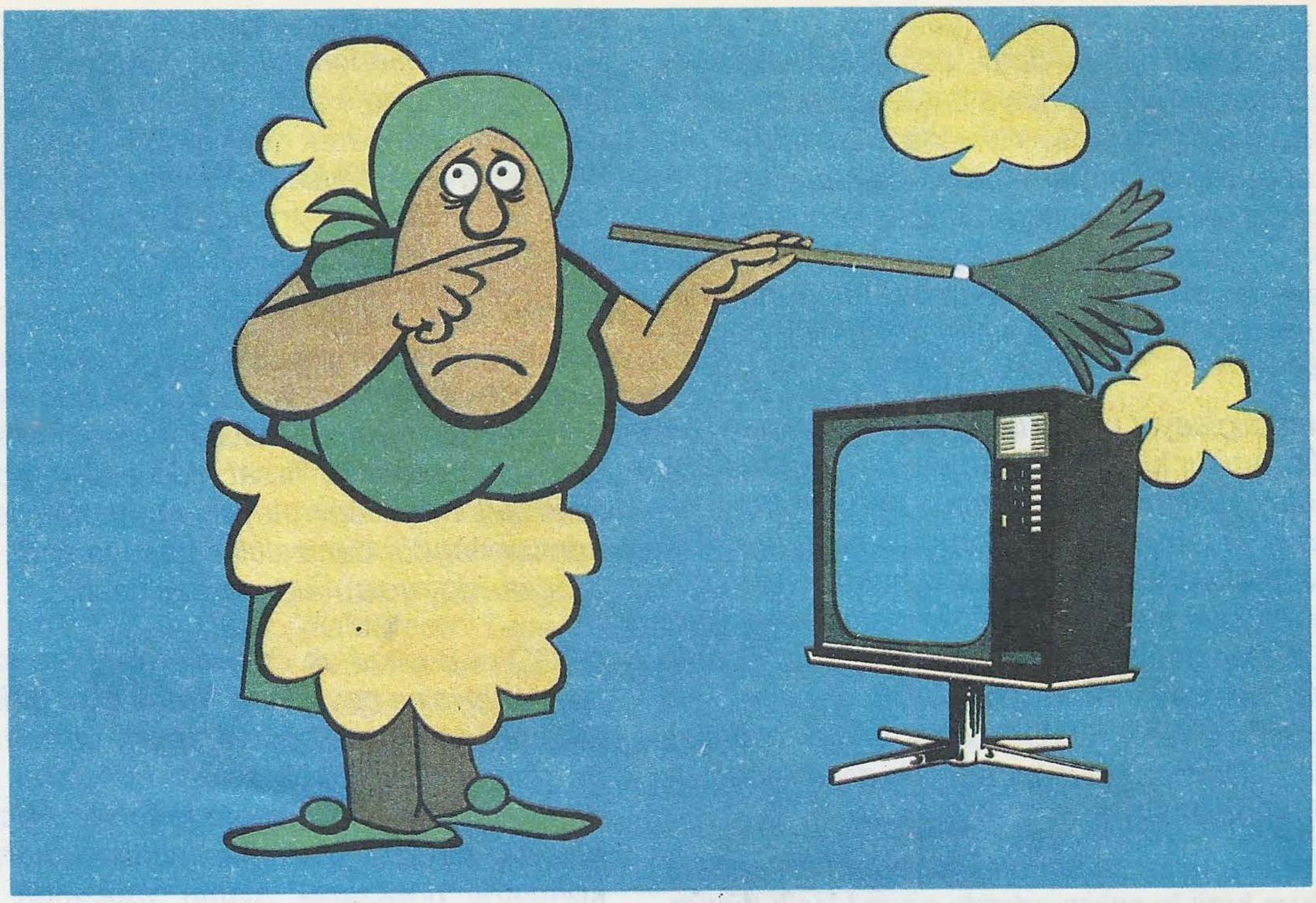
Öte yandan, analitik-rasyonel yaklaşım, yeni kuşağın modern bilime çok daha yakın olmasını sağlıyor, onu öğrenip kullanmasını kolaylaştırıyor. Bilimsel ve teknik eğitim için bilgisayarların kullanımı da hızla yaygınlaşıyor. Geleneksel eğitim yöntemlerinden hem daha etkin, hem de daha ekonomik oldukları için, başlangıç yatırımı büyük olmasına karşın bilgisayarlar okulları ağır ağır fethediyor.

Başlarda ''bilgisayar-destekli-eğitim'' için ''zaman-paylaşımı''na dayalı sistemler kullanılmaktaydı. 1960'larda PLATO (Programmed Logic for Automatic Teaching Operations) gibi programlarla ortaya çıkan bu yöntem, büyük bir merkezî makineye telefon ya da kablo ağıyla bağlanan **''dumb''** (aptal) terminallerden oluşuyor. Tüm yazılım, merkez makinenin belleğine yükleniyor ve sistemi paylaşan öğretmen ya da öğrenciler tarafından gerektiğinde yararlanılıyor.

Ama bu tür zaman-paylaşımı sistemleri, merkezî bilgisayarla olan bağlantıların sürekli açık tutulması nedeniyle çok pahalı. Ayrıca, kullanım zamanlarının genellikle belirli saatlerde çakışması, sistemi gereğinden fazla büyütüyor. En önemlisi de, bilgisayardan beklenen temel işlevi cevaplamıyor: Programlamayı öğrenmek ve programlayarak kullanmak. Bu nedenle, kullanıcı için daha tatmin edici bir ilişki sunan kişisel bilgisayarlar, daha özerk ve küçük sistemler yaygınlaşıyor.

Kuşkusuz bilgisayarlar, eğitim bütçelerinin önemli bir kalemi durumuna gelecekler. Bu konudaki teknik ve ekonomik sorunlar hem gereksinimlere, hem de gelişen teknolojiye bağlı olarak değişiyor. Ama değişmeyen bir başka başlık, en az işin bu yönü kadar önemli: Çocukların bilgisayarla kurduğu ilişkinin dünyayla olan ilişkilerini nasıl etkileyeceği.

New-Yorklu bir ilkokul öğrencisi, bilgisayar için ''Tamamen aptal'' diyor, ''her şeyi benim söylemem gerekiyor.'' Durum gerçekten buysa, bilgisayarların insanlaşacağına dayanan eski bilim-kurgu öyküleri o kadar ürkütücü değil, belki hiçbir zaman da gerçekleşemeyecekler. Ama insanların bilgisayarlaşması daha önemli ve gerçek bir sorun demeti oluşturuyor.



# Seymour Papert ve LOGO Evreni

*Olay, büyüklüğü ve türünde ilkti. Dr. Seymour Papert, M.I.T.'nin pırıltılı, sakallı peygamberi, iki haftalığına kentteydi. Yaz boyunca, o ve özenle seçilmiş sekiz öğretim görevlisi bu iki hafta süresince verilecek seminer için çalışmışlardı. Seminer, "**Bilgisayar: Öğrenme İçin Bir Dil**" olarak adlandırıldı. Çalışma alanları, bilgisayar dillerinin karşılaştırılması, temel bilgisayar teknolojisine giriş ve Papert'in kendi buluşu LOGO'nun derinlikli araştırılmasıydı. LOGO, M.I.T.'nin Yapay Zekâ Laboratuvarları'nda çocuklara yönelik geliştirilmiş bir bilgisayar dilidir. Gücü basitliğinden gelir. Renge, grafiğe ve çocuklara yönelik bir dildir. İtici kuvvetleri bir kamlumbağa, bir şeytan ve 32 ekran cinidir.*

*Kamlumbağa, nerde olduğunu gösteren renkli bir izi arkasında bırakarak belli yönlerde hareket eden üçgen bir hayvandır. Şeytan, uzaklıkları ölçerek, çeyrek daireleri gözleyerek, koşulları denetleyerek düdük taşıyan bir hakem gibi ekranda ordan oraya koşar. Son olarak, ekran cinleri vardır; pembe kar tanelerini, gri A'ları, küf kırmızısı denizaltıları siyah ekranda taşıyan çevik, görünmez hayaletler.*

*LOGO'dan asıl umulan; yalnızca yeni bir bilgisayar dili olması değil, çocukların elleri yerine kafalarıyla kurmaya başlayabilecekleri bir ortam da olmasıdır. Yalnızca düşüncenin dokunabileceği bir oyuncak topluluğu, elektronik bir tak-boz takımıdır. Küçük çocukların içine girebilmelerini engelleyen ya da hayli sınırlayan mevcut öteki dillerin üstünlüğünün tersine, LOGO, çocuğa sınırlı, zengin bir güç verir.*

*Her bilgisayar dilinin öncelikleri ve sınırları vardır. Açıktır ki, LOGO'dan başka dillerce daha kolay yüklenilebilecek, çoğu daha bilimsel veya iş yaşamına dönük sorunlar mevcuttur. Fakat meraklı küçük çocuk için bundan daha çok potansiyele sahip bir dünyayı düşlemek güçtür. Papert kitabına (Zihin Fırtınaları) ''Çocuklar, bilgisayarlar ve Güçlü Düşünceler'' altbaşlığını verdi. Onun, politik imalar olmaksızın, ''Devrimci Düşünceler''de diyebileceği de düşünülebilir.*

*Papert, uzun zamandır Piaget'nin arkadaşıydı ve birlikte çalışıyorlardı; onun klasik düşüncelerini birçok yolla teknoloji çağına aktardı. Çocuklarda en etkin öğrenmenin modellerden ve örneklerden kaynaklandığı düşüncesiyle, geometrinin yapısal olarak uygulandığı bir ortam yaratmaya koyuldu. Üç yaşındakiler parlak kavuniçi çizgilerle LOGO evrenini ikiye bölerler. Anaokulu çocukları kaplumbağaları 90° döndürerek kareler yapmayı öğrenirler. İlkokul birinci sınıftakiler küçük çapta çizimlere dalarlar, bu arada zevkle, adım adım matematiğe girerler.*

*LOGO birkaç yıldan beri varlığını sürdürmektedir ve önemli ölçüde deneme geliştirme aşaması geçirmiştir. Yine de, ''Zihin Fırtınaları''nın yayınlanmasından, Papert'in Texas Instruments ve Apple ile birlikte yaptığı LOGO'nun yetkinliğini geliştirme çalışmalarından başka, dil M.I.T.'nin araştırma tesislerinin dışına belirgin olarak çıkmamıştır. İki haftalık seminerin düzenlenmesinin nedeni dili ve Papert'in düşüncelerini eğitim çevrelerine aktarmaktı.*

*New York'un her yanından, yayınevlerinden, bilgisayar merkezlerinden, LOGO'nun Eylül'de başlangıç yapmasının hedeflendiği okullardan ve Iowa'daki bir çiftlikten 60 katılımcı Ağustos başında o sabah geldiler. Makina başındaki zamanı arttırmak için iki grup halinde çalıştılar.*

*Her katılımcı günde iki saati Dr. Paperţ'in Piaget'nin öneminden, bilgisayarlaşmanın uluslararası etkisine kadar değişen konulardaki derslerini dinleyerek geçirdi. Bir başka saat, her gün özel bir eğitsel seminerde geçirildi. Üç saat ise, Texas Instruments makinalarında LOGO'nun büyük ölçüde serbest bırakılmış araştırmasına ayrılmıştı.*

*Seminer, teori ve pratiğin güçlü, etkileyici bir bireşimiydi, Katılımcılar -büyük bir çoğunluğu öğretmendi- hergün LOGO evrenini araştırmak için önemli bir fırsat elde ettiler. Konferans salonunu çevreleyen monitörlerde ormanlar doğdu, denizaltılar pusuya yattı, taksiler durdu, yolcu aldı ve uzaklaştı.*

*LOGO'nun gücü ve çekiciliği rengine ya da terminolojisine dayanmaz. Daha çok, girilebilme kolaylığından doğar. Papert, ülkenin önde gelen matematikçilerinden biri olarak kendi alanının en ileri gelenlerindendi. Ancak o çok dar bir kesimce anlaşılabilecek karanlık konularla uğraşarak alanın yapısında olan seçkinciliği artırmak yerine, matematiği içine kolay girilebilir, neredeyse insanın ikincil bir yapısı haline getirmeye yöneldi.*

*Geleneksel matematik eğitimi sayısal şeylere karşı yaygın bir rahatsızlığa ve dirence yolaçarak korkunç bir bedel gerektirmektedir. Papert'in umudu ve inancı odur ki, gelecek neslin matematiğe bakışını ve eğitimin tümünü dönüştürebilecek giriş biçimi erken öğrenilmiş, doğal öğrenilmiş, deneyerek öğrenilmiş matematik ilkeleriyle sağlanabilecektir.*

*Üç saatlik bilgisayar seansları katılımcılara mikrobilgisayarı ve LOGO dilini yakından tanıma fırsatını verirken, çoğu için iki haftalık deneyimi özellikle değerli kılan Papert'in günlük dersleriydi.*

*Papert, tatlı aksaklıkları da olan, tuhaf, şaşırtıcı bir konuşmacıdır. Ona dalgın profesör demek, onu soytarılıkla suçlamayı göze almak demektir. Eksantriklikleri çekiciliğini derinleştirir, anlayışının parlaklığını daha farkedilir kılar. Doğru sözcükleri bulmak için didinirken podyumu dalgınca arşınladığı, düşünürken düşünceleriyle neredeyse güreştiği görülürdü.*

*Dinleyicilerden birini dinlerken düşünceli düşünceli başını sallar, bir yandan da parmağını sakalına sürterdi. Mikrofonla sürekli soru-*

nu vardı. Birkaç kez mikrofon, odayı gök gürlemesine benzer bir sesle doldurarak düştü, ama o, bu gürültüden habersizdi bile. Ortaya çıkmaya başlayan teknolojiden ve bize sağladığı güçten sözetti. Birçok özel varsayımın ve önyargının hatalı olduğunu sessizce gösterdi. Basitleştirilmiş ya da dönüştürülmüş eğitim hakkında konuştu.

## Geleceğin Teknolojisi

Bilgisayar yongalarının üretim maliyetlerinin müthiş düşüşünü görüyoruz. Öyle ki, bilgisayarın kalbi, merkezi işlem birimi artık bir doların altında elde edilebilmektedir. Şu anda, masraf bilgisayardan değil, ekrandan ve klavyeden ileri gelmektedir. Görüntü tübünden, sayısal saattekinden farklı olmayan düzekran TV'ye geçişi sağlayacak teknoloji halen üretilmektedir. Tepsi büyüklüğündeki plastik bir levha hantal klavyenin yerini almak için hazırdır (Çocuk Televizyonu Atölyesi'nde yapılmış bir tanesiyle oynadım). Üretilen miktar artıp düzekranın ve iki boyutlu klavyenin maliyeti düştüğünde, bilgisayarlar 100 dolardan daha ucuz olabilecektir. Bu, beş yıl içinde gerçekleşebilir.

## Zamanımızın Teknolojisi

Eski bir M.I.T. öğrencisi ve halen, bilgisayar/videodisk teknolojisi üreten bir kuruluşun üyesi olan Robert Moll, seminere konuk konuşmacı olarak katıldı. Katılımcının seslenerek, dokunarak ya da odanın bir ucundan işaret ederek denetleyebileceği çok yeni bir sistemin gösterimini yaptı. Ülkenin değişik yerlerinden bir grup yöneticinin aynı odada olduğu sanısını veren telekonferans teknikleri yeni geliştiriliyor. Bir videodisk sistemi, izleyiciyi Colorado'nun Aspen kasabasında, herhangi bir caddeden geçerek, herhangi bir yöne dönerek, yüzyılın başına ait evler hakkında bilgi ve fotoğraf isteyerek, istediğinde kasabayı kuşbakışı görerek dilediğince gezebilmesini sağlıyor. Bu sistemde, istenilen kesitin yalnızca dokunarak seçilebilmesi mümkün.

## Öğrencilerin Bilgisayara Oranı

Papert, daha fiyatların son düşüşü başlamadan önce bile, her çocuğa bir bilgisayar düşmesi gerektiğini savunuyordu. Eğitsel yararlarını ve bilgisayarlı eğitimden geçmiş kitlelerin, yeni atılımlarda üstünlük kurmaya katkılarını kanıtlarla gösteriyor. Bir çocuğun eğitimi'nin halen 25.000-40.000 dolar olduğunu belirtiyor. Şu anda 500 dolara satılan bilgisa-

yarlarla bir çocuk, eğitim masrafının yaklaşık % 5'ine bir kişisel bilgisayar sahibi olacaktır-

hatta arasıra değiştirebilecek veya kalitesini yükseltebilecektir.

## İşgücü Üzerindeki Etki

Papert, Business Week'in yaygın bilgisayarlaşma nedeniyle işleri ortadan kalktığından yeniden eğitime gereksinimi olanları 45.000.000 olarak tahmin ettiğini önemle belirtti. Gelecek yıllarda eğitimin, toplumun birincil ilgi alanı olarak ortaya çıkacağı tahmini üzerinde konuştu. Bununla birlikte, gelecekte öğrenci-öğretmen oranı, yetişkinlerin yeniden eğitimine verilen önem, öğretmenin bireysel rolündeki değişim yerleşik eğitim görüşümüzdekinden çok farklı olabilir.

## Bilgisayar Uzmanlarının Rolü

Matematikçilerin ve bilgisayar uzmanlarının rolü üzerine: Önümüzdeki geleceği yalnız tahmin etmek değil, onu tasarımlamak fırsatı da bulunuyor. Beyaz, erkek ve teknokratik yeteneklere sahip kişiler, geçmişte çok kazançlı olmuş, şimdi de toplumun geri kalanına bütünüyle egemen olmak için beklemektedir. Dil yaratma fırsatına sahip olanların artan zorlukta ve içine girilmesi zorlaşan diller kurma, böylece halen ayrıcalıklı olanların gücünü arttırma olanakları vardır.

Bunun yanında, yabancı olanın gizemini kaldırma, örtülü olanı yalınlaştırma, bilgisayarı içine kolay girilebilir ve demokratik kılma fırsatı da vardır. Papert'in deyişiyle, önümüzdeki seçenekler ekonomik ya da eğitsel olmaktan önce politiktir. Öyleyse, teknolojik gücü, sınırlı bir kesime bırakma veya yayma yeteneğine sahibiz.

## Bilgisayarcılık

Bir dilin, düşüncenin biçimi ve gelişimi üzerinde derin ve karmaşık etki yapması bağlamında, Papert bilgisayarcılığı ırkçılık ve cinsel baskıyla eş görmektedir. Bugüne dek, bilgisayar dili dar bir kesimce anlaşılan erkeksi bir jargon olma eğiliminde olmuştur. LOGO gibi aydınlık bir dünyada bile sisteme yerleşmiş beş biçim (roket, uçak, top, kutu ve kamyon) umulandan ya da tahmin edilen çok daha oğlan/askeri-yönelimlidir. Bu bile, Papert'in gelecek on yılın önemi üzerinde ağırlıkla durmasını doğrulamaktadır. Papert'in beğendiği bir mecaz, QWERTY klavyenin daktilo için geliştirilmiş olmasıdır. QWERTY, en eski, en yavaş daktilolardaki bir sorunun üstesinden gelmek için harflerin yerlerinin düzenlenmesiyle geliştirildi: En çok kullanılan tuşların aralıklarla yerleştirilerek kilitlenmesinden kurtulmak. Teknolojinin böylesi hız sorunlarının çok ötesini ulaşmış olması gerçeği şimdiyi dek, daha akılcı bir klavye düzenlemesine izin vermedi.

Alışkanlıklar ve köklü gelenekler, en eski rastlantısal yargıların varlığını sürdürmesi ve gelecekte de sürdürme eğiliminde olması demektir. Papert, dinleyicilerine şimdiki yargılarının dallanmasını göz önüne almalarını önermektedir. Eğer geleceğin QWERTY'leri olacak fikirler geliştireceksek bunlar, iyi düşünülmüş, uzak görüşlü QWERTY'ler olsun.

## Çocukların Bilgisayarlara Bakışı

Piaget'nin ilk çalışmaları çok küçük çocukların canlılığı hareket etme yeteneğiyle tanımladıklarını ortaya koydu. Buna göre bulutlar canlıydı, fakat ağaçlar değil. Gittikçe artan sayıda araştırma, neyin canlı olduğu ölçütünün hareketten duyguya kaydığını ortaya koymaktadır. Ses ve ışıkla tepki gösteren elektronik oyunlar olan Merlin ve Simon gibi oyuncaklar hem canlı hem de cansız olarak görülürler.

Böyle oyunların ''beyinleri olduğu'' düşünülür, hatta ''aldatmakla'' suçlanırlar (bilgisayarın kazanmadaki duyarsız yeteneği karşısında).

Mattel ve arkadaşlarınca özenle seçilmiş tebrikler ya da çok-kötü tonu veren oyunlar, bu tonları tümüyle yanlış yorumlayan küçük çocuklarda geri teperler-yitirmekten sevinen, kazanmaktan üzülen birini düşünmek bu çocukların deneyiminin ve imgeleminin ötesindedir.

Merlin'in bir üç-taş oyununu kazandığında çaldığı ton sempatik değildir, fakat alaycıdır. Gittikçe canlılara benzeyen bilgisayarlar çağına girdikçe yaşamı neyin oluşturduğu ölçütünün, gerçekten derin ve karmaşık sonuçları olabilir.

## Hard Mastery-Soft Mastery

Papert, son konuşmasının önemli bir kısmın, karşı karşıya koyduğu ''hard mastery'' ve ''soft mastery'' kavramlarına ayırdı. Hard mastery, büyük ölçüde çıktı yönelimli, erkek egemenliğinde, hiyerarşik ve karmaşıktır. Soft mastery ise, adım adım yürüyen işlem yönelimli, basit, etkileşimli ve içine kolay girilebilirdir. Hard mastery, matematik ve fen eğitiminin geleneksel yaklaşımına egemendir-bu konular, somut modeller üzerine kurulmaya veya araştırmaya açık değillerdir. Daha çok, birkaç -eğer varsa- günlük yaşamdan örnekle ustalaşılacak, piramit yapılardır. Kosinüs ve derece, kuantum ve vektör kavramları birçok kişi için şaşırtıcı, hatta iticidir.

New York Kenti Okullar Yönetim Kurulu'ndan yaşlıca, akıllı bir bayanın sözleriyle ''biz, yer-

*leşik eğitimin sarsılmaz hard mastery yapısına karşı sonuçsuz bir savaş veren soft mastery'ye yönelik bir kültüre tanıklık etmeye başladık''.*

*Geleceğin eğitiminin tasarımlanmasında etkin bir rol almamız eğitimle uğraşanlar için önemli bir mücadeledir. Okullar dağılmış ve teknoloji öncesi durumda kaldıkça, seçkin bir azınlığın geleceğin oluşumunu yönlendirmesi küçük bir olasılıktı. Şimdi bilgisayar çağının eşiğinden geçerken ilk yüz yüze geldiğimiz mücadele, en büyük tehlike, bilgisayar dili gibi basit ve birincil bir şeyin bir azınlık tarafından belirsizce yönlendirilmesi ve önemsenmemesidir.*

*LOGO sisteminin yapısı hakkında sorgulanabilecek ve yeniden göz önüne alınabilecek hâlâ çok şey var. Ufak bir girdi karşılığında sesli, renkli ve hızlı çıktı almanın çocuğu daha zorlu mücadelelere karşı isteksizleştireceği bir gerçektir. Özensiz bir incelemeyle, kuşkusuz, böyle çabuk ve basit ödüllendirmenin, dilin etkileşimli yapısına karşın, televizyonun kötülükleriyle koşut bulanlar olacaktır.*

*Papert, bilgisayarların beyaz gömlekli doktorlara ait olduğu mitine inanan küçük çocuklara başlangıç sağlayacak bir dil yarattı. Yine de, Papert'in olağanüstü yeteneklerini biçim ve bilgi kullanımı gibi daha az dinamik, öteki dillerin daha doğrudan sonuçlandırabileceği alanlara yöneltmesini umut etmek için yeterli neden vardır.*

*LOGO'nun işadamlarınca, bilimadamlarınca, araştırmacılarca, hatta bilgisayar hobicilerince kullanılması olası gözükmüyor. Öncelikleri çok fazla bir ağırlıkla grafiğe ve çocuklara yönelik. Yine de, LOGO evreninin müthiş bir çekiciliği var. Bazıları onu basitçe ''çocukların bilgisayar dili'' olarak görmekteler.*

*Aynı şekilde, daha ileri kullanımlarla ilgili sorunlarla uğraşılmadıkça, LOGO zamanla bırakılacak bir dil olacaktır-eğitim bisikletlerinin bilgisayar karşılığı.*

*Eğer böyle olursa, LOGO bu uzun dönemli sorunlara hitap etmezse veya edemezse, LOGO'nun eğlenceli, ama pratik olmayan bir dil olarak görüldüğü 8 yaşından daha büyüklerin daha yerleşik ve karmaşık dillerin kullandıkları bir geleceği rahatlıkla görebiliriz.*

*Anlaşılması güç, jargonlu, sıradan insanı konu dışı bırakacak diller yaratma yeteneğimiz*

*vardır. Bilgisayar egemenliğinde olacak olan geleceği toplumun tüm bireylerine açacak diller yaratma yeteneğimiz de vardır. Bir yanda, Hesse'nin Magister Ludi'sinde öngördüğünden pek farklı olmayan, çok iyi eğitilmiş, erkek egemenliğindeki seçkin bir topluluğun, kendisinden gittikçe geri kalan toplumdan kendilerini soyutladıkları bir gelecek vardır.*

*Öte yanda, kendine özgü riskleri yok değilse de, zenginliği paylaşma, toplumun bütün bireylerinin teknolojinin gücünden yararlanmamasını sağlama fırsatı vardır.*

*Seçkin kesimden olan Papert'in, seçkinciliği yoketmeyi amaçlaması büyük bir umut duymak için bir nedendir. LOGO'nun her kusurunun ortadan kaldırılmamış olması yeni bir dil için şaşırtıcı bir sorun değildir. Papert'in kendi başına her sorunu çözmesi beklenmemelidir. O, başkalarının sürdürecekleri ve geliştirecekleri bir süreci, bir teknolojiyi, bir dünya görüşünü harekete geçirdi. Bunu yaparken, de kaplumbağadan, şeytandan ve bir avuç ekran cininden çok daha güçlü ve sürekli bir şey yaşam verdi.*

*Richard Eyster*
*Creative Computing, Aralık, 1981*



# LOGO Nedir?

Bu soruya çeşitli yollarla cevap vermeyi kararlaştırdık. Cevapların her biri LOGO'nun farklı veçhelerinin yarattığı meselelerle uğraşıyor. Her cevap ilave sorular doğurur. Ek sorular, dili kullanmamıza imkân verecek çeşitli bilgi düzeyleri açısından temel nitelikte olmamakla beraber, yine de bilgisayarlar ve bilgisayar kullanımı alanında kavrayışınızı ve anlayışınızı geliştirmenize yardımcı olacaktır.

## LOGO Bir Dildir

BASIC, mikro bilgisayarlar için geliştirilmiş -muhtemelen- en çok tanınan ve en sık kullanılan dildir. Neredeyse her kişisel bilgisayar türü için farklı bir versiyonu olduğunu biliyoruz. LOGO, Forth, Pascal ve Fortran öbür genel bilgisayar dillerine verilen adlardır. Yazılı diller gibi, bilgisayar dilleri de, bir dizi kurallara dayalı olarak kavranabilir bir sıra ile birbiri ardınca gelen kelimeler ve sözdizim kaidelerinden oluşurlar. Herhangi bir veri lisanı öğrenmek, o dilin hem kelime dağarcığını, hem de kelimelerini birbirlerine bağlayacak sistemi edinmek demektir. Mesela, Türkçe'de ''onun bir gün güzel ikigün'' ifadesi bir sürü hatayı içeren bir kullanımdır. Kelimelerin dizilişi (gün güzel) hatalıdır; ve ''ikigün'' gibi mevcut olmayan bir kelime kullanılmıştır. Türkçe öğrenmekte olan bir öğrenci böylesi bir cümle sarfettiğinde karşısındakiler onu anlamaya çabalarken, bilgisayarlar çok daha az affedicidirler ve sizin ne söylemeye çabaladığını yorumlamak için kesinlikle en ufak bir gayret sarfetmezler. Yukarıda örneğini verdiğimiz hatalardan herhangi biri, bilgisayarın ne söylendiğini hiç anlamaması ve mesajı hepten reddetmesi için yeterli sebep olur. İnsan okuyucunun aksine bilgisayarlar sizin niyetlerinizi tahmin etme kabiliyetine sahip değildirler. Bu yüzden de açık ve kesin olmak özellikle önem taşır.

Kelime dağarcığının kapsamı, esnekliği ve hacmi önemlidir ve lisanın konuşmayı kolaylaştırıp kolaylaştırmadığını büyük ölçüde belirler. İnsan dillerindeki bu olgunun bir örneği, Eskimolar'ın lisanında kar'ın farklı durumlarını anlatmak amacıyla çok fazla sayıda kelime olmasıdır. Kendilerini çevreleyen yüzey şekilleri ile tabiî manzaranın çok mühim bir tezahürü olan kar'ın değişik hallerini çeşitli ayrıntılarıyla incelikli bir şekilde dile getirebilmek

Eskimolar için hayati önem taşır. Halbuki Türkiye'de kar, çevrenin o kadar da ehemmiyetli bir parçasını oluşturmadığı için, kar'ın birbirinden farklı olabilecek durumlarını izaha kalkıştığınızda, Türkçe'nin size pek fazla yardımcı olamadığını keşfedersiniz. Yumuşak, sulusepken, veya yarı erimiş karı anlatabilirsiniz ama, bu bile, Eskimolar'ın kar'ın her bir ayrı hali için apayrı kelimeler kullanmaları ile kıyaslandığında, işiniz hiç de kolay değildir. Çoğu bilgisayar dilleri kendi aralarında rakamlar ile ilişkileri hakkında kolayca konuşabilirler; örneğin SIN (sinüs), COS (kosinüs) ve SQR (karekök) BASIC'in sayısal fonksiyonları tarif için kullandığı örneklerdir.

LOGO'nun komut yapısında yansıyan iki özel ilgi alanı vardır. İmajları ekranda üretmeye yarayan belli genişlikte bir kelime dağarcığı, (örneğin FORWARD-ileri, BACK-geri, RIGHT-sağ, LEFT-sol, PENUP-kalem yukarı, PENOOWN-kalem aşağı) ve dilin çeşitli yönlerini tasvir eden öbür kelimeleri mevcuttur (kelime, cümle, liste). LOGO ayrıca inceden inceye sayısal çözümleme imkânları da sağlar, ama çoğu bilgisayar dillerinin aksine bu faaliyeti, uğraşının odak noktasını oluşturmaz.

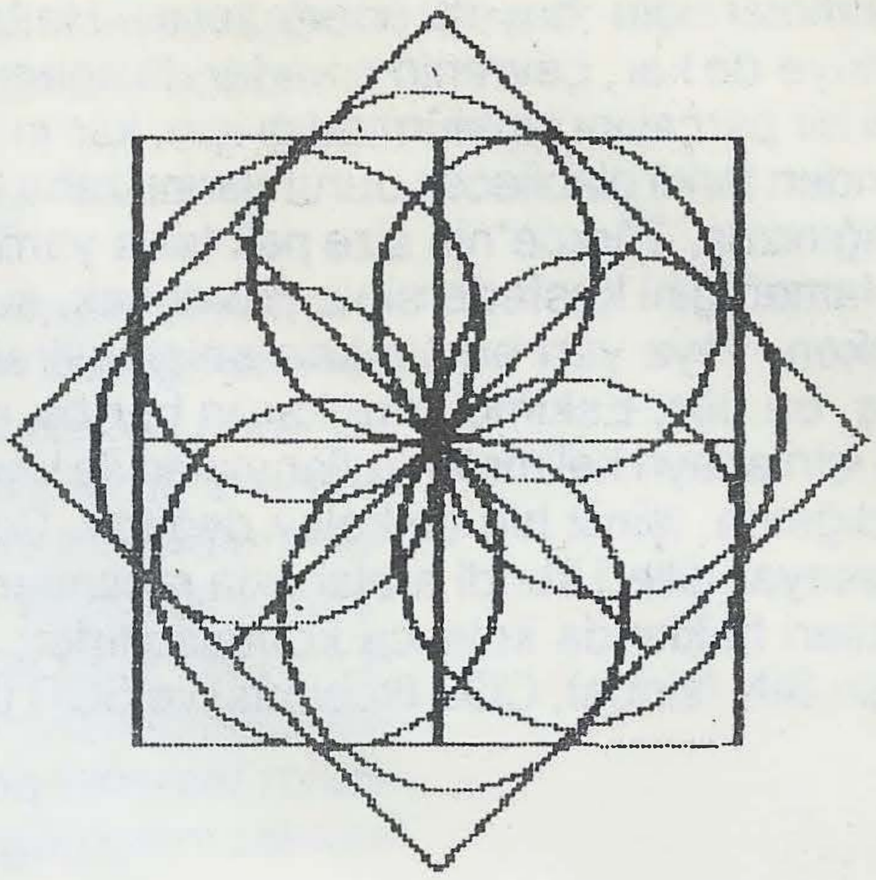
## Çocuklar İçin Bir Dil Olarak LOGO

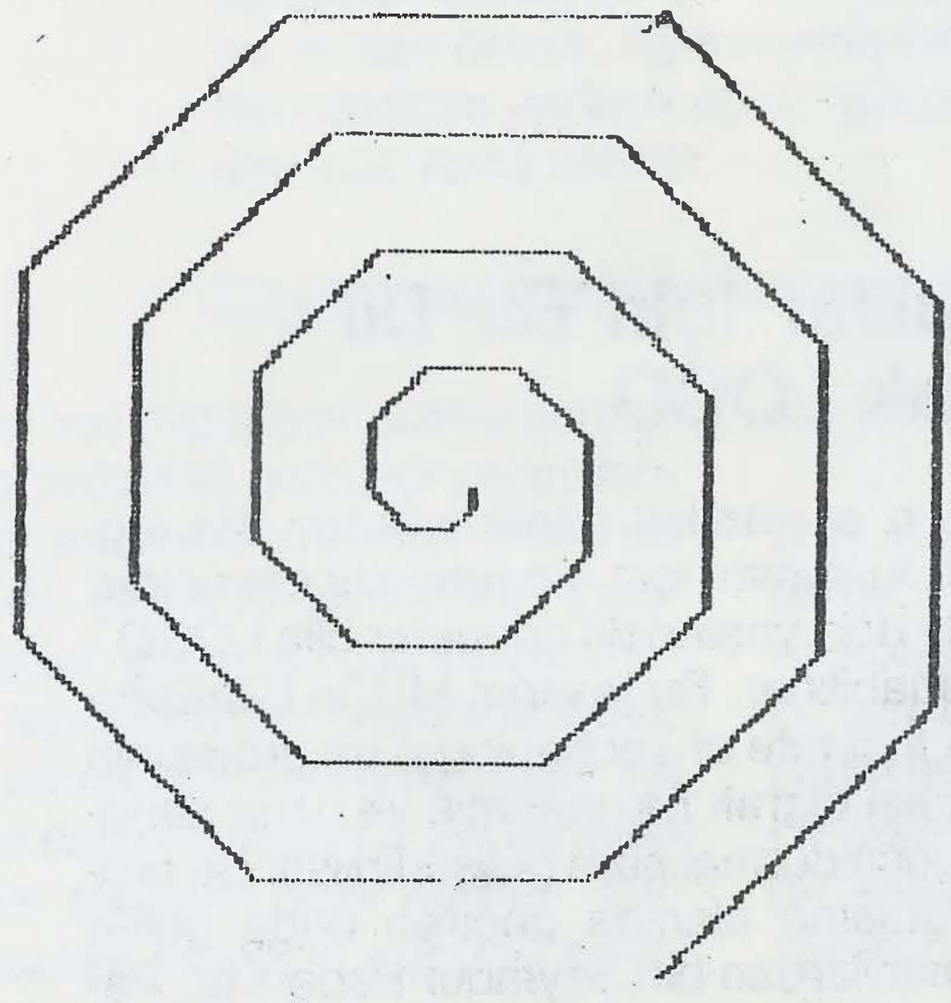
Logo'nun en radikal niteliklerinden biri, çocukların kullanımı için tasarlanmış olmasıdır. Özellikle dört yaşındaki çocuklar bile LOGO'yu kullanabilirler. Bu tasarım MIT'in (Massachusetts Institute of Technology) bir projesinin bir parçası olarak hazırlanmış ve uzun seneler boyunca durmaksızın geliştirilmişti. Bu projeyi uygulama alanına geçiren önde gelen araştırmacılardan biri Seymour Papert'tir. Papert'in kitabı **Mindstorms: Children, Computers and Powerful Ideas** LOGO'nun gerisinde yatan felsefeyi bir hayli ayrıntılı şekilde tartışır.

Başlangıçta LOGO tıpkı öbürleri gibi bir bilgisayar dili idi: Yani, özellikle kelimeler, kelime ya da harf listeleri ve rakamlarla iş görürdü. Papert kısa sürede, üzerinde uğraştığı oniki veya onüç yaşındaki çocukların kullandığı bilgisayar işleme kavramlarının çoğunun, uygun bir biçimde ifade edildikleri takdirde daha küçük yaştaki çocuklar tarafından da kullanılabileceğini farketti. Neticede, "**Turtle**" (tosbağa) adı verilen sistemi buldu. Bu, ekranda gezindikçe nerelerde bulunduğunu gösteren izler bırakan bir şekildir. Bu şeklin çıkarabileceği bazı kalıplar arkadaki sayfada gösterilmiştir.

**Turtle** bilgisayar ile yapılabilecek ilginç bir konu yaratır. Böylelikle çocuklar Turtle'ın yeteneklerini araştırırken bir taraftan da bilgisayar işlemenin disiplinini ve yöntemlerini öğrenirler. Bu genç çocukların öğrenme sürecinin çeşitli veçhelerini de yansıtır. Çocuklar mesela







''rakamlar'' gibi can sıkıcı bir biçimde çok incelikli tarif edilmiş şeyleri pek seyrek öğrenmeye kalkışırlar. Çocuklar için öğrenmek, kendilerini çevreleyen dünyayı bir tür keşfe çıkmaktır. LOGO onlara keşfedilecek bir bilgisayar dünyası sunar.

# LOGO'yu Öğrenmek

## Neler Gerekir

LOGO, fizik görünümü ile bir kaset veya disk durumunda iken, onu herhangi bir bilgisayar yazılımından ayırdetmek mümkün değildir, ama uygulama olanakları onu kendine mahsus bir sınıfa koyar. Eldeki ''eğitim'' bilgisayar programlarının çoğu bilgisayara yüklendiğinde, bilgisayarın ortaya çıkardığı sorulara nasıl cevap verileceğine dair talimatlar verir - bu bazı kelimeler yazarak veya belli toplamlara cevap vererek olabilir. Bir başka deyişle, bilgisayarın ekranda gösterdiği, çocuğun bilgisayar ile ne yapacağını belirler.

Bu eğitim programlarının sunduğu ödev daha ziyade oyma testeresi ile kesilmiş tahta parçalarından ibaret bilmece oyununu andırır - hani şu, bir sürü parçalar vardır da resmi tamamlayabilmek için, her birini doğru yerde kullanmak gerekir, işte ona. Bilmeceyi tamamladığınızda, geriye resmi bozmak ve yeniden yapmaktan başka bir iş kalmaz. LOGO ile durum hayli farklıdır. Çocuğun önüne bir mesele koymak yerine, yani eksik kelimeleri bulmasını istemek yerine, LOGO çocuğa bilgisayarın anladığı bir kelimeler demeti ile bunları biraraya getirmesini sağlayacak kurallar sunar.

Bunları kullanarak çocuk, bilgisayarın kendi istediğini yapmasını sağlayabilir. Bu tıpkı, çocuğun eline kocaman bir kutu Lego vererek kendi fikirlerini ve kabiliyetlerini geliştirebilmesine imkân hazırlamaya benzer.

Şimdiye kadar gösterdiğimiz üzere, LOGO belli bir kelime dağarcığına ve sözdizimine -ya da bu kelime dağarcığını birleştirmeye yarayan bir yönteme- sahip bir bilgisayar dilidir.

LOGO oynarken başvurulan ''kurallar'', mesela Uzay Fatihleri (Space Invaders) oyununun ''kuralları''ndan çok daha karmaşıktır.

Ama, mükafatları da çok daha büyük olur! Bir kere dile hakim oldunuz mu, bilgisayara istediğinizi yaptırabilirsiniz, hem de ''oyun''un sizi zorladığından daha basit bir şekilde.

Bu özellikler şu anlama geliyor: İster ebeveyn, ister öğretmen olarak LOGO'yu çocuklara öğretirken, eğer çocuk LOGO'dan azami derecede istifade etmek istiyorsa, bir miktar sizin yardımınıza ihtiyaç duyacaktır. Bu bölümün başlıca konusu, öğretmenliğini yaptığınız çocuğa ya da çocuklara birlikte keşfine çıkabileceğiniz çeşitli fikirler kazandırmaktır.

Bölüm boyunca fikirler genel bir düzen içinde sunulmuştur. Bu genel düzen, bilgisayara nelerin yazılması gerektiğini açık seçik gösterir, ayrıca tecrübe edilecek veriler (rakamlar, kelimeler) için de önermelerde bulunur. Dokuz ya da daha yukarı yaşlardaki çocuklar için bu programları bilgisayara yazmak ve büyüklerin asgari yardımı ile programların çalışmasını sağlamak kolay olmalıdır. Böyle olmakla beraber, çocuk şayet bu fikirleri ve programları genişletmeyi arzu ederse, yine de büyüklerin ek bir yardımda bulunması gerekebilir.

Bu tıpkı herhangi bir yabancı dili öğrenmeye benzer - öğrenme işini kendi başınıza gerçekleştirebilirsiniz, ama konuşacak hiç kimse bulamazsanız, bilginin kavrayabildiğiniz miktar ve edindiğiniz tecrübe hayli mahdut kalacaktır.

## LOGO'nun Kullanımı Ne Şekil Alır?

İzleyen bölümlerde LOGO'da mevcut olan kelime dağarcığı anlatılacaktır. LOGO'nun bazı özelliklerini canlandırmak ve kullandığı bazı dil yapılarını göstermek amacıyla burada ürettikleri kalıpları gösteren birkaç program ve çizim verilmiştir. Bu ve öbür programların nasıl çalıştığı daha sonraki bölümlerde bulunabilir. Aşağıdaki ilk üç örnek, Turtle'ı, ekranda şekiller ve dizaynlar çizmek için kullanıyor. Dördüncü örnek, Logo'nun İngilizce'yi keşfetmek için sahip olduğu potansiyeli canlandırıyor.

```
TO SQUARE
FORWARD 60
RIGHT 90
FORWARD 60
RIGHT 90
FORWARD 60
RIGHT 90
FORWARD 60
RIGHT 90
END
```

Bu program bilgisayara nasıl bir kare şekli çizileceğini öğretiyor. Siz bunu bilgisayara yazdığınızda mikro bilgisayarın sayfa 1888'deki kalıpları yapmaya yarayan maharetlerinden istifade edebilirsiniz. Bu çizim Turtle'ı kareyi çizme işlemini bitirdikten sonraki konumda gösteriyor. Resimler Turtle olmadan daha doğru düzgün göründüğü için, bu metinde yer alan çoğu çizimlerden Turtle'ı çıkarmaya karar verdik.

Bu programı bir kere yazmanız yeter. Başlangıçtaki TO bilgisayara sizin ona bir şeyin nasıl yapılacağını öğretmeye hazırlandığını söyler. Programı yazmayı tamamladığınızda sadece SOUARE (Kare) yazmanız Turtle'ın sayfa 1888'deki gibi kareyi ekrana çizmesine yetecek de artacaktır bile.

```
TO CIRCLE
REPEAT 36 [FORWARD 10 RIGHT 10]
END
TO PATTERN
REPEAT 8 [CIRCLE SQUARE RIGHT 45]
END
```

Bu program sizin bilgisayara evvelce öğrettiğiniz iki şekli kullanıyor: Kare ve daire çizmeyi. Bunu sekiz defa tekrarlayarak sayfa 1888'de gösterilmiş ikinci kalıbı elde ediyor.

```
TO SPIRAL :SIDE :ANGLE :INCREMENT
FORWARD :SIDE
RIGHT :ANGLE
SPIRAL :SIDE + :INCREMENT :ANGLE :INCRE
  MENT
END
```

Bu programı yazdığınızda, HELEZON 3 45 2 yazarak çalışmasını sağlayabilirsiniz. Bu sayılar: YAN: AÇI ve :NİCELİK ARTIŞI kelimelerinin yerine kullanılır. Program üçüncü kalıbı oluşturuncaya kadar tekrar tekrar çalışır.

```
TO TALK
PRINT [NASILSINIZ?]
MAKE "X READLIST
IF :X = [İYİYİM] [PRINT [BUNU DUYDUĞUMA
  MEMNUN OLDUM] STOP]
IF :X = [PEK İYİ DEĞİL] [PRINT [ÇOK YAZIK
  İNŞALLAH YAKINDA KENDİNİZİ DAHA İYİ
  HİSSEDERSİNİZ] STOP]
PRINT [SAHİ Mİ!]
END
```

Bu program, mesajları ekrana yazar ve onları cevaplandırmanızı bekler. Karşılıklı konuşma muhtemelen şöyle olacaktır:

```
NASILSINIZ?
? İYİYİM
BUNU DUYDUĞUMA MEMNUN OLDUM
```

veya,

```
NASILSINIZ?
? DİZİM AĞRIYOR; MASAYA VURMUŞTUM
  DA.
SAHİ Mİ!
```

Bu programlar LOGO'nun bir dil olarak sahip bulunduğu önemli bazı veçhelerini canlandırıyor. Kullanmaya başlamazdan önce, bu veçhelerin neler olduğunu tespit etmekte de yardımcı olabilirler. LOGO'yu öğrenen çocuklar açısından bu özellikler hayati bir önem taşımazlar. Ama, dilin içinde düşüneceği çerçeveyi **oluştururlar** ve LOGO'yu öbür dillerle kıyaslamamıza yardım edecek bir kıstas sunarlar.

## Etkileşimlilik

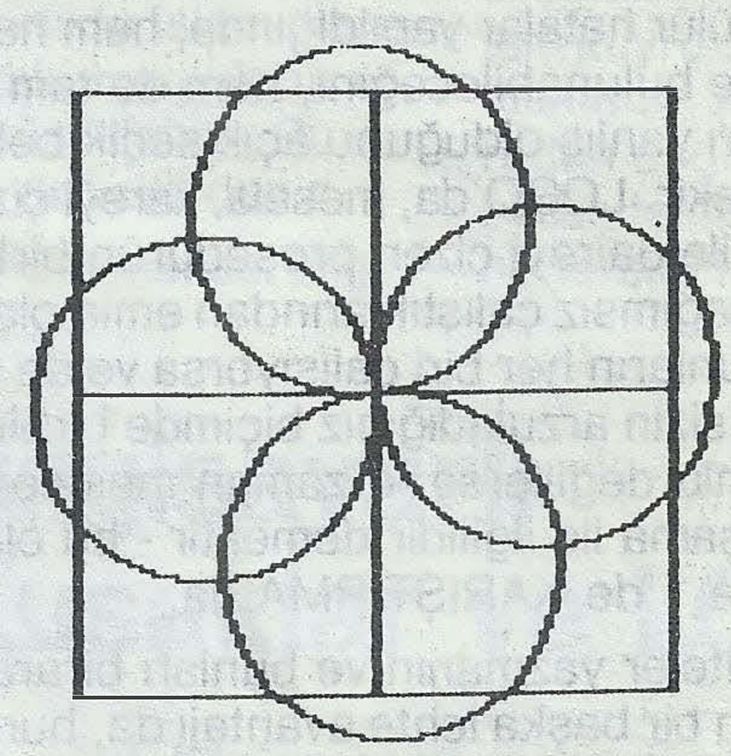
BASIC ile LOGO'yu öbür eski bilgisayar dillerinden ayırdeden özelliklerinden biri, sizin herhangi bir şeyi denemenize ve bu tecrübenin sonuçlarını derhal görmenize imkan tanımalarıdır. Bu nokta eğer programı çocuklar kullanacaksa özellikle önemlidir; çünkü bütün çocuklar hemen kontrol edebilecekleri çabuk sonuçlar görmeye ihtiyaç duyarlar. Bazı bilgisayar dilleri, en basit ödevleri tamamlamak için dahi sizi, bilgisayar komutlarının kalıplarını sıkı sıkıya takip etmeye mecbur bırakırlar. LOGO ise, sizin bir tek komut yazmanıza imkan tanıyacaktır: RETURN veya ENTER anahtarına (bilgisayarınızda anahtarın adının ne olduğuna bağlıdır bu) basınız, komutunuzun gereği hemen yerine getirilecektir. Mesela, ÖNE 20 (RETURN) yazdığınız anda hemen Turtle'ı 20 birim öne hareket ettirecektir.

## İşlem Usulü

Bilgisayar programını BASIC'te yazdığınız zaman, bir program bir tek işlem yapar. Bir oyun oynamak, hesaplarınızı yapmak, plak koleksiyonunuzu kaydetmek ve buna benzer işlemler için bir program yazmayı seçebilirsiniz. Bu ödevlerin herhangi biri bir tek basit program tarafından yerine getirilir. Karmaşık programlar yüzlerce satır uzunluğunda olabilirler. Herhangi bir bilgisayar programında olduğu gibi, bir noktadaki hata tek bir programın çalışmasını engelleyebilir. Bu hataların (''Debugging'', bir sistemin hatalarını giderme) bulunması çok uzun zaman alabilir ve bir hayli zorluk çıkarabilir. Bilgisayar jargonunda BASIC gibi bir dil, ''yapılanmamış'' kelimesiyle ifade edilir.

Oysa ki LOGO ''yapılanmış'' bir dildir. Bu da, belli bazı ödevleri yapabilmeniz için ''prosedürler'' ismi verilen küçük programlar yazmanız anlamına gelir (yani, yukarıda gösterildiği üzere, HELEZON, KARE gibi...). Yukarıda ortaya atılan (öne sürülen) büyük programların uygulanabilmesi, mesela bir oyun oynamak beş veya on, hattâ daha da fazla sayıda küçük prosedürlere ihtiyaç gösterebilir.

Bu yöntem sanki insana pek çok iş çıkarıyormuş gibi gelebilir, ama aslında bir sürü potansiyel zorluğun önünü alır. Her prosedür genellikle bir tek özel işlem yapar. Yukarıda gösterilen şekli çizmek isterseniz, bilgisayara bir karenin ''NASIL'' çizileceğini öğreten prosedürü yazarak bu işi yapabilirsiniz: Yani, prosedür ''KARE'' ile başlarsınız ve sonra da DAİRE izleyerek. Bir üçüncü prosedür ki biz buna ''KARIŞTIRMA'' ismini verebiliriz, bu iki prosedürü bir araya getirebilir. KARE, DAİRE ve KARIŞTIRMA böylece üç prosedürün adı olur. Bir kez Turtle'a bir şey yapmayı öğrettiniz mi, artık bundan sonra sadece prosedürün adını yazarak aynı şekli tekrar ve tekrar çizmesini sağlayabilirsiniz.



Bu örneklerde KARE ve DAİRE isimleri Turtle'ın yukarıda gösterildiği gibi kareler ve daireler çizmesini temin edecektir. KARIŞTIRMA prosedürü kalıbı oluşturmak amacı ile bu şekilleri biraraya koyar. Önce, KARE prosedürünü kullanarak dört kare çizer. Sonra, Turtle'ı 90 derece sola döndürür ki ilk daireyi çizmeye başladığında bu daire yukarıda gösterilen noktaya yerleşebilsin (isabet edebilsin?). Daha sonra DAİRE prosedürü ile alakalı dört daire çizer. Bu prosedürlerin birbirleri ile nasıl uyuştuğu aşağıdaki diagramda görülebilir:

```
LEVEL 1                              LEVEL 2
TO MIX
REPEAT 4 [SQUARE RIGHT 90]           SQUARE
                                     REPEAT 4 [FORWARD 50 RIGHT 90]
                                     END

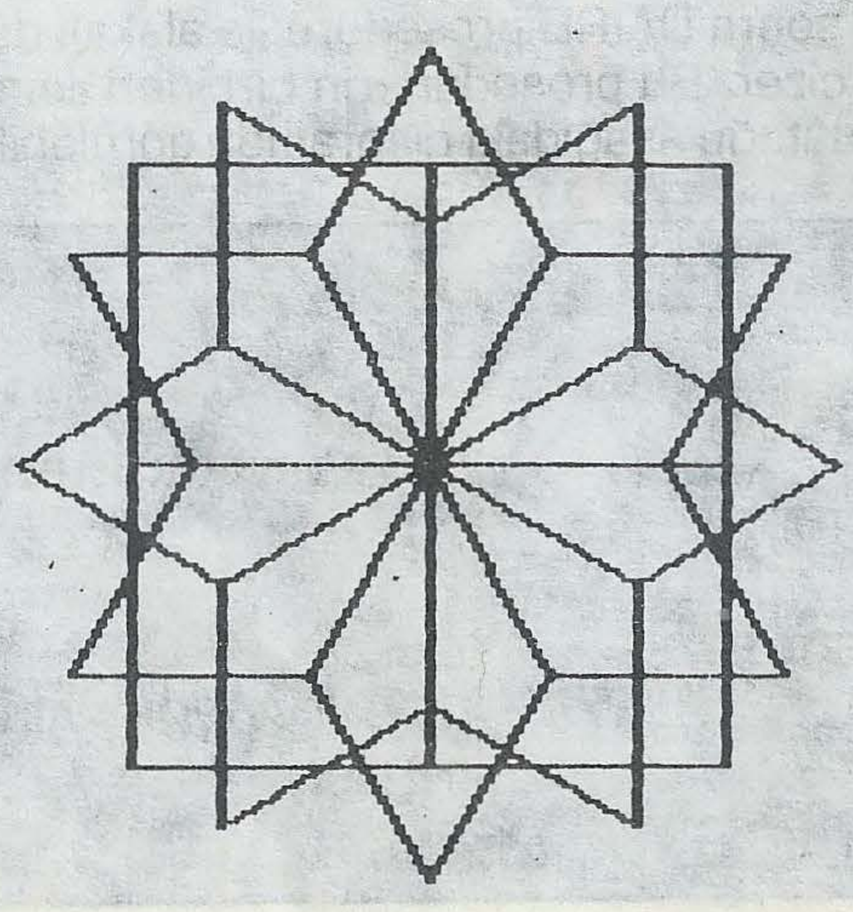
LEFT 90
REPEAT 4 [CIRCLE RIGHT 90]           CIRCLE
                                     REPEAT 36 [FORWARD 5 RIGHT 10]
                                     END

END
```

Yukarıdaki diyagram, programların birkaç aşamada ayarlanabileceğini -bu olayda iki kere- gösteriyor. Aşama 2 kareler veya daireler çizer. Aşama 1 aşağısındaki aşamayı, arzulanan etkiyi yaratabilmek amacıyla kullanıyor. KARIŞTIRMA KARE prosedürüne ihtiyaç duyduğunda, onu (KARE) adıyla ''çağırır''. KARE prosedüründe SON'a ulaşıldığında, bilgisayar KARIŞTIRMA'daki bir sonraki komut uyarınca çalışır (bu olayda bu talimat SAĞA 90 idi). 4 DEFA TEKRARLA bilgisayara bir kare çizmesini ve sonra dört defa sağa dönmesini söyler. Prosedürleri bu şekilde kullanarak, her prosedür en alt aşamada ayrı olarak test edilebilir. Bu nokta, bilgisayarın kullanımını basitleştirmek açısından çok önemlidir.

Gözle görülür hatalar yapıldığında, hem hatanın nerede bulunabileceğini, hem de tam tamına neyin yanlış olduğunu açık seçik belirtmeniz gerekir. LOGO'da, mesela, kareyi çizen prosedür ile daireyi çizen prosedürün birbirlerinden bağımsız çalıştıklarından emin olabilirsiniz. Bunların her biri çalışıyorsa ve de şekiller hâlâ sizin arzuladığınız biçimde birbirleriyle uyumlu değillerse, o zaman mesele bir sonraki aşama ile ilgilidir demektir - bu olayda, Aşama 1'de KARIŞTIRMA'da.

Küçük üniteler yazmanın ve bunları biraraya getirmenin bir başka lehte avantajı da, bunlara pek çok çeşitli durumlarda başvurulabilmesidir. Mesela, aşağıdaki kalıp, aynı KARE prosedürünü alıp ve bir başka PIRLANTALAR (DIAMONDS) prosedürü onunla birlikte kullanılarak yapılmıştır. Tek tek prosedürleri bu yolla karıştırmak bilgisayar işlemede pek çok vakit kazandırdığı gibi, böylece programlamanın daha da kolay yapılabilmesini sağlar. Çünkü LOGO ''prosedürel'' bir dildir.



## Geliştirilebilirlik

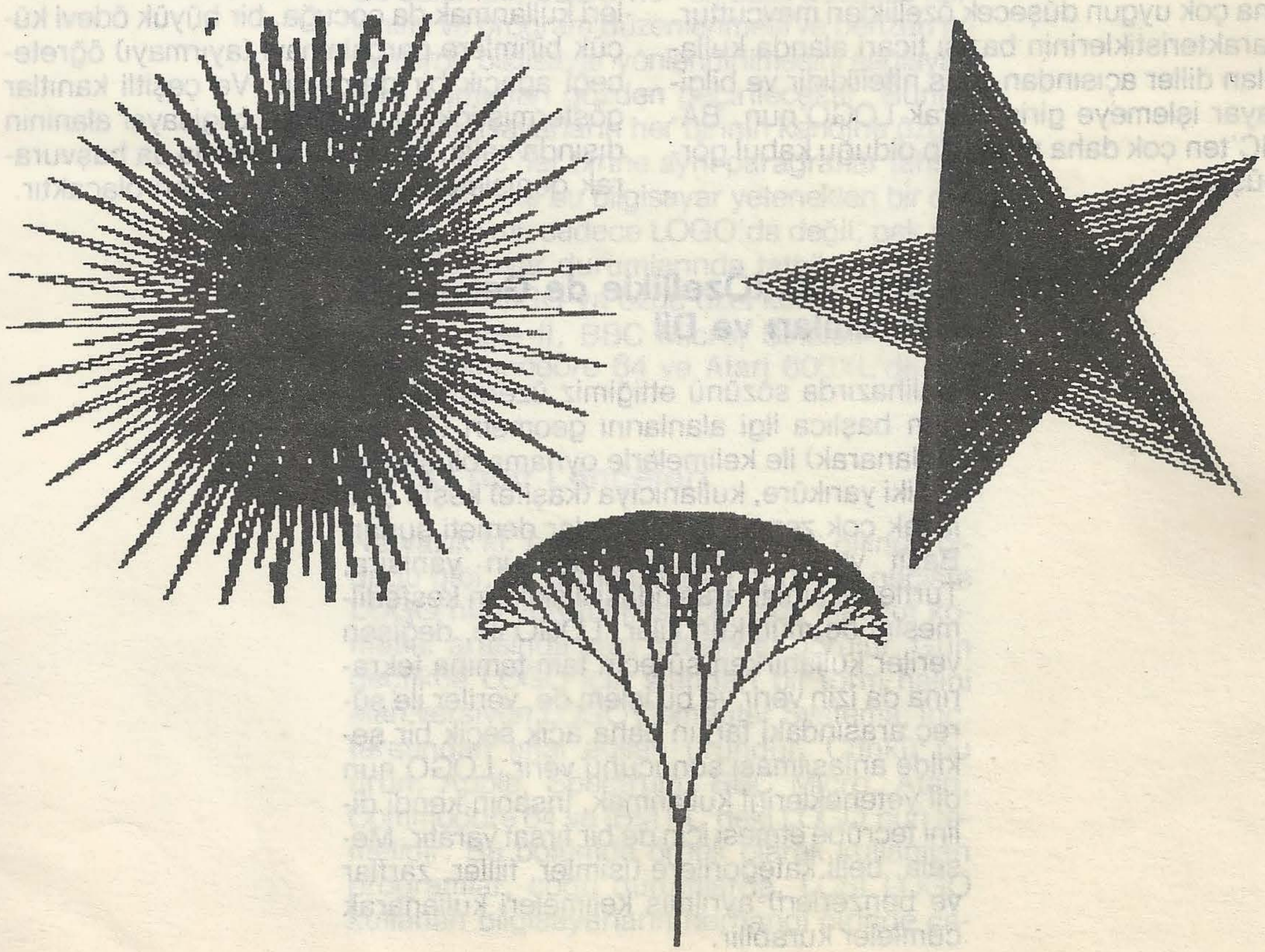
Bir şekil çizmek veya bazı belirli ödevleri uygulamak üzere bir prosedür yazdığınızda, prosedürün adını yazarak bilgisayarın istediğiniz işlemi yapmasını sağlayabilirsiniz. Böylece mesela, bilgisayara bir kare çizmesini örğettiyseniz belirlediğiniz kareyi çizmesini ''KARE'' kelimesini kullanarak sağlayabilirsiniz. Bu da, prosedürün ismini (bu olayda KARE) tıpkı LOGO dilinde mevcut olan kelimeleri (ÖNE, SOL ve benzerleri) kullandığınız gibi kullanıyorsunuz demektir).

İşte bu özelliğe bilgisayar jargonunda "**extensible**" (geliştirilebilir) deniyor. Ve sizin dile ekleyeceğiniz kelimeler de LOGO'nun ilk oluşturulmuş kelime dağarcığı gibi davranacaklardır. Dilin bu özelliği bazı olumlu yanlar taşıyor; ki bunların en önemlisi, tek bir kelime ile bir prosedürler demetine atıfta bulunabilmesi. Bu avantaj, size prosedürleri yeni bir komut gibi işlemden geçirerek, bir grup programı bir başkasının içinde kullarıma yeteneği kazandırır.

KARIŞTIRMA prosedürünün diyagramı, bir prosedürün öbürlerini çağırmak için nasıl kullanıldığını gösteriyor.

## Yinelemelilik (Recursivity)

Sizin yaratabilmenize imkan veren etki türleri açısından çok kuvvetli bir özelliktir bu. Genellikle, bir bilgisayarın, belli bir prosedürün sonuna ulaştığında kendi adını taşıyan bir satır ile karşılaşmasına ''recursion'' adı verilir. Bu isim bilgisayara aynı prosedürü tekrarlamasını söyler. Aynı şeyi tekrar ve tekrar yapmak faydalıdır, ama özel olarak fevkalade bir tarafı da yoktur. Recursion'un en önemli özelliği, bilgisayara aynı süreci tekrarlama komutunu vermesidir - **ama,** her seferinde farklı kelimeler veya sayılar ile. Böylece, prosedürün sonuna gelindiğinde ve tekrardan çağırıldığında, bilgisayara aynı çalışmayı yeniden yapması komutunu her seferinde sayıları veya kelimeleri değiştirerek süreç boyunca verdiğinizde, aynı eylemler aynı sıra ile vukû bulur, ama farklı veriler ile (sayılar veya kelimeler). Hiç kuşkusuz sizin de farkettiğiniz üzere, bu yöntem çok çeşitli sonuçlar elde edebilmeyi mümkün kılar. Sizin recursive programlar marifetiyle üretebileceğiniz bazı çizimler aşağıda gösterilmiştir. Programlar çok muhtasar olabilir ve BASIC'teki muadillerinden çok daha kısadırlar.



# LOGO Ne Öğretir?

## Bilgisayar Okur-Yazarlığı

Çok çeşitli anlamlarda kullanılan bir terimdir bu. En geniş manada bilgisayar dilini (dillerini) kullanabilme yeteneği anlamına gelir. Bu tarifte, LOGO bilgisayar okur-yazarlığı yeteneklerini can-ü gönülden geliştirir. Bir dil olarak LOGO, genellikle ticari işyerlerinde pek kullanılmaz. Bunun sebepleri temelde tarihîdir; kısmen de, bilgisayar programlarının mali tahminlerde bulunmasına, kelime-işlem yeteneklerine sahip olmasına ve benzeri işlemler yapabilmesine ihtiyaç duyan çoğu büro sistemlerinin gereklerinden ileri gelir. Şüphesiz, bu çalışmalar LOGO'da da yapılabilir, ama bu işlere kalkışıldığında çok can sıkıcı olabilir ve çok yavaş yürüyebilir.

Sanayide genellikle pek rağbette olmayan bir bilgisayar dilini çocuklara öğretmek ilk nazarda tuhaf görünebilir. Böyle olmakla beraber, tıpkı dershane kimyasının sanayi dünyasındaki kocaman teknelerden ziyade deney tüpü ile imbikte gerçekleştirildiği gibi, bilgisayar işlemeye de en iyi LOGO'yu kullanarak girizgâh yapılabilir. Bölüm 9'da tartışacağımız gibi LOGO'nun okul çocuklarının heves ve meraklarına çok uygun düşecek özellikleri mevcuttur. Karakteristiklerinin bazısı ticari alanda kullanılan diller açısından esas niteliklidir ve bilgisayar işlemeye giriş olarak LOGO'nun, BASIC'ten çok daha münasip olduğu kabul görmüştür.

## Düşünme

Bilgisayarları ister eğlence için, ister çalışma için kullanın, tüm kullanıcıların ortaklaşa sahip oldukları bazı disiplinler mevcuttur. Hepsi de bilgisayara bilgi, mantık yüklerken, kayıt yaparken -başarılı bir çalışma için- malzemenin (verilerin) inceden inceye doğru olmasına dikkat etmenizi mecburi kılar. Bu, hem bilgisayar oyununun kullanımı, hem de metin işleme için doğrudur (geçerlidir). Değerleri konusunda hiçbir kuşkuya yer olmayan bu disiplinleri öğretmenin yanısıra LOGO daha fazla şeyler de öğretebilir. LOGO'yu kullanmak öbür bilgisayar programlarının içermediği düşünme ve öğrenme yeteneklerini geliştirir.

Bunun böyle olmasının başlıca nedeni, çocuğun bilgisayara KARE yapmasını ya da YAZ'masını öğretmesidir. Çocuk bilgisayara istediklerini yaptırmaya çabalarken, bilgisayarın neyi bilmeye ihtiyaç duyduğunu ve çalışma sırasındaki işlemleri hangi sıra dahilinde yapması gerektiğini keşfetmek zorundadır. İşte, mantıkî düşüncenin gelişmesinin temelini de bu keşfetme mecburiyeti oluşturur. Prosedürleri kullanmak da çocuğa, bir büyük ödevi küçük birimlere parçalamayı (ayırmayı) öğreteceği apaçık bir gerçektir. Ve çeşitli kanıtlar göstermiştir ki, bu kabiliyet bilgisayar alanının dışında kalan başka durumlarda da başvurarak genişletebilecekleri bir beceri olacaktır.

## Matematik (Özellikle de Geometri) Kavramları ve Dil

Halihazırda sözünü ettiğimiz üzere, LOGO'nun başlıca ilgi alanlarını geometri (Turtle'ı kullanarak) ile kelimelerle oynama oluşturur. Bu iki yarıküre, kullanıcıya (kaşife) keşfe çıkılacak çok zengin bir imkanlar demeti sunar. Basit yarıkürelerin sunuluşunun yanısıra, Turtle rakamlar arasındaki ilişkilerin keşfedilmesini de mümkün kılar. LOGO ile, değişen veriler kullanırken sürecin tam tamına tekrarına da izin verir ve bu işlem de, veriler ile süreç arasındaki farkın daha açık seçik bir şekilde anlaşılması sonucunu verir. LOGO'nun dil yeteneklerini kullanmak, insanın kendi dilini tecrübe etmesi için de bir fırsat yaratır. Mesela, belli kategorilere (isimler, fiiller, zarflar ve benzerleri) ayrılmış kelimeleri kullanarak cümleler kurabilir.

Çocuk bu tip bir programla oynarken İngilizce sentaksa ait bazı yanlış anlamalar gün ışığına çıkar.

## Deney Yapma

LOGO'nun çocuklara sunduğu en orijinal ve faydalı kolaylık -muhtemelen- bir çevreyi denetleme ve onun nasıl çalıştığını keşfedebilme kabiliyetidir. Bu tür bir durum, tabii dünyada benzeri kolay kolay bulunmayacak bir şeydir.

Bilhassa, okul çağındaki bir çocuk için, hangi deneyi yaparsa yapsın, her seferinde aynı neticeyi verdiği pek enderdir. Bir bilgisayarda yaratılmış şartlar ise, istenen süreçleri her zaman aynen tekrarlayacaktır. Mantıkî açıdan tutarlı bir çevre temin edildiğinde, LOGO çocukların mantıkî deneyler planlamasına ve Turtle'ın yaptıklarına sebeplerini görmesine veya kurdukları problemlere bilgisayarın verdiği cümlelerin ya da matematik çözümlerin hangi nedenlere dayandığını görmelerine yardım edebilir.

## Sonuç

LOGO,, çocuklara özel olarak uygulanabilen, çok yönlü bilgisayar dilidir. Bölümün geri kalan kısımlarında, LOGO'nun sağladığı kelime dağarcığı ve aynı zamanda da çeşitli yaşlardaki çocuklarla LOGO'nun kullanılabileceği bazı yollar canlandırılacaktır.

Çoğu çocuklar LOGO'ya Turtle grafik komutları ile başlarlar. Hiçbir şekilde mecburi olmamakla birlikte, bu programlama güzergahı, bilgisayarınıza her ne yaptırdıysanız etkilerini görebilmeniz için bir araç sunar. Ancak Turtle'ı bu şekilde kullanarak henüz sayı ve sözcükleri işleyemezsiniz. Bunları LOGO'nun temel kavramlarını açıkladıktan sonra ilerki bölümlerde anlatacağız..

## Turtle Grafikleri

Bu bölümde, LOGO'nun en basit düzeyde kullanımı ve program düzenlenmesi ve benzeri işlemlerde bilgisayar yönlendirilmesini sağlayan bazı komutları gözden geçirilecektir. Bunlar farklı bilgisayarların her birinin kendine özgüdür ve yine her birine aynı paragraflar tahsis edilmiştir. İşte bu bilgisayar yetenekleri bir defa edinildi mi, sadece LOGO'da değil, pek çeşitli bilgisayar durumlarında tatbik edilebilir. Yine de, LOGO'ya özgü özel komutlar ve tesirleri Apple II, BBC Micro, Sinclair Spectrum, Commodore 64 ve Atari 600XL'de ele alınmıştır.

## LOGO'nun Lehçeleri

Ne yazık ki, hemen her bilgisayar dilinde olduğu gibi, bir bilgisayardan öbürüne geçişte LOGO'nun yürütülmesi için elde mevcut komutlar arasında bazı farklılıklar görülür. Gün geçtikçe LOGO için standart bir form halini alan versiyon LOGO Computer Systems, Inc. tarafından imal edilen üründür. Çünkü bu ürün, Apple, Spectrum, BBC Micro, Atari, Commodore 64 ve IBM PC'deki LOGO'nun temelidir. Bu bölümde örnek olarak kullanılan programlar, çoğu durumlarda, LCSI LOGO kullanan bilgisayarların herhangi birinde ça-

lışacaktır. RML 380Z ve TI 99/4A için olan LOGO bazı önemli farklılıklar arzeder. TI LOGO, ekranda dolaşabilen ve LOGO'ya özgü hareketli ''ekran cinleri'' **(sprites)** yaratır. Aynı derecedeki bir başka makinada bu özellik görülmez; Atari 600'ün LOGO'su daha az mikyasta olmakla beraber benzer ve becerilere sahiptir. TRS-80'in LOGO'su sadece Turtle grafiklerine münhasırdır, ama bu alanda çok geniş imkanlar sunar.

İzleyen prgoramlar LCSI LOGO'da yazılmıştır ve adı geçen bilgisayarlarda işleyecektir. Turtle grafik olanların çoğu da öbür bilgisayarlarda kullanılabilecektir.

## Karşılıklı Modda Turtle

LOGO'yu bilgisayarınıza yükler yüklemez ''LOGO'YA HOŞ GELDİNİZ'' kabilinden bir mesajla karşılaşacaksınız. İşte LOGO'nuz kullanılmaya hazır sizi bekliyor. Şayet Turtle'ı kullanmak istiyorsanız SHOWTURTLE (veya kısa için ST) yazmanıza gerek vardır. Küçük bir üçgen veya yukarı doğru işaret eden benzer bir şekil, aşağı yukarı ekranın ortalarında bir yerde belirecektir. Bu durumda, buna HOME konumu denir. Turtle'ı bir başka yere hareket ettirir ve HOME yazarsanız, Turtle bu noktaya geri dönecek ve her daim yukarı dönük olacaktır. Üçgenin işaret ettiği yön çok mühimdir, çünkü siz kendisine hareket etme komutunu verdiğinizde Turtle'ın ne yöne gideceğini belirler.

Turtle da (tüm öbür hayvanlar gibi) ÖNE veya ARKAYA hareket eder ve bu özelliği, siz ona bilgisayar ekranında yukarı veya aşağı hareket etmesini söylediğinizde herhangi bir bulanık anlamın doğmasını engeller. Turtle hareket ettikçe, daha önce hangi noktalardan geçtiğini size haber veren bir çizgi bırakır arkasında. Komutu tamamlayabilmek için Turtle'e kaç adım atacağını bildirecek bir rakam vermeniz gerekir, mesela: ÖNE 50 (RETURN) DÖN 30 (RETURN) ve bu böylece devam eder gider. Böylece Turtle sizin belirlediğiniz birim kadar işaret ettiği yönde ileri gidecektir. Eğer HOME konumunda ise, ÖNE komutu onu ekranda kullandığınız ekranın boyutlarına bağlı olarak bir ya da iki inç yukarı kaldıracaktır.

Turtle'ın yapabileceği çizimler düşünüldükte bu işlem hayli sınırlı bir şeydir; dimdik dikine bir hat yeteneklerinin gelip dayandığı nokta ise! Faydalı öbür iki komut SAĞ ve SOL'dur. Bu komutlar ile birlikte kullandığınız rakamlar, Turtle'ın ne derece sağa veya sola döneceğini belirler. Böylece, SAĞA 90 (RETURN) Turtle'ı 90 derece döndürür ve yüzünü sağa çevirir. ÖNE 50 (RETURN) Turtle'ı öne doğru ve sağa 50 birim ileri hareket ettirir.

SAĞA 270 besbelli ki SOLA 90 ile aynı tesiri yapacaktır. SAĞA ve SOLA komutlarının Turt-

le'ı sadece çevirdiğine ama asla ileri (öne) doğru hareket ettirmediğine dikkat ediniz. LOGO'da çok sık kullanılan bir misalde olduğu gibi, bir kare çizmek için, aşağıda gösterilen komutlara başvurulabilir. Komutların kısaltılmış biçimleri, bilgisayara yazım işlemini kısa sürede tamamlayabilmek için kullanılmaktadır.

```
FD 70
RT 90
FD 70
RT 90
FD 70
RT 90
FD 70
RT 90
```

RETURN ya da ENTER anahtarlarına (FD 70 veya RT 90 gibi) her bir komuttan sonra basılmalıdır. Komutların kısa aralıklarla birbirlerini izlemeleri birkaç ilginç mesele yaratır. Bir kere, Turtle'ın kare çizme işlemini fiilen en son RT 90 komutunu girmeden evvel tamamladığını göreceksiniz. Bazı hallerde Turtle'ın bir şeklin çizimini tamamladığında, başlangıçta olduğundan farklı bir yöne bakıyor olmasını arzu edebilirsiniz. Genellikle, yine de Turtle'ın çizimin sonunda başlangıçtaki ile aynı yönde bulunmasının daha yardımcı olduğunu keşfedeceksiniz. Bunun böyle olduğundan emin olmak için, işin nihayetinde prosedüre son bir SAĞA 90 komutunu eklemek makuldür.

Nihai SAĞA 90 ilave edince Turtle'ın 360 derece dönerek yürümesini sağlamışsınız demektir. Bu işlem, Dairevi Turtle Seyahati Teoremi **(Total Turtle Trip Theorem)** adı verilen teoremde izah edilecektir. Eğer Turtle herhangi bir bölgenin sınırları etrafından tam bir yolculuk yaparsa ve bunu başladığı durumda tamamlarsa, tüm bu hareketin toplamı 360 derece olacaktır.

LOGO ile oynarken, bu teoremin, Turtle'ın neden beklediğiniz şekli çizmediğini bulup çıkarmakta çok faydalı olduğunu göreceksiniz.

Yöntem ile sonuç arasındaki fark, Turtle'ın karşılıklı modda kullanılmasının yarattığı bir başka meseledir. Yukarıdaki kare çizimi öne doğru hareket ve sonra sağa dönüş ile elde edilmişti. HOME konumundan başlayarak, tam tamına aynı çizim şu yolla yapılmıştı:

```
RT 90
FD 70
LT 90
FD 70
LT 90
FD 70
LT 90
FD 70
```

Dairevi Turtle Seyahati Teoremi'ni uygularsanız, Turtle'ın sadece 180 döndüğünü görürsünüz. SAĞ dönüş SOL dönüşlerin biri tarafından iptal edilmiştir, ki böylece geriye yalnızca iki SOL dönüş kalır. Gerçekten de, Turtle sonuç-

ta ekranın dibine dönük vaziyette ve başlangıç konumundan 180 derece mesafede kalır. Bununla birlikte, karenin çizimi tam tamına önceki örneğe benzer. Böylece, sonuç tamamen aynı olsa da yöntem bir hayli farklıdır. Bu durum Turtle -genel olarak bilgisayarlar- kullanılarak yapılan çoğu şeyler için geçerlidir. Bazı yöntemlerin öbürlerine kıyasla daha hızlı çalıştığını ve daha derli toplu olduğunu, ama sonucun her durumda aynı çıkabileceğini kısa sürede farkedeceksiniz.

Öbür yönlerden, bilgisayarlar yine de affedicidirler: Beklediği ile sizin ona verdiğiniz arasındaki farkı görmemezlikten gelebilir. Turtle'ı karşılıklı modda kullanmak, bilgisayara kat'i bilgi vermek mecburiyetinde olduğunuzu öğrenmeniz için iyi bir tecrübe sağlar. Bilgisayar, ne FD20'ün ne de F D 20'ün neyin nesi olduğunu anlamaz. Eğer Turtle ile karşılıklı modda oynarken, bilgisayara verdiği bilginin mutlaka ve mutlaka doğru bilgi olması gerektiğinin öğrenmesine yetecek zaman harcanırsa bilgisayarlarda meydana gelen bazı yanlış anlamalar engellenebilir. Bilgisayarı karşılıklı modda kullanma hakkında son bir noktaya daha işaret edelim:

Buraya kadar anlattıklarımızdan da anlaşılabileceği gibi, Turtle'ı bu şekilde kullanırken, çizimleri düz çizgilerle oluşturmak eğrilerle çizmekten çok daha kolaydır. Eğriler LOGO'nun başka imkanlarının çiziminde çok yararlıdır ki bunu ileride işleyeceğiz.

## Turtle'ın Eğitimi

Bu bölümde şimdiye kadar Turtle'ın karşılıklı modda kullanımını -verdiğiniz komutları anında yerine getirmesi halini- inceledik. Bilgisayara tekrarlanmasını istediğiniz şeyleri öğretmek yararlıdır - böylece, her seferinde aynı bilgiyi yazmak zorunda kalmadan arzuladığınız işlemin yapılmasını sağlayabilirsiniz. Birinci Bölüm'de gösterildiği gibi, bu iş, bilgisayara bir şeyi **nasıl** yapılacağı öğretilerek gerçekleştirilir ve "**prosedür**" adı verilir. Eğer bilgisayarın kare çizmesi örneğini ele alırsak, o zaman prosedür şu şekilde yazılabilir:

```
TO SQUARE
FD 70
RT 90
FD 70
RT 90
FD 70
RT 90
FD 70
RT 90
END
```

İlk dikkat edilmesi gereken şey, başlangıçtaki TO'dur. Bu ibare, bilgisayara, belli aralıklarla bizim belirleyeceğiniz komutlar vereceğinizi gösterir. SQUARE (kare) de önemli tabiî; bu da, sizin kullanacağınız prosedürün adını gösterir. Komutlarınızı birbiri ardınca yazdığınız zaman, RETURN anahtarınıza rağmen Turtle artık sizin tek tek komutlarınızı yerine getirmediğini göreceksiniz. Şimdi sadece kursörü gerisin geriye yeni bir hattın başına getirecek, böylece sizin bir sonraki komutu yazmanıza hazırlanacaktır. Komutları yazma işini bitirdiğinizde, bu özel prosedür için gerekli komutların hepsini yazıp bitirdiğinizi anlatmak amacıyla SON yazmayı unutmamanız gerekir.

Daha sonra karşılıklı moda dönersiniz ve Turtle'ı evvelce olduğu gibi kullanmaya devam

edebilirsiniz. Böyle olmakla birlikte, şimdi KARE yazarsanız, Turtle kareyi, kenarları 70 birim uzunlukta olmak üzere, belirlediğiniz şekilde çizecektir. Bu Turtle ekranın neresinde olursa olsun çizilebilir ve Turtle'ın ''home'' konumunda olmasını gerektirmez. Şüphesiz karenin ekrana sığamayacağı durumlar da olabilir, ama Turtle yine de kareyi çizmeye çalışacaktır. Burada, KARE'ye tekrar TO SQUARE yazarak başvurmamanız önemlidir. Eğer böyle bir şey yaparsanız, KARE prosedürünün zaten belirlenmiş olduğunu anlatan bir mesaj alırsınız. KARE için bilgisayara ne türlü komut verirseniz verin, bu hepsi için geçerlidir. Kenarları 70 birim uzunluğundaki bir kare için komut vermişken, bir kenarını 40 birim olmasına karar verirseniz, bilgisayara yeniden FD 70'in yerine geçmek üzere FD 40 komutunu vermenin gerekir. Ve tabiî bu kareye yeni bir isim de vermeniz icap eder (muhtemelen, KARE 40). Böylece ÖNE (FORWARD), SAĞ ve SOL komutlarını kullanarak, Turtle'ı ekranın her tarafında oynatarak çeşitli kareler çizebilirsiniz. Aynı süreç öbür şekiller için de tekrarlanabilir ve böylece bir dizi resim veya çizimin yapılması sağlanabilir. Bunu tecrübe etmeye değer ve hayli ilginç çizimler ortaya çıkarılabilir.

## Prosedürlerin Adlandırılması

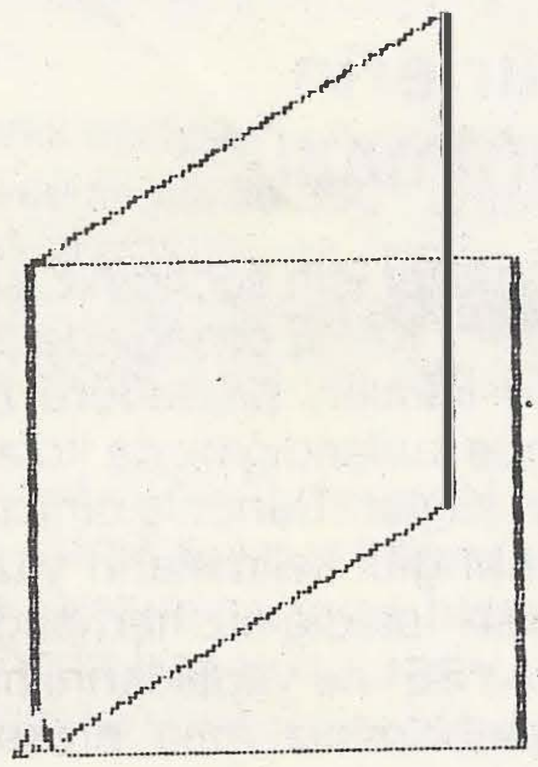
Yarattığınız şekiller için seçtiğiniz isimler tasvir edici olabilir - KARE örneğinde olduğu gibi. Bu tasviri isimler, prosedürü bir başka program içinde kullandığınızda kolaylıkla hatırlayabilmeyi sağlar. Bununla birlikte, PARALELPROGRAM gibi kelimelerin yazımı biraz incelikli olabilir! İstediğiniz herhangi bir ismi -NURİ veya AYŞE- ne yaptıklarını hatırlamak şartıyla kullanabilirsiniz. Ama, prosedürün ismini bir başka şey için kullandığınız kısaltma haline getiremeyeceğinizi aklınızda tutun. PARA PARALELPROGRAM'ı karşılıyorsa, aynı zamanda PARADOKS'un da karşılığı olarak da kullanılmamalıdır. İki prosedür istiyorsanız o zaman her ikisine de ayrı isimler vermelisiniz. Prosedür isimlerine rakamlar ve semboller ekleyebilirsiniz, ama aralık veremezsiniz, yani boşluk bırakamazsınız. Bir örnek olarak, farzedelim ki, TWİSTSHAPE adlı bir prosedürünü çağırmak istiyorsunuz; bunun için twist ile shape arasına bir nokta (TWİST.SHAPE) koyarsınız. Böylece, bilgisayar bunun tek bir isim olduğunu anlayacaktır.

## Prosedür İçinde Prosedür

Prosedürler kendi başlarına olduğu gibi, öbür prosedürlerin içinde de kullanılabilir. KARE prosedürünü bir baklava (pırlanta şekli) çizmek için de uygulayabilir ve arka sayfada gösterildiği üzere başka bir çizimin parçası haline getirebilirsiniz. Kare'nin nasıl çizileceğini evvelce belirlediğinize göre, yalnızca BAKLAVA (DIAMOND) prosedürünü ve bu ikisini bir araya getirecek prosedürü belirlemenize ihti-

yaç vardır. Şayet Turtle'ın BAKLAVA'yı (DIAMOND) 60 derece döndürecek ilk açıyı yaparsanız, sonra ikinci 120 dereceyi de yapmanız gerekecektir. Kare ile baklava prosedürlerini bir araya getiren prosedüre PART adını verebilirsiniz.

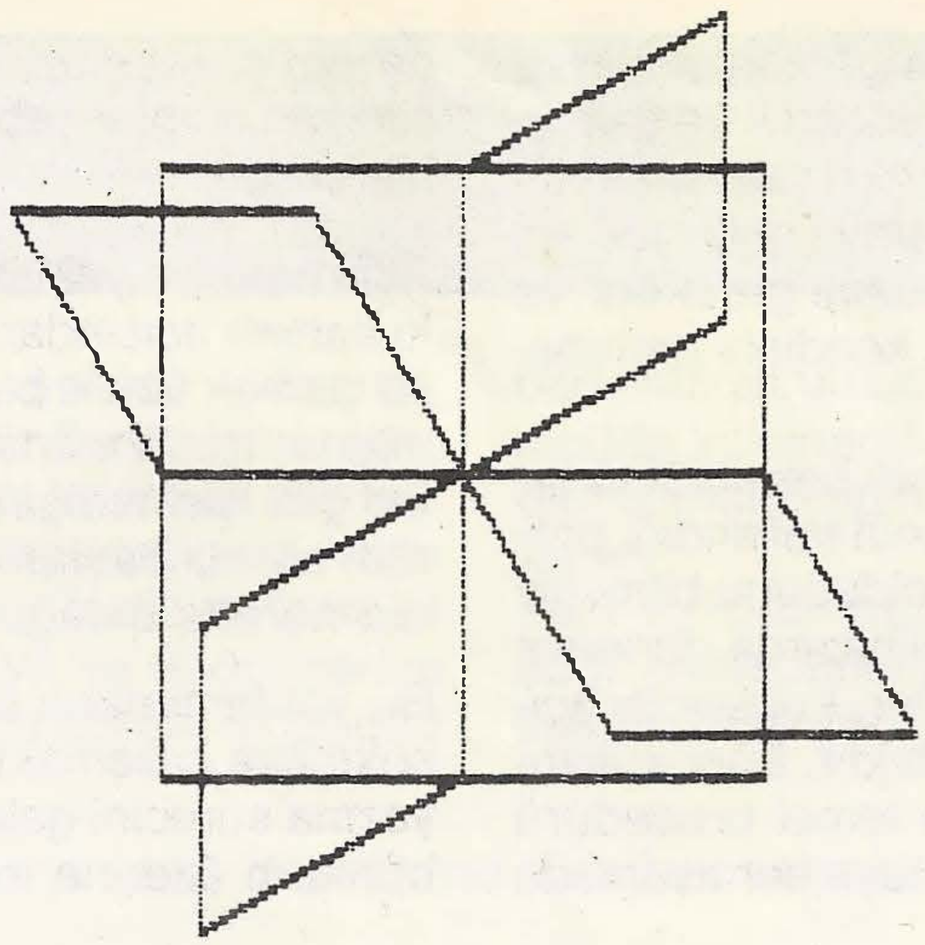
```
TO PART
SQUARE
DIAMOND
END
```



Şimdi aşağıda gösterilen çizimi yapmak için PART'ı kullanabilirsiniz.

```
TO PATTERN
PART
RT 90
PART
RT 90
PART
RT 90
PART
RT 90
END
```

Açıyı ve PART'ın kaç defa tekrarlanacağına dair sayıyı değiştirmek çok çeşitli tesirler yaratır. Bu hayli basit örneği alır, ne olduğunu ve bazı ayrıntılarıyla nasıl yazıldığını kavramaya çalışırsak, ortaya çıkan süreç, prosedür kullanan bilgisayarcılığın bazı temel ilkelerini canlandıracaktır.



Bilgisayarın kalıbı üretmek için bilgisayarın neler yaptığını gösteren bir diagram çizerseniz, muhtemelen şöyle bir şey ortaya çıkacaktır:

```
1. AŞAMA
TO PATTERN
PART

2. AŞAMA
TO PART
SQUARE

3. AŞAMA
TO SQUARE
FD 50
RT 90
FD 50
RT 90
FD 50
RT 90
FD 50
RT 90
END
TO DIAMOND
FD 30
RT 60
FD 30
RT 120
FD 30
RT 60
FD 30
RT 120
END

DIAMOND
END

RT 90
PART

TO PART
SQUARE

TO SQUARE
```

En basit prosedürler sağda gösterilmiştir ve siz prosedürlerin her birini yazarak ve çalışıp çalışmadıklarını denetleyerek başlayabilirsiniz. Bunların bir kısmını bir araya getirmek, en aşağıdaki iki aşama çalışmasını gerektirir ve siz de bunu PART yazarak kendiniz denetleyebilirsiniz.

Şayet KALIP prosedürü şimdi beklediğiniz işlemi yapmazsa, asıl meselenin kullandığı prosedürlerde değil, KALIP'ta olduğunu bilmeniz gerekir. Çünkü, hepsini halihazırda denemiş bulunuyorsunuz. Prosedürler, kullanıcısı açısından çok mühim bir avantajdır. Eğer kullanmadan evvel en mühim ve temel prosedürü denerseniz ve sonra denemeye her aşamada devam ederseniz, meselenin hangi noktada ortaya çıktığını çabucak ve kolayca keşfedebilirsiniz.

Alternatif bir yaklaşım ve belki de çocukların kullanımı açısından daha tabiî bir yol, bir kalıp çizmek üzere bir program yazmaya başlamaktır. İşin sonuna geldiklerinde ve bekledikleri gibi işlemediğini keşfettiklerinde, o zaman tüm süreci baştan sona kadar kat edip hatanın nerede olduğunu bulmak zorundadırlar.

Bu yöntemi daha ileri bir aşamada işin içine sokmaya çalışmaktan ziyade, küçük birimler yazma sürecini geliştirmek ve baştan itibaren bunların üzerine inşa etmek daha iyidir.

## Tekrarlama Eylemleri

Şu ana kadar çok sayıda prosedürün, aynı tür komutları tekrar tekrar yazma ile ilgili olduğunu görüp içinize fenalıklar gelmiş olabilir. Mesela, KARE'yi tarif ederken FD 70 ile RT 90'ın yazımını dört defa tekrarladınız. Önceden görülebileceği gibi, bu işi daha kısa yoldan yapabilecek bir yöntem vardır. Aslında bu işlemi yapacak, çok ufak farkla arzeden, iki yol mevcuttur.

### Tekrar

Biraz önce bahsettiğimiz üzere, KARE prosedürünü tarif ettiğinizde FD 70 ile RT 90'nın yazımını dört defa tekrarladınız. Bir başka deyişle, karenin her iki kenarı öne doğru belli bir mesafededir ve sağa döner. Bu ilke herhangi bir şekil için geçerlidir. Dolayısıyla, bir eylemin ya da eylemlerin kaç defa tekrarlanmasını istiyorsanız, prosedürü şöyle yazabilirsiniz:

```
TO HEXAGON
REPEAT 6 [FD 50 RT 60]
END
```

Turtle'ın tekrarlamasını istediğiniz eylemler mutlaka parantez [] içinde ve doğru bir sıra halinde yazılmalıdır. Hat üstünde ve parantezlerin dışında TEKRAR komutunu ve işlemin kaç defa tekrarlanmasını arzu ettiğinizi yazın. 60 derece açısı "Dairevi Turtle Seyahati Teoremi"ne dayandırılarak kolaylıkla hesaplanabilir: Çünkü, altı köşeli şekil ile her açının değeri 360 derece altıya bölünür, ki bu da 60 derece sonucunu verir. Altı'dan en iki kenara kadar tüm şekillerin açılarını hesaplamak çocuklar için hayli eğlenceli bir alıştırma olabilir.

TEKRAR'dan bu bağlamda olduğu kadar, bilgisayarınıza öğrettiğiniz prosedürlerin tekrarında da yararlanabilirsiniz. Yukarıda örneği dönersek, KALIP şu yolla da üretilebileceğini görürüz.

```
TO PATTERNA
REPEAT 4 [PART RT 90]
END
```

TEKRAR komutunun bu şekilde kullanımı, prosedür tamamlandığında Turtle'ın nerede, hangi noktada bulunacağını kesinlikle bilecek-siniz demektir. Turtle'ın arzu ettiğiniz kalıbı çizdiğinden emin olmak için, Turtle'ın KARE'yi tamamlandığında, BAKLAVA'ya ve öbürlerine geçmezden evvel ne tarafa dönük olduğunu bilmeniz zorunludur... İşte, Turtle'ı karşılıklı modda kullanarak etrafta 360 derece dolandırma alışkanlığını geliştirmenin yararı buradadır.

Yeni kalıplar yaratmak amacıyla, dizaynlar öbür prosedürlere gitgide daha çok başvurdukça, daha da çok disiplin gerekir.

## Kuyruklu Yineleme (Tail recursion)

TEKRAR'ı kullanmak istemezseniz, işte size bir prosedürün ya da bir dizi eylemin birkaç defa tekrarlanmasını sağlayacak alternatif bir yöntem.

**Tail recursion** hayli karmaşık bir işlemin adıymış gibi gelse de, temelde çok basit bir süreçtir. Bir işlemin kaç defa tekrarlanmasını istediğinizi kesinlikle bildiğiniz durumlarda, yukarıda izah edilen TEKRAR yöntemi çok daha iyidir. Ama, Turtle'ın herhangi bir şeyi kaç defa yapacağını *bilmediğiniz* durumlarda ise, tail recursion faydalıdır. Şöyle çalışır: Ilk evvela, eskiden olduğu gibi, prosedürün ismini belirlersiniz:

```
TO PATTERN1
PART
RT 5
PATTERN1
END
```

Bu sefer, kalıbı oluşturan birimin nasıl çizileceğini Turtle'a söyledikten sonra (ki bu olayda PART ve beş derecelik sağa dönüş) tüm işlemi tekrar ve tekrar yapmasını sadece kendi ''ismi''ni -KALIP1- çağırarak söylersiniz.

Bu tekrarlama işlemi siz bilgisayarı durdurunca kadar devam edecektir. Bunun nedeni, programın sonunda bilgisayarın her KALIP 1'e ulaşmasında gerisin geriye başlangıca dönmesi ve böylece hiçbir zaman gerçekten SON'a varamaması ve Turtle'a da durmasını söyleyememesidir. Siz bilgisayara durmasını söyleyebilirsiniz: Apple'da kontrol anahtarı ile G'ye beraber basarak; Spectrum'a CAPS ve BREAK'e; Atari'de BREAK'e ve BBC'de de ESCAPE anahtarlarına basarak. Anahtarı bilmiyorsanız kullanıcı el kitabı sizin özel bilgisayarınız için ne yapmanız gerektiğini gösterecektir. Bunu yaptığınızda sürecin hangi noktasında duracağınızı kesinlikle denetleyebilmeniz uzak bir ihtimal. Bu sınırlılık hali, kalıp veya resim yaparken bloklar oluşturmak için başvurabileceğiniz kare ve altıgen gibi şekillerin tarifinde tail recursion tekniklerinin faydasını çok azaltıyor. Kendi içinde tamam olan bir kalıp ve kaç kare tekrarın gerekli olduğunu tam olarak kestiremediğiniz kalıplarda, tail recursion en iyi yöntemdir.

## Bir Daire Çizimi

Daha önce de sözünü ettik: LOGO ile düz çizgiler çizmek, eğriler çizmekten daha kolaydır. Bununla birlikte, ekranın hassasiyeti daire gibi görünen bir şeyin çok kısa çizgiler çizilerek ve birkaç derece çevirerek üretilebilmesi anlamına gelir. Bunun üzerinde düşünmenin bir başka yolu ise, öbür şekilleri çizerken Turtle'ın 360 derece döndürürsünüz, başka türlü dendikte bir "devrim" yaparsınız. Dolayısıyla, Turtle biraz öne çizip sonra da azıcık sağa çevirirseniz ve başlangıç noktanıza varana dek ilerlemeye devam ederseniz, elinizde bir adet çizilmiş daire olacaktır. Aşağıdaki prosedür tam tamına bu yolla bir daire üretecektir.

```
TO CIRCLE
REPEAT 36 [FORWARD 10 RIGHT 10]
END
```

Kaç derece çevirmeniz gerektiğini, TEKRAR için kullanılan sayıyı 360 dereceye bölerek hesaplayabilirsiniz. Şayet TEKRAR 18 ise, o zaman 20 derece çeviriniz; eğer 72 ise o zaman da 5 derece çeviriniz. FORWARD ile kullandığınız numarayı, oluşturulacak dairenin boyutunu belirler. Bazı daireler ötekilere kıyasla daha inandırıcı görünür. O zaman amacınıza en iyi hizmet edecek bileşim sayısını tecrübe ede ede bulmak daha iyi olur.

DAİRE prosedürü ile, tüm prosedürü tekrar yazmadan çok çeşitli sayıları tecrübe etmek yararlıdır ve bu iş değişkenler kullanılarak yapılabilir. Bu olgunun alakalı olduğu çok sayıda durum sözkonusudur. Birincisi, prosedürü belirlerken, değiştirmek isteyebileceğiniz sayılar için bilgisayar isimleri vermeniz gereklidir. Mesela daireyi alırsak, o zaman şöyle bir hal alabilir:

TO CIRCLE: NUMBER: SIDE: ANGLE

Bir kelimenin hemen önünde kullanılan iki nokta üst üste (:) bu kelimenin değişkenin ismi olduğunu ve prosedür içinde geçtiği zaman da bu değişkenin değerinin kullanılacağını gösterir. Geçtiği zaman da prosedürün başlığında kullanıldığında ise daha ziyade sınıflandırmada bir etiket gibidir. O zaman tüm prosedür şöyle görünürdü:

```
TO CIRCLE :NUMBER :SIDE :ANGLE
REPEAT :NUMBER [FORWARD :SIDE RIGHT :ANGLE]
END
```

Bir başka ifade ile, prosedürü ilk yazdığınızda, etiketlerin adlarını yazarsanız -KENAR ya da -AÇI gibi. Prosedürü tekrar çağırdığınızda ise bu etiketlere tekabül eden sayıları yazmak zorundasınız. Bilgisayar bundan sonra prosedür uyarınca çalışacaktır. Şayet Acornsoft LOGO kullanıyorsanız, prosedürün başlığında kullanılan değişken isimlerinin başında ' bulunur (iki nokta üst üste yerine).
Prosedür TEKRAR'a ulaştığında bilgisayar bir sayı beklediğini bilir ve bu sayı: SAYI adlı etikette bulunabilir. O yüzden de bu sayıyı kullanır ve böylece, prosedürün bundan sonraki aşamasına ulaştığında :KENAR adlı etikette depolanmış ÖNE değerine yükselir ve sonunda da: AÇI'dan değerini alır. Değişkenler herhangi bir prosedürde kullanılabilir, tail recursion kullananlar dahil. Değişkenlerden tail recursion'dan yararlanan bir daire üretiminin şöyle bir prosedüre ihtiyacı olurdu.

```
TO CIRCLE1 :SIDE :ANGLE
FORWARD :SIDE
RIGHT :ANGLE
CIRCLE1 :SIDE :ANGLE
END
```

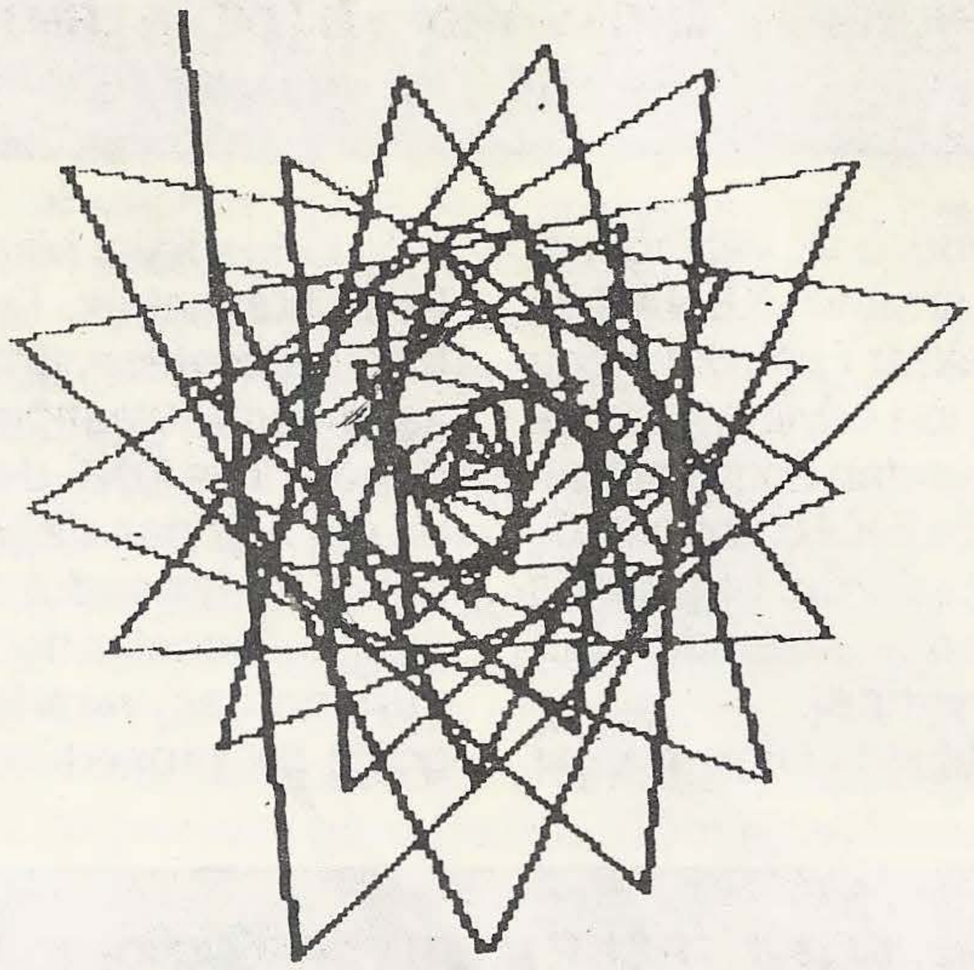
Bu prosedürdeki esas farklılık son satırdadır - DAİRE 1 :KENAR :AÇI. DAİRE 1 yazmak yeterli olmaz, çünkü bu prosedür, değişkenleri için ancak bazı sayılar (değerler) verdiğiniz takdirde çalışabilir. Bu nedenle, en son satırdaki başlığa dahil edilen etiket isimlerini kullanmanız gerekli. Bu durumda prosedür aynı sayılarla başlayacaktır. Aynı sayılar her işleme süresinde kullanılmak zorunda değildir. İzleyen HELEZON (SPRİAL) programı bu noktayı canlandırmaktadır.

```
TO SPIRAL :SIDE :ANGLE :INCREMENT
FORWARD :SIDE
RIGHT :ANGLE
SPIRAL :SIDE + :INCREMENT :ANGLE :INCRE
  MENT
END
```

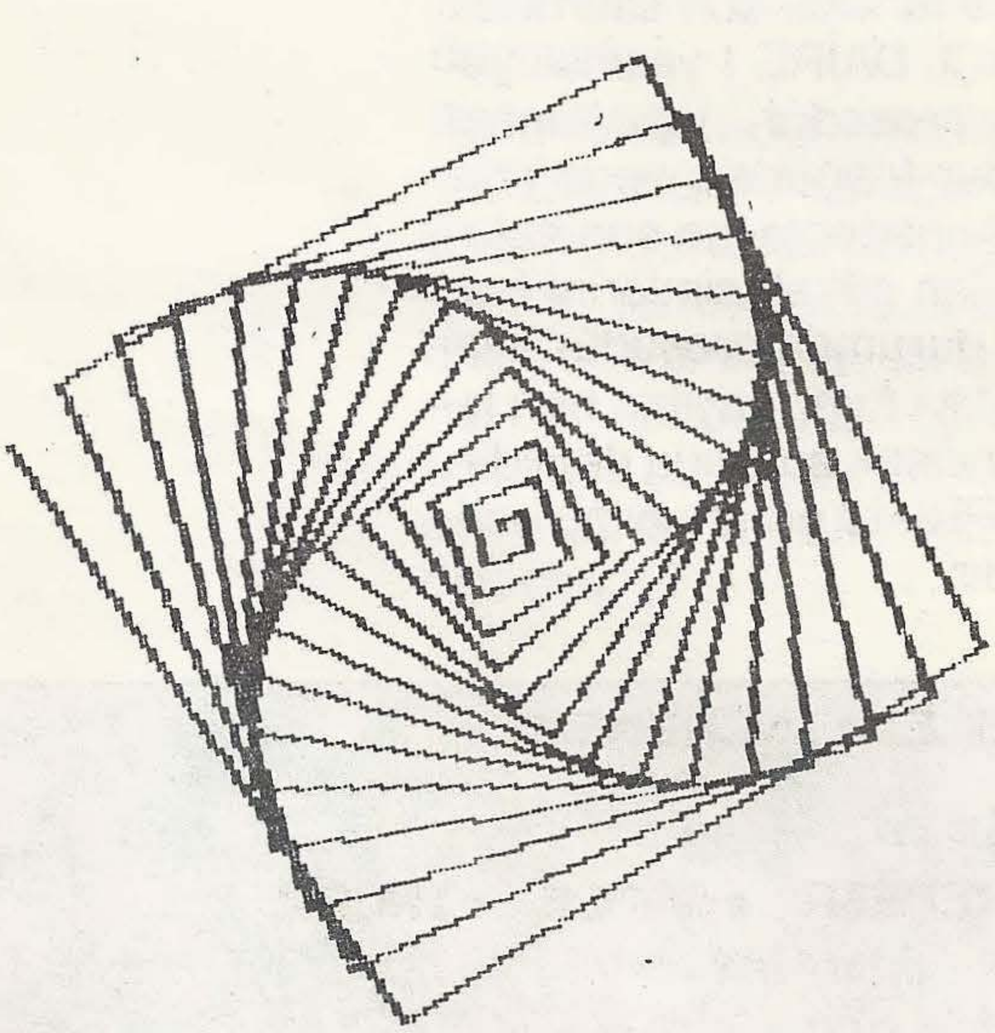
Bu prosedür hayli çok yönlüdür, bir dizi çizim üretecektir, ki bunların bir örneği arka sayfada gösterilmiştir.

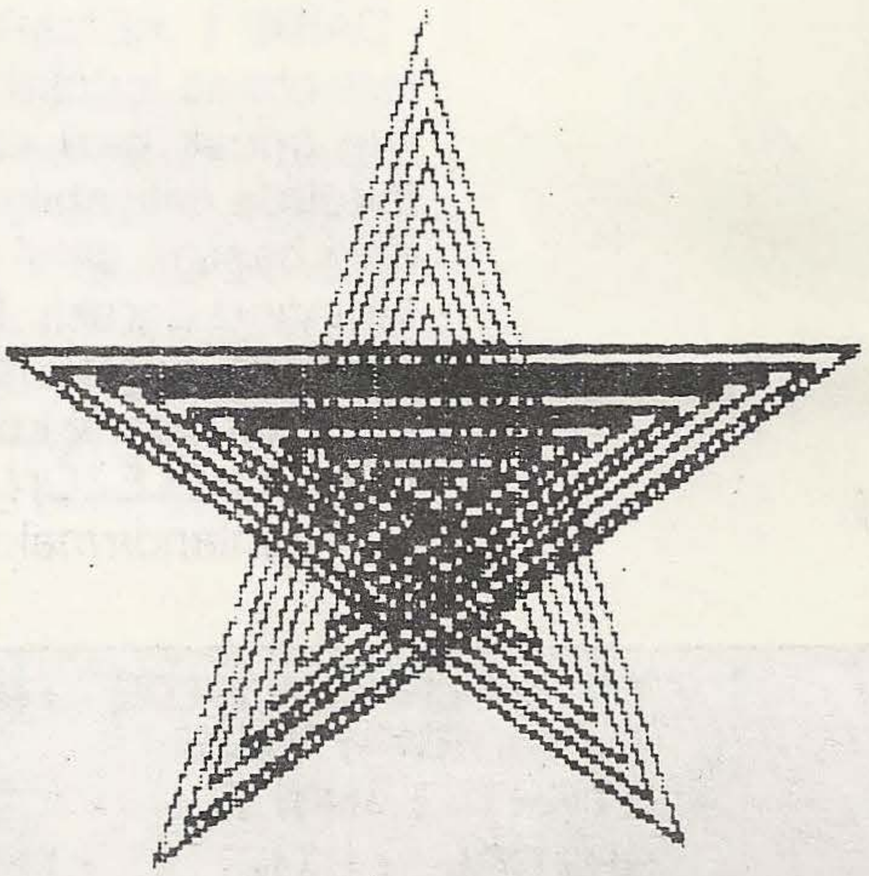
Çalıştırmak amacıyla, prosedürdeki ilk işleme süresi için kenarın uzunluğunu belirleyiniz.

Turtle bunu çizer ve sonra da sizin seçtiğiniz derece sayısı kadar döner. Şimdi, aynı işlemleri yeniden yazmasını söyleyen bir satıra ula-



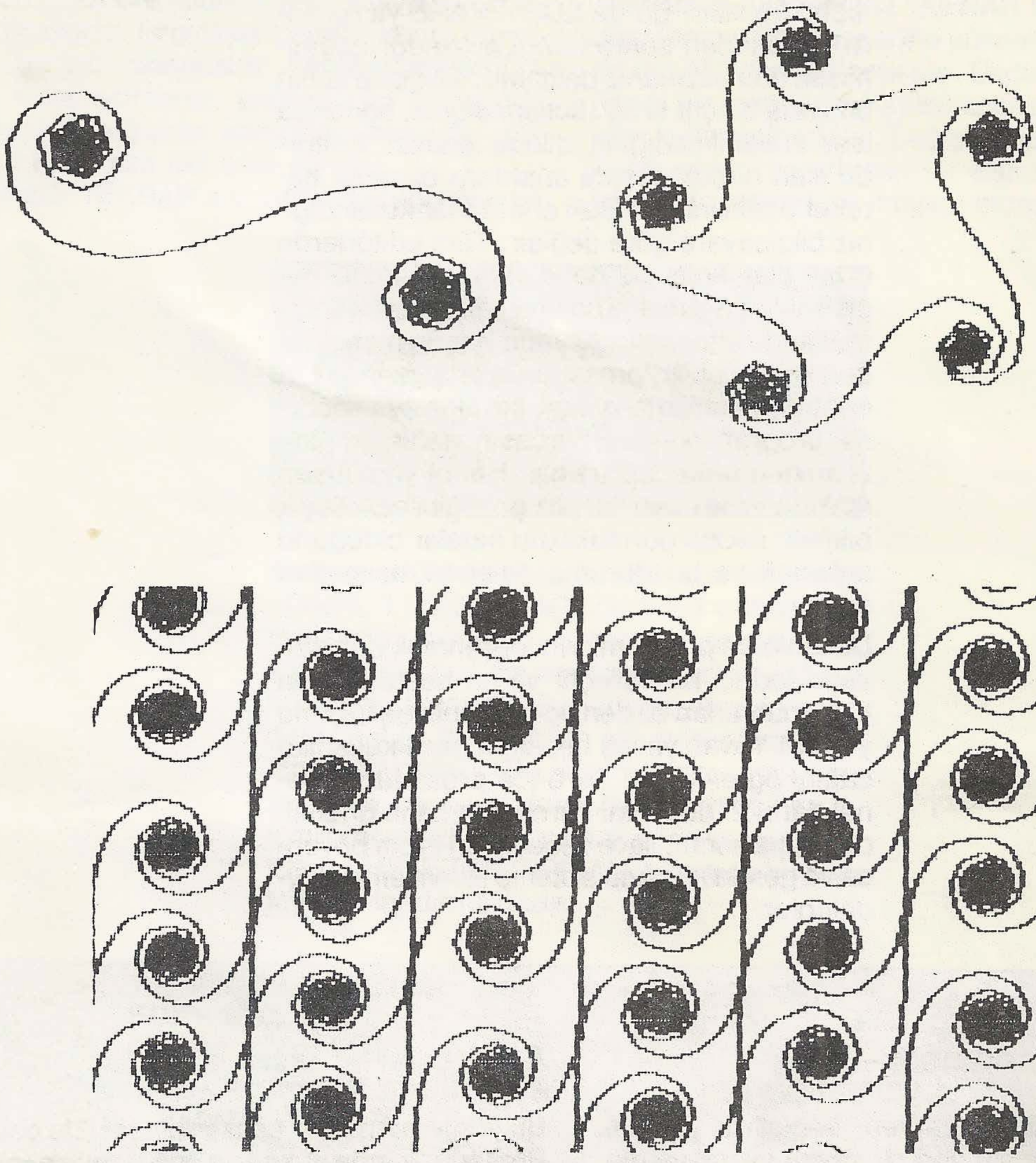
şır, ama bu sefer sayı :KENAR'ın eski değeri ile ve buna ek olarak da :ARTMA (INC.) değeri ile çalışmasını söyleyecektir. Bazı örneklere bakarsak bu işlem daha da açık seçik hale gelir.





Bu işlem sonucunda yukarıda gördüğünüz beş kenarlı yıldız şekli oluşur. Çizim HOME durumundan başlar ve Turtle'ın ilk yaptığı şey, yukarı doğru giden 5 birimlik bir hattır. Turtle daha sonra 144 derece sağa döner. Bu noktada artık HELEZON'a (SPIRAL) (:KENAR + :ARTIŞ) :AÇI :ARTIŞ satırına ulaşmıştır. Bilgisayar bu satırın :KENAR (5) için kullanılan değerin :ARTIŞ (3)'ın değerine eklenmesi ve bunun da yeniden :KENAR etiketine depolanması anlamına geldiğini bilir. :KENAR ile :ARTIŞ aynı kalır. Şimdi HELEZON'dan ikinci geçişte :KENAR'ın değeri 8'dir, böylece Turtle öne doğru 8 birim ilerler ve dereceden döner. Üçüncü ilmek (spiral) 11 birim uzunluğunda bir hat oluşturur, dördüncü ise 14 birimlik ve bu böylece sürer. Spiral (helezon) siz CONTROL anahtarını kullanarak ona durmasını söyleyene kadar büyümeye (artmaya) devam edecektir. Bu, tail recursion'un KALIP 1'inden da-

ha güçlü bir şekildir, çünkü her spiral (helezon) dizaynı yapışında :KENAR için farklı değer kullanır. Kullandığınız açıya kenar yerine artmış bir değer ilave edebileceğinize de dikkat buyurunuz. Sonuçta elde edebileceğiniz bazı tür kalıplar aşağıda gösterilmiştir.



Öbür önemli ve dikkat edilmesi gereken nokta parantez () kullanımıdır. Bilgisayarların bunlaın ne yaptığını bilmesi mecburi değildir. Bunlar size ne olup bittiğini rahatlıkla takip etmenizi sağlar, bilhassa daha önce kaydettiğiniz programları yeniden yazma aşamasına geldiğiniz zaman.

## Prosedürleri Düzenleme (Editing)

Programlarınız her zaman ilk seferde çalışırsa gerçekten uzman bir programcı olacaksınız. Böyle olmakla birlikte, çok basit değişiklikler yapmak için bütün prosedürü yeni baştan yazmanız çok sıkıntı verici bir iş olduğunu söylemeye gerek bile yok.LOGO bu işi, EDİTOR'a getirerek prosedürü değiştirmeyi mümkün kılar. Bu da EDİT ''KARE ya da soru işaretinden sonra hangi prosedür çağırılmışsa onu yazmanız demektir. Yalnızca ismin başında bir çift tırnak kullanıldığına, sonunda ise, kullanılmadığına dikkat ediniz. Editörde iken birçok sayıda anahtara basarak hareket edebilirsiniz: Belli anahtarlar kullandığınız bilgisayara göre değişir. Tüm editörlerde ortak olan kolaylık, daha yeni yazdığınız değişiklikleri işlemeksizin sizin dışarı çıkmanıza müsaade etmesidir. Aslında bunun hiçbir değeri kalmayabilir, prosedürün orta yerinde birdenbire kafanızda parlak bir fikir uyanıverirse, programın yanlış parçasını değiştirdiğinizi aniden farkedebilirsiniz. Hangi komutların makinenizde neye karşılık geldiğini açık seçik bilmek, neden durmaksızın hatalar olduğunu anlamak ve bu durumu önlemek açısından önemlidir.

Düzenleme programlarını öğrenmek daha ziyade tedrici bir süreçtir ve en başta hemen tüm anahtarları birden öğrenmenin de lüzumu yoktur. Yavaş yavaş hepsinin nasıl kullanılacağını öğrenirsiniz. İlk önce prosedürün genel hareket düzenini öğrenmek daha önemlidir. Daha sonra, ilave denetimleri işlem boyunca ve gerektikçe ana sisteme eklemlemek faydalı olur.

### Apple Uyarlaması

Bir prosedürü yazarken, değişiklik yapmak amacı ile gerisin geri bir şeyler yazmaya mezun değilsiniz ve bu iş için mutlaka editörü kullanmanız gerekir. Böylece, önceki satırı yukarı kaldırmak için CTRL-P ve yandaki satıra geçmek için de CTRL-N'yi kullanmalısınız. Yukarı ve aşağı ok ilmeç denetimleri bu editör ile çalışmaz. Hiçbir değişiklik yapmadan editör'den çıkmak için CTRL-G'yi kullanınız, ki bu, devam etmekte olan herhangi bir programı durdurmak amacı ile de kullanılır. Prosedürlerde değişikliklerle birlikte karşılıklı moda geri dönmek için CTRL-C'yi kullanınız.

### Atari Uyarlaması

Bu yöntem Apple'a benzer bir şekilde çalışır. Prosedür içinde yukarı ve aşağı veya sağa ve sola hareket etmek için imleç denetim anahtarlarını kullanabilirsiniz.

Karşılıklı moda dönmek istediğinizde ESCAPE anahtarına basınız ve böylece prosedüre yazdığınız değişiklikler ile ilgili işlem yerine getirilmiş olur.

Herhangi bir değişiklik yapmak istemiyorsanız, o zaman BREAK anahtarı editörü bitirmeden durduracak ve prosedürü dokunulmamış bırakacaktır.

## Spectrum Uyarlaması

Yukarıda olduğu gibi EDİT''KARE yazarak düzenleme modunu girersiniz. İmleç kontrülleri CAPS SHIFT ile 5, 6, 7 ve 8 anahtarlarını kullanan Spectrum'daki olağan yöntemle çalışırlar ve prosedür ile size manevra yapma imkanı verirler. Öbür değişiklikler ve editörden değişikliklerle çıkabilmek için mutlaka E modunda olmalısınız. Bunu da CAPS SHIFT ile SYMBOL SHIFT'e aynı anda basarak elde edebilirsiniz. Ekranın dibinde sağ köşede E işareti böylece belirecektir. Bir kere E moduna girince, modifikasyonlarla birlikte çıkmayı isterseniz, büyük harf C'ye basınız. Değişiklikler olmadan editörden çıkmak isterseniz, o zaman CAPS ve BREAK anahtarlarına birlikte basmalıdır.

## Acornsoft Uyarlaması

Karşılıklı moddaki satırlara benzeyen düzenleme satırları, bilgisayardaki BASIC satırlarının düzenlenmesinde kullanılan komutlarla aynı kelimeleri kullanır. LOGO prosedürlerini düzenlemek amacıyla EDIT''KARE yazarak edit moduna giriniz veya prosedürünüzün özel adını yazınız. Klavyenin sağındaki imleç anahtarları, imleçi sağa-sola hareket ettirmek için kullanılabilir.

Değişiklik yapmayı tamamladığınızda, COPY anahtarına basılarak editörden çıkacak ve yazdığınız değişiklikleri yerine getirecektir.

Şayet, yüklediğiniz editörden değişik yapmadan çıkmak isterseniz; o zaman ESCAE anahtarına basınız.

## Şekiller Kitaplığı

Resimler veya kalıplar oluşturacağımız zaman birlikte kullanabileceğimiz basit şekilleri çizmek için bir dizi prosedürler yapmak faydalı olabilir. Değişkenleri öğrendiğimizde bunları prosedürlerimize eklemlemek makuldur. Böylece mümkün olabildiğince ufak bir hal de alırlar. Şekil prosedürlerinde değişkenler kullanma, her türden ve her boyutta şekiller çizmeyi çok kolaylaştırır. Aşağıdaki şekiller, genel olarak şekil çizimini incelemek için iyi bir temel oluşturur.

TRIANGLE
RT. TRIANGLE
SQUARE
RECTANGLE
DIAMOND
PENTAGON
CIRCLE

**Üçgen**

```
TO TRIANGLE :SIDE
REPEAT 3 [FORWARD :SIDE RIGHT 120]
END
```

**Kare**

```
TO SQUARE :SIDE
REPEAT 4 [FORWARD :SIDE RIGHT 90]
END
```

### Dik Üçgen

SIDEA ve SIDEB'nin değerleri dik üçgenin kısa kenarlarını belirler. Hipotenüs ise belirlenen noktaya dönerken bir doğru çizen SETPOS'tan yararlanarak çizilir. Bu işlemin sonunda kaplumbağa ikinci çizgiye paralel kalacaktır.

```
TO RT.TRIANGLE :SIDEA :SIDEB
MAKE "SAVEPOS POS
FORWARD :SIDEA
RIGHT 90
FORWARD :SIDEB
SETPOS :SAVEPOS
END
```

### Dikdörtgen

```
TO RECTANGLE :SIDEA :SIDEB
REPEAT 2 [FORWARD :SIDEA RIGHT 90
  FORWARD :SIDEB RIGHT 90]
END
```

### Baklava

```
TO DIAMOND :SIDE :ANGLE
REPEAT 2 [FORWARD :SIDE RIGHT :ANGLE
  FORWARD :SIDE RIGHT 180 - :ANGLE]
END
```

### Beşgen

```
TO PENTAGON :SIDE
REPEAT 5 [FORWARD :SIDE RIGHT 72]
END
```

### Daire

```
TO CIRCLE :SIDE
REPEAT 36 [FORWARD :SIDE RIGHT 10]
END
```

# Zeki Bilgisayar Yardımıyla Eğitim

## 1. Giriş

*Bugünlerde, 'zeki' sıfatını taşımayan eski bilgisayarlar yerine, 'zeki' bilgisayar yardımıyla eğitim/öğrenme* ***(Intelligent Computer Assisted Instruction/Learnieg)*** *sistemlerinden söz etmek tercih ediliyor. Bu durumun bir nedeni geleneksel hesaplama yöntemlerine uymayan her sistemi 'zeki' diye tanımlama modası ise, bir nedeni de eğitsel sistem geliştiricilerinin ciddi bir şekilde yapay zekâ* ***(AI)*** *teknik ve yöntemlerine yönelmiş olmaları. Göreceğimiz gibi, bu yönelim sistem geliştiricilerinin eğlenceli yeni alet ve yöntemlerle uğraşma heveslerinden değil, CAI (bilgisayar yardımıyla eğitim)'e geleneksel yaklaşımların eğitsel sistemlere ilişkin iddealleri gerçekleştirememesinden kaynaklandı. Geleneksel alet ve yöntemler bu iş için yetersizdi.*

*CAI tarihinde görülen akımları inceledikten sonra, Yapay Zeka'nın (YZ) CAI açısından taşıdığı önem üzerinde duracağız. Bu makalede, YZ'nin eğitsel sistemlere uygulanmasından bu yana ortaya çıkmış olan başlıca yaklaşımlara örnek olarak üç sistem ele alınmakta. Bu prototipik sistemlerin YZ'nin gelecekteki eğitsel sistemlerin temel taşı olduğuna dair inancımızı güçlendirdiğini söyleyebiliriz; ancak, bunların en uygun bileşenlerinin neler olacağı, ya da nasıl bir felsefeye dayanmaları gerektiği üzerinde fikir birliği sağlanmış değil.*

## 2. Kısa Tarihçe

*Eğitimde bilgisayar kullanımının tam bir tarihçesini vermek niyetinde değiliz; bu konu kısa süre önce O'Shea ve Self tarafından ayrıntılı bir şekilde ele alınmıştı (1). Nitekim, başlıca akımların bu kısa tarihçesini hazırlamakta onların yazısından bir hayli yararlanmış bulunuyoruz.*

*O 'Shea ve Self eğitimde bilgisayar kullanımı konusunda onbir yaklaşım tespit etmişler:*

*— Doğrusal* ***(Linear)*** *programlar;*
*— Dallanan programlar;*
*— Üretici bilgisayar yardımıyla öğrenme;*
*— Matematiksel öğrenme modelleri;*
*— TICCIT;*
*— PLATO;*
*— Benzeşim;*
*— Oyunlar;*
*— Problem çözme;*
*— Kurtarıcı tarzlar;*
*— Diyalog sistemleri.*

*Onların da işaret ettiği gibi, bu liste gelişimlerin tam olarak kronoloji sırasını vermemekle birlikte, eğitimde bilgisayar kullanımının ele alınışındaki akımları yansıtıyor. Bu yaklaşımların her biri üzerinde biraz duracağız. Ve, bugün çoğu moda olmadığı için yetersizliklerinden de söz edeceğiz.*

### Doğrusal Programlar

*Doğrusal Programlar operant pekiştirme ilkesine dayanmaktaydı. Operant davranış, pekiş-*

tirme yoluyla koşullanan bir davranış birimidir (Skinner 2). Bu yaklaşımda istenmeyen davranışlar tamamen görmemezlikten gelinir; dolayısıyla, eğitime uygulandığında, doğrusal programlarda kullanılan eğitim malzemelerinin öğrencinin istenen davranışı gösterme şansını artıracak biçimkde dikkatle seçilip düzenlenmesi gereklidir.

Operant koşullanmanın başlıca savunucusu olan davranışçı psikolog B.F.Skinner'a göre, pekiştirme mekanizması olarak insan öğretmenin yerine pekâlâ bir makine de geçebilirdi. Bu görüş, programlanmış öğretme makinasını doğurdu. Öğrencinin elle çalıştırdığı bu makineler eğitim malzemesini çerçeveler halinde sunuyor, öğrencinin her çerçeveye bir cevap vermesi gerekiyordu (ve daha önce de belirttiğimiz gibi, çerçeveler doğru cevap ihtimalini yükseltecek biçimde düzenlenmişti). Makinenin üzerindeki pencereler sayesinde öğrenci cevabını sorunun yanına kaydediyor, sonra da makinenin içindeki önceden hazırlanmış kağıdı cevabının üzerine getiriyor ve cevabı ister yanlış ister doğru olsun, bir sonraki soruya geçiyordu.

Yetersizliği uzun zamandan beri kabul edilen bu yaklaşımın belki de en önemli kusuru bireyselleştirme ve zengin bir geri bildirim sağlayamaması: bütün öğrenciler aynı malzemeyi aynı düzende alıyorlar, ve geri bildirim cevapları dikkate almadığı için pek az anlam taşıyor.

## Dallanma Programlar

Dallanan program yaklaşımı bireyselleştirme ve geri bildirim sağlama gereğinin anlaşılmasının ürünü olarak ortaya çıkmıştır. Bu programlar, öğrencinin cevabı doğruysa bir sonraki soruya geçmesine izin vermek, cevap yanlışsa da bunu belirtip bir şans daha tanımak yoluyla, öğrenciye tepki gösterir.

Bu programların gelişimine katkıda bulunmak amacıyla yazar dilleri oluşturulmuştur. Program yazarının öğrencilerin cevaplarını tek tek ele alabilmesi için, bu diller büyük ölçüde IF ve GOTO emirlerine dayanır. Yazar dillerinin kolaylığı öğretmenleri kendi ders malzemelerini üretmeye teşvik etmiş ve bu yolla da bilgisayarların sınıfa girmesi konusundaki tereddütlerin aşılmasına bir ölçüde katkıda bulunmuştur. Ne var ki, bu tür programlar geliştirmek isteyen herkes ortak bir sorunla karşılaşmaktadır: ele alınacak durumların sayısı arttıkça, gerekli dalların sayısı da artar. Öyle ki, sonunda profesyonel bir programcının bile programlamakta güçlük çekeceği gerçek bir dal patlaması ortaya çıkabilir. Örneğin, programın cevaplarla, tek tek ilgilenebilmesi için, her olası cevap için koşullu birer dalı olması gereklidir. Böylece, her dal ideal olarak cevapları gene tek tek hesaba katacak yeni bir durum anlamına gelir. Bu süreç yalnızca ikişer olası cevabı bulunan dört soruyu kapsadığında, ele alınması gereken durum sayısı 2 X 2 X 2 X 2, yani 16 olmaktadır.

Dolayısıyla, bu yaklaşım teknik sorunlar içeriyor. Belki daha da önemlisi, eğitimin öğrenciyi de 'işin içine katması' gerektiği giderek

daha iyi anlaşılmakta. Oysa öğrencilerin kendilerine ne anlatılırsa onu öğrenecekleri inancına dayalı dallanan programların bu ilkeyi gerçekleştirebilmesi pek kolay sayılmaz.

## Üretici Bilgisayar Yardımıyla Öğrenme

Üretici sistemlerin en önemli özelliği, öğrencilere uygun zorlukta sorular üretip çözümlerini getirebilmesidir. Böylece (problem seçimi açısından) bireyselleştirilmiş eğitim mümkün olmaktadır.

Dallanan program yaklaşımıyla yeterli sayıda egzersiz oluşturmanın çok güç olduğu aritmetik tekrar çalışmalarında büyük kolaylık sağlayan üretici sistem, program değişkenlerini mantıklı bir şekilde kullanarak öğrencinin istediği kadar problem üretebilir. Bunun da ötesinde, konu hakkındaki bilgisine dayanarak problemlerin zorluk düzeyini denetleyebilir.

Fakat bu sistem tekrar tipi egzersizlerle sınırlıdır ve bir programın uygun problemler saptayabilmesi için problemlerin ilgili olduğu konu alanının tam bir biçimde temsil edilmesi gerektiğinden, yalnızca çok iyi yapılandırılmış konularda başarı gösterir.

## Matematiksel Öğrenme Modelleri

İsminden de anlaşılacağı gibi, bu yaklaşım öğrenme teorisini metametiksel işaretler sistemiyle (notasyonuyla) ifade etmeyi konu alır. Çeşitli öğretme stratejilerinin sonuçlarını tahmin edebilecek teoriler geliştirip, bunlara dayanan bilgisayar programlarıyla belirli bir durumda en uygun öğretme eylemini seçebilmek amacıyla oluşturulmuştur.

Buradaki başlıca sorun modeli geliştirmenin güçlüğü. Öğrenme süreci hakkında zaten pek az şey biliniyor ve -bu yöntemde olduğu gibi- öğretilecek konunun ne olduğuna bakmaksızın yeterli bir model geliştirilebileceği kuşkulu.

## Ticcit

Aşağıda anlatacağımız PLATO gibi TICCIT **(Time-SHared Interactive Computer Controlled Information Television**/Zaman Paylaşımlı Etkileşimli Bilgisayar Denetimli Bilgi Televizyonu) da 1971'de Amerikan Ulusal Bilim Vakfı (NSF) öncülüğünde kurulmuştu. NSF'nin 5 yıllık bir süreyi kapsayan 10 milyon dolarlık yatırımı, bilgisayar yardımıyla eğitimin geleneksel yöntemlerden daha ucuz ve etkili olup olmayacağını araştırmaya yönelikti. Yani, TICCIT mevcut öğretim uygulamalarına yardımcı bir sistem değil, temel öğretim vasıtası olarak tasarlanmaktaydı.

Malzemeler bir eğitim psikoloğu, (sistemin değerlendirmeye yönelik olarak sunduğu iki konu, yani kalkülüs matematik ve İngilizce kompozisyon için birer tane) alan uzmanı, bir eğitsel **(instructional)** tasarım teknisyeni, bir değerlendirme teknisyeni ve bir paketleme uzmanından oluşan bir ekip tarafından hazırlanmıştı ve üç aşamalı bir metin halinde sunulmaktaydı: genel ilkelerin bir özeti, bu ilkelere dair örnekler, ve son olarak da öğrencinin uygulayacağı egzersizler.

Her TICCIT sistemi 128 terminale birden hizmet verebilen iki mikrobilgisayarın kontroiündeydi. Terminaller bir klavye, (grafikler için) bir renkli televizyon, renkli videobantlar, önceden kaydedilmiş mesajları iletecek bir hoparlör ve öğrencinin mesajlarını alacak bir klavyeyle ışık kaleminden oluşuyordu. Klavyede HELP, EXAMPLE, HARD, SKIP, BACK gibi düğmeler vardı. Bu sayede, öğrenci malzemenin akışını denetleyebiliyordu.

Sistemin uygulanışı, özellikle ekonomik yönün aşırı vurgulanmasından ve ilgili taraflar arasında her iki konunun da içerikleri ve öğretim metodolojileri konusunda görüş ayrılığı doğmasından kaynaklanan çelişik tepkilerle karşılandı. NSF'nin bağımsız değerlendirmesinin pek de beklenmedik sonuçlarına göre, TICCIT kurslarını tamamlayan öğrenciler TICCIT'le çalışmamış olanlara nazaran daha başarılı görünüyordu. Ne var ki, bu sonuçlar TICCIT'in iyi öğrencilere eğilim gösterdiği ve terk oranlarının geleneksel eğitime nazaran daha yüksek olduğu iddialarıyla karşılandı. Başarılı sayılabilecek sonuçlarına karşın, TICCIT yaygın bir kullanıma kavuşmadı.

## Plato

1960'da doğan ve NSF desteğine çok sonraları kavuşabilen PLATO **(Programmed Logic for Automatic Teaching Operation**/Otomatik Öğretim Operasyonu için Programlanmış Mantık)ın TICCIT'den en önemli farkları:

**(sürecek)**

# Sayılarla Oynamak

Pek çok insan bilgisayarlarla sayılar arasındaki bağın çok açık ve kesin olduğunu düşünür. Logo içinde kaplumbağanın kullanımını daha önce tanıtmıştık. Kamlumbağayla oluşturulan biçimlerin çoğu, dönülen açı ile 360 derece arasındaki ilişkiye dayanır. Dolayısıyla çizdiğimiz geometrik şekilleri sayıların bir ifadesi olarak görüp sayılar arasındaki ilişkileri incelemek mümkündür. Bir başka seçenek ise, çok güzel şekiller üretmek ve sayılar üzerine hiçbir şey öğrenmeden bilgisayar programları yazmanın mantığını öğrenmektir. Gerçekten de Logo'nun pek çok çocuğa ve öğretmene en büyük yardımı, yaptıklarının farkında olmadan ''matematik yapma''larına imkan tanımasıdır!

Kamlumbağa grafik programlarında sayıların esas kullanım alanı çizilecek şekillerin kenar uzunlukları ya da kaplumbağanın dönme açısı için girdilerdir. Bu şekillerin bir kaçını incelersek, sayıların oluşturulan şekli belirlemedeki rolü çabucak görülebilir.

## Kaplumbağa grafiğinde sayıların kullanımı.

Üretilen şekli etkileyen etkenlerden birisi, dönülen derece sayısı ile 360° arasındaki ilişkidir. Bu ilişkiyi göstermek için iki örnek program verceğiz. Diğer matematiksel ilişkileri göstermek için bunlara benzer pek çoğu verilebilir.

PLOYGON ve SPIRAL örneklerimizin, ayrık programlar olarak ilginç şekiller ürettiklerini ve birbirleriyle tamamen ilişkisiz olduklarını, ama eğer aralarında bağ kurulursa çok daha genel ilkeler gösterebileceğini farkedeceksiniz.

```
TO POLYGON :SIDE :ANGLE
FORWARD :SIDE
RIGHT :ANGLE
POLYGON :SIDE :ANGLE
END

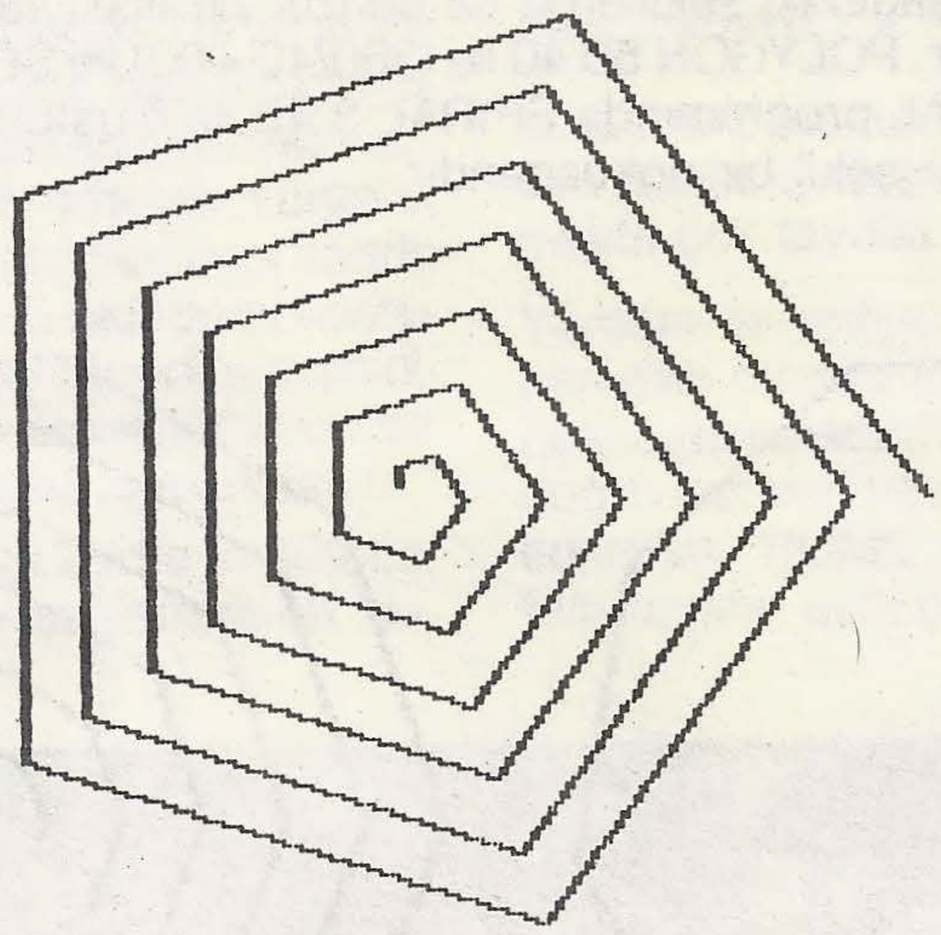
TO SPIRAL :SIDE :ANGLE :INCREMENT
FORWARD :SIDE
RIGHT :ANGLE
SPIRAL :SIDE + :INC :ANGLE :INC
END
```

Bu prosedürlerde iki ya da daha fazla sayı verilmiştir. :SIDE değişkeni, çizilen şeklin boyutunu, ya da SPIRAL programı için şekle ne kadar küçük başlanacağını belirler. :ANGLE için verilen sayı sonuçta çizilen şeklin nasıl bir biçim olacağını belirler. Bu ilişkiyi göstermek için POLYGON iyi bir örnektir. Daha önce gördüğümüz gibi eğer kaplumbağa bir şeklin dış tarafından dolaşıp başladığı yere dönüyorsa, 360 derece ya da 360 derecenin katları kadar döndürülmelidir. Bu nedenle, bir kare (ya

da dört kenarlı poligon) için yapılan dört dik dönüş 360/4 = 90 derece olarak hesaplanabilir. Üç kenarlı bir poligon (üçgen) 360/3 = 120 derecedir. Bu örnekleri çoğaltıp, kenar sayısı ile her köşe açısı arasındaki ilişkiyi gösteren bir tablonun kolayca yapılabileceği açıktır. POLYGON'un aşağıdaki çeşidi, şekli tamamlamak için kaç kenar çizileceğini bu ilişkiyi kullanarak hesaplar. Bir önceki prosedürden farklı olarak 360 derece ile açılar arasında tam bir ilişki varsa, sonsuz tekrarlı değildir. Eğer açılar tam katlara karşılık gelmezse (örneğin 83 derece alınırsa), program tekrarlıdır.

```
TO POLYGON :SIDE :ANGLE
REPEAT 360/ :ANGLE [FORWARD :SIDE
  RIGHT :ANGLE]
IF REMAINDER 360 :ANGLE = 0 [STOP]
POLYGON :SIDE :ANGLE
END
```

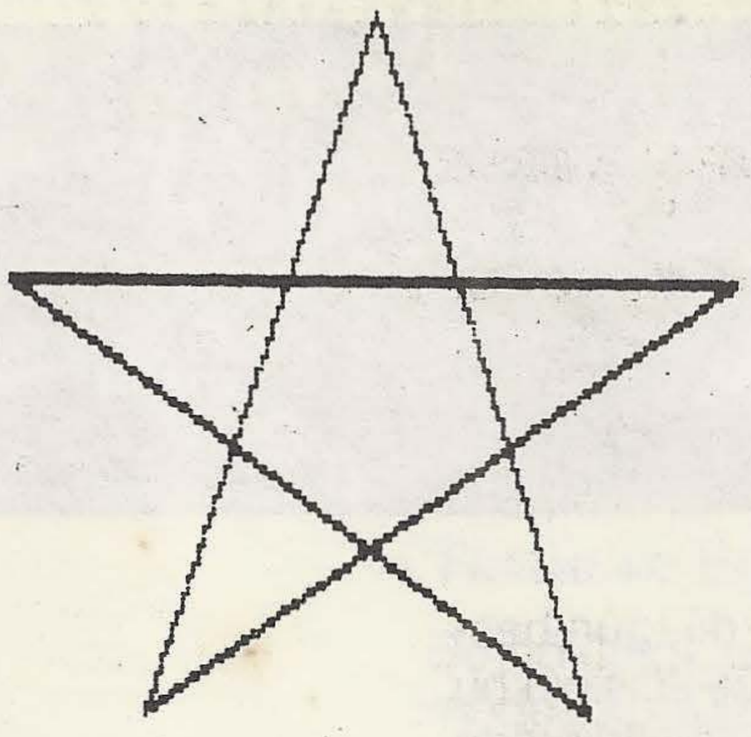
Eğer açıyı 72 derece alırsak, bir düzgün beşgen çizildiğini göreceksiniz. Dönüş açısının bu değeri SPIRAL programıyla kullanıldığında beş kenarlı bir spiral üretilir. Eğer SPIRAL ile yola çıkarsak, SPIRAL 3723, aşağıdaki şekli oluşturacaktır:

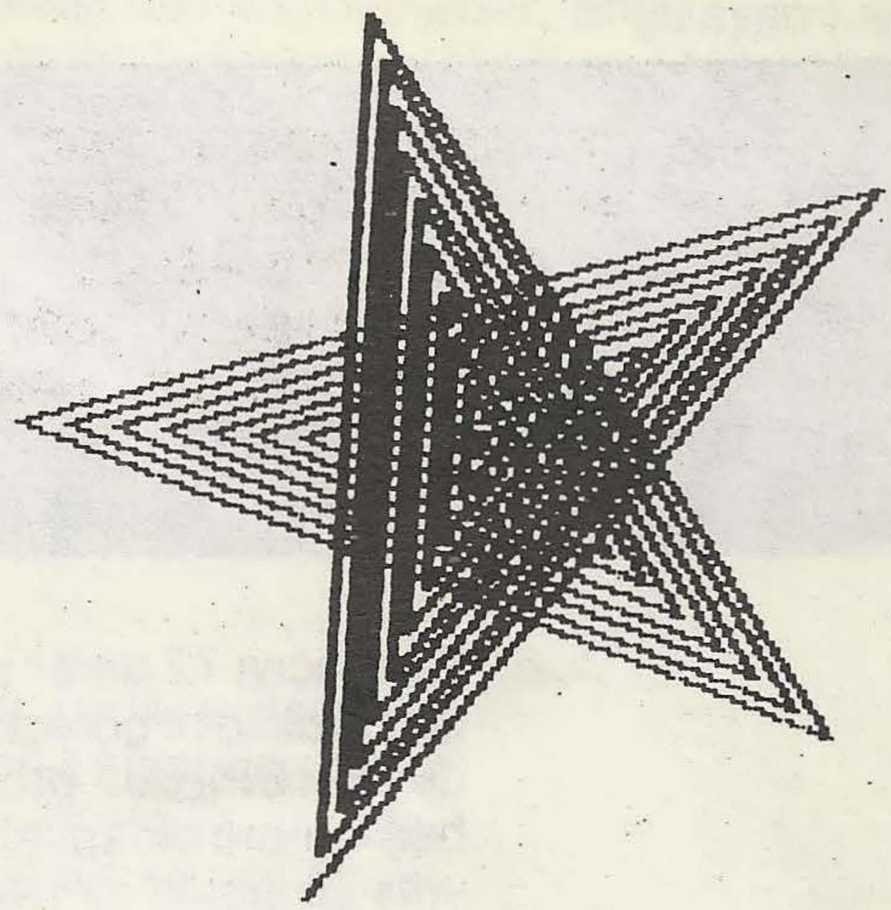


Daha önce belirttiğimiz gibi, :SIDE ve :INCREMENT için 3 değeri, ilk çizilen kenarın uzunluğunu ve prosedünün her tekrarında bunun ne kadar arttırılacağını gösterir. Ortaya çıkan şekil, 72 derece kullanıldığında üretilen spiralin aynı zamanda beş kenarlı bir şekil olduğunu gösterir.

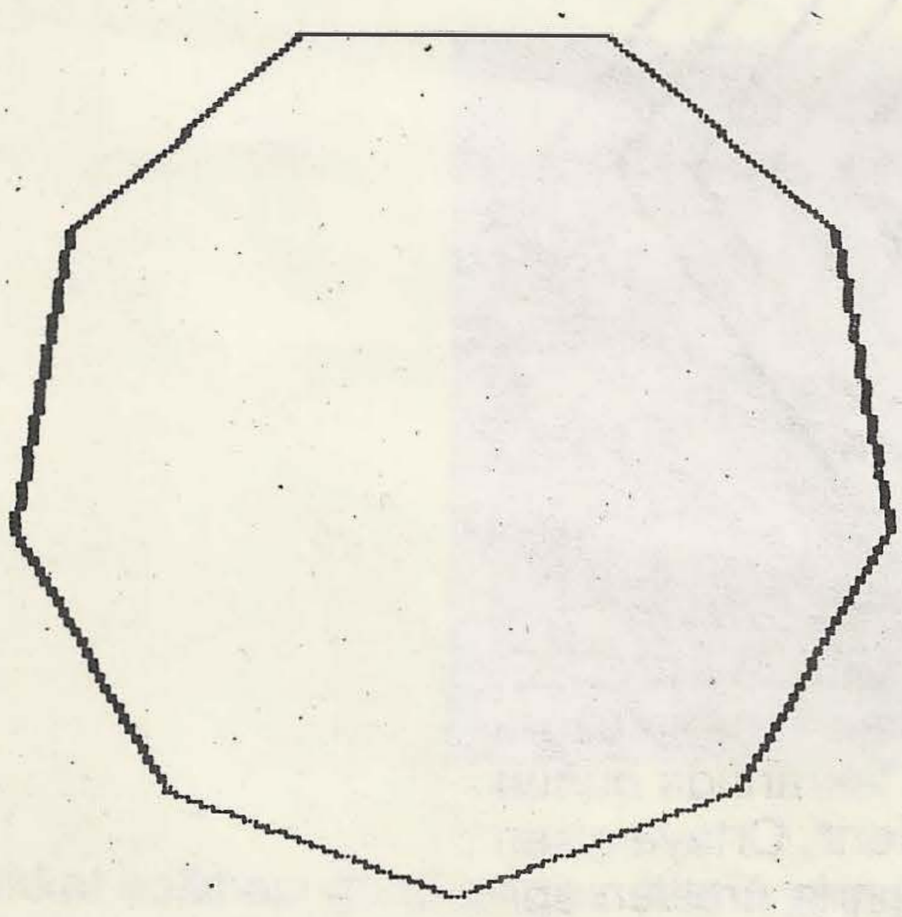
Açının derecesi katlarına, örneğin 144 dereceye çıkartıldığında daha ilginç şekiller üretilebilir. 144 sayısı, 360 dereceyi tam olarak bölmez, iki buçuk kere çıkartır. Eğer POLYGON 60 144'ü denerseniz, aşağıdaki şekillerden soldakini elde edersiniz:

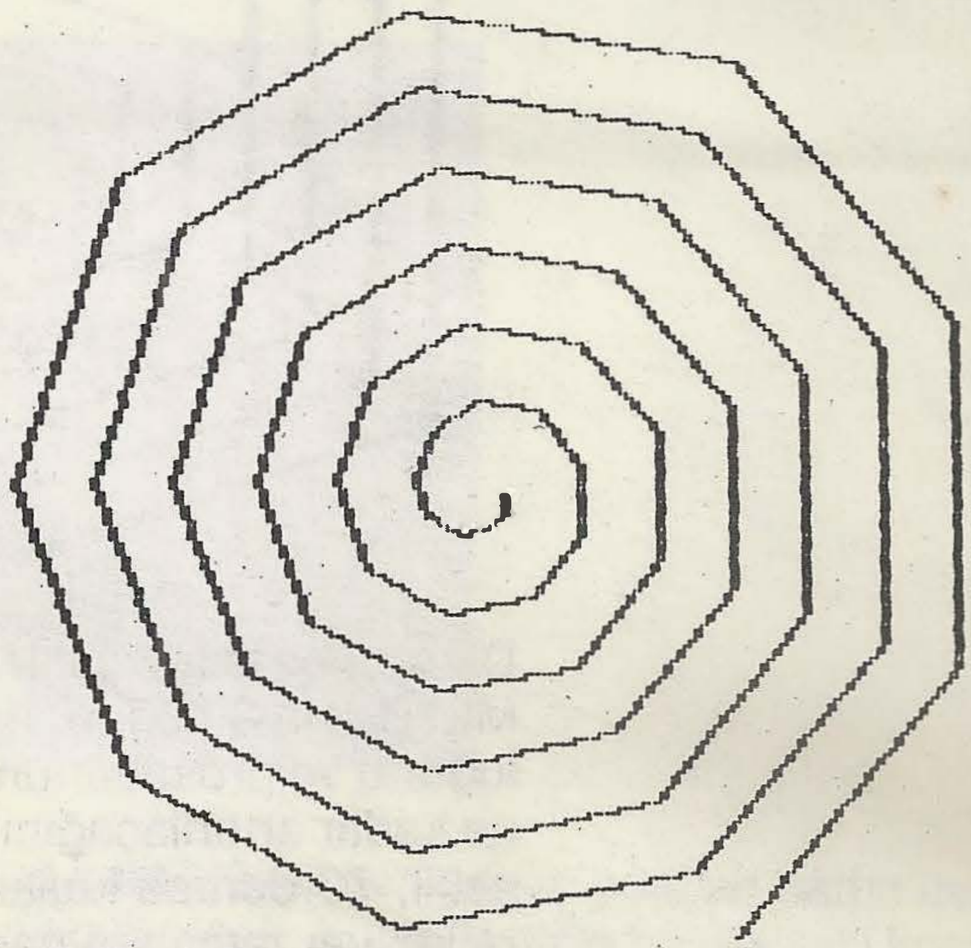
Çıkan şekil beş köşeli yıldızdır, çünkü 720 (360'ın iki katı), 144'e tam olarak bölünebilir. Bu şekli üretirken kaplumbağa 720 dereceye eşit olan iki tur tamamlanır. Bu durum SPIRAL'de de gözlenebilir; örneğin SPIRAL 3 144 2, sağdaki şekli oluşturur.



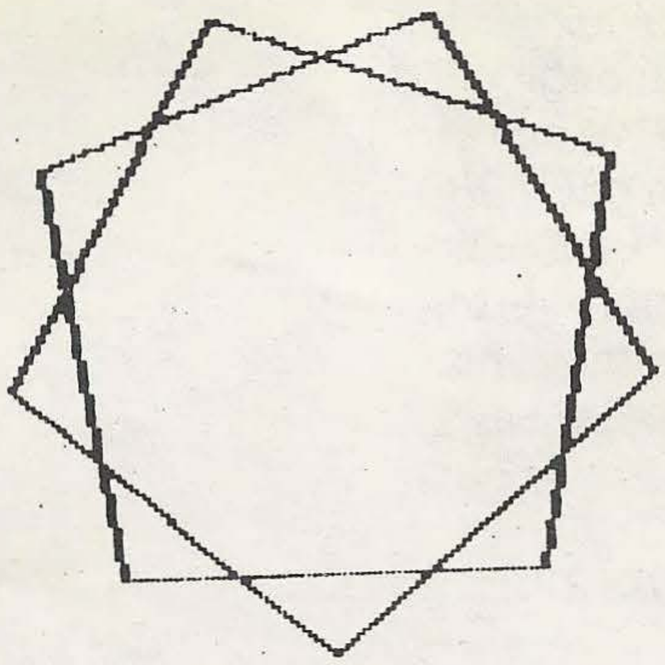


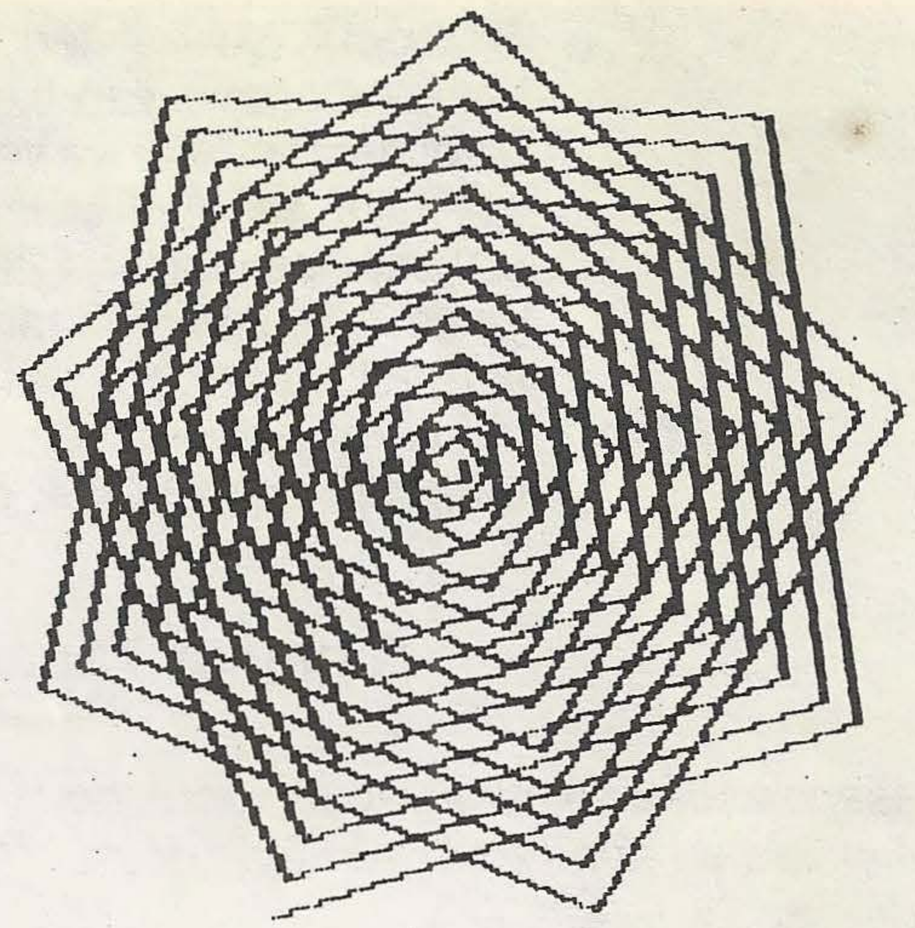
Aşağıdaki şekillerde de benzer bir ilişki vardır. POLYGON 50 40 ile (360/40 = 90) ve SPIRAL programında SPIRAL 3 40 ile oluşturulan şekil bir dokuzgendir.





Açıyı bir kez daha ikiye katlarsanız (böylece 80 derece olur ve 720'yi tam olarak böler), aşağıdaki şekiller oluşur: POLYGON 50 80 ve SPIRAL 3 80 1.

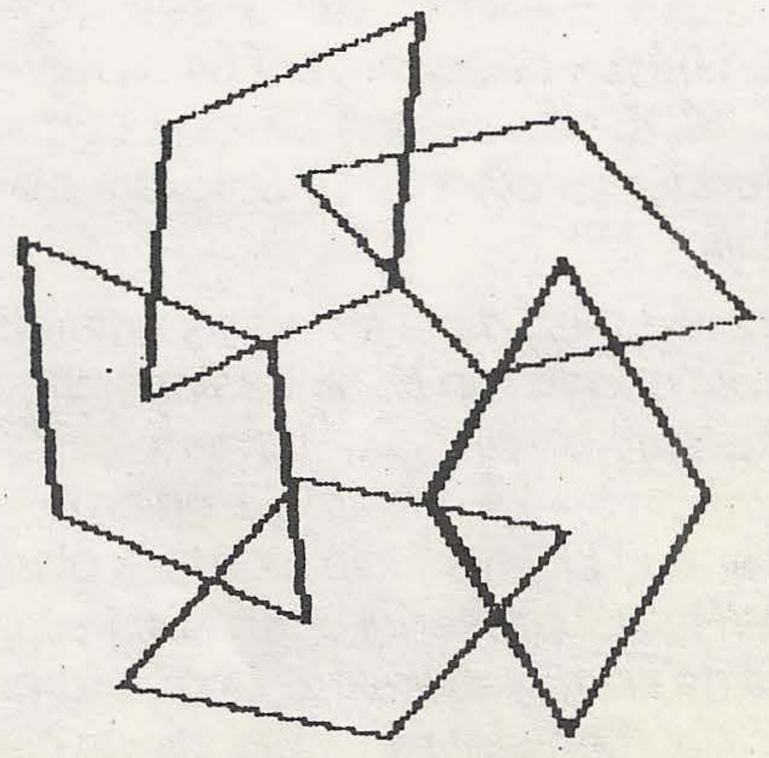


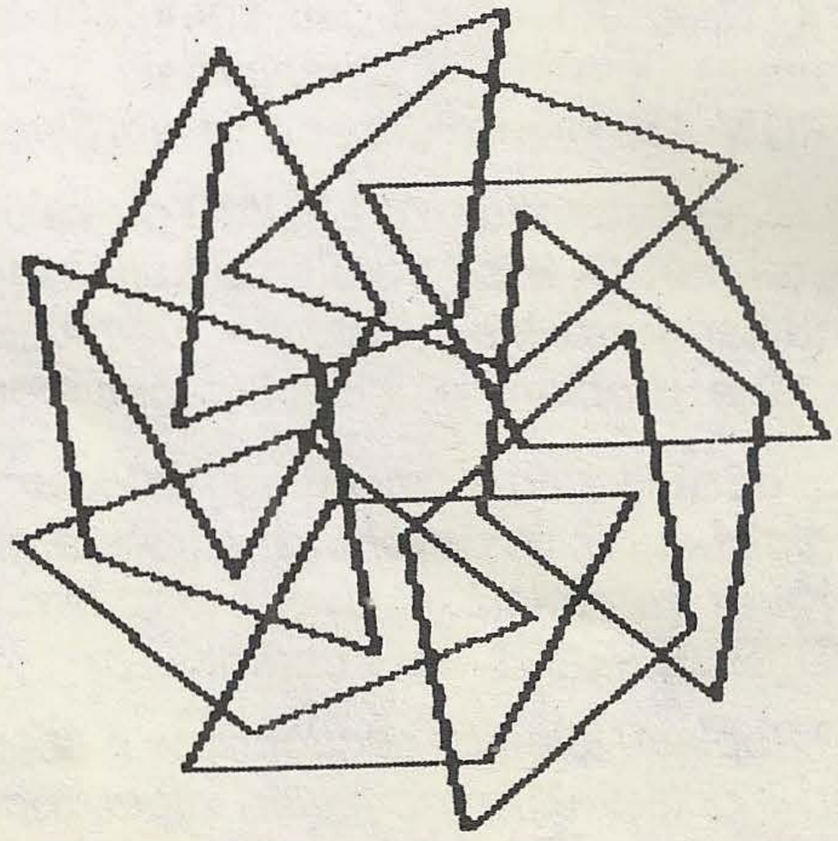


Aşağıdaki program, bir dizi içiçe geçmiş baklava şeklini (eşkenar dörtgeni) nasıl oluşturacağınızı gösterir. Son açıyı değiştirmek, oluşturulan dörtgenlerin sayısını farklılaştırır.

```
TO SWIRL :SIDE
LEFT 30
DIAMOND :SIDE
RIGHT 30
PENUP
FORWARD :SIDE
PENDOWN
RIGHT 72
SWIRL :SIDE
END
```

Şimdi gelen şekiller, son RIGHT açısı için sırayla 72 ve 40 derece verilen durumları gösterir.





Bir çocuk bu ilişkileri bir kez kavradığında, kamlumbağanın yarattığı şekiller artık bir sürpriz olmaktan çıkar ve verilen değere bağlı olarak ne oluşacağını tahmin etmek heyacanlı bir çalışma haline gelir. Çekici şekillerin çoğu 360 ya da 720 dereceye tamamlanmayan sayılarla üretilir. Eğer çocuk tamamlananların üretimini anlamakla başlarsa, tamamlanmayanlar üzerinde daha büyük istekle çalışacaktır.

## *Toplama Yapan Programlar!*

*Basit hesaplamaları kolayca yapan küçük hesap makinelerinin yeri henüz tümüyle doldurulamadı.*

*Ama bilgisayarlar, çocuk cevap verirken onun hesaplamalarını kontrol edebildiği için yararlıdır. LOGO'da, sayı hesaplarını kontrol eden programlar son derece kolay yazılabilir. Aşağıda vereceğimiz örnek, çocuğun yaş ve yeteneklerine bağlı olarak geliştirilip genişletilebilir.*

*Daha önce gördüğümüz gibi, sayıların matematik düzeni çocuklara oldukça tanıdık ve kolay gelecek biçimdedir: 4 + 5, 7 * 2 gibi. Kuşkusuz bölme (/), çarpma (*), programcılık (ve matematikte) kullanılan (küçük), (büyük) işaretlerine alışmak biraz zaman alabilir. Son iki işareti akılda tutmak için, sayılardan büyük olanının işaretin açık ucunda, küçüğünün de kapalı tarafta olduğunu belirtmek yararlı olur. Çocukların çoğu bunlara zaten alışkındır.*

*Bu tür programları yazarken, programın komutları tam olarak uyguladığına emin olmak önemlidir. Vereceğimiz örnekte, şu ana kadar tanıtmadığımız bazı LOGO özellikleri kullanılmıştır. Programın nasıl çalıştığını tam olarak açıklamak için ilerki sayfalardaki LIST ve WORD'le ilgili kısmı okumanız yararlı olacaktır.*

*Üst-düzey prosedür, ALLSUMS'dır ve üç sayı gerektirir. İlk ikisi, sorularda kullanılan sayıların hangi aralıkta yer aldığını, üçüncüsü de kaç tane problem oluşturulacağını belirler.*

*Prosedürün kalan kısmı ne tip bir toplama istediğinizi sorar ve cevabınızı ekranda gösterir. Karar verdiğinizde, uygun problem tipini (ADD, SUBTRACT, vb.) çağırır.*

*ADD (toplam) işlemini seçtiğinizde neler olduğunu incelersek, aynı ilkeleri diğer toplama tiplerine de uygulayabiliriz. İlkin ADD, tanımladığınız aralıkta kalan iki sayıyı belirtecek CHOOSE prosedürünü kullanır. Sonra bu iki sayıyı 3 + 15 = ? şeklinde yazar. Ardından, cevabın ne olduğunu hesaplar ve ''TOTAL (toplam) adlı bir değişkende saklar. Klavyeden cevabın verilmesini bekledikten sonra (FIRT READLIST), verilen değer (:ANSWER) ile :TOTAL'de saklanan değerlerin aynı olup olmadığını kontrol eden CHECKANSWER'ı (cevap kontrol) çağırır. Eğer aynıysalar, cevabı kutlayan bir yorum yazar; değilse WRONG (yanlış) prosedürünü çağırır. Bu durumda cevabın yanlış olduğu belirtilip çocuğun bir kez daha denemesi istenir. Bu komut, cevabı tekrar kontrol eder ve doğru cevap bulunana kadar yanlış olduğunu yazar. Doğru cevaptan sonra ADD'e geri dönülür ve prosedür tekrarlanır. Tanımlanmış sayıda problem çözülene kadar bu sürer. Bu sayıya ulaşılınca tekrar ALLSUMS'a ve dolayısıyla kullanıcıya dönülür.*

*Diğer toplama tipleri de yaklaşık aynı yolla üretilir, ama SUBTRACT (çıkarma) ve DIVIDE (bölme) için birkaç uyarı gerekebilir. Bu düzeyde, çıkartma probleminin sol terafında daha büyük sayının bulunması, örneğin 3—8 = ? yerine 8—3 = ?, gerekecektir. IF; A :B[MAKE'' A:A + :B] satırı ile iki sayı toplanarak :B'den büyük değerde yeni bir :A sayısı yapılar. DIVIDE'da, seçilen iki sayı çarpılarak C oluşturulur ve böylece C, B ile tam olarak bölünebilir:*

*Bu prosedürler, daha karmaşık aritmetik kontrol prosedürleri için iyi bir başlangıçtır. Mesajları karmaşıklaştırmak, formatı değiştirerek toplamaları 5 + ? = 11 gibi düzenlerde vermek, farklı sunuç biçimleri için bir örnek olabilir. Ayrıca WRONG prosedürünün kaç kez çağrılacağını da sınırlayabilirsiniz. Çoğu proramda olduğu gibi, genişletme ve geliştirme olasılıkları sonsuzdur.*

## Toplama yapan örnek prosedürler

*Aşağıdaki ALLSUMS, SORT, ADD, SUBTRACT, MULTIPLY, DIVIDE, CHOOSE, CHECKANSWER, CHOSEN prosedürleri şimdilik temel matematik işlemlerin tümünü örneklemeye yeterlidir:*

```
TO ALLSUMS :MAX1 :MAX2
PR [İSTEDİĞİNİZ TOPLAMA TİPİ İÇİN BİR
      SAYI SEÇİN, KARIŞIK İŞLEM
      İSTİYORSANIZ 5'E ASIN]
PR [1, TOPLAMA İÇİN]
PR [2, ÇIKARTMA İÇİN]
PR [3, ÇARPMA İÇİN]
PR [4, BÖLME İÇİN]
MAKE ''CHOICE READCHAER
IF :CHOICE = ''5 [SORT]
IF :CHOICE = ''1 [ADD: APPLES:
      PEARS]
IF :CHOICE = ''2 [SUBTRACT: APPLES
      :PEARS]
IF :CHOICE = ''3 [MULTIPLY: APPLES
      :PEARS]
IF :CHOICE = ''4 [DIVIDE: APPLES:
      PEARS]
END
```

```
TO SORT :MAX1 :MAX2 :NUMBER
MAKE ''X (INT :NUMBER/3)
ADD :MAX1 :MAX2 :X
SUBTRACT :MAX1 MAX:2 :X
MULTIPLY :MAX1 :MAX2 :X
DIVIDE :MAX1 : MAX2 : X
END
```

```
TO ADD :MAX1 :MAX2 :NUMBER
IF :NUMBER = 0 [STOP]
MAKE ''A CHOOSE :MAX1
MAK E''B CHOOSE :MAX2
```

```
PR (SE :A [+] :B [=?])
MAKE ''TOTAL :A + B
MAKE ''ANSWER FIRST READLIST
CHECKANSWER :ANSWER :TOTAL
ADD :MAX1 :MAX2 (:NUMBER-1)
END
```

```
TO SUBTRACT :MAX1 :MAX2 :
NUMBER
IF: NUMBER = 0 [STOP]
MAKE ''A CHOOSE :MAX1
MAKE ''B CHOOSE :MAX2
IF :A   :B [MAKE ''A :A + :B]
PR (SE :A [—] :B [= ?])
MAKE ''TOTAL :A - :B
MAKE ''ANSer FIRST READLIST
CHECKANSWER :ANSWER :TOTAL
SUBTRACT :MAX1 :MAX2
(:NUMBER-1)
END

TO MULTIPLY :MAX1 :MAX2
:NUMBER
IF :NUMBER = 0 [STOP]
MAKE ''A CHOOSE :MAX1
MAKE ''B CHOOSE :MAX2
PR (SE :A [X] :B [= ?])
MAKE ''TOTAL :A * :B
MAKE ''ANSWER FIRST READLIST
CHECKANSWER :ANSWER :TOTAL
MULTIPLY :MAX1 :MAX2
(:NUMBER—1)
END
```

```
TO DIVIDE :MAX1 :MAX2
:NUMBER
IF :NUMBER = 0 [STOP]
MAKE ''A CHOOSE :MAX1
MAKE ''B CHOOSE :MAX2
MAKE ''C :A * :B
PR (SE :C [/] :B [= ?])
MAKE ''TOTAL :C / :B
MAKE ''ANSWER FIRST READLIST
CHECKANSWER :ANSWER: TOTAL
DIVIDE :MAX1 :MAX2
(:NUMBER1)
END


TO CHOOSE :NUMBER
OUTPUT 1 + RANDOM :NUMBER
END


TO CHECKANSWER :ANSWER
:TOTAL
IF : ANSWER = :TOTAL [PR CHOSEN
     :PRAISE] [WRONG :TOTAL]
END
```

```
TO WROGN :TOTAL
PR [DOĞRU DEĞİL]
PR [TEKRAR DENE]
MAKE ''ANSWER FIRST READLIST
IF :ANSWER = :TOTAL [PRINT (IYI)]
     [YANLIŞ :TOTAL]
ALLSUMS :APPLES :PEARS
END

TO CHOSEN :PRAISE
OUTPUT ITEM (1  + RANDOM COUNT
     :PRAISE) :PRAISE
END

MAKE ''PRAISE [BERBAT] [İYİ]
     [BU DOĞRU] [MÜKEMMEL]
```

# İkili Sayılar

Kısa bir program kullanırken, bilgisayara sayıları hem normal ondalık formda, hem de ikili ya da başka bir sayı tabanına göre yazdırmak mümkündür. Bu program kısa gözükmesine karşın, iyi çalışır, çünkü küçük ya da büyük olsun, ikili sayıların hesaplanmasında aynı yöntem kullanılır. Aşağıdaki çizelge 1'den 10'a kadarki sayıları ikili formda gösteriyor:

| | |
|---|---|
| 1 | 1 |
| 2 | 10 |
| 3 | 11 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |
| 8 | 1000 |
| 9 | 1001 |
| 10 | 1010 |

Ondalık saymada olduğu gibi her sayı, bulunduğu sütuna (haneye) göre farklı değer taşır (yüzler, onlar, birler gibi); dolayısıyla ikili saymada 1, her sütunda farklı bir değerdedir. Ikili tabanda en sağ sütun 1'leri, sonraki 2'leri, üçüncü 4'leri, dördüncü 8'leri vb. gösterir. Yukarıda gösterildiği gibi, on tabanına göre 10, ikili sistemde bir sekiz ve bir ikiden oluşmuştur. Sütunların ilişkisini açıklamanın bir başka yolu da, ondalık bir ''1''in her sütunda, sağındaki sütundan on kat büyük değerde olduğudur (100 sayısının yüzler sütunundaki ''1'', onlar sütunundakinin on katı değerdedir). İkili sistemde de aynı ilişki vardır, bu kez bir sütundaki 1, sağındaki sütundakinin iki katı değerdedir (1110, bir 8, bir 4, bir 2 ve bir tane de 1 anlamına gelir). Sütunlar arasındaki ilişkiyi akılda tutarsak, ondalık sayıları ikiye bölüp kalanı saklayarak, bunları ikili sisteme çevirebiliriz. Bu yolla yalnızca tam sayıları kullanabilirsiniz, yoksa 3'ü 2'ye böldüğünüzde 1.5 çıkacak ve işlem yürümeyecektir! Eğer örnek olarak 6'yı alırsak, ikili sisteme çevrilmesi şöyle olur:

6 bölü 2 = 3 (kalan 0)
3 bölü 2 = 1 (kalan 1)
1 bölü 2 = 0 (kalan 1)

Şimdi kalanları alır ve sondan başa doğru okursanız, ikili sistemde 110 yaptığını göreceksiniz. Eğer altı yerine üç'ü denerseniz, yaklaşık aynı yöntem kullanılabilir ve sonuç 11, doğ-

rudur. Bu prosedür üzerine biraz daha çalışırsanız nasıl çalıştığını ve ikili sayıyı neden yukarıdaki örnekte olduğu gibi alttan üste doğru okumanız gerektiğini anlayacaksınız. On bir'i ikili olarak yazmaya çalıştığımızı düşünelim. Üst tarafında şu satır yazılı bir çizelge oluşturabilirsiniz:

```
16   8   4   2   1
```

Her ikiye bölme, sayının tek ya da çift oluşuna bağlı olarak yapılır. Eğer tek ise, kalanı, çizelgenin uygun sütununa yazarız. Onbir'in ikiye ilk bölünmesi, sayı tek olduğundan kalan verir ve bu da uygun sütuna yazılır:

```
16   8   4   2   1
                 1
```

Bu durumda beş'i ikiye bölmeniz gerekiyor. Bunda da bir kalan vardır ve ikinci sütuna yazılır:

```
16   8   4   2   1
                 1   1
```

Şimdi ikiye bölünecek olan sayı 2'dir. Çift olduğu için kalan vermez ve dörtler sütununa sıfır girilir:

```
16   8   4   2   1
            0   1   1
```

Şimdi gelinen aşamada elimizde 1 var. Bir'i ikiye tam olarak bölemediğimizden, kalan 1 olur ve sekizler sütununa girer. Artık elde sayı kalmamıştır ve sıfır bölü iki sıfır verdiğinden işlem burada durdurulabilir. Böylece ondalık 11; bir sekiz, bir iki ve bir tane de bir'e ayrıştırılmış oldu:

```
16   8   4   2   1
     1   0   1   1
```

Bu oldukça uzun açıklama LOGO ya da başka bir bilgisayar dilinde programlama için önemli bir noktayı açığa çıkartıyor. Eğer problemi analiz etmediyseniz, programlama hedefine ulaşmanız da hemen hemen imkansızdır.

İyi programlama sürecine gerçek anlamda girişmeden önce ikili sayı sistemi tam olarak özümsenmelidir.

Ondalık sistemi ikiliye çeviren bir LOGO programı şöyle yazılabilir:

```
TO IKILI :SAYI
IF :SAYI =0 [OUTPUT"]
OUTPUT WORD IKILI (INT :SAYI/2) (REM
      AINDER INT :SAYI/2)
END
```

Bu kısa bir program gibi gözükse de, boyundan büyük işler becerir! INT ve REMAINDER **(kalan)**, matematik kökenli bir sözcüktür. INT, sayının tam sayı kısmını verir; dolayısıyla INT 5 2, çıkış olarak 2.5 değil 2 verecektir. REMAINDER, bir sayı bir başkasına bölündüğünde kalan sayıyı keşfetmenin kullanışlı bir yoludur. REMAINDER INT 5 2, bize 1 verecektir (INT 5, 2'ye bölündüğünde REMAINDER 1'dir). REMAINDER her zaman iki sayıyla kullanılır; ilki, ikincisine bölünen sayıdır.

```
BINARY (INT :SAYI/2) (REMAINDER
      INT :SAYI/2)
```

Programın çekirdeği bu satırdır. Yukarıda açıklandığı gibi **(:SAYI)**, sayısı 2'ye bölünüp, kalanı **(REMAINDER INT: SAYI 2)** olarak saklanır. Bununla birlikte, prosedürün sonsuz olmasa da, tekrarlı olduğuna dikkat ediniz. Prosedürü tüm düzeyleriyle çizerseniz aşağıdaki şekli alacaktır.

```
IKILI 11
OUTPUT WORD IKILI (INT 11/2)
  (REMAINDER INT 11 2)
                    '' 1011
```

```
IKILI 5
OUTPUT WORD IKILI (INT 5/2)
  (REMAINDER INT 5 2)

                    '' 101
```

```
IKILI 2
OUTPUT WORD IKILI (INT 2/2)
  (REMAINDER) INT 2 2)

                    '' 10
```

```
IKILI 1
OUTPUT WORD IKILI (INT 12)
 (REMAINDER INT 1 2)
                    '' 1
```

```
IKILI 0
IF SAYI=0 [OUTPUT'']
                    ''
```

OUTPUT'un ayrıntılı açıklaması daha ileride verilecektir. Şu an için bilmeniz gereken, prosedürün her düzeyinin yalnızca bir tek bilgi parçası OUTPUT edebildiği (çıktı verebildiği), böylece İKİLİ'nin, prosedürün en alt düzeyinden başlayarak üste doğru yürüdüğü ve daha sonra WORD'un, her düzeydeki çıktıları biraraya getirdiğidir. Sonuç olarak, en alt düzey bir boşluk verir. Bir üstteki düzeyde, WORD bunu REMAINDER'da saklanan cevapla birleştirir ve çıktı olarak ''1 verir. Bir sonraki düzeyde sıfır eklenir ve 1011'e ulaşana kadar bu sürer.

Tekrarın bu şekilde yapılması önemlidir; böylece ikili sayı (büyükler solda olmak üzere) doğru bir sıralamayla çıkacaktır. Dolayısıyla, prosedür en alt düzeyden başlamalı ve sayıları tekrar biraraya getirerek size cevabı vermelidir. İKİLİ'nin verdiği cevabı TERSİKİLİ prosedürü ile kontrol edebilirsiniz:

```
TO TERSİKİLİ :SAYI
IF :SAYI='' [OUTPUT 0]
OUTPUT (2* TERSİKİLİ BUTLAST :SAYI)
     ×(LAST :SAYI)
```

Bu prosedür de bir önceki gibi çalışır, ama bu kez son sayıyı dikkate almaz BUTLAST: SAYI) ve sonra TERSİKİLİ'nin bir sonraki düzeyini çağırır. Aşağıdaki şekil bu prosedürün nasıl çalıştığını gösteriyor:

```
TERKSİKİLİ 1011
OUTPUT (2* TERSİKİLİ 101)+1
                    2*5 +1=11
```

```
TERSİKİLİ 101
OUTPUT (2* TERSİKİLİ 10)+1
                    2*2 +1=5
```

```
TERSİKİLİ 10
OUTPUT (2* TERSİKİLİ 1) +0
                    2 +1 +0=2
```

```
TERSİKİLİ 1
OUTPUT (2* TERKSİKİLİ) +1
                    2*0 +1= +1
```

```
IF 0='' [OUTPUT 0]
                    0
```

Birlikte kullanıldıklarında bu prosedürler, sayıları bir tabandan ötekine çeviren bir programa temel olabilir. Yukarıdaki açıklama karmaşık görülebilir ama, LOGO'nun karışık konuların (burada sayı tabanı matematiği ve programlaması), halledilmesinde ne kadar kullanışlı olduğunu gösteriyor. Bilgisayarın sizin yerinize düşünemediğini de açıkça sergiliyor!

Şimdi gelen programlar sayıları bir tabandan ötekine çevirir ve ayrıca toplamlar yapar ve sonucu tekrar uygun tabana çevirerek cevabı verir.

Kuşkusuz bu, diğer toplama türlerine de uygulanabilir. (NOT: Aşağıdaki programlarda RADIX, bir sayma sisteminin tabanı için kullanılan teknik bir terimdir.)

## LOGO'yla Yapılabilen Diğer Aritmetik İşlemler

LOGO'da sayılarla uğraşırken yararlı olabilecek diğer sözcükleri de belirtmek gerekiyor. Bunlardan bir çifti PRODUCT **(çarpım)** ve SUM **(toplam)** oluşturur. Bu ikisi, iki sayıyı alır ve PRODUCT'da çarpar, SUM'da toplar. İkiden fazla sayıyı birlikte toplamak ya da çarpmak istediğinizde de bunları kullanabilirsiniz, ama bu kez tüm sayıları ve komutu parantez içine almanız gerekir. (SUM 3 4 45 14), sonuç olarak 66'yı verecektir. (PRODUCT 3 14 5) ise 210 sonucu verecektir. Bu komutları kullanarak SUMS örneğini daha büyük çocuklar için daha karmaşık bir duruma getirebilirsiniz.

## RANDOM

RANDOM, rastlantısal sayılar ürettiği için kullanılır. Anlaşılacağı gibi, tanımlanmış bir aralık içinde rastlantısal tam sayılar seçer.

RANDOM 10, sıfırdan dokuza kadar olan aralıkla rastlantısal bir sayı üretecektir. (Bu işlemde, aralığın içindeki her tam sayı eşit seçilme olasılığına sahiptir.) Bir ile on arasında bir sayı seçmek için, seçilen sayıya 1 eklemek gerekir. CHOOSE prosedürünü bunu işleme sokar:

```
TO CHOOSE :NUMBER
OUTPUT 1 + RANDOM :NUMBER
END
```

Eğer seçilen sayıya bir eklemek yerine aralığın üst sınırını bir artırırsanız (ki bu durumda 11 olur), bir ile on arasında değişen bir aralık elde etmezsiniz. Sıfır da seçilebilir durumda olduğundan, şimdi aralık sıfırdan ona kadar olacaktır. RANDOM'ı, bir zar atışını simüle etmekte şöyle kullanabilirsiniz.

```
TO ZAR
OUTPUT 1+RANDOM 6
END
```

RANDOM kullanılan prosedürleri kontrol edeceğiniz bir başka komut da RERANDOM'dır. RANDOM içeren prosedürü run etmeden önce RERANDOM'ı yazarsanız, bu prosedürün bundan sonraki run edilişlerinde yaklaşık olarak aynı rastlantısal sayılar dizisini elde edersiniz. Böylece prosedürü çeşitli kereler çalıştırırken ortaya çıkan sorunlar azalır, en azından sayılar sürekli değişip durmaz.

LOGO'ların çoğunda SIN **(sinüs)** ve COS **(kosinüs)** gibi trigonometrik ifadeler de vardır. Problem çözmeyi kolaylaştıran bir başka komut da SQRT **(karekök)**'dür. En son olarak, kesirli sayıları en yakın tam sayıya yuvarlayan ROUND komutunu da sayabiliriz. INT ile ROUND arasındaki farkı burada belirtmek yararlı olacaktır: INT 5.9 = 5 olmasına karşın ROUND

5.9=6'dır. Yaklaşık değerli sonuçların istendiği problemlerde ROUND kullanışlı olabilir. (ROUND:NUMBER) ifadesi INT (:NUMBER+0.5)'e eşittir. Eğer, örneğin karışık faiz hesabı yapmak isterseniz, kuruşların hesabı karıştırmasını böylece önleyebilirsiniz.

## Sonuçlar

Bu kısımda gördüğünüz gibi, LOGO sayısal işlemler için geniş olanaklar sunuyor. Gösterilen tüm programlar, yeni matematiksel düşünceler geliştirmek üzere yeni durumlara uyarlanabilir. İkili sistem için sunulan problemler sonsuzdur; ayrıca negatif sayıların açtığı geniş alanı da unutmamalı. Bütün bunları gözönüne alırsanız, LOGO'yla hem matematiksel problemlerin, hem de programlama tekniklerinin kavranacağı açıktır.

## Sözcüklerle, Üstelerle Oynamak

LOGO'nun kökeninde, yapay zeka araştırmalarında yaygın olarak kullanılan LISP **(List Processor: Liste İşlemcisi)** adlı bir dil bulunduğu için; sözcükler ve listeler tarihsel olarak LOGO'nun çekirdeğini oluştururlar. Çocukların bilgisayar kullanabilmeleri yönünden, Seymour Papert'in kaplumbağayı eklemesi önemli bir hamleydi. Ama kaplumbağa kullanılmaya başladıktan sonra bile, listenin sentaksı ve ilkelerinin çoğu aynen bırakıldı. Bir başka deyişle, sözcük ve liste işlemenin bazı özellikleri farkına varmadan LOGO'da kullanılıyor. LOGO'nun sözcükler ve listelerle oynama yetisi çok çeşitli sözcük oyunlarını olanaklı kılar. LOGO daha genel olarak büyük veri tabanları için hazırlanan programların geliştirilmelerini de kolaylaştırır.

### Sözcük Nedir?

LOGO'da bir sözcük, aynen Türkçe'deki gibi karakterlerden (harflerden) oluşur. ARABA, KAPI, MASA birer sözcüktür. Ayrıca LOGO'da tanım biraz daha geniştir: 345.5, YAKLAŞ! ve YILDIZ2 de sözcüktür. Daha biçimsel bir deyişle, **bir LOGO sözcüğü, bir öncesi ve sonrası boş bırakılan bir karakter kümesidir.** Her karakter, sözcükteki diğer karakterlerden ayrılabilen bir elemandır. (Bu ileride çok önemli olacak bir nokta!)

FORWARD, IF, BACK vb. LOGO'da halen kullandığımız sözcüklerdir. Yeni bir prosedür yarattığımızda ona bir de ad verdiniz, örneğin "KARE: Bu da biz sözcüktür. Çift tırnağın **("")** nedeni daha sonra açıklanacaktır.

## Liste Nedir?

Bir LOGO listesi, köşeli parantezler arasına yerleştirilmiş sözcük ya da sözcük kümeleridir. GUZEL EV veya PRINT COK BASARILI gibi ifadeler, kaplumbağa grafiğinde kullandığımız listelerdir. İlkinde listenin iki elemanı vardır; GUZEL ve EV. İkinci listenin de iki elemanı vardır; PRINT ve COK BASARILI. Eğer bir sözcükten fazlasının listede bir bütün olarak ele alınmasını istiyorsanız, bu sözcükleri köşeli paranteze alın: [COK BASARILI] gibi.

## Sözcük ve Listelerle Neler Yapabilirsiniz?

Bilgisayar dilindeki sözcük ve listeler üzerinde konuşurken ortaya çıkan sorunlardan biri, bunları tartışırken Türkçe sözcük ve cümleler kullanıyor olmamızdır. Türkçe'de sözcükleri kullanışımızla ilgili birkaç örneğe bakarsak, LOGO sözcükleri ve listelerinin kullanım olanakları da belirginleşir.

KEDI MUZ KARD KOPEK YUMURTA

Yukarıdaki adların tümü sözcüktür. Bunlar nesneleri belirtirler, ama kendi içlerinde sınırlı anlamları vardır. Belirli bir kedinin ya da yumurtanın tavuğa mı, ördeğe mi ait olduğu konusunda hiçbir bilgi yoktur. Tek bir sözcüğü yapabildiği, anlamını veren nesneyi ya da duyguyu göstermekten ibarettir.

ZERRIN OZKAN IVEDIK CADDESI 56
ANKARA KIZILAY

OZAN TANYER HATAY CAD 42 IZMIR

LEVENT TUNCEL INONU CAD 31
ESENTEPE

Yukarıdaki üç grup ad ve adres, bir adres defterinden olabilir. LOGO terimleriyle, her biri bir adrakaşa ait bilgi taşıyan üç liste olarak belirtilebilirler. Adres defterlerinin çoğunda olduğu gibi, adlar alfabetik sıraya sokulurlar. Eğer defterinize

ONUR ARICA
BUYUKDERE CADDESI 88
LEVENT

eklemek isterseniz, öteki adresleri buna göre kaydırmanız gerekir. Ama bu sıralamayı, bilgisayara basitçe onun adını vererek sağlayamazsınız. Bunu yapmak için standart yol, adları geldiği sırada ele almaktır. İlk boşluk kolaydır; yalnızca FIRST adres diyebilirsiniz. Sonraki adres için ilk değil, ilkten sonraki olarak tarif edebilirsiniz. Listenin sonuna gelindiğinde LEVENT TUNCEL için LAST girdi kullanılabilir. Bir kere listede konum belirlendikten sonra yeni ekleme, çıkartma ya da yer değiştirmeler yapılabilir. Listeyi soyadlarına göre belirlemek isteseniz bile, her bir kişiyle ilgili bilginin toplu halde tutulması gerekir. Bunun için listeyi oluşturan parçalar **(alt listeler)**, paranteze alınır; bu da bir listeler listesi verir; örneğin

[[ZERRIN OZKAN] [IVEDIK CAD 56] [ANKARA KIZILAY]]

[[ONUR ARICA] [BUYUKDERE CAD 88] [LEVENT]]

Ad, adres ve semtin ayrı listelerde yeralma nedeni, adresin aynı parçalarını belirleyebilmek ve ayrı ayrı da kullanabilmektir. Eğer çok uzun bir adres defteriniz varsa ve ESENTEPE'de kimlerin oturduğunu öğrenmek istiyorsanız, listelerin üçüncü parçalarına bakabilirsiniz. Adresler tek ve uzun listeler halinde olsaydı aradığınızı ayırmak için daha zor olacaktı.

Örnek olarak listelerin şöyle düzenlendiğini düşünün:

[ZERRIN OZKAN IVEDIK CAD 56 ANKARA KIZILAY]

[OZAN TANYER HATAY CAD 42 IZMIR]

Eğer listeler böyle düzenlenmiş olsaydı, Zerrin Özkan'ın adresindeki son parça ANKARA yerine KIZILAY olacaktı. Ozan Tanyer'in adresinde bu sorun yoktur, ama listeleri hazırlarken ileride ortaya çıkabilecek tüm sorunları gözönüne almak gerekir.

Listelerle yaptığımız işlemleri sözcüklere de uygulayabiliriz. Bu durumda sözcükleri listeler halinde belirlemek yerine, harfleri sözcükler halinde belirleyebiliriz. Böylece daha önce baktığımız sözcük listesi alfabetik sıraya dizilebilir, sözcük uzunluğuna ya da herhangi önemli bir şeye göre kümelenebilir.

Bir sözcüğü, kendisini oluşturan harflerle ayıran bir program yazdığınızı düşünelim. Eğer sözcük yerine bir liste verirseniz, aynı programın bu kez listeyi sözcüklere ayırdığını göreceksiniz. Bu LOGO'nun olağandışı bir özelliğidir, sunduğu olanaklar ise ileride tartışılacaktır.

## PRINT

Sözcükler ve listelerle oynamaya başlamadan önce, bu komutu öğrenmeniz gerekir; çünkü bundan sonra yazacağınız prosedürler kaplumbağa-çizimli biçimler durumunda görüntülenmeyecek, ekrana yazılacak. PRINT, bu yazımı yapmanın en basit komutudur:

```
PRINT ''MERHABA
MERHABA
```

Yukarıda görüldüğü gibi, ardından çift tırnak ve biz sözcük olan PRINT, aynen BASIC'te olduğu gibi çift tırnaktan sonra gelen sözcüğün yazılmasını sağlar. PRINT bir listeyle kullanıldığında, listenin **(köşeli parantezin)** içindekileri yazdırır:

```
PRINT [IYI GUNLER]
IYI GUNLER
```

```
PRINT [''IYI ''GUNLER]
''IYI ''GUNLER
```

PRINT, ekranda boş bir satır bırakılmasını da sağlar. Bunun için bilgisayarın ''boş sözcük'' ya da ''boş liste'' yazmasını istemek gerekir:

```
PRINT ''
PRINT [ ]
```

Eğer PRINT '' MERHABA'yı, MERHABA sözcüğüyle çift tırnak arasında bir aralık bırakarak yazarsanız, bilgisayar önce bir satırı boş bırakacak, sonra da MERHABA prosedürünü bulamadığını yazacaktır. Bunu yazmadan önce bilgisayar iki şey yapmaya çalıştı: Önce, aralığı gördüğü için bir satırlık boş yer bıraktı, sonra da MERHABA adlı prosedürü aradı, bulamadığı takdirde hata mesajı verecekti. PRINT'in kısaltması PR'dir.

## Sözcük ve Listelerdeki Elemanların Konumunu Tarif Eden Komutlar

FIRST LAST BUTFIRST BUTLAST

Bu komutları, biri sözcük öteki liste olan iki örnekte inceleyelim. Sözcük için ''MERHABA, liste için de [GÜNAYDIN SANA GUNES]'i alalım. Bunların liste mi, sözcük mü oldukları noktalama işaretleriyle belirtilir. Şimdi komutları sırayla uygulayalım:

```
PR FIRST ''MERHABA
M
PR FIRST [GUNAYDIN SANA GUNES]
GUNAYDIN

PR LAST ''MERHABA
A
PR LAST [GUNAYDIN SANA GUNES]
GUNES

PR BUTFIRST ''MERHABA
ERHABA
PR BUTFIRST [GUNAYDIN SANA GUNES]
   SANA GUNES

PR BUTLAST ''MERHABA
MERHAB
PR BUTLAST [GUNAYDIN SANA GUNES]
GUNAYDIN SANA
```

Görüldüğü gibi FIRST **(ilk)** ve LAST **(son),** sözcüğün ya da listenin ilk ve son elemanlarını belirtir. BUTFIRST **(ilki dışında)**, sözcüğün ya da listenin ilk elemanı çıkartıldığında geri kalan

kısmı belirtir. Bu komutlar birlikte de kullanılabilir:

```
PR FIRST BUTFIRST ''MERHABA
E
PR FIRST BUTFIRST [GUNAYDIN SANA
GUNES]
SANA

PR LAST BUTLAST ''MERHABA
B
PR LAST BUTLAST [GUNAYDIN SANA
GUNES]
SANA
```

Örnek listemiz üç parçadan oluştuğu için, başından da, sonundan da başlansa, FIRST BUTFIRST ve LAST BUTLAST aynı sonucu verecektir. Listeler böyle ayrılırken genellikle bir taraf seçilir ve o taraftan işlem yürütülür. LAST BUTFIRST ya da FIRST BUTLAST'ı kullanabilirsiniz, ama bu komutların vereceği sonuçları önemle dikkate almak gerekir.

Buraya kadar açıklanan komutlarla, bir listenin ya da sözcüğün seçtiğimiz haflerini ya da sözcüklerini yazdırabilirsiniz. Aşağıdaki prosedür, bu komutları kullanarak ilginç biçimler oluşturma yollarını gösterecektir.

SÖZCÜK.AÇMA ile SÖZCÜK.AÇMA1 prosedürleri son satırları hariç aynıdır. SÖZCÜK.AÇMA'daki BUTLAST'ın SÖZCÜK.AÇMA1'de BUTFIRST olduğuna dikkat ediniz.

```
TO SÖZCÜK.AÇMA :AD
IF EMPTYP :AD [STOP]
PR :AD
SÖZCÜK.AÇMA BUTLAST :AD
END
```

```
SÖZCÜK.AÇMA ''HUSEYIN
HUSEYIN
HUSEYI
HUSEY
HUSE
HUS
HU
H
?
```

```
? SÖZCÜK.AÇMA [TUGRUL VE ASLI
PASTAYI SEVER]
TUGRUL VE ASLI PASTAYI SEVER
TUGRUL VE ASLI PASTAYI
TUGRUL VE ASLI
TUGRUL VE
TUGRUL
?
```

```
SÖZCÜK.AÇMA1 :AD
IF EMPTYP :AD [STOP]
PR :AD
SÖZCÜK.AÇMA1 BUTFIRST :AD
END
```

```
SÖZCÜK.AÇMA1 ''HUSEYIN
HUSEYIN
USEYIN
SEYIN
EYIN
YIN
IN
N
```

```
SÖZCÜK.AÇMA1 [TUGRUL VE ASLI
PASTAYI SEVER]
TUGRUL VE ASLI PASTAYI SEVER
VE ASLI PASTAYI SEVER
ASLI PASTAYI SEVER
PASTAYI SEVER
SEVER
```

```
SÖZCÜK.AÇMA2 :AD
IF EMPTYP :AD [STOP]
PR LAST :AD
SÖZCÜK.AÇMA2 BUTLAST :AD
END
```

```
SOZCUK.ACMA2 ''HUSEYIN
N
I
Y
E
S
U
H
?
```

```
SOZCUK.ACMA2 [TUGRUL VE ASLI
PASTAYI SEVER]
SEVER
PASTAYI
ASLI
VE
TUGRUL
?
```

```
SOZCUK.ACMA3 :AD
IF EMPTYP :AD [STOP]
TYPE LAST :AD
TYPE''
SOZCUK.ACMA3 BUTLAST :AD
END
```

```
?SOZCUK.ACMA3 ''HUSEYIN
NIYESUH?
```

```
?SOZCUK.ACMA 3
[TUGRUL VE ASLI
PASTAYI SEVER]
SEVER PASTAYI ASLI VE TUGRUL
```

## Sözcük ve Liste Prosedürlerinde Tekrar Kullanımı

Başka bir komutun açıklamasına geçmeden, bir sözcük ya da liste prosedürlerinde tekrarlamanın önemini belirtmek yararlı olacaktır.

Önceki prosedürlerde bu nokta açıkça görülmektedir. Tekrarlamayı etkin kullanabilmek için prosedürün başlık satırında bir değişkeni tanımlamak önemlidir. Böylece, parçalamak istediğiniz herhangi bir sözcük ya da listeyle prosedürü çağırabilirsiniz. Örneğin MERHABA'yı alalım:

MERHABA bütün halde bir sözcüktür.

ERHABA, MERHABA'nın BUTFIRST parçasıdır; ilk parça çıkarıldığında geri kalan kısımdır. Prosedürü yine çağıralım, ama bu kez BUTFIRST :AD'ı kullanalım. Bu, SOZCUK.ACMA prosedürünü ''ERHABA'yla çağırmaya benzeyecektir. Prosedür ERHABA'yla yinelenecektir. Bu kez E çıkarılacak ve RHABA kalacak ve böyle sürecektir. Dolayısıyla tekrarlama, aynı prosedürün birinci, ikinci ve sonra üçüncü harflerin çıkarılmasıyla geriye kalan kısımlar üzerinde uygulanmasını sağlar.

IF EMPTYP :AD [STOP] satırı önemlidir, çünkü :AD'da daha başka bir sözcük ya da liste parçası kalıp kalmadığını denetlemektedir.

Eğer kalmamışsa, prosedür durur; varsa bir sonraki harf ya da sözcüğe geçer.

SOZCUK.ACMA2, PR:AD'ı PR LAST:AD'la değiştirerek sözcüğün bir harfini ya da listenin bir sözcüğünü yazdırır. Yapılan, listeyi ters çevirmektir. TYPE'ı kullanarak sonucun tümünü bir satıra yazdırabilirsiniz. PRINT, her kullanıldığında yeni bir satırı başlatır. TYPE ise yeni bir satıra geçmeyi durdurur. Ardından bir aralık bırakılan TYPE'', harfler ya da listenin parçaları arasında aralık bırakır. Eğer SOZCUK.ACMA3'ten bu satırı çıkarırsanız, tüm harflerin ve sözcüklerin birbiri ardısıra yazıldığını göreceksiniz.

## Sözcükleri ve Listeleri Birleştirme

Buraya kadar sözcükleri ve listeleri parçalamakla ilgilendik. Bunları yeniden biraraya getirmek için kullanılan komutlar da vardır.

## WORD

Bu komut, iki parça bilgi gerektiren ve bunları birleştiren bir işlemdir. Örneğin, WORD ''KALI ''YOR'un çıktısı KALIYOR olacaktır. Eğer üç parçayı birleştirmek istiyorsanız WORD ve girdileri paranteze almak gerekir:

(WORD ''KALI ''YOR ''UM), KALIYORUM'u verir.

## LIST

Tahmin edilebileceği gibi, WORD'la aynı şekilde çalışır ve sözcükleri liste oluşturmak için birleştirir. LIST ''AHTAPOT ''GUZELDIR, [AHTAPOT GUZELDIR] listesini verir. İkiden fazla parça için gene paranteze almak gerekir: (LIST ''AHTAPOT ''COK ''GUZELDIR), [AHTAPOT COK GUZELDIR] çıktısını verir. Listelerde değişiklik yapma olanağı WORD'a göre biraz daha geniştir. Yukarıda gördüğümüz gibi, adreslere bakarken bir listeler listeniz olabilir, böylece aşağıdaki yöntem de kullanılabilir:

```
(LIST [MEHMET BORA] [26 NISAN 1960]
      [MEZUN])
[[MEHMET BORA] [26 NISAN 1960]
      [MEZUN]]
```

Böylece üç küçük liste, daha büyük bir liste içine alınmış oldu.

## SENTENCE (SE)

Bu komut, bir prosedürden gelen son mesajı yazar ve sözcüklerle, listelerle ya da her ikisinin karışımıyla kullanılabilir. Birleştirilecek ve liste haline getirilecek iki şey gerektirir. Eğer SE [SINEMA VE OYUNCULAR] [KAMERA VE IZLEYICILER] verirseniz, SINEMA VE OYUNCULAR KAMERA VE IZLEYICILER sonucu alırsınız.

Eğer bu çıktının ekranda yazılmasını istemeseniz, liste parantezsiz olarak yazılacaktır. Fakat bu yeni uzun listeden, eski iki listeye geri dönemezsiniz. SENTENCE komutunun kullanılması, eski liste işaretlerinin ortadan kalkmasına neden olur.

## FPUT VE LPUT

Şimdiye dek, birleştirmede kullanılan komutlar, gerekli bilginin doğru sırayla verilişiyle tanıtıldı. Bu durum her zaman zorunlu değildir! Başka durumlarla uğraşabilmek için LPUT ve FPUT komutlarını ele alalım. Bu komutlar bir sözcüğü listenin başına (FPUT) ya da sonuna (LPUT) yerleştirirler. Eğer PR LIST FPUT ''SES [GORUNTU] verilirse, liste [SES GORUNTU] OLUR. PR LIST LPUT ''SES [GORUNTU] verildiğinde, liste [GORUNTU SES] durumuna gelir. Formüle etmek gerekirse, listeye eklenen parça listede ilk sırayı alır. Bu liste boş da olabilir; örneğin PR LIST FPUT ''FILM [] için [FILM] çıktısı alırız.

## COUNT

Bu komut bir listedeki parçaları sayar. COUNT [SAHIN HUSEYIN BULCAY MUAMMER], 4 sayısını verecektir. COUNT [[YILBASI] [BAYRAM]], 2 sayısını verecektir.

## ITEM

Çoğunlukla COUNT'la birlikte kullanılır. Diyelim ki yukarıda COUNT ile sıraladığımız dostlarımızın adlarını değişken yaptık ve bu değişkene :DOST dedik. Bu durumda PR ITEM 2 :DOST ifadesi; bu listenin ikinci elemanı olan HUSEYIN'i çıktı olarak verecektir. Tatillerimizi değişken yapsaydık PR ITEM 2 :TATIL için BAYRAM yanıtını alacaktık. Aşağıdaki prosedürde olduğu gibi iki komut birlikte kullanılabilir:

```
TO CHOOSE : EMRE
PR ITEM (1 + RANDOM COUNT:EMRE):
     EMRE END
```

Bu prosedür bir listeyi veya sözcüğü alır ve önce :EMRE listesindeki parçaların sayısını bulur. Sonra bulduğu sayıyı üst sınır alarak rastgele bir sayı seçer.

(1 + RANDOM COUNT :EMRE)

Bu, PR ITEM X :EMRE'yi kullanmak için ITEM'e bir sayı verir. Bu örnekte neler olduğunu görelim. 'EMRE aşağıdaki liste olsun:

[MUZIK RESIM SINEMA TIYATRO HEYKEL]

COUNT:EMRE, 5 sayısını verecektir. 1+RANDOM 5 ifadesi de 1 ile 5 arasında bir sayı verecektir; bunun 3 olduğunu kabul edelim. Şimdi CHOOSE ile PRINT ITEM 3:EMRE birlikte kullanıldıında, çıktı olarak üçüncü eleman olan SINEMA alınacaktır.

Aşağıdaki prosedür şimdiye dek kullanılan bazı komutlarla cümleler oluşturmaktadır. Çıkan cümleler belki şimdiye kadar yazılmış en güzel Türkçe cümleler değil, ama en azından yepyeni.

```
TO CUMLA
PR (SE CHOOSE :OZNELER CHOOSE
     :ZARF CHOOSE :FIIL)
CUMLE
END
```

```
MAKE ''ADLAR [COCUKLAR KEDILER
    KOPEKLER AHTAPOTLAR]
MAKE ''FIIL [YUZUYORLAR CALISIYORLAR
    YIYORLAR KACIYORLAR]
MAKE ''ZARF [HIZLA SESSIZCE GIZLICE
    DIKKATLICE]
```

Elde edilebilecek cümlelerden bazıları!

```
OGLANLAR HIZLA YUZUYORLAR
KEDILER GIZLICE YIYORLAR
AHTAPOTLAR DİKKATLICE
CALISIYORLAR
```

## Prosedürlerinize Bilgi Verme

Bütün prosedürler, üzerinde çalışacakları bilgi gereksinirler. Sözcük ve listelerle oynarken, değişkenler, kaplumbağa grafiğindeki kadar önemlidir, ama farklı şeyler için kullanılırlar. Örneğin :EMRE gibi değişkenler, prosedürde kullanılacak sözcük ve listeler için bir depo görevi görürler. Prosedürlerinize bilgi vermek için birtakım yollar vardır. Hangisini seçeceğiniz, prosedürü nasıl kullanacağınıza bağlıdır.

## MAKE

Değişkenlerinize bir değer vermenin başka bir yolu da MAKE'dir.

MAKE ''EMRE [MUTLU YILLAR]

satırı, :EMRE'nin [MUTLU YILLAR]'ı, içermesini sağlar. Şimdi PR:EMRE derseniz [MUTLU YILLAR] yazacaktır. Bu komutu, EMRE'nin değerinin, kullanılacağı prosedüre özgü kalacağı durumlarda kullanabilirsiniz. Böylece ''EMRE olarak işaretlediğiniz bütün sözcükleri ve listeleri her prosedürde çağırabilirsiniz, başka bir deyişle ''EMRE bir genel değişkendir. Daha önceki örnekte verilen ''ADLAR, ''FIIL ve ''ZARF da birer genel değişkendi. Bir başka prosedürde ''ADLAR geçtiğinde, yine aynı sözcük listesini kullanacaktır.

## READCHAR ve READUST

Bilgisayarı tek tuşa dokunarak işletebilmek için bazı kaplumbağa prosedürlerinde, özellikle SINGLEKEY'de, bu komutlar kullanılmaktadır. Klavyeden bir girdi verilene dek her iki komut da çalışmadan bekler. READCHAR, klavyeden girilecek bir tek karakteri bekler. Bir kez tuşa basıp bilgi verdikten sonra, prosedürün kalan kısmı boyunca bu korunur. Öte yandan READUST için bir dizi tuşla girilmesi gerekir, girilen bilgi bir sözcük ya da liste olabilir. Yazmayı bitirdikten sonra RETURN tuşuna basılmalıdır. Kullanımları gözönüne alındığında, bu iki komutun MAKE ile yakın bağı vardır. MAKE ''DENIZ READUST, hangi sözcükler ya da liste yazılmışsa bunları DENIZ'ın değeri olarak almayı sağlar. Aşağıdaki örnek bu komutların nasıl kullanıldığını göstermektedir:

```
TO SOHBET
PRINT [ADINIZ NEDIR?]
MAKE ''YAN READUST
PRINT (SE [KAC YASINDASINIZ,] :YAN ''A)
PRINT [EMIN MISINIZ!]
MAKE ''X READCHAR
IF :X = ''Y [PRINT [DURUST OLDUGUNUZA
SEVINDIM]] [PRINT [YALNIZCA
     KADINLAR
YASLARI HAKKINDA YALAN SÖYLER!]]
PRINT SE [BUGUN NE YAPIYORSUNUZ]
     :YAN
MAKE ''YAN2 READUST
IF: YAN2 = [HIC BIR SEY] [PRINT [KAFANA
TAKMA]] [PRINT 4SE [HOSUNUZA GITTI MI]
     : YAN2''?)]
END
```

Bu çok basit bir programdır, ama liste komutlarının nasıl kullanıldığını göstermek için yeterlidir.

Basketbolu, okulda olup bitenleri ya da başka konuları tartışmak için benzer programlar yazarak eğlenebilirsiniz. İyi işlemesi için, bilgisayarı kullananın ne gibi yanıtlar verebileceğini doğru tahmin etmelisiniz. Eğer MAKE'ı READCHAR ve READLIST'le birlikte kullanırsanız, bilgisayarı kullananın yazdığı her şeye başvurabilirsiniz; örneğin yukarıdaki prosedürde :YAN2 ya da :X böyledir.

Prosedür tekrarlıysa, uygun bir noktada yeni bir sözcüğü ya da cümleyi eklemek ve :YAN2 ya da :X'in değerlerini gereğince belirlemek elinizdedir.

## OUTPUT

OUTPUT komutunun amacı bir prosedürden ötekine bilgi aktarmaktır. Bu komutla ilgili en önemli nokta, her prosedürün çıktı olarak yalnızca bir tek bilgi parçası verebileceğidir.

Kaplumbağayla konuşmaya alıştığınızdan OUTPUT'u kullanmada ustalaşmak biraz zor olabilir. Liste kullanan programlarda, vereceğiniz bilginin iletileceği yeri de belirtmeniz gerekir. Bu açıklamalar, bir iki örnekle daha kolay anlaşılacaktır. İlkin, daha önce kullandığımız CHOOSE prosedürünü ele alalım:

```
TO CHOOSE :EMRE
PRINT ITEM (1 + RANDOM COUNT :EMRE):
    :EMRE
END
```

Burada bir başka prosedürü de kullanmak için PRINT'i OUTPUT'la değiştirmeliyiz:

```
TO CHOOSE :EMRE
OUTPUT ITEM (1 + RANDOM COUNT
    :EMRE):EMRE
END
```

Bir tek sözcüğün ekrana yazılması yerine, cümle kuran bir prosedürle birlikte kullanabilirsiniz:

```
TO TWITTER
PR (SE CHOOSE: ADLAR CHOOSE: ZARF
    CHOOSE:FIIL)
TWITTER
END
```

Biraz daha karmaşık olan aşağıdaki son örnek, LOGO'yla oluşturulacak diyaloglar üzerine aydınlatıcı olacaktır. Bu örnekte bir doktorla hastasının eğlendirici sohbeti oluşturuluyor:

```
TO DOKTOR
MAKE "YORUM [[DUSUNCENIZE GORE]
[OYLE HISSEDIYORSUNUZ KI]
    [SOYLEDIGINIZE GORE]]
MAKE "PAD [[DEVAM EDIN LUTFEN]
    [DINLIYORUM ANLATIN]
[PEK COK INSAN BOYLE
    DUSUNUR]]
MAKE "ZARF [[BEN SEN] [BENIM SENIN]
[SEN BEN]]
PRINT [NE GIBI BIR SORUNUNUZ VAR]
COMMUNICATE
END
```

```
TO COMMUNICATE
MAKE "INPUT = [GULE GULE] [PR [TEKRAR
GELIN, KONUSURUZ] STOP]
IF RANDOM2 = 0 [PRINT CHOOSE :PAD]
[RESPOND: INPUT]
COMMUNICATE
END
```

```
TO RESPOND :INPUT
PRINT (SE (CHOOSE :YORUM) (SWITCHED:
INPUT :ZARF)
END
```

```
TO SWITCHED :INPUT :ZARF
IF EMPTYP :INPUT [OUTPUT []]
OUTPUT FPUT (COMPARE FIRST :INPUT
:ZARF) (SWITCHED (BUTFIRST :INPUT
:ZARF)
END
```

```
TO COMPARE :INPUT :ZARF
IF EMPTYP :ZARF [OUTPUT :INPUT]
IF :INPUT = FIRST FIRST :ZARF
[OUTPUT LAST FIRST :ZARF] [OUTPUT
COMPARE :INPUT BUTFIRST :PRONOUN]
END
```

```
TO CHOOSE :EMRE
OUTPUT ITEM 1 + (RANDOM COUNT
    :EMRE)
END
```

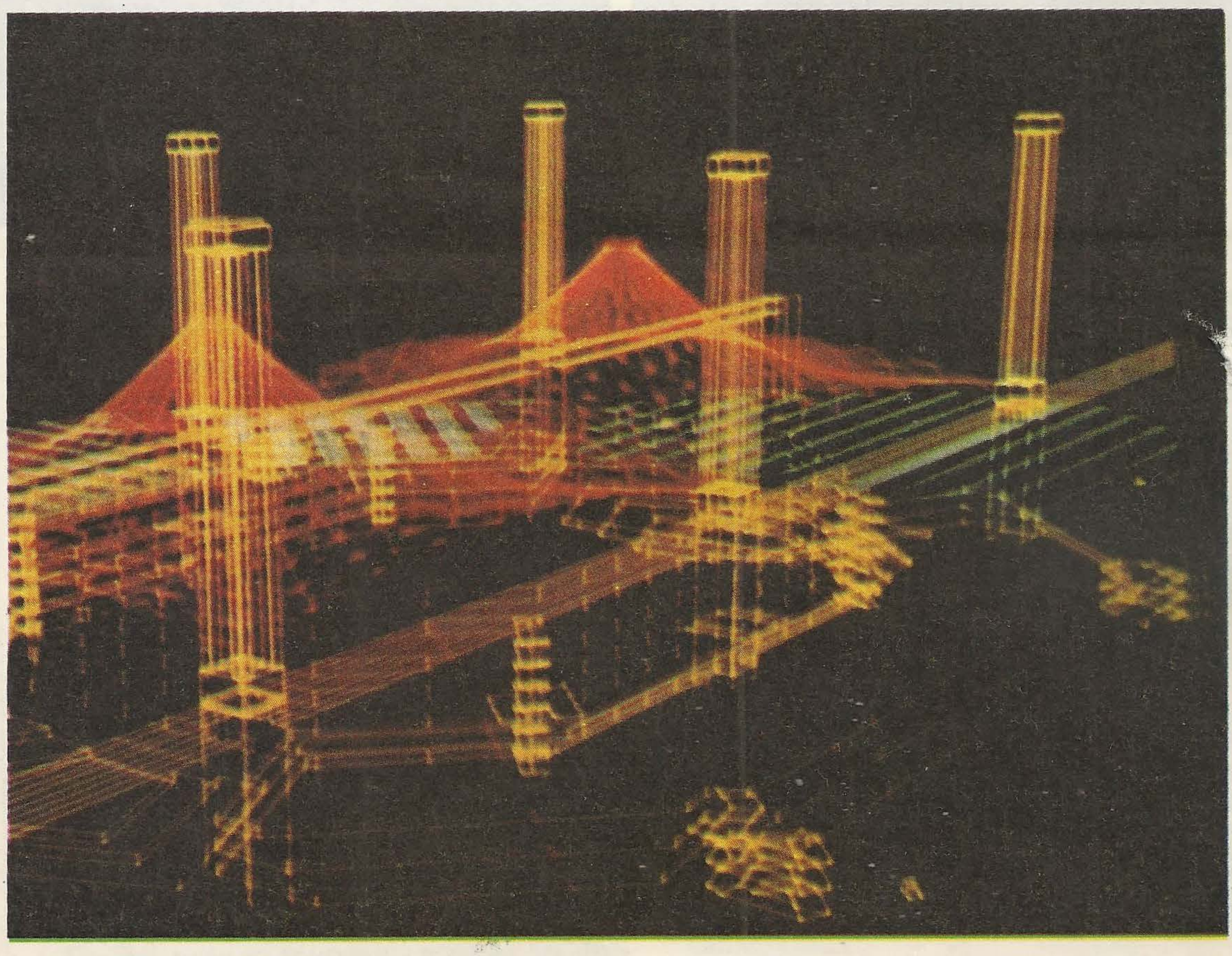
Verilen DOKTOR programı önce YORUM, PAD ve ZARF değişkenlerini tanımlıyor ve her birine bir yorum ya da sözcük listesi veriyor. Sonra da karşısındaki konuşmacı (hasta)nın yanıtlarına bağlı olarak konuşmayı sürdürüyor. Sonuçta da, GULE GULE yanıtı aldığında duruyor.

## Sözcükleri ve Listeleri Kullanmak İçin Kontrol Satırları

DOKTOR programı, liste işlemlerinde test satırlarının bazı kullanımlarını göstermektedir. Kamplumbağa grafiğinde olduğu gibi, IF x = y [doğruysa yapılacak] [yanlışsa yapılacak] test satırlarının sözcükler ve listelerle kullanılış biçimidir. Sözcükler ve listelerle kullanmak için tasarımlanmış birtakım kontrollar aşağıda kısaca tarif edilmiştir. Bazıları ötekilerden daha değerlidir, ancak bütünlüğü korumak için tümüne yer verilmiştir. Bütün bu kontroller bir bilgi parçasını bir başkasıyla karşılaştırırlar. Bir listenin öbürüyle aynı biçime sahip olup olmadığını veya bazı tür bilgiler içerip içermediğini denetlerler. Her durumda, kontrol satırının yapısı aynıdır: IF test sözcüğü (: denetlediği şey) (denetlediğine zıt olan şey)

### EMPTYP

Henüz görmüş olduğumuz gibi, özellikle tekrarlı yöntemlerde, listedeki sözcüklerin ya da



1938

sözcükteki harflerin tükenip tükenmediğini denetlemekte kullanılan bir kontroldur. Bazı durumlarda IF EMPTYP :OLCAY [STOP] gibi bir satır, işlemi veya programı sonuçlandırmak için kullanılabilir. EMPTYP'i OUTPUT için de kullanabilirsiniz- yukarda gösterildiği gibi.

## EQUALP

Bu, = simgesine denktir ve listelere, sayılara, sözcüklere uygulanabilir. Bunlar birbirinin aynı olmalıdırlar, yoksa FALSE (yanlış) çıktısı verir.

```
IF EQUALP :YUCEL ''GUNAYDIN [PRINT
    [SELAM]]
```

eğer GUNAYDIN tek sözcükse mesajı yazacaktır, fakat liste biçimindeyse yazmayacaktır. Başka deyişle, YUCEL'de ''GUNAYDIN varsa doğrudur, fakat [GUNAYDIN] varsa yanlıştır.

## MEMBERP

Bir şeyin, ki bu bir sayı, sözcük ya da liste olabilir; başka bir listede de bulunup bulunmadığını denetler. Örneğin, MEMBERP 34 [12 23 24 34 45] doğrudur. MEMBERP 34 (12 [34 45] yanlıştır.

## WOROP, NUMBERP VE LISTP

Bu üç işlem, verilen bilginin sözcük, sayı veya liste biçiminde olup olmadığını denetler. Programda hata bulma aşaması olduğu için, prosedür içersinde pek önemli değillerdir. Daha önce belirtilmiş olduğu gibi, bir sözcük gibi işlem süren bir sayı, liste halinde olan bir sözcük kullanmak olasıdır. IF WORDP :YUCEL [PRINT [BU BIR SOZCUKTUR]] sorun ortaya çıktığında belirlemeyi yapacaktır.

## Sonuçlar

Bu kısımda LOGO'da kullanılan pek çok yeni yöntem ve komut tanıtıldı. Daha öncekiler gibi bunlar da, kişisel tercih ve isteklere bağlı olarak sonsuz sayıda farklı biçim almaya uygundur.

# Zeki Bilgisayar Yardımıyla Eğitim (2)

*— 100 terminale kadar varabilen geniş şebekeler,*
*— terminallere en son teknolojik yeniliklerin bağlanması,*
*— ders malzemelerinin kullanıcılar tarafından geliştirilebilmesidir.*

PLATO sisteminin (terminal sayısı arttıkça ekonomik verimlilik yükselir şeklindeki) felsefesinde, iktisadi yön (hemen hiçbir varlık göstermeyen) pedagolojik ilkelere ağır basmakta.

*PLATO zengin bir öğrenme ortamı oluşturmak gerektiği, iyi nitelikteki ders malzemelerinin varlığını koruyacağı ve sonunda yaygın kullanıma ulaşacağı gibi müphem fikirlere dayanır. Dolayısıyla PLATO hakkındaki değerlendirmeler ister istemez sübjektif oluyor: Sık sık bozulmasından şikayet etmekle birlikte, kimi durumlarda sisteme başvurmaktan hoşlandıklarını belirtiyorlar. Öğretmenler de bu konuda çelişik duygular içindeler - bazıları bu sistem aracılığıyla ders malzemesi geliştirmeyi zevkli bir uğraş olarak görüyor, fakat bir daha PLATO'yu kullanmak istemediğini bildiren öğretmenlerin oranı da yüzde onu aşıyor.*

*PLATO'nun NSF desteğine kavuşmasından bu yana donanım fiyatları düştü ve mikrobilgisayarlar okullarda yaygın bir kullanıma ulaştı. Dolayısıyla, başlangıçta bu sistemin gerekçesi olarak gösterilen ekonomik denklemin değişkenleri yeni değerler almış ve sistem mali açıdan çekiciliğini kaybetmiş bulunuyor.*

## Benzeşimler

*Bir sürecin bilgisayar benzeşimlerinin eğitimdeki kullanımı, öğrencilerin seyrederek (burada, ekranda gösterilen benzeşimlerin sonuçlarını seyrederek) öğrendikleri varsayımını temel alır. Gelişkin benzeşim ortamlarında, öğrenci süreç ilerledikçe değişkenlerin değerlerini değiştirerek veya başlangıç girdilerini seçerek aktif bir rol oynayabilmektedir. Bu da, öğrenci, örneğin biyolojik evrimi ya da bir bilgisayar program parçasının nasıl çalıştığını öğrenmekteyse son derece yararlı olmakta, ilk durumda bir evrim sürecinin sonuçları (mesela bir meyve sineğinin görünümünde gerçekleşen değişimler), ikincisinde de program değişkenlerinin değerleri gösterilebilmektedir.*

*Elbette, benzeşimlerin örneğin İngilizce kompozisyon gibi bir konuda hiçbir anlamı yok, fakat kullanılabildikleri konularda insana kendi fikirlerini araştırma olanağı tanıdıkları için öğrenciler tarafından bir hayli tutuluyorlar. Kısacası benzeşimler, ders konularının tek bir perspektifiyle sınırlı, ama gene de önemli bir rol oynuyorlar.*

## Oyunlar

*Bilgisayarlarda oyun kullanımı bir hayli yaygın. Gerçekten de oyunlar, son yıllarda İngiltere'de bu kadar çok mikrobilgisayar alınmış olmasının belki de başlıca sebebini oluşturmakta: Mikrobilgisayarlar genellikle alıcıyı teşvik etmek için eklenmiş bir oyun paketiyle birarada satışa sunuluyorlar. Varolan talebi karşılamak üzere küçük, fakat hızla büyüyen bir oyun-üretim endüstrisi doğmuş bulunuyor ve saygın bilgisayar dergileri, meraklıların kendi bilgisayarlarına aktarabilmesi için sayfalarca oyun program şifreleri yayınlıyorlar.*

*Oyun programlarının cazibesinin kaynağı ne, ve bu durum eğitim açısından ne anlama geliyor? Bilgisayar oyunları olgusu öylesine yaygın ki, bu konuyu önemsememek imkânsız.*

*O'Shea ve Self, Banet [4] ile Malone'un [5,6] araştırmalarına dayanarak bu cazibenin nedenlerinden bazılarını saptamışlar. Banet'in araştırmasına dayanan ilk nedenler dizisi bilgisayarların oyun ortamında kendini gösteren özellikleriyle ilgili:*

*— Oyun durumunun sunuluşunda ve başarının ödüllendirilmesinde görsel ve işitsel efektler kullanılır;*
*— Oyunun oyuncuyu zorlama derecesi yükselebilir - can sıkacak ölçüde basitleşmesi şart değildir;*
*— Oyun (bir uzay gemisini kullanmak gibi) hayali ögeler içerir;*
*— Bilgisayar, oyuncunun tepki gösterme süresini ölçüp skor hesaplayabilir.*

*Malone'un araştırmasındaysa çocukların 25 oyun arasından tercihleri incelenip, oyunların*

çekiciliğinin şu üç özelliğe bağlı olduğu tahminine varılmış:

— Zorluk: Ulaşılabileceği kesin olmayan bir hedef vardır;
— Merak: Oyuncu olacaklar hakkında bilgisine dayanan bazı beklentilere sahiptir, fakat bu beklentiler her zaman gerçekleşmeyebilir.
— Hayal: Oyun duyularla algılanmayan zihinsel imgeler uyandırır.

Oyunların eğitim açısından önemi, çocuğun dikkat ve ilgisini toplayabilmeleri, yani geleneksel eğitimde pek kolay görülmeyen bir durumu sağlayabilmeleridir. Buradan hareketle, oyunlar eğitsel açıdan yararlı kılınabilir mi sorusu doğuyor. Yani, normal koşullarda öğrenilerek kazanılan becerileri gerektiren ve bu becerileri oyun ortamında etkileşim yoluyla oyuncuya hissettirmeden veren bir oyun olabilir mi?

Bu sorunun cevabı henüz bilinmiyor, fakat oyuncunun mümkün olduğunca az ''GO'' komutu ile Tombstone'dan Redgulch'a gitmek zorunda olduğu ''Batı Nasıl Kazanıldı'' (Bitzer [7]) oyunu temel aritmetik beceriler gerektirmekte. Çocukların aritmetik becerileri yükseldikçe oyundaki başarıları da artıyor.

Çocuğun oyunun vermeye çalıştığı becerileri bu oyunu oynaması sonucunda kazanıp kazanmadığıysa pek kesin sayılmaz. Reswick'in [8] bulgularına göre çocuk, oyunu ne kadar çok oynarsa oynasın, genellikle becerilerinin tümünü kullanmasını gerektirmeyen sınırlı bir dizi stratejiyle yetiniyor. Bu sorun bugün söz konusu oyunu bilgisayarla eğitim ortamında kullanma çabasında olan Burton ve Brown [9] tarafından ele alınmakta. Henüz sonuçlar ortaya çıkmış değil; fakat oyun gerekli becerileri kazandıracak olsa bile -ki bu mümkün görünüyor- asıl sorun bu süreç içinde oyunun çekiciliğini kaybedip etmeyeceği.

## Problem Çözme

Öğrencinin bir bilgisayar programı yazmasını gerektiren ve bilgisayar programı oluşturma eyleminin genel problem çözme becerilerini geliştireceği inancına dayanan problem çözme yaklaşımı, en ateşli savunucusu olan Seymour Papert sayesinde bugün hem yaygın bir kullanıma, hem de en yetkin biçimi olan LOGO'ya (Papert ve ark. [10]) kavuşmuş durumda. Bu yaklaşım ''yaparak öğrenme''ye dayanan ve eğitimin ''açıklama''dan ziyade ''uğraşma'' süreci olarak görülmesi gerektiğini vurgulayan bir eğitim felsefesini temel alıyor.

LOGO'nun en önemli yönlerinden birisi, program yazacak öğrenciye son derece zengin araçlar sağlaması. Bu, yaklaşım açısından hayati bir önem taşıyor. Çünkü programlamacılık en iyi koşullar altında bile içkin güçlükler taşır ve yetersiz bir ortamda bu yaklaşımın hedeflerinin gerçekleşmesi, yani öğrencinin düşünme hakkındaki düşüncelerinin berraklaşması ve düşünme alışkanlıklarının gelişmesi beklenemez.

Peki, öğrenci bu hedeflere nasıl ulaşıyor? Bu soruyu cevaplamak pek kolay sayılmaz, hattâ bu hedeflere ulaşılabildiği bile kesin değil. Yalnız şunu söyleyebiliriz ki, bu yaklaşım öğrenciye kullanmaktan hoşlanacağı bir ortam sunmayı esas alıyor. Daha açık söylemek gerekirse, öğrencinin programını yazmakta kullandığı dilin güçlü, fakat öğrenilmesi ve kullanılması kolay bir dil olması gerekli; ortam ''dost'', yani örneğin öğrenciyi küçük hatalardan dolayı gereksiz yere cezalandırmayan ya da öğrencinin yapmak istediği şeylere keyfi sınırlamalar getirmeyen bir ortam ve ele alınabilecek bir dizi ilginç problem sunuluyor.

Programın gerçekleştirilmesine gereğince (örneğin renkli grafikler yoluyla) cevap verebilecek şekilde örgütlenmiş bir ortam sağlanması koşuluyla, küçük programlar bile öğrenci için çok aydınlatıcı olabilir. Öğrencinin geliştirdiği her bir programı kendi düşünsel süreçleri üzerinde bir deney olarak görebiliriz, programların gerçekleştirilmesiyle ortaya çıkan deneysel sonuçlarsa zengin bir geri-besleme kaynağı oluşturur. Eğer ortam özenle hazırlanmışsa, öğrencinin o ana kadar oluşturmuş olduğu yapı üzerinde giderek daha karmaşık programlar inşa etmesine olanak sağlayacaktır. Öğrenci kendisine en uygun gelen hızla ve istediği yönde ilerleme şansına sahiptir. Pratikte de öğrenci yalıtılmış durumda değildir ve programını başkalarına göstererek onların fikirlerinden de yeni şeyler öğrenebilmektedir.

Ne var ki bu yaklaşımın, en azından okullarda karşılaştığı bir sorun var: Ders programlarına kolayca yerleştirilememesi. Problem çözme veya düşünme hakkında, düşünme genel-

*likle kendi başına bir ders konusu olarak görülmüyor ve dolayısıyla örneğin LOGO faaliyetleri bilgisayar eğitimi başlığı altında ele alınıyor. Oysa problem çözme yaklaşımı bir öğrencinin problem çözme yeteneklerini geliştirse de, bu mutlaka programlama becerilerini de geliştireceği anlamına gelmez. Gerçekten de, çoğunlukla (belki de henüz çocuk yaşta olan) öğrenci Papert'ı çok mutlu edecek bir şekilde ''kafadan'' program yapacaktır. Fakat Dijkstra gibi geleneksel bir bilgisayar bilimcisi bunu nasıl karşılar? Bu bilim adamlarından birincisi bunu yaratıcı keşif olarak değerlendirirken, ikincisi belki de (öğrencilerin BASIC dilini öğrenmesi karşısında yapmış olduğu gibi) zihinsel sakatlanmadan söz edecektir.*

*Her ikisi de bir haklılık payı içeren bu farklı görüşleri uzlaştırmak için epey çaba gerekmektedir.*

## Kurtarıcı Tarzlar

*McDonald [11] bilgisayarın öğrenci adına örneğin bilgi toplamak, hesap yapmak gibi işler yapmasını kurtarıcı tarzda işlem görmek olarak adlandırıyor. Bu tarzda, bilgisayar öğrenciyi önündeki konu hakkındaki bilgilerini artırmayacak tipteki görevleri yapmaktan kurtarmakta.*

*Bilgisayardan bu şekilde yararlanmanın avantajı, öğrencinin ders konusu üzerinde dikkatini toplama olanağı bulması. Aşikâr bir dezavantajdan da söz edilemez (tabiî bilgisayarın makul bir süre içinde cevap vermesi, bozulmaması, kolay iletişim kurulabilmesi, sonuçlarının doğru olması, vb. koşullarıyla).*

## Diyalog Sistemleri

*İsminden de anlaşılacağı gibi, diyalog sistemleri yukarıda ele alınan yaklaşımlardan farklı olarak öğrenciyle sistem arasında bir diyalog oluşmasını hedefler. Bunları, yalnızca sistemin ''konuştuğu'' ve öğrencinin rolünün soruları cevaplamakla sınırlı kaldığı geleneksel CAI sistemlerinin kesin olarak tek yönlü iletişim sürecine bir tepki olarak görebiliriz. Tabii, bunlar da öğrencinin öğretmenine soru sorabildiği alışılmış özel öğretmenlik uygulamasını yansıtıyorlar. Özel öğretmenlik sisteminin en önemli üstünlükleri öğretmenin uygun soruları yöneltmek yolmuyla öğrencinin olası sorunlarını tahmin etmek zorunda kalmaması (bunları öğrenci kendisi de belirtebilir) ve öğrenciye ders hakkında kendi fikirlerini geliştirme olanağı vermesi (ki bu da, her öğretmenin kabul edeceği gibi, dersi anlamanın en iyi yollarından biridir).*

*Özel öğretmenliğin diğer bazı özellikleriyse doğal dilde (genellikle konuşma yoluyla) gerçekleştirilmesi, iki tarafın da (yani gerek öğrencinin, gerekse öğretmenin) inisiyatifi alabilmesi, soruların atlanabilmesi veya başka sorularla desteklenebilmesi, konunun değiştirilebilmesi ve genel bir yönelimi ya da amacı bulunmasıdır. Bir sistemin bütün bu özelliklerle donatılması bugün mümkün değil. Doğal dille anlaşma (sınırlı bazı söylem alanlarında bir oranda başarı kazanılmış olmasına karşın) YZ araştırmalarının çözümlenmemiş sorunlarından birini oluşturuyor ve bir araştırma konusu olarak konuşma işlemleme henüz bebeklik çağını yaşıyor. Gene de, araştırmacılar bu tür sistemlerde karşılaşılan teorik sorunları mümkün olduğunca incelemiş bulunuyorlar. Bu sistemlere ilişkin tipik sayılabilecek üç örnek ileride ele alınacak.*

## 3. Tartışma

*Şu ana kadar sözü edilen ilk on yaklaşım, genel çizgileriyle, geleneksel yazılım mühendisliği yöntemleriyle uygulanabilir: Çözümleri doğaları gereği algoritmiktir ve bir programın geçerlilikle kaldığı süre içinde, geleneksel bilgisayar dilleriyle geliştirilebilirler. Değişik bir yaklaşım gereği ancak geliştiriciler sistemlerini öğrencilerin ihtiyaçlarına karşı daha duyarlı kılmak istedikleri zaman ortaya çıkmaktadır.*

*PLATO gibi sistemlerin son derece sınırlı davranış kalıplarından fazla bir şeyler sunabilme yetenekleri temelde kısıtlıydı; bu da, yetkin eğitsel uygulamalara geçmek isteyen bir geliştiricinin PLATO'nun yazar dili kullanmasından kaynaklanan başa çıkılması çok güç bir dal patlamasıyla karşılaşmasından ileri geliyordu. Daha önce de belirttiğimiz gibi, PLATO teknolojik ilerleme dalgaları üzerinde karaya oturdu. (Kaderin garip bir cilvesi, aynı dalgalar ''zeki'' Bilgisayar Destekli Eğitim (BDE) araştırmalarını da geriletti: Okullarda henüz sınırlı bir güç taşıyan ve YZ sistemleri için yetersiz olan mikro-bilgisayarlar kullanıl-*

*makta ve şimdilik çabalar bilgisayarlaştırılmış eğitimin önemli sorunlarının incelenmesinden ziyade bu bilgisayarlara ders malzemesi yetiştirilmesi üzerinde yoğunlaşmakta.)*

*Öte yandan YZ programları algoritmik benzerlerinden temelde ayrılıyorlar. Bu ikincilerin prosedür kontrolüne ağırlık tanımasına karşılık, YZ programları daha ziyade işlenecek olan verilerin doğasıyla ilgileniyor. Verilerin (YZ araştırmacıları bunu bilgi diye adlandırmayı tercih ediyor) aktarmak istediği şeyi doğal bir şekilde ifade edecek ve aynı zamanda da işlem kolaylığı sağlayacak şekilde temsil edilebilme, sunulabilmesi konusu, YZ çalışmalarının odak noktalarından birini oluşturuyor.*

*Temsil edilmeye çalışılan bilgi türleri şunlar:*

*- Olgusal ifadeler;*
*- Metinler;*
*- Müphem ifadeler.*

*Bunların her biri eğitsel bir sistem açısından potansiyel bir değer taşımaktadır. Olgusal ifadeler konu hakkındaki bilgileri (örneğin, Peru'nun başkenti Lima'dır) ve öğrencinin bilgisini (örneğin, Jim Smith Lima'nın Peru'nun başkenti olduğunu biliyor) ifade etmekte kullanılabilir. Genellikle doğal dil biçiminde olan metin, sistemle öğrenci arasında diyaloğu mümkün kılmak için gereklidir. Müphem ifadeler eğitim sürecinin tümünde görülür. Bunlar bir öğrencinin bilgisine dair kanıyı temsil eder (ör., Jim Smith'in Lima'nın Peru'nun başkenti olduğunu bildiğinin kesinlik derecesi **p**) ve mesela tıp gibi konularda yargısal bilgiyi yansıtabilir (örneğin, enfeksiyon türü primer bakterimiyse, bu durum organizanın bakteroides olduğuna işaret etmektedir).*

*YZ araştırmacıları pek çok temsili şema üzerinde çalışmıştır. Bu çalışmaların değişik üslupları hakkında bir fikir verebilmek için, bunların bazılarından kısaca söz edelim.*

*Yaygın bir kullanımı bulunan **predicate** mantık gerek olgusal ifadeleri, gerekse bunlar arasındaki ilişkileri temsil edebilmektedir. **Predicate** mantığın başlıca avantajı uzun bir araştırma geçmişine sahip bulunması ve artık iyice öğrenilmiş olması, dezavantajı ise psikolojik geçerliliği bulunmadığının, yani insan bilgisinin temsil edilme biçimine tekabül etmediğinin düşünülmesidir.*

*Semantik şebekeler de BDE'de oldukça uzun bir kullanım tarihçesine sahiptir. Örneğin SCHOLAR sistemi (Carbonell [12]) Güney Amerika coğrafyasının temsilinde bu şebekelerden yararlanmıştır. Şebeke (düğüm halinde) kavramlarla (bunları birbirine bağlayan yaylar halinde) ilişkilerden oluşur. Dolayısıyla, örneğin Lima ve Peru nod, ''başkentidir'' ilişkisi ise bunların arasında bir yaydır.*

*Psikologlar tarafından **schemata** adı altında uzun bir zamandan beri kullanılagelen çerçeveler, örneğin restoranda bir yemek gibi stereotipik durumların temsilinde kullanılan bir yöntem oluştururlar. Bir çerçevede değişik bilgi türlerinin girdiği depolar vardır. Örneğin, bazı depolar sözkonusu durumda her zaman doğru olan bilgileri (ör. yemek masada yenir), (dolgu diye adlandırılan) bazıları da ör. hangi yemeğin yenileceği gibi) bir sorunun cevaplandırılmasını gerektiren bilgileri içerir. Çerçeveler bir öğrenme teorisinin oluşturulmasında rol oynamış oldukları için özel bir önem taşımaktadır. Bu teoriye göre, tecrübe tek tek çevçevelerdeki yuvaların doldurulmasına olanak sağlar ve mevcut çerçeveler yardımıyla da yeni çerçeveler formüle edilir. ''Dolayısıyla, öğrenme tecrübe sonucunda **schemata**'ların (çerçevelerin) depolanması ve uyarlanması şeklinde ele alınmalıdır'' [1].*

*YZ araştırmacılarının yeni yeni ilgi gösterdikleri bir konu da (kökenlerini gene psikolojiden alan) üretim sistemleridir. Bu sistemler, çeşitli değişikliklerle, uzman sistemlerin geliştirilmesinde kullanılmıştır. Üretim, iki yarısı bulunan bir kuraldır, örneğin IF A THEN B (A gerektirir B) gibi. Eğer A doğruysa, o zaman (bir tür eylem olan) B gerçekleşir. Uzman sistemlerdeki uzmanlık genellikle bu türden bir dizi kural ile bunlara ilişkin bir yorumlayıcı şeklinde temsil edilir. GUIDON [14] eğitim sisteminde kullanılmış olan MYCIN bunun başlıca örneklerinden birini oluşturur.*

*YZ ile algoritmik programlar arasındaki bir diğer önemli fark da kontrolün uygulanış biçiminde görülür. İkinciler kesinlikle lineer bir dizisel tarzda çalışırlar. YZ'ın zengin bilgi yapılarının en iyi biçimde kullanılabilmesi içinse daha yetkin kontrol yöntemleri gereklidir. Dolayısıyla, YZ programları paralel kontrol ve bilgi durumuna bağlı kontrolün önemini ortaya çıkarmış ve belirsiz ya da yetersiz bilgi duru-*

munda karar verme mekanizmalarına başvurmuşlardır.

Eğitsel etkileşimin doğasını muhtemel olay dizilerine ilişkin sabit ve ister istemez sınırlı bir bakıştan ziyade, öğretmenin herhangi bir anda elinde bulunan bilgiler açısından incelemek daha uygun olduğundan, bu tür kontrol yöntemleri eğitsel sistem geliştiricileri tarafından ilgiyle karşılanmakta. Nitekim, GUIDON'da, çeşitli koşullar altında ne olacağını belirleyen söylem yönetimi ve eğitsel eylem kuralları bulunur.

## 4. En Yeni Gelişmeler

Eğitsel bir sistemde nelerin bulunup nelerin bulunmaması gerektiği konusunda büyük bir kafa karışıklığının hüküm sürdüğü bir dönemde, Self [15] en önemli üç bileşeni net bir şekilde ortaya koymuştu:

Bir öğretme programının genel performansını belirleyen bilgi üç başlık altında toplanabilir:

(a) Nasıl öretileceği ve (genelde öğrenciler) hakkında bilgi,
(b) Öğretilen konu hakkında bilgi,
(c) Öğretilen kişi hakkında bilgi (yani özel olarak tek bir öğrenci hakkında bilgi).

O günden bu yana, her bir alanda nelerin bilgi sayılacağı ve bir öğretme programının genel performansını daha yetkin kılmak için bu bilginin nasıl temsil edilmesi gerektiği konusunda tartışmalar süregelmiştir. Özellikle, öğrenci modeli diye adlandırılan yukarıdaki (c)) maddesi üzerinde durulmaktadır ve bunun ele alınmasında birbirinden farklı üç felsefe sözkonusudur:

- Uzman modeli;
- Bilgi işlem modeli;
- Genetik model.

Bu üç sisteme, tartışmaya bulunmuş oldukları katkı açısından bir göz atalım.

## GUIDON

Uzman sistemlerin bugünlerde topladıkları ilgi, GUIDON'u [14] incelemeniz için doğal bir seçim haline getirmekte. MYCIN uzman sisteminden yararlanan GUIDON bulaşıcı hastalıklar konusunda eğitim vermektedir. GUIDON'un amacı, MYCIN bilgi üssündeki uzmanlığı GUIDON'la etkileşim içinde bulunan öğrencilere aktarmaktır. İnceleme için GUIDON'u seçmemizin bir diğer önemli nedeni de, MYCIN bilgi üssünün bulaşıcı hastalık teşhisinde bir problem çözücü olarak üstün niteliğinin yaygın bir şekilde kabul edilmiş olmasıdır. Bu alanda gerçekten de uzman olan MYCIN, bunun yanısıra, kendi davranışlarını açıklama yeteneğine de sahiptir ve örneğin, belirli klinik bilgileri niçin istediğini ya da niçin şu veya bu hipotez üzerinde durmakta olduğunu açıklayabilmektedir.

GUIDON'ın kullandığı ''vaka incelemesi yöntemi''nde önce MYCIN alanda bir problem çözer, GUIDON da onun kullandığı problem çözme stratejisini inceledikten sonra, aynı vakayı çözmesi istenen bir öğrencinin kaydettiği gelişmeyi denetleyebilir. Yani öğrencinin (i) vakayla ilgili gerekli verileri isteyip istemediğini (ve ii) doğru sonuçlara ulaşıp ulaşmadığını saptar. Bu da öğrencinin değişik açılardan değerlendirilmesini sağlar. Öğrenmeninse bu etkileşim içinde gerçekleşeceği düşünülmektedir.

Ne var ki, ilk tecrübeler sistemin bazı yetersizliklerini ortaya koymuştur [18]:

> Uzmanların bilgisini GUIDON'la ''geri aktarma''yı denediğimizde, uzmanların teşhise yaklaşımının ve kurallar hakkındaki anlayışının açık bir şekilde temsil edilmemiş olduğunu görüyoruz. MYCIN bir kuraldaki kavramların nasıl bütünleştiğine dair bir kodlama içermediği için, GUIDON kuralların doğruluğunu ispatlayamıyor. Araştırma boşluğununyapısı ve bunu aşma stratejisi kural kavramlarının sıralanışında içkin olduğundan, GUIDON MYCIN'in problem çözme yaklaşımını tam olarak ifade edemiyor. Bu nedenle, bilgiye dayanan bir sistemi bilgisayar yardımıyla eğitim programına dönüştürmek gibi yalım olacağı beklenen bir iş, kural temelinin ve bu kuralların üzerinde yükseldiği bazın ayrıntılı bir şekilde yeniden incelenmesine dönüşmüş bulunuyor...

(sürecek)

*Burada en önemli sorun, MYCIN'in bilgi üssünün temsil edilmesinde kullanılan formalizmden kaynaklanmakta, MYCIN'in yapımında elde edilen başarının başlıca faktörlerinden biri olan üretim kurallarına uygun tekbiçimli temsil, eğitimde tuhaf sonuçlara yol açabilmektedir. Sorun yaratan alanlar Clancey tarafından ele alınmıştır.*

*Bu sorunlardan bir kısmını değerlendirmeden önce, MYCIN'in nasıl akıl yürüttüğünü (kurallarını nasıl yorumladığını) anlamalıyız. Her kural bir veya daha çok hipotezle bir veya daha çok hedeften oluşur. MYCIN geriye doğru zincirleme bir strateji izler, yani hedefini seçtikten sonra bu hedefe varan hipotezlerin uygunluğunu inceler. Fakat tek bir kurala dayanarak derinlemesine bir araştırma yapmaz ve hedefe yönelen tüm kuralların gözönüne alınmasını gerektiren daha genel bir strateji izler. Bu akıl yürütme yöntemini kolaylaştırmak için, ortak bir hipotezi bulunan kurallar ve ortak ber hedefi bulunan kurallar arasında bağlantılar kurulmuştur.*

*Bu şekilde birbirine bağlanacak kuralların belirlenmesi kural-yazarının seçimine bağlıdır, kurulacak bağlantıları onun düşünce tarzı belirler. Bu tür bağlantıların gerekçesini kural doğrulamaları oluşturur. MYCIN'de dört tip kural doğrulamaları görüyoruz:*

*— Tespit*
*— Nedensei*
*— Gözlem olgusu*
*— Alan olgusu*

*Tanım doğrulamaları bir nesnenin özelliklerini temel alır. Örneğin ''eğer organizma gram-negatif anaerobik çubuk ise cinsi bakteroid olabilir'' kuralı, bir bilinmeyen hakkındaki laboratuvar gözlemlerine dayanmaktadır. Yani doğrulanması bir organizmanın özellikleriyle ilgilidir.*

*Çevre verisiyle doğrulanan kurallara bir örnek: ''Eğer hasta erkekse hamile ya da emzirme durumunda olamaz.''*

*Alan verisiyle doğrulanan kurallara bir örnekse, ''eğer ilaç ağızdan verilmiş ve gastroentestinal yolda az adzorbe olmuşsa, yetersiz miktarda verilmiştir'' olabilir. Tanım gereği, bir ilacın yeterli miktarda verilmiş sayılması için vücutta yeterince yüksek bir düzeyde bulunması gereklidir.*

*Görüldüğü gibi, bu tür doğrulamalarda, en azından yukarıdaki örneklerde, kuralın kendisi eğitsel açıdan yeterince açıktır. Fakat başka alanlarda bu durum farklı olabilir. WHY (Niçin?) sisteminin kullanımında, geliştiriciler öğrencilerin yanlış fikirlerinden bazılarının alandan değil, gerçek-dünyaya ilişkin bilgi modellerinin yanlış olmasından kaynaklandığını görmüşlerdi. Bir doğrulama gerçek-dünya bilgisini gerektiriyorsa, karmaşıklık düzeyi öğrencinin bu doğrulamayı kabul edip kendi (belki de yanlış) modelini ona göre düzeltmesini sağlayacak ölçüde yükseltilmelidir.*

*MYCIN'de nedensel doğrulamalar daha da karmaşık bir konudur. Örneğin ''eğer hasta 8 yaşından küçükse ona tetrasiklin vermemek gerekir'' kuralı, altında yatan nedensel süreçten (ilacın gelişmekte olan kemiklerde birikmesi) ve toplumsal sorunlardan (kalıcı dişlerin kararması) hiç bahsetmez. ''Uzman işini yaparken doğrulamayı düşünmek zorunda olmadığına göre öğrencinin de bilmesi gerekmez'' savıyla öğrenme süreci gözardı edilmektedir. Bu sava cevap olarak, uzman kuralı doğrulama aracılığıyla öğrenmiş olduğuna göre aynı şeyin öğrenci için de geçerli olacağı söylenebilir.*

*Bir de MYCIN'in kurallarında içkin bulunan stratejik enformasyona bakalım. Bir kuraldaki hipotezlerin sırası, hedeflere hangi sırayla yönelindiğini etkiler. Ancak bu sıranın gerekçesi kural ağının dışında kaldığı için, program hedeflerin niçin belirli bir sıra içinde ele alındığını açıklayamamaktadır. Ayrıca, bir hedef için denenen kuralların sırası da, hipotezlerin ve dolayısıyla alt-hedeflerin nasıl bir sırayla ele alınacağını etkiler. Kullanıcının soruları bu hedeflere bağlı olduğuna göre, kuralların sırası soruların dizilişini de belirler. Dolayısıyla, program soruların niçin belli bir diziliş içinde sorulduğunu açıklayamamaktadır. Tabii, burada cevap olarak kural sırası gösterilebilir, fakat bunun tatmin edici bir cevap olmadığı açıktır.*

*Pratikte, MYCIN'de kural sırası nihai teşhisi etkileyebilecek ölçüde rol oynamıyor. Bu sıra, kuralların sisteme verilme sırası tarafından belirlenmiş, sabit bir sıra. Yani, MYCIN'in nasıl bir sırayla hedeflere yöneldiğini ve sorular sorduğunu belirleyen akılcı bir gerekçe sözkonusu değil. Fakat, bu konunun öğrenciler açısından önemsiz olduğu söylenemez. Bir hi-*

potez üzerinde odaklanıp uygun soruları seçmek, insanın akıl yürütme tarzının tipik özelliğidir. MYCIN'in bu açık stratejik bilgiye gerek göstermemesi, kuralları formüle eden uzmanın ya da bu kuralları öğrenmeye çalışan öğrencinin değil, sistemin çıkarsama mekanizmasından ileri gelmektedir.

İlke olarak, MYCIN'in açık stratejik enformasyon içermemesi için hiçbir sebep yok; hatta böyle bir enformasyon normal görevlerindeki becerisini de yükseltebilir. Ne var ki, üretim sistemlerinde stratejik kontrol için alanın yapısal tanımı gereklidir. Alanın temel bir yapısı bulunduğu, daha önce sözünü ettiğimiz doğrulama bağları tarafından da kanıtlanmaktadır. Burada, GUIDON'un sorunu yapının ''tutamak''larının bulunmaması ve bu nedenle açıklama yeteneğinin hedefleri ve kural hipotezlerini düz bir şekilde tartışmakla sınırlı kalmasıdır.

Neyse ki, MYCIN'in kural ağaçları geniş ve sığ olduğundan, GUIDON bir hayli tatmin edici açıklamalar sunabilmektedir. Fakat ağaçların dar ve derin olduğu SACON yapısal analiz alanına uygulandığında, ''ders programı dejenere olup, ağacın dallarından birinin uzun bir tanımına dönüşmüştür''.

MYCIN akıl yürütme adımlarını yalnızca basit bir şekilde belirten, kesinlikle ayrıntılı bir şekilde açıklayamayan düz bir kurallar sistemidir. Kuşkusuz, bir kuralın nasıl oluşturulduğu temel önem taşır: MYCIN'in kuralları anlamayı değil, beceriyi geliştirmek üzere oluşturulmuştur. Clancey'e göre:

> Bu çalışmadan, temsil sistemi olarak türetme kurallarından yararlanmayanlar da dahil olmak üzere, bilgi-tabanı programları için de geçerli bazı şeyler öğrendik. Birincisi, insan yalnızca iyi bir YZ programına karşılıklı-etkileşimli bir takarak yeterli bir öğretim sistemi elde edemez. İkincisi: Bir açıklama sisteminin akılyürütme adımlarını tekrarlamak ve soruları tanımaktan daha fazla şeyler yapabilmesi gerekir.

## SPADE

SPADE bilgisayar programlarının yazmak için yapılandırılmış planlar hazırlamayı ve bu planları hatalardan arındırmayı öğretmek üzere hazırlanmış, henüz deneysel safhadaki bir sistemdir. Bu sistem, programın belli bir bilgisayar dilinde yazılması aşamasındansa, planlamaya ve planın öngördüğü tasarımının hatalardan arındırılmasına daha büyük bir önem atfeden bir programlama anlayışının ürünüdür.

Dolayısıyla SPADE tasarlama sürecinin açık bir modelini içerir. Bu model Papert'in Kaplumbağa LOGO'su dilinde çalışma yapan çocukların protokollerinden türetilmiştir. Burada sözkonusu olanı, belki de ''model'' teriminden ziyade, ''modeller üreten bir çerçeve'' anlatımı daha iyi tarif ediyor. Örneğin bu çerçeve ideal öğrencinin bir modelini içeriyor; ayrıca kendisine kaynaklık eden öğrenci protokollerinin de modellerini içeriyor; gücül olarak öğrencilerin etkileşimlerinin de modellerini içerebilir. Bu son türden modeller, çerçevede halen varolan modeller ''çarpıtılarak'' geliştirilir.

SPADE bir bakıma kendi içerdiği çerçeveyi sınamaya yarayan bir deneysel araç olarak da

tarif edilebilir. Bu sınama, öğrencilerin etkileşim sırasında sözkonusu planlama ve tasarlama sürecini ne kadar başarılı bir biçimde kullandıkları gözden geçirilerek gerçekleştirilir. Bu gözden geçirmeye ilişkin bazı sorular şunlar olabilir: ''Kullanılan kavramlar hangi ölçüde yeterli?'', ''Sistem öğrencilerin kendisiyle ne türden etkileşimlere gireceğini önceden kestirebiliyor mu?''

Geliştiricisi SPADE'i ''sınırlı bir didaktik sistem'' diye tarif ediyor ve tam teşekküllü bir öğretim sisteminin oluşturulmasında bir ara aşama olarak yorumluyor. Ancak programlama öğreten öğretim sistemlerinin geliştirilmesinin önünde duran bir engel, programlamanın çok iyi anlaşılan bir faaliyet olmayışıdır. SPADE bu faaliyetin bazı yönlerini somutlamamızı sağlıyor. Öğretilecek alan hakkında iyi bir kavrayışın yanında, tam teşekküllü bir öğretim sisteminin hangi bilgilerin ne zaman ve nasıl iletilmesi gerektiğini de bilmesi gerekir. Tasarlama süreci hakkında açık bir model içeren SPADE, bu soruların da yanıtlanmasını mümkün kılıyor.

SPADE'in dayandığı programlama anlayışında, planın bir bilgisayar dili halinde kodlanmasının önemli görülmediğini söylemiştik. Dolayısıyla modeldeki hiyerarşinin üst düzeylerindeki kurallar planlama ve planların hatalardan arındırılması ile ilgili; ancak görece düşük düzeyli kurallar belli dillerin belli yapılarını hesaba katıyor.

Model bir dizi prosedürden oluşuyor. Prosedürlerin her biri ise, kendi çeviricileri tarafından harekete geçirilen bir kurallar paketinden ibaret. Hiyerarşinin en üst düzeyinde ise, ''geliştir'', ''tamir et'', ''dene'' gibi adları olan kontrol prosedürleri var. Bazı kontrol kurallarına örnek olarak şunlar sayılabilir:

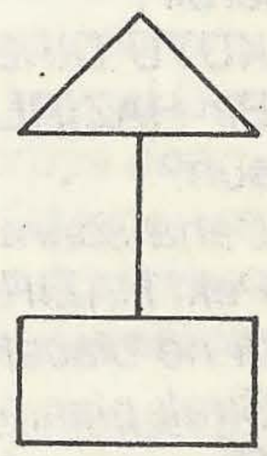
1) IF ''dene''deysen ve bir hata varsa THEN hatanın yerini ve türünü bildir.
2) IF ''geliştir''deysen ve bir alt-prosedürün tanımlanması tamamlanmışsa, THEN o alt-prosedür için ''dene''yi başlat.

Model belli problemlerle sınırlı çözümler ürettiğinden, her problemin özgül tanımını gerektiriyor. Çözüm ancak tanımda kullanılan terimleri kullanabilir. Bu ise SPADE'ın uygulanabilirliğini, çözümü problemin tanımından belli, basit problemlerle sınırlıyor.

Ama bizim asıl ilgilendiğimiz konu SPADE'in Kaplumbağa LOGO alanında nasıl çalıştığını göstermek. Bu SPADE için ideal bir alandır, çünkü karmaşık olmayan bir dizi ilginç problemler içerebilir. LOGO'yla çalışmış olan öğrencilerin, bu çalışma seanslarını yansıtan protokollerinin bulunması, SPADE'i kullanmayı ve içerdiği modeli LOGO'ya göre geliştirmeyi mümkün kılmaktadır.

SPADE'in bu kullanımına bir örnek olarak, bir örnek olarak, bir kuyu çizimi problemini vereceğiz. Kuyuyu çizdiren program aşağıda verilmiştir. Programda kullanılan KUYU ve CATI komutları, kullanıcının kendisinin tanımlayacağı prosedürlere işaret etmektedir:

```
TO DILEK KUYUSU
10 RIGHT 90
20 KUYU
30 FORWARD 50
40 LEFT 90
50 FORWARD 100
60 LEFT 90
70 FORWARD 50
80 RIGHT 120
90 CATI
END
```



Bir öğrenci bu problemin çözümünü öğrenmek için nasıl bir yol izleyecektir? Bunu görmek için bir dizi etkileşimi gözden geçirelim:

S1: Bugün tanımlayacağın prosedürü nasıl adlandırmak istiyorsun?
C1: Dilek Kuyusu
S2: Ne tür bir plan kullanacaksın?
(A) Problemi daha önce çözülmüş problemlere benzer olarak tanımlayan bir plan
(B) Problemi alt problemlere çözümleyen bir plan
(C) Problemi alternatif bir şekilde formüle eden bir plan
C2: Sıralı.

SPADE'in henüz üzerinde durmadığımız bir yönü, problem tanımının nasıl girileceği. Bunun için sistem, prosedürün adı verildikten sonra tarifini ister. Tarif yalın yüklemler olarak verilmesi gerekir. Örneğin bu örnekte öğrencinin Dilek Kuyusunun CATI, DIREK ve KUYU'dan oluştuğunu; catı'nın bir UCGEN olduğunu; DIREK'in bir DOGRU olduğunu; KUYU'nun bir KARE olduğunu, CATI'nın DIREK'İN ÜSTÜNDE (ABOVE) olduğunu bildirmesi gerekir. SPADE yüklemler arasındaki

içerme ilişkilerine ve UCGEN, KARE vs. gibi genel kullanıma açık bir dizi objeye sahiptir. Bu, sisteme öğrencinin planını değerlendirme ve planın adımlarını kurma imkânını verir.

SPADE'in ayrıca hiyerarşik bir planlar sınıflandırması vardır. En tepede üç tür plan bulunur: Tanımlama, çözümleme, yeniden formülasyon. Ancak örneğimizde soruyu cevaplandıran öğrenci, sistemin menü aracılığıyla verdiği seçenekleri dikkate almadı. Ancak aşağıda da görüldüğü gibi SPADE, öğrencinin bu tarz önceden kestirilemeyen seçimleri ile ilgili olarak da bazı varsayımlarda bulunabilir:

S3: PLAN-COZUMLE-SIRALI istediğini varsayıyorum. Bu sıralı planan ana adımları nelerdir?
C3: KUYU DIREK CATI
S4: Bir HAZIRLA adımı eklemek istiyor musun?
C4: Daha sonra.
S5: Peki HAZIRLA adımını erteliyorum. Şimdi ne olacak?
A5: Sıralı planın ikinci arabirimine git.
S6: Bu seçime bağlı arabirimi de dahil etmek istiyor musun?
A6: Evet.

Burada SPADE DILEK KUYUSU'nun sıralı bir plan şeklinde çözümleneceğini varsaymaktadır. S4 ve S6 SPADE'in planlama süreci hakkındaki bilgisini örneklemektedir. Bu bilgi sıralı planların genellikle bir ilkdeğerleme ya da HAZIRLA adımları olduğunu (örneğin kaplumbağanın istenen yöne bakmasını sağlamak için) ve sıralı birimlerin çeşitli aşamaları arasındaki arabirimlerin seçime bağlı (ihtiyari) olduğunu içermektedir.

S4, aynı zamanda yukarıdan aşağı ve soldan sağa bir planlama stratejisinin izlenmekte olduğunu gösteriyor; ama bu örnekte öğrenci sık sık yapıldığı gibi, hazırlama adımını ertelemeyi tercih etmiştir.

Böyle bir söyleşinin yararı, modeldeki eksiklikleri hemen açığa çıkarabilmesindendir. Böylesi söyleşilerin çözümlemesinden hareketle Miller ''asgari-kapsam'' ilkesini geliştirmiştir: **''Planlama kararları, ileride yapılacak değişikliklerin kapsamını asgariye indirecek bir sıra içerisinde alınmalıdır.''** Aşağıdan yukarı planlama ilkesi, bir ölçüde bu ilkeye yaklaşmaktadır, ama ancak bir ölçüde. Örneğin bu strateji yukarıdaki örnekte, HAZIRLA adımının KUYU'dan önce tasarlanmasını önerdi. Oysa öğrenci HAZIRLA adımını sona ertelemeyi tercih etmiştir. Bu durumda, öğrenci Miller'ın ''asgari-kapsam'' ilkesini uyguluyor olabilir. Çünkü öğrenci önce HAZIRLA'yı, sonra KUYU'yu tasarlasa ve sonra da KUYU'nun bir HAZIRLA adımı değil de bir arabirim gerektirdiğini farketseydi, o zaman HAZIRLA adımı fazladan olacaktı.

SPADE'in hedefi programlama sırasında, özellik yapısal planlamaya ve hatalardan arındırmaya ağırlık vererek öğrenciye kılavuzluk etmek. Ama bugüne kadarki çalışmalar, önce daha az iddialı hedeflere ulaşılması gerektiğini göstermektedir. Bu hedefler şu soruların yanıtlanmasını içermektedir:

— Tasarlama sürecine ilişkin olan kavramlar hangileridir?
— Bu kavramlar Kaplumbağa LOGO'su dışındaki alanlar için de yeterli olacak mıdır?
— Bu kavramların dile getirilmesi öğrenme sürecine yardımcı olmakta mıdır?
— Sistem öğrenciye hangi ölçüde yardımcı olmaktadır?

## WUSOR

WUSOR, ya da tam adıyla WUSOR-1, WUMPUS oyunu için geliştirilmiş ve uzman bilgisine dayalı bir ''antrenör'' sistemdir. Ancak biz burada doğrudan doğruya WUSOR'un kendisiyle değil, öğrencilerin modellerinin nasıl üretilebileceği ile ilgileneceğiz. Bu alanda WUSOR'un sınırlılıklarından hareketle değerli bazı fikirler üretilmiştir. Biz özellikle Goldstein'ın hazırlamış olduğu gelişme grafikleri üzerinde duracağız.

*GUIDON'un öğrenci modeli, Goldstein'ın örtüşme modeli diye adlandardığı bir modeldir. Bu modelde öğrencinin bilgisi, edinmesi gereken uzmanlık alanının toplam bilgisinin bir alt-kümesi olarak kavramsallaştırılır. Örneğin GUIDON'da öğrenci modeli öğrencinin MYCIN'in uzmanlık alanından öğrencinin edinmiş olduğu düşünülen bilgilerden oluşur. Bu tür sistemlerde öğrenme de, bu alt-kümenin sınırlarının büyümesi olarak kavramsallaştırılır. Bu, doğru cevaba ulaşılmasının önündeki kritik engelin bir kural ya da olgunun eksikliği olduğu durumlarda sistemin müdahaleleri ile sağlanır.*

*Goldsetin bu öğrenme modelinin aşırı derecede basit olduğunu ve yeni bilgilerin nasıl eski bilgilerden hareketle evrildiğini hesaba katmadığını iddia ediyor. Bu yüzden de daha yetkin bir temsil sistemi olarak gelişme grafiklerini öneriyor. Bilginin gelişmesine ağırlık veren sistemler uzman-merkezli olmaktan çok öğrenen-merkezli olarak tarif edilebilir.*

*Bir gelişme grafiğinde düğümler bazı becerileri, düğümleri birbirine bağlayan yaylar ise becerilerin nasıl birbirlerinden evrildiğini temsil eder. Aradaki bağların temsil ettiği şeyler, örneğin şunlar olabilir: Analoji, uzmanlaşma, genelleme, ön-koşul, sapma. Bu sistem aynı zamanda, düğümlerin kuralları, yayların ise aralarındaki karşılıklı ilişkileri temsil ettiği şeklinde de düşünülebilir.*

*Bu yaylar içinde en ilginç olanları sapmaları temsil edenlerdir, çünkü bunlar öğrencinin yanlış anlamalarını da temsil etme olanağını verirler. Yanlış kurallar, doğru kurallardan sapmalar olarak gösterilir. Dolayısıyla yanlış kuralların doğru uygulanmasından kaynaklanan hatalar yakalanabilir. Eğer hatalar hakkında bir ön çalışma ve çözümleme yapılırsa, bu sistemin doğru kuralların yanlış uygulanmasından kaynaklanan hataları yakalayabilmesinin önünde de ilkece bir engel yoktur.*

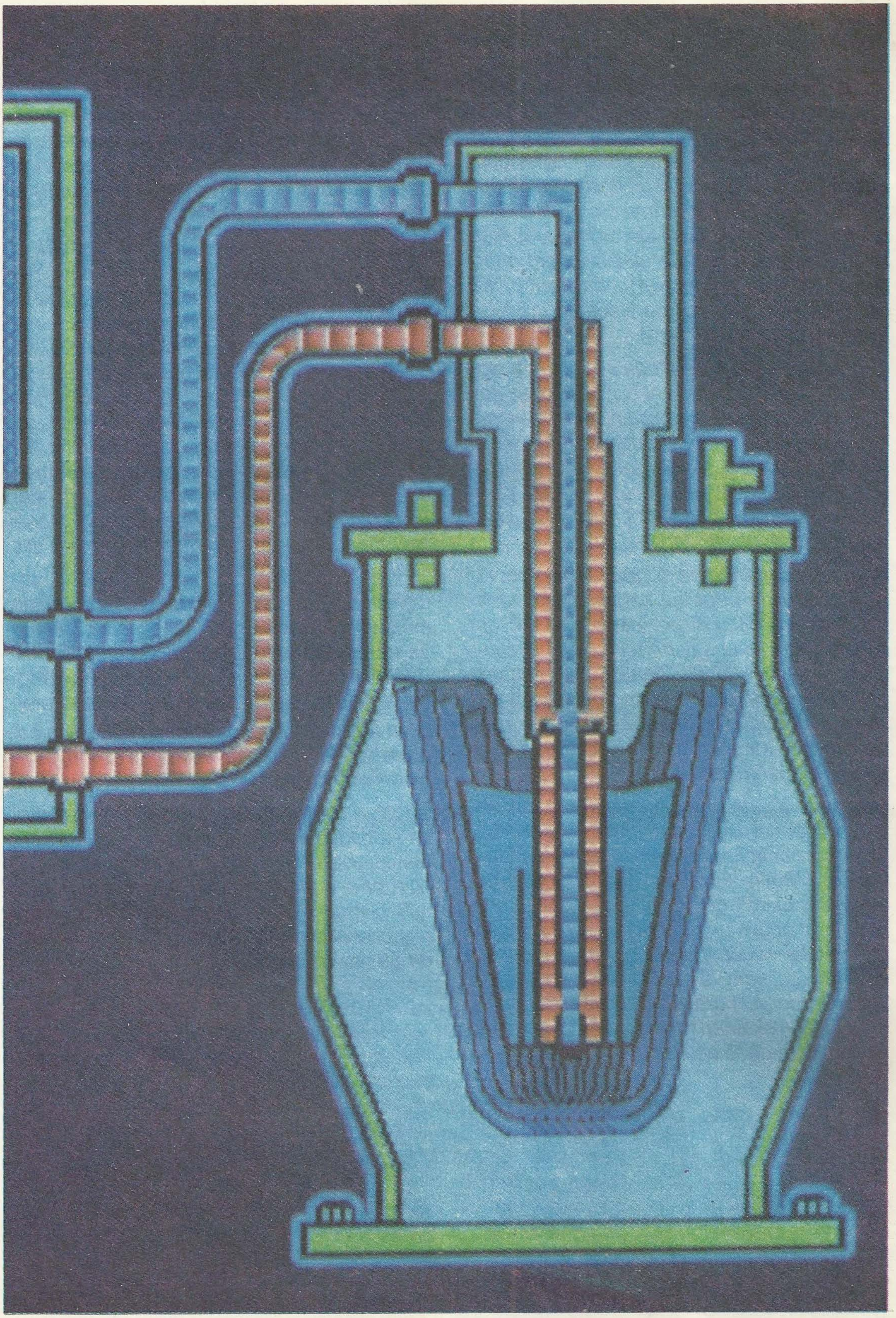
*Her düğümün birden fazla yayı olabileceğinden (aynı bilgi birden fazla şekilde edinilebilir), grafik tek bir bilgi gelişme çizgisi tarif etmez. Bu hem her öğrencinin kendine özgü öğrenme tarzının temsil edilebilmesine, hem her duruma göre farklı öğretmen müdahalelerine olanak tanır. Böylelikle öğrenci önceden tasarlanmış bir öğretim tekniği ile kısıtlanmış olmuyor.*

*Zeki eğitim tekniklerinin hedeflerinden biri, bireyselleştirilmiş öğretimdir. Hernekadar bireyselleştirilmiş eğitim, öğrencilerin öğrenme alışkanlıklarının ve tercihlerinin hesaba katılmasından elbette olumlu yönde etkilenirse de, geleneksel eğitim teknikleri bu bilgilerle ne yapabileceklerini kestirememişlerdir. Eğer öğrenci verilen malzemeyi öğrenememişse, bu, malzemenin kötü verilmiş olmasından mı, öğrencideki bir yetersizlikten mi kaynaklanmaktadır? Gelişme grafikler bu soruya doğrudan bir karşılık getirmektedirler: Öğrenci modeli grafiğindeki yaylar öğrencinin öğrenme alışkanlıklarını ve tercihlerini simgelemektedir.*

*Öğretmenlerin karşılaştığı bir başka sorun da, mümkün açıklamalardan hangisinin öğrenmeye en çok katkısı olacağını kestirebilmektir. Öğrencinin bilgilerinin bir modeli bulunduğu takdirde, öğretmen belli bir açıklama stratejisinin tutarlı bir biçimde istenen becerinin edinilmesini sağlayıp sağlamadığını belirleyebilecektir.*

*Gelişme grafiğinin bir sınırlılığı ise yapılandırılmış bilgi gerektiren alanlarla başa çıkamamasıdır. Aynı eksiklik GUIDON için de geçerlidir, ama daha önce gördüğümüz gibi, stratejik bilginin uygulanabilmesi için yapılandırılmış bilgi elzem olduğundan, bu eksiklik aslında hayati bir eksikliktir.*

L. Ford



1950

# Bilgisayarlaşmanın İktisadi Sistem Üzerindeki Etkileri

Beşinci Kuşak bilgisayarlarla ilgili çalışmaları anlatırken, bilgisayar teknolojisi ve sanayiindeki olası yeni gelişmeleri gözden geçirmiş, toplumun bilgisayarlaşmasının gündelik hayat ve özellikle de çalışma hayatı üzerindeki etkilerinden bazılarına zaten değinmiştik. Örneğin, VLSI teknolojisindeki gelişmelere bağlı olarak Üçüncü Dünya ülkelerinin en önemli kaynaklarından biri olan ucuz işgücüne duyulan uluslararası talebin azalmakta olduğunu gördük. Özellikle gelişmiş ülkelerde otomasyon ve robotlar sayesinde elde edilen üretkenik artışıyla birlikte ele alındığında, bunun gelişmiş ve azgelişmiş ülkeler arasındaki gelişmişlik farkını, hiç değilse orta vadede, daha da büyüteceği açıktır. Çok yüksek düzeyde tümleştirmenin, Üçüncü Dünya ülkeleri üzerindeki ikinci bir olumsuz etkisi ise, özellikle bu ülkelerin uzmanlaşmış oldukları montaj sanayilerinin önemini azaltmak.

Mikro-elektronik alanındaki gelişmelerin bu kadar merkezi bir önem kazanması, sadece bu teknolojilerin devrimci niteliğinden ötürü değildi. Bu alandaki teknolojik sıçrama, dünya iktisadi sisteminin tarihinin üçüncü büyük krizine girdiği bir sırada gerçekleştirildi. Bu, çeşitli iktisatçı, politikacı ve yazarları mikro-elektronik sanayiine krizin aşılmasında merkezi bir rol atfetmeye yöneltti.

Ancak bu sırada kimi uzman ve yazarlar mikro-elektronik teknolojinin böyle bir etkisi olamayacağını, aksine sözkonusu teknolojinin sermayenin merkezileşmesinin berisinde yatan iktisadi rasyoneli aşındırarak, bugün varolan kapitalist dünya sisteminin varlık nedenini ortadan kaldırdığını öne sürüyorlar. Diğer yandan başka bazı yazarlar ise, 1980'den beri krizin aşılması yönünde önemli gelişmeler kaydedildiğini, dünya iktisadi sisteminin mikro-elektronik alanındaki gelişmelerle tutarlı yeni bir sermaye birikim modelini yapısallaştırmak üzere olduğunu iddia ediyorlar. Bu ikinci gruptaki yazarların iddiası mikro-elektronik teknolojinin ürünlerinin yaygınlaşmasının gündelik hayatı köklü bir biçimde değiştirmeyeceği değil. Ancak iktisadi sistemin temel parametrelerinin -toplumun varlığını idame ettirebilmesi için gerekli çalışmanın ücretli emek biçimini alması, ücretli işçilerin üretim araçları ve süreci üzerindeki mülkiyet ve denetim haklarından yoksun olması, sermayenin merkezileşmesi, toplumsal ürünlerin (ve hatta giderek her şeyin) meta (mal) biçimini alması- değişmediğine işaret ediyorlar.

# Enformatik Sanayiinin Emek Süreci Üzerindeki Etkileri

## Bandta Çalışmanın Sınırları

1970'lerin başında, özellikle Fiat, Volvo gibi otomobil fabrikalarında yeni bir duruma tanık olundu. Bu fabrikaların bazı atölyeleri bandta çalışmayı terkedip, işi yeniden ekipler halinde örgütlemeye başladılar. Aslında çoğu defa band (yani konveyörler) tamamen terkedilmiyordu. Ekipler kendi alet ve gereç stoklarıyla çalışıyorlardı. Ancak araç, gereç ve hammadde donatımı için yine bir konvoyöre bağlı oluyorlardı. Konveyörün sürati işin ritmini tespit etmiş oluyordu.

''İşe duyulan ilgiyi yeniden canlandırma'' hedefi olarak gerekçelenen bu atılımla bandta ve özel becerilerine başvurulmadan çalıştırılan işçiler bandtan çıkarılıp yeniden -1920'lere kadar geçerlikte olan teknik işbölümünü andırırcasına- ekipler halinde ve mesleki uzmanlıklarına da gerek duyulduğu intibaı verilerek çalıştırılmaya başlandılar. Bu ekipler üretimin bütün bir safhasından sorumlu oluyorlar, maddeye uyguladıkları değişimin, faaliyetin anlamını üretimin genel kapsamı içinde kavrama olanağına daha çok yaklaşıyorlardı. Değişime uğrayacak madde üzerinde birden fazla faaliyet ya da iş türü uygulamaları gerekebiliyordu. Bütün bu değişimler işe duyulan ilgiyi canlandırmanın, emekçiyi işine karşı ''sorumlu'' kılmanın yolları olarak sunuluyordu. Gerçekten de bandta çalışmanın, taylorizm, fordizm uygulamalarının iş sürecine kazandırdıkları ''yoğunlaşma'' (intensification) bir sınıra erişmiş, işin bu tür örgütlenişine gelen tepkiler terazide gittikçe ağır basmaya başlamıştı: İşe ilgisizlik, işe gelmeme (absentéisme, désaffection), turn-over üretim masrafları üzerinde ağırlıkları artan ''tepkilerden'' yalnızca bazıları. Örneğin Kasım 1970'te Philips şirketinin yayınladığı bir bülten işe gelmemezliği karşılamak üzere istihdam ettikleri işçi sayısının, tüm personelin % 15-20'sini teşkil etmekte olduğunu bildiriyor, yine de neticenin memnunluk verici olmadığını ekliyordu.

Bandta çalışmanın yeniden örgütlenişine paralel olarak (yarı özerk ekipler halinde) ve de özellikle elektronik içeriği ağır olan dallarda, montaj bandından, işin ''modüller'' halinde örgütlenmesine geçişin yanısıra, sanayide robotiğin gelişmesinden, otomatlar kullanımından gittikçe söz edilmeye başlandı. Bunlar taylorizmin prensiplerini çiğneyip yeni işbölümü prensipleri doğurmaktan çok, taylorizmin, ''işin bilimsel örgütlenişinin'' prensiplerine eklemlenerek işlemeye başladılar. Yani işin bölünmüş ve tekrara dayanan bir iş olmasına eklendiler. Otomatlar ya çalışma bandının içinde işçi yerine konarak kullanılıyorlar, ya da birkaç montaj bandı koordine etmeye yarayabiliyorlar.

Robotlar ise kalifiye bir işçinin yerine getirebileceği işlemleri gerçekleştirecek programlara sahipler ve kullanımları genelleşme yolunda.

## Hizmetler Sektörü ve Büro Faaliyetlerinde Enformatikleşme Sonrası Evrim

Yeni birikim modelinde, emek gücü değerini düşürmenin başlıca yolu hizmet sektörü ve büro faaliyetlerinin enformatikleşmesiyle sağlanacak olan üretkenlik artışına bağlı gözüküyor. 1945'den 1960'lara kadar olan tarihî kesitte emek gücünün kendini yeniden-üretme süreci içinde tüketmekte olduğu metalardan yalnızca sanayi mamulü olanların birim başına değeri düşme eğilimi kaydetmişti. Şimdi ise, emek gücünün kendini yeniden-üretme sürecine giren hizmet sektörü mamullerinin üretim koşulları enformatikleşmeyle değişecek, böylece aşılanan üretkenlik artışı bu hiz-

*metlerin birim başına içerdikleri ''çalışma zamanı''nı (değişim değerini) düşürebilecektir.*

## Büro Faaliyetlerinin Enformatikleşmesi

*Bu konudaki ilk deneme ve çabalar büro faaliyetleri arasında basit olup tekrara dayanan işlerin otomasyonunu hedeflemiştir (daktiloyla yazı, düzeltmeler, baskı, stokaj, dosyalama, döküman, araştırma vs.). Bu hedeflere erişmeyi mümkün kılan makine ve ekipmanlar, donatımlar bugün artık normal şartlarda satılmaktadır. Etüd edilmekte olan bir başka sistem ise, büro faaliyetleri arasında bütün bir faaliyet tipini otomatikleştirebilme imkânları sağlayacaktır; örneğin tüm satış faaliyetlerinin idaresi gibi (satışların kaydı, gönderileceklerin hazırlanması, fatura kesimi vs. gibi). Bu durumda bilgisayarın görevi, işlerin koordinasyonunu yapmak, çeşitli görevlerde, çeşitli yerlerde işbaşında olanlara yapılması gereken işlemleri bildirmek.*

*Böylece şimdiye kadar ''işin bilimsel örgütlenişi''nin, ''taylorizmin'' uygulanmadığı büro faaliyetleri, bu tür işbölümünün prensiplerince örgütlenmeye doğru yöneliyorlar. Dolayısıyla buralarda çalışanların işi asrın başından beri sanayi işçilerinin uğramakta olduğu akibetin aynına uğruyor: İşler, faaliyetler daha çok bölünüyor, basitleşiyor, tekrara dayanır oluyor. Yani buradaki emek gücü kalifikasyonunu kaybediyor, karar mercileri ile icra unsurları arasındaki uçurum derinleşiyor, kutuplaşma yoğunlaşıyor.*

### Büro faaliyetlerinin enformatikleşmesinin mekân ve şehir üzerindeki etkileri

*Hizmet sektöründe karar ve idari mercilerle, uygulama görevleri arasındaki kutuplaşmanın kentsel örüntüde ve mekânsal yerleşimde önemli etkileri olacaktır. Bu olgu, Paris gibi hizmet sektöründe ihtisaslaşmış, sanayilerini kendi yöresinin dışına ihraç etmiş, sürmüş bir şehir için son derece belirgin bir şekilde ortaya çıkmaktadır. Paris yöresini ilgilendiren şehirsel planlar 1950'lerden beri bu yöredc ki sanayilerden arınmayı kendilerine hedef edinmişlerdi. Bu hedefe yönelik uygulamalar 1960'lardan itibaren sistematik bir şekilde gerçekleştirilmeye başlanmış, Paris'in hizmet sektöründe uzmanlaşması şehir planlarının diğer bir hedefi olarak -kâğıt üzerinde- belirmişti. Böylelikle 60'larda uygulanmasına ağırlık ve-*

*rilen bu planın, 70'lerde sonuçları maddileşmiş, Paris sanayilerinden, işçilerinden önemli oranda arınmıştır. Bu şehir hizmet sektörünün yoğunlaştığı, büyük şirketlerin, çokuluslıların idari bürolarının yerleştiği, uluslararası bir merkez haline gelmiştir.*

*Şimdi ise büro faaliyetlerinin enformatikleşmesiyle yeni bir eğilim doğmaktadır. Otomatizmin genelleşmesi, enformatiğin, sermayenin idare ve dolaşımıyla ilgili faaliyetlerde uygulanması, bu faaliyetler alanında bir yandan işgücünün dekalifikasyonunu, diğer yandan ise, bütün kalifikasyon, karar ve sorumluluun, sayıları sınırlı, belirli görevlerde kutuplaşmasını getirmiştir. Bu eğilim ise dekalifiye olan işgücünün, merkezi konum ve şehirlerden taşra ya da uzak banliyölere doğru ya da bu işgücünü karşılamak ve yerleşmesini sağlamak üzere kurulmuş ''yeni şehirler''e (villes nouvelles) doğru ihracını mümkün kılmaktadır. Dolayısıyla Paris yalnızca, global olarak tüm hizmet sektörünün, büro faaliyetlerinin merkezileştiği bir yöre değil, bu sektör içinden, yani hizmet sektörü ve büro faaliyetleri arasında, yalnızca karar ve idari mercilerin yerleştiği, en kalifiye görevlerin merkezileştiği bir yöre oluyor.*

*Bu sektörden, bu tip faaliyetlerden olup, karar ve idari görevlere yakından bağlı olmayan, icraya, uygulamaya yönelik, kalifikasyonsuz görevler Paris'i ''yeni şehirler''e, uzak banliyö ve taşraya gitmek üzere terkediyorlar. İcraya uygulamaya bağlı ve artık fazla bir kalifikasyon gerektirmeyebilecek görevler, faaliyetler -1960'larda dekalifikasyon süreci ve taylorizasyonu tamamlanmış sanayi faaliyetleri için olmuş olduğu gibi- Paris dışına sürülebiliyorlar. Şubat 1980'de devlet sorumluları Paris'ten, periferisindeki ''yeni kent''lere 700.000 görevin, yani kişinin ihraç edileceğini açıklıyorlardı.*

*''La Défense'' ve ''Gare de Lyon'' gibi mahallelerin bugünkü işlev ve görünümleri bu çifte eğilimed oğrudan bağlı. La Défense ve Gare de Lyon-Quai de Bercy yörelerinin görünümlerini belirleyen gökdelenler, idari kuleler, çokuluslu şirketlerin, büyük monopollerin büro faaliyetlerini, idari işlevlerini, karar mercilerini barındırıyorlar. Bunlar, devlet mekanizmasına, politik karar mercilerine yakın olmaları gereken personel ve büro faaliyetlerini barındıran, prestijli enternasyonal -kulesel, gökdelensel- vitrinler. Halihazırda bu kulelerdeki personelin ve servislerin önemli bir kısmı yukarıda sözü edilen ''yeni kentlere'' gönderilecek 700.000 kişiye dahil. Yani burada tariflenen süreç, büro faaliyetlerinin ''yerinden yönetim'' siyasetince, coğrafi konum değiştirmeleri süreci tamamlanmış değildir. Bu faaliyetleri yeni kentlere ihraç ettiren siyasetin resmî gerekçesi Paris'te kâfi miktarda inşa edilmiş büro alanı (metrekare hesabıyla) olmadığı, yeni kentlerde boş bürolar bulunduğunu ileri sürmektir. Yeni kentlere niye büro inşa edildiği sorusu sorulmamaktadır. Enformatikleşmenin kent örüntüsü üzerindeki mekânsal etkilerinin bu kadarla kalmayacağını, çok daha yaygın ve kökten değişiklikleri beraberinde getireceğini düşünmek gerek. Örneğin ''ulaşım sorunsalında'' büro faaliyetlerinin enformatikleşmesi ve bu tür donatıma bağlı çalışma yöntemlerinin gelişmesiyle, ileri bir safhada kişilerin, profesyonel faaliyetlerini (kısmen ya da tamamen) oturdukları evden gütme imkânlarına erişebilecekleri düşünülen, planlanan imkânlardan. Bunun dışında şehirlerarası ulaşımı etkilemesi beklenen bir olgu da ''telekonferansların'' genelleşmesi. Telekonferans ikiden fazla kişinin uzaktan ''toplantı'' yapabilmelerini sağlayan ekranlı bir telekomünikasyon sistemi. Bu sistem 60'lardan beri ya video-konferans, ya odio-konferans ya da bilgisayarlı konferans olarak denenmekte. İletişimin ''gerçek-zamanda'' gerçekleşmesi, idari personelin, işadamlarının, şirketin çeşitli şehirlerdeki şubeleri arasında yapılacak toplantılar için uçağa binip yer değiştirme zorunlalığını kaldırıyor.*

*Aşağıda sözü edilecek elektronik posta, telematik vs.nin sağladığı ''ulaşımdan tasarruf'' imkânları üzerinde biraz düşünülecek olursa, bu birikim modelinde, kapitalist üretim tarzı için önemi bilinen kişi ve malların taşımı sorununun (dolayısıyla orta ya da uzun vadede petrol sorununun) yeni bir örüntü içinde karşımıza çıktığı görülür. Biraz spekülasyon yapılacak olsa muhakkak ki bu birikim tarzında önemli olanın insan ve malların dolaşımından, ulaşımından çok, veri ve enformasyonun ulaşım ve iletişimine bağlı olduğu ileri sürülebilir.*

## Telekomünikasyonların Enformatiğe Bağlı Evrimleri

*Enformatiğin televizyon ve telefonu da katarak (entegre ederek) kurmuş olduğu ''veri iletişim'' sistemine ''telematik'' (ya da teleenformatik) deniyor. Telematiğin mümkün kıldığı hizmetler çeşitli. Ancak burada dikkatimizi iki türlü hizmet çekecektir: 1. Halkın emrine yeni, kolektif bir altyapı sunulması, 2. Veri iletişim ağları oluşturulması ve parasal ''fonların'' elektronik transferi ile elektronik posta servisleri.*

***a) Veri İletişim Ağları (Réeseaux de Transmission de Donnés)***

*Bu ağlar gelişecek olan yeni birtakım servislerin altyapısını oluşturdukları için bu alanda dinamik bir siyasetin vazgeçilmez bir açısı. Bunlar telekomünikasyon sisteminin özel ve en kârlı hizmetleri olarak gerçekliklerini kazanmaktalar. Bu işletim ağlarına örnek olarak Fransa'da gelişmesine çok ümit bağlanmış TRANSPAC sisteminin niteliği açıklanabilir. Bu ağ, nümerik bilginin (verilerin) paket halinde iletişimini sağladığından daha ucuz bir hizmet sunabiliyor, genelleşme şansını dab una borçlu. Devlet dairelerinin, çokuluslu şirketlerin, büyük irma ve bankaların şubeleri, çeşitli servis ve ajansları arasında idareyi, veri iletişimini süratle sağlamak üzere (stokların, muhasebenin, fiş sisteminin idareleri için) kullanılmaya elverişli gözüken, en ucuz sistem Transpac.*

***b) Fonların Elektronik Transferi***

*Bankalararası ya da banka ve çeşitli şubeleri arası, ya da herhangi bir ticari alış-verişte gerçekleştirilen her türlü ödeme, borçlanma, kredi gibi işlemlerin ''gerçek zamanda'' yapılabilmesini bilgisayar ve terminallerarası veri alış-verişiyle sağlayan bir sisteme, bir ağa dayanıyor fonların elektronik transferi. Tedavül hızını çok süratlendireceği için enlasyonist etkileri olacağı varsayılabilir.*

***c) Elektronik posta***

*Bugün herhangi bir coğrafi noktadan diğerine 4-6 dakika içinde 21 x 29.7 ebatlı bir dökümanı ileten cihazlar mevcut. Bunun yanısıra aynı faaliyeti birkaç saniye içinde gerçekleştiren sistemler de mevcut. Fransa bu alanda 1981'de kullanılır duruma gelecek olan ve geniş halk kitleleri için telekopi sistemi geliştirmekte. Gelişmiş, sanayileşmiş ülkelerin ço-*

ğundaki posta bütçe açıkları gözönüne alınacak olursa enformatiği bu alanda uygulamak için yapılan araştırmaların nedeni kolayca anlaşılacaktır. Şirketin çeşitli şubeleri, ajansları arasındaki mektuplaşmanın, fatura, finans belgeleri ve çeklerin posta aracılığıyla gidiş gelişlerinin azalacağı, devlet daireleri arası muhaberatın bu elektronik postanın imkânlarından yararlanarak azalacağı tahmin edilebilir. Tyler'e bakılırsa yalnızca fatura ve parasal tedavül belgelerinin posta aracılığıyla değil de elektronik yolla iletilmesi halinde posta hizmetleri % 30 oranda hafifleyecek. Basın ve gazeteciliğin bu imkânları yaygın biçimde kullanması bu hafiflemeyi daha da önemli kılacaktır. Gazetelerin bir yerde basılıp dağıtımının o yerden itibaren yapılması yerine, yalnızca redaksiyon ve mizanpaj merkezi bir yerde gerçekleştirilip, çeşitli noktalardaki matbaalara fak-simile olarak iletilebilmekte ve gazetenin baskısı (tiraji) ve dağıtımı o noktalardan yapılmaktadır.

## Yeni Bir Tüketim Standartı Oluşması

Yukarıda sözü edilmiş olan ''telematik'', yeni tüketim standartını oluşturan teknik altyapının önemli bir parçası. İlerde maddelere ayırarak açıklayacağım yeni tüketim normunu oluşturan alanların hepsinde telematik kullanımını bulmak mümkün. Zira yaygınlaşacak olan, ''domestik telematik'' kullanımıdır. Örneğin bir ev hanımı daktilo klavyesini andıran bir klavyeyi kullanarak evindeki televizyon ekranı üzerinde belirmesini sağlayacağı çeşitli haber ve enformasyona erişebilecektir (meteorolojik bülten, siyasi haberler, telefon rehberinin verileri, mağazaların açılış ve kapanış saatleri vs. gibi).

Şimdi teknik açıyı bırakıp genelleşecek tüketim alanlarını ayrı ayrı incelemek mümkün. Burada 3 alan üzerinde durulacaktır; kültür-eğitim, boş-zaman tüketimi (loisirs) ve sıhhat. Bunlar tabii ki işçi sınıfının da uzun zamandır zaten tüketmekte olduğu hizmetler. Ancak yeni birikim modelinin özelliği, üretim şartlarının enformatikleşmesiyle bu hizmetlerin niteliğinin de değişmekte olduğudur. Seri üretime geçiş ve bunun sonucunda metanın niteliğinin değişimi, sözkonusu hizmetlerin ''demokratikleşmesinin'' koşulu olarak belirmektedir. İkinci Dünya Savaşı'ndan sonra, işçi sınıfının tüketim sepeti Batı ülkelerinde nasıl kapitalist ilişkiler içinde üretilmiş mallarla dolmuşsa, şimdi de işçi sınıfı ve orta sınıflar yeni şartlarda üretilen ''kültür'', ''sıhhat'', ''boş zaman'' gibi hizmetlerin en önemli müşterisi olarak belirmektedirler. Bunun oluşabilmesi için, bu hizmet dallarındaki üretim şartlarının değişmiş, önemli bir üretkenlik artışı kaydedilmiş olması gereklidir. Enformatik, teknik işbölümünü yeniden örgütlemesiyle ve mümkün kıldığı üretkenlik artışıyla bunun maddi koşulu olarak belirmektedir. Tıpkı 1926'da yassı-çelikte beliren yeni haddeleme teknolojisinin yarattığı yeni vasıflara sahip çeliğin 1945-1974 arası ''boom''u bilinen tüketim mallarına sağladığı gelişme imkânı gibi.

''Kültür'' konusuna bir örnekle başlanabilir: O da Paris'teki ünlü kültür kompleksi ''Centre Georges Pompidou''nun (diğer adıyla Beaubourg'un)k ütüphanesi. Bu kütüphanenin Fransa için bir özelliği var: Buraya giren okuyucu istediği kitabın kod numarasını kütüphane memuruna verip, kitabın kendisine teslim edilmesini beklemiyor, kodu enformatize bir katalogdan bulup, gidip raflardan kitabını kendi alıyor. (1981'den itibaren ise okuyucu hizmete açılacak terminallerde arayacaktır bibliyografyasını). Böylelikle Paris kütüphanelerinde ortalama yarım ile bir saat arasında değişen bekleme süresi burada sıfıra inmiş oluyor. Oysa bu özellik, hem söz konusu hizmetin niteliğini, hem de tüketicisini farklı kılıyor. Önce tüketicinin değişik niteliği hakkında: Geleneksel kütüphanenin geleneksel tüketicisi çoğunlukla kitaplarla çalışmayı kendine iş edinmiş, bir dökümanı bir-birbuçuk saat bekleyebilen araştırıcılar, öğrenciler, entelektüeller. Kitabı okuyucusuna bekletmeden sunabilen kütüphane, vakti daha kısıtlı olanı, kütüphaneye gidip gelmeyi alışkanlık, uzmanlık edinmemiş olanı da çağırabilme imkânına sahip; yalnız entelektüelleri değil, ev kadınlarını, öğle ya da akşam tatilindeki memuru, belki işçiyi kendine çekebilmeyi amaçlıyor. Beaubourg'da olduğu gibi, sayısı 1 milyonu aşan dokümanı yönetebilen, günde 10—18 bin kişilik bir kitlenin bu dokümanlara beklemeden erişmesini sağlayabilen yönetimsel teknolojinin temelinde ise enformatik var. (1979'da kütüphanenin enformatize kataloğunun ''multimedia'' bir katalog haline gelmesi söz konusuydu, yani mevcut video-kasetlerin, filmlerin, slaydların, plakların bu katalogda yer alması bekleniyordu.)

*Ancak yeni kitlelerin kütüphane ''tüketmeye'' başlamaları için mevcut kitapları enformatize bir kataloga göre tanzim etmenin yetmeyeceği aşikâr. ''Sunulan hizmetin de niteliğinin değişmiş olması gerek.'' Bu kütüphanelerin belirli uzmanlık yapıtlarından, bilimsel araştırma kitaplarından çok, genel ve ''vulgarisation'' niteliği de taşıyan yapıtlara yönelmesi, bunun yanısıra örneğin ev kadınlarını ilgilendiren örgü, moda, ev döşeme dergilerinin bulundurulmasına da titizlik gösterileceği beklenmelidir. Nitekim 1978'de görüştüğüm Beaubourg kütüphanesi müdürü, ellerindeki dokümanların henüz çok uzmanlaşmış bir nitelik taşıdığını, fazla bilimsel kitap bulundurduklarını ve projenin bunu değiştirmek olduğunu anlatıyordu. Gerçekten de 1978'de günde 10-18 bin kişinin girip çıktığı bu kütüphane okuyucusunun % 40'ını öğrenciler, % 20'sini özellikle taşra şehirlerinden gelen turistler oluşturuyordu. Okuyucu sayısının % 40'ını öğrencilerin oluşturması ve ortalama yaşın 300u geçmeyişi her ne kadar Beaubourg kütüphanesinin amaçlarına ters düşüyorsa da, bu kütüphanenin, bu türde geliştirilecek kütüphaneler için bir prototip niteliği taşıdığını unutmamak gerek. Bunun gibi, dışarıya ödünç kitap vermeyen, ''kamu okuma kütüphaneleri'' (bibliothéque de lecture publique) denen ve Beaubourg'da geliştirilen enformatik katalog sistemine göre yönetilecek olan kütüphaneler Fransa'nın Lyon, Grenoble, Nice gibi taşra şehirlerinde de kurulmak üzeredir. Bu tür bir kütüphane olan New York Public Library ise 7-8 yıldır işlemektedir ABD'de.*

*Bunun gibi araçlara dayanan bu tip kültür yayılmasının yanısıra üniversite ve araştırma mercilerinin kredileri sistematik bir siyasetle sürekli kısılmakta ve teknisist alanlar lehine yeniden örgütlenmektedir. Bazı yayınevlerinin, kitap basımının Fransa'daki krizi de kanımca bu süreç içinde değerlendirilmelidir.*

## Boş Zaman Tüketimi

*Kültürün yanısıra ''demokratikleşen'', belirli boş zaman kullanış tarzlarından söz edilebilir. Bu konuda da tabana yayılmayı mümkün kılan, bu hizmetlerin üretim koşullarındaki değişimler. Örneğin, seyahatler konusunda: Müşterek seyahatler, grup halinde turlar, ''club méditerranné'' türü tatiller çoktan beri mevcut. Ancak şimdi bu konudaki rezervasyon gibi büro işleri çeşitli açılardan otomatikleşmekte. Kişiler belirli seyahatleri için gerekli rezervasyonların tümünü (tren ya da uçak biletlerini, otel odalarını, tiyatro biletlerini,*

*Hertz'den otomobil kiralamayı) evlerindeki terminallerden yapabileceklerdir. Yukarıda sözü edilmiş olan Transpac ağı buna da yarayabilecektir. 1981'de Fransa'da Velizy yöresinde 2 binden fazla aile deneysel olarak bu teknik imkânlarla donatılacaktır.*

*Bilindiği gibi tren, uçak gibi ulaşım konularında rezervasyon sistemi birkaç yıldır enformatiğe bağlanmış bulunuyor. Uçak şirketleri arasındaki rekabetin bugün hâlâ kayıt, rezervasyon, bilet kesme gibi faaliyetleri azaltmaya yönelik olduğu ileri sürülebilir. (Örneğin Lacker adlı ''charter'' şirketi, uyguladığı düşük bilet fiyatlarını, müşterilerini, rezervasyon yapmadan uçurtmasına borçlu.) Bütün bunlar bu alanda otomasyonun bitmemiş olduğuna ve genişlemesine devam eden bir enformatik donatım pazarıyla karşı karşıya olunduğuna delalet eden işaretler.*

*''Sıhhat'': Diagnostik yapmakta da kullanılan enformatiğin, sıhhat, bakım gibi alanlarda da kitlelere daha geniş tüketim imkânları yaratabilmesi çoklukla hastane, dispanser ve muayenehanelerin hastalar hakkında (hastalıkları, görmüş oldukları tedavi türleri, verilmiş ilaçlar, yapılmış ameliyatlar ve ameliyat sonrası, takip edilmesi gereken bakımları hakkında) bilgisayarlarca yönetilen fişler tutabilmelerine dayanıyor.*

## Çalışma Zamanı/Boş Zaman Oranının Değişmesi

*Yukarıda maddeleri sayılan tüketim standartının yerleşmesi, haftalık çalışma süresinin kısalmasını, buna mukabil boş zaman süresinin uzamasını beraberinde getirecektir. Gerçekten de bu yeni tüketim normunun yerleşme arifelerinde olan Fransa, Almanya gibi ülkelerde bir yıldır kuvvetle gündeme gelmiş bir slogan; ''haftalık çalışma süresinin 40 saatten 35 saate indirilmesi'' gittikçe önem kazanıyor. Bu iki eğilimin -yeni bir tüketim standartı oluşması eğilimiyle iş süresinin kısalması eğiliminin- eş zamanda gelişmesi herhalde bir rastlantı olmasa gerek. Aksine, bunların eş zamanda oluşması karşımıza tarihsel bir gereksinim olarak çıkıyor, yeni birikim modelinin ge-*

*reksinimi olarak. Yukarıda da belirtilmiş olduğu gibi ekonomiye üretkenlik artışı aşılanmasının nedeni sömürü oranını yükseltme gereğinden kaynaklanır. Sömürü oranı ya gerçek ücretlerin de artmasına müsaade edecek kadar, ya iş süresinin ücretler düşmeden kısalmasına izin verecek kadar, ya da ikisini de mümkün kılacak bir oranda yükselmiş olacaktır ki... yeni bir birikim safhasına girilmiş olunsun. Bizi ilgilendiren 1980'ler ve sonraları safhasında iş süresinin de kısalması teknik olarak gerekli. Zira birikimi sağlayacak söz konusu kültür, sıhhat vs. gibi alanlarda tüketim vakit de gerektiren bir olgu.*

*Sendikaların, işçi örgütlerinin haftalık iş süresini 35 saate indirmek istemeleri muhakkak ki sermaye birikimini hızlandırmak, sermayeye yardımcı olmak için değil. Önemli olan böyle bir isteğin, ifade edilebilen, dolayısıyla belirli bir güç ve bilince dayanan bir isteğin, işçi sınıfınca ifade edilebilmesi anıyla, sermaye için böyle bir durumun yaratılması gereğinin -yani iş süresinin kısalması gereğinin- yaklaşık olarak aynı zamanlara rastlaması. 45'lerden bu yana işçi sınıfı mücadelesinin ve ücretlerin artmaya devam edişinin sonucunda ortaya çıkan sömürü oranını artırma imkânsızlığına sermaye sınıfının bulduğu cevap, yeni bir makine, yeni bir teknoloji ve yeni üretkenlik artışı ve bunların gerektirdikleri yeni tüketim standartlarının yerleşmesini sağlamaktı. Bu standartlar yalnız malların niteliğini değil, ne zaman ve nasıl tüketileceklerini de belirleyen standartlar. Bu sermayenin nasıl salt bir zenginlik, bir üretme kapasitesi olmakla kalmadığını, nasıl bir üretim ilişkisi olduğunu, bir sistemi, kapitalist üretim tarzını belirleyen bir üretim ilişkisi olduğunu örnekleyen belgelerden biri. Ekonomik ve siyasi tarihi yazan öznenin bir sınıf olmadığını, kapitalist üretim tarzı tarihinin ne sermaye sınıfınca ne de işçi sınıfınca yazıldığını, bu tarihin iki sınıf arasındaki mücadeleden -ve özellikle üretim süreci içindeki mücadeleden- doğan dinamikçe yazıldığını ifade eden bir gerçek.*

## Devlet rolündeki değişiklik

*Yukarıda tasvir edilen sürecin yanısıra; değişen ekonomik gerçekleri değişen kurumlar, değişen örgütler çerçevelemektedir. Bunların içinde ''devletin'' değişen niteliklerine yüzeysel olarak da olsa değinmeden geçmek olanaksız. Önce, bir evvelki birikim tarzıyla karşılaştırarak hemen söylenebilecek olan; 1980'lerde devletin ''sosyal devlet'' (Etat providen-*

ce) niteliğini tamamen terkedeceğidir. O kadar ki neredeyse ''kamu hizmeti'' (service public) kavramı anlamını tamamen kaybedecektir. Ekonomideki neo-liberalizm akımları bu oluşumun ideolojisi olarak karşımıza çıkmaktadır. Fransa'da bu değişimin uygulayıcısı olan siyasi varlık kuşkusuz Başbakan Raymond Barre'dir. Siyasetini gerekçelemek için kullandığı terimler, uyguladığı programın tümünü özümlemektedir. ''Gerçek fiyatlar'' siyaseti örneğin. Gerçek fiyatlardan söz ederek devlet ''subvansiyonu'' sayesinde, şimdiye kadar maliyetinden daha alçak fiyatlar ödendiği varsayılan hizmetlere (ulaşım vasıtalarında bilet fiyatları, sosyal sigorta kesintileri gibi) bundan böyle, piyasa maliyet fiyatlarına göre hesaplanmış bedeller ödeneceği ilan edilmektedir. Yeni liberalizmin görüşleri uygulanmaktadır: ''Piyasa'', pazar denilen bir olgu vardır ve bu olguyu derleyen, yöneten ''tabii'' kurallar, doğa kanunları vardır (arz talep kanunu gibi). Devletten istenen, bu kanunların işleyebilmesi, varlık ve gerçekliklerini ispatlayabilmeleri için gerekeni yapmasıdır. Yani devletten, hükümet edenden çok fazla bir şey yapmaması, pazarın kendi ''tabii'' kanunlarınca yönetilmesine izin vermesi istenmektedir; bilinen, malum ''bırakınız yapsınlar''a itaat etmesi. Raymond Barre'in dilinde de bu siyaset ''gerçek fiyatlar uygulama'' siyaseti olarak şekilleniyor.

1980'ler ve sonralarında devletin oynayacağı rol 1945-74 modelinde devletin oynamış olduğu rolün tamamen aksi. İkinci Dünya Savaşı'ndan sonraki entansif birikim modelinde işçi sınıfının, orta sınıfların ekonomiye sınai sermayenin en önemli müşterisi olarak dahil edilebilmesi için, devletin, bu sınıfların alım gücünü garanti etmesi gerekiyordu. Ekonomik buhranların (özellikle 1929 krizinin) ve getirdikleri işsizliğin anısını silmek gerekiyordu. Bunu yapacak sosyal varlık, devletti. (Bu senaryonun ideoloğu da bilindiği gibi Keynes'di.) İşçinin kapitalist üretim tarzına yalnızca üretici olarak değil, aynı zamanda önemli bir tüketici olarak katılması için satın alma yeteneğinin belirli bir ''güven duygusu içinde, tüketme imkânına dönüşebilmesini sağlamak (kötü günler için para biriktirmesini engellemek)'' gerekiyordu. Batı Avrupa ekonomilerinde devletin İkinci Dünya Savaşı sonrasında açtığı ''sosyal güvenlikler'' şemsiyesi (ihtiyarlık, emeklilik, hastalık, işsizlik vs. sigortaları) bunun önemli bir parçasıydı. (Gelişen bir başka açı da tüketim kredileriydi.)

Şimdi ise ve yine liberal ekonomiye dönüş siyaseti içinde, sosyal sigortaların iş ve işçi bulma kurumlarının tümü Fransa gibi bir memlekette bir yandan enformatikle donatılırken, diğer yandan kârlılık prensiplerince yönetilebilmek üzere örgütleniyorlar, özel şirketlere devirleri hazırlanıyor. Bunun yanısıra, elektrik ve gaz idarelerinin, posta ve telefonun aynı evrimlerden geçeceklerini ileri sürmek için kâhin olmak gerekmiyor. Bu evrimlerin ''enformatikle donatım'' açısı bu kurumlarda hayli zamandır gerçekleşmiş bulunuyor. Eskiden ''kamu hizmeti'' dediğimiz bu hizmetlerin bu tarzdaki yeniden örgütlenişi karşısında François Châtelet gibi bir düşünür, kişinin vatandaşlık statüsünün yerine bir başka statü, bir ''kullanan'', ''tüketen'' (usager) statüsü yerleşmekte olduğunu açıklıyor.

Yukarıda sözü edilen kamu hizmetlerinin özel sektöre devrinin yanısıra merkezi devletin kendisine bağlı yerel mercilere gücünden, görev ve sorumluluklarından bir kısmını terketmesi, yani ''yerinden yönetim'' (= décentralisation) sürecinde ilerlemesi beklenmeli, devlet görüntüsündeki merkezi bir devlet mekanizmasıyla idare olunan bir memlekette, örneğin vilayetlerin (préfectures) sorumluluklarının kapsamının değişeceği, hareket alanlarının (coğrafi anlamda değil) genişleyeceği gündemdedir.

Aslında ''yerinden yönetim''e geçiş süreci gibi sunulan bu olgu enformatizasyona yakından bağlı olup, merkezi devlet mercilerinin konum ve sorumluluğundan hiçbir şey eksiltmediği gibi, merkezi niteliğini daha da derinleştirecektir. Enformatik geniş bir fişleme, verileri stoklama imkânı yaratmaktadır. Rutin ve güncel işler ve prosedürler için yerel mercilerin sorumluluk ve görevlerini genişletmekte bir sakınca yoktur. Ancak olağandışı herhangi bir sorun, bir prosedür sözkonusu olduğunda merkezi merciler, yerel mercilerin fişlemiş oldukları verileri enformatik sayesinde inceleyip, ona göre ve ''kendileri'' karar verebileceklerdir. Dolayısıyla ''yerinden yönetilen'' güncel, otomatik prosedürler olacak, ince kararlar konusunda merkezi merciler işe karışabileceklerdir. Bir benzetme yapmak gerekirse şöyle ifade edilebilir: Bugüne kadar a.b.c... gibi maddeleri aşmayan yerel mercilerin görev ve sorumlu-

*luk alanları, bundan böyle belki alfabenin son harfine kadar bütün harfleri (maddeleri, alanları) dahil edecek şekilde genişleyecektir. Ancak bundan böyle, bütün bu harflerin kapsam ve içeriği de değişmiş, daralmış ve fakirleşmiş olacaktır.*
*Yazının burasında, değişen birikim modelinin uluslararası ekonomik işbölümlerinde önemli gelişmelere ve yeni gerçeklere yol açacağının ileri sürülmesi doğal karşılanacaktır. Ancak bu konunun ve 1980 sonrasının birikim modelinde, Batı ekonomilerinde ''işçi sınıfı oluşumundaki'' değişiklikler konusunun incelenmesi hernekadar zorunluysa da, yazının boyutlarını çok genişleteceğinden bunlara burada değinmemeyi tercih ettim.*

Nora Şeni (''Ekonomik Kriz ve Yeni Bir Birikim Modeli Senaryosu Üzerine'' yazısından Toplum ve Bilim, 14 Yaz, 1981)

## Mikro-Elektronik Devrim

*Temel topluluk birimlerinde kökleşmiş ademimerkeziyetçi bir gelişmeyi engelleyen sanayileşmeci dönemin büyük çaplı teknolojilerinden farklı olarak, otomatizasyon gerçekten de toplumsal açıdan değişken bir nitelik taşır. Büyük çaplı teknolojilerin kilit-teknolojiler olmalarına karşın mikro-elektronik, bir kavşak-teknolojisidir; çünkü belli bir gelişme tipini ne yasaklamakta, ne de kabule zorlamaktadır. Elektronükleer ya da uzaysanayilerinin tersine, özdenyönetime olduğu ölçüde aşırı-merkeziyetçiliğe, hatta özdenyönetilen merkeziyetçiliğe bile hizmet edebilir. Bununla birlikte, yeni ve uzun bir kapitalist birikim süreci başlatamaz.*

*Bu durum çeşitli yabancı araştırmalarda ve özellikle de Federal Alman Ağır Sanayi İşçileri Sendikası'nın (Frankfurt'taki I.G. Metall) otomasyon bölümü sorumlusu Günter Friedrichs tarafından kapsamlı biçimde kanıtlanmıştır. Robot ve büro teknolojisi klasik çalışma gereçlerinin hem daha üstün, hem de üretilen birim başına daha ucuza malolan donanımlarla değiştirilmesine imkan vermektedir. Üstelik bunlar, yatırımdan (değişmeyen sabit sermaye), iş gücünden (değişken sermaye) ve hammaddelerden (değişmez dolaşan sermaye), özellikle de enerjiden ''aynı zamanda'' tasarruf sağlamaktadır. Radikal yenilikleri buradadır. Bu yenilik ''mikro-elektronik devrimi'' deyimini tümüyle doğruluyor.*

*Elektrik motorunun ortaya çıkışından bu yana ilk kez, birinci sektördeki (üretim araçlarının üretimi) üretkenlik, ikinci sektördeki (tüketim malları üretimi) üretkenlikten daha hızlı ve kalıcı biçimde gelişmektedir. İşçisi olmayan ilk fabrikalar, robotların yeni robotlar ürettikleri fabrikalardır. Üretilen birim başına düşen değişmeyen sabit sermaye değeri hızla azalmakta ve değişken sermaye değeri de bakım emeğinin gerektireceği bir düzeye inme eğilimine*

*girmektedir. Kısacası teknik devrim, ekonomik akıl yürütmenin temellerini sarsan bir dönüşümü peşinden sürüklüyor. Bu, geçmişteki teknik devrimlerde olduğu gibi yalnızca üretilen birim başına düşen sabit sermaye değerini düşürmekle kalmaz; çabucak büyüyen bir meta hacmini üretmek üzere harekete geçirilmiş sermayenin ''bütününün'' de küçülmesi sürecini başlatır.*

*Devlet, yatırım arzusunu uyardığı zaman, buna bağlı olarak üretim araçlarının yenilenişini ve kâr haddinin yeniden oluşumunu, ama aynı zamanda sermayenin de tahribatını (işletmelerin kapanışı, varolan donanımların sökülüp kaldırılması yoluyla) ve iş alanlarının ortadan kalkışını da uyarmış olur. Federal Almanya'da yapılan bir araştırmaya göre, sınai donanım alanına yatırılan yüz milyar Deutsche Mark, 1955-1960 arasında 2 milyon yeni iş, 1960-1965 arasında da 400 bin yeni iş yaratabiliyordu. 1965-1970 arası dönemde sözkonusu tutar 100 bin iş alanını ''ortadan kaldıracak'' ve 1970 ile 1975 arasında bu miktar 500 bin iş alanını bulacaktı. Fransa'da da sanayide ''gözlemlenen'' işgücü gerilemesi, yılda 53 bin iş alanına kadar çıkar.*

*Şu halde yatırımın yapacağı hamle, en azından teknolojik dönüşüm son aşamasına ulaşmadığı sürece, büyümeyi hızlandıramaz. Kaldı ki otomatizm üzerine kurulu yeni bir büyüme süreci, daha yeni yeni boy göstermeye başlayan bir kültürel dönüşümü ve ekonomik sistemlerin kendi dönüşümlerini öngörmektedir. Çalışma süresi, bundan böyle, mübadele değerinin ölçüsü olamayacağı gibi, mübadele değeri de ekonomik değerin ölçüsü olamaz. Ücret emeğin niceliğine göre belirlenemeyeceği gibi, bir gelire hak kazanmak da bir işte çalışmaya bağlanamaz.*

*Bu dönüşümlerin tanımı ve politik yönetimi, yaşanan bunalımın temel bilinmez-beklentileridir.*

## Olmayacak Kapitalizm

*Mevcut bunalımın yeni olan yönü, kapitalizmin teknolojik değişmeler aracılığıyla bunalıma karşılık verme çabasının artık kapitalist akılcılık çerçevesi içinde tutulamayışıdır. Bu değişimler, sermayelerin ve iş alanlarının tahribini hızlandırarak, büyüyen miktarlardaki malın, hızla küçülen miktarlardaki sermaye ve emekle üretilmesine imkân vermekteler. Kuşkusuz ''daha azla daha çoğu yapmak'', ekonomi-politiğin her zaman hedeflediği şey olmuştur; ancak bu hedefe ''mutlak'' niceliklerin büyümesini (sermayeden daha hızla büyüyen bir üre-*

*tim nedeniyle daha verimli olan sermaye) yalnızca ''görece'' büyüklüklerin (üretilen birim başına daha az sermaye ve emek) izlemesi halinde ulaşabilirdi. Oysa gördük ki otomatizasyon, büyüyen miktarlardaki maddi ve maddi olmayan malların üretilmesiyle değerlenebilecek mutlak sermaye miktarlarının azalmasına neden olmaktadır.*

*Miktarı azaltılmış bu sermayenin, geçmişteki-ne oranla çok daha verimli olduğu bir gerçektir. Ne ki asıl bu çok yüksek düzeydeki verimlilikleri sorun yaratmaktadır. Tam otomatik bir fabrikada canlı emek miktarı, tıpkı ücret şeklinde dağıtılan emek miktarı ve (Marksist anlamda) ürünün ekonomik değeri gibi sıfıra doğru inme eğilimi gösterir. Bir başka deyişle otomatizasyon, potansiyel alıcılarla birlikte emekçileri de ortadan kaldırır. Üretimin tüketimi, bu durumda, satınalma gücünün, klasik ekonomik devrenin dışındaki yöntemlerle, yani meta mübadelelerine dayanmayan ve emeği ödüllendirmeyen yöntemlerle dağıtılması yolundan sağlanabilir. Ürünlerin meta değeri (mübadele değeri) de, otomatik makine donanımının bakım, yeniden üretim ve çalışma masrafına eşit bir büyüklüğe yönelir.*

*Burada, Marx'ın* **Grundrisse***'lerde açıkladığı, ''mübadele değerine dayalı üretimin çöküşü''-yle karşı karşıyayız. Başka bir deyişle, üretimin hedefi artık mübadele değerinin, artı-emeğin ve sermayeyi büyütmek üzere artı-değerin büyümesi ''olamaz'', zira otomatik fabrika artık hemen hemen artı-değerden başka bir şey üretmemektedir. Otomatizasyon yaygınlaştıkça ve artı-değerin birikimi ve oranı da arttıkça, bu artı-değerin sermayeleştirilmesi imkânsızlaşır; verimli bir yeniden-yatırım imkânı kalmaz.*

*Şu halde, ekonomik yönetimin özel biçim ve hedefleri kapitalizmin öngördüklerinden ibaret ''kalamayacağı'' gibi, toplumsal ilişkiler de emek gücünün satılmasına, yani ücret düzenine bağlı kalmaya devam edemez. Bu yönetim, sosyalist de ''olamaz'', çünkü ''herkese emeğine göre'' ilkesi olaylar içinde aşılmış ve üretim sürecinin toplumsallaşması (ki Marx'a göre, bu sürecin sonunda sosyalizm kurulur) çoktan tamamlanmıştır. Dolayısıyla otomatizasyon, kapitalizmin de, sosyalizmin de ötesinde bir yere götürmektedir. Bu yer, temel bilinmez-beklentidir ve başlıca iki biçim alabilir: Tam anlamıyla (Alain Touraine'in deyimiyle) ''programlanmış'' teknokratik toplum biçimi ve Marx'ın ''komünist'' olarak tanımladığı, özgürleşmiş toplum biçimi... Bu ikinci toplumda gerekli olan gerektiğince üretilecek ve üretim her bireyin zamanının ancak çok kısıtlı bir bölümünü işgal edecek ve dolayısıyla ''emek'' (ücretli emek) başlıca faaliyet olmaktan çıkacaktır.*

*Bu iki toplum biçiminin zorunlu ortak yanı, güvence altına alınmış bir gelir ve bir işe sahip olma hakkı arasındaki ayrılıktır. Gene de, temelde birbirlerinden tamamen farklıdırlar.*

## Çalışma Toplumunun Sonu

*Mikro-elektronik devrim, emeğin ortadan kaldırılması çağını başlatır. Bunun iki anlamını da kavramak gerekir:*

*a) Gerekli emek miktarı, maddi üretimlerin ve örgütlenme faaliyetlerinin çoğunda, marjinal hale gelecek ölçüde hızla azalır.*
*b) Emek, emekçinin işledii maddeyle karşı karşıya gelmesini gerektirmez. Maddenin değiştirilmesi, anında, eksiksiz ve egemen bir faaliyetin sonucu değildir artık.*

*''Marksistlerin'' çoğunluğunca bilinmeyen ya da kapitalizmin yozlaştırıcı etkilerine atfedilen bu gelişmeler, Marx'a bakıldığında, tam tersine kapitalizmin ortadan kalkışını ve kömüniz-*

*min nesnel olgunlaşmasını haber vermekteydi. Gerçekten de Marx, ''iş araçlarının otomatik süreç içindeki dönüşümünüyle'', bireysel emeğin ''anındalığı ve öznelliği içinde ortadan kaldırılarak toplumsal emeğe dönüşmesinin'' koşut olarak ilerleyeceğini öngörür. Emeğin toplumsallaştırılması, böylece, meslekler adını verdiğimiz kişisel becerileri ortadan kaldırır, çünkü bireye, kişisel yeteneklerini yanında taşıyan bir insan olma yerine, toplumsal açıdan belirlenmiş yetenekleri taşıyan, ''aracısız bir toplumsal faaliyet''e katılan toplumsal bir birey niteliği verir. Bu evrim hemen hemen tamamladı.*

*Marx'ın öngörüsüne uygun olarak, işin doğası (önceden belirlenmiş niteliği) işle ilgisini kesti: ''Çalışıyorum'' ya da ''bir işim var'' deniyor, ''şunu ya da bunu yapmaya çalışıyorum'' denmiyor. İş artık, ''şu ya da bu meslekte çalışıyorum'' denildiği zamanlardaki gibi, bireyin ''özgün varlığı içinde kendisini belirleyen'' bir faaliyetle buluşmasını içermiyor. Çalışmak herhangi bir şeyi yapmadır; şunu ya da bunu, burada ya da şurada hiç önemli değil!*

*Bu ''belirlenmiş içeriğe ilgisizlik'' yine de, Marx'ın öngördüğü gibi, bireyin toplumla tıpkı ''organik bedeni''ymişçesine özdeşleşmesini sağlayamadı; tersine, işin geneline karşı bir soğumaya yol açtı. İş artık yalnızca, şöyle ya da böyle, hızla küçülen ve kişisel olarak kullanılan bir becerinin sahipliğini kendi tekellerinde sürdüren ''profesyoneller'' azınlığı için bir sosyal kimlik ve kişisel oluşum kaynağıdır.*

*Nitekim İsveç'te, çalışmanın hayatlarının en önemli boyutu olduğunu düşünen kadın ve erkeklerin oranı 1955'te yüzde 33'ken, 1977'de yüzde 17'ye düşmüştür. Federal Almanya'da boş zamanlarında yaptıkları şeylere işlerinden daha çok önem verdiklerini söyleyen kişilerin oranı da, 1962'de yüzde 36'yken, 1976'da yüzde 56'ya yükselmiştir. Fransa'da da, 1981 sonbaharında ücretliler arasında gerçekleştirilen bir ankette şu soru yöneltiliyordu: ''İmkân bulabilseydiniz her türlü profesyonel faaliyetten vazgeçer miydiniz?'' Yanıtlar, ''İşe bağlılığın sağlanan gelirle orantılı olarak arttığını'' ortaya koyuyor (ama beri yanda, gelirin ihtisaslaşma ve ücretliye tanınan kişisel sorumlulukla birlikte arttığı da biliniyor). Böylece, ayda 4 bin ile 6 bin Frank arasında kazananların yüzde 61'i yukardaki soruya olumlu karşılık verirken, 10 bin Frank'ın üzerinde ka-*

*zananların yüzde 34.5'inin karşılığı olumludur; ortalamaysa yüzde 44.9'dur.*

*İşe karşı bu soğuma, oluşum halindeki sosyo-kültürel dönüşümler iinde en önemlisidir. Sanayileşmeciliğin ideolojik ve etik temellerini yıkmaktadır. İşe karşı soğuma, mevcut bunalımın en yüksek evresinden önce de belli olmakla birlikte (bu, daha önce gördüğümüz gibi, bunalımın belirleyici bir ögesiydi), otomatizasyonun beraberinde getirdiği, toplumsal açıdan gerekli çalışma zamanının azalışıyla daha da hızlanmıştır. Bu nedenle her yerde,* ***toplumlar, bu azalışı, gerçekliği gözlerden saklayacak bir biçimde kullanırlar.*** *Ne ki hangi ideolojinin bayraktarlığını yaparsa yapsın,* ***bundan böyle herkes için tam günlük ücretli iş olamayacağı ve de ücretlendirilmiş çalışmanın insanların hayatında odak noktası, hatta ağırlıklı faaliyet olarak bile kalamayacağı olgusunu kabul etmeyen her politika yalandır.***

## Bir İşçi Muhafazakârlığı

*Çalışma, bir işe ve bir gelire sahip olma hakkı uzun süre birbirine karıştırıldı. Bundan böyle karıştırılamaz. Bu olgu, işsizliğin ödeneğe bağlanması ve erken emeklilik uygulamalarıyla birlikte kabul edilip, gizlenmeye çalışılıyor. Kuzey Avrupa ülkelerinde işsizlere, eski ücretlerinin yüzde 70'inin süreyle bağlı olmaksızın ödenmesi ya da 55 yaşından yukarı işçilerin, kimi bunalım içindeki sektörlerdeyse 50 yaşından yukarı olanların ücretlerinin yüzde 70 ila 90'ı üzerinden emekliye sevk edilmeleri, bir gelire sahip olma hakkını, bir işe sahip olma hakkından kendiliğinden ayırmaktadır.*

*Yine de bu ayrım, gizli kapaklı bir şekilde gerçekleşmektedir. Bu ayrım, işsizliği rastlantısal ve geçici bir olay gibi, işsizlik ödentisini de bir hak gibi değil bir sadaka gibi değerlendirerek, artık bundan böyle hiç kimse için asla tam gün çalışma olmadığını, olamayacağını kabullenmeyi önlemektedir. İşsizlik de işsizler de, sürekli tam gün çalışma sanki kural ve ilkeymiş ve öyle kalmalıymış gibi ele alınmaktadır: Ya tam gün ücretli olunmalıdır, ya da ödeneği karşılığında, ücretsiz dahi olsa her türlü faaliyetten elini çekmiş işsiz! Tam gün çalışma veya tam işsizlik seçenekleri, toplumsal açıdan gerekli çalışma süresinin olgusal biçimde azalışını zımnen yadsımakta ve bu azalmayı en eşitsiz bir biçimde paylaştırmaktadır. Tam gün ilkesini ve* ***onun aracılığıyla işverenlerle ücretliler arasındaki sosyal ilişkilerin niteliğini*** *korumak üzere, bu azalıştan yararlananları (işsizler) cezalandırıp kenarda bırakmaktadır.*

*İşverenlerle ücretliler arasında, ikincilerin birinciler karşısındaki boyun eğiş ve bağımlılıkları üzerine kurulu bu ilişkiler, gerçekten de ancak, iş ücretlinin başlıca uğraşısıysa sürebilir. Ücretli o zaman, işveren karşısındaki bağımlılığını, hayatı üzerinde etki yapan bir güç olarak duyar; çünkü hayatı tümüyle iş çevresinde örgütlenmiş ve yoğunlaşmıştır. Yok, eğer iş haftada otuz saati ya da daha az bir süreyi aşmazsa o zaman herhangi bir diğer faaliyetten farksız, hatta zaman zaman çok daha önemsiz bir konuma düşecektir. Bundan da işçinin işverenle arasındaki varlıksal boyun eğiş ilişkisi etkilenecektir; işgücü artık işverenin kararlarını ve gücünü uysallıkla kabullenmeyecektir. Tam gün ilkesine sahip çıkılması temelde,* ***verimin etiği üzerine kurulu egemenlik ilişkilerini korumaya*** *yöneliktir.*

*Egemenlik ilişkisinin ideolojik temellerini korumayı amaçlayan bu politikanın yüksek bir toplumsal maliyeti vardır. Ve faal nüfusu, ister istemez iki yönlü bir bölünmeye sürükler:*

*Bir yanda korunan ve istikrarlı, tam gün istihdam edilen, sanayileşmeciliğin geleneksel değerlerinin koruyucusu, işine ve toplumsal statüsüne bağlı bir seçkin-işçiler kesimi; öte yanda bir işsizler kitlesi ve vasıfsız, statüsüz, geçici, öylesine sıradan işlerde çalıştırılan işçiler...*

*Tüm sanayileşmiş toplumlar, istikrarlı işçi sınıfının eski düzenin korunmasında tutucu bir rol oynamaya çağırıldığı, (özellikle Japonya'da her zaman gündemde olan) bu ''ikileşmeye'' doğru yol almaktalar. Buna karşın işi paylaşma, iş saatlerini azaltma, özdenüretim ve faaliyetlerini yaygınlaştırarak ücret düzenini ortadan kaldırma ve herkese yaşamasını güvenceye alacak bir gelir sağlama mücadelesinde, ''mümkün olan'' tek toplumsal özne ''çalışmaya karşı ilgisizler'' kitlesidir. Solla sağ arasındaki sınır, her zaman olduğu gibi açıkça iş dünyasından geçiyor.*

## Yaşayan-Ölü Kapitalizm

*Otomatizasyonun etkisi karşısında, tam gün istihdama imkân veren istikrarlı işlerin sayısının hızla azalması nedeniyle, emek seçkinlerinin egemenliği ancak az ya da çok otoriter yollardan sağlanabilirken, giderek büyüyen çalışmayanlar yığınının toplumsal denetimi, aşağıda özet bir tipolojisi verilen az ya da çok özgün politikaları gerektirir:*

*I. Çalışmayanların ''sınıflandırılması ve toplum dışına itilmeleri'' şimdiden görülebilen eğilimlerdir. İşsizliğin yaygınlığı ve çalışmayanların bir ödeneğe bağlanması, çok sayıdaö nlemle gizlenmektedir: Zorunlu eğitim süresinin uzatılması; hiçbir geleceği olmadığı bilinen eğitim stajları ve yüksek öğretim için burs vb. yollarla para ödenmesi, askerlik süresinin uzatılması; özgür işgücünün ekonomik açıdan verimli bulmayacağı işler için, düşük ücretli yarı askeri bir ''zorunlu hizmet'' getirilmesi; silah üretiminin artması; sınırlı savaşlar, vs...*

*Bu önlemlerin ortak özelliği, kendileri için tam günlük, istikrarlı işler bulunmayan kesimlerin toplumdan dışlanmasıdır. İşbaşındaki rejimlerin yapısı ve politik geleneğine göre, bu ayırım az ya da çok otoriter bir biçim alır: Irk ayrımcılığı, gulaglar, zorunlu yarı askeri hizmet; çoğunluğu işsiz olan halk kitlelerini biraraya toplayan Kuzey Amerika tipi kent gettoları ya da gecekondu semtleri; ve beri yanda da genç işsizler çeteleri; sonu gelmez öğrenim ve eğitimler için belirsiz burslular; vekalet edenler, ek hizmet yapanlar, mevsimlik çalışanlar, vs...*

*Bütün şıklarda düzenli işi olmayan kesimler, çoğunlukta oldukları zaman bile (bazı Amerikan kentlerinde veya Güney Afrika'da olduğu gibi) ''toplum tarafından kenarda bırakılmışlardır''. Her türlü faaliyet ve toplumsal katılma onlara kendiliğinden yasaklanmıştır. Şöyle ya da böyle, bu keşimlerin toplumsal açıdan daha aşağı ve uyumsuz oldukları kabul edilmektedir. Toplum, bu dışlamanın kurbanlarından her gün yaka silker ve onlara vermek lütfunda bulunduğu sadakayı başlarına kakar.*

*Emek seçkinlerinin hegemonyası, aslında hakların eşitliğini alaşağı eden ve hemen hemen ırkçı kinler ve küçümsemelerle kendini doğrulayan bir katmanlaşma pahasına sürüp gider.*

*II. Toplumsal faaliyetin ''iki yönlü katmanlaşması'', dışlamanın teknokratik ve örgütlü bir değişik türüdür. Halkı, kendiliğinden, iki sektörden birine ya da diğerine ait oluşlarına göre ayırır: Yüksek verimin geçerli olduğu ve maliyetleri etkileyen ileri sanayilerle hizmetlerin (özellikle taşımacılık, iletişim ve dağıtım) uluslararası rekabetine açık sektöre ve düşük verimin hüküm sürdüğü, ilk sektörün maliyetleri içine girmeyen, personel hizmet ve bakım-*

*ları, zenaatlar, eğlenceler, dinlence sanayileri, vs. gibi kârlılık ve verim ilkeleri saptamaya gerek olmayan faaliyetlerin sektörüne...*

*İkinci sektör (ki bu buluşun öncüleri alay edercesine ''konuk'' sektör diye adlandırıyorlar) politik açıdan olduğu kadar ekonomik ve ideolojik açılardan da açıkça birincisine bağımlıdır. Emek seçkinleri, üretkenlikleri ve verim etikleri ile hegemonyalarını sürdürmektedir. İstediği kadar azınlıkta olsunlar, bu seçkinler küçük mesleklerin, küçük hizmetlerin ve küçük meta üretimlerinin, kendi gölgesi altında üremesine göz yuman yönetici sınıfı oluşturur. Küçük hizmet ve üretimlerse (Japonya ve İtalya'da olduğu gibi) birinci sektörün esneklik ve gerektiği zaman uyum yeteneği gösteremediği kimi alanlarda, ikinci elden görev yapmaya çağırılabilirler.*

*Guy Aznar'ın deyimiyle bu ''uzamsal'' ikilikçi düzenleme, her şeye rağmen ancak geçici ve kısa süreli olabilir. Gerçekten de, bir yandan emek seçkinleri, sayısal olarak, giderek daha bir daralma eğilimine girecektir. İkinci sektörde sığınak arayan, aynı zamanda umursamaz ve kanaatkâr bir yaşam tarzına sahip kitlenin büyümesi, ister istemez, birinci sektörün meta üretimine sunulmuş, çıkış yollarının daralmasına eşlik edecektir. Öte yandan, kişisel bakım ve hizmet yükümlülerinin çoğalması, maddi olmayan üretimlerin sanayileştirilmesi ve enformatizasyonu eğilimiyle çelişki halindedir. Bu eğilim ancak, ikinci sektördeki halkın büyüyen bir bölümünü, şu ya da bu biçimde, her türlü işten bağımsız bir gelir güvencesi sağlayarak kârlı faaliyetlerden yoksun bırakmakla gerçekleşebilir.*

*III. ''Özdenüretim ve özdengözetim.'' Yönetici sınıf için çözülecek sorun üç yönlüdür:*

*a) Maddi olmayan malların sani üretimi için bir talep yaratmak;*
*b) Enformatikleşmiş ve otomatikleşmiş üretimden dışlanmış halk yığınlarına, bu üretimle rekabete girişemeyecek uğraşılar bulmak;*
*c) Bu uğraşıları, talebi ödeme gücüne erişecek biçimde ücretlendirmek.*

*Bu sorun tek bir yanıt kabul eder: Halk sunulan üretimi tüketmek için ücretlendirilmelidir. Tüketim, ücreti hak eden bir işle özümlenebilen bir uğraş haline gelmelidir. İnsanlar, maddi olmayan mallardaki tüketimlerine göre, bu tü-*

*ketimin aynı zamanda tüketilen malların yönlendirdiği, kişilerin bizzat kendi kendilerini ürettikleri bir üretken faaliyet oluşu ölçüsünde ücret almalıdırlar. Mallar, tüketiciler, tüketim faaliyetiyle, toplumun gereksindiği şekle girebilmeleri için kendi tüketicilerini satın almaktadırlar.*

*''Meta düzeni''nin, bilerek ya da bilmeyerek, başlangıç için gençlere ve yaşlılara uygulama eğiliminde olduğu çözüm işte budur. ''Eğitim ve sağlık tüketimi toplumsal bakımdan yararlı bir iş olarak kabul edildiğinden, çocukların ve yaşlıların ücrete bağlanması bir yardım gibi görülmemektedir'' diyor Jacques Attali.*

*Birinciler, birtakım stajları, eğitim ve öğretimleri tüketmek ve bu yoldan beklentilerinde ve görünür ideolojilerinde, toplumun kendilerinden olmalarını istediği şeye uygun, ''normal'' yurttaşlar gibi özdenüretime göre ücretlendirileceklerdir. Çok geçmeden, bu staj ve eğitimler, enformatikleştirilmiş ve ücretlendirilmiş özdenöğrenim ve özdenyetiştirme programları şeklinde, erişilebilir hale gelecektir. Okul çağından itibaren, çocuklar kendilerine normalleştirici elektronik oyuncakları tüketme imkânını verecek cep harçlığını devlet babadan alacaklardır: Gerçekten de, oyunun içinde varolan ilkelere karşı çıkmak ve onları aşmak uygulamada imkânsız hale getirilmiştir, çünkü bu ilkeler çocuğun, hiçbir zaman otoritesine başkaldıramayacağı bir özne tarafından konmuş gibi görünmektedir. Bu ilkeler nesnelerin dünyasına, mekanizmanın işleyişine aittir. Ve oyun-alet programlayıcılarının ideolojik seçişlerini, sınırlanamayan ''nesnel gereklilikler'' hesabına geçirir.*

*Yaşlılar da öte yandan, toplumsal faaliyetlerden dışlanmalarını kabul etmek için ücretlendirilmişlerdir; ama bu ücretlendirmenin koşulu (kişisel eğilimlerin ve bünyenin dikkate alınmasıyla) ''sağlığın maliyetini önemsizleştirmeye imkân veren bir yaşam profilinin'' kabulüdür. ''Hasta olmak, bundan böyle, yalnızca en uygun profili taklit ederek, kendi kendinin ideal bir kopyasını tüketerek onarılabilecek hem bir günah, hem bir suç ve hem de bir kabahat olarak yaşanır... İnsan her an bir aynanın içindeki, kopya edilecek bir modele, benzerliğin standardizasyonuna doğru gelişeduran, kendi kendisinin uslu ve hoşgörülebilir bir kopyasına bakar.'' ''Ölümlerin ayrıntılı nedenleri, kopya edilmesi gerekenin temel parametreleri ve anormal davranışların sosyal ve bireysel maliyetinin sayısal tahmini, düzenli biçimde yayınlanır... Ekonomik politika burada artık büyük kollektif donanımların yönetilmesinde değil, (...) ama tüm ekonomik mercilerin, sigorta primleri veya yardımlardaki değişmeyle ayarlanmış davranış ilkelerinin açık üretimindedir; sermaye devleti, kişilerden ilkeye (norma) uygunluklarını satın alır, kopyalarını tüketmeleri için insanlara para verir ve benzerliğin talebini yaratır.''*

*Bu model daha başka normalleştirme boyutlarında da geliştirilebilir. Jacques Attali'nin kapitalist toplumun bunalımdan çıkış için seçtiği yöneliş olarak değerlendirdiği modelin, gerçekte kapitalizmle yalnızca ''formel'' benzerlikleri vardır. Vatandaşlara sağlanan gelirler ücret ''görünümünü'' alır; tüketilecek ürünler mal ''görünümünü'' kazanırlar; ama bu görüntüler içi boş görüntülerdir. Korunmuş olan,* ***kapitalist sistem değil, ama kapitalizmin başlıca araçları ücret ve pazar olan egemenlik sistemidir.*** *Gerçekten de üretimin amacı sermaye birikimi ve birikimin değerlendirilmesi değildir ve bundan böyle de olamaz. Şimdi onun birincil hedefi, toplumun denetimi ve topluma egemen olunmasıdır. Sunulan ürünler alışverişin ve kârın -tüketicilerin tüketmek üzere ücretlendirildiği ve üreticilerin azınlıktaki bir tabaka oluşturdukları bir toplumda anlamını yitiren bir kavram- maksimizasyonu gözetilerek değil, ama denetimin ve yönlendirmenin maksimizasyonu gözetilerek sunulmaktadır. Bu ürünler esas olarak, gücü artık mülkiyetin değil, ama denetim mekanizmasının denetimi üzerine kurulu bir yönetici sınıfının ellerinde, iktidar araçları haline gelir. Üretim mekanizmasıyla denetim mekanizması bir ve aynı şey olup çıkar. Artık kapitalist burjuva bir toplumdan çok, neredeyse askeri bir hiyerarşi içindeki bir teknokrasiyle yönetilen totaliter bir topluma daha çok yaklaşılmıştır. Alain Touraine'in teknokrasi tanımı, özünü yitirirken kapitalizmin görünümlerini kurtaran bu topluma mükemmelen uyuyor: Sözkonusu olan amaca uygunluğunun yitip gittiği ve yöneticilerinin efendi değil hizmetkar oldukları bir teknik sistemin egemenliğidir.*

Andre Gorz, Cennetin Yolları, Çev. Turhan Ilgaz,
AFA Yayınları, 1985

# SÖZLÜK

## A

**Açma (Open):** Bant ya da disket üzerinde bir kütüğün yazılma ya da okunmaya hazırlanması.

**Adres (Address):** 1. Bir komutun herhangi bir (sabit ya da değişken) değerin konumunu bildiren kısmı. 2. Herhangi bir bellek bölgesinin konumunun belirlenmesi. Bir adres genelde 8 bitten oluşmakla birlikte, daha büyük makinelerde 16 ve 32 bitlik adresler de kullanılmaktadır.

**Adres Sicili (Address Register):** Yürütülmekte olan komutun adresini içeren sicil.

**Akış Şeması (Flowchart):** Programlanacak problemin çözümlenmesini izleyen aşama. Bu aşama yapılması istenen bütün işlemler, sıraları gözetilerek grafik simgelerle gösterilir.

**Akustik Eşleştirici (Acoustic Coupler):** Sayısal bir sinyalin, normal telefon cihaz ve hatları kullanılarak iletilmesini mümkün kılan bir veri iletişim aracı.

**Alan (Field):** Herhangi bir kaydın, hangi bilgilerin girileceği ve bazen uzunluğunun da önceden belirlenmiş olduğu bir bölgesi.

**ALGOL (ALGOrithmic Language-Algoritmik Dil):** Bir bilgisayar dili. İki versiyonu vardır ALGOL-60 ve ALGOL-68. İkincisi birincinin, yapılandırılmış programlama yetenekleri içerecek şekilde geliştirilmiş biçimidir. Her ikisi de özellikle özel algoritmaların kodlanmasında kullanılır.

**Algoritma:** Belli bir problem türünü çözmek için gereken işlemler sırasını belirleyen bir sınırlı kurallar bütünü.

**Alt-program (Subprogram):** Kendi içinde bütün bir program olmamakla birlikte, bağımsız bir işi görebilen bir komutlar bütünü. Alt-programların kullanılabilmesi için bir program tarafından çağrılabilmeleri gerekir. Alt-programlar, rutin, yordam ve prosedür adıyla da anılabilir.

**Ana Bellek (Main Store):** Merkezi işlem birimine ilişkin olan bellek. Bazen *iç bellek* (internal store), *dolaysız erişimli bellek* (immediate access memory) ya da *birincil bellek* (primary memory) gibi adlarla da anılır.

**Anahtar (Key):** Aranmakta olan veri kaydını tanımlamaya yarayan alan ya da bilgi parçası (örneğin bir adresler kütüğü için anahtar soyadı olabilir).

**APL (A Programmanin Language-Bir Programlama Dili):** Özel bir karakter seti kullanan bir bilgisayar dili. İleri matematik uygulamalar için özellikle kullanışlıdır.

**Ara Bellek (Tampon, Buffer):** Bir çevresel gereç ile merkezi işlem birimi arasında alışverişi yapılmakta olunan verilerin geçici olarak saklandığı bellek bölgesi. Bu çevresel gereçle merkezi işlem birimi arasında hız farklarını telafi eder.

**Arabirim (Interface):** Farklı işlemciler ve çevresel gereçlerin çalışma tarzları arasındaki farkları telafi etmek üzere kullanılan donanım ve yazılımlar.

**Araya Girme (Interrupt):** Bir çevresel gereçten gelen ve halen yürütülmekte olan yordam ya da alt-programın durdurularak ikinci bir işlemin yürütülmesini mümkün kılan sinyal. Bu ikinci işlem tamamlandıktan sonra bilgisayar ara verdiği yordam ya da alt-programa geri döneri.

**Arıza (Bug):** Programdaki bir hata ya da

donanımda bir kusur.

**Aritmetik İşlemci (Arithmetic Operator):** Yapılması istenen aritmetik işlemin türünü belirten simge (örneğin +, —, *, /).

**Aritmetik Mantık Birimi (Arithmetic Logic Unit):** Merkezi işlem biriminin aritmetik ve mantık işlemlerini yapmaktan sorumlu olan bölümü.

**Aritmetik Sicili (Arithmetic Register):** Halen yürütülmekte olan işleme ait değer ve sonuçların geçici olarak saklandıkları özel bellek bölgesi.

## B

**Bağdaştırıcı (Adaptör, Adaptor):** Farklı bağlantıları olan aygıtlar arasında bağlantı kurmayı sağlayan mekanik arabirim. Bkz. Çevresel Gereçler.

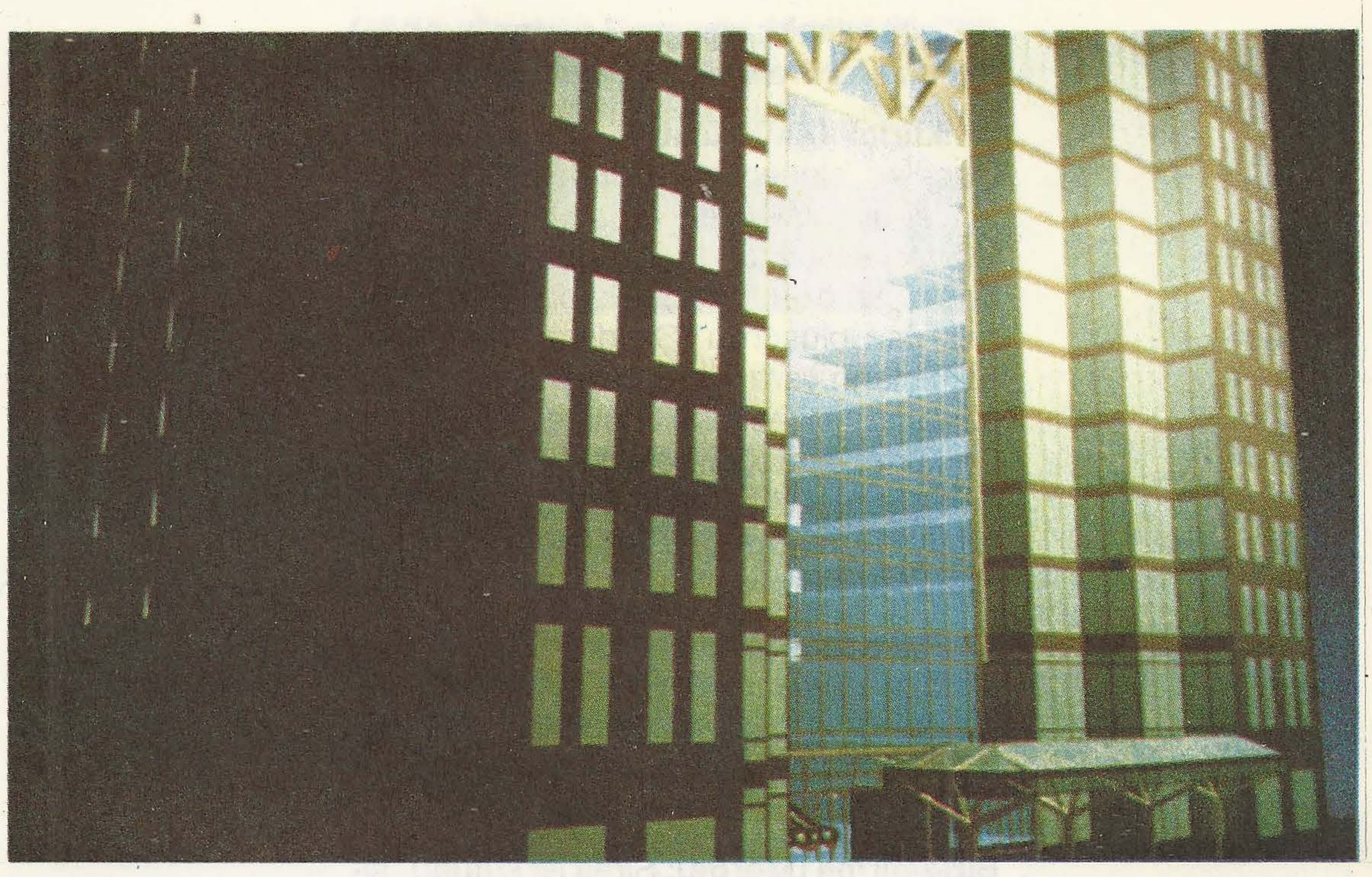
**BASIC (Beginners' All-Purpose Symbolic Instruction Code-Başlayanlar İçin Genel Amaçlı Simgesel Komutlar Kodu):** Kişisel bilgisayarlarda en yaygın olarak kullanılan bilgisayar dili. Hem çeviricili, hem derleyicili çeşitli versiyonları mevcuttur.

**Baud:** Telgraf ve telefon hatlarında veri iletim hızını ölçmek için kullanılan bir birim. Bir baud genellikle saniyede bir bite tekabül eder.

**Bayt (Byte):** Sekiz bitten oluşan bir bilgi birimi. Genellikle her karakter bir bayt tarafından temsil edilir. Bkz. Bit, Karakter.

**Bellek (Memory):** Bilgisayarın veri ve komutların saklandığı bileşeni.

**Benzeşik Bilgisayar (Analog Computer):** Bilgileri temsil etmek için sürekli değişen



fiziki niceliklerin aldığı değişik değerleri (örneğin voltaj değerini) kullanan bilgisayarlar.

**Benzetleme (Simulation):** Herhangi bir sistemin ve işleyişinin bir modelinin bilgisayar kullanılarak oluşturulması ve bu yolla sözkonusu sistemin davranışlarının ve işleyiş ilkelerinin incelenmesi.

**Beşinci Kuşak Bilgisayarlar:** 1990'ların başından başlayarak kullanılması öngörülen bilgisayarlar. Doğal dillerle ve insan sesiyle komut kabul edebilecekleri bekleniyor.

**Bilgisayar Ağı (Şebeke; Network):** Birbirlerinin bilgi işlem ya da depolama yeteneklerinden yararlanmak üzere birbirlerine bağlanmış bir bilgisayar sistemleri dizisi. Zaman zaman sistemler birbirlerinden çok uzak mesafelerde olabilir. ''Yıldız'', ''Dai-

re'' gibi terimler ağın nasıl düzenlenmiş olduğunu tarif etmek için kullanılır.

**Biriktirici (Akümülatör, Accumulator):** Önceleri işlenenler üzerinde aritmetik işlemler uygulayan elektronik bir gereç adı olarak kullanılırdı. Bu genellikle Aritmetik Birimi'nin özel bir siciliyle ona bağlı devrelerden oluşurdu. Şimdilerde aritmetik sonuçların yaratıldığı herhangi bir değer saklama bölgesi biriktirici adı ile anılıyor. Biriktirici bir değer saklar, yeni bir değer geldiğinde ise iki değerin toplamını yaratarak bu yeni değeri saklamaya başlar. Bkz. Aritmetik ve Mantık Birimi.

**Birinci Kuşak Bilgisayarlar:** Henüz transistörlerin kullanılmaya başlanmadığı bilgisayarlar. Bunlar arasında EDSAC, EDVAC ve ENIAC sayılabilir. İlk ticari ürünler ise LEO ve Univac 1'dir.

**Birleştirme (Merge):** Sıralı iki kütüğü, sıralı tek bir kütük halinde birleştirme.

**Bit:** En küçük veri birimi. Yani ikili sayı tabanında her bir rakamın -1 ya da 0'ın- temsil ettiği bilgi birimi. Bazı bilgisayarlarda her bitin teker teker adreslenmesi mümkündür.

## C

**CPS (Characters Per Second; Saniyede Karakter Sayısı):** İki donanım gereci arasındaki veri iletim hızını ölçmek için kullanılan bir birim. Bkz. Baud.

**COBOL (COmmon Business Oriented Language - Sıradan İşletmelere Yönelik Dil):** Özellikle sınıflandırma ve metin-işlem yetenekleri yüksek, işletmeler için geliştirilmiş bilgisayar dili.

**CP/M (Control Program Monitör):** Z80 mikroişlemci yongasını kullanan kişisel bilgisayarlarda kullanılan işletim sistemi.

# Ç

**Çentik (Flag):** Ancak iki değerden birini alabilen bir değişken ya da gösterge. Durumu bir koşulun durumunu, örneğin bir listenin sıralanmış olup olmadığını, bir araya girme sinyalinin alınıp alınmadığını gösterir.

**Çevirici (Interpreter):** Yüksek düzeyli bir dilde yazılmış bir programın komutlarını teker teker, düşük düzeyli bir dile ya da makine diline çeviren program.

**Çevresel Gereç (Peripheral Device):** Merkezi işlem birimine bağlanabilecek ek bellek birimlerini, giriş ve çıkış gereçlerini kasdetmek için kullanılan genel bir terim.

**Çevrim Zamanı (Cycle Time):** Belli bir donanım işleminin başlangıcı ve yeniden başlangıcı arasında geçen süre.

**Çıkış Gereci (Output Device):** Bilgisayardan gelen sinyalleri insanların okuyabileceği ya da daha sonra bilgisayarın yeniden işleyebileceği bir biçime çeviren çevresel gereç.

**Çift Yönlü Veri Aktarımı (Duplex Data Transmission):** İki nokta arasındaki verilerin aynı anda her iki yönde birden aktarılması.

**Çizici (Plotter):** Kâğıt üzerine çizgiler çizmeye yarayan bir çıkış gereci.

**Çok İşlemcili Sistemler (Multiprocessor Systems):** Birden fazla minikişlemci içe-

ren sistemler. En önemli özellikleri paralel veri işlemi mümkün kılmalarıdır.

**Çoklu Erişim (Multi-Access):** Bilgisayarı birden fazla kullanıcının, görünürde aynı anda kullanmasını mümkün kılan erişim tarzı. Özellikle zaman paylaşımlı sistemlerde etkileşimli olarak çalışmak için kullanılır. Bkz. Zaman paylaşımı, Karşılıklı Mod.

## D

**Dallanma (Branch):** Programın normal sıralı işleyişinden bir sapma. Dallanmalar koşullu ya da koşulsuz olabilir.

**DBE-Doğrudan Bellek Erişimi (DMA-Direct Memory Access):** Bir çevresel gereçten belleğe, merkezi işlem biriminden geçmeden veri aktarma. Bkz. Merkezi İşlem Birimi.

**Derleme Hatası (Compilation Error):** Derleme aşamasında ortaya çıkan bir hata (örneğin sözdizin hatası-syntax error).

**Derleyici (Compiler):** Yüksek düzeyli bir dilde yazılmış bir programı düşük düzeyli bir dile ya da makine diline çeviren bir program. Yüksek düzeyli dildeki her komut birden fazla makine dilinde komuta çevrilir. Bu şekilde, birincisinden bağımsız, yürütülebilir yeni bir program yaratılmış olur.

**Disk Düzenlemesi (Disk Formatting):** Herhangi bir disk ya da disketin okuma ve yazma işlemlerini mümkün kılmak üzere, kanal ve kesim numaraları gibi güdüm bilgileri verilerek hazırlanması.

**Dizey (Array):** Aynı ad ve adresle erişile-



bilecek bir dizi bellek bölgesi. Bu bölgedeki her unsur (yani her erişilmek istenen her değer) dizeyin adı ve dizey göstergeleri (subscript) ile birlikte gösterilir. Örneğin tek boyutlu SINIF dizeyinin sekizinci elemanına SINIF(8) adresiyle, iki boyutlu OKUL dizeyinde herhangi bir değere ise

OKUL(x,y) adresiyle erişilr.

**Doğrudan Erişim (Direct Access):** Herhangi bir veriye doğrudan ulaşma olanağını veren erişim yöntemi. Bir kütüğün herhangi bir kaydı, diğer kayıtlardan bağımsız olarak okunabiliyorsa, o kütük doğrudan (veya ''rastgele'') erişimlidir. Bkz. Erişim, Rastgele erişim.

**Dolaylı Adresleme (Indirect Addressing):** Kendisi de başka bir adres içeren bir konumun gösterildiği adresleme yöntemi. Aranılan veri bu tarzda gösterilen ikinci adreste yeralır. Bkz. Adres.

**DOS (Disk Operating System-Disk işletim Sistemi):** Bir işletim sisteminin diskler üzerindeki kütük ve programların işletimi ile ilgili olan bölümü.

**Döküm (Listing):** Bir programdaki komut, veri ve yorumların sıraları korunarak yapılmış bilgiyazımı.

**Döngü (Loop):** Belli bir koşul doğrulanıncaya kadar tekrarlanan komutlar dizisi.

**Dönüşlü Alt-program (Recursive Subprogram):** Komutları arasında kendi kendisini çağıran bir komut içeren bir alt-program.

**Dördüncü Kuşak Bilgisayarlar:** Halen kullanılmakta olan bilgisayarlar. Özellikleri yonga (chip) teknolojisinin kullanılması, düşük bellek maliyetleri ve az sayıda işletim sisteminin yaygın olarak kullanılmasıdır.

**Düşük Düzeyli Dil (Low Level Language):** Her komutu bir makine dili komutuna tekabül eden makineye yönelik bir dil.

**Düzenleyici (Editor):** Kullanıcının veri ve programlarını gözden geçirip düzeltmesini sağlayan program.

# E

**Elde Çentiği (Carry Flag):** İkili toplama sırasında bir elde hasıl olduğu zaman 1 değerini alan bir bit. Kullanıcı elde çentiğinin durumunu sınayabilir.

EOF (End of File-Kütük Sonu): Özel bir güdüm karakteri. Bir kütükte son kayıttan sonra, kütüğün bittiğini işaretlemek için kullanılır. Bkz. Kütük.

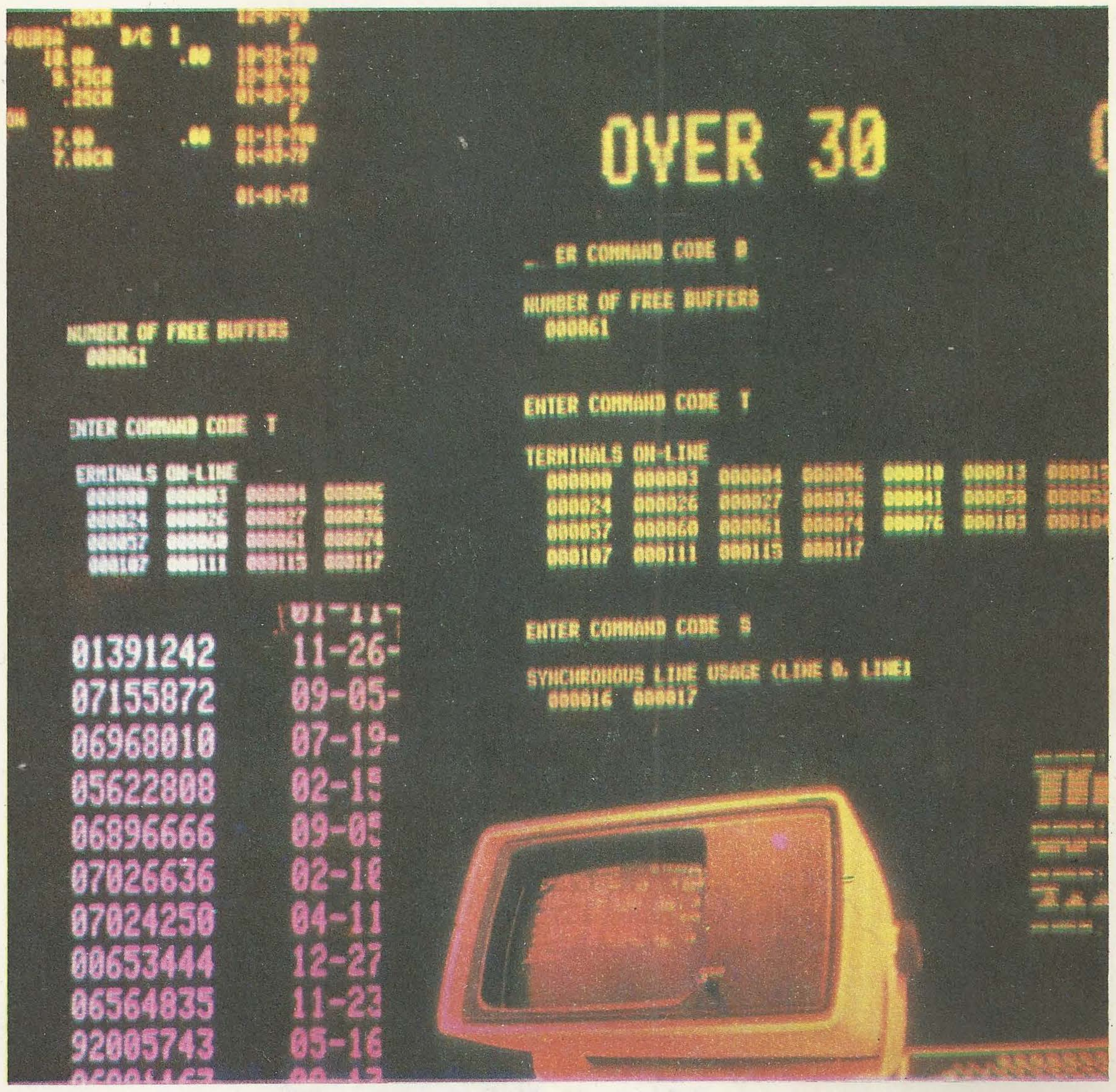
**EPROM (Erasable PROM-Silinebilir PROM):** Özel bir PROM türü. Buraya yerleştirilen programlar özel bir süreçle (morötesi ışınlara tabi tutularak) silinebilir. Bkz. ROM, PROM.

**Erişim (Access):** Bir bellek ünitesine (disk, disket, RAM ya da ROM) giriş/çıkış işlemi. İki aşamada gerçekleşir: Arama ve yazma veya okuma. İlk başlaması normal olmayan disketlere erişimde bu aşamalara bir yenisi eklenir: Uygun dönüş hızına ulaşıncaya kadar geçen bekleme süresi. Bu ise toplam erişim süresini uzatır. Bkz. Doğrudan Erişim, Rastgele Erişim, Seri Erişim, Sıralı Erişim.

Erişim Zamanı (Access Time): Bir erişim talebi ile verinin kullanılabilir hale geldiği an arasında geçen süre.

**Eşzamanlı Veri Aktarımı (Synchronous Data Transmission):** Kesintisiz olarak işlemekte olan ve aynı saat tarafından kontrol edilen iki gereç arasında veri aktarımı.

**Eşzamansız Veri Aktarımı (Asynchronous DataT ransmission):** Bir karakterin aktarım sürecinin bitiminin diğer karakterin aktarımının başlamasını sağladığı veri aktarım tarzı.




**F**

**FORTRAN (FORmula TRANslator-Formül Çeviricisi):** Bilimsel ve istatistiki amaçlarla yaygın olarak kullanılan bilgisayar dili. Çeşitli versiyonları mevcuttur.

# G

**Gerçek Zamanlı Sistemler (Real-Time Systems):** Dış bir kaynaktan sürekli değişen verileri alıp bunları yeni veriler gelene kadar hızlı bir biçimde işleyebilen bilgisayar sistemleri.

**Gıyabi Değer (Default Option):** Herhangi bir değer ya da komutun verilmemesi halinde bilgisayarın otomatik olarak izleyeceği yol.

**Giriş Gereci (Input Device):** Makinenin okuyabileceği bir düzende girilmiş verileri kabul eden ve merkezi işlem birimini elektrik sinyalleri halinde gönderen bir çevresel gereç.

**Giriş/Çıkış Gereci (Input/Output Device):** Aynı zamanda hem giriş, hem çıkış birimi olarak kullanılabilen bir çevresel gereç. Bazen giriş/çıkış gereçleri aynı kabinde muhafaza edilen aslında iki ayrı gereçten oluşabilir.

**Güdüm Karakteri (Control Character):** Bilgiyazımı yapılmayan bir karakter. Klavyeden girildiği ya da bir çevresel gereće gönderildiği zaman, veri olarak değil, çeşitli fonksiyonların işleyişini kontrol eden bir sinyal olarak yorumlanır.

# H

**Harekete Geçirme (Aktive Etme, Activate):** Bir yordam veya alt-programı harekete geçiren işlem. Bkz. Alt-program, Yordam.

**Harfsayısal Veri (Alphanumeric Data):** Genellikle alfabenin karakterlerinden ve

rakamlardan oluşan karakter seti. Bazen bu set daktilolarda bulunan diğer işaretleri de (noktalama vb.) içerir.

**Hata (Error):** Programın istenen sonucu gerçekleştirmesini engelleyen bir özelliği.

## İ

**İkili Sayı Sistemi (Binary Notation):** Tabanı iki olan, yani yalnızca 0 ve 1 rakamlarını kullanan sayı sistemi.

**İkinci Kuşak Bilgisayarlar:** İlk kez transistör kullanılarak yapılan bilgisayarlar. Transistörlerin kullanılması işlem hacmi ve hızını artırdığı gibi, bilgisayarı işletmek için gereken güç miktarını da hatırı sayılır ölçüde azalttı. En önemli ikinci kuşak bilgisayarlardan biri Atlas'tı.

**İlkdeğerleme (Sıfırlama, Initialise):** Sayaç ve değişkenlere belli bir ilk değerin -çoğunlukla sıfır- verilmesi. Bu, genellikle bir program ya da alt-programın başında yapılır.

**İmleç (Cursor):** Ekranda o anda kullanıcının erişebileceği konumu gösteren ve genellikle yanıp sönen işaret. Bazı bilgisayarlarda imleç adreslenebilir, yani kullanıcı koordinatlarını vererek, imlecin ekranın istediği bir konumuna gitmesini sağlayabilir.

## K

**Kabarcık Bellek (Bubble Memory):** Bilgilerin çok küçük zerreciklerin manyetize edilmesiyle saklandığı bellekler. Bu çok

yüksek hacimlerdeki bilgilerin çok küçük alanlarda saklanmasını mümkün kılar.

**Kanal (Track):** Herhangi bir manyetik bant ya da diskte üzerine verilerin yerleştirildiği bir yol. Bantlarda bandın kenarına paralel bir dizi kanal bulunur. Disk ve disketlerde ise kanallar içiçe geçmiş halkalar halindedir.

**Kapama (Close):** Bir disket ya da manyetik bant üzerindeki bir kütüğe erişim işlemleri tamamlandığında yapılması gereken işlem. Bu kullanıcının görmediği bir dizi işlemin -örneğin, ara belleğin boşaltılması- yapılmasını sağlar.

**Kapı (Port):** Terminalden gelen sinyallerin merkez bilgisayara girdiği nokta.

**Kaplumbağa (Turtle):** LOGO ve ondan türemiş dillerde kullanılan bir çizim gereci. Bu yerde hareket edebilen elektro-mekanik bir gereç olabileceği gibi, böyle bir gerecin hareketlerini benzetleyen ekrandaki bir figür de olabilir.

**Karakter (Character):** Bilgisayarda temsil edilebilecek herhangi bir simge. Örneğin, rakamlar, harfler, boşluk, noktalama işaretleri vb.

**Karakter Kodu (Character Code):** Alfabetik ve diğer karakterleri ikili sayı sisteminde temsil eden kodlar. Farklı imalatçılar bu amaçla farklı kodlar kullanabilmektedir. Yine de sistemler arasında bilgi alışverişini kolaylaştırmak amacıyla belli bazı standart kodlara yaygınlık kazandırılmıştır. Bunların en önemlileri şunlardır: ASCII - Bilgi Değiştokuşu için Amerikan Standard Kodu (American Standard Code for Information Interchange); ISO7-7 Bitlik Kodun Standartlaştırılması için Uluslararası Örgüt (International Organisation for 7-bit Code); EBCDIC-Ondalıklı Değiştoku-

şu için Geliştirilmiş İkili Kod (Extended Binary Code for Decimal Interchange).

**Kayıt (Record):** Tek bir birim halinde işlem gören birbirleriyle ilintili veriler. Bu mantıki kayıttır. Bellekte saklanan fiziki kayıt, mantıki kayda ek olarak, bazı güdüm karakterleri de içerebilir.

**Kesim (Sector):** Herhangi bir manyetik bant ya da diskin adreslenebilir en küçük bölümü.

**Karşılıklı Mod (Interactive Mode, Conversational Mode):** Kullanıcının bilgisayarla kesintisiz iletişimde bulunabildiği ve komutlarına derhal karşılık alabildiği işleyiş tarzı.

**Kısmi Çift Yönlü Veri Aktarımı (Half Duplex Data Transimssion):** İki nokta arasında her iki yönde birden veri iletilmesini sağlayan veri aktarım tarzı. Ancak bu tarzda veriler her iki yönde birden aynı anda aktarılamaz.

**Kontrol Birimi (Control Unit):** Merkezi işlem biriminin program komutlarının yürütülmesini denetlemekten sorumlu bileşeni.

**Kütük (File):** Birbirleriyle ilişkili kayıtların belli bir düzen içerisinde birarada saklandığı mantıki bellek bölgesi.

## L

**LISP (LISt Processing-Liste İşlem):** Özellikle yapay zeka araştırmalarında kullanılan, liste işlem yeteneklerine sahip bilgisayar dili.

**Liste (List):** Sıralı olarak düzenlenmiş bir veriler yapısı (örneğin bir boyutlu bir dizey). Bağlı bir listede (linked list) hem ka-



1983

yıt hem veriyi, hem de bir sonraki veriyi gösteren bir gösterge içerir.

**LOGO:** Öncelikle çocuklara programlama mantığı öğretmek üzere geliştirilmiş, liste işlem yeteneklerine sahip bilgisayar dili. Kaplumbağa grafikleri ile ünlüdür.

## M

**Mantık Hatası (Logical Error):** Programın tasarımında bir hata (örneğin yürütülemeyecek bir komuta dallanma).

**Merkezi İşlem Birimi (MİB; Central Processing Unit-CPU): Ana bellek, aritmetik/mantık birimi ve kontrol biriminden oluşan, bilgisayarın temel bileşeni.**

**Metin-İşlem (Word Processing):** Yazı ve belge yazma, düzeltme, düzenleme, saklama ve basılı kopyalarını elde etme amacıyla geliştirilmiş yazılım paketi.

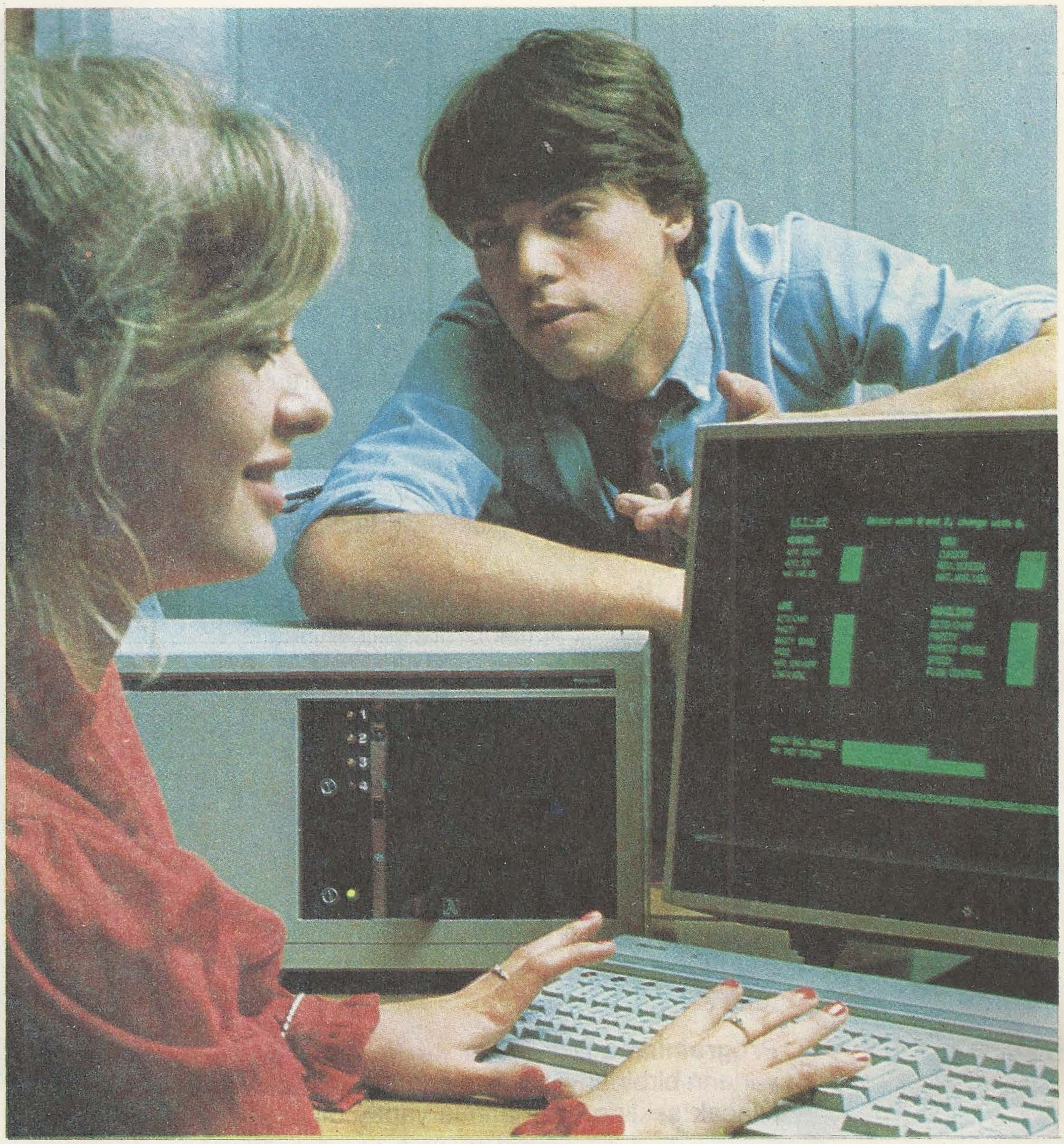
**Modem:** Sayısal sinyallerin benzeşik (analog) elektrik darbelerine çevrilmesini ve tersini sağlayan aygıt. Bu, telefon bağlantılarından yararlanmaya olanak verir.

## O

**Optik Lif (Fiber Optik, Optical Fibre):** Işık sinyallerinden yararlanarak çok yüksek hızlarda veri aktarımını mümkün kılan çok ince cam hat.

## Ö

**Öbek (Block):** Veri aktarımı sırasında ya



da bellekten veri alınırken tek bir birim olarak işlem gören kayıtlar bütünü.

**Örtüşme (Overlay):** Kapsamlı bir programın sadece işlenecek bölümlerinin ana belleğe alınarak parça parça işlenmesi.

# P

**Paket Aktarımı (Packet Switching):** Bilgisayar ağlarında otomatik iletişim için kullanılan bir yöntem. Her veri ''paket''i nereye yollandığını bildiren bir kod taşır.

**Paralel Veri Aktarımı (Parallel Data Transmission):** Veri bitlerinin farklı hatlar üzerinden aynı anda iletildiği veri aktarım tarzı.

**Parametre:** Bir ana programın alt-programa ya da bir alt-programın ana programa sağladığı değişken ya da değerler.

**PASCAL:** Özellikle yapılandırılmış programlamayı teşvik etmek üzere geliştirilmiş bilgisayar dili.

**Pencere (Window):** Belli bir program ya da program parçası ile ekranın bir bölümünün özel bir işlev için ayrılması.

**Piksel (Ekran Noktası, Pixel):** Herhangi bir görüntü biriminde (ekranda) kontrol edilebilecek en küçük ekran noktası.

**Program:** belli bir algoritmayı özgülleştirmek üzere yapılandırılmış bir komutlar bütünü.

**Programlama Dili:** İnsanların ve bilgisayarların birbirleriyle iletişmelerini mümkün kılacak şekilde hazırlanmış yapay dil.

**PROLOG (PROgramming in LOGic-Mantıkta Programlama):** Matematiksel mantık ve yüklemler mantığı temel alınarak geliştirilmiş bilgisayar dili. Beşinci Kuşak araştırmalarında özellikle kullanılan dil.

**PROM (PROgrammable Memory-Prog-**

**ramlanabilir ROM):** Bir ROM türü. İmalattan sonra da buraya kullanıcı bir program yerleştirebilir, ancak bir kere yerleştirdikten sonra, o program orada sabit kalır.

**Protokol:** Veri aktarımı sırasında, aktarılan verilerin düzenlerinin, veri aktarımının başlangıcını, bitimini bildiren sinyallerin neler olacağını belirleyen, üzerinde anlaşılmış işlemsel prosedürler.

## R

**RAM (Random Access Memory-Yaz-Oku Bellek):** Programcı tarafından okunabilen ve yazılabilen bellektir. Genellikle bir yonga üzerinde bulunur. İki türü vardır: Dinamik RAM'lerde bilgilerin belli fasılalarla (2 milisaniyede bir) yeniden yazılması, yani yenilenmesi gerekir. Statik RAM'larda ise böyle bir gereklilik yoktur.

**Rastgele Erişim (Random Access):** Belleğin herhangi bir kesimine kayıtlı bir kaydı okuma ya da yazmak için kullanılan yöntemlerden biri. Bu yöntem bir sonraki adresi önceden bilme olanağı vermez. Bir RAM bellek bölgesine ya da bir kütüğe rastgele erişimle ulaşılabilir. Bkz. Erişim, Doğrudan Erişim.

**Rastlantısal Sayılar (Random Numbers):** Aralarında belli, kestirilebilir bir ilişki olmayan, birarada bir örüntü (pattern) oluşturmayan sayılar dizisi. Genellikle bilgisayar dillerinde raslantısal sayılar üretmeye yarayan bir komut bulunur.

**RETURN:** Kontrolün bir alt-programdan onu çağırmış olan ana programa aktarılmasını sağlayan bir komut.

**ROM (Read Only Memory-Salt Okunur Bellek):** Programcının sadece okuyabileceği (yazamayacağı) bellek bölgesi. Buradaki yazılım buraya imalat sırasında yerleştirilir.

## S

**Sabit Noktalı Yazı Düzeni (Fixed-Point Notation):** Bu yazı düzeninde sayılar bir dizi basamak olarak gösterilir ve ondalık noktası doğru yerinde bulunur (örneğin 163.485). İşlemler sırasında program ve programcı da ondalık noktasının yerini sabit tutar. Bu yazı düzeni ile temsil edilebilecek sayıların büyüklüğü, kullanılmakta olan bilgisayarın yapısı tarafından sınırlandırılmıştır. Fakat işlemler çok hızlı olduğundan, genellikle ticari uygulamalarda sabit noktalı yazı düzeni tercih edilir. Bkz. Yüzer Noktalı Yazı Düzeni.

**Sayısal Bilgisayar (Digital Computer):** Bilgileri 1 ve 0'la gösterilen elektrik darbelerinin varlığı ya da yokluğu ile temsil eden bilgisayar.

**Sayısal Kontrol (Numeric Control):** Fabrikalarda kullanılan otomatik tornalar, tezgâhlar, bandlar gibi gereçlerin bilgisayarlar tarafından kontrolü. Bazı sayısal kontrol makineleri ancak verili bir programa göre, önceden belirlenmiş işlemleri yapabilir. Daha gelişkin olanları ise duyargalar (sensor) aracılığıyla edindikleri geribildirimden yararlanabilir.

**Seri Erişim (Serial Access):** Genelde sıralı erişimle aynı anlamda kullanılır. Herhangi bir veriye ancak ondan önceki bütün veriler ele alındıktan sonra ulaşılabi-



lir. Örneğin sıralı bir kütükte bir kayıt, ancak ondan önceki bütün kayıtlar ele alındıktan sonra yazılabilir veya okunabilir. Bkz. Erişim, Sıralı Erişim.

**Sıçrama (Jump):** Bkz. Dallanma.

**Sınama Biti (Control Bit):** Veri aktarımı sırasında gönderilmiş olan verinin doğruluğunu sınamak için kullanılan, veriyi temsil eden bayta eklenen ek bir bit.

**Sıralama (Sort):** Herhangi bir veri kütüğündeki kayıtların, önceden belirlenen bir kurala göre düzenlenmesi.

**Sıralı Erişim (Sequential Access):** Disk, manyetik şerit vb. gibi ortamlarda normal kayıt sırasını izleyerek okuma/yazma yöntemi. Yazma işlemi sırasına her kayıt bir öncekinden sonraki fiziksel konuma yazılır. Okuma işlemi sırasında da veri çıkış sırası, normal fiziksel sırayı izler. Yani doğrudan (rastgele) erişimin tersidir. Bkz. Erişim, Seri Erişim.

**Sıralı Veri Aktarımı (Serial Data Transmission):** Veri bitlerinin aynı hat üzerinde teker teker aktarıldığı veri aktarım tarzı.

**Sibernetik (Cybernetics):** İnsan davranışlarının kimi özelliklerine, özellikle de geribildirimden (feed-back) yararlanma yeteneğine sahip sistemlerin incelenmesi.

**Sicil (Register):** Bazen korunmalı olabilen ve özel amaçlarla kullanılan bir bellek bölgesi.

**Sistem Analizi (Systemes Analysis):** Bilgisayarlaştıracak bir işin gereklerinin, bilgisayar uygulamasının mali olurluğunun, mevcut donanım ve yazılım olanaklarıyla ilişkilendirilerek incelenmesi.

**Sonsuz Döngü (Infinite Loop):** Programın durdurulması dışında bir çıkışı olmayan bir döngü.

**Sözdizin (Syntax):** Belli bir programlama dilinin unsurlarını (örneğin karakterleri) bir komut oluşturmak üzere nasıl biraraya getirilebileceğini belirleyen kurallar.

# T

**Taşıt (Bus):** Bilgisayarın bir dizi bileşeninden gelen sinyallerin ortak olarak paylaştığı bir hat (örneğin bütün giriş/çıkış birimlerinden gelen sinyaller G/Ç taşıtını kullanacaktır).

**Taşma (Overflow):** Aritmetik işlemlerin sonucu bilgisayarda saklanamayacak kadar büyük bir sayı verdiği zaman meydana gelen hata.

**Taşma Çentiği (Overflow Flag):** Aritmetik işlemler sırasında herhangi bir taşma meydana geldiği zaman 1 değerini alan bir bit. Durumu kullanıcı tarafından sınanabilir.

**Tek Yönlü Veri Aktarımı (Simplex Data Transmission):** Verilerin ancak tek yönde aktarılabildiği veri aktarım tarzı.

**Terminal:** Uzak bir konumdan bilgisayarla iletişim kurmayı mümkün kılan herhangi bir giriş/çıkış birimi.

**Tokalaşma (Handshake):** Herhangi bir nedenle sinyal alışverişinde bulunacak iki gerecin iletişim kurmak üzere birbirlerine sinyal yollamaları.

**Tümleme (Complementation):** Negatif sayıları göstermede kullanılan bir yöntem. Bu yöntemde orijinal sayıdaki her 1 yerine 0, her 0 yerine 1 yazılır. Örneğin, 10 tabanlı 43'ün iki tabanlı yazılışı 0010100 11'dir. Tümleşiği ise 11010100'dır. Tümleme ikili tabanda çıkarma yapılırken kullanılır.

# U

**Uzman Sistem (Expert System):** Yapılandırılmış bilgilerden oluşan belli bir alanda, gerçek bir uzmanın sahip olduğu bilgilere sahip ve gerçek bir uzman gibi akıl yürüten bir bilgisayar sistemi.

# Ü

**Üçüncü Kuşak Bilgisayarlar:** İlk kez tümleşik devre kullanarak yapılan bilgisayarlar. Tümleşik devre kullanımının en önemli sonucu, bilgi-işlem hız ve hacminin artması yanında, bilgisayar serilerinin (IBM 360'lar, ICL 1900'ler vb.) yapılabilmesiydi.

# V

**Veri Aktarımı (Veri İletimi, Data Transmission):** Bir veri bağını kullanma süreci. Bkz. Veri Hattı.

**Veri Bağı (Data Link):** Herhangi iki nokta arasında sayısal olarak kodlanmış verilerin aktarımını sağlayan iletişim devresi.

**Veri Bankası (Data Bank):** Büyük veri kütükleri ya da veri tabanları kümesi.

**Veri Tabanı (Data Base):** İstenen belli bir düzene göre yapılandırılmış verilerin saklanmasını sağlayan özel yazılım.

# Y

**Yankı (Echo):** Veri Aktarımı süreci içerisinde gönderilen verinin alındığında, karşılaştırma amacıyla gönderen noktaya tekrar gönderilmesi.

**Yapay Zeka (Artificail Intelligence):** Bilgisayarların zeka gerektiren bazı davranışlarda bulunabileceği iddiası ve bu amaçla geliştirilmekte olan araştırma programları. İki tür yapay zeka iddiası vardır: Aşırı (ya da felsefi) yapay zeka iddiasına göre insanlar da zaten bilgisayarlar gibi düşünür. Ilımlı (ya da teknik) yapay zeka iddiasına göre, bilgisayarların çalışma şekli ile insanların düşünme şekilleri arasında bir benzerlik olmasa bile, bilgisayarlar insana özgü bazı zeki davranışları benzetleyebilir.

**Yapılandırılmış Program (Structured Programming):** Bir işi giderek küçülen ve yalınlaşan bölümlere bölme gerekliliğini vurgulayan bir programlama yaklaşımı.

**Yarılamalı Arama (Binary Search):** Herhangi bir sıralı listede bir veri kaydının aranması için bir yöntem. Bu yöntemde önce, listenin ortasındaki kayıt aranan veri ile karşılaştırılır, eğer ortadaki kaydın aranmakta olan kayıttan önce gelmesi gerektiğine karar verilirse (kütüğün sıralama ilkesine göre), kütüğün ilk yarısı arama dışı bırakılır. Ve ikinci yarı için aynı işlem tekrar edilir. Aranan kayıt bulununcaya kadar bu böyle sürer.

**Yerel Bilgisayar Ağı (Loacal Area Network-LAN):** Birimlerin aynı konumda bulunduğu bilgisayar şebekesi. Ayırdedici

özelliği sistemlerin birbirlerine telekomünikasyon hatları ile değil doğrudan kablolarla bağlanmış olmalarıdır.

**Yineleme (Iteration):** Belli bir sonucun, aynı işlemler dizisi, belli bir koşul doğrulanıncaya kadar üstüste yinelenmesi ile elde edilmesi.

**Yürütme Hatası (Execution Error):** Programın yürütülmesi sırasında ortaya çıkan bir hata.

**Yüzer Noktalı Yazı Düzeni (Floating Point Notation):** Bu yazı düzeninde sayılar ondalık bir kesir ve onu çarpan tabanın belli bir üssü ile gösterilir. Örneğin 165.487 sayısı şu şekilde gösterilebilir: 0.165487 x 103. Bu yazı düzeni bilgisayarda temsil edilebilecek sayıların büyüklüğünü artırır. Ancak bu işlem hızı ve hassasiyeti pahasına olur. Bkz. Sabit Noktalı Yazı Düzeni.

# Z

**Zaman Paylaşımı (Time Sharing):** Bilgisayar çoklu erişimi mümkün kılan bir sistem. Her kullanıcıya sistemin kaynaklarından yararlanması için belli bir zaman ayrılır; ancak bu zaman aralıkları o kadar kısadır ve o kadar hızlı değişir ki, kullanıcı sistemi kesintisiz olarak kullanıyormuş gibi görünür.

**Zeki Terminal (Intelligent Terminal):** Merkez bilgisayara erişime gerek kalmadan, sınırlı bazı veri-işlemleri kendi başına gerçekleştirebilen terminal.

**Zincirleme (Chaining):** Ana belleğe sığmayacak kadar kapsamlı bir programın parça parça belleğe alınarak işlenmesi.

# İçindekiler

**Cilt 1**


| Sayfa | |
|---|---|
| 13 | BİLGİSAYAR: BÜYÜK BİLİNMEYEN |
| 15 | Veriler |
| 22 | Hesap makinesinden bilgisayara |
| 22 | Geçmiş ve şimdiki durum |
| 28 | Bilgisayarların evrimi: Uygulamalar |
| 33 | BİLGİSAYARIN MANTIĞI |
| 33 | Benzeştirmeli bilgisayarlar |
| 33 | Sayısal bilgisayarlar |
| 37 | Bir bilgisayardan nerede ve nasıl yararlanılmalıdır? |
| 38 | Bilgisayarların sınıflandırılması |
| 43 | Bilgisayar ağları |
| 45 | SAYILAMA SİSTEMLERİ |
| 45 | Darbeler |
| 46 | Anahtarlama |
| 47 | Sayılama tabanları |
| 50 | 2 tabanlı sayı sistemi |
| 52 | 16 tabanlı sayı sistemi |
| 55 | 8 tabanlı sayı sistemi |
| 56 | 2 tabanlı sayı sisteminde işlemler |
| 60 | 2 tabanlı toplama |
| 61 | 2 tabanlı çıkarma |
| 72 | MANTIK ÖGELERİ |
| 72 | İşlemciler |
| 73 | Boole cebiri işlemcileri |
| 79 | Kablolu mantık |
| 80 | Mantık devreleri |
| 80 | NOT (DEĞİL) işlemcisi |
| 82 | AND (VE) işlemcisi |
| 89 | OR (VEYA) işlemcisi |
| 90 | XOR (DAR VEYA) işlemcisi |
| 90 | Diğer zıtlayıcılar |
| 97 | Tümleşik devreler |
| 97 | Birleşik devreler |
| 97 | Uygulama alanları |
| 99 | Diğer tümleşik devre türleri |
| 107 | VERİ TAŞIMA KODLARI |
| 107 | Bit ve bayt |
| 110 | ASCII kodu |
| 111 | Güdüm işaretleri |
| 113 | Simgeler ve sayılar |
| 114 | Harfler |
| 114 | Uygulama alanları |
| 117 | ASCII kodlarının taşınması |
| 121 | Şifrenin alım sırasında kontrolü |
| 121 | Güvenlik kodları |
| 125 | KİŞİSEL BİLGİSAYAR SİSTEMLERİ |
| 125 | Minik işlemcinin temel yapısı |
| 126 | MİB'in yapısı |
| 126 | MİB'in çalışma şekli |
| 129 | Bir atlama komutunun evreleri |
| 131 | Bir programa nasıl ara verilir? |

| Sayfa | |
|---|---|
| 133 | Veri alışverişi |
| 138 | Bellekler |
| 139 | Çevresel gereçler |
| 142 | Klavyeler |
| 142 | Bilgiyazarlar |
| 142 | Disk bellek birimleri |
| 146 | MİB'in komutları |
| 147 | Veri taşıma komutları |
| 147 | Aritmetik ve mantık komutları |
| 148 | Atlama komutları |
| 148 | Giriş/çıkış komutları |
| 148 | Özel komutlar |
| 154 | PROGRAMLAMA |
| 154 | Programlama dilleri |
| 154 | Yüksek düzeyli programlama dilleri |
| 155 | Derleme |
| 156 | BASIC |
| 156 | FORTRAN |
| 156 | COBOL |
| 157 | ÇÖZÜMLEME VE AKIŞ ŞEMALARI |
| 157 | Bir programın hazırlanışı |
| 158 | Çözümleme |
| 162 | Bireşim |
| 168 | Akış şemaları |
| 168 | Sık kullanılan simgeler |
| 170 | Daha az kullanılan grafik simgeler |
| 172 | Bir uygulama örneği |
| 174 | Döngüler |
| 175 | Sınırları belirlenmiş döngüler |
| 176 | Sınırları parametrelere bağlı döngüler |
| 176 | Verilerin bittiğini anlayan döngü |
| 177 | Kod deişmesiyle durdurulabilen döngüler |
| 186 | Çok adımlı döngüler |
| 190 | Döngüden çıkış |
| 193 | Akış şemalarının kullanılışı |
| 200 | Bilimsel uygulamalar |
| 203 | Yinelemeli yöntemin uygulama alanları |
| 209 | Yapılandırılmış programlama |
| 220 | VERİLERİN SAKLANMASI |
| 220 | Yığın bellekler |
| 222 | Manyetik şeritler |
| 227 | Manyetik diskler |
| 229 | Manyetik davullar |
| 231 | Disketler |
| 231 | Kütükler |
| 236 | Manyetik şerit üzerinde kütüklerin düzeni |
| 238 | Manyetik diskler üzerinde kütüklerin düzeni |
| 238 | Sıralı düzen |
| 238 | Zincirleme sıralı düzen |
| 239 | Anahtarlı sıralı düzen |
| 241 | Doğrudan erişimli düzen |
| 246 | Küçük ve kişisel bilgisayarlarda arşiv düzeni |
| 248 | Doğrudan erişimli kütükler (Random access files) |
| 248 | Sıralı kütükler |

| Sayfa | |
|---|---|
| 248 | Dizinli kütükler |
| 254 | Dizinli kütüğün işletimi (DSEY sistemi) |
| 255 | Verilere erişme işlemleri |
| 255 | DSEY (ISAM) kütüğünün genel yapısı |
| 256 | İki tabanlı ağacın yapısı |
| 256 | DSEY ağaç yapısı |
| 258 | Açık kütüklerin ve ara belleklerin sayısı |
| 259 | Dolaylı adresleme |
| 260 | Bölümleme |
| 260 | Çarpma |
| 261 | 11 ile çarpma |
| 261 | Kare alma |
| 261 | Bir uygulama örneği: Aile bütçesi |
| 261 | Hesap planı |
| 262 | Adresin değerini belirleyen program birimi |
| 269 | Veri girişi sınama birimi |
| 271 | Kütüklerde veri arama |
| 271 | Seri arama |
| 275 | Yarılamalı arama |
| 275 | Verilerin sıralanması |
| 283 | Veri sıkıştırma |
| 283 | Büyük sistemlerde kütük yönetimi |
| 284 | Karşılaştırmalı veri bankaları |

**Cilt 2**

| | |
|---|---|
| 291 | PROGRAMLAMA |
| 291 | Gereken çıktıların tanımlanması |
| 292 | Giriş verilerinin ve işlem algoritmalarının sınanması |
| 292 | Bilgisayarın seçimi |
| 295 | İşletim sistemleri |
| 296 | İşletim sisteminin işlevleri |
| 298 | Disketle kütük yönetimi |
| 298 | Düzenleme |
| 299 | Fihrist |
| 299 | Kapsam |
| 305 | İşletim sistemlerinin kütük yönetim işlemleri |
| 311 | BASIC'e geçiş işlemleri |
| 314 | Bir işletim sistemini değerlendirirken kullanılan ölçütler |
| 314 | Programların bölümlenmesi |
| 316 | Parametrelerin aktarılması |
| 317 | BASIC PROGRAMLAMA DİLİ |
| 320 | Genel özellikler |
| 321 | İşlem konumları |
| 321 | BASIC komutlarının yapısı |
| 322 | CTRL tuşunun işlevleri |
| 322 | BASIC'in etkileşimli kullanımı (Doğrudan ya da aracısız konum) |
| 323 | İşlemcilerin simgeleri, anlamları ve öncelikleri |
| 323 | Aritmetik işlemciler |
| 323 | Karşılaştırma işlemcileri |
| 326 | Mantıksal işlemciler |
| 326 | Fonksiyonel işlemciler |
| 337 | Tam sayılar için bölme işlemcisi |
| 338 | Bir bölme işleminde artanı hesaplayan MOD işlemcisi |
| 338 | Sabitler ve değişkenler |
| 338 | Sayısal sabitler |
| 339 | Hassasiyet |
| 341 | Sayısal değişkenler |

| Sayfa | |
|---|---|
| 343 | Değişik türden değişkenlerin çevrilmesi |
| 343 | Yuvarlama |
| 343 | Değer verme sırasında yuvarlama |
| 345 | Hesaplarda yuvarlama |
| 345 | Mantık işlemlerinde yuvarlama |
| 345 | Yapılandırılmış değişkenler |
| 346 | Dizeyler |
| 346 | Karakter zincirleri |
| 352 | Zincir bölümleri |
| 352 | Güdümler |
| 353 | Program yazma ya da saklama güdümleri |
| 363 | Manyetik şerit güdümleri |
| 363 | Programların basılması ve liste halinde dökümü |
| 363 | Programların yürütülmesi ve durdurulması |
| 363 | Eklemleme (MERGE) |
| 364 | Komutlar |
| 364 | Programların başı ve sonu |
| 368 | Değişken türünün saptanması |
| 370 | Değer verme komutları |
| 370 | LET |
| 372 | DATA READ ve RESTORE |
| 382 | LSET ve RSET |
| 382 | SPACE(N) |
| 384 | SPC(N) |
| 385 | SWAP |
| 385 | Döngü komutları |
| 385 | FOR...NEXT |
| 393 | WHILE...WEND |
| 397 | Atlama komutları |
| 397 | GOTO... |
| 397 | ON...GOTO... |
| 401 | Koşullu komutlar |
| 409 | Koşullu komutlarda gerçek sayısal değişkenlerin kullanımı |
| 411 | Alt-program çağırma ve programların eklemlenmesi |
| 413 | Bir uygulama örneği: Dizel mi, benzinli mi? |
| 417 | Hatalar ve tanıları |
| 418 | Veri girişi için tanı |
| 434 | Fonksiyonlar |
| 435 | Nokta fonksiyonunun tanımı |
| 437 | İnterpolasyon ve eksterpolasyon |
| 439 | Fonksiyonların analitik gösterimi |
| 440 | BASIC fonksiyonları |
| 440 | Sayısal (Nümerik) fonksiyonlar |
| 440 | ABS(R) |
| 441 | CDBL(R) |
| 442 | CNTRL(R) |
| 442 | CSNG(R) |
| 442 | FIX(R) |
| 443 | INT(R) |
| 443 | SNG(R) |
| 443 | SQR(R) |
| 444 | RND(R) |
| 447 | ATN(R) |
| 447 | COS(R) |
| 447 | SIN(R) |
| 447 | TAN(R) |
| 448 | Trigonometrik fonksiyonların kullanımı |
| 449 | EXP(R) |
| 450 | LOG(R) |
| 450 | Karakter zinciri fonksiyonları |
| 450 | ASC(A$) |
| 450 | CHR$(N) |

**Sayfa**

452 CVI(A$)
452 CVS(A$)
452 CVD(A$)
454 HEX$(N)
454 INKEY$
455 INSTR(N,A$,B$)
457 LEN(A$)
457 MID$(A$,NS,NC)
458 MKI$(N)
458 OCT$(N)
460 RIGHT$(A$,N)
460 STR$(R)
469 STRING$(N,K)
469 VAL(A$)
469 Özel fonksiyonlar
469 EOF$(N)
470 FRE(A)
470 INP(A)
470 LOC(N)
471 LPOS(N)
472 OUT N,M
472 PEEK(N)
472 POKE N,M
473 POS(N)
474 Kullanıcı tarafından tanımlanan fonksiyonlar
483 Sayısal fonksiyonlar: İkinci dereceden denklemlerin çözümü
485 Karakter zinciri fonksiyonları: Büyük ve küçük harflerin kontrolü
488 Özel uygulamalar
496 Uygulama örneği: Isı akışının hesaplanması
496 Genel çözüm
499 Alt programlar
520 Dizeyler
521 Dizeylerin düzenlenmesi
521 Bir ve çok boyutlu dizeyler
522 Dizeylerde verilerin yerleştirilmesi
529 Dizeylerin kullanımı
531 Belli sayıda basamaktan oluşan değişkenler üzerinde işlem yapmak
537 İstatistiksel uygulamalar
547 Verilerin girişi ve çıkışı için fonksiyonlar
548 Veri girişi için fonksiyonlar
549 INPUT
552 INKEY$
554 INPUT$(N)
555 LINE INPUT
555 Veri çıktıları için fonksiyonlar
556 L PRINT
557 TAB(N)
557 LPRINT USING
560 Bilgiyazımın yapılandırılması
560 Parametrelenmemiş tabloların üretimi
566 Parametrelenmiş tabloların üretimi

**Cilt 3**

579 Bilgiyazarın kullanımı
580 Bir iğneli bilgiyazarın yapı ve özellikleri
583 Bilgiyazım örnekleri
588 Çubuk diyagramların çizimi
594 Ekran terminalinin kullanımı
594 Bir ekran terminalinin yapısı ve özellikleri
596 İmleç'in kullanımı

| Sayfa | |
|---|---|
| 596 | Ekran maskeleri |
| 605 | Fonksiyon tuşları |
| 617 | Menüler |
| 629 | Histogramların üretimi |
| 637 | Kütük yönetimi |
| 638 | Sıralı kütüklere erişim için fonksiyonlar |
| 642 | Verilerin sıralı bir kütüğe eklenmesi |
| 642 | Doğrudan erişimli kütüklere (Random files) erişim için fonksiyonlar |
| 654 | Verilerin yazı düzenleri ve kodlanmaları |
| 659 | Veri seçimi |
| 660 | Güdümler ve özel fonksiyonlar |
| 661 | Bölümlenmiş kütükler (Multi volume) |
| 662 | Büyük sistemlerde kütük yönetimi |
| 662 | Stok hesabı |
| 665 | Stokların ve envanterin düzeltilmesi (Yeniden değerlendirme) |
| 665 | Temel mallar listesi |
| 665 | Temsil (Simulation) |
| 665 | Diğer yöntemlerle ilişki |
| 666 | Kullanım örneği: Bir müşteri kartoteksinin kurulması |
| 666 | Kütüğün açılması (1000 alt-program) |
| 670 | Menü (2000 alt-program) |
| 670 | Giriş (3000 alt-program) |
| 670 | Güncelleştirme (4000 alt-program) |
| 672 | Arama (5000 alt-program) |
| 673 | Bilgiyazım (6000 alt-program) |
| 673 | Özel bilgiyazım fonksiyonları |
| 675 | Verilerin sıralanması |
| 676 | Verilerin bellekte sıralanması |
| 680 | Disket üzerinde sıralama |
| 683 | OLISORT hazır programı |
| 689 | Yapabildikleri |
| 691 | İşletim türleri |
| 695 | Bir veri işleme yönteminin geliştirilmesi |
| 695 | Giriş evresi |
| 695 | Ekran maskesinin hazırlanması |
| 699 | Veri maskesinin görüntülenmesi |
| 702 | Verilerin girilmesi |
| 703 | Disket üzerinde saklama aşaması |
| 709 | Yazma |
| 712 | Okuma |
| 717 | Ana program |
| 717 | Derleme |
| 717 | Hazırlık |
| 724 | Derlenmiş |
| 724 | Döküm |
| 725 | Ham program |
| 727 | Derleyicinin özel işlevleri |
| 727 | Uzlaşımların belirlenmesi |
| 730 | Hataları saptayan fonksiyonlar |
| 731 | Özel kodlar |
| 731 | Bağlama |
| 732 | BASIC 80'i tamamlayan diğer özel komutlar |
| 732 | Genel kullanıma yönelik komutlar |
| 738 | Çizimlerin komutları |
| 740 | Özel komutlar |
| 740 | Ses üretimi |
| 740 | Joystick (Oyun çubuğu) ve pedal |
| 741 | Işıklı kalem |
| 746 | BASIC'in lehçeleri |
| 746 | Karakter zincirlerinin kullanımı |
| 747 | Disket üzerinde arşivlerin kullanımı |
| 755 | G/Ç fonksiyonlarının kullanımı |

**Sayfa**

757 Çevresel gereçlerin kullanımı: Arabirimlerin işlevi
760 Veri kabulü ve iletimi

769 YAZILIM

770 Program mimarisi
772 Modelleme
772 Makroskopik model
774 Mikroskopik model
775 Programların yapılanması
776 Yukarıdan aşağı geliştirme tekniği
780 Yazılım hataları
784 Programın sınanması
785 Sınama yöntemleri
787 Aşağıdan yukarı yöntem
787 Yukarıdan aşağı yöntem
788 Big Bang yöntemi
788 Sandviç yöntemi
788 Neden sonuç grafikleri
788 Hatalardan arındırma
790 Yazılımın güvenliği
793 Genel uygulama programları
793 Yerleşim haritası tekniği
798 Giriş durumu
799 Güdüm durumu
801 Yerleşim haritası tekniğinin fonksiyonları
813 Matematik fonksiyonlar
813 Özel sayısal fonksiyonlar
818 Arama fonksiyonları
820 Mantık fonksiyonları
820 AND (İfade listesi)
820 IF (Koşul doğru yanlış)
822 NOT (Koşul)
822 ISNA (Hücre)
822 ISERROR (Hücre)
822 OR (Koşullar)
822 Argümansız fonksiyonlar
825 Yerleşim haritası tekniğine ilişkin uygulama örnekleri
825 Apartman giderlerinin bölünmesi
832 Stok yönetimi
835 Değişmelerin çözümlenmesi (WHAT IF)
843 Yinelenen işlemler için fonksiyonlar
845 Yerleşim haritasının gelişimi
847 Kütüklerin yönetimi ile ilgili programlar
847 Kütüklerin oluşturulması ve alanların tanımlanması
849 Verilerin girilmesi ve güncelleştirilmesi
849 Arama (FIND)
851 Sıralama (SORT)
851 Verilerin işlenmesi
858 Tabloların bilgiyazımı
858 Metinlerin işlenmesi

**Cilt 4**

867 ASSEMBLER DİLİ

867 Minik işlemcinin yapısı ve işleyiş biçimi
869 Siciller
873 Merkezi işlem biriminin işleyiş biçimi
874 Minik işlemcinin iç yapısı
881 Aritmetik ve mantık birimi (AMB)
882 İki tabanlı toplama
884 İki tabanlı çıkarma

| Sayfa | |
|---|---|
| 884 | Tümleme |
| 890 | İki tabanlı çarpma |
| 890 | İki tabanlı bölme |
| 891 | Boole işlemleri |
| 892 | Kaydırma işlemleri (Shift) |
| 892 | Durum sicili |
| 892 | Elde (Carry) |
| 892 | Sıfır |
| 892 | Taşma (Overflow) |
| 892 | Negatif |
| 893 | Yarım elde |
| 894 | Tamlama |
| 897 | Özel çentikler |
| 897 | AMB ara belleği |
| 897 | Güdüm birimi |
| 898 | Komutların yerine getirilmesi |
| 899 | Sinyallerin zamansal yönetimi |
| 904 | Araya girmenin kullanılması |
| 906 | Deste bellek (İstif depo belleği) |
| 910 | Assembler komutlarının yapısı |
| 912 | Adresleme yöntemleri |
| 913 | Örtük adresleme |
| 913 | Mutlak ya da doğrudan adresleme |
| 915 | Dolaylı adresleme |
| 915 | Dizinli adresleme |
| 917 | Göreli adresleme |
| 917 | Sicil adreslemesi |
| 918 | Deste adreslemesi |
| 918 | Assembler komutları |
| 918 | Koşullu komutlar |
| 921 | Aritmetiksel komutlar |
| 922 | Mantıksal komutlar |
| 922 | Veri aktarımı için komutlar |
| 929 | Atlama komutları (BRANCH) |
| 930 | Bir komutun atlanması için komutlar (SKIP) |
| 930 | Bir alt-programın çağırılışı ve ana programa geri dönüş için komutlar |
| 932 | Değişik komutlar |
| 933 | Assembler komutlarının çevrilmesi için komutlar |
| 942 | Assembler ile programlamaya örnek |
| 945 | Rockwell 6502 minik-işlemcisinin Assembler'ı |
| 947 | Z80 Zilog minik-işlemcisinde kullanılan Assembler |
| 951 | COBOL DİLİ |
| 951 | Genel |
| 953 | Kaynak program |
| 955 | Derleme ve bağlama |
| 957 | COBOL programı ve taslak oluşturma kuralları |
| 960 | Bir COBOL programının yapısı |
| 964 | Adlandırma bölümü |
| 965 | Çevre bölümü |
| 965 | Configuration section |
| 965 | Özel isimler paragrafı |
| 966 | File control paragraf |
| 967 | Veri bölümü |
| 967 | File section |
| 968 | Verilerin açıklanması. BLOCK CONTAINS bildirimi (Öbek içeriği bildirimi) |
| 968 | LABEL RECORD (Etiket) bildirimi |
| 969 | RECORDİNG MODE bildirimi (Kaydetme tarzı) |
| 970 | RECORD CONTAINS bildirimi (Veri kaydı uzunluğu bildirimi) |
| 970 | Veri kayıtlarının yapısı |
| 973 | WORKING STORAGE SECTION |

| Sayfa | |
|---|---|
| 973 | 77. düzey |
| 973 | 88. düzey |
| 974 | 66. düzey ve RENAMES bildirimi (Yeniden adlandırma bildirimi) |
| 975 | REDEFINES bildirimi |
| 976 | PICTURE bildirimi (Maske bildirimi) |
| 978 | Sayısal alanların açıklanması |
| 978 | Alfabetik alanların açıklanması |
| 979 | Harfsayısal alanların açıklanması |
| 979 | PROCEDURE DIVISION (Prosedür bölümü) |
| 988 | Bir PROCEDURE DIVISION'ın yapılandırılması için örnek |
| 992 | COBOL komutları |
| 994 | G/Ç fiilleri |
| 995 | Kütüklerin açılması ve kapanması: OPEN ve CLOSE fiilleri |
| 995 | Okuma için kullanım: OPEN INPUT |
| 997 | Yazma için kullanım: OPEN OUTPUT |
| 997 | Okuma ve yazma için kullanım: OPEN I O |
| 1000 | Ek: OPEN EXTEND |
| 1000 | Kütükler üzerinde yazma veya okuma: READ ve WRITE fiilleri |
| 1003 | Bilgiyazım yazım kütüklerine yazma |
| 1003 | Satır atlamaları için bildirimler |
| 1004 | Sayfaların atlanması için bildirimler |
| 1004 | LINAGE bildirimi |
| 1006 | ACCEPT ve DISPLAY komutları |
| 1009 | Aritmetiksel fiiller |
| 1009 | Bellekteki verilerin gösterimi: USAGE bildirimi |
| 1013 | Aritmetiksel ifadeler |
| 1015 | COMPUTE (HESAPLA) komutu |
| 1020 | ADD fiili |
| 1023 | SUBTRACT fiili |
| 1024 | MULTIPLY fiili |
| 1024 | DIVIDE fiili |
| 1025 | Verilerin gönderilmesine ve işlenmesine ilişkin fiiller |
| 1026 | MOVE fiili |
| 1027 | Harfsayısal MOVE |
| 1030 | ALL bildirimi |
| 1036 | Sayısal MOVE |
| 1038 | Inspect komutu |
| 1038 | Sayma işlevi |
| 1040 | ALL bildirimi |
| 1040 | LEADING bildirimi |
| 1040 | CHARACTERS bildirimi |
| 1040 | BEFORE bildirimi |
| 1041 | AFTER bildirimi |
| 1041 | Yerine geçirme işlevi |
| 1041 | INSPECT komutunun genel yazı düzeni |
| 1043 | STRING komutu |
| 1048 | UNSTRING komutu |
| 1049 | Denetleme fiilleri |
| 1052 | IF önermesi |
| 1053 | IF komutunun yürütme mantığı |
| 1056 | NEXT SENTENCE seçeneği |
| 1057 | IF komutuna bağlı koşullar |
| 1057 | İlişki çözümlemesi |
| 1064 | İşaret çözümlemesi |
| 1064 | Koşul çözümlemesi |
| 1067 | Sınıf çözümlemesi |
| 1067 | Mantıksal işlemcilerin kullanımı |
| 1071 | GOTO komutu |
| 175 | DEPENDING ON bildirimi |
| 1076 | PERFORM fiili |
| 1084 | EXIT komutu |

| Sayfa | |
|---|---|
| 1085 | Aynı yordamın yinelemeli uygulamaları |
| 1085 | VARYING... FROM... BY bildirimi |
| 1085 | STOP komutu |
| 1086 | COBOL'da tabloların yönetimi |
| 1089 | Tek boyutlu tablolar |
| 1095 | İki boyutlu tablolar |
| 1100 | Üç boyutlu tablolar |
| 1103 | Bir tablonun yüklenmesi |
| 1111 | Tablo dizinleri |
| 1112 | Tablolarda arama için SEARCH fiili |
| 1115 | Sıralı SEARCH'ün kullanılışı |
| 1119 | SEARCH ALL komutu |
| 1120 | Bir tablonun sıraya sokulması |
| 1124 | Verilerin sıraya sokulması (Sort) |
| 1127 | INPUT PROCEDURE ve RELEASE fiili |
| 1127 | OUTPUT PROCEDURE ve RETURN fiili |
| 1129 | Ek kütüklerin kullanımı |
| 1132 | DS kütüklerinin ENVIRONMENT DIVISION'da tanımlanması |
| 1133 | DS kütüklerinin DATA DIVISION'da tanımlanması |
| 1133 | PROCEDURE DIVISION'daki komutlar |
| 1133 | READ fiili |
| 1134 | WRITE fiili |
| 1134 | DS kütüklerine özgü komutlar |
| 1134 | REWRITE fiili |
| 1134 | DELETE fiili |
| 1135 | START fiili |
| 1136 | CLOSE fiili |
| 1136 | Programlararası iletişim: Alt-programlama |
| 1145 | CALL fiili |
| 1145 | EXIT PROGRAM komutu |
| 1147 | Uygulama örnekleri |


## Cilt 5


| | |
|---|---|
| 1115 | FORTRAN DİLİ |
| 1155 | BASIC'e oranla FORTRAN'ın bellibaşlı özellikleri |
| 1158 | Aritmetik mantık ve karşılaştırma fonksiyonları |
| 1158 | FORTRAN'da değişkenler ve sabitler |
| 1160 | Değişken türünün bildirilmesi |
| 1162 | Yapılandırılmış değişkenlerin boyutlandırılması |
| 1162 | COMMON ve EQUIVALENCE komutları |
| 1164 | Atama komutları |
| 1166 | FORTRAN komutları |
| 1166 | GOTO komutu |
| 1166 | Döngüler |
| 1167 | Dolaylı döngüler |
| 1168 | DO...WHILE...biçimi (...'yı doğruysa yürüt) |
| 1168 | Koşullu komutlar |
| 1170 | Alt-programların kullanımı |
| 1172 | Yürümeyi bitiren ve durduran komutlar |
| 1172 | FORTRAN fonksiyonları |
| 1173 | Kitaplık fonksiyonları |
| 1173 | Tür değiştirme |
| 1174 | İşlem fonksiyonları |
| 1175 | Karakter zinciri fonksiyonları |
| 1175 | Matematik fonksiyonlar |
| 1175 | Kullanıcı tarafından programlanan fonksiyonlar |
| 1177 | G/Ç komut ve düzenleri |
| 1180 | G/Ç düzenleri |
| 1181 | Tamsayıların yazı düzeni |

| Sayfa | |
|---|---|
| 1182 | Gerçek sayıların yazı düzeni |
| 1182 | Karakterlerin yazı düzeni |
| 1182 | Mantıksal sabitlerin yazı düzeni |
| 1182 | Sayfa düzeni tanımlayıcısı |
| 1182 | Giriş tanımlayıcıları |
| 1182 | Çıkış tanımlayıcıları |
| 1183 | Ortak tanımlayıcılar |
| 1183 | Ölçek katsayısı |
| 1183 | Yazı düzeninin tekrarlanması |
| 1183 | Düzenlenmemiş G/Ç komutları |
| 1184 | Giriş komutları |
| 1184 | Çıkış komutları |
| 1184 | FORTRAN ile programlama örneği |
| 1188 | Kütüklerin yönetimi |
| 1193 | Disket üzerindeki kütüklerin yönetimi |
| 1196 | FORTRAN'a özgü komutlar |
| 1199 | Özel uygulamalar |
| 1200 | Komutların normal düzeni |
| 1204 | PASCAL DİLİ |
| 1205 | Yapılandırılmış programlama |
| 1207 | Programların düzenlenmesi |
| 1208 | İlk kod |
| 1208 | Yapılandırılmış akış şemaları |
| 1209 | Kontrol yapıları |
| 1209 | Sıra |
| 1209 | Seçim |
| 1215 | Tekrar |
| 1217 | PASCAL'de veriler |
| 1217 | Veri türleri |
| 1219 | Tamsayı veriler |
| 1221 | Gerçek sayılar |
| 1227 | Karakterler |
| 1228 | Mantıksal veriler |
| 1228 | Tür tanımları |
| 1229 | Sabitlerin tanımlanması |
| 1229 | Değişkenlerin tanımlanması |
| 1230 | Standart olmayan veri türleri |
| 1231 | Ex novo tanımlanıp sıralanan veri türleri |
| 1233 | Alt cümle veri türleri |
| 1234 | İlk PASCAL komutları |
| 1234 | Aritmetik ve boole ifadeler |
| 1235 | PASCAL fonksiyonlarının kullanılması |
| 1236 | Atama komutu |
| 1237 | Çıkış komutları: READ ve READLN |
| 1241 | Çıkış komutları: WRITE ve WRITELN |
| 1242 | Bir PASCAL programının yapısı |
| 1249 | Denetleme komutları |
| 1249 | Yineleme komutları |
| 1249 | FOR...TO...DO... komutu |
| 1251 | WHILE...DO... komutu |
| 1251 | REPEAT...UNTIL... komutu |
| 1253 | Koşullu komutlar |
| 1253 | IF...THEN...ELSE... komutu |
| 1255 | CASE...OF... komutu |
| 1257 | Koşulsuz atlama komutu: GOTO... |
| 1259 | PASCAL'ın özel yönleri |
| 1261 | PASCAL'de programlama örnekleri |
| 1266 | ÖZET: BİR FATURALAMA İŞLEVİ GELİŞTİRİLMESİ |

| Sayfa | |
|---|---|
| 1266 | Sorunun çözümlenmesi |
| 1273 | Kullanılan kütükler |
| 1275 | Ana menü |
| 1277 | Kütük yönetimi |
| 1286 | Fatura düzenleme |
| 1305 | Faturanın yazılması |
| 1305 | Diğer fonksiyonlar |
| 1316 | BİLGİLERİN İLETİLMESİ |
| 1316 | Farklı iletişim tarzları |
| 1319 | Basit yarım ve tam çift yönlü kanallar |
| 1321 | Telefon hatlarıyla iletişim |
| 1321 | Sinyallerin özellikleri |
| 1324 | Modülasyon teknikleri: Modemler |
| 1326 | Bağlantıların yapısı |
| 1326 | Adım adım (Point to point) bağlantı yapısı |
| 1326 | Çok adımlı (Multipoint) yapı |
| 1329 | Çok düzeyli yapı (Multiplexer) |
| 1330 | İletişim arabirimleri |
| 1333 | Kısa mesafeli bağlantılar |
| 1337 | İletişim protokolleri |
| 1337 | Handshake (Tokalaşma) |
| 1340 | XON/XOFF yöntemi |
| 1340 | ENQ/ACK yöntemi |
| 1349 | BSC protokolü |
| 1352 | HDLC protokolü |
| 1356 | Veri iletişim şebekeleri |
| 1356 | Şebekelerin topolojisi ve özellikleri |
| 1356 | Zincirleme şebekeler |
| 1356 | Yıldızlı şebeke |
| 1356 | Halkalı şebeke |
| 1359 | Ağaç ya da hiyerarşik şebeke |
| 1360 | Kamu şebekeleri |
| 1360 | Yerel şebekeler |
| 1362 | Standart yapılara doğru |
| 1364 | Veri iletiminde BASIC komutları |
| 1368 | İŞLETİM SİSTEMLERİ |
| 1368 | Tekli kullanım ve çoklu kullanım |
| 1369 | Batch ve etkileşimli işlem |
| 1369 | Çoklu programlama |
| 1371 | Zaman bölüşümü (Time sharing) |
| 1372 | Gerçek zamanda işleme (Real time) |
| 1373 | Kütük yönetim sistemi |
| 1374 | UNIX kütüklerinin yönetim yapısı |
| 1379 | Eşzamanlı süreçler |
| 1381 | Süreçler arasında iletişim |
| 1383 | BASIC'DE GRAFİK BİLGİ İŞLEM |
| 1384 | Grafik bilgi işlemin uygulama alanları |
| 1384 | İş programlarında grafik |
| 1384 | Bilgisayar destekli tasarım (CAD) |
| 1385 | Mekanik tasarım |
| 1385 | Elektronik tasarım |
| 1385 | Mimari tasarım |
| 1386 | Endüstriyel tasarım |

| Sayfa | |
|---|---|
| 1386 | İmge işlem (İmage processing) |
| 1387 | Canlandırma (Animasyon) |
| 1387 | Grafik kullanıma yönelik çevresel gereçler |
| 1388 | Giriş gereçleri |
| 1389 | Grafik levha |
| 1394 | Özel gereçler |
| 1397 | Çıkış gereçleri |
| 1397 | Grafik bilgiyazar |
| 1403 | Çizici (Plotter) |
| 1407 | Tek renkli monitör |
| 1410 | Renkli monitör |
| 1411 | Uzmanlaşmış gereçler |
| 1414 | Grafik için BASIC komutları |
| 1414 | Doğru kesitlerinin çizimi |
| 1418 | Bir doğru parçasının ekranda döndürülmesi (Rotation) |
| 1426 | Bir çemberin çizimi |
| 1430 | Bir fonksiyonun grafiğinin görüntülenmesi |
| 1433 | Doğru parçaları ile bir fonksiyon grafiğinin çizimi |
| 1435 | Bir doğrunun çizimi |
| | **Cilt 6** |
| 1443 | Kartezyen eksenlerinin görüntülenmesi |
| 1443 | İki nokta yardımıyla bir doğrunun çizimi |
| 1450 | En küçük karelerin doğrusal regrasyonu |
| 1464 | Herhangi bir fonksiyon grafiğinin çizim programı |
| 1467 | Karakterlerin grafik biçimde görselleştirilmesi |
| 1502 | Histogramlar ve dilimli diyagramlar |
| 1530 | Grafik modda bellek yönetimi |
| 1530 | Grafik bellek |
| 1532 | Ekran belleğinin adreslenmesi |
| 1532 | Y ekseni üzerinden adresleme |
| 1537 | X eksenine göre adresleme |
| 1539 | Belleğin doğrudan adreslenmesi |
| 1546 | Grafik imgelerin saklanması |
| 1555 | Alan ve çevrelerin hesaplanması |
| 1564 | Figürlerin bellekte saklanması için BASIC komutları |
| 2565 | Ekran pencereleri |
| 1565 | Pencere oluşturma programı |
| 1587 | Ekran çerçevelerinin yönetiminde BASIC komutları |
| 1588 | Grafik vektörler ve figür tabloları |
| 1590 | Yer değiştirme vektörlerinin kodlanması |
| 1593 | Figür tabloları yönetim programı |
| 1604 | Bütünün bölümlerinin yönetimi (Alt-program 4100) |
| 1605 | Figürün yer değiştirmesi (Alt-program 6000) |
| 1610 | Şeklin döndürülmesi (Alt-program 4500) |
| 1611 | Disk üzerine kayıt (Alt-program 5000) |
| 1614 | Disketin okunması (Alt-program 5200) |
| 1614 | Grafik figürlerin yaratılması için hizmet programları |
| 1614 | Şekil düzenleyicisi |
| 1623 | Şekil yükleyicisi |
| 1623 | Grafik figürlerin yönetimi için BASIC komutları |
| 1633 | Üç boyutlu grafik (3D) |
| 1634 | İki değişkenli fonksiyonlar |
| 1634 | 3D gösterimi |
| 1637 | 3D grafik problemleri |
| 1658 | Bilgisayarla resim canlandırma (Animasyon) |
| 1658 | Hareketli cisimler |
| 1658 | Hareketli cisimlerin yazılımla yönetilmesi |
| 1658 | Resim canlandırma |
| 1661 | Zemin yönetimi |

**Sayfa**

1661 Hareketli bir cismin gösterimi
1663 Canlandırmayla ilgili sorunlar
1669 Hareketli cisimlerin donanımla yönetimi
1670 Hareketli cismin bellekte saklanması
1673 Hareketli cismin görüntülenmesi
1674 Cismin hareketi
1674 Hareketli cisimlerin yönetiminde bellek alanının kullanımı
1674 Uygulama örneği
1679 Hareketli cisimlerin gelişmiş yönetimi
1679 Hareketli cisim tanımı
1679 Görüntüleme
1681 Yapay zekâ
1681 Oluşturucu kurallar, düzenleyici kurallar
1681 Felsefe ve yapay zekâ
1682 Soyut bir makine olarak bilgi-işlem sistemleri
1684 Bilgi-işlem sistemleri ve diğer makineler
1685 Bir bilgi-işlem sistemi olarak sinir sistemi
1686 Bilgi ve davranış
1687 Kuralların anatomisi
1687 Bir örnek: MIU sistemi
1687 Aksiyonlar, çıkarım kuralları ve türetmeler
1694 Sınıflandırma sorunları
1695 Bir örnek: Pratisyen bir bilgisayar
1701 Otomatik pratisyenin kısıtlılıkları
1701 Bilgilerin temsil edilişi ile ilgili sorunlar
1704 Öğrenme sorunu
1705 Otomatik pratisyenin öğrenmesi
1710 Daha gelişkin bir program
1710 Öğrenme programı zeki mi?
1715 Önermeler mantığının sınırları
1716 Yüklemler mantığı
1716 Nitelikler ve ilişkiler
1717 IF-THEN işlemcisi
1718 Sabitler ve değişkenler
1718 Niceleyenler
1719 Kapsam sorunları
1720 Aksiyomlar
1722 Bir örnek: Soyağacı ve soyadları
1725 Fonksiyonlar
1725 Fonksiyonlar ve değişkenler
1727 Uzman sistemler
1727 Oyun oynayan uzman sistemler
1728 Diğer uzman sistemler
1728 Akılyürütme bileşeni

**Cilt 7**

1731 Çıkarım mekanizmaları
1731 İkili çözüm
1731 Birim sonuç çözümü
1731 Aşırı çözümlülük
1732 Uzman sistemlerde belirsizlik sorunu
1733 Basit bir uzman sistem
1738 Uzman sistemlerde kullanılan akılyürütmenin derinliği
1740 Prolog: Bir programlama dili olarak mantık
1741 Prolog'a giriş
1746 Prolog ve diğer diller
1748 Prolog'un sınırları ve önemi
1751 Yapay zekânın sınırları

| Sayfa | |
|---|---|
| 1752 | Biyolojik varsayım |
| 1760 | Psikolojik varsayım |
| 1760 | Bir örnek: Algılama ve tanıma |
| 1768 | Psikolojik varsayımın tarihi |
| 1768 | Doğal bilimler ve insan davranışı |
| 1769 | Fizyoloji, fenomenoloji ve psikoloji |
| 1771 | Epistemolojik varsayım |
| 1772 | Fizik ve yapay zekâ |
| 1773 | Dilbilim ve yapay zekâ |
| 1774 | Ontolojik varsayım |
| 1776 | Ontolojik varsayımın tarihi |
| 1779 | Ontolojik varsayımın ötesinde |
| 1780 | Gödel teoremi |
| 1780 | Gödel teoreminin tarih-öncesi |
| 1781 | Paradokslar |
| 1783 | Tipler kuramı |
| 1784 | Matematik, mantık ve üst-mantık |
| 1785 | Tutarlılık |
| 1787 | Tamamlanmamışlık |
| 1788 | Sayı kuramının tamamlanmamışlığı |
| 1791 | Gündelik hayatın fenomenolojisi |
| 1792 | Zekâ gerektiren davranışlarda gövdenin rolü |
| 1796 | Satranç ve biçim-zemin olgusu |
| 1800 | Bütünsellik |
| 1803 | Fiziksel durumlar ve insanlık durumları |
| 1806 | Durumlar ve ilgi alanları |
| 1807 | Programcının sezgileri |
| 1807 | Formel kural ve olguların bağlamı |
| 1811 | Düalizm ve yapay zekâ |
| 1812 | Değerler ve sınırları |
| 1814 | İlgi ve seyir |
| 1817 | Zekâ ve sanatsal yaratım |
| 1818 | Zekâ ve ihtiyaçlar |
| 1819 | Alternatif bir insan tasarımı |
| 1819 | Kavramsal dönüşümler |
| 1821 | Kültürel dönüşümler |
| 1823 | Beşinci kuşak bilgisayarlar |
| 1825 | Beşinci kuşak bilgisayarların ticari ve teknolojik arka planı |
| 1825 | Bilgisayar teknolojisinin tarihi |
| 1826 | Destek sanayiler |
| 1838 | Süper bilgisayarlar |
| 1839 | Japonların katkısı |
| 1840 | Çok yüksek ölçekli tümleştirme |
| 1849 | Japonlar ve VLSI |
| 1850 | Firmalar arası işbölümü ve beşinci kuşak tasarısı |
| 1872 | Beşinci kuşak tasarısının teknolojik hedefi |
| 1874 | Paralel işlem ve veri akışı sistemleri |
| 1875 | Bilgisayarlı kuşaklar ve logo |
| 1878 | Eleştiriler |
| 1886 | LOGO nedir? |
| 1886 | LOGO bir dildir |
| 1887 | Çocuklar için bir dil olarak LOGO |
| 1888 | LOGO'yu öğrenmek |
| 1888 | Neler gerekir? |
| 1890 | LOGO'nun kullanımı ne şekil alır? |
| 1892 | Etkileşimlilik |
| 1892 | İşlem usulü |
| 1895 | Geliştirilebilirlik |
| 1895 | Yinelemelilik |
| 1896 | LOGO ne öğretir? |
| 1896 | Bilgisayar okur-yazarlığı |
| 1896 | Düşünme |

| Sayfa | |
|---|---|
| 1896 | Matematik kavramları ve dil |
| 1897 | Deney yapma |
| 1897 | Sonuç |
| 1897 | Turtle grafikleri |
| 1897 | LOGO'nun lehçeleri |
| 1898 | Karşılıklı modda Turtle |
| 1900 | Turtle'ın eğitimi |
| 1901 | Prosedürlerin adlandırılması |
| 1902 | Prosedür içinde prosedür |
| 1904 | Tekrarlama eylemleri |
| 1904 | Tekrar |
| 1905 | Tail Recursion |
| 1906 | Bir daire çizimi |
| 1910 | Prosedürleri düzenleme |
| 1910 | Apple uyarlaması |
| 1910 | Atari uyarlaması |
| 1911 | Spectrum uyarlaması |
| 1911 | Acornsoft uyarlaması |
| 1911 | Şekiller kitaplığı |
| 1916 | Sayılarla oynamak |
| 1916 | Kaplumbağa grafiğinde sayıların kullanımı |
| 1920 | Toplama yapan programlar |
| 1923 | İkili sayılar |
| 1926 | LOGO'yla yapılabilen diğer aritmetik işlemler |
| 1926 | RANDOM |
| 1927 | Sonuçlar |
| 1927 | Sözcüklerle, listelerle oynamak |
| 1927 | Sözcük nedir? |
| 1928 | Liste nedir? |
| 1928 | Sözcük ve listelerle neler yapabilirsiniz? |
| 1929 | Sözcük ve listelerdeki elemanların konumlarını tarif eden komutlar |
| 1932 | Sözcük ve liste prosedürlerinde tekrar kullanımı |
| 1932 | Sözcükleri ve listeleri birleştirme |
| 1935 | Prosedürlerinize bilgi verme |
| 1938 | Sözcükleri ve listeleri kullanmak için kontrol |
| 1939 | Sonuçlar |
| 1951 | Bilgisayarlaşmanın iktisadi sistem üzerindeki etkileri |


# Okuma parçaları


Bilgisayarlar düşünebilir mi? s.18
Sayı dokuyucu- Bir buluşun hikâyesi s.25
Gündelik yaşamımızda bilgisayarlar s.40
Okulda bilgisayar s.48
Bilişim devrimi s.58
Televizyonunuzdaki terminal s.68
Bilgisayara nasıl danışılır s.84
İletişimde devrim s.92
Yazılım korsanlarına karşı şifreleme s.102
Unutmayan bellekler s.134
Salt okunur belleklerin çalışma şekli s.144
Eb-Ül-İz'in ''Robot Adamları'' s.164
İleti, iletişim ortamının kendisidir s.180
Fiber-optik teknolojisi s.194
Konuşan makineler s.216
Video oyununu icat eden kişi s.234
Otomatik bürolar s.252
Özel amaçlı veri bankaları s.280
Çizgi film ve bilgisayar s.300
Bilgisayarla tasarım s.328
Yonga, pul ve diğerleri s.359
Klavye ve minik işlemci için konçerto s.376
Bilgisayarların geleceği nasıl olacak? s.402
Turizmatik doğdu s.420
İnsan ve bilgisayar: Karşılıklı etkiler s.431
Direksiyonda bir bilgisayar s.477
Öğretmen olarak kaplumbağa s.492
Bilgisayar ve inek s.516
Yapay zekâ s.540
Bilgisayar ve zekâ s.584
Verilerin korunması s.612
Oyun için birkaç KB s.646
Anlayış için bir kişisel bilgisayar s.684
Sammie ve otomobil s.719
Silisyumdan bilgisayara s.765
Hangi yazılım s.777
Bürolarda otomasyon (1) s.807
Bürolarda otomasyon (2) s.827
Bürolarda otomasyon (3) s.852
Bilgisayar yardımıyla teşhis koyma (1) s.935
Bilgisayar yardımıyla teşhis koyma (2) s.983
Bilgisayar yardımıyla teşhis koyma (3) s.1031
İnsan-makine söyleşisi s.1078
Bilgisayarla hava tahminleri s.1140
Niçin video oyunları (1) s.1222
Niçin video oyunları (2): Bir zekâ oyununa doğru s.1269
Niçin video oyunları (3) s.1295
Profesyoneller için savaş oyunu s.1343
Bilgisayar ve radar insanları s.1511
Kişisel bilgisayarların anatomisi s.1548
Bilgisayar ve yelkenli s.1583
Bilgisayar düşünebilir mi? s.1762
Mecazi ifadelerin önemi (1) s.1808
Mecazi ifadelerin önemi (2) s.1828
5. kuşak bilgisayar sistemlerine genel bakış s.1841
Mecazi ifadelerin önemi (3) s.1853
Seymour Papert ve LOGO evreni s.1879
Zeki bilgisayar yardımıyla eğitim (1) s.1913
Zeki bilgisayar yardımıyla eğitim (2) s.1940


2008

Hones