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:

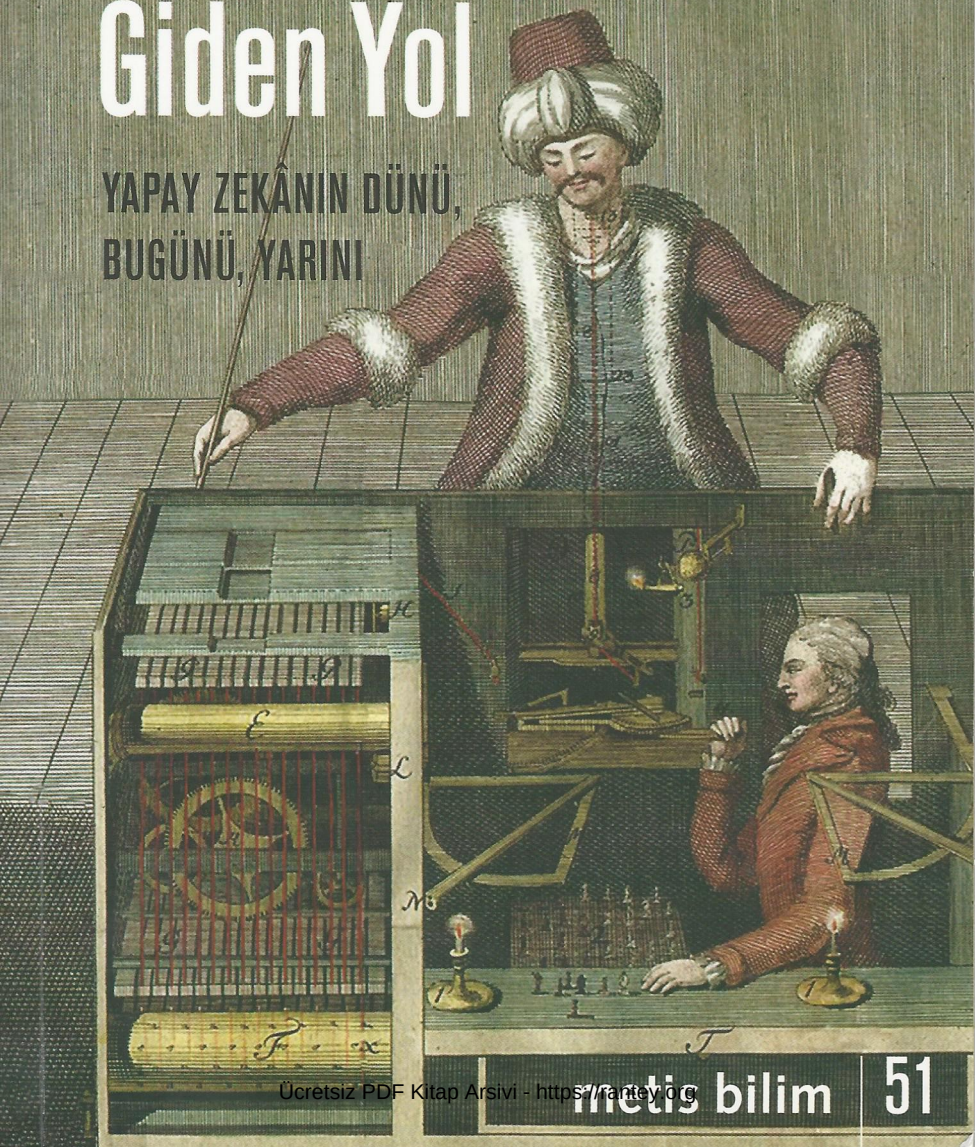
• Telegram: https://t.me/birkitapbir

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

MICHAEL WOOLDRIDGE

# Bilinçli Makinelere Giden Yol

## YAPAY ZEKÂNIN DÜNÜ, BUGÜNÜ, YARINI





metiş bilim 51

MICHAEL WOOLDRIDGE

# Bilinçli Makinelere Giden Yol

1966 yılında İngiltere'nin Wakefield şehrinde doğan Michael John Wooldridge, lisans eğitimini ve doktorasını Manchester Üniversitesi'nde tamamladı. 2001-05 yıllarında Liverpool Üniversitesi'nin Bilgisayar Bilimleri Bölümü'nün başkanlığını, 2008-11 yıllarındaysa aynı üniversitenin Elektrik Mühendisliği, Elektronik ve Bilgisayar Bilimleri Bölümü'nün başkanlığını yapan Wooldridge, 2014-21 yıllarında da Oxford Üniversitesi'nin Bilgisayar Bilimleri Bölümü'nün başkanlığını yürüttü. Halen Oxford Üniversitesi'nde bilgisayar bilimleri profesörü olarak hizmet veren ve araştırmalarında yapay zekânın çeşitli veçhelerine odaklanan Wooldridge, uzun ve verimli kariyeri boyunca pek çok ödül aldı. Bunlardan en yakın tarihli ikisini saymak gerekirse, yapay zekâ alanındaki katkılarından ötürü 2021 yılında uluslararası Yapay Zekâ Geliştirme Derneği'nin verdiği Üstün Eğitmen Ödülü'ne, bilgi işlem alanına yaptığı katkılardan ötürü de 2020 yılında Britanya Bilgisayar Topluluğu'nun verdiği Lovelace Madalyası'na layık görüldü. 300'den fazla akademik makalenin yanı sıra yapay zekâyla bağlantılı konularda ders kitapları da kaleme alan Wooldridge'in bir popüler bilim kitabı daha bulunuyor: *Artificial Intelligence: Everything You Need to Know About the Coming AI* (Yapay Zekâ: Yaklaşmakta Olan YZ Hakkında Bilmeniz Gereken Her Şey; 2018).

Metis Yayınları
İpek Sokak 5, 34433 Beyoğlu, İstanbul
e-posta: info@metiskitap.com
www.metiskitap.com
Yayınevi Sertifika No: 43544

Bilinçli Makinelere Giden Yol
Yapay Zekânın Dünü, Bugünü, Yarını
Michael Wooldridge

İngilizce Basımı:
The Road to Conscious Machines
The Story of Artificial Intelligence
Pelican Books, 2020



AnatoliaLit Ajans, İstanbul aracılığıyla
Felicity Bryan Associates Ltd ile yapılan
sözleşme temelinde yayımlanmıştır.



İlk Basım: Ekim 2022

Yayıma Hazırlayan: Özde Duygu Gürkan

Kapak Resmi: Wolfgang von Kempelen,
satranç otomatı "Türk", 1769.
Kapak Tasarımı: Emine Bora, Semih Sökmen

Dizgi ve Baskı Öncesi Hazırlık: Metis Yayıncılık Ltd.
Baskı ve Cilt: Yaylacık Matbaacılık Ltd.
Fatih Sanayi Sitesi No. 12/197 Topkapı, İstanbul
Matbaa Sertifika No: 44865

ISBN-13: 978-605-316-268-1

MICHAEL WOOLDRIDGE

# Bilinçli Makinelere Giden Yol

## YAPAY ZEKÂNIN DÜNÜ, BUGÜNÜ, YARINI

Çeviren: Özge Çelik

metis

Oxford Üniversitesi, Hertford College'ın
Müdürüne, Akademisyen ve Araştırmacılarına
ve Öğrencilerine

# İçindekiler

**EKLER**

# Teşekkür

BU PROJENİN en başından itibaren beni hep destekleyip teşvik eden, tavsiyelerini hiç eksik etmeyen temsilcim Felicity Bryan ve editörüm Laura Stickney'ye ne kadar teşekkür etsem az.

Ian J. Goodfellow, Jonathon Shlens ve Christian Szegedy'ye "Rakip Örneklerin Açıklanması ve Kontrol Altına Alınması" başlıklı çalışmalarındaki "panda/gibon" görsellerini kullanmama izin verdikleri için; Peter Millican'a birlikte yazdığımız bir makalenin malzemesinin bir kısmını bu kitaba uyarlayarak kullanmama izin verdiği için; Subbarao Kambhampati'ye bazı temel argümanlarım hakkında çok faydalandığım yorumlar yaptığı ve Pascal'ın kumarına dikkatimi çektiği için; Nigel Shadbolt'a tartışmalarımız, desteği ve yapay zekâ tarihiyle ilgili olarak aktardığı müthiş anekdotlar için; ve Reid G. Smith'e MYCIN sistemini anlatırken derste kullandığı bazı malzemeleri bu kitaba uyarlayarak kullanmama izin verdiği için gerçekten çok teşekkür ederim.

O kadar işinin gücünün arasında kitabın taslağını okuyup birbirinden kıymetli değerlendirmeler yapan çalışma arkadaşlarım, öğrencilerim ve bahtsız tanıdıklarıma büyük bir gönül borcum var. Ani Calinescu, Tim Clement-Jones, Carl Benedikt Frey, Paul Harrenstein, Andrew Hodges, Matthias Holweg, Will Hutton, Graham May, Aida Mehonic, Peter Millican, Steve New, André Nilsen, James Paulin, Emma Smith, Thomas Steeples, André Stern, John Thornhill ve Kiri Walden'a müteşekkirim. Yine de kalan hataların sorumlusu ise elbette benim.

2012'den 2018'e kadar dahil olduğum araştırma grubuyla birlikte Avrupa Araştırma Konseyi İleri Düzey Fon 291528'e layık görül-

müş olmak benim için bir onur. Julian Gutierrez ve Paul Harrenstein'ın başında olduğu dinamik, parlak ve müthiş destekleyici bir araştırma grubu, çalışmalarımı alabildiğine verimli, alabildiğine zevkli hale getirdi. Umarım daha çok uzun zaman birlikte çalışırız.

# Giriş

ELİNİZDEKİ KİTABI yazmayı neredeyse yarıladığım sıralarda, bir çalışma arkadaşımla yemeğe çıktık.

"Ne üstünde çalışıyorsun?" diye sordu.

Bir akademisyenden diğerine gayet olağan bir soruydu bu. Şaşılacak bir tarafı yoktu yani. Bilakis bunu bekliyor olmam, çoktan şöyle okkalı bir cevabı hazır etmiş olmam lazımdı.

"Bu seferki biraz farklı," dedim, "yapay zekâya giriş niteliğinde bir popüler bilim kitabı yazıyorum."

"Dünyanın böyle bir kitaba *daha* ihtiyacı var mı gerçekten?" diye burun büktü. "Mevzu ne tam olarak? Konuyu hangi açıdan ele alıyorsun?"

Söyledikleri içime oturdu tabii. Bu sefer akıllıca bir cevapla taşı gediğine koymam lazımdı. Nihayetinde şakayla karışık şöyle diyebildim:

"Yapay zekâ tarihini hüsranla sonuçlanan fikirler üstünden anlatıyorum."

Şöyle bir bana baktı, gülümsemesi yüzünden siliniyordu artık.

"Desene tuğla gibi bir kitap olacak."

Yapay Zekâ (YZ) benim hayatım. Daha 1980'lerin ortasında, henüz öğrenciyken gönül verdim bu işe, hâlâ da büyük bir tutkuyla bağlıyım. YZ'yi böylesine sevmemin nedeni, bu işten parayı kırarım diye düşünmem de değil (hiç fena olmazdı gerçi), dünyamızı değiştireceğine inanmam da (ileride göreceğimiz gibi, *gerçekten* çok önemli etkileri olacağı kanaatindeyim ama). Seviyorum çünkü bence her bakımdan müthiş bir konu. Felsefe, psikoloji, bilişsel bilim, sinirbilim, mantık, istatistik, iktisat ve robotik gibi birbirinden farklı pek

çok disiplinle sürekli alışveriş halinde. Seviyorum çünkü insanlığın hali ve *Homo sapiens* olarak durumumuz hakkındaki en temel soruları ele alıyor: İnsan olmak ne demek? İnsanlar hakikaten biricik midir?

## YZ Nedir Ne Değildir?

Amacım öncelikle YZ'nin ne olduğunu ve, belki daha da önemlisi, ne olmadığını anlatmak. Buna biraz şaşırıp, iyi de bu ortada değil mi zaten diyebilirsiniz. Uzun vadede YZ hayali hiç şüphesiz, zekâya dayalı eylemlerde bulunma açısından insanınki kadar geniş bir kapasiteye sahip, yani tıpkı biz insanlar gibi özfarkındalık sahibi, bilinçli ve otonom makineler inşa etmektir. YZ hayalinin bu versiyonuna bilimkurgu filmlerinde, dizilerde veya kitaplarda muhakkak denk gelmişsinizdir.

Söz konusu YZ versiyonu sezgisel olarak bildiğimiz bir şey, malumun ilamı gibi gelir. Halbuki, ileride göreceğimiz gibi, bunun gerçekte ne anlama geldiğini anlamaya çalışırken pek çok güçlükle karşılaşırız. İşin aslına bakarsanız, yaratmak istediğimiz şeyin tam olarak ne olduğunu veya insanda nasıl mekanizmalar sonucu ortaya çıktığını anlıyor değiliz. Ayrıca YZ'nin amacının sahiden bu olduğu konusunda bir uzlaşı falan da yoktur. Bilakis gayet tartışmalı bir meseledir bu. Böyle bir YZ'nin –arzu edilirliği şöyle dursun– yapılabilir olup olmadığı konusunda bile bir mutabakat yoktur.

Dolayısıyla *büyük hayal* olarak niteleyebileceğimiz bu YZ versiyonunu doğrudan ele almak bayağı zordur. Buradan harika kitaplar, müthiş filmler ve video oyunları çıksa da, YZ araştırmaları esasen bu sularda yüzmez. Büyük hayal hiç şüphesiz son derece derin felsefi soruları gündeme getirir. Gelgelelim bu YZ versiyonu hakkında yazılanların çoğu spekülasyondan başka bir şey değildir, hatta bir kısmı evlere şenliktir. YZ her zaman parlak biliminsanları kadar kaçık tipleri, şarlatanları ve üçkâğıtçıları da cezbetmiştir.

Öyle veya böyle, YZ hakkındaki genelgeçer tartışmaların ve basının bitmek bilmez YZ sevdasının, ayrıca YZ hakkındaki haberlerde temcit pilavı gibi ısıtılıp ısıtılıp önümüze konan kaygı verici distop-

yaların (YZ işimizi elimizden alacak, bizden daha akıllı olup çıkacak, üstün zekâlı YZ zıvanadan çıkıp insanlığın kökünü kazıyacak) odağında bu büyük hayalden başka bir şey yoktur. Anaakım medyada YZ hakkında yazıp çizenlerin çoğu konuya yeterince hâkim değildir veya dünyadan bihaberdir. Yazdıklarının büyük bölümü, eğlenceliyse de, teknik açıdan çerçöptür.

Elinizdeki kitapta işte bu anlatıyı değiştirmek, YZ'nin *gerçekte* ne olduğunu, YZ araştırmacılarının *gerçekte ne üstünde* ve *nasıl* çalıştığını anlatmak istiyorum. Yakın gelecekte, gerçek YZ bu büyük hayalden oldukça farklı olacak. Onun kadar dikkat çekici değil belki ama, ilerleyen bölümlerde göreceğimiz üzere, başlı başına heyecan verici. Bugün anaakım YZ araştırmaları, halihazırda insan beyni (ayrıca, potansiyel olarak, insan bedeni) gerektiren ve alışıldık bilgi işlem tekniklerinin bir çözüm sunmadığı belli başlı görevleri yerine getirecek makineler inşa etmeye yoğunlaşıyor. Yaşadığımız yüzyıl bu alanda önemli gelişmelere tanık oldu, YZ'nin günümüzde böylesine el üstünde tutulması biraz da bundan. Yirmi sene önce, otomatikleştirilmiş çeviri araçları bilimkurgu filmlerinde görebileceğiniz türden bir YZ teknolojisiydi; halbuki son on yılda, gündelik gerçekliğin alelade uygulamalarından biri haline geldi. Bütün kısıtlarına rağmen bu çeviri araçları her gün dünyanın dört bir yanında milyonlar tarafından kullanılıyor. Önümüzdeki on yıl içinde, sözün gerçek zamanlı olarak isabetli bir biçimde tercüme edildiğini, ayrıca yeni yeni artırılmış gerçeklik araçlarının, içinde yaşadığımız dünyayı algılama, anlama ve bu dünyayla ilişki kurma tarzlarımızı değiştirdiğini göreceğiz. Sürücüsüz otomobiller gerçekçi bir ihtimal olarak ufukta beliriyor. Sağlık alanını dönüştürecek, muhtemelen hepimizin yararlanacağı uygulamalar halihazırda kullanılmaya başlanmış gibi görünüyor: YZ sistemlerinin bir röntgen veya ultrason filmindeki tümör gibi anomalileri teşhis etmede insandan daha iyi olduğu anlaşılıyor; giyilebilir/takılabilir teknoloji, YZ ile birlikte, sağlık durumumuzu düzenli olarak takip etme imkânı sunuyor; kalp rahatsızlığı, stres, hatta demans konusunda erken uyarıda bulunuyor. YZ araştırmacıları *tam olarak* böyle şeyler üstünde çalışıyor işte. Ve YZ konusunda beni heyecanlandıran da *tam olarak* böyle şeyler. Dola-

yısıyla YZ anlatısının asıl bunlarla ilgili olması gerektiğini düşünüyorum.

Günümüzde YZ'nin durumunu ve genel itibarıyla o büyük hayalle neden ilgilenmediğini anlamak istiyorsak, YZ yaratmanın neden zor olduğunu da iyice anlamamız lazım. Son altmış yılda YZ'ye müthiş bir emek harcandı (ve alana ciddi miktarda araştırma fonu akıtıldı). Yine de üç vakte kadar hepimizin robot hizmetkârları olacakmış gibi görünmüyor ne yazık ki. YZ'nin zorluğu nereden geliyor öyleyse? Bu sorunun cevabını anlamak için, önce bilgisayarın temelde ne olduğunu ve neler yapabildiğini bir anlamamız lazım. Bu da bizi matematik alanındaki en derin sorulara ve yirminci yüzyılın en büyük zekâlarından biri olan Alan Turing'in çalışmalarına getiriyor.

## YZ'nin Tarihi

İkinci amacım, başlangıcından itibaren YZ'nin hikâyesini anlatmak. Her hikâyenin bir olay örgüsü olmalıdır ve var olan bütün hikâyeler için aslında sadece yedi temel olay örgüsü olduğu söylenir. YZ'ye cuk oturan olay örgüsü bunlardan hangisi o zaman? Meslektaşlarımın çoğu bunun "Fakirlikten Zenginliğe" türü bir olay örgüsü olmasını canıgönülden isterdi muhtemelen, ki birkaç akıllı (veya düpedüz şanslı) YZ'ci için gerçekten böyle olmuştur da. Bununla beraber, ileride anlaşılacak nedenlerle, YZ hikâyesini rahatlıkla "Canavarı Alt Etmek" türüne de dahil edebiliriz tabii. Bu örnekte canavar, pek çok YZ probleminin çözümünün neden göz korkutucu derecede zor olduğunu gözler önüne seren, bilgi işlemsel karmaşıklık denen soyut bir matematik teorisidir. "Arayış" türünde bir olay örgüsü de olur aslında, çünkü YZ'nin hikâyesi Kutsal Kâse'nin peşine düşen ortaçağ şövalyelerininkini andırır: yoğun dini duygular, iflah olmaz bir iyimserlik, yanıltıcı ipuçları ve hüsran. Ama nihayetinde, YZ'ye en çok uyan olay örgüsü "En Dibi Görüp Zirveye Tırmanmak"tır, çünkü daha yirmi sene öncesine kadar akademideki itibarına şüpheyle yaklaşılan ve çok dar bir kesime hitap ettiği düşünülen YZ, bugün bilimin en canlı, en takdir gören alanlarından biri merte-

besine yükselmiş durumdadır. Gerçi YZ'nin olay örgüsünü "Zirveye Tırmandıktan Sonra En Dibi Görüp Tekrar Zirveye Çıkmanın Ardından Tepetaklak Olup Sonunda Bir Kere Daha Zirveye Yükselmek" diye nitelendirmek daha doğru olabilir. Yarım yüzyılı aşkın sürede, YZ pek çok araştırmaya konu oldu, araştırmacılar kim bilir kaç kere zeki makineler hayalini gerçekleştirmemizi sağlayacak çığır açıcı atılımlar yaptıklarını iddia ettiler, ama her seferinde de bunların nihayetinde aşırı iyimser temennilerden başka bir şey olmadığı ortaya çıktı. Sonuç olarak YZ patlama ve çöküş döngüleriyle maluldür. Son 40 senede böyle en az üç döngü yaşandı. Son 60 yılda pek çok noktada öylesine şiddetli bir çöküş yaşandı ki YZ'nin sağ çıkamayacağı düşünüldü, ancak her seferinde tekrar ayağa kalkmayı başardı. Bilimi cehaletten aydınlanmaya sürekli bir ilerleme olarak tasavvur ediyorsanız eğer, duyacaklarınıza bayağı bir şaşıracaksınız.

Bugün bir kere daha YZ alanında patlama yaşanıyor, heyecan dorukta. Beklentilerin tavan yaptığı böyle bir zamanda, YZ'nin inişli çıkışlı hikâyesinin tekrar tekrar anlatılması elzem geliyor bana. Araştırmacılar bir kere daha YZ'yi kaderini gerçekleştirmeye sevk edecek sihirli bileşeni keşfettiklerini düşünüyorlar. Soluksuz, heyecanla yapılan mevcut YZ tartışmaları da tam olarak bu çerçevede ilerliyor. Google'ın CEO'su Sundar Pichai'nin şu sözleri haber oldu: "YZ insanlığın üstünde çalıştığı en önemli şeylerden biri. Sözgelimi elektrikten veya ateşten bile daha önemli belki."[1] Bundan önce de Andrew Ng, YZ'nin "yeni elektrik"[2] olduğu minvalinde laflar etmişti. Bu kibrin geçmişte neyle sonuçlandığını unutmamamız lazım. Evet, gerçekten ciddi ilerlemeler oldu, ve evet, bunlar hakikaten heyecan verici, kutlamaya değer şeyler, ama nihayetinde yolun sonuna, yani bilinçli makinelere ulaşılmış değil. Bir YZ araştırmacısı olarak 30 yıllık tecrübem, bu alanla ilgili olarak ortaya atılan iddialar konusunda saplantı derecesinde ihtiyatlı davranmayı öğretti bana. Çığır açan yenilik iddialarına gayriihtiyari şüpheyle yaklaşıyorum. YZ'ye belki de her şeyden önce bolca tevazu zerk etmek gerekiyor. Bu noktada, Antik Roma'nın *auriga*'larıyla ilgili hikâye aklıma geliyor hemen: Muzaffer generale kentteki zafer yürüyüşü sırasında eşlik eden bir *auriga* (köle), yürüyüş boyunca generalin kulağına

tekrar tekrar şöyle fısıldarmış: *Memento homo*, "unutma, sadece insansın sen".

Bu yüzden kitabın ikinci kısmını günahıyla sevabıyla YZ'nin hikâyesini geniş bir kronolojide anlatmaya ayırdım. YZ'nin hikâyesi II. Dünya Savaşı sonrasında ilk bilgisayarların yaratılmasının hemen ardından başlıyor. YZ'nin patlama yaptığı her dönemi tek tek ele alacağız: önce YZ'nin Altın Çağı, dizginlenemeyen bir iyimserlik dönemidir bu, bir süreliğine farklı farklı cephelerin hepsinde birden hızlı bir ilerleme kaydediliyormuş gibi görünmüştür; sonra "bilgi çağı", bu dönemin büyük fikri dünya hakkındaki bütün bilgimizi makinelere vermektir; ve yakın geçmişten günümüze uzanan davranışsal dönem, bu periyotta YZ'nin odağında robotların olması gerektiği savunulur. Bunların her birinde, o dönemi şekillendiren fikirler ve insanlar üstünde duracağız.

## YZ'nin Geleceği

Hüsrana uğrayan fikirlerle dolu uzun bir tarihin bağlamında, bugün YZ ile ilgili her şey etrafında bir yaygara koparıldığını anlamak (ve buna paralel olarak heyecanımızı biraz yatıştırmak) önemli; diğer taraftan, şu an itibarıyla YZ hakkında iyimser olmak için gayet haklı gerekçeler olduğu da ortada. Sahiden de bilimsel olarak çığır açıcı gelişmeler yaşandı ve bu gelişmeler son on yılda, "büyük veri"nin erişilebilir hale gelmesi ve bilgisayar gücünün ucuzlamasıyla birlikte, alanı kuranların adeta mucize gözüyle bakıp sevinçle karşılayacağı türden YZ sistemlerini mümkün hale getirdi. Dolayısıyla üçüncü amacım da YZ sistemlerinin şu an için bilfiil neler yapabildiğine ve çok geçmeden neler yapabilir hale geleceğine –ve kısıtlarına– dikkat çekmek olacak.

Bu da YZ ile ilgili korkular konusunda bir tartışmaya getiriyor bizi. YZ hakkındaki genelgeçer tartışmalarda, YZ'nin dünyayı ele geçirmesi gibi distopik senaryoların hâkim olduğundan bahsetmiştik. YZ alanındaki güncel gelişmeler ise *gerçekten* kaygılanmamız gereken konuları gündeme getiriyor: sözgelimi YZ çağında istihdamın doğası ve YZ teknolojilerinin insan haklarını nasıl etkileyebile-

ceği. Fakat robotların dünyayı ele geçirip geçirmeyeceği vb. tartışmalar manşetlere taşınınca asıl önemli meseleler arka planda kalıyor. Dolayısıyla anlatıyı bu açıdan da değiştirmek istiyorum. Asıl ilgilenmemiz gereken meseleler üstünde durarak, nelerden korkmamız nelerden korkmamamız gerektiğini apaçık ortaya koymaya çalışacağım.

Son olarak, biraz da eğlenelim istiyorum tabii. Bu yüzden son bölümde büyük YZ hayalini, özfarkındalık sahibi, bilinçli ve otonom makineleri ele alacağız. Hayalin ayrıntılarına inip, bunu gerçekleştirmek ne anlama gelir, böyle makineler neye benzer, insan gibi olur mu yoksa olmaz mı diye soracağız.

## Bu Kitabı Nasıl Okuyalım?

Kitabın kalanı şu kapsamlı amaçlar etrafında şekillendi: YZ'nin ne olduğunu ve neden zor olduğunu anlatmak, YZ'nin hikâyesini yani patlama yaptığı dönemlerde başı çeken fikirleri ve insanları anlatmak, ve son olarak şu anda YZ'nin neler yapabileceğini ve neler yapamayacağını ortaya koymak, biraz da uzun dönem YZ ihtimalleri yani bilinçli makinelere giden yoldan bahsetmek istiyorum.

Bu kitabı yazmanın en keyifli taraflarından biri de, genelde önemli olmakla beraber biraz da bıktırıcı olan akademik teamülleri bir yana bırakmaktı. Dolayısıyla, en önemli noktalarda kaynak gösterdim tabii ama, genel olarak nispeten az referans verdim.

Kapsamlı referanslar vermekten kaçınmanın yanı sıra teknik detaylardan daha doğrusu *matematiksel* detaylardan da uzak durdum. Bu kitabı okuyarak, tarihi boyunca YZ'ye yön veren başlıca fikir ve kavramları genel hatlarıyla kavrayacağınızı umuyorum. Aslına bakarsanız bu fikir ve kavramların çoğu doğası itibarıyla bayağı matematikseldir, ama Stephen Hawking'in bir kitaptaki her denklemin okur sayısını yarı yarıya düşüreceği yolundaki düsturunun gerçek hayatta bir karşılığı olduğunun da gayet farkındayım. Dolayısıyla daha fazlasını isteyenler için, kitabın sonuna biraz daha teknik ayrıntıya giren bazı ekler koydum, ayrıca okuma önerilerinde bulundum.

Kitabı yazarken *epey* seçici davrandım. Pek bir seçeneğim yoktu doğrusu. YZ uçsuz bucaksız bir alan; son altmış yıl içinde YZ'yi etkileyen farklı farklı fikirlerin, geleneklerin ve düşünce ekollerinin hepsini birden hakkıyla ele almak imkânsızdı. Bu yüzden ben de belli başlı noktaları, YZ denen girift dokumanın başlıca düğümlerini öne çıkarmaya çalıştım.

Son olarak, elinizdeki bir ders kitabı değil. Yani bu kitabı okuyarak, yeni bir YZ şirketi kurmanıza veya Google ya da Facebook ekibine katılmanıza imkân sağlayacak becerilerle donanmış olmayacaksınız. Daha ziyade, YZ'nin ne olduğuna ve gidişatına dair genel bir fikir edineceksiniz. Umarım bu kitabı okumak, YZ *gerçeği* hakkında doğru dürüst bilgi sahibi olmanıza katkıda bulunur, ve umarım kitabı okuduktan sonra, bu anlatıyı değiştirmeme yardım edersiniz.

*Oxford, Mayıs 2019*

BİRİNCİ KISIM

# YZ Nedir?

# 1
# Turing'in Elektronik Beyinleri

Sizi şu soruyu dikkate almaya davet ediyorum:
"Makineler düşünebilir mi?"

ALAN TURING (1950)

HER HİKÂYENİN bir yerden başlaması gerekir ve YZ'nin hikâyesi de birçok yerden başlatılabilir, çünkü YZ hayali öyle veya böyle kadim bir hayaldir.

Bu hikâyeyi, demire can veren tanrı Hephaistos'la başlatabiliriz.

Bir başhahamın 1600'lerin Pragı'nda, kentteki Yahudi nüfusu antisemitlerin saldırılarından korusun diye kilden büyülü bir Golem yaratmasıyla başlatabiliriz.

James Watt'ın 18. yüzyıl İskoçyası'nda inşa ettiği buhar motorları için regülatör dediği yaratıcı bir otomatik kontrol sistemi tasarlayarak günümüz kontrol teorisinin temellerini atmasıyla başlatabiliriz.

Genç Mary Shelley'nin 19. yüzyıl başlarında bir yaz İsviçre'de hava feci bozduğundan bir villaya kapanmak zorunda kalıp eşi şair Percy Bysshe Shelley ile bednam aile dostları Lord Byron'ı eğlendirmek için Frankenstein'ın hikâyesini yazmasıyla başlatabiliriz.

Lord Byron'ın kızı Ada Lovelace'in 1830'larda mekanik hesap yapma makinelerinin huysuz mucidi Charles Babbage ile dostluk kurmasıyla, bu dostluğun da parlak bir genç olan Ada'nın makineler nihayetinde yaratıcı olabilir mi diye kafa yormaya başlamasına vesile olmasıyla başlatabiliriz.

On sekizinci yüzyılın otomat sevdasıyla canlılık yanılsaması

yaratmak için özel olarak tasarlanmış makinelere duyulan ilgiyle başlatabiliriz.

Gördüğünüz gibi seçenek çok, ama bence YZ hikâyesinin başlangıcı, bizzat bilgi işlemin hikâyesinin başlangıcıyla kesişiyor, ki bunun başlangıç noktasını net bir biçimde biliyoruz: King's College, Cambridge, 1935 ve sıradışı bir öğrenci, parlak bir genç, Alan Turing.

## Cambridge, 1935

Düşünsenize, bugün gelmiş geçmiş en ünlü matematikçilerden biri sayılan Alan Turing'in ismi, daha 1980'lere kadar matematik ve bilgisayar bilimi alanları dışında neredeyse hiç bilinmiyordu. Bu isim matematik ve bilgi işlem alanlarındaki öğrencilerin bir şekilde kulağına çalınıyordu belki ama Turing'in tam olarak neler yaptığı veya trajik bir biçimde çok vakitsiz hayatını kaybettiği genelde bilinmiyordu. Bunun nedenlerinden biri, Turing'in en önemli çalışmalarından bazılarının II. Dünya Savaşı sırasında Birleşik Krallık hükümeti için gizlice yapılmış olması, dolayısıyla bu önemli çalışmaların sonuçlarının 1970'lere kadar gün yüzü görmemesiydi.[1] Bir diğeri de dönemin önyargılarıydı tabii; Turing geydi ve o zamanlar Birleşik Krallık'ta eşcinsellik suç sayılıyordu. 1952'de hakkında bir dava açıldı ve genel ahlaka karşı suçlardan hüküm giydi. Ceza olarak, cinsel arzularını azaltması beklenen ne idiği belirsiz bir hormon ilacı verildi Turing'e, yani bir tür "kimyasal kastrasyon"a uğradı. İki sene sonra da öldü, görünüşe göre intihar etmişti, daha 41 yaşındaydı.[2]

Turing'in hikâyesini bugün hepimiz, olması gerektiği kadar değilse de biraz biliyoruz artık. Bu hikâyenin en bilinen tarafı, II. Dünya Savaşı sırasında Müttefik Devletler'in şifre kırma merkezi olan Bletchley Park'ta yaptığı çalışmalarla ilgili. 2014 yapımı bir Hollywood filmi olan *The Imitation Game / Enigma*'nın (tarihsel olguları doğru bir biçimde vermekten ziyade salt bir görsel şölen sunsa da) bunda payı büyük. Müttefiklerin zaferinde belirleyici bir rol oynaması bakımından Turing'in bu çalışmaları hiç kuşkusuz büyük bir alkışı hak ediyor. Fakat YZ araştırmacılarının ve bilgisayar bilimci-

lerin Turing'e böyle hürmet etmesinin nedenleri çok daha farklı. Turing neresinden baksanız bilgisayarın mucididir ve bunun hemen ardından YZ alanını icat eden de yine çok büyük ölçüde o olmuştur.

Pek çok açıdan kayda değer biri olan Turing'le ilgili en dikkat çekici gerçeklerden biri bilgisayarı tesadüfen keşfetmiş olmasıdır. 1930'larda Cambridge Üniversitesi'nde matematik öğrencisi olan Turing, dönemin belli başlı matematik problemlerinden *Entscheidungsproblem*'i çözmek gibi zamanın ötesinde bir hedef koymuştu kendine (etkileyici olduğu ölçüde göz korkutan bu terim Almancada "**karar problemi**" anlamına gelir). Matematikçi David Hilbert'ın 1928'de ortaya attığı bu problemde, belli bir tarife uyarak cevaplanamayacak soruların olup olmadığı sorulur. Burada Hilbert'ın kastettiği "Tanrı var mı?" veya "Hayatın anlamı nedir?" gibi sorular değildir elbette, matematik alanındaki karar problemleridir. Karar problemleri evet veya hayır şeklinde bir cevabı olan matematiksel sorulardır. Birkaç örnek verecek olursak:

- 2 + 2 = 4 müdür?
- 4 x 4 = 16 mıdır?
- 7919 asal sayı mıdır?

Bu karar problemlerinin hepsinin cevabı "evet"tir. İlk ikisinin cevabının evet olduğu gayet açık zaten, değil mi? Üçüncünün cevabını bulmak içinse –asal sayılarla sağlıksız, saplantılı bir ilişkiniz yoksa şayet– birazcık uğraşmanız lazım. Öyleyse bu son soru üstünde biraz duralım.

Asal sayı, hatırlarsınız, sadece kendisine ve 1'e tam olarak bölünebilen tamsayıdır. Şimdi, bunu aklınızda tutarak, son sorunun cevabını –en azından teoride– kendi başınıza bulabilirsiniz. 7919 gibi büyük bir sayı söz konusuysa, bu sorunun cevabını bulmanın sıkıcı olmakla beraber belli, dosdoğru bir yolu vardır: 7919'u tam olarak bölebileceğini düşündüğünüz her sayıyı deneyip bakarsınız. Ve bunları dikkatlice denedikten sonra, hiçbirinin bölmediğini görürsünüz: 7919 gerçekten de asal sayıdır.[3]

Burada önemli olan böyle soruları cevaplamanın kesin ve net metotlarının olmasıdır. Söz konusu metotlar belli bir *zekâ* gerektir-

mez. Bunlar olsa olsa birer *tariftir*, mekanik bir biçimde tekrar etmek yeter. Cevabı bulmak için yapmamız gereken tek şey tarife *bire bir* uymaktır.

Soruyu (yeterli zaman verildiği takdirde) cevaplamayı garanti eden bir tekniğimiz olduğu sürece, "*n* asal sayı mıdır?" türü soruların **karar verilebilir** olduğunu söyleriz. Burada vurgulamaya çalıştığım özetle şu: "*n* asal sayı mıdır?" şeklinde bir soruyla karşılaştığımızda, yeterince zaman verildiği takdirde, cevabı kesinlikle bulabileceğimizi biliriz. Tarife uyarsak, sonunda doğru cevabı buluruz.

*Entscheidungsproblem*, örneklerini gördüğümüz türden matematiksel karar problemlerinin hepsi karar verilebilir midir veya –ne kadar uzun süre uğraşırsanız uğraşın– cevabını nasıl bulacağınızın hiçbir tarifi *olmayan* sorular da var mıdır diye sorar.

Çok temel bir sorudur bu, matematiğin sadece birtakım tariflere uymaya indirgenip indirgenemeyeceğine kafa yorar. 1935'te kendisine bu temel soruyu cevaplamak gibi göz korkutucu bir hedef koymuş olan Turing, hedefine baş döndürücü bir hızla ulaşmayı başarmıştır.

Derin matematik problemlerinin çözüm yollarının karmaşık denklemler ve uzun ispatlardan geçtiğini düşünürüz. Bazen sahiden de öyledir; sözgelimi İngiliz matematikçi Andrew Wiles gayet iyi bilindiği üzere Fermat'nın Son Teoremi'ni ispatladıktan ancak yıllar sonra matematik camiası bu yüzlerce sayfalık ispatı anlayıp hakikaten doğru olduğuna kanaat getirmiştir. Standardın bu olduğunu düşününce, Turing'in *Entscheidungsproblem* için bulduğu çözüm kesinlikle çok acayiptir.

Her şey bir yana, Turing'in ispatı kısa ve nispeten erişilebilirdir (genel çerçeveyi çizdikten sonra, ispat aslında birkaç satırdır). En önemlisi de Turing *Entscheidungsproblem*'i çözmek için bire bir uyulacak bir tarif fikrini kesinleştirmesi gerektiğini fark etmiştir. Bunun için de bir matematiksel problem çözme makinesi icat etmiştir. Bugün böyle makinelere onun anısına **Turing makinesi** diyoruz. Bir Turing makinesi, asal sayılarla ilgili örneğimizde olduğu gibi, belli bir tarifin matematiksel betimlemesidir; yaptığı tek şey, tasarımında temel alınan tarife uymaktır. Öte yandan şunu vurgulamak

isterim ki, her ne kadar Turing bunları "makine" olarak adlandırdıysa da, soyut bir matematiksel fikirden başka bir şey değildi Turing makineleri. Derin bir matematik problemini *bir makine icat ederek* çözme fikri en hafif tabirle alışılmışın dışındaydı, o dönemde matematikçilerin çoğu bunu hayretle karşılamıştır herhalde.

Turing makineleri çok güçlü şeylerdir. Aklınıza gelip gelebilecek her türlü matematiksel tarif bir Turing makinesi olarak kodlanabilir. Bütün matematiksel karar problemleri bir tarife uyularak çözülebiliyorsa, herhangi bir karar problemi için bu problemi çözecek bir Turing makinesi tasarlayabilmeniz gerekir. Hilbert'ın problemini çözmek için yapılması gereken tek şey, herhangi bir Turing makinesinin cevaplayamayacağı bir karar problemi olduğunu göstermekti. Ve Turing de tam olarak bunu yaptı işte.

Turing'in bir sonraki numarası, makinelerinin *genel amaçlı* problem çözme makineleri haline getirilebileceğini göstermekti. Verdiğiniz *her* tarife uyan bir Turing makinesi tasarladı. Bugün bu genel amaçlı Turing makinelerine **Evrensel Turing Makinesi**[4] diyoruz: Bir bilgisayar, en temel özelliklerini düşününce, cisme kavuşmuş bir Evrensel Turing Makinesi'dir. Bir bilgisayarın çalıştırdığı programlar, asal sayılar örneğindeki gibi, sadece birer tariftir.

Esas konumuz bu olmasa da, Turing'in yeni icadını kullanarak *Entscheidungsproblem*'i nasıl çözdüğünden bahsetmekte fayda var. Müthiş yaratıcı bir çözüm bu, ayrıca YZ'nin nihayetinde mümkün olup olmadığı sorusuyla da ilgili.

Turing, makinelerinin *diğer Turing makineleri hakkındaki soruları cevaplamak* üzere programlanabileceği kanaatindeydi. Şu karar problemini düşündü: Elimizde bir Turing makinesi ve ilgili girdi varken, cevabı bulduğunda makinenin nihayetinde duracağı garanti mi yoksa çalışmaya sonsuza dek devam edebilir mi? Çok daha kapsamlı olmakla beraber önceki örneklerimizdeki gibi bir karar problemi bu. Şimdi, bu problemi çözebilecek bir makine olduğunu düşünün. Turing bu soruya cevap verebilen bir Turing makinesinin çelişki yaratacağını anlamıştı. Dolayısıyla bir Turing makinesinin durup durmadığını kontrol etmenin bir tarifi olamaz, bu yüzden de "Bir Turing makinesi durur mu?" sorusu **bir karar verilemez prob-**

lemdir. Turing sadece bir tarife uyarak çözülemeyen karar problemleri olduğunu tespit etmişti. Böylece Hilbert'ın *Entscheidungsproblem*'ini de çözmüştü: Matematik sadece birtakım tariflere uymaya indirgenemez.[5]

Bu sonuç 20. yüzyıl matematiğinin en büyük başarılarından biriydi. Tek başına bu bile Turing isminin matematikçiler arasında ölümsüzleşmesini garantilerken, ispatının bir de yan ürünü olmuştu: genel amaçlı bir problem çözme makinesinin, Evrensel Turing Makinesi'nin icadı. Makinelerini icat ederken, bu makineleri bilfiil inşa etmek gibi bir fikri yoktu Turing'in; ama icat ettikten hemen sonra Turing de başkaları da bunu düşündüler tabii. Konrad Zuse savaş zamanı Münih'te Alman Havacılık Bakanlığı için Z3 diye bir bilgisayar tasarladı. Günümüz bilgisayarlarına pek benzememekle beraber onların pek çok temel bileşenini bünyesinde barındıran bir bilgisayardı bu. Atlantik'in diğer yakasında Pennsylvania'da John Mauchly ve J. Presper Eckert'ın başında olduğu bir ekip askeri topların menzil tablosunu hesaplamak için ENIAC diye bir makine geliştirdi. Parlak bir matematikçi olan Macar John von Neumann'ın yaptığı birtakım ince ayarlarla ENIAC, modern bilgisayarın temel mimarisini ortaya koyan bir makine haline geldi (bildiğimiz türden bilgisayarların mimarisine onun anısına Von Neumann mimarisi diyoruz). Savaştan sonra İngiltere'de Fred Williams ve Tom Kilburn'ün inşa ettiği Manchester Baby dünyanın ticari olarak satılan ilk bilgisayarına, Ferranti Mark 1'e kapı araladı. Turing 1948'de Manchester Üniversitesi'ndeki ekibe katıldı, hatta bu bilgisayarda çalışacak ilk programlardan bazılarını yine bizzat Turing yazdı.

1950'ler itibarıyla modern bilgisayarın bütün temel bileşenleri geliştirilmiş durumdaydı. Turing'in matematiksel tahayyülünü hayata geçiren makineler artık gerçek olmuştu. Böyle bir bilgisayara sahip olmak için gereken tek şey yeterince para ve yeterli büyüklükte bir binaydı (Ferranti Mark 1 için her biri yaklaşık 5 m uzunluğunda, 2,5 m yüksekliğinde ve 1,25 m genişliğinde iki depo gerekiyordu; makinenin 27 kW yani o dönemde üç hanenin ortalama tüketimi kadar elektrik sarfiyatı vardı). Ama o zamandan beri giderek küçülüyor ve ucuzluyorlar tabii.

## Elektronik Beyinler Tam Olarak Ne Yapar?

İkinci Dünya Savaşı'ndan sonra ilk bilgisayarlar inşa edilince, dünyanın dört bir yanındaki gazetelerin akılda kalıcı manşetlere bayılan editörleri bu mucizevi yeni icadı şu sözcüklerle müjdeledi: *elektronik beyin.* Korkutucu derecede karmaşık olan bu makineler görünüşe bakılırsa matematik konusunda göz kamaştırıcı becerilere sahipti, sözgelimi son derece dolambaçlı aritmetik problemlerin onlarcasını bir insanın hayal bile edemeyeceği kadar hızlı ve doğru bir biçimde işleyebiliyorlardı. Bilgisayarlarla pek haşır neşir olmayanlar, böylesine görevleri yerine getirebildiklerine göre bu makinelere üstün bir zekâ bahşedilmiş olmalı diye düşünmüştür herhalde. Böylece *elektronik beyin* cuk oturan bir ifade gibi geldi insanlara ve yerleşti. (Bilgisayarlara yeni yeni ilgi duymaya başladığım 1980'lerin başında bu tür bir dil kullanımına hâlâ rastlayabiliyordunuz.) Halbuki, elektronik beyinler inanılmaz işe yarayan ve insanların müthiş zor bulduğu bir şey yaparlar yapmasına ama, yaptıkları şey *zekâ* gerektirmez. YZ'yi ve neden bu kadar zor olduğunu anlamak için, bilgisayarın tam olarak ne yapmak üzere tasarlandığını –keza neleri yapamayacağını– iyice anlamak lazım.

Unutmayın, Turing makineleri, ve bilgisayar biçimindeki tezahürleri, *talimatlara uyan makinelerden* başka bir şey değildir. Tek amaçları budur, sadece bunu yapmak üzere tasarlanmışlardır, yapabilecekleri tek şey budur. Bir Turing makinesine verdiğimiz talimatlara günümüzde algoritma veya program diyoruz.[6] Çoğu programcı etkileşime girdiği şeyin aslında bir Turing makinesinden ibaret olduğunun farkında değildir muhtemelen. Böyle olmasının gayet anlaşılır bir sebebi var: Turing makinelerini programlamak inanılmaz derecede zahmetli ve sıkıcıdır. İstediğiniz yaştan istediğiniz bilgisayar bilimi öğrencisine sorun, kendi girişiminin nasıl da hüsranla sonuçlandığını anlatıp doğrulayacaktır bunu. Dolayısıyla biz de, Turing makinelerini programlamayı kolaylaştırmak için, Python, Java ve C gibi daha üst düzey diller inşa ediyoruz. Bütün bu diller aslında makinenin kimi nahoş ayrıntılarını programcıdan gizleyerek onu

daha erişilebilir hale getirir. Yine de hâlâ inanılmaz zahmetli ve sıkıcı bir şeydir bu; programlama bu yüzden zordur, bilgisayar programları bu yüzden sık sık çöker ve iyi programcılar bu yüzden iyi kazanır.

Burada size programlama öğretmeye kalkışacak değilim, yine de bilgisayarların ne türden talimatlara uyabildiğine dair genel bir fikir edinmenizde fayda var. Bir bilgisayarın yapıp yapabileceği, kabaca, şu tür talimatlara uymaktır:[7]

- *B*'ye *A* ekle
- Sonuç *C*'den büyükse *D* yap, değilse *E* yap
- *G*'ye kadar tekrar tekrar *F* yap

*Her* bilgisayar programı nihayetinde bu türden bir dizi talimattan ibarettir. Microsoft Word ve PowerPoint böyle talimatlardan ibarettir. Call of Duty ve Minecraft böyle talimatlardan ibarettir. Facebook, Google ve eBay böyle talimatlardan ibarettir. Web tarayıcınızdan Tinder'a, cep telefonunuzdaki bütün uygulamalar böyle talimatlardan ibarettir. *Zekâya dayalı makineler inşa edeceksek, bu makinelerin zekâsı nihayetinde böyle basit, açık talimatlara indirgenebilir olmalıdır.* YZ ile imtihanımız esasen budur. YZ'nin mümkün olup olmadığı sorusu nihayetinde böyle bir dizi talimata uyularak zekâya dayalı davranışın üretilip üretilemeyeceği sorusuyla aynı kapıya çıkar.

Bölümün kalanında, bu gözlemi biraz daha derinlemesine ele almak ve YZ açısından içerimlerini ortaya koymak istiyorum. Ama daha önce, sizi bilgisayarların aslında işe yaramaz şeyler olduğuna inandırmaya çalışan biri pozisyonuna düşmemek adına, bilgisayarların belki de şu aşamada göründüğünden çok daha kullanışlı olduğunu gösteren bazı özelliklere dikkat çekmek istiyorum.

Birincisi, bilgisayarlar *hızlıdır*. Çok çok çok hızlıdır. Şüphesiz herkesin bildiğini düşündüğü bir olgudur bu ama gündelik hayat deneyimlerimizin bilgisayarların ne kadar hızlı olduğunu gerçekten kavramamıza katkı sağladığı da pek söylenemez. Gelin, somutlaştıralım bu önermeyi. Tam hızda çalışan orta halli bir masaüstü bilgisayar, yazı yazma esnasında *her bir saniyede örneğimizdeki gibi*

*talimatların 100 milyar kadarını* yerine getirebilir. Karşılaştırma için galaksimizde yaklaşık 100 milyar yıldız var desek, yine de kafamızda tam bir yere oturtamayabiliriz bunu. O zaman şöyle düşünelim: Diyelim ki bilgisayarın yaptığını kendiniz yapmayı deniyorsunuz, talimatları bizzat yerine getiriyorsunuz. Her 10 saniyede 1 talimatı yerine getirseniz, hiç ara vermeden 7/24 ve 365 gün çalışsanız, bilgisayarın *sadece 1 saniye* içinde yaptığını yapmanız tam *3700 yıl* sürer.

Tariflere uyan bir makine olarak şüphesiz bilgisayara kıyasla *çok* yavaş kalırsınız. Üstelik aranızda önemli bir fark daha olacaktır: Bunca zamanı böyle talimatlara uyarak geçirirken hata yapmamanıza imkân yoktur. Buna karşılık bilgisayarlar çok nadir hata yapar. Programlar sürekli çöküp duruyorsa, bunun sebebi hemen her zaman programı yazanın bir hatasıdır, genelde bilgisayarın kendisinden kaynaklanan bir hata olmaz. Modern bilgisayar işlemcileri gayet güvenilirdir, ortalama 50 bin saate kadar çalışmaları beklenir. Bu kadar saatin her bir saniyesinde on milyarlarca talimatı aynen yerine getirirler.

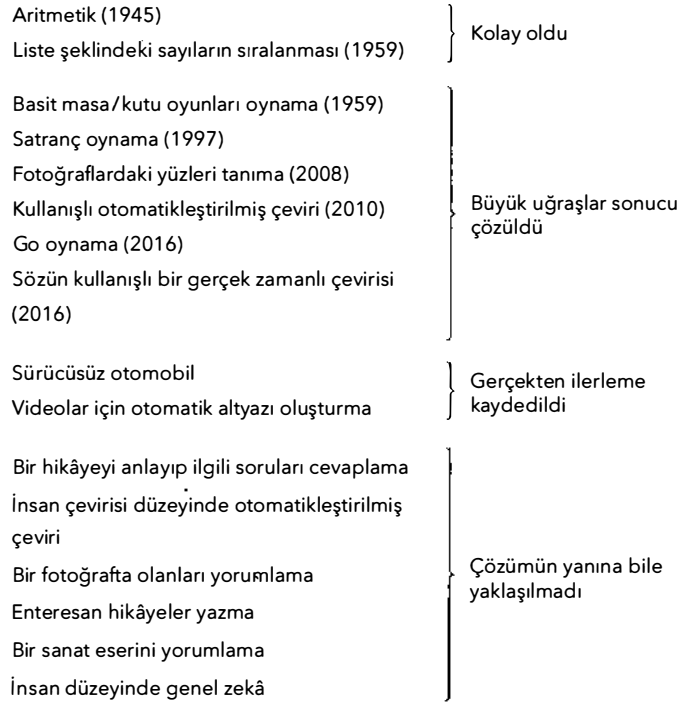
Son olarak, bilgisayarın sadece talimatlara uyan bir makine olması, birtakım kararlar veremeyeceği anlamına gelmez. Bilgisayar elbette karar vermeye *muktedirdir*, sadece ona karar verme işini tam olarak *nasıl* yapacağına dair kesin talimatlar vermemiz gerekir. Ardından, bilgisayar bu talimatları ayarlayıp kendisine uygun hale getirebilir; sadece bilgisayara bunu nasıl yapacağına dair de talimatlar vermemiz gerekir. İleride göreceğimiz gibi, böylece bilgisayar zamanla davranışını değiştirebilir yani *öğrenebilir*.

## YZ Neden Zor?

Demek ki neymiş? Bilgisayarlar gayet *güvenilir* bir biçimde *son derece basit* talimatlara *çok çok hızlı* uyabiliyorlarmış; açık ve net biçimde tanımlandığı takdirde, birtakım *kararlar verebiliyorlarmış*. Şimdi, bilgisayarların bizim için yapmasını isteyebileceğimiz kimi şeyleri bu şekilde kodlamak çok basittir, kimilerini kodlamaksa hiç de kolay değildir. YZ'nin neden zor olduğunu, yıllar içinde görüldü-

ğü üzere alanda ilerleme kaydetmenin neden bir hayli güç olduğunu anlamak için, bu şekilde kodlanması kolay problemlere ve zor problemlere bakmakta fayda var. Şekil 1 bilgisayarların yapmasını isteyebileceğimiz kimi görevleri ve bilgisayarların bunları yapmasını sağlamanın ne derece zor olduğunu gösteriyor.

Listenin en tepesinde aritmetik yer alıyor. Bilgisayarların aritmetik işlemler yapması çok kolay oldu, çünkü her türlü aritmetik işlem (toplama, çıkarma, çarpma, bölme) son derece basit bir tarifle yapılabilir. Şimdi hatırlamıyorsanız bile, vaktiyle size okulda öğre-




**Şekil 1:** Bilgisayarların yerine getirebilmesini istediğimiz kimi görevler zorluk derecesine göre sıralanmıştır. Parantez içinde verilen tarihler problemin yaklaşık ne zaman çözüldüğünü gösteriyor. Bilgisayarların listenin en sonunda yer alan görevleri yerine getirmesinin nasıl sağlanabileceği konusunda şimdilik bir fikrimiz yok.

tilenler de bu türden tariflerdir. Böyle tarifler doğrudan bilgisayar programlarına tercüme edilebilir, dolayısıyla doğrudan aritmetik uygulamaları içeren problemler bilgi işlemin en erken dönemlerinde çözülebilmiştir. (Turing'in 1948'de Manchester Üniversitesi'ndeki ekibe katılınca Manchester Baby için yazdığı ilk program uzun bölme üzerineydi; 20. yüzyılın en derin matematik problemlerinden birini çözdükten sonra okulda öğrendiği aritmetiğe dönmek, Turing açısından tuhaf bir deneyim olmalı.)

Ardından sıralama geliyor; mesela liste şeklinde alt alta yazılmış sayıların küçükten büyüğe sıralanması veya isimlerin alfabetik olarak sıralanması. Bunun hiç de yapay zekâlık bir tarafı yokmuş gibi geliyor, değil mi? Çünkü sıralamanın son derece basit tarifleri vardır. Gelgelelim bu aşikâr tarifler o kadar yavaştı ki insanı afakanlar basıyordu, yani hiç de kullanışlı değillerdi: 1959'da QuickSort diye bir teknik icat edildi, sıralama yapmanın gerçekten randımanlı bir yolu vardı artık. (İcadından bu yana geçen 50 senelik zaman zarfında kimse QuickSort'tan açık ara daha iyisini yapamadı.)[8]

Sıralamadan sonra ise daha uğraştırıcı problemler geliyor. YZ'nin hikâyesi açısından temel bir nedenden dolayı, masa oyunlarını iyi oynamanın ciddi bir zorluk oluşturduğu ortaya çıktı. Masa oyunlarını iyi oynamanın aslında gayet basit ve zarif bir tarifi vardır. Bu tarifin dayandığı **arama** denen tekniği bir sonraki bölümde daha sık duyacağız. Sorun şu ki, arama yoluyla oyun oynamanın basit tarifini programlamak gayet kolay olmakla beraber bu yöntem en ıvır zıvır oyunlar dışında çalışmaz, çünkü çok zaman ve çok bilgisayar belleği gerektirir. Evrendeki her bir atomu kullanarak bir bilgisayar inşa etseydik bile, bu bilgisayarın gücü basit arama (*naive search*) tekniğini kullanarak satranç veya Go oynamaya asla yetmezdi. Aramanın yapılabilir olması için bir şey daha lazım ve, bir sonraki bölümde göreceğimiz gibi, YZ işte tam da bu noktada devreye giriyor.

Bu durum, yani bir problemi nasıl çözeceğimizi teoride bildiğimiz halde ilgili tekniklerin pratikte bir işe yaramamaları çünkü karşılanamayacak kadar büyük miktarlarda bilgi işlem kaynakları gerektirmeleri, YZ alanında hayli sık rastlanan bir durum olduğu için ilgili onlarca araştırmayı da beraberinde getirmiştir.

Listenin devamındakiler pek de bu türden problemler değildir. Fotoğraflardaki yüzleri tanıma, otomatikleştirilmiş çeviri ve sözün kullanışlı bir gerçek zamanlı çevirisi çok daha farklıdır, çünkü alışıldık bilgi işlem teknikleri –masa oyunlarının aksine– böyle problemleri çözmek için nasıl tarif yazılacağı konusunda en ufak bir ipucu bile vermez bize. Bunun için bambaşka bir yaklaşım gerekir. Söz konusu örneklerin her birinde problem, kitabın ilerleyen bölümlerinde adını daha sık duyacağımız, **makine öğrenmesi** denen bir teknikle çözülür.

Listede bir sonraki grupta yer alan sürücüsüz otomobiller etkileyici bir problemdir, çünkü araba kullanmak dümdüz bir iş gibi gelir bize, bu yetinin *zekâ* ile bir ilgisi olduğunu düşünmeyiz. Gelgelelim bilgisayarlara araba kullandırmanın dehşet zor bir iş olduğu ortaya çıkmıştır. Başlıca sorun, arabanın nerede olduğunu ve etrafında neler olup bittiğini anlamasının gerekmesidir. New York'un vızır vızır işleyen bir kavşağında sürücüsüz bir otomobil düşünün: sürekli hareket halinde bir sürü araba, ayrıca yayalar, bisikletliler, yol çalışmaları, trafik levhaları, yol işaretleri. Yağmur veya kar yağıyor belki ya da hava sisli (New York'ta bunların üçü de aynı anda olabilir pekâlâ), işler karışık yani. Böyle bir durumda başlıca zorluk *ne yapacağına karar vermek* (yavaşla, hızlan, sola dön, sağa dön vb.) değildir, daha ziyade *etrafında neler olup bittiğini anlamaktır*: Nerede bulunduğunu, başka hangi araçlar olduğunu, bunların konumunu ve ne yaptığını, yayaların nerede olduğunu vb. tespit etmektir. Bütün bu enformasyona sahipseniz, ne yapmanız gerektiğine karar vermeniz genelde kolay olur. (Sürücüsüz otomobillere özel zorlukları kitabın ilerleyen bölümlerinde ele alacağız.)

Bu grubun ardından, nasıl çözeceğimiz konusunda en ufak bir fikrimizin bile olmadığı problemler gelir. Bir bilgisayar nasıl karmaşık bir hikâyeyi anlayıp bununla ilgili sorulara cevap verebilir? Roman gibi nüans bakımından zengin bir metni nasıl tercüme edebilir? Bir fotoğrafta olup bitenleri nasıl yorumlayabilir? Sadece fotoğraftaki kişileri teşhis etmeyi kastetmiyorum burada, *fotoğrafta olanları bilfiil yorumlamaktan* bahsediyorum. Nasıl enteresan bir hikâye yazabilir veya bir sanat eserini, sözgelimi bir tabloyu yorum-

layabilir? Ve son olarak, bilgisayarlarla ilgili en büyük hayal, genel amaçlı ve insan düzeyinde zekâya dayalı bilgisayarlar yaratmak, bütün bunlar içindeki en büyük zorluk olarak karşımızda durmaktadır.

Öyleyse YZ alanında ilerleme kaydetmek, bilgisayarların Şekil 1'deki görevlerden giderek daha fazlasını yapmasını sağlamak anlamına geliyor. Şekil 1'de zor sayılan görevlerin böyle olmasının nedeni şu ikisinden biridir: Birincisi, belli bir problem için *teoride* iş gören bir tarif, çok fazla hesaplama zamanı ve belleği gerektirmesi nedeniyle *pratikte* çalışmayabilir. Satranç ve Go gibi masa oyunları bu kategoriye girer. İkincisi, problemi nasıl bir tarifle çözebileceğimiz konusunda en ufak bir fikrimiz bile olmayabilir (örneğin yüz tanıma), bu yüzden de yepyeni bir şeye ihtiyacımız vardır (örneğin makine öğrenmesi). Bugün YZ'nin ele aldığı problemlerin çok büyük bir bölümü bu iki kategoriden birine girer.

İsterseniz Şekil 1'deki en zor problem olan genel amaçlı, insan düzeyinde zekâ problemini, YZ'nin o büyük hayalini biraz daha ayrıntılı ele alalım şimdi. Tahmin edeceğiniz üzere bu problem büyük ilgi görmüştür. Ve bu konu üstünde çalışan ilk ve en etkili düşünürlerden biri de kadim dostumuz Alan Turing'den başkası değildir.

## Turing Testi

1940'ların sonuyla 1950'lerin başında ilk bilgisayarların geliştirilmesi, modern bilimin bu harikulade özelliklerinin taşıdığı potansiyel konusundaki kamuoyu tartışmalarını hararetlendirdi. O dönemde, söz konusu tartışmalara yapılan en bilinen katkılardan biri, Massachusetts Teknoloji Enstitüsü'nden (Massachusetts Institute of Technology, MIT) matematik profesörü Norbert Wiener'ın *Sibernetik*'iydi. Kitap, makineler ile hayvanların beyinleri ve sinir sistemleri arasında çok açık paralellikler kuruyor, ayrıca YZ ile ilişkili pek çok fikre değiniyordu. Konuya hâkim ve matematik ehli olanlar dışında okunup kavranması bir hayli zor olmasına rağmen, kamuoyunda büyük ilgi gördü. Makineler "düşünebilir" mi gibi sorular basında ve radyo programlarında ciddi ciddi tartışıldı (1951'de BBC Radyo'daki konuyla ilgili bir programa Turing bizzat katılmıştır). Henüz adı

konmuş olmasa da, YZ fikrinin eli kulağındaydı.

Kamuoyu tartışmalarının etkisiyle, Turing yapay zekânın olabilirliği üstüne ciddi ciddi kafa yormaya başladı. Kamuoyu tartışmalarında ortaya atılan "Makineler asla *şunu yapamaz*" türü iddialardan (italiklerin yerine *düşünemez*, *akıl yürütemez*, *yaratıcı olamaz* vb.ni koyarak okuyun) son derece rahatsızdı. "Makinelerin düşünemeyeceğini" savunanların ağzını kapatmak istiyordu. Bu amaçla, bir test yapmayı önerdi; bugün bu testi onun adıyla anıyoruz. **Turing Testi** ilk defa betimlendiği 1950'den bu yana bir hayli etkili oldu, bugün hâlâ çok ciddi araştırmalara konu oluyor. Ama ne yazık ki, şimdi bahsedeceğimiz nedenlerle, makinelerin düşünemeyeceğini öne sürenlerin ağzını kapatmayı başaramadı.

Turing'in ilham kaynağı Taklit Oyunu'ydu. Bu Viktoryen salon oyununda, sadece karşınızdaki kişinin sorulan sorulara verdiği yanıtlardan yola çıkarak, aklındakinin bir kadın mı yoksa erkek mi olduğunu tahmin etmeye çalışıyordunuz. Turing YZ için de böyle bir test yapılabileceğini düşünüyordu. Bu test genel olarak şöyle tanımlanır:

> İnsan katılımcılar bir bilgisayar klavyesi ve monitörü aracılığıyla sorular sorarak karşı tarafla etkileşime geçer. Karşıdaki ya insandır ya da bilgisayar programı ama soruyu soran bunu önceden bilmez. Etkileşim sadece yazılı olarak ve soru-cevap şeklinde gerçekleşir: Katılımcı bir soru yazar, ardından da monitörde cevabını görür. Soruyu soranın görevi, karşısındakinin bir insan mı yoksa bilgisayar programı mı olduğunu belirlemektir.

Şimdi, diyelim ki soruları cevaplayan aslında bir bilgisayar programı, ama belli bir noktadan sonra soruyu soran kişi karşısındakinin bir program mı yoksa insan mı olduğunu tam olarak kestiremiyor. Bu durumda, Turing'e göre, programın insan düzeyinde zekâ gibi bir şeye sahip olduğunu kabul etmek gerekir. Turing burada çok akıllıca bir hamle yapmakta, bir programın "gerçekten" zekâya (veya bilince vb.) dayalı olup olamayacağına dair her türlü soru ve tartışmadan ustalıkla sıyrılmaktadır. Bir programın "gerçekten" düşünüp düşünmediğinin (veya bilinçli, özfarkındalığa sahip vb. olup olmadığının) bir bakıma çok da önemi yoktur bu bağlamda, çünkü ni-

hayetinde yaptığı şey *"gerçek şey"den ayırt edilemez*. Burada anahtar sözcük *ayırt edilemez*'dir.

Turing testi bilimde standart bir tekniğin derli toplu bir örneğidir. İki şeyin aynı mı yoksa farklı mı olduğunu anlamak istiyorsam, makul bir test yaparak bu ikisini birbirinden ayırt edebiliyor muyum ona bakarım. Eğer biri bu testten geçerken diğeri geçemiyorsa, gönül rahatlığıyla farklı olduklarını söyleyebilirim. Eğer bu tür bir testle ikisini birbirinden ayıramıyorsam, farklı olduklarını iddia edemem. Turing'in testi makine zekâsını insan zekâsından ayırt etmekle ilgilidir; bir insanın, insan zekâsının ürettiği davranış ile makine zekâsının ürünü olan davranışı birbirinden ayırıp ayıramadığına bakar.

Yalnız burada bir şeye dikkat etmemiz lazım. Yıllardır YZ'nin ayırt edici özelliklerini ortaya koymaya çalışan pek çok yaklaşımın sıkıntılı bir tarafı vardır: YZ'yi kullanılan *teknik* veya *metotlara* göre tanımlarlar. Örneğin favori YZ tekniğiniz (bugün YZ'nin moda tabirlerinden rasgele birkaçını cümle içinde kullanacak olursam) "Zamansal Açıdan Yinelenen Optimal Öğrenme" ise, YZ'yi Zamansal Açıdan Yinelenen Optimal Öğrenmeyi kullanarak Turing testini geçme görevi olarak tanımlamaya meyledebilir, başka her türlü yaklaşımı dışarıda bırakabilirsiniz. Dolayısıyla zekâya dayalı davranışa yönelik testin, bu davranışı elde etmek için kullanılan teknik veya metotlardan *bağımsız* olmasını isteriz. Turing testi bunu soruyu soran ile soru sorulanı net biçimde ayırarak yapar: Soru soranın elinde devam etmek için sadece girdiler ve çıktılar, yani gönderdiği sorular ve ardından aldığı cevaplar vardır. Soruların gönderildiği bilgisayarın diğer ucundaki şeyin iç yapısını incelememize izin verilmemesi, elimizde sadece girdi ve çıktılar olması bakımından, Turing testinde karşımızdaki tam bir *karakutudur*.

Turing'in testini anlatan makalesi "Bilgi İşlem Makineleri ve Zekâ" 1950'de prestijli bir uluslararası dergi olan *Mind*'da yayımlandı.[9] Bundan önce de YZ benzeri fikirlere değinen makaleler yayımlanmıştır, ancak ilk defa Turing bu konuya modern dijital bilgisayar açısından yaklaşmıştır. Bu bakımdan makalesi genelde ilk YZ yayını kabul edilir.

## Turing Testine Yüzeysel Yaklaşımlar

Turing testi gayet basit ve zariftir, kolayca anlaşılır. Ama ne yazık ki YZ'nin kutsal kâsesi haline gelmiştir ve bunun da günümüze kadar uzanan birtakım talihsiz sonuçları olmuştur.

Sorun, Turing testini geçmeye yönelik girişimlerin çoğunun, zekâya dayalı davranışla ilgili asıl meselelerle uğraşmak yerine, soruyu soran kişinin kafasını karıştırıp onu karşısındakinin bir insan olduğuna inandırmak için ucuz numaralara başvurmasıdır. Bunun en bilinen örneği, MIT'den Alman-Amerikalı bir bilgisayar bilimci olan Joseph Weizenbaum'un 1960'ların ortasında yazdığı ELIZA programıdır. Weizenbaum ELIZA'yı sırf rakipleri arasından sıyrılıp testi geçmeyi başaran bir program ortaya koyma niyetiyle yazmamıştır, aslında başlı başına ciddi ve etkili bir bilimsel deneydir bu, ama ne yazık ki genelde YZ'ye ve özelde Turing testine yüzeysel yaklaşımların alametifarikası haline gelmiştir.

ELIZA, danışanıyla konuşan bir psikiyatrist rolü yapar, sorduğu açık uçlu sorularla danışanının kendi üstüne düşünmesini sağlamaya çalışır. Öncelikle "aile", "arkadaşlar", "yalnız", "üzgün" vb. anahtar sözcükleri seçer, ardından bu sözcüklerle ilişkili senaryo şablonlarını devreye sokar ve bu çerçevede sorularına devam eder. Bunun numara olduğunu anlamak için cin gibi biri olmaya falan gerek yoktur. Örneğin sistemin güncel versiyonuyla gerçekleştirilen şu diyaloğa bir göz atalım (italikler ELIZA'nın cevapları):[10]

Çok yalnızım.
*Genelde mi böyle?*

Evet. Hemen her gün.
*Neden hemen her gün diyorsunuz?*

Ailem yüzünden sanırım.
*Ailenizden bahseder misiniz biraz?*

Benden nefret ediyorlar.
*Neden sizden nefret etiklerini düşünüyorsunuz?*

Bana kötü davranıyorlar.
*Çok yalnızım diyorsunuz?*

Hayatın anlamı ne?
*Hayatınızın bununla bir ilgisi vardır belki.*

Varoluşun amacı ne?
*Planlarınız nedeniyle mi varoluşun amacı ne diye soruyorsunuz?*

Başlangıç umut verici ama birkaç cümleye kalmadan etkileşimin ne kadar yüzeysel olduğu can sıkıcı bir biçimde açığa çıkıyor, ELIZA'nın bu diyaloğu hiç anlamadığı ortada (Weizenbaum'un da böyle bir iddiası olmamıştır zaten, deneyin amacı bu değildir).

Weizenbaum ELIZA'nın akademik camiada ve diğer çevrelerde böylesine büyük bir etki yaratacağını beklemiyordu muhtemelen. Programın kullanıcılarının bu basit deneyi ciddiye aldığını görünce şaşırmıştı. ELIZA çok geçmeden YZ folklorunun bir parçası haline geldi. Hatırlıyorum da ben doktora öğrencisiyken, 1980'lerde, sosyalleşme konusunda birtakım sıkıntılar yaşadığı için kendini yalnız hisseden lisansüstü öğrencilerinin geceler boyunca ELIZA'yla sohbet ettikleri, onunla kurdukları ilişkiyi gerçek hayatta kuramadıkları insan ilişkileri yerine koydukları konuşulurdu. Hatta ELIZA'nın bilmeden –inşallah sadece bir şehir efsanesidir bu– zaten depresyonda olan bir öğrenciyi intihara teşvik ettiği söylenirdi. Ayrıca Weizenbaum da sekreterinden programı denemesini istediğinde şaşırtıcı bir tepkiyle karşılaştığını; sekreterin diyalog devam ederken bir yerden sonra, bu etkileşimin aslında gerçek olmadığını gayet iyi bildiği halde, Weizenbaum'dan odadan çıkmasını rica ettiğini anlatır.

ELIZA'nın mirası her yıl düzenlenen Loebner Ödülü biçiminde hâlâ yaşıyor bugün. Turing bir gün birilerinin bu testi sahiden deneyeceğini düşünmüş müdür bilmiyoruz ama 1990'da Amerikalı milyoner mucit Hugh Loebner tam olarak bunu yapmaya karar vermiştir. Loebner Vakfı her sene Turing testine girmeye aday bilgisayar programları için çağrı yapar, amaç programın jüri heyetini aslında insan olduğuna ikna etmesidir, kazanan için yüz bin dolarlık bir ödül vardır.

Loebner'ın yarışması ilk günden beri pek çok tartışmaya konu olmuş, sözgelimi tanınmış YZ bilimcilerinden Marvin Minsky tarafından "reklam kokan hareketler" olarak nitelendirilmiştir. Yarışma

her sene en azından bir tane fazlasıyla yanıltıcı manşete sebebiyet vermektedir. Sorun, Loebner yarışmasına katılan programların çoğunlukla ELIZA temasının çeşitlemelerinden ibaret olmasıdır. Soru soranları insan düzeyinde bir kavrayış, sağduyu ve idrak sergileyen anlamlı bir sohbete dahil ederek kazanmak yerine ucuz numaralarla yanlış yönlendirmeye ve yanıltmaya çalışmaktadırlar. Örneğin 2009 yılının kazananı olan, tanınmış YZ yorumcusu ve bilgisayar satrancı uzmanı David Levy'nin yazdığı "Do-Much-More" programının gerçekleştirdiği bir sohbetten alınan şu fragmana bir göz atalım (italikler Do-Much-More'un cevapları):

Selam, n'aber?
*İyilik, senden?*

Benden de iyilik. Sen kimsin?
*Mühendislik okuyorum. Kim olduğum konusunda bir fikir veriyordur herhalde bu.*

Valla çok da bir fikir vermiyor açıkçası.
*Deneyler hakkında daha fazlasını öğrenmeyi ne kadar çok istediğimi anlatamam.*

Nerelisin?
*Bilmem, ama aynı anda üç yerde birden olamayacağım açık.*

Hangi şehirde yaşıyorsun?
*Hangi şehir mi? Şaşırtmacalı bir soru mu bu? Çünkü bütün şehirler birbirine benziyor.*

Yo, değil. Nerede yaşıyorsun?
*Nehrin oradaki eski mandırada galiba.*

İlginç, çiftçi misin o zaman?
*Bir fizyoloğa sorsan, benim için çiftçi demez. Bolca –koca bir bira fıçısını dolduracak kadar– sudan, artı biraz demirden, fosfordan ve gayet alışıldık başka bileşenlerden oluştuğumu söyler.*

Böyle basit bir soru için biraz garip bir cevap olmadı mı bu, ne dersin?
*Kesinlikle evet derim. İzle de gör!*

ELIZA'dan daha sofistike belki ama sohbete katılanın bir insan olmadığı hâlâ çok açık.

ELIZA, YZ araştırmacılarının adını duyunca bile bunaldığı bir fenomenin, yani internet sohbet robotunun doğrudan atasıdır. Çoğunlukla Twitter gibi sosyal medya platformları aracılığıyla, kullanıcıları sohbete dahil etmeye çalışan internet tabanlı programlardır bunlar. Ciddi araştırmalar anlamlı bir sohbet edebilen programlar geliştirme problemini konu edinirken, internet sohbet robotlarının ezici çoğunluğu tıpkı ELIZA gibi anahtar sözcük tabanlı senaryo şablonları kullanmanın pek ötesine geçmez, dolayısıyla ürettikleri sohbetler de bir o kadar yüzeysel ve sıradandır. Bu tür sohbet robotları YZ değildir.

## Yapay Zekâ Çeşitleri

Turing testi, bu tür saçmalıkların önünü açtıysa da YZ hikâyesinin önemli bir parçasıdır, çünkü böylece ilk defa, yeni yeni ortaya çıkan bu alana meraklı araştırmacıların ulaşılacak bir hedefi olmuştur. Amacın ne sorusuna verecek dosdoğru ve net bir cevapları vardır artık: *Turing testini gerçek anlamda geçebilecek bir makine inşa etmek.* Bugün ciddi bir YZ araştırmacısının kolay kolay böyle bir cevap vereceğini sanmıyorum, ama Turing testinin tarihsel açıdan önemli bir rol oynadığını ve hâlâ bize söyleyecek önemli şeyleri olduğunu düşünüyorum.

Çekiciliği büyük ölçüde sadeliğinden geliyor elbette, ama ne kadar açık ve net olursa olsun testin YZ ile ilgili birçok sorunlu noktayı gündeme getirdiği de ortada.

Diyelim ki Turing testinde soru soran kişi sizsiniz. Aldığınız cevaplara bakınca karşınızdakinin bir sohbet robotu olmadığına kanaat getiriyorsunuz: Karşınızdaki her neyse sorularınızı tıpkı bir insan gibi anladığını gösteriyor ve bir insanın vereceği türden cevaplar veriyor. Ama bir de bakıyorsunuz ki karşınızdaki aslında bir bilgisayar programı. Turing testi bakımından konu kapanmıştır. Program insan davranışından ayırt edilemeyen bir şey yapmaktadır: tartışmanın sonu, nokta. Ama mantıksal açıdan hâlâ en az iki ayrı olasılık var:

1. Program, diyaloğu tıpkı bir insan gibi, *gerçekten anlar*.

2. Programın aslında bir şey anladığı falan yoktur ama insanınki gibi bir idraki *simüle edebiliyordur*.

Bu iki iddia arasında dünyalar kadar fark var. Birincisi, program *gerçekten anlar* iddiası ikincisinden çok daha güçlüdür. Zira ikincisi için sadece anlıyormuş *gibi görünen* bilgisayarlar yapabiliyor olmamız yeterlidir.

Sanırım YZ araştırmacılarının –ve muhtemelen bu kitabın okurlarının– çoğu 2. tip programların, en azından teoride, uygulanabilir olduğunu rahatlıkla kabul edecektir. Ancak 1. tip programların yapılabilir olduğunu kabul etmeleri için bunların çok daha ikna edici olması gerekir. Ayrıca bir programın 1. tip olduğu iddiasını nasıl ispatlayacağımız da belli değildir. Turing testi böyle bir iddiada bulunmaz. (Böyle bir ayrım Turing'i uyuz ederdi herhalde: Testi icat ederken başlıca amaçlarından birinin tam da böyle ayrımlar hakkındaki tartışmalara son vermek olduğunu vurgulardı.) Öyleyse 1. tip bir program inşa etmek, 2. tip bir program inşa etmekten misliyle iddialı ve tartışmalı bir amaçtır.

*Gerçekten de insan gibi anlayan (bilinci olan vb.)* programlar inşa etme amacına **güçlü YZ** denir; yine anlıyormuş gibi görünmekle beraber *söz konusu niteliklere gerçekten de sahip olma* iddiası taşımayan programlar inşa etme amacına ise **zayıf YZ** denir.

## Turing Testinin Ötesinde

Turing'in ayırt edilemezlik testinin pek çok çeşitlemesi vardır. Mesela testin çok daha güçlü bir versiyonunda, gündelik hayatta kendini insan gibi göstermeye çalışan robotlar olduğunu hayal edebiliriz. Bu bağlamda "ayırt edilemezlik"in son derece talepkâr bir anlamı olduğu düşünülür: insanlardan ayırt edilemeyen makineler. (Robotları ameliyat masasına yatırıp incelemenize izin verilmediğini varsayarsak, test hâlâ bir karakutudur.) Yakın bir gelecekte, bu tür şeylerin kurmacanın ötesine geçemeyeceği söylenebilir. Robotları insanlardan ayırmanın zor olduğu bir dünyayı anlatan güzel bir film vardır

mesela: Ridley Scott'ın 1982 tarihli klasiği *Blade Runner / Ölüm Takibi*'nde genç Harrison Ford, görünüş itibarıyla güzel bir genç kadının aslında robot olup olmadığını tespit etmek için gizli testler yapar. 2014 yapımı bir film olan *Ex Machina* da benzer temaları ele alır.

Ufukta *Blade Runner* senaryolarının gerçekleşmesi gibi bir ihtimal olmasa da, araştırmacılar hakiki zekâyı gerçek anlamda sınayacak, sohbet robotlarında başvurulan türden numaraları yemeyecek Turing testi çeşitlemeleri olup olmadığını sormaya başladılar. Bununla ilgili basit fikirlerden biri *kavrayışın* test edilmesi, söz konusu fikrin tezahürlerinden biri ise **Winograd şablonlarının** kullanılmasıdır. Şablonları oluşturan kısa soruları şu örneklerle somutlaştırmak iyi olabilir:[11]

> Önerme 1a: Belediye meclisi üyeleri şiddetten korktukları için göstericilere izin vermediler.
>
> Önerme 1b: Belediye meclisi üyeleri şiddetten yana oldukları için göstericilere izin vermediler.
>
> Soru: Şiddetten korkan / yana olan kim?

Gördüğünüz gibi iki önerme arasında çok küçük bir fark var (altı çizili) ama bu ufacık fark anlamı tamamen değiştiriyor. Testin amacı iki ayrı durumda "onlar"ın kime işaret ettiğini tespit etmek. Önerme 1a'da "şiddetten korkanlar"ın meclis üyeleri olduğu ortada, Önerme 1b'de ise "şiddetten yana olanlar" derken göstericilerin kastedildiği (meclis üyelerinin göstericilerin şiddete başvurmasından kaygılandığı) anlaşılıyor.

Başka bir örneğe bakalım:

> Önerme 2a: Aldığım hediye kahverengi valize sığmıyor çünkü çok küçük.
>
> Önerme 2b: Aldığım hediye kahverengi valize sığmıyor çünkü çok büyük.
>
> Soru: Çok küçük/büyük olan ne?

Önerme 2a'da valizin küçüklüğünün, Önerme 2b'de ise hediyenin büyüklüğünün kastedildiği aşikârdır.

Okuma yazması olan yetişkinlerin çoğunun rahatlıkla altından kalktığı Winograd şablonları, sohbet robotlarının ve Loebner yarış-

masına katılan programların ucuz numaralarını yakalar. Dolayısıyla şu sonuca varıyoruz: Doğru cevabı verebilmek için metni hakikaten *anlamanız*, söz konusu senaryo hakkında *bilgi* sahibi olmanız gerekir. Mesela Önerme 1a ile 1b arasındaki farkı anlamanız için gösteriler (şiddete varabilirler) ve belediye meclisinin üyeleri (gösterilerin yapılıp yapılmamasına karar verme gücüne sahiptirler ve şiddete yol açabilecek durumların ortaya çıkmasından kaçınırlar) hakkında bir şeyler bilmeniz gerekir.

YZ'nin bünyesinde barındırdığı benzer bir başka zorluk da insanın dünyasını ve bu dünyadaki ilişkilerimizi yöneten yazılı olmayan kuralları anlamayı gerektirmesidir. Psikolog ve dilbilimci Steven Pinker'dan aldığımız şu kısa diyaloğa bakalım:

> Bob: "Ayrılalım."
> Alice: "Kim o kadın?"

Bu diyaloğu izah edebilir misiniz? Tabii ki edebilirsiniz. Pembe dizilerin alametifarikalarından biridir bu: Alice ve Bob'un bir ilişkisi vardır, Bob ayrılmak istediğini söyleyince Alice kesin başka bir kadın için terk edildiğini düşünür ve hemen bu kadının kim olduğunu öğrenmek ister. Alice'in çok kızgın olduğunu da tahmin edebiliriz.

Bir bilgisayarın böyle diyalogları anlar hale gelmesi için nasıl programlanması gerekir peki? Bir hikâyeyi anlamak, hatta hikâye yazmak için muhakkak bu tür bir yeti gereklidir. Bir bilgisayar programının *EastEnders* gibi bir pembe diziyi takip edebilmesi için, böyle bir sağduyu, insanın kanaat, arzu ve ilişkilerinin işleyiş tarzlarını anlamayı sağlayan gündelik bir kapasite olmazsa olmazdır. Herkesin sahip olduğu bu kapasite ayrıca hem güçlü YZ'nin hem de zayıf YZ'nin başlıca gereklerinden biri gibi görünmektedir. Bunun nasıl becerileceğine dair fikirlerimiz biraz havada kalıyor, herhangi birinin bir sonuca ulaştığını görmedik henüz. Makinelerin böyle senaryoları anlayıp ilgili soruları cevaplamayı becerir hale gelmesine daha çok var.

## Genel YZ

Büyük YZ hayali sezgisel olarak aşikârsa da, ne anlama geldiğini veya onu bulduğumuzu nereden bileceğimizi tanımlamak beklenmedik derecede zordur. Bu nedenle güçlü YZ'nin, YZ hikâyesinin önemli ve heyecan verici bir parçası olmakla birlikte, günümüz YZ araştırmalarıyla pek bir alakası yoktur. Bugün bir YZ konferansına gitseniz, bu konuda tek kelime bile duymazsınız herhalde; konferans sonrası gidilen meyhanede gecenin bir yarısı muhabbet buraya gelebilir ama.

Biraz daha küçük bir hedef ise genel amaçlı ve insan düzeyinde zekâya dayalı makineler inşa etmektir. Şu sıralar buna genellikle **Yapay Genel Zekâ** (YGZ) veya **Genel YZ** deniyor. YGZ kabaca bir insanın çok çeşitli düşünsel yetilerine –ortalama bir insan kadar veya daha üst düzeyde doğal dilde sohbet etme (karş. Turing testi), problem çözme, akıl yürütme, içinde bulunulan ortamı algılama vb.– sahip bir bilgisayara ulaşmak anlamına geliyor. YGZ'ye muktedir bir sistem Şekil 1'deki görevlerin hepsini ve daha fazlasını yerine getirebilir. YGZ literatürü bilinç veya özfarkındalık gibi meselelerle genelde pek ilgilenmez, dolayısıyla YGZ bir bakıma zayıf YZ'nin daha zayıf bir versiyonu olarak düşünülebilir.[12]

Yalnız bu daha küçük hedef bile günümüz YZ'sinin merkezinde yer almıyor. YZ araştırmacıları daha ziyade beyin gerektiren görevleri gerçekleştirebilen bilgisayar programları inşa etmeye odaklanmış durumdalar; Şekil 1'de listelenen problemleri çözerek adım adım ilerliyorlar. Bilgisayarların gayet spesifik görevleri yerine getirmesini sağlamaya yönelik bu yaklaşıma zaman zaman **dar YZ** de dendiğini görüyoruz ama YZ camiasından kimse bu terimi kullanmaz. Mesela başlıca YZ konferanslarından birinde bu lafı duyacak olursanız, bilin ki söyleyen alan dışından biridir. Yaptığımız şeye "dar YZ" demiyoruz çünkü bununla kastedilen "YZ"nin *kendisinden* başka bir şey değil – bunu öğrenince robot uşak sevdalıları hayal kırıklığına uğrayacak, robotların ayaklanıp dünyayı ele geçirmesiyle ilgili kâbuslar görenlerin de yüreğine su serpilecektir herhalde.

Artık YZ'nin ne olduğunu bildiğimize, YZ'yi gerçekleştirmenin neden zor olduğuna dair genel bir fikir edindiğimize göre şimdi de şunu sorabiliriz: Öyleyse YZ araştırmacıları bu konuda tam olarak nasıl bir yol izliyor?

## Beyin mi Zihin mi?

Bir bilgisayarda insan düzeyinde zekâya dayalı davranış üretmek için nasıl bir yol izleyebiliriz? YZ tarihsel olarak bu probleme yönelik başlıca iki yaklaşımdan birini benimsemiştir. Birinci yaklaşım, kabaca, *zihni* modellemeye çalışmaktır: hepimizin hayatımızı sürdürürken başvurduğumuz bilinçli muhakeme, problem çözme vb. süreçleri yani. Bu yaklaşıma **simgesel YZ** denir çünkü sistemin hakkında akıl yürüttüğü şeyleri temsil eden simgeleri kullanır. Diyelim ki bir robotun kontrol sistemi dahilinde "oda451" simgesi yatak odanız için, "temizleOda" simgesi de bir odayı temizleme faaliyeti için kullanılıyor. Robot ne yapılacağını çözdükçe, bu simgeleri açıkça kullanır; sözgelimi "temizleOda(oda451)" eylemini gerçekleştirmeye karar vermesi, yatak odanızı temizlemeye karar verdiği anlamına gelir. Kullandığı simgeler robotun içinde bulunduğu ortamda *bir anlama gelmektedir*.

Simgesel YZ 1950'lerin ortasından 1980'lerin sonuna kadar otuz küsur sene boyunca YZ sistemleri inşa etmeye yönelik en popüler yaklaşım olmuştur. Pek çok artısı olan simgesel YZ'nin belki de en önemli üstünlüğü *şeffaf* olmasıdır: Robot "temizleOda(oda451)" sonucuna vardı mı, ne yapmaya karar verdiğini hemen anlayabiliriz. Simgesel YZ'nin böylesine popüler olmasının bir nedeni de bizim bilinçli düşünme süreçlerimizi yansıtıyor gibi görünmesi bence. Simgeler –sözcükler– çerçevesinde "düşünürüz" ve ne yapılacağına karar verirken kendimizle çeşitli eylemlerin artıları ve eksileri hakkında zihinsel bir diyalog yürütebiliriz. Simgesel YZ'nin bunların hepsini yakalamak* gibi bir hedefi vardı. Ve 2. Bölüm'de göreceği-

* İng. *capture*: Elde edilen enformasyonun bilgisayarın kullanabileceği bir biçime sokulması.

miz üzere, 1980'lerin başında doruk noktasına ulaşmıştı.

Zihni modellemenin alternatifi ise *beyni* modellemektir. Böyle uç bir olasılık, bir bilgisayar dahilinde insan beyninin (ve hatta sinir sisteminin) tamamını simüle etmeye çalışmak olacaktır. Nihayetinde insan düzeyinde zekâya dayalı davranış üretebildiğini kesin olarak söyleyebileceğimiz tek şey insan beynidir. Bu yaklaşımın sorunlu tarafı beynin hayal bile edemeyeceğimiz kadar karmaşık bir organ olmasıdır: İnsan beyninin birbiriyle bağlantılı yaklaşık 100 milyar bileşeni vardır ve yapılarıyla işleyişlerini henüz yeterince anlayamamış olduğumuzdan bunları kopyalayamayız. Yakınlarda bunun mümkün olabileceğini düşünmek pek gerçekçi gelmiyor bana, belki de asla mümkün olmayacak (yine de ne yazık ki bazılarını denemeye devam etmekten alıkoyamadı bu).[13]

Gelgelelim bunun yerine, beyinde bulunan kimi yapılardan ilham alıp bunları zekâya dayalı sistemlerin bileşenleri olarak modelleyebiliriz. Bu araştırma alanına, beynin mikro yapısında bulunan hücresel enformasyon işleme birimleri olan nöronlardan dolayı, **nöral ağlar** deniyor. YZ'nin ortaya çıkışını önceleyen nöral ağların incelenmesi anaakım YZ'nin yanı sıra gelişmiştir. Bu alanın içinde bulunduğumuz yüzyılda etkileyici bir ilerleme kaydetmiş olması, günümüzde YZ'nin müthiş bir patlama yapmasına vesile olmuştur.

Simgesel YZ ve nöral ağlar, metodolojileri birbirinden çok farklı olan iki ayrı yaklaşımdır. Son altmış yıl içinde ikisi de ara ara gözden düşüp yeniden rağbet görmüştür, hatta ileride göreceğimiz üzere iki ekol arasında kimi tatsızlıkların yaşandığı bile olmuştur. Ne var ki yeni bir bilimsel disiplin olarak ortaya çıktığı 1950'lerin ortasından itibaren, YZ giderek daha hâkim hale gelmiştir.

İKİNCİ KISIM

# Bu Noktaya Nasıl Geldik?

# 2

# YZ'nin Altın Çağı

TURING'İN KENDİ ADIYLA ANILAN testi takdim ettiği makalesi "Bilgi İşlem Makineleri ve Zekâ"yı bugün artık YZ disiplinine yapılan ilk kayda değer bilimsel katkı olarak kabul ediyoruz. Fakat bu katkı biraz münferit kalmıştı, çünkü o tarihte YZ disiplini falan yoktu, hatta daha adı bile konmamıştı. Aynı şekilde bir araştırmacılar topluluğu da yoktu. Sadece Turing testi gibi spekülasyona dayalı kavramsal katkılar vardı, YZ sistemleri henüz ortaya çıkmamıştı. On yıl kadar sonra, 1950'lerin sonunda işe durum tamamen değişecekti ama: YZ ayrı bir disiplin olarak karşımıza çıkıyordu, adı konmuştu, araştırmacılar zekâya dayalı davranışın ilkel bileşenlerini sergileyen ilk sistem denemelerini gururla görücüye çıkarabiliyordu artık.

Sonraki yirmi yılda tam bir YZ furyası yaşandı. Bir iyimserlik, büyüme ve bariz ilerleme dalgasıyla, 1956-74'te **YZ'nin Altın Çağı** yaşandı. Hüsranın h'si yoktu ortada, her şey mümkünmüş gibi görünüyordu. Bu dönemde inşa edilen YZ sistemleri YZ kanonunda efsanedir. Hemen hepsinin bilgisayar meraklılarının yüzüne bir gülümseme konduran, SHRDLU, STRIPS ve SHAKEY gibi acayip isimleri vardı; bu isimlerin kısa olması, hepsinin büyük harfle yazılması, o zamanlar bir bilgisayar dosyasını isimlendirmenin böyle kısıtları olmasından kaynaklanıyordu (YZ sistemlerini bu şekilde isimlendirmek uzun zaman önce bir zorunluluk olmaktan çıktıysa da bu gelenek hâlâ yaşatılıyor). Bu sistemleri inşa ederken kullanılan bilgisayarlar günümüz standartlarına göre hayal edilemeyecek kadar kısıtlı ve sinir bozucu derecede yavaştı, üstelik bunları kullanması da

çok zordu. Bugün elimizin altında olan yazılım geliştirme araçları o dönemde henüz ortalarda yoktu, hem olsalar bile dönemin bilgisayarları bu araçları kaldırmazdı zaten. Bilgisayar programcılığının *hacker* kültürü büyük ölçüde bu dönemde ortaya çıkmıştır. YZ araştırmacıları gece çalışıyordu, çünkü bilgisayarlar mesai saatlerinde daha önemli işler için kullanılıyor, ancak geceleri boş oluyordu. Ayrıca karmaşık programlarını sırf çalıştırmak için bile birbirinden yaratıcı programlama numaraları icat emek durumunda kalıyorlardı. Akabinde standart teknik haline gelen bu numaraların çoğu, şimdi pek hatırlanmasa da, esasen 1960'ların ve 1970'lerin YZ laboratuvarlarında doğmuştur.[1]

Ama 1970'lerin ortasına doğru YZ'de ilerleme durdu, ilk baştaki basit deneylerin çok da ötesine geçilemedi. Büyük vaatlerde bulunan YZ'nin aslında pek bir yere gitmediğine inanmaya başlayan araştırma fonu sağlayıcılar ve bilim camiası az kalsın bu alanın fişini çekiyordu.

Bu bölümde YZ'nin ilk yirmi yılını ele alacağız. Söz konusu dönemde inşa edilen başlıca sistemlerin bazılarına bakıp, alanda geliştirilen en önemli tekniklerden biri, bugün bile hâlâ pek çok YZ sisteminin merkezi bileşenlerinden olan "arama" tekniği üstünde duracağız. Ayrıca soyut bir matematiksel teorinin, 1960'ların sonuyla 1970'lerin başında geliştirilen bilgi işlemsel karmaşıklığın, YZ alanındaki pek çok problemin neden temelde zor olduğunu nasıl açıklamaya başladığını okuyacağız. YZ uzun zaman bilgi işlemsel karmaşıklığın gölgesinde kalmıştır.

Âdet olduğu üzere Altın Çağ'ın başlangıç noktasından, Amerikalı genç bir akademisyen olan John McCarthy'nin alana adını verdiği o 1956 yazından başlayalım.

## YZ'nin İlk Yazı

McCarthy, çağımızda ABD'nin teknolojinin en büyük öncülerinden biri haline gelmesinin önünü açan o akademisyenler kuşağındandı. 1950'lerden 1960'lara adeta kayıtsız bir dehayla icat ettiği bir dizi bilgi işlem kavramı bugün bize gayet doğal, ezelden beridir hep var-

mış gibi geldiğinden, vaktiyle bunları bilfiil icat etmek gerektiğini tahayyül etmekte zorlanıyoruz. McCarthy'nin en bilinen katkılarından biri, yıllar yılı YZ araştırmacılarının tercih ettiği programlama dili olan LISP'tir. En ideal koşullarda bile bilgisayar programlarını okumak zordur, ama mesleğimin düpedüz akıl sır ermeyen standartlarına göre dahi LISP'in bir acayip olduğu söylenir, çünkü LISP'te (bütün (programlar (böyle (görünür)))). Programcıların bununla ilgili, nesilden nesle aktarılan bir geyiği vardır: LISP'in açılımı ne? Lots of Irrelevant Silly Parentheses, yani "gerekli gereksiz bir sürü parantez".[2]

McCarthy LISP'i 1950'lerin ortasında icat etti. İnsan gerçekten hayret ediyor çünkü neredeyse 70 yıl sonra bugün hâlâ düzenli olarak öğretiliyor ve dünyanın dört bir yanında kullanılıyor. (Ben her gün kullanıyorum mesela.) Şöyle düşünün: McCarthy LISP'i icat ettiğinde ABD Başkanı Dwight D. Eisenhower'dı, Nikita Hruşçov Sovyetler Birliği Komünist Partisi'nin Birinci Sekreteri'ydi ve Çin'de Mao Zedung ilk beş yıllık kalkınma planının hayata geçirilmesine başkanlık ediyordu. Yeryüzündeki bütün bilgisayarların sayısı bir elin parmaklarını geçmiyordu. İşte McCarthy'nin böyle bir dünyada icat ettiği programlama dili bugün hâlâ düzenli olarak kullanılıyor.

Göçmen bir ailenin çocuğu olarak Boston'da dünyaya gelen McCarthy'nin matematiğe olağanüstü bir yatkınlığı olduğu daha küçük yaşında anlaşılmıştı. California Teknoloji Enstitüsü (Caltech) Matematik bölümünden mezun olduktan sonra, yani henüz yirmili yaşlarında, New Hampshire'daki Dartmouth Üniversitesi'nde doçent olarak görev yapmaya başladı. Uzun zamandır bilgi işleme büyük ilgi duyan McCarthy Darmouth'a geldikten sonra, 1955'te Rockefeller Enstitüsü'ne bir proje sundu; buradan kaynak sağlayabilirse, Darmouth'ta bir yaz okulu düzenlemeyi ümit ediyordu. Eğer akademisyen değilseniz, yetişkinler için "yaz okulu" fikrini biraz yadırgayabilirsiniz, ama aslına bakarsınız günümüzde bile sevilip sayılan, iyice oturmuş bir akademik gelenektir bu. Dünyanın dört bir yanından ilgi alanları ortak araştırmacıları bir araya getirerek tanışmalarını ve belli bir süre birlikte çalışmalarını sağlama esasına da-

yanır. Bilhassa yaz aylarında düzenlenmesinin sebebi ise, tahmin edeceğiniz üzere, derslerin bitmiş olması, dolayısıyla ders yükünden azade akademisyenlerin yazın araştırma için daha fazla zamanının olmasıdır. Genelde albenili bir lokasyonda düzenlenir, ayrıca dolu dolu bir sosyal etkinlik programı da böyle yaz okullarının olmazsa olmazlarındandır.

Unutulmaz bir yaz okulu düzenlemek istiyorsanız eğer, starlarla dolu bir katılımcılar listesi de şarttır. Bugünden geçmişe bakınca, Dartmouth yaz okulunun YZ alanını sonraki onyıllarda tanımlayacak olan belli başlı kişilerin çoğunu bir araya getirmiş olduğunu görüyoruz. Yalnız McCarthy'nin davetliler listesindeki bir isim özellikle etkileyici: John Forbes Nash Jr. Princeton menşeli matematikçi altı yıl önce doktorasını verirken, "işbirliksiz oyun" kavramını (sadece 28 sayfa uzunluğundaki tezinde) ortaya atmıştı. Nash'in fikirleri müteakip onyıllarda iktisat teorisinin temel taşları haline geldi, hatta matematikçiye 1994'te bir Nobel Ödülü kazandırdı. Ne var ki Nash çalışmalarının böylesine takdir görmesinin tadını çıkaramayacaktı maalesef. Doktoradan birkaç yıl sonra yavaş yavaş Nash'i tüketmeye başlayan paranoyalar ve kuruntular nihayetinde ünlü matematikçinin akademik hayattan yıllar yılı uzak kalmasına yol açtı. Ne mutlu ki zamanla yeterince iyileşip 1994'te Nobel Ödülü'nü kendisi alabilmiştir. Nash'in hayatı pek çok ödül kazanan bir kitaba ve bu kitaptan yola çıkan bir filme, *A Beautiful Mind / Akıl Oyunları*'na konu olmuştur.[3]

Dartmouth'un katılımcılar listesi bir nedenle daha dikkat çekicidir. Zira yaz okulu akademisyenlerin yanı sıra sanayiden, hükümetten ve ordudan isimlere de evsahipliği yapmıştır (hatta bu listede, 1960'larda kan dondurucu bir biçimde bir nükleer savaşın nasıl "kazanılacağını" tartışmasıyla bednam California merkezli düşünce kuruluşu RAND Corporation'ın temsilcileri de göze çarpmaktadır). Ondan on yıl kadar önce de Manhattan Projesi kapsamında ABD'de akademi, sanayi, hükümet ve ordunun çeşitli kapasitelerini birleştirerek ilk atom bombasını geliştirme, ABD'nin bilimsel ve teknolojik gücünü apaçık gösterme denemesinde bulunulmuştur. Bu akademi, sanayi ve hükümet/ordu kombinasyonu, ABD'de II. Dünya Savaşı'nı

izleyen onyıllarda gelişen bilgisayar teknolojisinin ayırt edici bir özelliğidir, keza sonraki altmış yılda ABD'nin YZ alanında dünya lideri haline gelmesinde önemli bir rol oynamıştır.

1955'te finansal kaynak için Rockefeller Enstitüsü'ne başvururken, düzenleyeceği etkinliğe bir isim vermesi gereken McCarthy buna "yapay zekâ" demiştir. Zamanla usanç verici bir YZ geleneği haline gelecek olan bir yaklaşımla, McCarthy'nin bu etkinlik için gerçekçilikten nasibini almamış yüksek beklentiler içinde olduğunu görüyoruz: "Eğer dikkatle seçilmiş bir grup biliminsanı yaz boyu hep beraber bu konu üstünde çalışırsa ... kayda değer bir ilerleme sağlanabileceği görüşündeyiz."[4]

Yaz okulunun sonunda katılımcılar kayda değer bir ilerleme sağlayamadı belki ama McCarthy'nin bulduğu isim tuttu ve YZ adıyla yeni bir akademik disiplin oluştu.

Sonradan, McCarthy'nin kurucusu olduğu alana verdiği isim pek çoklarınca talihsiz bir seçim olarak nitelendirilmiştir. Bir kere, *yapay* aynı zamanda *çakma* veya *sahte* olarak anlaşılabilir ve kimse *sahte bir zekâ* istemez. Dahası *zekâ* sözcüğü asıl meselenin *akıl* olduğunu düşündürür. Halbuki YZ camiasının 1956'dan beri canla başla üstünde çalıştığı görevlerin çoğu, insanlar bu görevleri yerine getirirken, belli bir *zekâ* gerektiriyormuş gibi değildir. Bilakis, önceki bölümde gördüğümüz üzere, YZ'nin altmış yılı aşkın bir süredir uğraştığı en önemli, en zor problemlerin büyük bölümü esasen *düşünsel* problemlermiş gibi bile görünmez. Bu olgu alana yabancı olanları afallatır hep, kafalarını karıştırır.

Öyle veya böyle, McCarthy'nin alan için seçtiği isim yapay zekâydı ve bu isim günümüze kadar geldi. McCarthy'nin yaz okuluna katılanların ve onları izleyen akademik nesillerin araştırmaları günümüze kadar kesintisiz devam etti. O yaz okulunda bugün bildiğimiz modern biçimini alan YZ, bu başlangıç döneminde gerçekten büyük umutlar vadediyor gibi görünüyordu.

Dartmouth yaz okulunu heyecan uyandıran bir büyüme dönemi izledi. En azından bir süreliğine, hızlı bir ilerleme kaydediliyor gibiydi. Müteakip onyıllarda, bu yaz okuluna katılan dört isim alana damga vurdu: McCarthy Silikon Vadisi'nin göbeğinde Stanford'ın

YZ laboratuvarını kurdu, Marvin Minsky Cambridge, Massachusett'te MIT'nin YZ laboratuvarını kurdu, Alan Newell ve doktora danışmanı Herb Simon Carnegie Mellon Üniversitesi'ne (CMU) gitti. Bu dört YZ'ci ve öğrencileriyle birlikte inşa ettikleri YZ sistemleri, bizim kuşaktan YZ araştırmacıları için birer efsanedir.

Bununla beraber Altın Çağ'ın çok naif bir tarafı da vardı. Araştırmacıların alanın muhtemel ilerleme hızı konusunda biraz yüksekten uçması, bugün bile YZ'nin peşini bırakmayan beklentilerin ortaya çıkmasına yol açtı. 1970'lerin ortasına doğru o güzel günler sona erecek, talihsiz bir gerileme başlayacak; müteakip onyıllarda YZ tekrar tekrar bir yükselişe bir düşüşe geçtiği bir döngüye hapsolacaktı. Tarih bu dönemi pekâlâ eleştirip yargılayabilir, benimse, bu dönemin figürlerini ve yaptıkları işleri muhabbetle anmaktan başka türlüsü gelmiyor elimden.

### Böl ve Yönet

Gördüğünüz gibi, açık ve net biçimde tanımlanmış bir hedef olmadığı için, Genel YZ'ye doğrudan yaklaşmak öyle kolay değildir. Bu yüzden YZ'nin Altın Çağı'nda benimsenen strateji *böl ve yönet* stratejisi oldu. En baştan itibaren bütün bir genel zekâ sistemi inşa etmeye çalışmak yerine, genel amaçlı YZ için gerekli görünen çeşitli *yetileri* tek tek tespit edip bunları sergileyebilen sistemler inşa etme yaklaşımı benimsendi. Yaklaşımın altında yatan varsayım, söz konusu yetilerin tek tek her birini sergileyen sistemler inşa edebilirsek, ileride bunların birleştirilip bir bütün haline getirilebileceğiydi. Bu temel varsayımın zamanla iyice kabul görmesiyle YZ araştırmalarının standart metodolojisi de şekillenmiş oldu. Bir an önce zekâya dayalı davranışı meydana getiren ayrı ayrı yetileri sergileyen makineler inşa etmek istiyorlardı artık.

Araştırmacıların üstünde yoğunlaştığı belli başlı yetiler nelerdi peki? Birincisi, ve ileride anlaşılacağı kadarıyla en zorlusu, aslında kanıksadığımız bir şeydi: **algılama**. Belli bir ortamda zekâya dayalı davranışlar sergileyecek bir makinenin bu ortam hakkında enformasyon alabilmesi gerekir. Yaşadığımız dünyayı görme, işitme, do-

kunma, koku ve tat almadan oluşan beş duyumuz gibi çeşitli mekanizmalar yoluyla algılarız. Bu nedenle araştırma dallarından biri, böyle mekanizmaların muadillerini sağlayan **sensörler** inşa etmekten oluşur. Günümüzde robotların onlara içinde bulundukları ortam hakkında enformasyon sağlayan envai çeşit yapay sensörü vardır: radarlar, kızılötesi menzilölçerler, ultrasonik menzilölçerler, lazer radarlar vb. Ancak böyle sensörler *inşa etmek* –ki hiç de azımsanacak bir iş değildir bu– problemin sadece bir parçasıdır. Optiği ne kadar iyi olursa olsun, görüntü sensörü kaç piksel olursa olsun, dijital bir fotoğraf makinesinin yaptığı nihayetinde gördüğünü kılavuz çizgilerle parçalara ayırıp her bir hücreye renk ve parlaklığı belirten bir sayı tayin etmektir. Böylece en iyi dijital kameralarla donatılmış bir robotun eline geçen nihayetinde upuzun bir sayılar listesi olacaktır. Dolayısıyla algılamanın ikinci zorluğu bu ham sayıları yorumlamanın, yani gördüğü şeyin ne olduğunu anlamanın zorluğundan gelmektedir. Ve bunun bilfiil sensör inşa etmekten kat kat zor bir problem olduğu ortaya çıkmıştır.

Genel zekâ sistemleri için gereken başlıca yetilerden bir diğeri ise deneyimden öğrenmektir. Bu yüzden YZ araştırmalarının bir dalı makine öğrenmesine yoğunlaşmıştır. Tıpkı "yapay zekâ" gibi "makine öğrenmesi"nin de talihsiz bir isimlendirme olduğu söylenebilir. Çünkü makinenin bir şekilde kendi zekâsını geliştirmesi, sıfırdan başlayıp giderek daha akıllı hale gelmesi gibi anlaşılmaktadır. Halbuki makinenin öğrenmesi insanınkinden farklıdır; burada söz konusu olan, verilerden öğrenmek ve veriler hakkında tahminlerde bulunmaktır. Örneğin geçtiğimiz on yılda makine öğrenmesinin en büyük başarılarından biri fotoğraflardaki yüzleri tanıyabilen programlar oldu. Bu genelde şöyle yapılır: Öğrenilecek olan neyse, programa onun örneklerini sağlarsınız. Bir program insan yüzlerini tanıyacaksa, insan yüzleriyle isim etiketleri olan bir sürü fotoğraf vererek onu eğitirsiniz. Burada hedef, programa daha sonra sadece bir fotoğraf sunulduğunda oradaki kişinin adını doğru olarak verebilmesini sağlamaktır.

Zekâya dayalı davranışla ilgili görünen diğer iki yeti ise **problem çözme** ve **planlama**dır. İkisi de verili birtakım eylemlerle belli

bir amaca ulaşabilmeyi gerektirir; buradaki zorluk, eylemlerin nasıl bir sıra izleyeceğini bulmaktır. Örneğin satranç veya Go'da amaç oyunu kazanmaktır; eylemler olası hamleler, zorluk ise ne zaman hangi hamlenin yapılacağını kestirmektir. Problem çözme ve planlamanın en temel zorluklarından bir diğeri ise, *teoride* gayet yapılabilirmiş gibi görünmelerine rağmen, olası alternatiflerin sayıca çokluğunun pratikte bunları yapılabilir olmaktan çıkarmasıdır.

Zekâyla ilişkili en üst düzey yeti muhtemelen akıl yürütme, muhakemedir: Mevcut olgulardan yola çıkarak sağlam bir biçimde yeni bilgiler edinmeyi içerir. Çok bilinen bir örnek verecek olursak, "Bütün insanlar ölümlüdür" bilgisine ve "Michael bir insandır" bilgisine sahipseniz, akıl yürüterek "Michael ölümlüdür" sonucuna varabilirsiniz. Sahiden zekâya dayalı bir sistem, yeni edindiği bu bilgiden yola çıkarak daha başka sonuçlara da varabilir. Örneğin "Michael ölümlüdür" bilgisine sahip olmak, "gelecekte Michael ölecektir" ve "Michael ebediyen ölü olarak kalacaktır" vb. sonuçlara varmanıza imkân sağlar. Otomatikleştirilmiş muhakeme, bilgisayara *mantık çerçevesinde* akıl yürütme yetisini vermekle ilgilidir. 3. Bölüm'de, uzun süre boyunca YZ'nin başlıca amacının bu tür akıl yürütme olduğunun düşünüldüğünü, ancak zamanla bunun anaakım bir görüş olmaktan çıktığını, yine de otomatikleştirilmiş muhakemenin hâlâ önemli bir konu sayıldığını göreceğiz.

Son olarak, **doğal dili anlama** bilgisayarların İngilizce veya Çince gibi insan dillerini yorumlayabilmesini gerektirir. Günümüz itibarıyla bir programcı bir bilgisayara uyulacak bir tarif (algoritma veya program) verecek olsa, bunu *yapay* bir dil kullanarak, herhangi bir belirsizliğe mahal vermeyecek kadar kesin biçimde tanımlanmış kurallar çerçevesinde özel olarak inşa edilmiş bir dille yapması gerekir. Python, Java ve C gibi tanınmış örneklerinden bildiğimiz bu diller İngilizce veya Çince gibi doğal dillerden *çok çok* basittir. Uzun zaman boyunca, doğal dilleri anlamaya yönelik başlıca yaklaşım, nasıl ki bilgisayar dillerini tanımlayan kesin kurallar varsa, doğal diller için de tanımlayıcı kesin kurallar ortaya koymaya çalışmaktan ibaretti. Ancak bunun imkânsız olduğu anlaşıldı. Doğal diller bu şekilde tanımlanamayacak kadar esnek, belirsiz ve akışkan-

dır; gündelik durumlarda kullanım şekilleri, bu dilleri kesin olarak tanımlamaya yönelik girişimleri boşa çıkarır.

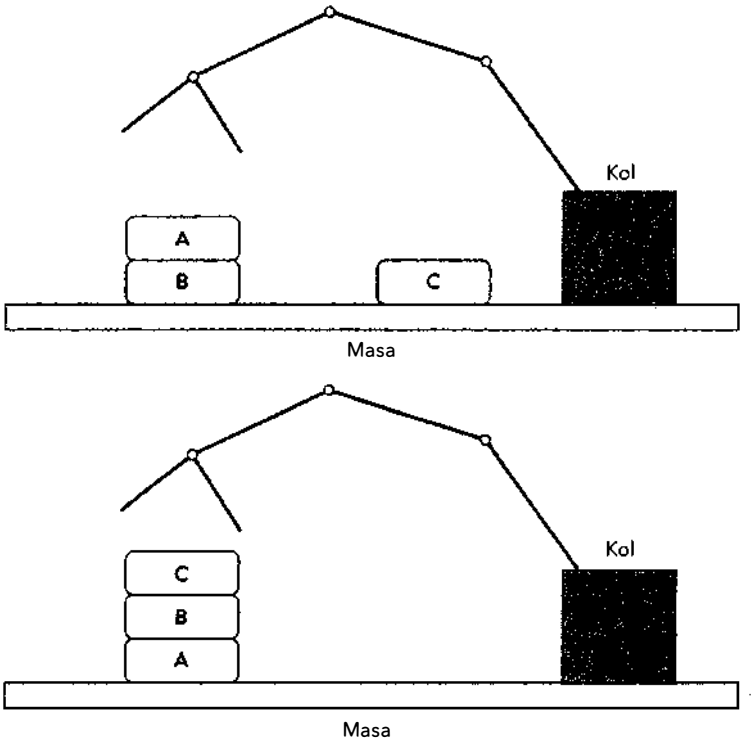
## SHRDLU ve Blok Dünyası

Altın Çağ'ın en çok takdir edilen başarılarından biri de SHRDLU sistemiydi (bu acayip isim, o dönem kullanılan dizgi makinelerinin klavyesinin bir sütununda harflerin yukarıdan aşağıya böyle sıralanmasından geliyor – bilgisayar programcıları sadece meslekten olanların anlayabileceği şakalara bayılıyorlar gerçekten). Stanford Üniversitesi doktora öğrencilerinden Terry Winograd'in 1971'de geliştirdiği[5] SHRDLU, YZ için başlıca iki yetiyi sergilemeyi amaçlıyordu: problem çözme ve doğal dili anlama.

SHRDLU'nun problem çözme bileşeninin temelinde, YZ alanındaki en meşhur deney senaryolarından biri haline gelecek olan **Blok Dünyası** vardı. Belli sayıda renkli nesneden oluşan (prizma, küp ve piramit şeklinde bloklar) bir ortam simülasyonuydu bu. Gerçek bir robot inşa etmek yerine bir ortamı simüle etmenin gerekçesi, problemin karmaşıklığını daha rahat baş edilebilir bir seviyeye getirmekti. SHRDLU'nun Blok Dünyası'nda problem çözme, bir kullanıcının talimatları doğrultusunda ve bir robot kol marifetiyle birtakım nesneleri düzenlemek anlamına gelir. Şekil 2 bunu görselleştiriyor: İlk kısımda Blok Dünyası'nın baştaki halini, ikinci kısımdaysa nihai olarak hedeflenen halini görüyoruz. Burada işin püf noktası, başlangıç durumunu *nasıl* hedef duruma getireceğimizi bulmak.

Bu dönüşümü gerçekleştirmek için sadece belli eylemleri gerçekleştirebiliyorsunuz.

- *X nesnesini masadan al.* Burada robot kol *x* nesnesini (prizma veya piramit) masadan alır. Bu ancak o anda *x* nesnesi masanın üstündeyse ve robot kol boştaysa yapılabilir.
- *X nesnesini masaya koy.* Bu ancak o anda robot kol *x* nesnesini taşıyorsa yapılabilir.
- *X nesnesini y nesnesinden al.* Bu ancak o anda robot kol boştaysa, *x* nesnesi *y* nesnesinin üstündeyse ve *x* nesnesinin üstünde bir şey yoksa yapılabilir.




Şekil 2: Blok Dünyası. Üstte başlangıç durumunu, altta hedef durumu görüyoruz. Asıl soru şu: İlk halinden ikinci haline nasıl gelebiliriz?

- *X nesnesini y nesnesinin üstüne koy*. Bu ancak o anda robot kol *x* nesnesini taşıyorsa ve *y* nesnesinin üstünde bir şey yoksa yapılabilir.

Blok Dünyası'nda olanlar nihayetinde sadece bu eylemlerden ibarettir. Şekil 2'deki başlangıç durumundan hedef duruma nasıl geçeceğimizi planlarken sırayla şu eylemlerle başlayabiliriz:

- A nesnesini B nesnesinden al
- A nesnesini masaya koy
- B nesnesini masadan al

Blok Dünyası'nın bu eylemler gerçekleştirildikten sonraki hali nasıldır? Hedef duruma ulaşmak için bundan sonra sırasıyla hangi eylemlerin gerçekleştirilmesi gerektiğini bulabildiniz mi? (Zor değil sadece biraz uğraşmanız lazım.)

Blok Dünyası YZ alanında en çok işlenen senaryolardan biridir herhalde, çünkü bu dünyadaki görevler (birtakım şeyleri bir yerden alıp başka bir yere koymak) gerçek dünyada robotların yapmasını beklediğimiz türden şeyleri akla getirir. Ne var ki SHRDLU'daki (ve müteakip pek çok araştırmadaki) haliyle Blok Dünyası senaryosu çok kısıtlı olduğu için pratikte hakikaten işe yarayacak YZ teknikleri geliştirmeye pek bir katkı sağlayamaz.

Birincisi, Blok Dünyası'nın her türlü dış etkiye *kapalı* olduğu varsayılır. Buna göre Blok Dünyası'nda değişime yol açan tek şey SHRDLU'dur. Mesela yalnız yaşamaya benzetilebilir bu: Eğer yalnız yaşıyorsanız, yarın sabah uyandığınızda, anahtarlarınızı dün akşam nereye bıraktıysanız orada bulursunuz; ama yalnız yaşamıyorsanız, dün akşam başka birinin anahtarlarınızı ellemiş olma olasılığı her zaman vardır. Yani SHRDLU *x* bloğunu *y* bloğunun üstüne koyduktan sonra, kendisi ellemediği sürece, gönül rahatlığıyla onu yine orada bulacağını varsayabilir. Ama gerçek dünya böyle değildir: İçinde yaşadığı dünyadaki tek aktörün kendisi olduğu varsayımıyla yola çıkan her YZ sistemi çoğu zaman bir şeyleri yanlış anlar.

İkincisi, ve belki daha önemlisi, burada Blok Dünyası bir *simülasyondur*. SHRDLU aslında nesneleri bir robot kolla bir yerden alıp bir yere koymaz, sadece koyarmış gibi yapar. Bir dünya modelini korur ve eylemlerinin bu dünyadaki etkilerini modeller. Dünya modelini inşa ederken gerçek dünyaya hiç bakmaz ve kendi modelinin doğru olup olmadığını kontrol etmez. Fazla basite kaçan bir varsayımdır bu; zaten sonradan araştırmacılar da bir robotun gerçek dünyada karşılaşacağı sahiden zor problemlerin çoğunun gözardı edildiğini ileri sürmüşlerdir.

Bu noktayı daha iyi anlamak için, isterseniz biraz da "x nesnesini y nesnesinden al" eylemi üstünde duralım. SHRDLU açısından burada tek bir eylem söz konusudur: Robotun eylemin *tamamını* tek adımda gerçekleştirdiği varsayılır, eylemin *fiilen ne gibi şeyleri içer-*

*diğiyle* ilgilenmesine gerek yoktur. Dolayısıyla kolu "kontrol eden" programın tek yapması gereken, verili görevi yerine getirmek için gerçekleştirilecek böyle eylemlerin nasıl bir sıra izlemesi lazım onu bulmaktır. Bu eylemlerin fiilen nasıl gerçekleştirileceği gibi karışık bir meseleyle ilgilenmesi gerekmez. Ancak gerçek dünyada, sözgelimi bir depodaki robotun bu eylemi gerçekleştirmesi için, söz konusu iki nesneyi teşhis edebilmesi, kolun doğru yere varıp ilgili nesneyi almasını sağlayarak karmaşık bir hareket problemini uygulamaya geçirmeyi başarması gerekecektir. Bu "atomik" eylemin son kısmı, yani kolun ilgili nesneyi alması bile hiç de azımsanacak bir iş değildir, gerçekte gayet önemlidir; gerçek dünyada robotların basit nesneleri oynatmasını sağlamak bile olağanüstü güçtür, bugün bile hâlâ aşılması gereken bir zorluk olarak karşımızda durmaktadır. Bununla ilgili olarak şöyle bir anım var: 1994'te, daha genç bir akademisyenken, Amerikan Yapay Zekâ Derneği'nin Seattle'da düzenlediği bir konferansa katılmıştım. Konferans kapsamında "ofis temizleme yarışması" gibi bir şey yapıyorlardı. Çok net hatırlıyorum, robotların bu işi yapmakta bu kadar beceriksiz olmasına şaşırıp kalmıştım. Amaç, yarışmanın adından da anlaşıldığı üzere, ofis benzeri bir ortamda güvenli bir şekilde hareket edebilen, etraftaki çöpleri tanıyıp toplayabilen bir robot inşa etmekti, yani SHRDLU'dan yirmi yıl sonra Blok Dünyası'nı ete kemiğe büründürmek gibi bir şeydi bu. En iyi robotlar bile gülünç derecede yavaştı, verilen görevi yerine getirmek şöyle dursun, buna az çok yaklaşan bir şeyi bile zar zor yapıyorlardı. Burada robotların bir günahı yok tabii, asıl mesele benim naifliğim, böyle problemlerin aslında ne kadar zor olduğunu anlamamış olmamdı. *Bizim* bunları ıvır zıvır görevler olarak görmemiz haliyle yanılmamıza, bu görevleri yerine getirecek makineler yapmayı da ıvır zıvır bir iş olarak görmemize yol açıyor.

Blok Dünyası caziptir çünkü sezgiseldir, kolay açıklanır, ayrıca gerçek dünyadaki robotların çözmeye çalıştığını tahayyül edebileceğimiz bir problemi andırıyor gibidir. Ama biraz daha yakından bakınca, bir görevi tamamlamak için gerçekleştirilecek eylemlerin nasıl bir sıra izlemesi gerektiği problemine odaklandığı görülür. Bu da çözülmesi gereken bir problemdir elbette ama bir robotun "gerçek"

blok dünyasında karşılaşacağı tek problem değildir, hele ki en zor problem hiç değildir.

SHRDLU'nun en çok takdir edilen özelliklerinden biri, kullanıcının sisteme talimat verirken kullandığı dilin de sistemin geribildirim sağlarken kullandığı dilin de bildiğimiz İngilizce gibi görünmesidir. SHRDLU ile diyaloğun kabaca neye benzediğine dair kısa bir örnek vereyim (italikler SHRDLU'nun cevapları):

BÜYÜK KIRMIZI BİR PRİZMA AL.
*TAMAM (Alır.)*

PİRAMİDİ AL.
*HANGİ PİRAMİDİ KASTETTİĞİNİ ANLAMADIM.*

ELİNDE TUTTUĞUNDAN DAHA UZUN BİR PRİZMA BUL VE ONU KÜPÜN İÇİNE KOY.
*"ONU" DERKEN, ELİMDE TUTTUĞUMDAN DAHA UZUN OLAN PRİZMAYI KASTETTİĞİNİ ANLIYORUM.*
*TAMAM (Bulup koyar.)*

KÜPÜN İÇİNDE NE VAR?
*MAVİ PİRAMİT VE MAVİ PRİZMA.*

PİRAMİT NEYİN ÜSTÜNDE DURUYOR?
*KÜP.*

KAÇ TANE PRİZMA KÜPÜN İÇİNDE DEĞİL?
*DÖRT.*

BUNLARIN EN AZ BİRİ, ALMANI SÖYLEDİĞİM PRİZMADAN DAHA DAR MI?
*EVET, KIRMIZI KÜP.*

ALTINDA BİR ŞEY VAR MI?
*EVET VAR, MASA.*

MASA PRİZMALARI ALABİLİR Mİ?
*HAYIR.*

BİR PİRAMİDİN ALTINDA BİR PRİZMA OLABİLİR Mİ?
*EVET.*

Gördüğünüz gibi ELIZA'nın fersah fersah ötesinde bir diyalog bu; sistem daha zengin, daha doğal, ayrıca karmaşık yapılarla başa çıkabiliyor (BUNLARIN EN AZ BİRİ, ALMANI SÖYLEDİĞİM PRİZMADAN DAHADAR MI?) ve "onu" örneğinde olduğu gibi bu sözcükle diyalogda daha önce geçen bir şeyin kastedildiğini anlayabiliyor.[6] Sistemin bu veçhesi –kullanıcının ilk bakışta bildiğimiz İngilizce gibi görünen bir dil aracılığıyla sistemle etkileşime girebilmesi– nedeniyle, SHRDLU 1970'lerin başında büyük bir coşkuyla karşılandı. Gelgelelim daha sonra anlaşıldı ki burada sadece son derece kısıtlı bir senaryo (Blok Dünyası) söz konusu olduğu için SHRDLU böylesine zengin diyaloglar yaratabiliyordu. Diyaloglar ELIZA'daki gibi şablon şeklinde değildi belki ama yine de çok kısıtlı bir senaryo çerçevesinde ilerliyordu. Sistem ilk çıktığında, cisimleştirdiği tekniklerin çok daha genel doğal dili anlama sistemlerine kapı aralayabileceği yolunda bir umut vardı ama uzun vadede bu beklentiler suya düştü.

Elli yıl sonra bugün durduğumuz yerden SHRDLU'nun kısıtlarını çok rahat görebiliyoruz tabii. Bununla beraber, sistemin o dönemde ne kadar büyük bir yankı uyandırdığını ve YZ sistemlerinde bir dönüm noktası olduğunu da unutmamamız lazım.

## Robot SHAKEY

YZ hep robotlarla ilişkilendirilmiştir, bilhassa da medya alanında. 1927 tarihli Fritz Lang klasiği *Metropolis*, YZ'yi robotlara eşitleyerek, kendinden sonra gelen onlarca tasvir için bir şablon oluşturmuştur: iki kolu, iki bacağı, bir kafası olan... ve ölüm saçan bir robot. Bugün bile popüler basında neredeyse YZ ile ilgili tek bir makale dahi yoktur ki yanına *Metropolis*'in "makine insan"ının hık demiş burnundan düşmüş bir robot resmi iliştirilmiş olmasın. Genelde robotların ve özelde insansı robotların kamuoyunun gözünde YZ'nin alametifarikası haline gelmesinin çok da şaşırtıcı bir tarafı yok bence. Ne de olsa insanların arasında yaşayan, çalışan robotlar fikri YZ hayalinin en bariz tezahürüdür herhalde – kim bir dediğini iki etmeyen bir robot uşağı olsun istememiştir ki?

Dolayısıyla Altın Çağ'da robotların YZ hikâyesinin nispeten kü-

çük bir kısmını oluşturduğunu öğrenince şaşıracaksınız muhtemelen. Gayet dünyevi nedenleri var bunun: Robot inşa etmek pahalıya patlar, zaman alır ve açıkçası *zordur*. Yani 1960'lar veya 1970'lerde bir doktora öğrencisinin tek başına çalışarak araştırma düzeyinde bir YZ robotu inşa etmesine imkân yoktu. Sırf bu işe tahsis edilmiş büyük bir araştırma laboratuvarı, kendini bu işe vakfetmiş mühendisler ve başkaca tesisler olması lazımdı; ve her halükârda, YZ programlarını çalıştıracak kadar güçlü bilgisayarlar böyle otonom robotlar için fazla büyük, fazla ağırdı. Sonuç olarak araştırmacılar için SHRDLU gibi gerçek dünyada çalışıyor*muş gibi* görünen programlar inşa etmek, gerçek dünyada bütün karmaşıklığıyla *bilfiil* işleyen robotlar inşa etmeye kıyasla, daha kolay ve daha ucuzdu.

YZ'nin erken dönemlerinde çok sayıda YZ robotu yoktu belki ama bu konuda müthiş bir deney yapılmıştı: Stanford Araştırma Enstitüsü'nde 1966-72'de gerçekleştirilen SHAKEY projesi.

SHAKEY hareket eden ve gerçek dünyada kendisine verilen görevleri nasıl yerine getireceğini kendi başına bulan bir robot inşa etmeye yönelik ilk ciddi girişimdi. SHAKEY'nin içinde bulunduğu ortamı algılaması, nerede olduğunu ve etrafında neler olduğunu anlaması; kullanıcıların verdiği görevleri nasıl yerine getireceğini kendi başına çözüp planlaması, planladıklarını bilfiil yapması ve bütün bu zaman zarfında her şeyin başta tasarlandığı gibi ilerlemesini sağlaması lazımdı. Söz konusu görevler bir ofis ortamında birtakım nesnelerin, sözgelimi kutuların bir yerden bir yere taşınmasını gerektiriyordu. İlk bakışta SHRDLU'ya benzetilebilir ama arada belirleyici bir fark var: SHAKEY gerçek bir robottu, gerçek nesneleri hareket ettiriyordu. Ve böyle bir robot yapmak çok daha büyük bir zorluktu.

SHAKEY'nin bütün bunları yapabilmesi için, gerçekleştirilmesi birbirinden zor görünen bir dizi YZ yetisini bünyesinde toplaması gerekiyordu. Her şeyden önce, mühendisliği zor bir işti bu: Geliştirenlerin robotu bilfiil inşa etmesi gerekiyordu. Ofis ortamında rahat hareket edecek kadar küçük ve kıvrak olmalıydı, etrafında nelerin bulunduğunu doğru bir biçimde anlamasını sağlayacak kadar güçlü sensörlere ihtiyacı vardı. Bu nedenle SHAKEY bir televizyon kamerasının yanı sıra nesnelerle arasındaki mesafeyi belirlemesini sağ-

layan lazer menzilölçerlerle donatıldı; ayrıca etraftaki engelleri tespit etmesini sağlayan, "kedi bıyığı" denen çarpma dedektörleri vardı. İkincisi, SHAKEY'nin içinde bulunduğu ortamda her yöne hareket edebilmesi lazımdı. Üçüncüsü, SHAKEY'nin kendisine verilen görevleri nasıl yerine getireceğini planlayabilmesi gerekiyordu. Robotu geliştirenler bu iş için STRIPS'i (Stanford Research Institute Problem Solver: Stanford Stanford Araştırma Enstitüsü Problem Çözücüsü)[7] tasarladılar. Bu sistem artık bütün YZ planlama tekniklerinin atası kabul ediliyor. Ve son olarak, bu yetilerin hepsinin birbiriyle uyum içinde çalışmasını sağlamak gerekiyordu. Hangi YZ araştırmacısına sorsanız size şunu söyler: Bu bileşenlerden sadece birinin bile çalışmasını sağlamak başlı başına zor bir iştir, hepsinin birlikte çalışmasını sağlamak ise misliyle zor bir iş.

Son derece etkileyici olmakla beraber, SHAKEY dönemin YZ teknolojisinin sınırlarını da gözler önüne serer. SHAKEY'nin çalışması için, tasarımcılar robotun karşılaşacağı güçlükleri büyük ölçüde basitleştirmek zorunda kalmıştır. Mesela SHAKEY'nin TV kamerasından aldığı veriyi yorumlama yetisi son derece sınırlıydı, başka bir deyişle etraftaki engelleri tespit etmenin pek ötesine geçmiyordu. Sırf bunu yapabilmesi için bile ortamın özel olarak boyanması ve dikkatle aydınlatılması gerekmişti. TV kamerası için çok fazla güç gerektiğinden kamera sadece gerekli olduğu zamanlarda açılıyordu ve kamera açıldıktan sonra işe yarar bir görüntü elde edinceye kadar yaklaşık on saniye geçiyordu. Geliştiriciler ayrıca dönemin bilgisayarlarının kısıtlarına da toslayıp durdular: SHAKEY'nin bir görevi nasıl yerine getireceğini netleştirmesi 15 dakikayı bulabiliyordu; ayrıca bu zaman zarfında hiç hareket etmeden, çevresinden tamamen izole bir şekilde duruyordu. SHAKEY'nin yazılımını çalıştıracak kadar güçlü bilgisayarlar robotun oradan oraya taşıyamayacağı kadar büyük ve ağır olduğundan, SHAKEY yazılımı çalıştıran bilgisayarla bir radyo bağlantısı kuruyordu.[8] İşin özü, SHAKEY somut bir problemi çözmek için kullanmaya pek uygun değildi.

Muhtemelen ilk gerçek otonom mobil robot olan SHAKEY birbirinden heyecan verici bir dizi yeni YZ teknolojisine öncülük etti. Bu yüzden tıpkı SHRDLU gibi SHAKEY de YZ'nin tarihini anlatan

kitaplarda başköşeye yerleştirilmeyi fazlasıyla hak ediyor. Diğer taraftan SHAKEY'nin kısıtları, YZ'nin gerçek otonom robotlar hayaline ulaşmak için ne kadar uzun bir mesafe katetmesi gerektiğini ve bunun ne denli zor göründüğünü de ortaya koymuş oldu.

### Problem Çözme ve Arama

İnsanı diğer hayvanlardan ayıran başlıca yetilerinden biri de hiç kuşkusuz problem çözmedir. İnternet sincapların[9] veya kargaların[10] yiyeceğe ulaşmak için karmaşık problemleri nasıl da çözdüğünü gösteren videolarla doluysa da, soyut problemleri çözme yetisi söz konusu olduğunda (işin ucunda yiyecek olmadığında dahi) hiçbir hayvanın insanın yanına yaklaşamadığı ortadadır. Ayrıca problem çözme şüphesiz zekâ gerektiren bir şey gibi görünmektedir. İnsana zor gelen problemleri çözebilen programlar inşa edebilseydik, YZ yolunda kuşkusuz büyük bir ilerleme kaydetmiş olurduk, değil mi? Bu ve benzeri nedenlerle Altın Çağ'da problem çözme yoğun bir biçimde üstünde durulan bir konuydu. Dönemin YZ araştırmacılarının sıkça yaptığı tipik alıştırmalardan biri, bir bilgisayarın gazetenizin bulmaca sayfasında bulabileceğiniz türden bir problemi çözmesini sağlamaya çalışmaktı. Gelin, meseleyi biraz daha açmak için, bu tür bulmacaların klasik örneklerinden olan Hanoi Kuleleri'ne bakalım:

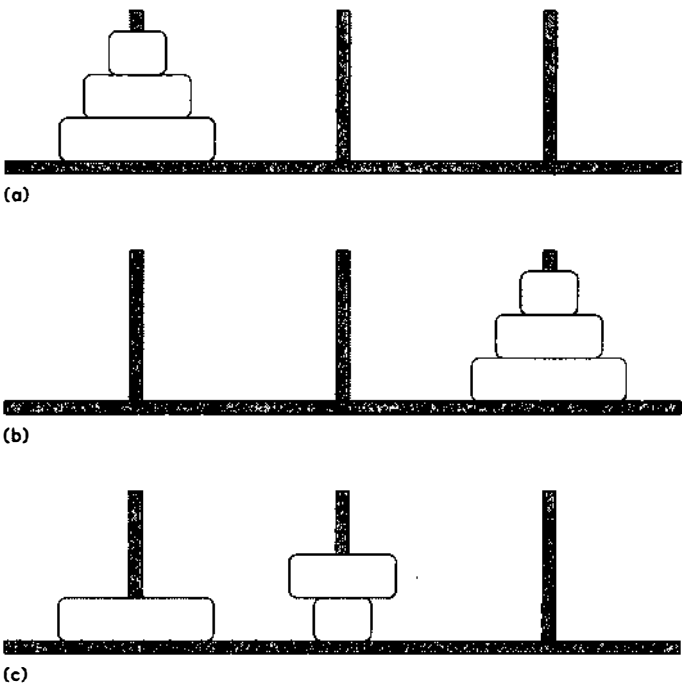
> Ücra bir manastırda 3 direk ve 64 altın disk vardır. Farklı ebatlardaki diskler bu direklere geçirilmiş olarak durur. Başta, bütün diskler en soldaki direkte duruyordur; keşişler diskleri teker teker bir direkten diğerine geçirmektedir. Amaçları bütün diskleri en sağdaki direkte toplamaktır. Bu süreçte keşişlerin uyması gereken iki kural vardır:
>
> 1. Bir seferde sadece tek bir disk hareket ettirilebilir.
> 2. Bir disk, kendisinden daha küçük bir diskin üstüne koyulamaz.

Bütün diskleri en sol direkten en sağdakine taşımak suretiyle problemi çözmek için, bu iki kurala uyarak, hangi eylemleri hangi sırayla gerçekleştirmeniz gerektiğini bulmanız lazım.

Bu hikâyenin kimi versiyonlarında, keşişler görevini tamamlayınca dünyanın sonunun geleceği söylenir. Birazdan göreceğimiz

üzere, bu hikâye doğru olsaydı bile, boşuna uykularınız kaçmasın hiç derdim, çünkü keşişler 64 diski en sağdaki direğe geçirene kadar evrenin sonunun çoktan gelmiş olacağı açık. Zaten bu yüzden bulmacanın mevcut versiyonlarında hep çok daha az sayıda disk vardır. Bulmacanın görsel halini Şekil 3'te bulabilirsiniz: 3 diskli bu versiyonda (a) başlangıç durumudur, (b) hedef durumdur, (c) ise kurallara aykırı bir durumu gösterir.

Öyleyse Hanoi Kuleleri gibi bir problemi çözmek için nasıl bir yol izleyebiliriz sorusunun cevabı arama tekniğinden geçer. Yalnız şuna dikkat etmemiz lazım: YZ sularında dolaşırken "arama" teri-




**Şekil 3:** Hanoi Kuleleri YZ'nin Altın Çağı'nda çokça incelenmiş klasik bir bulmacadır. (a) bulmacanın başlangıç durumunu, (b) ise hedef durumu gösterir. (c)'de ise kurallara aykırı bir durumu görürüz (üstteki disk alttakinden büyüktür).

mini duyduğunuzda bilin ki burada kastedilen web'de arama yapmak (yani google'lamak veya baidu'lamak) değildir. YZ'de arama temel bir problem çözme tekniğidir, bütün olası eylem tarzlarını sistematik bir biçimde gözden geçirmeyi gerektirir. Bilgisayarınızdaki bir satranç oyunundan arabanızdaki uydu navigasyon sistemine kadar envai çeşit program esasen aramaya dayanır. Tekrar tekrar gündeme gelen bu teknik YZ'nin köşetaşlarından biridir.

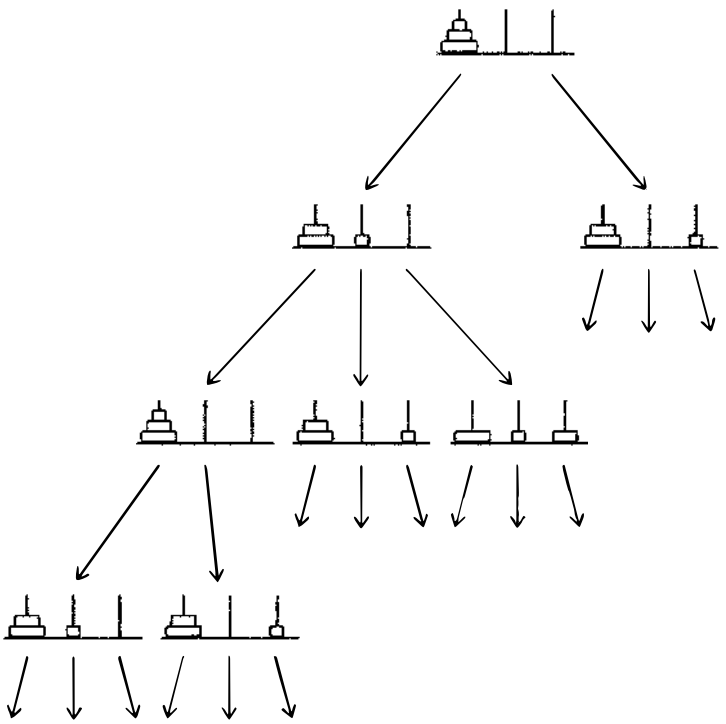
Hanoi Kuleleri türü problemlerin yapısı temelde aynıdır. Blok Dünyası örneğinde de olduğu gibi, problemin **başlangıç durumun**dan ulaşılması amaçlanan **hedef duruma** gelmek için hangi eylemleri hangi sırayla gerçekleştireceğimizi bulmaya çalışırız. YZ terminolojisinde "durum" belli bir problemin belli bir zamandaki hali anlamında kullanılır.

Hanoi Kuleleri türünden problemleri çözmek için aramayı kullanacaksak şu prosedürü izleyebiliriz:

- Birincisi, mevcut her eylemin başlangıç durumu üstündeki etkisini göz önünde bulundururuz. Bir eylemi gerçekleştirmenin etkisi, problemi dönüştürerek yeni bir hal almasına neden olmasıdır.
- Eylemlerden biri hedef durumun ortaya çıkmasını sağladıysa, başardık demektir: Bulmacanın çözümü bizi başlangıç durumundan hedef duruma ulaştıran eylemler ve bunların sırasıdır.
- Ama sağlamadıysa, meydana getirdiğimiz her durum için bu süreci tekrarlar, her bir eylemin bu durumlar üstündeki etkisini göz önünde bulundururuz vs.

Bu tarifi aramaya uyguladığımızda bir **arama ağacı** ortaya çıkar. Şekil 4'te böyle bir arama ağacından bir kesit bulabilirsiniz.

- En başta, sadece en küçük diski hareket ettirebiliriz ve sadece ortadaki veya en sağdaki direğe geçirebiliriz. Yani ilk hamle için iki eylem ve bunlara paralel olarak iki yeni durum söz konusu olabilir.
- Küçük diski ortadaki direğe geçirecek olursak (başlangıç durumunun sol alt ucundaki görseldeki gibi), bu yeni durumda üç eylem olasılığı ortaya çıkar: Küçük diski en soldaki direğe geçirebiliriz, küçük diski en sağdaki direğe geçirebiliriz veya ortanca diski en sağdaki direğe geçirebiliriz (ortadaki direğe geçiremeyiz çünkü böylece ortancayı küçüğün üstüne koymuş oluruz ki bu da kurallara aykırıdır).
- vs.



**Şekil 4:** Hanoi Kuleleri bulmacasının arama ağacından bir kesit.

Sistematik bir biçimde arama ağacını meydana getirirken, aşama aşama bütün olasılıkları değerlendirmiş oluyoruz. Yani bir çözüm varsa, bu süreci izleyerek çözümü eninde sonunda buluruz. (Bir önceki cümledeki "eninde sonunda" vurgusu önemli; neden önemli olduğunu ise şimdi göreceğiz.)

Öyleyse Hanoi Kuleleri bulmacasını çözmek için *optimum* hamle sayısı kaçtır (yani başlangıç durumundan hedef duruma en az kaç hamleyle ulaşabiliriz)? Bulmacanın 3 diskli versiyonu için bu sorunun cevabı 7'dir, problemi asla 7'den az hamleyle çözemezsiniz. Diğer taraftan 7'den fazla hamleyle çözebilirsiniz (böyle *çok sayıda*

çözüm vardır), ama bu da çözümü optimum olmaktan çıkarır, çünkü sadece 7 hamleyle çözmeniz mümkündür. Şimdi, tanımladığımız arama süreci, her aşamada bütün olasılıkları değerlendirerek bütün bir arama ağacı sistemini ortaya koyduğundan, bu yol alelade bir çözüm bulmayı değil, *en kısa* çözümü bulmayı garantiler. Üstelik, bilgi işlemsel tarifler söz konusu olduğunda, kapsamlı arama gayet basittir, bu tarifi uygulamaya geçirecek bir program yazmak kolaydır.

Ancak Şekil 4'ü üstünkörü incelediğimizde bile hemen fark ederiz ki, tanımladığımız bu basit kapsamlı arama düpedüz aptalcadır. Mesela arama ağacının en sol dalına bakacak olursanız, iki hamle sonra tekrar en başa döndüğünü görürsünüz, yani boşa kürek çekmişsinizdir. Hanoi Kuleleri bulmacasını çözmeye çalışan *siz* olsanız, çözümü ararken bir-iki kere boş bulunup böyle bir hata yaparsınız belki ama çok geçmeden bunu fark edip boşa çaba harcamaktan kaçınmaya başlarsınız. Ama verilen tarife uyan bir bilgisayar sizin gibi hareket etmez: Bu tür alternatifler evvelce bir yere varmadığını gördüğümüz durumları tekrar etmeleri bakımından zaman kaybından başka bir anlama gelmese bile, tarif bütün alternatifleri sistemli bir biçimde üretmeyi gerektirir.

Gelgelelim kapsamlı aramanın, verimsizliğine gelinceye kadar, daha temel bir sorunu vardır. Şöyle birkaç deneme yapacak olursanız, Hanoi Kuleleri bulmacasının bütün olası durumlarında 3 hamle yapma imkânı olduğunu görürsünüz. Yani bu bulmacanın **dallanma faktörü** 3'tür. Farklı arama problemlerinin dallanma faktörleri de farklı olacaktır. Örneğin Go oyununda dallanma faktörü 250'dir, yani oyunun verili bir durumunda her oyuncu için ortalama 250 olası hamle vardır. Bunu daha iyi anlamak için, bir arama ağacının dallanma faktörüne oranla ne kadar *büyüdüğüne*, ağacın verili bir aşamasında kaç farklı olası durum olduğuna bakalım. Örneğin Go oyununda:[11]

- Ağacın birinci aşamasında 250 durum vardır çünkü oyunun başlangıç durumu için 250 olası hamle mevcuttur.
- Arama ağacının ikinci aşamasında 250x250=62.500 durum olacaktır çünkü birinci aşamadaki 250 durumun her biri için her olası hamleyi göz önünde bulundurmamız gerekir.

- Arama ağacının üçüncü aşamasında 250x62.500=15,6 milyon durum olacaktır.
- Ve arama ağacının dördüncü aşamasında 250x15,6 milyon=3,9 milyar durum olacaktır.

Günümüzde alelade bir masaüstü bilgisayarın Go oyunu için arama ağacının ilk dört aşamasından fazlasını depolamasına yetecek kadar belleği yoktur, halbuki oyun yaklaşık 200 hamle sürer. Ve 200 hamlelik bir arama ağacındaki durumların sayısı havsalamızın almayacağı kadar çoktur. Evrenimizdeki atomların sayısından yüzlerce kat daha fazladır bu. Bildiğimiz kadarıyla bilgisayar teknolojisindeki hiçbir gelişme bu büyüklükte arama ağaçları yapmaya olanak sağlamayacaktır.

Arama ağaçları hızlı büyür. Absürd derecede, hayal bile edemeyeceğimiz kadar hızlı. **Kombinasyon patlaması** YZ pratiğine ilişkin en önemli problemlerdendir, çünkü YZ problemlerinde arama neredeyse olmazsa olmazdır.[12] Arama problemlerini hızlıca çözmenin şaşmaz tarifini bulabilseydiniz, yani kapsamlı aramayla aynı sonuca onca zahmete katlanmadan ulaşabilseydiniz bayağı meşhur olurdunuz; ayrıca an itibarıyla YZ için bir hayli zor olan pek çok problem bir anda kolaylaşıverirdi. Ama ne yazık ki bulamayacaksınız. Kombinasyon patlamasının etrafından dolaşamayız, onunla birlikte çalışmamız gerekiyor.

Kombinasyon patlaması YZ'nin ilk zamanlarından beri temel bir problem kabul edilmektedir. McCarthy Darmouth yaz okulunda üstünde çalışılacak konulardan biri olarak bunu seçmiştir mesela. Çok geçmeden daha ziyade aramayı randımanlı hale getirme meselesine odaklanılmıştır. Bunun için birçok farklı yaklaşım olduğu ortadadır.

Bunlardan biri, aramayı bir noktada yoğunlaştırmaktır. Bunun en bilinen yollarından biri ise **derinlik öncelikli arama**dır: Ağacı aşama aşama inşa edeceğinize bir dalı üstünden ilerler, o dalda çözüme ulaşana kadar veya buradan bir sonuç çıkmayacağına kanaat getirene kadar devam edersiniz. Dalın bir yerinde takılıp kalırsanız (örneğin Şekil 4'ün en sol dalında gördüğümüz gibi başa dönerseniz) bu dalı bırakıp bir sonraki dal üstünde çalışmaya başlarsınız.

Derinlik öncelikli aramanın başlıca avantajı arama ağacının tamamını depolamaya gerek kalmamasıdır, sadece üstünde çalıştığımız dalı depolasak yeter. Diğer taraftan şöyle de büyük bir dezavantajı vardır: Yanlış dalı seçecek olursak, incelememize ne kadar devam edersek edelim bir çözüme ulaşamayabiliriz. Yani derinlik öncelikli arama yapacaksak eğer, hangi dalı inceleyeceğimizi iyi bilmemiz lazım. İşte tam bu noktada **sezgisel arama** devreye giriyor. Sezgisel arama fikri "ampirik kurallar"a, aramamızı hangi noktaya yoğunlaştırmamız gerektiğini gösteren **sezgisel yöntemler**e başvurmaya dayanır. Çözmekle ilgilendiğimiz belli başlı problemler için genelde, aramayı mümkün olan en iyi istikamete yoğunlaştırmayı garantileyen sezgisel yöntemleri bulamayız ama kimi durumlarda pek iyi sonuç vermeyebileceğini bildiğimiz sezgisel yöntemleri bulabiliriz.

Gayet doğal bir fikir olan sezgisel arama yıllar içinde tekrar tekrar icat edilmiştir, dolayısıyla bunu kimin icat ettiğini tartışmak çok da anlamlı değildir. Diğer taraftan sezgisel aramanın bir YZ programında ilk defa ne zaman ve kim tarafından kapsamlı bir biçimde uygulandığını rahatlıkla söyleyebiliriz: dama oynayan bir programda, 1950'lerin ortasında, IBM çalışanı Arthur Samuel tarafından. Samuel bu programı yazarak YZ alanında birçok bakımdan çığır açmıştır. Birincisi, Samuel'ın dama oyuncusu YZ tekniklerini kanıtlamak için masa oyunlarını kullanma geleneğini başlattı. Eleştirilerden nasibini almakla beraber bugün bile hâlâ yaygın olarak kullanılan bir tekniktir bu. İkincisi, az önce sadece değindiğimiz ama birazdan ayrıntılı olarak ele alacağımız gibi, program sezgisel aramaya dayanıyordu. Ve son olarak, bu noktayı tartışmayı esasen daha sonraya bırakacağız ama, belki de gerçek anlamda ilk makine öğrenmesi programıydı bu: Program kendine nasıl dama oynanacağını öğretiyordu.

Samuel'ın programının başlıca bileşeni belli bir pozisyonu değerlendirme tarzı, bu pozisyonun belli bir oyuncu için ne kadar "iyi" olduğunu kestirebilmesiydi: Tahmin edersiniz ki oyuncu tahta üstündeki taşlarının pozisyonu "iyi" ise muhtemelen oyunu kazanacak, "kötü" ise muhtemelen kaybedecektir. Samuel taşların tahta üstündeki pozisyonunun değerini hesaplamak için birtakım özellikler-

den yararlanıyordu. Bu özelliklerden biri tahta üstündeki taşların sayısına bakmak olabilir mesela; ne kadar çok taşınız varsa o kadar iyi durumdasınız demektir. Program daha sonra bu farklı farklı özellikleri birleştirip pozisyonun durumuna dair tek bir genel değerlendirme yapıyordu. Bunu izleyen tipik bir sezgisel yöntem de taşların tahta üstünde en iyi pozisyona ulaşmasını sağlayacak bir hamle seçmek oluyordu.

Pratikte, sadece sezgisel yöntemlerden fazlası gerekliydi; dama rakibe karşı oynanan bir oyun olduğundan, karşınızdakinin muhtemelen nasıl hareket edeceğini de göz önünde bulundurmanız gerekiyordu. Samuel'ın dama oyuncusu rakibinizin sizin için mümkün olan en kötü hamleyi yapacağını varsayıyordu. Oyun oynamadaki en temel fikirlerden biri olan bu "en kötü senaryo" yaklaşımına (rakibinizin sizin en az puanı almanız için uğraştığını varsayarak en çok puanı almaya çalışmak) **minimaks arama** denir.

Samuel'ın programı damada hiç fena değildi. Birçok açıdan etkileyici bir başarı tabii, hele ki o dönemde ortalama bir bilgisayarın bugünkülere kıyasla ne kadar ilkel kaldığını düşününce daha da etkileyici: Samuel'ın kullandığı IBM 701 sadece birkaç metin sayfası uzunluğuna denk programların altından kalkabiliyordu. Günümüzde ortalama bir masaüstü bilgisayar bunun milyonlarca katı belleğe sahiptir, bundan milyonlarca kat daha hızlıdır. Bütün bu kısıtları göz önünde bulundurunca, damayı iyi oynamak şöyle dursun, sadece oynayabiliyor olması bile müthiş etkileyicidir.

Samuel'ın dama oyuncusu gibi sezgisel aramaya dayalı ilk çalışmalar daha ziyade amaca özel sezgisel yaklaşımları, "deneyip görelim" yaklaşımlarını benimsemiştir. Bununla beraber, 1960'ların sonunda Nils Nilsson ve Stanford Araştırma Enstitüsü'ndeki meslektaşları bu konuda bir atılım gerçekleştirerek, daha önce bahsettiğimiz SHAKEY projesinin bir parçası olan, A* tekniğini geliştirmişlerdir. Böylece bir sezgisel yöntemin "iyi" olup olmadığını anlamamızı sağlayan birtakım basit kurallar belirlenmiştir. O zamana kadar bir tahmin oyunundan çok da farklı olmayan sezgisel arama, A* ile beraber matematiksel olarak gayet iyi anlaşılan bir süreç haline gelmiştir.[13] A* bugün hesaplamanın en temel algoritmalarından biri ka-

bul ediliyor ve uygulamada yaygın olarak kullanılıyor. Dolayısıyla A* ile bir biçimde karşılaşma ihtimaliniz yüksek; araç içi navigasyon sistemlerinin yazılımını çalıştıran algoritma motoru bu mesela. Ancak A* ne kadar güzel olursa olsun kullanılan özel sezgisel yönteme bağlı olması bakımından sorunlu: İyi bir sezgisel yöntem hızlıca çözüme ulaştırırken, zayıf bir sezgisel yöntemin kıymeti olmuyor. Ve A* tekniği de belli problemler için nasıl iyi sezgisel yöntemler bulacağımız sorusuna herhangi bir cevap sunamıyor.

## YZ Karmaşıklık Bariyerine Tosluyor

Hatırlayacak olursak, Alan Turing bilgisayarı esasen o dönemin en büyük matematik problemlerinden birini çözmek için icat etmişti. Turing'in bilgisayarı aslında bilgisayarın yapamayacağı şeyler olduğunu, kimi problemlerin tabiatı gereği karar verilemez olduğunu göstermek için icat etmesi bilim tarihinin en büyük cilvelerinden biridir.

Turing'in bu sonuçları ortaya koymasından sonraki onyıllarda, dünyanın dört bir yanındaki üniversitelerin matematik departmanları bilgisayarın sınırlarını keşfetmeyi, neleri yapıp neleri yapamayacağını incelemeyi kendilerine iş edindiler. Bu iş daha ziyade tabiatı gereği karar verilemez olan (bilgisayarın çözemediği) problemler ile karar verilebilir olanları (bilgisayarın çözebildiklerini) birbirinden ayırmaya yoğunlaşıyordu. Dönemin ilgi çekici keşiflerinden biri de karar verilemez problemlerin *hiyerarşisiydi*: Bir problem sadece karar verilemez olmakla kalmayabiliyordu, aynı zamanda *yüksek derecede* karar verilemez olabiliyordu (bazı matematikçiler böyle şeylere bayılır).

1960'larda, bir problemin karar verilebilir olup olmadığının netleşmesiyle meselenin bitmediği anlaşılmaya başladı. Bir problemin Turing'in çalışması bağlamında karar verilebilir olması, *pratikte* muhakkak çözülebileceği anlamına gelmiyordu. Turing'in teorisine göre çözülebilir olan kimi problemler görünüşe bakılırsa bilfiil çözmeye yönelik her türlü girişime direniyordu; çözüm için gerekli bellek miktarını karşılamak mümkün değildi veya çözüm o kadar ama

o kadar yavaş ilerliyordu ki artık uygulanabilir olmaktan çıkıyordu. YZ problemlerinin pek çoğunun bu acayip kategoriye girdiği, rahatsız edici bir biçimde aşikâr hale geliyordu.

Turing'in teorisine göre hemen çözülüverecekmiş gibi görünen şu örneğe bakalım:

İşyerinizde birbirinden yetenekli dört çalışan var: John, Paul, George ve Ringo. Belli bir proje için bu kişilerden üçünün dahil olduğu bir ekip oluşturmanız gerekiyor. Ne yazık ki, aralarındaki şahsi anlaşmazlıklar nedeniyle, John ve Paul birlikte çalışamaz. Buna göre bir ekip oluşturabilir misiniz?

Bu durumda cevap tabii ki "evet"tir: John, George ve Ringo'dan veya Paul, George ve Ringo'dan bir ekip oluşturabilirsiniz. (Dikkat ederseniz problemin bunlardan *başka* bir çözümü yok.)

Şimdi bir kısıtlama daha getirelim: John ve George birlikte çalışamaz. Hâlâ bir ekip oluşturabilir miyiz? Evet: Paul, George ve Ringo'dan bir ekip oluşturabiliriz.

Son bir kısıtlama daha: Paul ve George birlikte çalışamaz. Bu son kısıtlamayla sorumuzun cevabı "hayır" olur, "yasak çiftler" kısıtlamalarına uyan üç kişilik bir ekip kurulamaz. Bu problem biraz daha genel bir biçimde şöyle ifade edilebilir:[14]

*n* sayıda kişinin bir listesi (yukarıdaki örnekte kişiler John, Paul, George ve Ringo olduğundan $n = 4$'tür) ve bir de "yasaklı çiftler" listesi veriliyor (örn. "John ve Paul birlikte olamaz"). Ardından *m* sayısı veriliyor; ilk listedekilerden tam olarak *m* sayıda kişiden oluşan ve yasaklı çiftler kısıtlamasına uyan, yani birlikte olmaması gereken iki kişinin yan yana gelmediği bir ekip oluşturmanın mümkün olup olmadığı soruluyor. (Problemin bir anlam ifade etmesi için *m*'nin *n*'den küçük olması gerektiği açık.)

İlk bakışta, problemin *teoride* çözülebilecek gibi göründüğünün altını çizelim. En basiti, *n* sayıda kişi arasından *m* kişilik bütün olası ekipler sıralanır, ardından da bu ekiplerin verili kısıtlara uyup uymadığı tek tek kontrol edilir. Bilgisayarda böyle bir tarif programlamak çok kolaydır. Dolayısıyla problem Turing'in sınıflandırmasına göre gereksiz faktörleri göz önünde bulundurmakla beraber karar verilebilir bir problemdir.

Peki bu tarifi kullanırsanız *kaç* olası ekibi kontrol etmeniz gerekecek? Diyelim ki 4 çalışan söz konusu ve 3 kişilik bir ekip oluşturulacak, bu durumda kontrol edilecek tam 4 ekip vardır. Kolay iş. Peki çalışan sayısı artarsa ne olur? Gelin şimdi de buna bakalım.

10 kişilik bir çalışan havuzu var diyelim ve bu sefer 5 kişilik bir ekip oluşturmak istensin, bu durumda 252 olasılığı değerlendirmek gerekir. Sıkıcı bir iş, ama yapılır mı yapılır. Dikkat ederseniz olası ekiplerin sayısı, çalışan sayısından veya hedef ekibi meydana getiren kişi sayısından çok daha hızlı artıyor.

100 çalışandan meydana gelen bir havuz olsun ve 50 kişilik bir ekip oluşturmak istensin. Bu durumda *100 milyarlarca* olası ekibi değerlendirmek gerekir. Günümüzde hızlı bir bilgisayar saniyede 10 milyar olası ekibi değerlendirebilir. İlk duyduğunuzda kulağa bayağı çok gibi geliyor, değil mi? Ne var ki bir daha düşününce, bütün alternatifleri değerlendirmek için gerçekten çok ama çok fazla zaman gerektiği anlaşılıyor; bu hesabın bitmesine kalmadan çoktan evrenin sonu gelmiş olabilir. Intel'deki parlak insanların daha hızlı bir çip geliştirmesini beklemek de çare değil: Mevcut bilgisayar teknolojisinde tasavvur edebileceğimiz hiçbir gelişme, bütün bu alternatifleri makul bir sürede kontrol edebilecek bir makinenin ortaya çıkmasıyla sonuçlanmayacaktır.

Burada tanık olduğumuz fenomen, kombinasyon patlamasının bir başka örneğidir. Kombinasyon patlamasının ne olduğunu arama ağaçlarından bahsederken görmüştük; arama ağacındaki her müteakip katman ağacın büyüklüğünü misliyle artırıyordu. Kombinasyon patlaması birbiri ardına bir dizi seçim yapmanız gereken durumlarda ortaya çıkar; her seçim olası sonuçların toplam sayısını *misliyle artırır*. Çalışanlardan ekip kurma örneğinde, yapmamız gereken seçimler belli bir kişiyi veya kişileri ekibe dahil edip etmemekle ilgilidir; aday havuzuna tek bir çalışan eklemek bile değerlendirmemiz gereken olası ekip sayısını *iki katına* çıkarabilir.

Kısacası, kısıtlamalara uygun olup olmadığını görmek için bütün müstakbel ekipleri tek tek araştırma yaklaşımı uygulanabilir olmaktan çok uzaktır. *Teoride* işler belki (yeterince zaman devam edersek doğru cevaba ulaşırız) ama *pratikte* işlemediği açıktır (bü-

tün alternatifleri değerlendirmek için gereken zaman, ulaşmamızın mümkün olmadığı kadar uzun bir süre haline geliverir).

Daha önce bahsettiğimiz basit kapsamlı arama ham bir tekniktir. Öyleyse sezgisel yöntemleri kullanalım desek, e onların da işleyeceğinin garantisi yoktur. Tek tek bütün alternatifleri değerlendirmeyi gerektirmemekle beraber uygun bir ekip bulmayı *garantileyen daha akıllıca* bir yol var mıdır peki? *Yoktur*. Gidişatı bir ölçüde iyileştiren bir teknik bulabilirsiniz ama nihayetinde *kombinasyon patlamasının etrafından dolaşamazsınız*. Bu problemi kesinkes çözen tarifler çoğu durumda uygulanabilir olmayacaktır.

Bunun sebebi elimizdeki problemin bir **NP-tam** problem olmasıdır. Alandan olmayanlar için bu kısaltmayı açalım hemen: NP, *nondeterministic polynomial-time*, yani "deterministik olmayan polinomsal zaman". Terimin teknik anlamı biraz karışık olsa da, meselenin özünü anlamak neyse ki kolaydır.

NP-tam problem, çalışanlardan ekip kurma örneğimiz gibi, bir kombinasyon problemidir. Bu problemde *çözümleri bulmak zordur* (çünkü çok sayıda olasılığı kontrol etmek gerekir) ama *bir çözüm bulup bulmadığınızı doğrulamak kolaydır* (ekip kurma örneğinde, olası bir çözümü sadece yasak çiftler barındırmamasına dayanarak doğrulayabiliriz).

NP-tam problemlerin bir ayırt edici özelliği daha vardır. Ama bunu anlamak için önce başka bir problemden bahsetmemiz lazım (bana güvenin, böyle yapmamın haklı gerekçeleri var). İyi bilinen bir hesaplama problemidir bu, belki duymuşsunuzdur: *seyyar satıcı problemi*.[15]

Seyyar satıcının biri, belli bir güzergâhtaki bir grup şehre uğrayıp sonunda başlangıç noktasına geri dönecek. Bütün şehirler bir yolla birbirine bağlı değil, ama bir yolla birbirine bağlı olan bütün şehirler için seyyar satıcı en kısa rotanın hangisi olduğunu biliyor. Seyyar satıcı bir depo benzinle belli bir kilometre yol gidebiliyor. Öyleyse seyyar satıcının bir depo benzinle bütün şehirlere uğrayıp başlangıç noktasına geri dönmesine olanak sağlayan bir rota var mıdır?

Ekip kurma örneğinde olduğu gibi bunda da bir kombinasyon problemi havası var. Hiç kafamızı yormadan, sadece bütün olası turları

sıralayarak ve tek tek her birinin bir depo benzinle yapılıp yapılamayacağını kontrol ederek çözebiliriz bu problemi. Ancak, tahmin edeceğiniz üzere, şehir sayısı artacak olursa olası tur sayısı da dramatik bir biçimde artar: 10 şehir için 3,6 milyon olası turu, 11 şehir için yaklaşık 40 milyon turu değerlendirmek gerekecektir vb.

Bu durumda, ekip kurma problemi gibi, seyyar satıcı problemi de kombinasyon patlamasından muzdarip demektir. Ama bunun dışında, iki problemin herhangi bir ortak noktası yokmuş gibi görünür. Zaten neden olsun ki? Ekip kurma ile bir güzergâhtaki en kısa rotayı bulmanın nihayetinde birbiriyle ne alakası olabilir? NP-tam problemlerin en can alıcı tarafı budur işte: Tamamen farklı görünseler dahi bunlar nihayetinde *aynı problemdir*.

Tam olarak ne kastettiğimi anlamak için, herhangi bir ekip kurma probleminin doğru cevabını hemen vermesi garanti olan bir tarif buldum diye düşünün. Sonra da diyelim ki bana bir seyyar satıcı problemi veriyorsunuz. Bu problemi alıp bir ekip kurma problemine çeviriyorum, tarifim bunu hemencecik çözüyor ve *elde ettiğim cevap bana verdiğiniz problemin cevabını veriyor*. Ne demek bu? Ekip kurma problemini çabucak çözmenin bir yolunu bulursanız, seyyar satıcı problemini *de* çabucak çözmenin bir yolunu bulmuş olursunuz. Ve yalnızca söz konusu iki problem için geçerli değildir bu, *bütün* NP-tam problemler için geçerlidir.

Bütün NP-tam problemlerin kolaylıkla birbirine dönüştürülebilme özelliği vardır. Bu fevkalade sonuç, bir NP-tam problem ayakta kalırsa hepsinin ayakta kalacağı veya biri düşerse hepsinin düşeceği anlamına gelir: *Tek bir* NP-tam problemi çabucak çözen bir tarif bulabilirseniz, *hepsini çabucak çözen bir tarif de bulmuş olursunuz*. Ama henüz hiç kimse herhangi bir NP-tam problem için randımanlı bir tarif bulabilmiş değil. Dolayısıyla, NP-tam problemlerin verimli bir biçimde çözülüp çözülemeyeceği, bugün bilim dünyasının hâlâ çözülememiş en önemli problemlerinden biri. **P vs NP problemi**[16] denen bu problemi bilim camiasını tatmin edecek herhangi bir biçimde çözebilirseniz, onu 2000 yılında matematikte "Milenyum Problemleri"nden biri olarak tanımlayan Clay Matematik Enstitüsü'nden bir milyon dolarlık bir ödül kazanırsınız. Uzman çevreler NP-

tam problemlerin muhtemelen verimli bir biçimde *çözülemeyeceğini*, ama aynı zamanda bunu kesin bir biçimde bilmekten de bir hayli uzak olduğumuzu düşünüyor.

Üstünde çalıştığınız problemin NP-tam olduğunu keşfederseniz, bilinen teknikler onu çözmeye yaramayacak demektir. Tam olarak matematiksel anlamda *zor* bir problemdir bu.

1970'lerin başında yayımlanan bir dizi makale NP-tam problemlerin temel yapısını ortaya koyuyordu. Amerikalı-Kanadalı matematikçi Stephen Cook'un 1971 tarihli çalışması ana fikri tesis etti, ardından da Amerikalı Richard Karp'ın makalesi Cook'un NP-tam problemlerinin başta düşünüldüğünden çok daha geniş kapsamlı olduğunu gösterdi. Bu onyılda YZ araştırmacıları bir süredir üstünde çalıştıkları problemlere NP-tamlık teorisini kullanarak bakmaya başladılar. Sonuçlar şok ediciydi. Nereye baksalar –problem çözme, oyun oynama, planlama, öğrenme, akıl yürütme– başlıca problemler hep NP-tam problemlermiş (veya beteri) gibi görünüyordu. Söz konusu fenomen o kadar yaygınlaştı ki sonunda "YZ-tam problem" diye esprisi yapılır oldu ("YZ'nin kendisi gibi *zor* bir problem" mânâsında); buna göre, bir YZ-tam problemi çözerseniz, hepsini çözebilirsiniz demekti.

Üstünde çalışılan problemlerin NP-tam (veya beteri) olduğunu keşfetmek ağır bir darbeydi. NP-tamlık teorisi ve sonuçları bütünüyle anlaşılmadan önce, bir anda müthiş bir atılımla böyle problemlerin kolay ya da –teknik terimle– **çözülebilir** hale geleceğine dair bir umut vardı hep. Teknik açıdan böyle bir umut hâlâ var, çünkü NP-tam problemlerin verimli bir biçimde çözülemeyeceği kesin değil. Ancak 1970'lerin sonunda NP-tamlığın ve kombinasyon patlamasının hayaleti YZ âleminin üstüne bir kâbus gibi çökmeye başladı. Alan bilgi işlemsel karmaşıklık biçiminde bir engele takıldı ve ilerleme durma noktasına geldi. Altın Çağ'da geliştirilen tekniklerin uygulama alanı Blok Dünyası gibi oyuncaklı mikro-dünya senaryolarının ötesine geçemiyor gibiydi. 1950 ve 1960'larda kaydedilen hızlı ilerlemenin beraberinde getirdiği iyimser tahminler erken dönemin öncü araştırmacılarının başına bela olmaya başladı.

## YZ Simyası

1970'lerin başında, en temel YZ problemleri konusunda kayda değer bir ilerleme kaydedilemediği ortadayken kimi araştırmacıların hâlâ büyük büyük iddialarda bulunuyor olması, bilim camiasının geniş kesimlerini giderek daha fazla hayal kırıklığına uğratıyordu. 1970'lerin ortasında eleştiriler aldı yürüdü, ortalık iyice kızıştı.

Bütün bu eleştirmenlerin (sayıları hiç de azımsanacak gibi değildi) en lafını sakınmayanı, en alenen iğneleyici olanı kuşkusuz Hubert Dreyfus'tı. 1960'ların ortasında YZ alanında ilerlemenin durumunu raporlayacak birini arayan RAND Corporation, bu iş için Dreyfus'la anlaşmıştı. Dreyfus'ın raporu için seçtiği başlık "Simya ve YZ" idi. Bu alanı ve alanda çalışanları nasıl küçümsediğini daha başlıktan anlamışsınızdır herhalde. Bir biliminsanının ciddi çalışmalarını alenen simyayla bir tutmak olağanüstü derecede aşağılayıcıdır. "Simya ve YZ"yi yenilerde tekrar okudum da, hayatta böyle bir bilimsel rapor daha görmedim desem yeridir; o nasıl bir hor görmek öyle, üstünden yarım yüzyıldan fazla geçmesine rağmen hâlâ şok edici.

Dreyfus'ın üslubunu onaylamayabiliriz, ama bu, YZ eleştirisinin bazı bakımlardan, bilhassa öncü YZ'cilerin şişirilmiş iddiaları ve abartılı tahminleri hususunda haklı olduğu gerçeğini değiştirmez. Sözgelimi Herb Simon (ilerleyen yıllarda iktisat alanında Nobel kazanmıştır) 1958'de şunları yazmıştır:

> Sırf sizi şaşırtmak veya şok etmek gibi bir niyetle söylemiyorum ama ... bugün yeryüzünde düşünen makineler, öğrenen makineler ve yaratan makineler var. Dahası yakın bir gelecekte bunları yapma kapasiteleri öylesine hızlı gelişecek ki, insan zihni ne kadar geniş bir uygulama alanı buluyorsa, bu makineler de o kadar geniş çapta her türlü problemin üstesinden geliyor olacak.

O döneme baktığımızda böyle pek çok iddia ortaya atıldığını görüyoruz, yani Simon'dan alıntı yapmamın nedeni bunun en abes YZ savunusu olması değil sadece en bilinen örnek olması. Zaten Simon da fanatik bir YZ taraftarı değil olsa olsa iyimser bir YZ'cidir. Ancak

bugün durduğumuz yerden geçmişe dönüp baktığımızda, böyle iddia ve tahminlerin gerçekdışılığının acı bir biçimde aşikâr olduğunu görüyoruz. Bugün YZ camiasından kimileri, o dönemde araştırmacıların böyle iddialarda bulunmalarını, olup bitenin heyecanına kapılmalarına bağlıyor; meseleye biraz daha şüpheyle yaklaşan kimileriyse, bütün bu yaygaranın daha çok araştırmalara maddi kaynak sağlamak için koparıldığını düşünüyor.

Meslekten biri olarak naçizane fikrimi sorarsanız, o dönemde bir naiflik varmış gibi geliyor bana; bunun YZ'yi olduğundan daha iyi göstermeye yönelik topyekûn kasıtlı bir girişim olduğunu sanmıyorum. Araştırmacılar yaptıklarından heyecan duyuyorlardı, belki daha önemlisi *yaptıkları şeylere inançları tamdı*. Ve başkalarının da inanmasını istiyorlardı. Kısıtlı bir biçimde öğrenebilen, planlayabilen, akıl yürütebilen programlar inşa etmişlerdi ve bu kısıtları aşıp uygulama alanını genişletmenin kolay olacağına inanıyorlardı.

Bu coşkunun en azından bir ölçüde hoş görülmesi gerektiğini düşünüyorum, çünkü ele alınan problemlerin *tam da tabiatı itibarıyla* bilgisayarlar tarafından çözülmesi zor olduğunu varsaymak için hiçbir neden yoktu. O dönemlerde, çığır açıcı bir gelişmeyle zor problemlerin kolay hale gelmesi ihtimali vardı hep. Ancak NP-tamlık anlayışının yaygınlaşmasıyla, YZ camiası da bilgi işlemsel bir problemin *zor* olmasının gerçekten ne anlama geldiğini anlamaya başladı.

## Altın Çağ'ın Sonu

1972'de, Birleşik Krallık'ta bilimsel araştırmalar için fon sağlayan başlıca kuruluşlardan biri, ünlü matematikçi James Lighthill'den YZ'nin mevcut durumunu ve yakın geleceğine ilişkin beklentileri değerlendirmesini istedi. Böyle bir talebin ortaya çıkma sebebinin, geçmişte olduğu gibi bugün de dünyanın en önde gelen YZ araştırma merkezlerinden biri kabul edilen Edinburgh Üniversitesi'ndeki YZ camiasına mensup akademisyenler arasında süregiden çekişmelere dair raporlar olduğu rivayet edilir. Lighthill o dönemde Cambridge Üniversitesi'nde Lucas Matematik Profesörü olarak görev yapmak-

taydı, Birleşik Krallık'ta bir matematikçinin sahip olabileceği en prestijli akademik unvana sahipti yani (Lighthill'in ardından bu pozisyona Stephen Hawking getirilmiştir). Uygulamalı matematik pratiği bakımından çok deneyimliydi. Ancak, bugün **Lighthill Raporu**'nu okuyunca, dönemin YZ kültürüne çok yabancı olduğu, öğrendikleri karşısında şaşkına döndüğü anlaşılıyor:

> ... ehil ve saygın biliminsanları ... YZ'nin ... genel evrim sürecinde yeni bir aşama olduğunu, 1980'lerde insan ölçeğinde bir bilgi tabanında genel amaçlı zekâya ulaşmanın mümkün olduğunu, 2000'ler itibarıyla ise insan zekâsını aşan makine zekâsı temeline dayanan büyüleyici imkânların gündemde olacağını yazıyorlar.

Lighthill'in raporu anaakım YZ'yi ıskartaya çıkarıyor, YZ camiasının başlıca problemlerden biri olan kombinasyon patlamasını çözemediğinin özellikle altını çiziyordu. Rapor çok geçmeden Birleşik Krallık'ın dört bir yanındaki YZ araştırma fonlarının bıçak gibi kesilmesine yol açtı.

ABD'de ise YZ araştırmalarına fon sağlayan başlıca kuruluşlar Amerikan Savunma Bakanlığı İleri Teknoloji Araştırma Projeleri Ajansı (Defense Advanced Research Projects Agency, DARPA) ve onun da öncüsü İleri Teknoloji Araştırma Projeleri Ajansı (Advanced Research Projects Agency, ARPA) olmuştu hep. Ancak 1970'lerin başı itibarıyla YZ'nin pek çok vaadini yerine getirememiş olması nedeniyle bu kuruluşlar da umudunu kaybetmiş, araştırma fonları aynı şekilde ABD'de de kesilmişti.

1970'lerin başından 1980'lerin başına kadar olan dönem zamanla **YZ kışı** olarak anılmaya başladı. Gerçi buna *ilk* YZ kışı demek daha doğru olur belki, ileride başkaları da gelecekti çünkü. YZ araştırmacısı dendi mi aşırı iyimser ve olur olmaz vaatlerde bulunan ama bunları asla yerine getiremeyen bir tip akla geliyordu artık. Bilim dünyasında YZ'nin "alternatif tıp"ınkinden hallice bir namı vardı. Saygın bir akademik disiplin olarak geri dönüşü olmayan bir gerileme dönemine girmiş gibi görünüyordu.

# 3
# Bilgi Güçtür

YZ'NİN 1970'LERİN ORTASINDA gördüğü zararın büyüklüğünü iyi anlamamız lazım. Bu tarihten itibaren pek çok akademisyen YZ'ye sözdebilim muamelesi yapmaya başladı; YZ kışında itibarının aldığı yara daha yeni yeni iyileşiyor. Bununla beraber, ta Lighthill Raporu'nun dolaşımda olduğu ve sonuçlarının hissedildiği zamanlarda bile dikkat çeken yeni bir yaklaşım vardı ki YZ'nin itibarının yerle bir olmasına yol açan problemlerin üstesinden gelmeyi vadediyordu. 1970'lerin sonu ile 1980'lerin başında dikkatleri üstüne çeken yeni nesil akademisyenlere göre, evvelki girişimlerin sorunu, arama ve problem çözme gibi genel yaklaşımlara fazla odaklandıklarından gözlerinin başka bir şey görmez oluşuydu. Bu "zayıf" metotlarda, zekâya dayalı her türlü faaliyetin elzem bir parçası olması gereken hayati bir bileşenin, *bilginin* eksik olduğu öne sürülüyordu. Francis Bacon 17. yüzyılda "Scientia potentia est" yazmıştır, yani "Bilgi güçtür". **Bilgi tabanlı** YZ'yi savunanlar da işte Bacon'ın bu düsturunu harfiyen alıyor, insan bilgisini açık bir biçimde yakalayıp kullanmayı YZ'nin ilerlemesinin anahtarı olarak görüyorlardı.

Böylece bilgi tabanlı YZ sistemleri ortaya çıktı. **Uzman sistemler** denen bu yeni kategori insan uzman bilgisini kullanarak dar sınırları iyi tanımlanmış problemleri çözüyordu. Uzman sistemler belli görevlerde YZ sistemlerinin insanlardan daha iyi bir performans sergileyebileceğine dair kanıtlar ortaya koyuyor; belki daha önemlisi, ilk defa, YZ'nin ticari çıkar sağlayacak şekilde uygulamaya koyulabileceğini gösteriyorlardı. YZ birdenbire para kazanma fır-

satı sağlayan bir alan haline gelivermişti yani. Ayrıca bilgi tabanlı YZ tekniklerini daha geniş bir kesime öğretmek mümkündü, böylece pratik YZ yetilerini uygulamaya kararlı bir YZ mezunları nesli ortaya çıktı.

Uzman sistemlerin Genel YZ olmak gibi bir iddiası yoktu. İnşa edilmesi amaçlanan sistemler kayda değer ölçüde insan uzmanlığı gerektiren, kapsamı oldukça dar bir biçimde tanımlanmış, gayet spesifik problemleri çözebilecekti. Genelde bir insan uzmanın uzun zaman içinde edindiği ve ender bulunan türden bir uzmanlık gerektiren problemlerdi bunlar.

İzleyen on yılda, bilgi tabanlı uzman sistemler YZ araştırmalarının başlıca odak noktası haline geldi. Sanayiden YZ alanına muazzam bir sermaye akışı oldu. 1980'lerin başında YZ kışı bitmişti artık; başka, daha büyük bir YZ furyası yoldaydı.

Bu bölümde, 1970'lerin sonuyla 1980'lerin sonu arasında yaşanan uzman sistemler furyasına bakacağız. İlk önce, insan uzman bilgisinin nasıl yakalanıp bilgisayara verildiğini göreceğiz. Bu noktada size dönemin en meşhur uzman sistemlerinden biri olan MYCIN'in hikâyesini anlatacağım. Matematiksel mantığın gücünü ve kesinliğini kullanarak uzman bilgisini yakalamanın daha iyi yollarının inşa edilmeye çalışıldığını ama nihayetinde bu amaca ulaşılamadığını göreceğiz. Ardından da YZ tarihinin başarısızlığıyla ün salmış en iddialı projelerinden birinin, insan uzman bilgisini kullanarak en büyük problemi yani Genel YZ problemini çözmeye çalışmış olan Cyc'ın hikâyesini dinleyeceğiz.

## İnsan Bilgisini Kurallarla Yakalamak

YZ sistemlerinde bilgiyi kullanma fikri hiç de yeni bir fikir değildi muhtemelen. Bir önceki bölümde gördüğümüz sezgisel yöntemler, Altın Çağ'da problem çözmeyi umut vadeden istikametlere odaklamanın bir yolu olarak yaygın biçimde kullanılıyordu. Böyle sezgisel yöntemler bir problem hakkındaki bilginin cisimleşmesi olarak anlaşılabilir, ne var ki bilgiyi ancak *örtük bir biçimde* yakalayabilirler. Bilgi tabanlı YZ ise önemli ve yeni bir fikre dayanıyordu: Bir prob-

lem hakkındaki insan bilgisi *açık bir biçimde* yakalanmalı ve bir YZ sistemi dahilinde *temsil edilmeliydi.*

İleride alacağı adla **bilgi temsili** için kullanılan en yaygın yöntem **kural**lara dayanıyordu. YZ bağlamında bir kural, farklı farklı bilgileri "eğer ..., o zaman ..." kalıbı biçiminde bir öbek olarak yakalar. Kurallar bayağı basittir aslında. Bir örnekle açıklayalım. Aşağıda YZ folklorunun parçası olan bazı kurallar var.[1] Hayvanları sınıflandırma görevinde bir kullanıcıya yardımcı olan bir uzman sistem için bilgiyi yakalıyorlar (kuralların nasıl kullanıldığı konusunda daha ayrıntılı bilgi için bkz. Ek A):

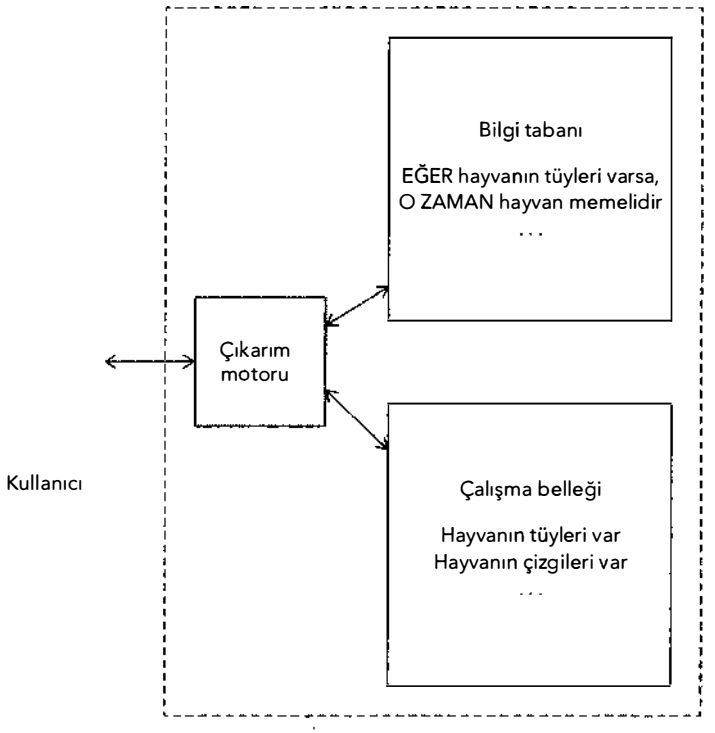
EĞER hayvan süt veriyorsa, O ZAMAN hayvan memelidir
EĞER hayvanın telekleri varsa, O ZAMAN hayvan kuştur
EĞER hayvan uçabiliyorsa VE yumurtluyorsa, O ZAMAN hayvan kuştur
EĞER hayvan et yiyorsa, O ZAMAN hayvan etoburdur

Bu kuralları şöyle yorumlarız: Her kuralın bir **şartı** ("EĞER" ile virgül arasındaki kısım) ve **sonucu** (O ZAMAN'dan sonraki kısım) vardır. Örneğin alttaki kuralda:

EĞER hayvan uçabiliyorsa VE yumurtluyorsa, O ZAMAN hayvan kuştur

Şart "hayvan uçabiliyorsa VE yumurtluyorsa"dır, sonuç ise "hayvan kuştur". Böyle bir kural genelde şöyle yorumlanır: Halihazırda sahip olduğumuz enformasyon şarta uyuyorsa, kural **tetiklenir**, yani sonuca varabiliriz. Dolayısıyla bu kuralın tetiklenmesi için iki enformasyon gerekiyor bize: sınıflandırmaya çalıştığımız hayvan uçabiliyordur ve sınıflandırmaya çalıştığımız hayvan yumurtluyordur. Bu enformasyona sahipsek kural tetiklenir ve hayvanın kuş olduğu sonucuna varırız. İlgili sonuç bize daha fazla enformasyon verir, bu enformasyon da müteakip kurallarda kullanılır, böylece daha fazla enformasyon elde edilir vs. Uzman sistemler genelde bir kullanıcı ile istişare biçiminde bir etkileşime giriyordu. Kullanıcı sisteme enformasyon sağlamak ve sistemin sorduğu soruları cevaplamakla yükümlüydü.

Şekil 5 tipik bir uzman sistemin yapısını gösteriyor. **Bilgi tabanı**




Şekil 5: Tipik bir uzman sistemin yapısı.

sistemin sahip olduğu bilgiyi (kurallar), **çalışma belleği** ise sistemin çözülmekte olan problem hakkında halihazırda sahip olduğu enformasyonu içerir (örn. "hayvanın telekleri var"). Son olarak **çıkarım motoru** da sistemin sahip olduğu bilgiyi problem çözme esnasında bilfiil uygulayan kısmıdır.

Deminki hayvan örneği gibi bir bilgi tabanı verildiğinde, bir çıkarım motoru iki farklı şekilde işleyebilir. Birincisi, kullanıcı sisteme eldeki belli bir problem hakkında bilinen her şeyi anlatır (bu örnekte, sınıflandırmaya çalıştıkları hayvanın çizgileri var mı yok mu, et yiyor mu yemiyor mu vb.), çıkarım motoru kurallarını kapsamlı bir biçimde uygulayarak mümkün olduğunca çok yeni enfor-

masyon elde etmeye çalışır, kuralları tetikler, elde edilen yeni enformasyonu çalışma belleğine ekler, bu yeni enformasyon ışığında tetiklenen herhangi bir kural var mı diye bakar, ve başka bir kural uygulama ve başka bir enformasyon elde etme imkânı kalmayıncaya kadar bu süreci tekrar eder durur. Bu yaklaşıma **ileriye doğru zincirleme** denir, veriden sonuçlara doğru çalışırız.

İkincisi ise **geriye doğru zincirleme**dir; tesis etmek istediğimiz bir sonuçtan başlayıp veriye doğru geri gideriz. Örneğin sınıflandırmaya çalıştığımız hayvanın etobur olup olmadığını tespit etme amacıyla başlayabiliriz. Son kural, hayvanın et yediğini tespit edebilirsek bu sonuca varabileceğimizi söyler bize. Öyleyse biz de hayvanın et yiyip yemediğini tespit etmeye çalışabiliriz. Bu sonuca varan herhangi bir kural olmadığından, doğrudan kullanıcıya sorabiliriz.

## MYCIN: Bir Uzman Sistem Klasiği

1970'lerde ortaya çıkan birinci nesil uzman sistemlerin ikonası MYCIN'dir herhalde[2] (birçok antibiyotik "-mycin" sonekiyle bittiğinden bu ismi almıştır). MYCIN ilk defa, YZ sistemlerinin önemli problemlerde insan uzmanlardan daha iyi performans sergileyebildiğini ortaya koymuş ve kendisinden sonra gelen onlarca sistem için bir şablon sunmuştur.

MYCIN bir doktor yardımcısı olarak düşünülmüştü, insanların kan hastalıkları konusunda uzman tavsiyesi verecekti. Sistem Stanford Üniversitesi araştırmacılarından oluşan bir ekip tarafından geliştirildi, ekibe Bruce Buchanan'ın başı çektiği YZ laboratuvarlarından uzmanlar ile Ted Shortliffe'in başı çektiği tıp fakültesinden uzmanlar da dahildi. Sistemin üstünde uzmanlaşması gereken konuları gerçekten bilen insanların projeye doğrudan katkı sağlaması, MYCIN'in böylesine başarılı olmasında önemli bir rol oynamıştı. Pek çok müteakip uzman sistem projesi, ilgili insan uzmanların olmazsa olmaz katılımını sağlamadığı için başarısızlıkla sonuçlanacaktı.

MYCIN'in kan hastalıkları hakkındaki bilgisinin temsilinde kullanılan kurallar daha önce verdiğimiz hayvan örneğinde olduğundan birazcık daha zengindir. Tipik bir MYCIN kuralı şöyledir mesela:

EĞER:
1) Organizma Gram metodu kullanılarak boyanmadıysa VE
2) Organizmanın morfolojisi çubuğu andırıyorsa VE
3) Organizma anaerobikse
O ZAMAN:
Organizmanın bakteroit olduğuna dair kanıt vardır (0,6)

Bu kural, [Türkçeleştirilmiş] MYCIN dilinde şu şekilde ifade edilir:

KURAL036:
ÖNCÜL: (*$*VE (AYNI BĞLM GRAM GRAMNEG)
(AYNI BĞLM MORF ÇUBUK)
(AYNI BĞLM SLNM ANAEROBİK))
EYLEM: (SONUÇ BĞLM KİMLİK
BAKTEROİT KESİNLİK 0,6)

MYCIN'in kan hastalıkları hakkındaki bilgisi beş seneyi aşkın bir zamanda kodlanıp işlendi. Nihai versiyonunda bilgi tabanında yüzlerce kural vardı.

MYCIN'in ikonalaşmasının nedeni uzman sistemler için olmazsa olmaz sayılan belli başlı bütün özellikleri bünyesinde toplamasıydı.

Birincisi, sistem bir insan kullanıcıya danışmaya benzer şekilde işlemek üzere tasarlanmıştı; sorular soruluyor, kullanıcı bunlara cevap veriyordu. Uzman sistemlerin standart modeli haline geldi bu ve MYCIN'in başlıca rolü yani teşhis de uzman sistemler için standart bir görev oldu.

İkincisi, MYCIN nasıl akıl yürüttüğünü *açıklayabiliyordu.* Akıl yürütmenin şeffaflığı YZ uygulamaları için çok önemli bir mesele haline geldi. Bir YZ sistemi önemli sonuçları olan bir problemi işliyorsa (MYCIN örneğinde hayat memat meselesidir bu), sistemin tavsiyelerine uyması istenenlerin bunlara güveniyor olması gerekir; bunun için de bir tavsiyeyi açıklama ve gerekçelendirme yetisi esastır. Deneyimlerimiz bize "karakutu" gibi işleyen, tavsiyede bulunmakla beraber bunları gerekçelendiremeyen sistemlerin kullanıcılar tarafından şüpheyle karşılandığını gösteriyor.

MYCIN'in bir başka önemli özelliği de neden belli bir sonuca vardığıyla ilgili soruları cevaplayabilmesiydi. Sonuca varan akıl yü-

rütme zincirini ortaya koyarak yapıyordu bunu, tetiklenen kurallarla ve bu kuralların tetiklenmesini sağlayan enformasyonla. Pratikte, uzman sistemlerin açıklama yetileri aşağı yukarı aynı kapıya çıkar. İdeal açıklamalar olmasalar da kolayca ortaya konurlar ve sistemin belli bir sonuca nasıl vardığına dair en azından fikir verirler.

Son olarak, MYCIN **belirsizlik**le, sağlanan enformasyonun doğruluğunun kesin olarak bilinmediği durumlarla baş edebiliyordu. Belirsizliği ele almanın, uzman sistem uygulamalarının ve daha genelde YZ sistemlerinin her yerinde bir gereklilik olduğu ortaya çıkmıştı. MYCIN gibi sistemlerde belli bir kanıta dayanarak kesin bir sonuca varıldığı hemen hiç görülmezdi mesela. Diyelim ki kullanıcının kan testi pozitif çıktı; bu durum sistemin dikkate alabileceği faydalı bir kanıt sağlar, ama testin hatalı olma ihtimali hep vardır ("yanlış pozitif" veya "yanlış negatif" olabilir). Veya sözgelimi hastada görülen bazı semptomlar belli bir rahatsızlığa *işaret ediyor* olabilir, ama hastanın kesinlikle bu rahatsızlıktan muzdarip olduğu sonucuna varmaya yetmez (lassa ateşinin semptomlarından biri öksürüktür, diğer taraftan sadece hastanın öksürmesine dayanarak lassa ateşinin olduğu sonucuna varamayız, yeterli bir kanıt değildir bu). Uzman sistemlerin yerinde hükümlerde bulunabilmeleri için böyle kanıtları birtakım ilkeler çerçevesinde değerlendirebilmeleri gerekir.

MYCIN'in belirsizliği ele alırken kullandığı tekniğe kesinlik faktörleri deniyor, bu sayısal değerler belli bir enformasyona ne derece inanıldığını veya inanılmadığını gösteriyordu. Kesinlik faktörleri sadece belirsizliği yakalama problemine mahsus bir çözüm olduğu için, sonradan çok eleştirildi. Belirsizliği ele alma ve belirsizlik konusunda akıl yürütme problemi YZ araştırmalarının başlıca konularından biri haline geldi. Günümüzde de hâlâ geçerliliğini koruyan bu durumu 4. Bölüm'de daha ayrıntılı ele alacağız.

1979'da yapılan denemelerde, on gerçek vaka üstünden değerlendirildiğinde, MYCIN'in kan hastalıklarını teşhis etmede gösterdiği performansın insan uzmanlarınkinden geri kalır bir yanı olmadığı, pratisyen hekimlerinkinden ise daha iyi olduğu görüldü. Belki de ilk defa bir YZ sistemi kayda değer bir görevde insan uzman seviyesinde veya üstünde bir kapasite sergiliyordu.

## Yeni Bir Furya

MYCIN 1970'lerde ortaya çıkan pek çok sansasyonel uzman sistemden sadece biriydi. Örneğin yine Stanford Üniversitesi menşeli bir DENDRAL projesi vardır. Bilgi tabanlı sistemlerin en bilinen savunucularından Ed Feigenbaum'un başını çektiği bu proje "uzman sistemlerin atası" olarak bilinir. DENDRAL, kütle spektrometrelerinin sağladığı enformasyon yoluyla, kimyagerlerin bir bileşiğin kimyasal yapısını belirlemesine yardım ediyordu. 1980'lerin ortasında DENDRAL artık her gün yüzlerce kişi tarafından kullanılan bir sistem haline gelmişti.

Digital Equipment Corporation (DEC) VAX bilgisayarlarının konfigürasyonuna yardımcı olması için R1/XCON sistemini geliştirdi. 1980'lerin ortasında, DEC 80.000'den fazla siparişin işlendiğini söylüyordu. O dönemde sistemin 5000 farklı sistem bileşeniyle ilgili olarak 3000'den fazla kuralı vardı. 1980'lerin sonunda, sistemin 17.500 kuralı vardı; sistemi geliştirenlerin ifadesine göre bu sayede şirket yaklaşık 40 milyon dolar tasarruf etti.

DENDRAL uzman sistemlerin kullanışlı olabileceğini, MYCIN sergilediği performansla insan uzmanları onların sahasında geçebileceğini, R1/XCON bu sayede iyi para kazanılabileceğini gösterdi. Bu başarı hikâyeleri çok ilgi çekti, YZ iş dünyası için hazır görünüyordu artık. Bekleneceği üzere, cezbedici bir hikâyesi olan bu alana ciddi bir sermaye akışı gerçekleşti, çok büyük yatırımlar yapıldı. Bu furyada kasasını doldurmak isteyen girişim şirketleri mantar gibi bitti. Böyle şirketlerin sunduğu tipik ürünlerden biri, uzman sistemler inşa etmek için bir yazılım platformu ve yanı sıra uzman sistemler geliştirip uygulamaya koymak için destek hizmetleriydi. Ya da bunun yerine, uzman sistem inşasında tercih edilen programlama dili olan LISP'i hızlıca çalıştırmak üzere özel olarak tasarlanmış bilgisayarlar da alabiliyordunuz. Bu LISP makineleri 1990'ların başına kadar piyasada kaldı ama PC'lerin yeterince ucuzlayıp güçlenmesiyle beraber, sadece LISP'i çalıştırabilen bir bilgisayara 70 bin dolar harcamak pek de iyi bir alışverişmiş gibi görünmemeye başladı.

Dünyanın dört bir yanından şirketler uzman sistemler dalgasını yakalamak için birbirleriyle yarışıyorlardı, üstelik sadece yazılım şirketleri de değildi bunlar. 1980'ler itibarıyla sanayi, bilgi ve uzmanlığın önemli varlıklar olduğunu, geliştirilirse kâr avantajı sağlayabileceğini anlamaya başlamış gibiydi. Uzman sistemler bu maddi olmayan varlığı maddi hale getirebilirmiş gibi görünüyordu. Bilgi tabanlı sistem konsepti aynı zamanda dönemin hâkim görüşlerinden birine, Batı ekonomilerinin sanayi-sonrası bir döneme girdiği fikrine de cuk oturuyordu. Buna göre ekonominin gelişmesini sağlayacak yeni fırsatlar öncelikle, imalat sanayisinden veya diğer geleneksel sanayilerden değil, bilgi tabanlı sanayi ve hizmetlerden gelecekti.

Geçmiş uzman sistemler deneyimine bugünden bakınca, bu furyanın ortaya çıkışında MYCIN, DENDRAL ve benzerlerinin başarı hikâyelerinden öte bir şey olduğunu görüyoruz: Uzman sistemler YZ'yi *erişilebilir* kılmıştır. Bir uzman sistem inşa etmek için illaki doktoranızın olması gerekmiyordu (biraz geride verdiğim hayvan örneğindeki kuralları gayet rahat takip etmişsinizdir diye düşünüyorum). Programlamayla haşır neşir olan herkes uzman sistemlerin ilkelerini öğrenebilirdi, hatta uzman sistemleri inşa etmek bildiğimiz programları inşa etmekten bir şekilde daha kolay görünüyordu. Böylece nur topu gibi bir iş unvanımız oldu: **bilgi mühendisi**.

İşin ironik tarafı, 1983'te Birleşik Krallık Hükümeti hevesle, sanki daha on sene kadar önce Lighthill Raporu İngiliz YZ'sinin neredeyse sonunu getirmiyormuş gibi, bilgisayar teknolojisi için bir araştırma finansmanı girişimi olan Alvey programını başlattı. Alvey'nin orta yerinde YZ vardı ama YZ daha bu kadar yeni itibar kaybına uğramışken kimse adını anmak istemiyordu sanki. Böylece YZ yerine *Zekâya-Dayalı Bilgi-Tabanlı Sistemler* demeye başladılar. YZ'ye YZ demediğiniz sürece, parlak bir gelecek onu bekliyordu.

## Mantık Tabanlı YZ

Kurallar insan bilgisini yakalama konusunda en yaygın yaklaşım haline geldiyse de, birbirinden farklı daha pek çok yöntem ortaya atıldı. Örneğin Şekil 6 psikolog Roger Schank ve Robert P. Abelson'ın ge-

İsim: RESTORAN

Roller: Müşteri, Garson, Aşçı, Kasiyer

Giriş koşulu: Müşteri açtır

Unsurlar: Yemek, masa, para, menü, bahşiş

Olaylar:
1. Müşteri restorana girer
2. Müşteri masaya oturur
3. Garson menüyü getirir
4. Müşteri sipariş verir
5. Garson yemeği getirir
6. Müşteri yemeği yer
7. Müşteri garsondan hesabı ister
8. Garson müşteriye hesabı getirir
9. Müşteri kasiyere para verir
10. Müşteri restorandan çıkar

Ana konsept: 6

Sonuçlar: Müşteri aç değil
Müşterinin parası azaldı
Restoranın parası arttı

**Şekil 6:** Alelade bir restoranda yemek yeme deneyimini betimleyen bir betik.

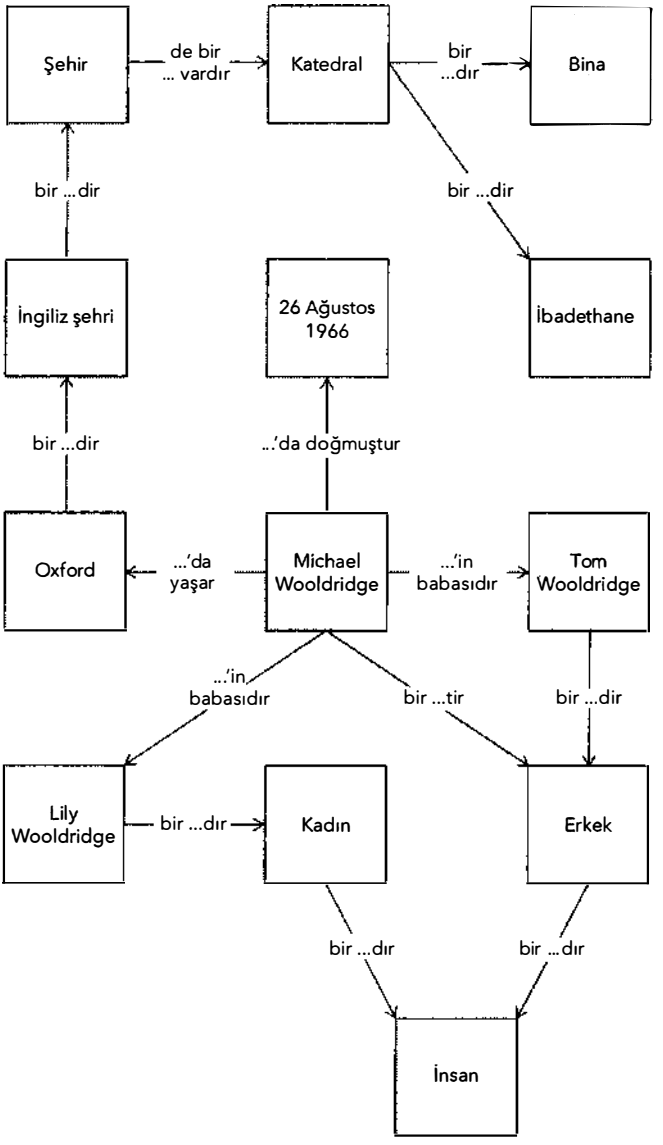
liştirdiği bilgi temsil yöntemi olan **betik**leri (basitleştirilmiş halini) gösteriyor. Schank ve Abelson'ın çalışması insan idrakine dair bir psikoloji teorisine dayanıyordu; buna göre, davranışımız kısmen klişeleşmiş örüntüler ("betikler") tarafından yönetiliyordu ve bunları aynı zamanda içinde yaşadığımız dünyayı anlamak için kullanıyorduk. Bu fikir YZ alanında da kullanılabilirdi. Şekil 6'ya tekrar bakalım mesela; burada basmakalıp bir restoran senaryosu betimlenmektedir. Betik çeşitli rollerdeki katılımcıları (müşteri, garson, aşçı, kasiyer), betiğin başlatılması için genelde gerekeni (müşteri açtır), betiğin kullanacağı fiziksel unsurları (yemek, masa, para, menü, bahşiş) ve en önemlisi, ilgili olayların genelde izlediği sırayı betimler

(1'den 10'a numaralandırılmıştır). Schank ve Abelson'a göre böyle bir betik, hikâyeleri anlamaya yönelik bir YZ programının parçası olarak kullanılabilirdi. Olaylar basmakalıp betikten *farklılaştığı* ölçüde hikâye enteresanlaşıyordu (eğlenceli, korkutucu, şaşırtıcı oluyordu). Örneğin komik bir hikâyede betik 4. adımdan sonra kesintiye uğrayabilir: Müşteri yemek söyler ama o yemek asla gelmez. Polisiye tarzında bir hikâye 9. adımı atlayabilir: Müşteri siparişini verir, gelen yemeği yer, ama hesabı ödemeden gider. Betikleri temel alarak hikâyeleri anlayabilen sistemler inşa etmeye yönelik pek çok girişimde bulunulmuş, ama bu girişimlerin başarısı sınırlı kalmıştır.[3]

Büyük ilgi gören bir diğer yöntem de **semantik ağlar**dır.[4] Bu son derece sezgisel ve doğal yöntem günümüzde tekrar tekrar icat edilmektedir. Aslına bakarsanız, bir bilgi temsil yöntemi icat etmeniz istense, muhtemelen siz de buna benzer bir şey yapardınız. Şekil 7'de basit bir semantik ağ var; benim hakkımdaki kimi bilgileri (doğum tarihim, ikamet yerim, cinsiyetim ve çocuklarım), keza dünya hakkındaki daha genel birtakım bilgileri (kadın insandır, katedral hem bina hem ibadethanedir vb.) temsil ediyor.

Bilgi tabanlı YZ ortaya çıkarken bir yerden sonra artık herkesin kendine özel, başka hiç kimseninkine uymayan ayrı bir bilgi temsil yöntemi var gibiydi.

Kurallar uzman sistemlerin fiili bilgi temsil yöntemi olarak yerleştiyse de, bilgi temsili meselesi YZ araştırmacılarının başına bela olmuştu. Bir kere kurallar basit kalıyor, karmaşık ortamlar hakkındaki bilgiyi yakalamaya yetmiyordu. MYCIN'de kullanılan türden kurallar, zamanla değişen veya birçok aktör (insan veya YZ) barındıran ortamlar ya da ortamın gerçekteki durumuyla ilgili belirsizlikler söz konusu olduğunda bilgiyi yakalamaya pek uygun değildi. Başka bir sorun da uzman sistemlerde bilgiyi yakalamak için kullanılan çeşitli yöntemlerin açıkçası biraz *keyfi* görünmesiydi. Araştırmacılar uzman sistemlerdeki bilginin gerçekte *ne anlama geldiğini* bilmek ve sistemin giriştiği akıl yürütmenin sağlamlığından emin olmak istiyorlardı. Kısacası, bilgi tabanlı sistemlere doğru dürüst bir matematiksel temel kazandırmak istiyorlardı. YZ araştırmacısı Drew McDermott'ın 1978 tarihli bir makalesi bu meseleleri iyi özetler.[5]




**Şekil 7:** Ben, çocuklarım ve yaşadığım yer hakkında birtakım enformasyon barındıran basit bir semantik ağ.

"Bir sistemin doğru olması yetmez," der McDermott, "anlaşılması da bir o kadar önemlidir."

1970'lerin sonunda, bilgi temsili için birleştirici bir yöntem olarak **mantığı** kullanma çözümü belirmeye başladı. Bilgi tabanlı sistemlerde mantığın nasıl bir rol oynadığını anlamak için, mantığın ne olduğunu ve neden geliştirildiğini biraz bilmekte fayda var. Mantık, akıl yürütmeyi ve bilhassa **sağlam muhakeme**nin **hatalı muhakeme**den farkını anlamak için geliştirilmiştir. Bu akıl yürütme tarzlarını birkaç örnekle açıklayalım:

Bütün insanlar fanidir.
Emma insandır.
Öyleyse Emma fanidir.

Burada **tasım** denen mantıksal akıl yürütme örüntüsünü görüyoruz. Örnekteki muhakemenin gayet akla yatkın olduğu konusunda bana katılırsınız herhalde: Eğer bütün insanlar faniyse ve Emma insansa, o zaman Emma fanidir.

Şimdi de şu örneğe bakalım:

Bütün profesörler güzeldir.
Michael profesördür.
Öyleyse Michael güzeldir.

Bir önceki örnekle kıyaslarsak, "Bütün insanlar fanidir" önermesini seve seve kabul etmekle birlikte, "Bütün profesörler güzeldir" önermesini –doğru olmadığı aşikârdır bunun– kabul etmeye yanaşmazsınız muhtemelen. Ancak *mantık* açısından, bu örnekteki akıl yürütmede herhangi bir terslik yoktur, gayet sağlamdır. Bütün profesörler gerçekten güzel olsaydı, Michael profesörse güzel olacağı sonucuna varmak gayet makul olurdu. Mantık, çıkış noktanızı oluşturan önermelerin (**öncül**ler) *gerçekten* doğru olup olmadığıyla ilgilenmez; öncüllerinizin doğru olması halinde, başvurduğunuz akıl yürütme örüntüsünün ve vardığınız sonuçların makul olup olmadığına bakar.

Sıradaki örneğimiz hatalı bir muhakemeyi gösteriyor:

Bütün öğrenciler çalışkandır.
Sophie öğrencidir.
Öyleyse Sophie zengindir.

Burada muhakeme örüntüsü sağlam değil, zira verili iki öncülden yola çıkarak Sophie'nin zengin olduğu sonucuna varmak pek makul görünmüyor. Sophie zengin de *olabilir* tabii, ama mesele bu değil; burada asıl üstünde durmamız gereken nokta, verili iki öncülden yola çıkarak bu sonuca varmanın makul olmaması.

Özetle, mantık esasen akıl yürütmenin örüntüsüyle ilgilidir. Son iki örnekte gördüğümüz tasım belki de en basit haliyle bu örüntüyü gösteriyor. Mantık bize ne zaman öncüllerden yola çıkarak güvenle birtakım sonuçlara varabileceğimizi anlatır. Bu sürece **tümdengelim** adı verilir.

Antik Yunan filozofu Aristoteles'in takdimi olan tasımlar bin yılı aşkın süredir mantıksal analizin ana çerçevesini oluşturmaktadır. Ancak tasımlar mantıksal akıl yürütmenin son derece kısıtlı bir biçimini yakalar, pek çok argüman biçimine hiç uymaz. Matematikçiler ezelden beridir sağlam muhakemeye büyük ilgi gösteriyor, genel ilkelerini anlamaya çalışıyorlar, zira matematik nihayetinde akıl yürütmeden başka nedir ki? Matematikçinin işi mevcut bilgiden yeni matematiksel bilgiler türetmek, başka bir deyişle tümdengelimdir. 19. yüzyıl itibarıyla, yapmakta oldukları şeyin ilkeleri konusunda matematikçilerin kafası net değildi. Bir şeyin doğru olması ne anlama geliyordu? Matematiksel bir kanıtın makul bir tümdengelim ortaya koyduğundan gerçekten emin olabilir miydik? 1+1'in 2'ye eşit olduğundan bile emin olabilir miydik ki?

On dokuzuncu yüzyılın ortasından itibaren matematikçiler bu konuları ciddi bir biçimde araştırmaya başladı. Almanya'da Gottlob Frege'nin geliştirdiği genel mantıksal kalkülüs sayesinde ilk kez modern matematiksel mantığın tam çerçevesini andıran bir şey belirmeye başlamıştı. Londra'da Augustus de Morgan ve Cork, İrlanda'da George Boole cebir problemleri için kullanılan tekniklerin nasıl mantıksal akıl yürütmeye uygulanabileceğini gösteriyordu. (Boole elde ettiği kimi bulguları 1854'te mütevazılıktan uzak, debdebeli bir başlıkla, *Laws of Thought* [Düşünmenin Yasaları] adıyla yayımladı.)

Yirminci yüzyılın başı itibarıyla modern mantığın temel çerçevesi büyük ölçüde oluştu, bugün matematik alanındaki hemen hemen bütün çalışmaların temelindeki dili sağlayan bir mantık sistemi

ortaya çıktı. Bu sistem **birinci dereceden mantık** adıyla biliniyor. Birinci dereceden mantık matematiğin ve akıl yürütmenin ortak dilidir. Tek başına bu çerçeveyle, Aristoteles ve de Morgan'ınkinden Boole ve daha nicelerininkine envai çeşit akıl yürütme tarzı yakalanabilir. Birinci dereceden mantık aynı şekilde YZ alanında da, o dönemde dağınık bir biçimde hemen her yerde görülen binbir çeşit, çoğu amaca özel bilgi temsil yöntemini birleştirebilecek bir çerçeve olarak görülmüştür.

Böylece **mantık tabanlı YZ** doğmuş oldu. İlk ve en etkili savunucularından biri John McCarthy'ydi – evet, Darmouth Yaz Okulu'nu düzenleyen, alana adını veren McCarthy. Mantık tabanlı YZ vizyonunu şöyle anlatıyordu McCarthy:[6]

> Bir ajanın, mantıksal önermelerle kendi dünyasına, amaçlarına ve mevcut duruma dair bilgiyi temsil edebileceği ve bu amaçlara ulaşmak için hangi eylem veya eylemlerin uygun olduğunu [tümdengelim yoluyla] belirleyerek ne yapacağına karar verebileceği fikrine dayanıyor.

Burada "ajan"la kastedilen YZ sistemidir, "mantıksal önermeler" ise tam da "Bütün insanlar fanidir" veya "Bütün profesörler güzeldir" türünden önermelerdir. Birinci dereceden mantık, matematiksel kesinliği sayesinde böyle önermelerin ifade edilebileceği zengin bir dil sağlar.

Şekil 8 McCarthy'nin mantık tabanlı YZ vizyonunu (biraz stilize bir biçimde) görselleştiriyor. Bir robotu Blok Dünyası gibi bir ortamda iş başındayken görüyoruz. Robotun bir kolu var, ayrıca ona içinde bulunduğu ortam hakkında enformasyon sağlayan bir algılama sistemiyle donatılmış. Robot dahilinde, ortamın mantıksal bir betimlemesinin, Üst(A, Masa), Üst(C, A) vb. önermelerle yapıldığını görüyoruz. Bu önermeler esasen birinci dereceden mantığın ifadeleridir. Şekil 8'de gördüğümüz senaryo bağlamında ne ifade etme niyetiyle kullanıldıkları gayet açık:

- Üst($x$, $y$) $x$ nesnesinin $y$ nesnesinin üstünde olduğu anlamına gelir,
- Yok($x$) $x$ nesnesinin üstünde hiçbir şey yok demektir,
- Boş($x$) de $x$'in boş olduğu anlamına gelir.

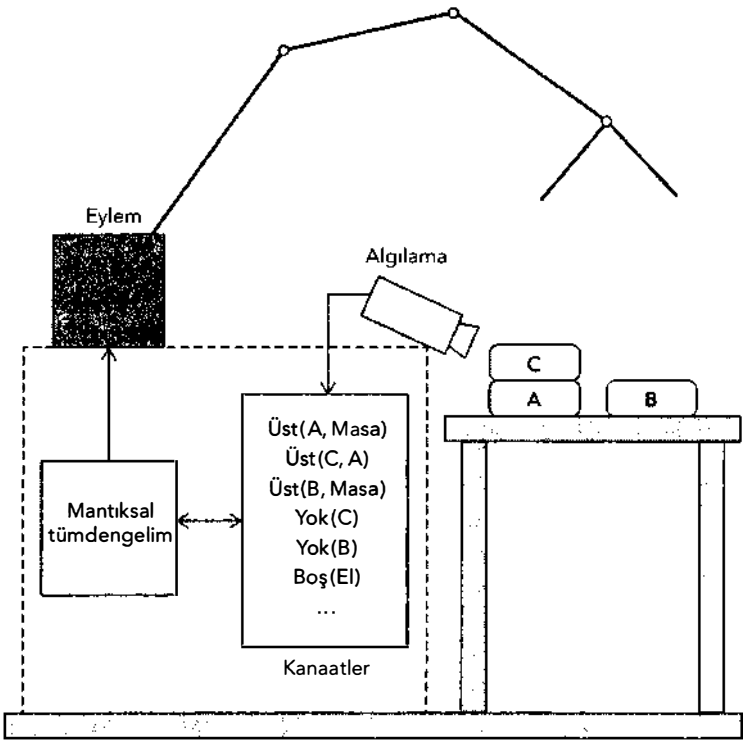
Mantık tabanlı YZ'de başlıca görevlerden biri, mevcut senaryoyu temsil edecek şekilde Üst($x$, $y$), Boş($x$) vb. ifadelerden oluşan bir dağarcık ortaya koymaktır.

Dahili mantıksal temsilin robot için rolü, ajanın içinde bulunduğu ortam hakkında sahip olduğu enformasyonun tamamını yakalamaktır. Bu temsile karşılık olarak **kanaat**ler teriminin kullanımı yaygınlaştı. Buna göre, robotun kanaatler sisteminin "Üst(A, Masa)" ifadesini içermesi, "Robot A'nın masada olduğunu düşünüyor" demektir. Gerçi böyle bir terminoloji kullanırken dikkat etmek lazım. Robotun kanaatler sisteminin içini doldururken aşina olduğumuz bir söz dağarcığından yararlanmak işimizi kolaylaştırabilir ("Üst($x$, $y$)" ifadesi insan okurlar için gayet açıktır). Fakat böyle bir ifadenin *anlamı*, robotun kanaatler tabanı dahilinde, nihayetinde *robotun davranışında oynadığı rolden* kaynaklanır. Bunun bir kanaat sayılabilmesi için robotun, bilgi tabanında bu veriye sahip olduğunda, $x$'in $y$'nin üstünde olduğunu düşünüyor gibi hareket etmesi gerekir. Ayrıca "Üst" gibi bir terimin seçilmiş olmasının manidar bir tarafı falan yoktur. Robotu tasarlayan kişi $x$ nesnesi $y$ nesnesinin üstündedir demek için pekâlâ "Qwerty($x$, $y$)" ifadesini de kullanabilirdi ve bire bir aynı sonuca ulaşırdı. YZ'de sık gördüğümüz bir sorun, robotu geliştirenlerin sistemlerinin bileşenleri için doğal olarak seçtiği terimlerin başka şeyler çağrıştırmaları, böylece kendilerine gerçekten sahip olduklarından daha fazla anlam yüklenmesine yol açmalarıdır.

Şekil 8'de algılama sisteminin görevi, robotun sensörlerinden sağlanan ham enformasyonu, robotun kullandığı dahili mantık formuna tercüme etmektir. Gayet önemli olduğundan, bu meseleye döneceğiz.

Son olarak, robotun fiili davranışı mantıksal tümdengelim sistemi (çıkarım motoru) yoluyla tanımlanır. Mantık tabanlı YZ'nin temel fikri, robotun ne yapması gerektiği konusunda mantık çerçevesinde akıl yürütmesidir.

Genel itibarıyla robotun döngüsel olarak şunları yaptığını tahayyül edebiliriz: sensörler vasıtasıyla ortamı gözlemleme, ortam hakkındaki kanaatlerini gözden geçirme, tümdengelim yoluyla bir son-




**Şekil 8:** McCarthy'nin mantık tabanlı YZ vizyonu. Mantık tabanlı bir YZ sistemi ortam hakkındaki görüşlerini mantıksal önermeler yoluyla açıkça temsil eder ve zekâya dayalı karar verme bir mantıksal akıl yürütme sürecine ayrılır.

raki eyleminin ne olması gerektiğini tespit etme, bu eylemi gerçekleştirme ve sonra tekrar başa dönüp bütün süreci yeniden başlatma.

Mantık tabanlı YZ'nin neden bu kadar etkili olduğunu iyi anlamak lazım. Bunun belki de en önemli nedeni, her şeyi gayet *saf* hale getirmesidir. Zekâya dayalı bir sistem inşa etme problemi, robotun ne yapması gerektiğinin mantıksal bir betimlemesini inşa etme problemine indirgenmiştir. Ve böyle bir sistem *şeffaftır*; bir şeyi *neden* yaptığını anlamak için, kanaatlerine ve akıl yürütme tarzına bakmamız yeter.

Mantıksal yaklaşımın böylesine etkili olmasının bu kadar bariz olmayan başka nedenleri de olduğu kanaatindeyim. Bir kere, akıl yürütmenin *bizim* verdiğimiz kararlar için temel oluşturduğunu düşünmenin cezbedici bir tarafı vardır. Ve akıl yürütmeyi *cümle cümle* yapıyormuşuz gibi görünür: Çeşitli eylem tarzlarının avantaj ve dezavantajlarına kafa yorarken zihnimizde kendimizle diyaloğa girebiliriz, kendi içimizde bu eylemlerin artılarıyla eksilerini tartışabiliriz. Mantık tabanlı YZ bu fikri yansıtıyor gibi görünür.

## Mantık Olarak Programlar

1970'lerin sonundan 1980'lerin ortasına kadar, mantık tabanlı YZ paradigmasının etkisi giderek arttı; 1980'lerin başı itibarıyla, bu paradigma anaakım YZ haline gelmişti bile. Artık öylesine etkiliydi ki araştırmacılar sadece YZ'de değil hesaplamanın daha pek çok alanında fayda sağlayabileceğine dair spekülasyonda bulunmaya başladı. Bilindiği üzere **mantık programlama** insanların program yazma tarzını kökten değiştirme iddiasıyla ortaya çıkmıştı. Mantık programlamayı pazarlamaya çalışanlar ürünleri hakkında şöyle şeyler söylüyorlardı: Programlama meşakkatlidir, zaman alır, hataya meyillidir çünkü tariflerin en ince detayına kadar düşünmeye zorlar bizi ki bununla başa çıkmak zordur. Mantık programlama bizi tam da bu dertten kurtarıyor işte. Mantık programlamayla, siz sadece bir problem hakkında bildiklerinizi ifade etmek için mantığın gücünü kullanmaya bakın, bırakın gerisini mantıksal tümdengelim halletsin. Tarif yazmaya son. Ne yapması gerektiğini tümdengelim yoluyla mantık programı kendisi bulsun.

Mantık programlama PROLOG[7] denen etkili bir dilde yapıldı. Mantık tabanlı YZ devrinin belki de en önemli mirası olan bu dil bugün hâlâ yaygın olarak öğretiliyor ve kullanılıyor. PROLOG büyük ölçüde, Birleşik Krallık'ta çalışan ABD asıllı araştırmacı Bob Kowalski ile Marsilya'da yaşayan iki Fransız araştırmacı, Alain Colmerauer ve Philippe Roussel'in çalışmaları sayesinde ortaya çıktı. 1970'lerin başında, Kowalski birinci dereceden mantıkla ifade edi-

len kuralların gerçek bir programlama dilinin temelini oluşturabileceğini fark etti. Kowalski'nin kafasında genel bir fikir vardı, detaya girmemişti. Ayrıntıları çözme işini, Kowalski'nin 1972'deki ziyaretinin ardından, Colmerauer ve Roussel halledecekti.

PROLOG çok sezgisel bir dildir. Örnek verecek olursak, daha önce gördüğümüz "Bütün insanlar fanidir" önermesi bu dilde [Türkçeleştirilmiş olarak] şöyle ifade edilir:

```
insan(emma).
fani(X) :- insan(X).
```

İlk satırdaki bir PROLOG "olgusu"dur, Emma'nın insan olduğunu ifade eder. İkincisi ise bir PROLOG kuralıdır, buna göre *X* fanidir, eğer *X* insansa.

PROLOG'a bir şey yaptıracaksak, ona bir amaç vermemiz gerekir. Diyelim ki amaç Emma'nın fani olup olmadığını tespit etmeye çalışmak. Bu şöyle ifade edilir:

```
fani(emma).
```

Bunu PROLOG'a bir amaç olarak sunarken "Emma fani midir?" diye, daha net bir ifadeyle "Sana verilen olgu ve kuralları göz önünde bulundurarak, Emma'nın fani olduğu sonucunu çıkarabilir misin?" diye soruyoruz. PROLOG bu amacı alıp, durumun gerçekten böyle olup olmadığını tespit etmek için mantıksal tümdengelime başvuruyor.[8] (PROLOG hakkında daha ayrıntılı bilgi için bkz. Ek B.)

Bu buzdağının görünen yüzü sadece, zira PROLOG'un yapabilecekleri bunu misliyle aşıyor. Kimi problemler için çok özlü, çok zarif PROLOG programları yazılabileceği anlaşılıyor. Sözgelimi David Warren'ın 1974'te yazdığı, Blok Dünyası'nı ve bunun ötesinde daha nice problemi çözebilen WARPLAN planlama sistemi için 100 satır PROLOG kodu gerekmiştir;[9] buna karşılık, aynı sistemi Python gibi bir dilde yazmaya kalksanız, muhtemelen binlerce satır tutar ve aylarınızı alırdı. Dahası PROLOG'da yazdığınız programlar sadece program değildi, aynı zamanda *mantıksal formüllerdi.* Bu nedenle PROLOG'da yazılan programlar McCarthy'nin mantık tabanlı YZ vizyonunu andırıyordu.

1970'lerin sonundan 1980'lerin başına kadar, PROLOG o kadar öne çıktı ki YZ *hacker*'larının tercihi olan programlama dili LISP'in tahtını sallamaya başladı. 1980'lerin başında, Japon hükümeti Beşinci Nesil Bilgisayar Sistemleri Projesi kapsamında PROLOG'a muazzam bir yatırım yaptı. O dönemde, ABD ve Avrupa hükümetleri Japon ekonomisinin kendi ekonomilerine kıyasla bir patlama yaşamasını endişeyle karşılıyordu. Japon hükümeti ise, teknolojiyi daha rafine hale getirmede ve ticarileştirmede gayet mahir olmasına rağmen, bilhassa hesaplama alanında kökten inovasyon bakımından o kadar da başarı sağlayamaması nedeniyle kaygılıydı. İşte Beşinci Nesil Bilgisayar Sistemleri Projesi tam da bu durumu değiştirerek Japonya'yı bilgi işlemde inovasyonun dünya çapında merkezi haline getirmeyi amaçlıyordu ve bu kapsamda yatırım için seçilen başlıca teknolojilerden biri mantık programlamaydı. Müteakip on yılda Beşinci Nesil projesine 400 milyon dolar gibi bir para akıtıldı. Proje Japonya'da bilgisayar biliminin temellerini inşa etme bakımından önemli bir rol oynamış olsa da, nihayetinde Japonya'nın bilgi işlem sanayisi atılım yaparak ABD'nin önüne geçemedi. Mantık programlama camiasından kimilerinin tahminlerinin aksine, PROLOG bir sıçrama yapıp bilgi işlemin başlıca genel dillerinden biri haline gelmedi.

Programlama dünyasını fethedemediyse de, PROLOG'u fiyasko olarak görmek büyük bir hata olur. Nihayetinde her gün dünyanın dört bir yanında programcılar, zarafetine ve gücüne minnet duydukları bu dille, seve seve üretken PROLOG programları yazıyorlar. Ama bana sorarsanız PROLOG'un daha önemli bir mirası var, daha soyut bir miras: Mantık programlama *bilgi işlem konusunda farklı bir düşünme tarzı* sağlıyor, bilgi işlem problemlerine ve bu problemlerin nasıl çözüleceğine dair bambaşka bir perspektif ortaya koyuyor ve programcılar nesillerdir bu yeni perspektiften yararlanıyor.

## Cyc: Uzman Sistemlerde Son Nokta

**Cyc projesi** bilgi tabanlı YZ döneminin muhtemelen en meşhur denemesiydi. Fikir babası Doug Lenat yetenekli bir YZ araştırmacısıy-

dı, olağanüstü bilimsel yetilerinin yanı sıra bir aydının sarsılmaz inancına ve vizyonu konusunda başkalarını ikna etme becerisine sahipti. 1970'lerde birbirinden etkileyici YZ sistemleri geliştirince iyice tanınan Lenat, 1977'de de genç bir YZ bilimcisinin kazanabileceği en prestijli ödül olan Bilgisayarlar ve Düşünce Ödülü'ne layık görüldü.

1980'lerin başında, Lenat "Bilgi güçtür" düsturunun sadece uzman sistemlerle sınırlı kalmaması gerektiği, bunun çok ötesinde uygulama alanları bulabileceği kanaatindeydi. Bunun Genel YZ'nin anahtarı olduğuna, o Büyük YZ Hayali'ne uzanan yolun buradan geçtiğine inanıyordu. 1990'da (Ramanathan Guha ile birlikte) şunları yazacaktı:[10]

> Zekâya uzanan bir kısayol, düşünce alanında henüz keşfedilmemiş Maxwell denklemleri olduğuna inanmıyoruz. ... Muazzam bilgi gereksinimini giderecek kadar güçlü bir biçimcilik yok. Bilgi derken öyle kuru kuruya, madde madde sıralanmış veya sadece belli bir alana özgü olguları kastetmiyoruz. İşin aslına bakılırsa, gerçek dünyada hayatımızı idame ettirmek için bilmemiz gerekenlerin çoğu kaynak kitaplarda yazanlar değil aklıselime dayanan şeyler. Örneğin her hayvanın belli bir ömrü vardır, hiçbir şey aynı anda iki ayrı yerde bulunamaz, hayvanlar acıyı sevmez. ... Yüzleşmesi en zor, YZ'nin 34 yıldır kaçmaya çalıştığı hakikat, bu uçsuz bucaksız bilgi tabanını elde etmenin rahat, zahmetsiz bir yolu olmadığıdır belki de. En azından başlangıçta, tek tek her savı manuel olarak girmeye emek harcamak gerekmektedir.

1980'lerin ortasında, Lenat ve çalışma arkadaşları bu "uçsuz bucaksız bilgi tabanı"nı yaratmaya giriştiler ve böylece Cyc projesi doğdu.

Akıllara durgunluk veren bir hedefti bu. Projenin Lenat'ın tahayyül ettiği şekilde yürümesi için, Cyc'ın bilgi tabanına "uzlaşımsal gerçekliğin", yani bildiğimiz anlamda dünyanın dört başı mamur bir tanımı gerekiyordu. Bunun ne kadar büyük bir zorluk olduğunu düşünün bir kere. Cyc'a şunları açık açık anlatmak gerekiyordu mesela:

> Dünya Gezegeni'nde, bırakılan bir nesne yere düşer,
> ve yere çarpınca hareket etmesi durur;

ama uzayda bırakılan bir nesne düşmez;
benzini biten bir uçak düşer;
uçak kazalarında genelde insanlar hayatını kaybeder;
türünü bilmediğiniz mantarları yemek tehlikelidir;
üstünde kırmızı işaret olan musluklardan genelde sıcak, mavilerden de soğuk su akar;
... vs.

Az çok eğitimli birinin ister istemez sahip olduğu *her türlü* gündelik bilginin Cyc'ın kendi dilinde *açık açık* yazılması ve sistemin bu bilgilerle beslenmesi gerekiyordu.

Lenat'ın hesaplarına göre projenin tamamlanması için bir kişinin yaklaşık 200 yıl emek harcaması gerekiyordu; bu zamanın çok büyük bir bölümü manuel olarak bilgilerin girilmesine gidecek, böylece Cyc'a dünyamız hakkında her şey, bizim bu dünyayı nasıl kavradığımız anlatılmış olacaktı. Lenat çok geçmeden Cyc'ın kendi kendine öğrenebilir hale geleceği konusunda iyimserdi. Kendini Lenat'ın tahayyül ettiği şekilde eğitme yetisine sahip olan bir sistem, Genel YZ probleminin etkili bir biçimde çözülmesi anlamına geliyordu. Dolayısıyla **Cyc hipotezi**ne göre Genel YZ esasen bir bilgi problemiydi ve uygun bir bilgi tabanlı sistem tarafından çözülebilirdi. Cyc hipotezi bir kumardı, bahislerin yüksek olduğu bir kumar, ama gerçekleşirse de dünyayı değiştirecek sonuçlar doğuracaktı.

Araştırma fonlamasının paradokslarından biri, kimi zaman abes derecede iddialı fikirlerin kazanmasıdır. Araştırmalara fon sağlayan bazı kuruluşlar alışılmışın dışına çıkarak, güvenli ama ağır ilerleyen projelerden ziyade, dünyayı yerinden oynatma iddiasındaki önerileri teşvik ederler. Cyc da bunlardan biriydi belki de. Her halükârda, Cyc 1984'te Austin, Teksas'taki Mikroelektronik ve Bilgisayar Konsorsiyumu'ndan (Microelectronics and Computer Consortium) alınan fonla başladı.

Ama yürümedi tabii.

Birincisi, daha önce hiç kimsenin insanın uzlaşımsal gerçekliğe dair bilgisinin tamamını düzenlemeye girişmiş olmamasından kaynaklanan bir problem vardı: Böylesi bir işe nereden *başlanır* ki? Her şeyden önce, kullanılacak söz dağarcığı konusunda anlaşmaya var-

mak ve bu dağarcığı meydana getirecek bütün terimlerin birbiriyle ilişkisini netleştirmek gerekir. Temel terim ve kavramları tanımlayıp bilgiyi bunlar etrafında düzenleme problemi debdebeli bir adla anılır: **ontolojik mühendislik**. Cyc projesi bağlamında, ontolojik mühendislik probleminin ortaya koyduğu zorluk, evvelce girişilmiş bütün projelerinkinden misliyle büyüktü. Lenat ve çalışma arkadaşları çok geçmeden, Cyc'ın kullanacağı bilginin nasıl yapılandırılacağına ve bu bilginin nasıl yakalanacağına dair başlangıçtaki fikirlerinin fazla naif ve kesinlikten oldukça uzak kaldığını gördüler. Ve bir yerden sonra, her şeye baştan başlamaları gerektiğini anladılar.

Projenin onuncu yılına yaklaşırken, Lenat bütün bu aksiliklere rağmen hâlâ iyimserdi. Nisan 1994'te, Lenat'ın bu iyimserliği Stanford Üniversitesi'nden bilgisayar bilimi profesörü Vaughan Pratt'in dikkatini çekti.[11] Pratt önde gelen hesaplama teorisyenlerinden biri olduğundan, matematiğin kesin ve titiz dilini kullanarak hesaplamanın asli doğası hakkında düşünmeye alışkındı. Cyc ilgisini çekince Lenat'tan bir sunum yapmasını rica etti, Lenat da bu ricayı kırmadı.

Pratt ziyarete gelmeden önce Lenat'a bir mektup yazıp genel hatlarıyla beklentilerini anlattı. Cyc hakkındaki yayımlanmış bilimsel makalelere dayanarak, ne türden yetiler görmeyi beklediğinden bahsediyordu. "Sunum açısından en iyisi, Cyc mevcut haliyle gerçekten nelere muktedirse o düzeyde bir beklentiyle gelmem olacak sanırım," yazıyordu. "Çok büyük beklentilerle gelirsem, muhtemelen hayal kırıklığına uğramış olarak ayrılırım. Beklentimi çok düşük tutarsam da ... kimi özellikleri üstünde haddinden fazla durup diğer özelliklerine haksızlık edebiliriz." Mektubuna yanıt alamadı.

Pratt 15 Nisan 1994'te Cyc'ı görmeye gitti. Sunum Cyc'ın mütevazı ama hakiki teknik başarılarının dosdoğru illüstrasyonlarıyla başlamıştı. Sözgelimi Cyc'ın bir ölçüde akıl yürüterek bir veritabanındaki kimi tutarsızlıkları tespit etmesi bu başarılardan biriydi. Buraya kadar her şey yolundaydı. Ardından Lenat Cyc'ın yeti yelpazesini biraz daha genişletmeye başladı. Sunumun ikinci kısmında, Cyc'ın bildiğimiz İngilizce yazılmış kimi koşullar doğrultusunda birtakım görseller getirebildiği gösterilecekti. "Kafa dinleyen biri" sorgusu, mayolu ve sörf tahtalı üç adam resmiyle sonuçlandı. Cyc

doğru bir biçimde sörf tahtası, sörf yapmak ve kafa dinlemek arasında bağlantı kurmuştu. Pratt Cyc'ın bu bağlantıya ulaşmak için nasıl bir akıl yürütme zinciri izlediğini yazarak gösterdi. Neredeyse 20 kural gerekmişti. Bunlardan bazıları size de biraz tuhaf gelebilir; mesela "Bütün memeliler omurgalıdır" da Cyc'ın bu sonuca varması için gereken kurallardan biriydi. (Pratt'in gördüğü Cyc versiyonunda yarım milyon kural vardı.)

Ardından Pratt Cyc'ın en temel özelliklerini görüp incelemek istedi: Gündelik hayata dair bilgisi nasıldı? Örneğin Pratt Cyc'a "Ekmek yenecek bir şey midir?" diye sordu. Evsahibi bu sorguyu Cyc'ın kullandığı kural diline tercüme etti. Cyc'ın cevabı "doğru" oldu. Bu kez de "Peki içilecek bir şey midir?" diye sordu Pratt. İşte o zaman Cyc işin içinden çıkamadı. Cyc'a açıkça ekmeğin içecek olmadığını anlatmaya çalıştılar. Hâlâ işe yaramıyordu. "Bir-iki kere daha denedikten sonra bu soruyu geçtik," diye anlatıyor Pratt. Devam ettiler. Cyc ölümün çeşitli nedenleri olduğunu biliyor gibi görünüyordu ama açlığın bunlardan biri olduğundan haberi yoktu sanki. Daha doğrusu açlık diye bir şeyin varlığından bihaber gibiydi. Pratt gezegenler, gökyüzü ve arabalar hakkında sorularla incelemesine devam etti. Ama çok geçmeden anlaşıldı ki Cyc'ın bunlar hakkındaki bilgisi üstünkörüydü, neyi bilip neyi bilmediğini kestirmek güçtü. Mesela gökyüzünün mavi olduğunu veya otomobillerin dört tekerleği olduğunu bilmiyordu.

Lenat'ın sunumunun uyandırdığı beklentilere dayanarak, Pratt Cyc'a sorulmak üzere yüzden fazla soru hazırlamıştı. Böylece Cyc'ın dünyamızı ne derece anladığını keşfetmeyi umuyordu. Örneğin:

Tom Dick'ten 7 cm uzunsa, Dick de Harry'den 5 cm uzunsa, Tom Harry'den kaç cm uzundur?
İki kardeşin her biri diğerinden daha uzun olabilir mi?
Şunlardan hangisi daha ıslaktır: Kara mı deniz mi?
Bir otomobil geri geri gidebilir mi?
Bir uçak geri geri uçabilir mi?
Bir insan nefes almadan ne kadar dayanabilir?

Cyc dünyamız söz konusu olduğunda aslında basit kaçan bu soruları

ne tam olarak anlayabildi ne de cevaplayabildi. Bu deneyim üstüne kafa yoran Pratt Cyc'ı eleştirirken dikkatliydi. Belki de Cyc genel zekâ yolunda bir ilerleme kaydetmiştir diye düşünüyordu. Buradaki sorun söz konusu ilerlemenin *ne kadar* olduğunun kestirilememesiydi, çünkü Cyc'ın bilgisinin dağılımı çok düzensizdi.

Cyc genel yapay zekâya uzanan yol hakkında bize ne öğretiyor? Pek de gerçekçi olmayan beklentileri bir yana bırakırsak, Cyc geniş ölçekli bilgi mühendisliğinde teknik açıdan sofistike bir alıştırma olarak karşımızda duruyor. Bize Genel YZ'yi vermemiş olabilir, ama geniş bilgi tabanlı sistemlerin gelişimi ve organizasyonu hakkında çok şey öğrettiği açık. Daha net bir ifadeyle, Genel YZ'nin esasen uygun bir bilgi tabanlı sistem yoluyla çözülebilecek bir problem olduğu yolundaki Cyc hipotezi henüz kanıtlanmış veya çürütülmüş değil. Cyc'ın bize Genel YZ'yi vermemiş olması bu hipotezin yanlış olduğunu göstermez, sadece söz konusu yaklaşımın işe yaramadığı anlamına gelir. Genel YZ'ye bilgi tabanlı yaklaşımlar yoluyla varmaya yönelik başka bir girişim *belki de* amacına ulaşacaktır.

Cyc bazı açılardan zamanının ötesinde bir girişimdi. Projenin başlamasından otuz yıl kadar sonra, Google **bilgi grafiğini** duyurdu. Google'ın arama hizmetlerini artırmak için kullandığı geniş bir bilgi tabanıydı bu. Bilgi grafiği gerçek dünyadaki varlıklar (mekânlar, kişiler, filmler, kitaplar, etkinlikler) hakkında geniş bir olgular yelpazesi sunuyor, bu olgular da Google'ın arama sonuçlarını zenginleştiriyordu. Google'a girdiğiniz sorguda genellikle dünyadaki gerçek şeylere işaret eden isimler ve daha başkaca terimler vardır. Diyelim ki "Madonna"yı google'ladınız. Alelade bir web aramasıyla Madonna sözcüğünün geçtiği web sayfalarına ulaşabilirsiniz. Ama "Madonna" karakter dizgisinin bir pop müzik şarkıcısına (Madonna Louise Ciccone) işaret ettiğini anlayan bir arama motoru daha çok işinize yarar. Yani Madonna'yı Google'da aratırsanız, bu sorgu sonucunda, şarkıcı hakkında sorulan rutin soruların (doğum yeri ve tarihi, çocukları, en popüler albümleri vb.) çoğunu yanıtlayan bir sürü enformasyona ulaşırsınız. Cyc ile Google arasındaki başlıca farklardan biri, bilgi grafiğinde bilginin manuel bir biçimde kodlanmamış, bunun yerine Wikipedia gibi web sayfalarından oto-

matik bir biçimde çıkarılmış olmasıdır – 2017'de sistemin 500 milyon varlık hakkında 70 milyar olguyu içerdiği öne sürülmüştür. Bilgi grafiği yapay zekâya talip değildir elbette. Ayrıca dünyaya dair bilgi tabanlı sistemler görüşü için asli olan bilfiil muhakemeyi ne ölçüde içerdiği açık değildir. Yine de bilgi grafiğine biraz Cyc DNA'sı kaçmış gibi geliyor bana.

Bugünden geçmişe yönelik bir değerlendirme yaparken Cyc'a haksızlık etmemeye çalışsak da, Cyc'ın YZ tarihindeki temel rolü, hakkındaki iddialı beklentileri alenen boşa çıkaran YZ furyasına ekstrem bir örnek teşkil etmesi olmuştur. Lenat'ın YZ'deki rolü efsaneleşip hesaplama folklorunun bir parçası haline gelmiş, hatta şakalara konu olmuştur: Bir şeyin ne kadar düzmece olduğunu ölçme birimine "mikro-Lenat" denir. Neden *mikro*? Hiçbir şey tam bir Lenat kadar düzmece olamaz da ondan.

## Görünen Çatlaklar

YZ'nin teoride gayet güzel olan pek çok versiyonu gibi bilgi tabanlı YZ de pratikte sınırlıydı. Yaklaşım, bize çocuk oyuncağı gibi gelen kimi akıl yürütme görevlerinde bile bocalayabiliyordu. **Sağduyuya dayalı akıl yürütme** görevini düşünün mesela, bunun klasik örneklerinden biri şöyledir:[12]

> Tweety'nin kuş olduğu söylenir, buna dayanarak Tweety'nin uçabileceği sonucuna varırsınız. Ama daha sonra öğrenirsiniz ki Tweety penguenmiş. Dolayısıyla daha önce vardığınız sonucu geri alırsınız.

İlk bakışta, bu problem mantıkla rahatça yakalanabilir gibidir. Daha önce bahsettiğimiz tasımlar iş görecek gibi görünür. YZ sistemimiz "eğer *x* kuşsa, o zaman uçabilir" bilgisine sahip olduğu takdirde, "*x* kuştur" enformasyonu sağlandığında, "*x* uçabilir" sonucuna varabiliyor olmalıdır. Buraya kadar her şey yolunda. Sıkıntı, Tweety'nin penguen olduğu söylendiğinde başlıyor. Bu noktada, daha önce vardığımız Tweety'nin uçabileceği sonucunu geri almamız gerekiyor. Ancak mantık bununla başa çıkamaz. Mantıkta, eklenen yeni enformasyon asla evvelce varılmış bir sonucu ortadan kaldır-

maz. Halbuki verdiğimiz örnekte olan tam da budur: Ek enformasyon ("Tweety penguendir"), daha önce vardığımız sonucu ("Tweety uçabilir") elememize yol açar.

Sağduyuya dayalı akıl yürütmedeki bir diğer sorun, çelişkilerle karşı karşıya kaldığımızda gündeme gelir. Literatürden standart bir örnek verecek olursak:[13]

Quakerlar pasifisttir;
Cumhuriyetçiler pasifist değildir;
Nixon Cumhuriyetçidir ve Quakerdır.

Bu senaryoyu mantıkla yakalamaya çalışmanın sonucu tam bir facia olur, çünkü Nixon'ın hem pasifist olduğu hem de pasifist olmadığı sonucuna varırız ki bu da bir çelişkidir. Böyle çelişkiler karşısında mantığın eli kolu bağlıdır: Bunlarla baş edemez, bize işe yarar bir şey söyleyemez. Ne var ki hemen her gün böyle çelişkili durumlarla karşılaşırız. Vergilerin bizim için iyi olduğu da söylenir kötü olduğu da, şarap bize yarar da dokunur da vb. Hayatın hemen her alanında böyledir. Yani bir kere daha aynı sorunla karşılaşıyoruz: Mantığı uygun olmadığı bir şey için kullanmaya çalışıyoruz. Matematikte, eğer bir çelişkiyle karşılaşıyorsanız, bir yerlerde bir hata yapıyorsunuz demektir.

Bize lafı bile edilmeyecek kadar basit gelen bu problemler 1980' lerin sonunda bilgi tabanlı YZ araştırmalarında karşılaşılan başlıca zorluklardandı. Alanın en parlak dimağları bu problemler üstünde çalışıyordu. Ama söz konusu problemleri asla çözemeyeceklerdi.

Ayrıca başarılı uzman sistemler inşa edip uygulamaya koymanın ilk başta düşünüldüğü kadar kolay olmadığı ortaya çıktı. Bu konuda yaşanan başlıca sıkıntı **bilgi çıkarma** problemiydi. Kabaca açıklamaya çalışacak olursam, insan uzmanlardan bilgi temin etme ve bu bilgiyi kurallar biçiminde kodlama problemidir bu. Sahip olduğu uzman bilgisini dile dökmek insan için genelde pek kolay değildir. İnsanın belli bir şeyi iyi yapması, onu nasıl yaptığını da gayet iyi bir biçimde anlatabileceği anlamına gelmez. Ayrıca anlaşıldı ki insanlar sahip oldukları uzmanlık bilgisini paylaşma meraklısı

değildi. Çalıştığınız şirket sizin yaptığınızı yapabilen bir programa sahip olsa, nihayetinde size ihtiyacı kalmazdı ne de olsa. YZ'nin insanın yerini almasıyla ilgili endişeler yeni değil yani.

1980'lerin sonuna doğru, uzman sistemler furyası da söndü gitti. Bilgi tabanlı sistemler teknolojisinin başarısızlıkla sonuçlandığı söylenemezdi, çünkü bu dönemde inşa edilen pek çok uzman sistem başarıyı yakaladı, hatta o zamandan bu zamana da daha nice başarılı uzman sistem inşa edildi. Ama bir kere daha, gerçekte olanlar, YZ furyasının yarattığı beklentileri karşılamaktan uzak görünüyordu.

# 4

# Robotlar ve Rasyonalite

Nedenini düşünmeyen, yaparken helak olur.
Yapmayan, bu nedenle helak olur.

W. H. AUDEN

FELSEFECİ THOMAS KUHN 1962 tarihli *Bilimsel Devrimlerin Yapısı*'nda, bilimsel yaklaşım geliştikçe, müesses bilimsel ortodoksluğun aşikâr başarısızlıkların basıncına dayanamayacağını öne sürer. Kuhn'a göre kriz anlarında yeni bir ortodoksluk ortaya çıkar ve müesses nizamın yerini alır: Bilimde bir paradigma değişimi yaşanır. 1980'lerin sonunda, uzman sistemler furyası sona erdiğinde, yeni bir YZ krizinin eli kulağındaydı. YZ camiası bir kere daha fazla yüksekten uçmakla, yerine getiremeyeceği vaatlerde bulunmakla, fiilen çok az şey ortaya koymakla eleştiriliyordu. Bu sefer sorgulanan, uzman sistemler furyasını körükleyen "Bilgi güçtür" düsturu değildi yalnızca, 1950'lerden beri YZ'nin temelini oluşturan varsayımlar, bilhassa da simgesel YZ'ydi. 1980'lerin sonunda YZ'ye yönelik en acımasız eleştiriler dışarıdan değil bizzat alanın içinden olan kişilerden geliyordu.

Hâkim YZ paradigmasının en yetkin ve etkili ismi, Avustralya doğumlu robotikçi Rodney Brooks'tu. 1954'te dünyaya gelmiş olan Brooks, YZ'ye karşı ses yükseltmesini bekleyeceğiniz türden biri değildi. YZ araştırmalarının yürütüldüğü en önde gelen merkezlerin üçünde, Stanford Üniversitesi, MIT ve Carnegie Mellon Üniversitesi'nde araştırmacı ve öğretim görevlisi olarak çalışmıştı. Brooks esasen gerçek dünyada faydalı işler yapan robotlar inşa etmekle ilgile-

niyordu. 1980'lerin başında, böyle robotlar inşa etmenin yolunun, dünya hakkındaki bilgileri robotun akıl yürütme ve karar verme süreçlerine temel oluşturacak bir biçimde kodlamaktan geçtiği fikri hâkimdi. Gelgelelim Brooks bundan giderek daha fazla şüphe duyar olmuştu. 1980'lerin ortasında MIT kadrosuna katıldı ve YZ'yi en temelden başlayarak yeniden düşünmeye girişti.

## Brooksçu Devrim

Brooks'un argümanlarını anlayabilmek için Blok Dünyası'na dönmekte fayda var. Hatırlayacak olursak, Blok Dünyası bir masanın üstünde duran çeşitli nesnelerden meydana gelen bir ortam simülasyonudur. Burada yerine getirilmesi beklenen görev, nesnelerin belli biçimlerde yeniden düzenlenmesidir. İlk bakışta, Blok Dünyası YZ tekniklerini ispatlamak için son derece makul bir zemin gibi görünür: Bir depo ortamını andırır (yeri gelmişken, projeye finansal kaynak sağlamak için yapılan sunumlarda bu noktanın özellikle altının çizildiğini düşünüyorum). Ancak Brooks'a ve onun fikirlerini benimsemiş olanlara göre, Blok Dünyası yanıltıcıydı. Böyle düşünmelerinin gayet basit bir nedeni var: Blok Dünyası bir *simülasyon* olduğu ölçüde gerçek dünyada blokları düzenleme görevinin zor taraflarının üstünü örter. Blok Dünyası'ndaki problemleri çözebilen bir sistemin, ne kadar akıllı görünürse görünsün, gerçek depoda hiçbir kıymeti yoktur, çünkü fiziksel dünyada *gerçek* zorluk algı gibi meselelerle uğraşmaktan kaynaklanır. Diğer taraftan Blok Dünyası'nda bunlar tümüyle gözardı edilir. Blok Dünyası 1970'ler ve 1980'lerin YZ ortodoksluğunun ne kadar yanlış, düşünsel anlamda iflas etmiş tarafı varsa hepsinin simgesi addedildi. Ancak bu durum Blok Dünyası ile ilgili araştırmaları durdurmadı. Blok Dünyası'nı kullanan makaleleri bugün hâlâ ara ara görebilirsiniz. Açıkçası benim de böyle birkaç makale yazmışlığım var.

Brooks'un benimsediği pozisyon üç temel ilkeye dayanıyordu. Birincisi, YZ'de kayda değer bir ilerlemenin ancak gerçek dünyaya **yerleştirilmiş** sistemlerle mümkün olduğu kanaatindeydi, yani doğrudan içinde bulunduğu ortamı algılayabilen ve eylemleriyle bu or-

tamı etkileyen sistemlerle. İkincisi, zekâya dayalı davranışın, genelde bilgi tabanlı YZ'nin ve özelde mantık tabanlı YZ'nin teşvik ettiği türden açık ve net bilgi ve muhakeme olmaksızın da ortaya konabileceğini öne sürüyordu. Ve son olarak, zekânın bir varlığın çevresiyle etkileşimi sayesinde **sonradan ortaya çıkan bir özellik** olduğunu savunuyordu.

Brooks 1991'de aktardığı şu meselde YZ'cilerin durumunu hicvederken meramını gayet özlü anlatıyor:[1]

> Diyelim ki 1890'lardayız. Yapay Uçuş (YU) bilim, mühendislik ve risk sermayesi çevrelerinde göz kamaştıran bir konu. Mucize bu ya, bir grup YU araştırmacısı bir zaman makinesiyle 1980'lere geliyor. Gelecekteki birkaç saatlik bu zamanın tamamını orta süreli bir uçuşta, ticari bir Boeing 747 uçağının yolcu kabininde geçiriyorlar. Kendi zamanlarına döndüklerinde yeniden doğmuş gibiler, çünkü YU'nun geniş ölçekte mümkün olduğundan eminler artık. Hemen kolları sıvayıp gördüklerinin bir kopyasını yapmaya girişiyorlar. Yatan koltukların, çift cam pencerelerin bire bir aynısını yapmaya bayağı yaklaşıyorlar. Ah bir de o acayip "plastiklerin" ne olduğunu çözebilseler, işte o zaman amaçlarına ulaşacaklarını biliyorlar.

*İnsan* zekâsı üstüne kafa yorarken genelde daha göz alıcı, daha somut veçhelerine odaklanırız; sözgelimi akıl yürütmeye, problem çözmeye veya satranç oynamaya, yani *düşünsel* faaliyetlere. Akademisyenler genelde böyle şeylere kıymet verir, böyle konularda iyi olmak ister. YZ araştırmalarının daha ziyade böyle faaliyetlere meyletmesinin nedeni bu önyargı olabilir. Akıl yürütme ve problem çözme zekâya dayalı davranışta bir rol oynuyordu belki ama, Brooks'a ve daha nicelerine göre, YZ'yi inşa etmek için doğru *başlangıç noktası* değildi.[2]

YU meseline dönecek olursak, Brooks'un ilk zamanlardan beri YZ'nin dayanaklarından biri olan "böl ve yönet" varsayımına da itirazı vardı: YZ araştırmalarında ancak zekâya dayalı davranış kurucu bileşenlerine (muhakeme, öğrenme, algılama) ayrıldığı takdirde ilerleme sağlanacağı düşünülüyordu ama bu bileşenlerin bir arada nasıl işlediğini değerlendirmeye yönelik bir girişim yoktu.

[YU araştırmacıları] projenin bir bütün olarak ele alınamayacak kadar büyük olduğu, aralarında işbölümü yapıp farklı alanlarda uzmanlaşmaları gerektiği konusunda hemfikirdirler. Keza uçaktaki diğer yolculara sordukları sorularla Boeing firmasının böyle bir uçak inşa etmek için altı bini aşkın çalışan istihdam ettiğini öğrenmişlerdir. ... Herkes belli bir işle meşguldür ama farklı gruplar arasında pek bir iletişim yoktur.

Son olarak, Brooks "ağırlığı" gözardı etmedeki saflığa dikkat çeker. Meseline şöyle devam eder:

Yolcu koltuklarını yapanlar iskelet için mevcut en iyi masif çeliği kullanmıştır. Aradan "Çelik boru kullanırsak ağırlıktan biraz daha tasarruf edebiliriz belki" sesleri yükseldiyse de nihayetinde "Bu kadar büyük ve ağır bir uçak uçabiliyorsa, ağırlıkla ilgili bir sorun olamaz" görüşü ağır basmıştır.

Brooks'un burada "ağırlık" dediği şey *bilgi işlem gücünü* anıştırmaktadır. Bilhassa, bütün karar verme süreçlerinin, mantıksal akıl yürütme gibi çok fazla bilgisayar işlem süresi ve belleğine ihtiyaç duyan süreçlere indirgenmesi gerektiği fikrine itiraz etmektedir.

Brooks'un meseline koyduğu başlık "Temsilsiz Zekâ"ydı. 1980'lerin uzman sistemler furyasının ortasında YZ okuyan bir lisans öğrencisiyken, bilgi temsilinin ve akıl yürütmenin YZ açısından merkezi olduğu öğretilmişti bana. Brooks'un makalesi ise alanım hakkında bildiğimi sandığım her şeyi reddediyor gibi görünüyordu. Açıkçası, o zamanlar sapkınlık gibi gelmişti bu bana. 1991'de Avustralya'da düzenlenen büyük bir YZ konferansından dönen genç bir meslektaşım, gözleri heyecandan fal taşı gibi açık, Stanfordlı (McCarthy burada çalışıyordu) doktora öğrencileri ile MIT'dekilerin (Brooks da bu üniversitedeydi) ağız dalaşına girdiğini anlatmıştı. Bir tarafta yerleşik bir gelenek vardı: mantık, bilgi temsili ve akıl yürütme. Karşısında ise yeni bir YZ hareketinin lafını sakınmayan, saygısız savunucuları. Kutsal geleneğe sırt çevirmekle kalmıyor, onunla üst perdeden dalga geçiyorlardı.

Brooks bu yeni hareketi savunan pek çok araştırmacıdan sadece biriydi. Daha nicesi benzer sonuçlara varmıştı; detaylar konusunda muhakkak hemfikir olmasalar da, birbirinden farklı yaklaşımlarında tekrar tekrar gündeme gelen belli başlı temalar vardı. Bunların en

önemlisi, eskiden YZ'nin kalbi olan bilgi ve muhakemenin tahtından indirildiği fikriydi. McCarthy'nin merkezde simgesel, mantıksal bir ortam modeli ve bunun etrafında dönen zekâya dayalı faaliyetlerden meydana gelen YZ sistemi vizyonu (Şekil 8) kesinkes reddediliyordu. Kimi ılımlı sesler muhakeme ve temsilin hâlâ oynayacağı bir rol olduğunu, gerçi bunun muhtemelen *başrol* olmadığını ileri sürerken, daha ekstrem sesler ikisini de tümüyle reddediyordu.

Bu noktayı biraz daha detaylı ele almakta fayda var. Hatırlayacak olursak, McCarthy'nin mantık tabanlı YZ görüşü, bir YZ sisteminin sürekli olarak belli bir döngüyü izleyeceğini varsayıyordu: İçinde bulunduğu ortamı algılayacak, akıl yürüterek ne yapacağını bulacak ve eyleme geçecekti. Ancak bu şekilde işleyen bir sistemde, sistem ortamdan *ayrılmıştır*.

Şimdi elinizdeki kitabı birkaç saniyeliğine bırakıp etrafınıza bir bakın. Bir havaalanının giden yolcu salonunda, kafede, trende, evinizde veya nehir kıyısında güneşe karşı uzanmış olabilirsiniz. Etrafa bakarken, ortamdan ve bu ortamın geçirdiği değişimlerden *kopuk* değilsinizdir. O *andasınızdır*. Algınız –ve eylemleriniz– ortama yerleşiktir, ortamla uyumludur. Elbette ara ara durup düşünürsünüz, elbette bu ortamdan koparsınız, ama böyle durumlar norm değil istisnadır.

Sorun, bilgi tabanlı yaklaşımın bunu yansıtıyor gibi görünmemesidir. Diyelim ki size McCarthy'nin mantık tabanlı YZ modeline göre tasarlanmış bir robot sunuyorum. Robot sürekli bir *algılama-akıl yürütme-eyleme geçme* döngüsü çerçevesinde faaliyet gösteriyor; sensörleriyle edindiği veriyi işleyip yorumluyor, algılamadan gelen bu enformasyonu kullanarak kanaatlerini tekrar gözden geçiriyor, ne yapacağı konusunda akıl yürütüyor, karar verdiği eylemi uygulamaya geçiyor ve başa dönüp bu döngüyü yeniden başlatıyor. Size gururla, tasarladığım robotun, yerine getirmesi beklenen görev hangisi olursa olsun, her zaman gerçekleştirmek üzere *en iyi* eylemi seçtiğini söylüyorum (bkz. Şekil 9).

$t_0$'dan $t_1$'e, robot ortamı duyumsar; $t_1$'den $t_2$'ye, sensör verilerini işleyip kanaatlerini buna göre tekrar gözden geçirir. $t_2$ ile $t_3$ arasında, akıl yürüterek hangi eylemi gerçekleştireceğine karar verir. Ve son


Şekil 9: Robot eyleme geçmek için en iyi zamanı seçer. Peki ama bu eylem ne zaman gerçekleştirilecek en iyi eylemdir? İçinde bulunduğu ortamı gözlemlediği zaman mı yoksa sonunda hangi eylemi gerçekleştireceğine karar verdiği zaman mı?

olarak $t_3$'te, kararını uygulamaya geçer.

Şimdi, robotun ne yapacağı konusunda her daim *en iyi* kararı verdiği iddia ediliyordu. Peki ama bu eylem $t_1$'de mi en iyi karardır yoksa $t_3$'te mi? Robot ortam hakkındaki enformasyonu daha $t_1$'de edinmiştir ama ancak $t_3$'te bu enformasyona göre harekete geçer. Böylesi bir yaklaşım pratikte pek gerçekçi değildir, yine de –belki şaşırtıcı bir biçimde– 1980'lerin sonuna kadar YZ araştırmalarının hemen hepsinde örtük bir biçimde bulunuyordu: YZ *pratikte* en iyi kararı değil *teoride* en iyi kararı (ne yapacağını belirlemeye çalışırken dünyanın hiç değişmediğini varsayarak) verebilen makineler inşa etmeye odaklanmıştı.[3]

Tam da bu nedenlerle, dönemin başlıca temalarından bir diğeri de sistemin kendini içinde bulduğu *durum* ile sergilediği *davranış* arasında doğrudan bir ilişki olması gerektiği fikriydi.

## Davranışsal YZ

YZ konusunda tipik eleştiriler öne sürseydi, Brooks'un fikirleri böylesine ilgi görmezdi. Ne de olsa 1980'lerin ortasında YZ camiası eleştirilere kulak tıkayıp yoluna devam etmeyi alışkanlık haline getirmişti. Brooks'un eleştirilerini Dreyfus tarzı bir saldırıdan ayıran, ikna edici bir biçimde alternatif bir paradigma ortaya koyması ve müteakip onyıllarda bunu etkileyici sistemlerle göstermesiydi.

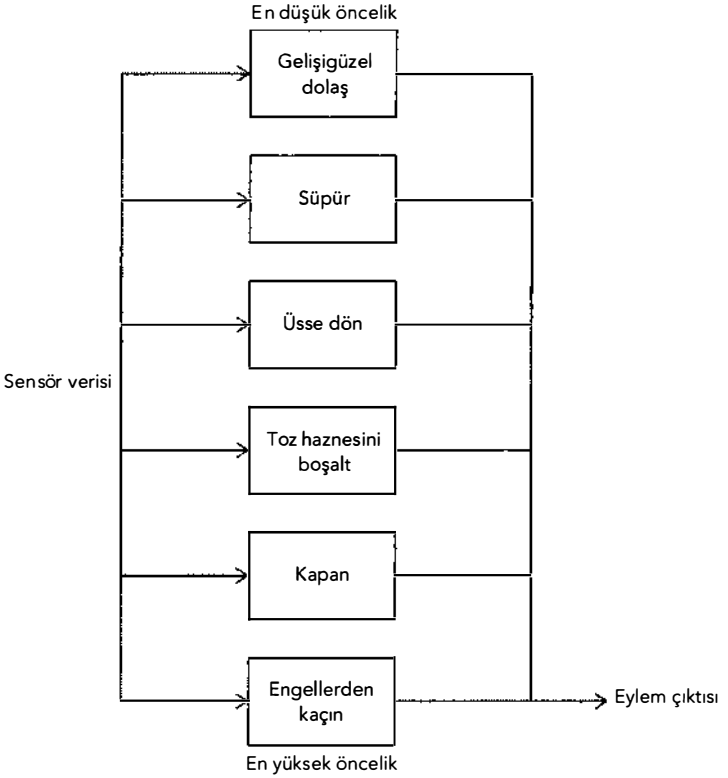
Bu yeni YZ paradigmasının **davranışsal YZ** olarak adlandırılmasının nedeni, birazdan göreceğimiz üzere, tek tek davranışların zekâya dayalı bir sistemin genel işleyişine nasıl katkıda bulunduğu üstünde durmasıydı. Brooks'un benimsediği yaklaşıma **kapsama mimarisi** deniyordu. O dönem ortaya atılan bütün yaklaşımlar içinde en çok bunun etkisi kalıcı olmuş gibi görünüyor. Dolayısıyla gelin kapsama mimarisinin mütevazı ama kullanışlı bir robotik uygulamasına biraz daha yakından bakalım: robot süpürge. (Brooks'un kurucularından biri olduğu iRobot şirketi Roomba diye bir robot süpürge üretmişti, o dönem çok tutan bu süpürgenin tasarımı Brooks'un araştırmalarına dayanıyordu.)[4] Robottan beklentimiz bir mekânı karış karış gezmesi, engellerden kaçınması ve kir veya toz tespit ettiğinde süpürmesi olsun; ayrıca şarjı azaldığında veya toz haznesi dolduğunda üssüne geri dönüp kapansın.

Kapsama mimarisinin metodolojisi öncelikle robotun yerine getirmesi beklenen davranışların tek tek tespit edilmesini gerektirir. Ardından robotumuzu inşa etmeye başlar, bu davranışlardan birini sergileyebilir hale getiririz; sonra tedricen diğer davranışları da ekleriz. Burada başlıca zorluk, davranışların birbiriyle nasıl ilişkili olduğunu düşünmek ve bunları, robotun doğru zamanda doğru davranışı sergilemesini sağlayacak şekilde düzenlemektir. Doğru düzenlemeyi bulmak için robotla kapsamlı denemeler yaparsınız.

Bizim robot süpürgemizin şu altı davranışı sergilemesi lazım:

- **Engellerden kaçın:** Bir engel tespit edersen, istikametini değiştir, rasgele bir yön seçip o istikamette ilerlemeye başla.
- **Kapan:** Üsteysen ve şarjın azsa, kendini kapat.
- **Toz haznesini boşalt:** Üsteysen ve haznede toz varsa, toz haznesini boşalt.
- **Üsse dön:** Şarj azaldıysa veya toz haznesi dolduysa, üsse geri dön.
- **Süpür:** Mevcut konumda toz olduğunu tespit edersen, süpürmeye başla.
- **Gelişigüzel dolaş:** Rasgele bir istikamet belirleyip o tarafa doğru ilerle.

Bir sonraki zorluk davranışların düzenlenmesidir. Brooks bunu **kapsama hiyerarşisi** dediği şey yoluyla yapar (bkz. Şekil 10). Kapsama




**Şekil 10:** Bir robot süpürge için basit bir kapsama hiyerarşisi.

hiyerarşisi hangi davranışın önce geldiğini belirler. Şekil 10'a göre daha altlarda yer alanlar, daha üstlerdekilere göre önceliklidir. Sözgelimi engellerden kaçınma, önceliği en yüksek olan davranıştır: Robot, engelle karşı karşıya geldiği takdirde, *daima* kaçınır. Bu eylemlerle ve Şekil 10'da görülen hiyerarşi biçiminde düzenlenmeleriyle problemin nasıl çözüldüğünü anlamak zor değil: Şarjı çok ve haznesi boş olan robot toz arayacak, bulunca süpürecek; şarjı azalınca veya haznesi dolunca da üsse geri dönecek.

Robot süpürgemizdeki davranışlar kural gibi görünebilir ama aslında çok daha basittir. Böyle davranışları bir robot bünyesinde uy-

gulamaya geçirmek, mantıksal akıl yürütme çerçevesinde gerçekleştirmekten bin kere daha az uğraştırıcı bir iştir. Davranışlar doğrudan basit elektrik devreleri biçiminde uygulanabilir, böylece robot değişen sensör verilerine çok çabuk tepki verebilir ve ortamla uyum içinde olur.

Müteakip onyıllarda, Brooks kapsama mimarisi çerçevesinde birbirinden etkili birçok robot inşa etti. Örneğin Brooks'un bugün Smithsonian Hava ve Uzay Müzesi'nde bulunan altı ayaklı robotu Genghis bir kapsama mimarisi çerçevesinde düzenlenmiş 57 davranışla kontrol ediliyordu.[5] Genghis'i bilgi tabanlı YZ tekniklerini kullanarak inşa etmek aşırı zor olurdu, hatta belki mümkün bile olmazdı. Böyle gelişmelerle, onyıllar boyunca tali yollarda ilerleyen robotik anayola geri döndü.

## Ajan Tabanlı YZ

1990'ların başında, davranışsal YZ devriminin başkahramanlarından biriyle tanışım. O zamanlar benim idolümdü. Bilgi temsili ve akıl yürütme, problem çözme ve planlama gibi YZ teknolojilerini reddettiğini söylüyordu. Gerçekten böyle mi düşünüyordu? "Tabii ki hayır," diye cevap verdi, "ama statükoyu onaylayan biri olarak tanınmak istemem." O zaman içime oturmuştu bu cevap, ama şimdi geriye dönüp bakınca, sadece naif bir yüksek lisans öğrencisini etkilemeye çalışıyordu belki de. Kendisi yaptıklarına sahiden inansa da inanmasa da, başkalarının onun yaptıklarına inandığı açıktı. Ve davranışsal YZ, tıpkı kendinden önceki gelenekler gibi, dogmalaşmaktan kurtulamamıştı. Çok geçmeden anlaşıldı ki, YZ'nin dayandığı varsayımlar hakkında önemli soruları gündeme getirmekle beraber, davranışsal YZ'nin de ciddi kısıtları vardı.

Sorun, *daha kapsamlı hale getirilememesiydi*. Tek istediğiniz evi süpürecek bir robot inşa etmekse, davranışsal YZ yeter de artar bile. Bir robot süpürgenin akıl yürütmesi, İngilizce konuşabilmesi veya karmaşık problemleri çözmesi gerekmez. Bunları yapması gerekmediği için de otonom robot süpürgeler inşa etmek için kapsama mimarisini (veya o dönem kabul gören bu tarz nice yaklaşımdan bir

diğerini) kullanabilir ve böylece gayet randımanlı bir çözüme ulaşabiliriz. Davranışsal YZ kimi problemlerde –esasen robotik alanında– son derece başarılı olduysa da YZ kapısını açan sihirli anahtar değildi. Sadece birkaç davranıştan fazlasını içeren davranışsal sistemler tasarlamak güçtür çünkü davranışların olası etkileşimlerini anlamak kolay iş değildir. Davranışsal yaklaşımlardan yararlanarak sistemler inşa etmek kara büyü yapmak gibi bir şeydi; sistemin çalışıp çalışmayacağını öğrenmenin tek yolu deneyip görmekti ve bu da zaman alıyordu, pahalıya mal oluyordu, sonunun nereye varacağı belli olmuyordu. Davranışsal bir yaklaşımla geliştirilen çözümler, etkinlik bakımından harikulade olmakla beraber, genelde son derece spesifik bir problemin en ince ayrıntılarına uygun olarak şekillendirilmiş çözümlerdi. Yani bir problemi çözerken öğrendiklerinizi kolay kolay başka bir problemin çözümünde kullanamıyordunuz.

Brooks bilgi temsili ve akıl yürütme gibi yetilerin zekâya dayalı davranışı inşa etmeye uygun temeller olmadığına dikkat çekmekte haklıydı. Onun robotları salt davranışsal yaklaşımlarla neler yapılabileceğini gösteriyordu. Ancak akıl yürütme ve bilgi temsilinin süreçte oynayacağı bir rol olmadığı fikrini bir dogma gibi savunmak bence yanlış bir adımdı. Akıl yürütmeyi (ister katı bir mantık çerçevesinde ister başka türlü) gerektiren durumlar *vardır*; bunu reddetmeye çalışmak, ne yapacağına mantıksal tümdengelim yoluyla karar veren bir süpürge inşa etmeye çalışmak kadar anlamsızdır.

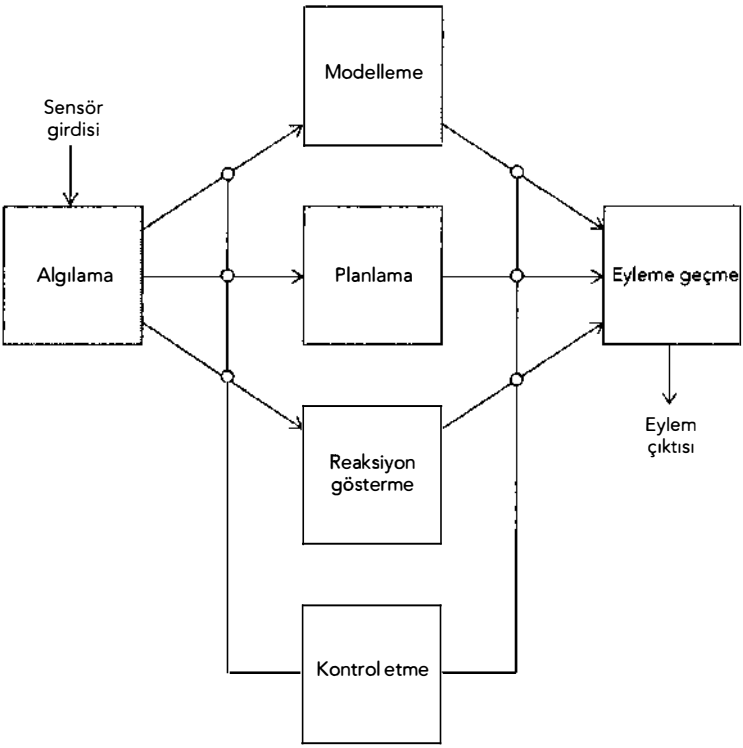
Yeni YZ'yi savunanların bir kısmı tavizsiz bir yaklaşımı benimsemişken –mantıksal temsil ve akıl yürütme kesinkes yasaktır– bir kısmı daha ılımlı bir çizgi tutturmuştur ve günümüze kadar alanda hâkimiyetini sürdüren de bu ikinci grup olmuştur. Ilımlılar davranışsal YZ'den önemli dersler almıştı, ama doğru çözümün davranışsal yaklaşım ile akıl yürütmeye dayalı yaklaşımın bir *kombinasyonu* olduğuna inanıyorlardı. Böylece YZ'de yeni bir istikamet ortaya çıktı: Brooks'un ortaya koyduğu belli başlı fikirleri kabul ederken, YZ'nin akıl yürütme ve temsil mirasında halihazırda başarıya ulaşmış olan ne varsa ona da kucak açıyorlardı.

YZ faaliyetlerinin odak noktası bir kere daha değişiyor, uzman sistemler ve mantıksal akıl yürütenler gibi *cisimsiz* YZ sistemlerin-

den, **ajan**lar inşa etmeye doğru kayıyordu. Ajanın "tam" bir YZ sistemi olması amaçlanıyordu, yani kendine yetecek, otonom olacak, belli bir ortamda bulunacak ve bir kullanıcı adına birtakım özel görevleri yerine getirecekti. Ajanın, mantıksal akıl yürütme gibi müstakil ve cisimsiz bir yetiden ziyade, tam ve entegre bir dizi yeti sergilemesi gerekiyordu. Salt zekânın *bileşenleri* yerine zekâya dayalı *tam* ajanlar inşa etmeye odaklanmanın, Brooks'un o hiç hazzetmediği safsatadan –YZ'nin zekâya dayalı davranışı bileşenlerine ayırarak (akıl yürütme, öğrenme vb.) ve bunları birbirinden ayrı olarak geliştirerek başarıya ulaşabileceği fikrinden– kaçınmayı sağlayacağı umuluyordu.

Ajan tabanlı YZ yaklaşımı doğrudan davranışsal YZ'den etkilenmiş ama mesajını biraz daha yumuşatmıştı. Bu hareket 1990'ların başında ortaya çıktı. O sıralarda "ajan" derken herkes başka bir şeyi kastediyordu. YZ camiasının bu konuda bir mutabakata varması 1990'ların ortasını buldu. Buna göre ajanlar üç önemli yetiye sahip olmalıydı. Birincisi, ajanlar reaktif olmalıydı: İçinde bulundukları ortama uyum sağlayabilmeli, davranışlarını ortamda meydana gelen değişikliklere adapte edebilmelilerdi. İkincisi, ajanlar proaktif olmalıydı: Sistematik bir biçimde çalışarak kendilerine verilen görevi kullanıcıları adına yerine getirebilmelilerdi. Üçüncü ve son olarak, ajanlar sosyal olmalıydı: Gerektiğinde başka ajanlarla beraber çalışabilmelilerdi. YZ'nin Altın Çağı'nda proaktif davranış, planlama ve problem çözme ön plandayken; davranışsal YZ reaktifliğin, belli bir ortamda cisimleşmiş ve bu ortama uyum sağlamış olmanın önemi üstünde duruyordu. Ajan tabanlı YZ ise ikisini de gerekli görüyor, ayrıca bunların yanına yeni bir şey daha ekliyordu: Ajanlar başka ajanlarla beraber çalışabilmelilerdi ve bunun için de *sosyal* yetilere sahip olmaları gerekiyordu. Sadece iletişim kurmak değildi burada söz konusu olan, görevlerini yerine getirmek için işbirliği, koordinasyon ve müzakereyi de becerebilmeleri gerekiyordu.

Ajan tabanlı YZ'yi kendisinden önce gelen diğer paradigmalardan ayıran da işte bu sonuncusu, ajanların *sosyal* olmalarının gerekliliğiydi. Geriye dönüp bakıldığında, YZ sistemlerinin birbiriyle nasıl etkileşime girebileceğini, girdiği takdirde hangi meselelerin gün-




**Şekil 11:** Tipik bir ajan mimarisi: TouringMachines.

deme geleceğini ciddi ciddi düşünmeye bu kadar geç başlanmış olması tuhaftır. Turing testinin vurguladığı sosyal yeti insanlarla İngilizce gibi doğal diller yoluyla etkileşime girme ve havadan sudan sohbet etme yetisiydi. Ajan tabanlı YZ açısından başlıca mesele *insanlarla* iletişim değil ajanların *başka ajanlarla* beraber çalışması fikriydi.

Tipik bir ajan tasarımını Şekil 11'de görebilirsiniz. **TouringMachines**[6] adıyla bilinen bu ajan mimarisinde ajanın genel kontrolü üç altsisteme ayrılıyordu. Reaksiyon altsistemi Brooks'un kapsama mimarisi gibi işliyordu; engelden kaçınma gibi hızlı tepki vermeyi gerektirirken akıl yürütmeye ihtiyaç duyulmayan durumlardan sorum-

luydu. Planlama altsistemi ajanın amaçlarına nasıl ulaşacağını bulmakla yükümlüyken, modelleme katmanı da diğer ajanlarla etkileşimleri ele alıyordu. Üç kontrol katmanına aracılık eden bir kontrol altsistemi tek tek her katmandan gelen tavsiyeleri dinleyip hangi tavsiyeye uyulacağına karar veriyordu. Genelde gayet basit bir karardı bu: Reaksiyon katmanı "DUR!" diyorsa, bu tavsiyeye kulak vermeniz muhtemelen hayırlı olacaktır. 1990'ların başında buna benzer pek çok ajan mimarisi geliştirildi.

## HOMER: Tam Ajan

Bu dönemde geliştirilen ajanlar arasında favorin hangisi diye soracak olursanız, HOMER derim. Bir robot denizaltı simülasyonu olan HOMER iki boyutlu bir "deniz dünyası"nda yer alıyordu.[7] Bu deniz dünyasında HOMER'a verilen görevler genelde birtakım paketleri bir yerden bir yere taşımakla ilgili oluyordu. Özetle SHRDLU'nun manevi oğlu gibi bir şeydi, hatta SHRDLU'nun başlıca kısıtlarından biri onda da vardı: Sadece bir simülasyondu HOMER. Ama çok daha sofistike bir simülasyon.

HOMER bir kullanıcıdan 800 kelimelik bir söz dağarcığından oluşan sınırlı bir İngilizceyle talimat alıyordu. Talimatları nasıl yerine getireceğini planlayabiliyor, planlarını gerçekleştirebiliyor, değişen koşullara uygun hale getirebiliyordu. Neler becerebildiğine dair genel bir fikir edinmek için, gelin HOMER'la gerçekleştirilen bazı diyaloglara bakalım. Bunlardan ilki yüzen bir tomrukla ilgili, HOMER tomruğu görebiliyor (düz yazılanlar kullanıcının girdileri, HOMER'ın girdileri italik, parantez içleri de benim yorumlarım):

Önünde ne var?
*Bir tomruk.*
Senin mi?
*Hayır, benim değil.*
Tomruk senin.
*Aa.* (Bu ünlem, HOMER'a önceden bilmediği bir şey söylendiğini gösteriyor.)
[...]

Başka tarafa dönerek tomruğundan uzaklaş.
*Tamam, dönüyorum.*
(Kullanıcı tomruğu kaldırır.)
Geri dön.
*Tamam, dönüyorum.*
*Tomruğu kaybettim!*

Bu diyalog sağduyuya dayalı akıl yürütmeye basit ama iyi bir örnek. HOMER tomruğu hatırlıyor, geri dönünce tekrar tomruğu görmeyi bekliyor, tomruğun ortadan kaybolduğunu fark edince şaşırıyor.

İkinci senaryoda HOMER iskelede bekliyor, iskelede bir koli var, görev ise koliyi oradan alıp bir mavnaya teslim etmek.

Koliyi haftaya cumartesi 21:00'de mavnaya bırak.
*Tamam.*
(HOMER planı hazırlar.)
Haftaya cumartesi iskeleye gidiyor musun?
*Evet.*
Neden?
*İskeleye bir koli almak için gideceğim.*
Koliyi almaya ne zaman gideceksin?
*20:56'da.*
Haftaya pazar koli nerede olacak?
*Mavnada.*
Haftaya bugün koli iskelede olacak mı?
*Hayır.*

Burada HOMER'ın zamanı, eylemlerinin içinde bulunduğu ortamı nasıl etkilediğini sağduyu temelinde anladığını görüyoruz (koliyi mavnaya bıraktıktan sonra, iskelede koli olmayacak). Ayrıca yaptığı planı gerçekleştirmesinin zaman alacağını anlamış olduğunu görüyoruz (iskeleden 20:56'da ayrılacak ki 21:00'de mavnaya varabilsin).

### YZ Yardımcıları

Ajan tabanlı YZ'nin kökeni robotik olsa da, birçok araştırmacı çok geçmeden bu paradigmanın yazılım dünyasına uygulandığında kul-

lanışlı sonuçlar vereceğini fark etti. Uzman sistemler gibi ajan tabanlı YZ'nin de genel YZ olmak gibi bir iddiası yoktu, daha ziyade işimizi görecek **yazılım ajanlarının** inşa edilebileceği düşünülüyordu. Yazılım ajanları masaüstü bilgisayarlar ve internet gibi yazılım ortamlarında işliyordu. Asıl fikir, e-postaları işlemek veya internette gezinmek gibi rutin gündelik faaliyetlerimize yardımcı olmak üzere bizimle birlikte çalışacak YZ destekli yazılım ajanları yapmaktı.

Yazılım ajanları fikrinin nasıl ortaya çıktığını anlamak için, bilgisayarlarla etkileşimimizin nasıl olduğunu ve bu konu hakkındaki fikirlerin zamanla nasıl evrildiğini biraz bilmemiz lazım.

İlk bilgisayarları kullananlar genelde Alan Turing gibi bu bilgisayarların tasarımına veya inşasına katkıda bulunmuş biliminsanları veya mühendislerdi. Bilgisayarların arayüzleri kullanışsızdı; sözgelimi Turing'in 1940'ların sonunda Manchester'dayken kullandığı Manchester Baby'de, bilgisayar belleğinin her bir biti için bir anahtarı "1" veya "0"ı gösterecek şekilde çevirerek ayarlamak gerekiyordu. Hiçbir şey kullanıcıdan gizlenmiyordu, makineyi programlayabilmesi için kullanıcı onun nasıl çalıştığını anlamak zorundaydı. Bu yolla programlama yapabilecek kadar teknik beceriye sahip insan sayısı haliyle çok azdı, dahası Manchester Baby'de edindiğiniz programlama becerisi Cambridge'deki bilgisayarda hiçbir işe yaramıyordu çünkü bu iki makine birbirinden dünyalar kadar farklıydı.

1950'lerin ortasında durum değişmeye başladı. Dönemin başlıca yeniliği **üst düzey programlama dilleri**ydi. Burada "üst düzey"le kastedilen, makinenin kimi alt düzey ayrıntılarının bilgisayarı programlayan kişiye açık olmamasıydı, yani belli bir bilgisayarı programlamak için ille de o bilgisayarın nasıl çalıştığını anlamak gerekmiyordu artık. Üst düzey programlama dilleri *makineden bağımsızdı*, sözgelimi bir bilgisayarda COBOL'da yazılan programın başka bir bilgisayarda da –az çok– çalışması beklenebiliyordu. Bir bilgisayarda edindiğiniz programlama becerisini başka bir bilgisayarda da kullanabiliyordunuz artık. Söz konusu yenilikler bilgisayarın kullanışlılığını dramatik bir biçimde artırdı. Bilgisayarlarla etkileşim tarzımızda "bilgisayar odaklı" bir bakış açısından "insan odaklı" bir anlayışa geçişin başlangıcıydı bu.

Müteakip altmış yıl boyunca bu trend devam etti, insan-bilgisayar etkileşimi giderek makine odaklı yaklaşımlardan daha insan odaklı olanlara doğru kaydı. 1980'lerde bu açıdan büyük bir sıçrama gerçekleşecek, zira Apple Bilgisayar firması 1984'te Macintosh ("Mac") bilgisayarını takdim edecekti. Mac geniş bir pazara yönelik ilk bilgisayardı, hedef kitlesinin belli bir alanda bilgisayar eğitimi almamış kişilerden oluştuğu açıkça belirtiliyor ve pazarlanırken bilhassa arayüzünün kullanım kolaylığı öne çıkarılıyordu. Mac'in grafik bir kullanıcı arayüzü (GUI) vardı, belge ve klasörleri temsil eden simgelerle dolu olması bakımından bir masanın üstünü andırıyordu; bir fareyle ekranda gördüğünüz bir imleci, bu imleçle de bütün simgeleri yönlendirebiliyordunuz. Bilgisayarınızın arayüzünü betimlemek için "masaüstü" teriminin kullanıldığını duymuşsunuzdur muhtemelen, ama bu terimin bilgisayar ekranınızı gerçek bir masanın üstüne benzetin diye özellikle seçildiği aklınıza gelmemiş olabilir. Bilgisayarınızın masaüstündeki belgeler fiziki bir masanın üstünde görebileceğiniz türden belgelerin benzerleri olarak tahayyül edilmiştir, masaüstündeki klasörler fiziki belgeleri düzenlemek için kullanabileceğiniz türden klasörler olarak düşünülmüştür, aynı şekilde bilgisayarınızdan kaldırmak istediğiniz belgeleri sürüklediğiniz masaüstündeki çöp sepeti de çalışma odanızdaki fiziki çöp kutusu göz önünde bulundurularak tasarlanmıştır. (Günümüzde bu benzerlik eskisi kadar çarpıcı değil artık, daha genç nesiller bu masaüstü metaforunun pek farkında değildir zannımca.)

1984'te Mac'in öncülük ettiği türden, masaüstü esasına dayalı, grafik kullanıcı arayüzleri bugün hâlâ standart durumunda.[8] Arayüzler hakkındaki en temel fikirlerin ta o zamanlardan bu zamanlara kadar böylesine az değişmiş olması, hele ki donanım alanında yaşanan ilerlemelerin ne kadar büyük olduğunu göz önünde bulundurunca, hakikaten çarpıcıdır. 1984 yılında Mac kullanan biri günümüzde üretilen bir Mac'in veya Microsoft Windows arayüzünün başına geçse, yolunu bulmakta pek zorlanmazmış gibi geliyor bana.

Öyleyse bu arayüzlerle, kullanıcının *aşina olduğu kavramlar* (masaüstü, belge, klasör, çöp kutusu) aracılığıyla bilgisayarla etkileşime girmesinin amaçlandığı, böylece etkileşim sürecinin daha

doğal ve dolayısıyla bilgisayarın da daha erişilebilir hale gelmesinin umulduğu söylenebilir.

1980'lerin sonunda, Mac'in yakaladığı başarının ardından, Apple'ın o zamanki CEO'su John Sculley insan-bilgisayar etkileşiminde bir sonraki Mac benzeri yeniliğin ne olabileceğine kafa yormaya başladı. Üstünde karar kıldığı fikre **Bilgi Gezgini** adını verdi, eli kulağında birçok teknolojinin (birkaç yıla kalmadan World Wide Web, yani Dünya Çapında Ağ ortaya çıkacaktı mesela) habercisi olan bir bilgi işlem vizyonuydu bu. Apple Bilgi Gezgini'ni tanıtmak için bu isimle bir video hazırlatıp 1987'de yayınladı.[9] Son derece spekülatif olan bu video, bir ürünün yol haritasının temelini şekillendirmekten çok, bir vizyonu gözler önüne sermeyi amaçlıyordu.

Videoda bir üniversite hocasını bilgisayarını kullanırken görürüz, profesörün bilgisayarı ile günümüz tablet bilgisayarları arasındaki benzerlik etkileyicidir. Tabletin kullanıcı arayüzü alışkın olduğumuz türden bir masaüstü arayüzü gibi görünmektedir, ama önemli bir istisnayla: Profesörün tablet ile etkileşimi bir yazılım ajanı aracılığıyla gerçekleşmektedir. Ajan, tıpkı HOMER gibi bildiğimiz İngilizceyi kullanarak etkileşime girer, ama HOMER'dan farklı olarak tabletin ekranındaki bir insan animasyonu şeklinde karşımıza çıkar. Ajan olanca kibarlığıyla profesöre gün içinde yapması gerekenleri hatırlatır, bir derse hazırlık için lazım olan materyallerle ilgili bazı aramalar yapar ve telefon görüşmelerini yönetir.

Video hayal ürünüdür (ve pek de komik olmayan esprilerle doludur) ama tarihsel açıdan önemli bir rolü vardır, hem de birden fazla açıdan. Birincisi, internetin nasıl çalışma ortamımızın rutin bir parçası haline geleceğinin sinyallerini verir. Videonun çekildiği dönemde şahısların, hatta şirketlerin çoğu için internet diye bir şey yoktu; internet daha ziyade üniversitelerin ve hükümetin (bilhassa ordunun) bir imtiyazı gibiydi. Video ayrıca tablet bilgisayarı öngörüyordu. Ama YZ camiası açısından videonun en önemli tarafı *bir bilgisayarla bir ajan aracılığıyla etkileşime girme* fikrini yerleştirmiş olmasıydı.

**Ajan tabanlı arayüz** insan-bilgisayar etkileşiminin öncekilerden tamamen farklı bir tarzını sunuyordu. Microsoft Word'ü veya Inter-

net Explorer'ı kullandığımızda, uygulamanın bizim taleplerimize karşılık olarak pasif bir rol oynadığını görürüz. Bu uygulamalar asla kontrolü veya inisiyatifi ele almaz. Eğer Microsoft Word bir şey yapıyorsa, biz menüden bir öğeyi seçtik veya bir yere tıkladık diye yapıyordur. Microsoft Word ile etkileşiminizde tek bir ajan vardır, o da sizsinizdir.

Ajan tabanlı arayüz işte bunu değiştiriyordu. Bilgisayar ne yapacağı söylenene kadar pasif bir biçimde beklerken, ajan tıpkı birlikte çalıştığınız bir insan gibi aktif bir rol alacaktı. Ajan bir yardımcı misali *sizinle birlikte çalışacak*, ne yapmak istiyorsanız onun yapılması için aktif bir biçimde sizinle işbirliğine girişecekti.

Bilgi Gezgini'ndeki ajan bir animasyonudur, video konferans bağlantısı yapan bir insan gibi görünür, İngilizcesi gayet akıcıdır. Bu yetiler dönemin YZ teknolojisinin fersah fersah ötesindeydi, hatta bugün bile biraz öyle. Ama bunlar meselenin sadece tali veçheleriydi. Asıl fikir, YZ'nin, bir yazılımı her buyurulanı pasif bir biçimde yerine getiren bir uşaktan ziyade aktif bir biçimde sizinle birlikte çalışan bir yardımcı haline getirebileceğiydi. Bu fikir tuttu, 1990'ların ortasında World Wide Web'in hızla genişlemesiyle iyice pekişti, yazılım ajanlarına ilgi hızla arttı.

Belçika doğumlu MIT Medya Laboratuvarı profesörü Pattie Maes'in büyük ilgi gören makalesinin başlığı zamanın ruhunu çok iyi yakalıyordu: "Fazla İş ve Enformasyon Yükünü Hafifleten Ajanlar".[10] Makalede Pattie'nin laboratuvarında geliştirilen çeşitli ajan prototipleri anlatılıyordu. Söz konusu ajanlar e-posta yönetimi, toplantı planlaması, haber filtreleme ve müzik tavsiyesiyle ilgiliydi. Mesela e-posta yardımcısı, kullanıcının e-posta aldığı zaman sergilediği davranışları gözlemleyecek (kullanıcı e-postayı hemen okuyor, dosyalıyor, hemen siliyor vb.) ve makine öğrenmesine dayalı bir algoritma kullanarak yeni bir e-posta geldiğinde kullanıcının ne yapacağını tahmin etmeye çalışacaktı. Ajan artık doğru tahminlerde bulunabildiğine kanaat getirdiği zaman inisiyatifi ele alıp e-postayı bu tahmin uyarınca işleyecekti.

Müteakip on yılda, böyle yüzlerce ajan geliştirildi. Bunların çoğu internete dayalıydı. World Wide Web'in ilk zamanlarında, arama

araçları henüz emekleme çağındaydı, internet bağlantıları günümüze kıyasla çok çok yavaştı. Bugün web'de rutin haline gelmiş olan pek çok görev o zamanlar fazla vakit alıyordu, dolayısıyla ajanların ağır işleri otomatikleştirebileceği umuluyordu.[11] 1990'ların sonunda, World Wide Web'e müthiş bir ilgi vardı. Yazılım ajanları hızla büyüyen web'e cuk oturan bir teknoloji gibi görünüyordu, dolayısıyla bu alanda faaliyet göstermek üzere yığınla girişim şirketi kuruldu. Bu gidişat ".com balonu"nun ortaya çıkmasında da rol oynayacaktı.

Aslına bakarsanız ben de bu hikâyenin bir parçasıydım. 1996 yazında birkaç meslektaşım, Londra merkezli iddialı bir girişim şirketine katılmamı teklif etti. O zaman üniversiteden aldığım maaşın üç katını veriyorlardı. En fazla yarım saniye düşünüp kabul ettim. Ajanları kullanarak web aramasını daha iyi hale getirmek gibi bir planımız vardı. World Wide Web'i bir kütüphaneye dönüştürmeyi umuyorduk. Ama iş dünyası hakkında en ufak bir fikrimiz yoktu ve, dürüst olmak gerekirse, ticari yazılım inşa etmekten anladığımız da söylenemezdi. Çok geçmeden anlaşıldı ki yatırımcının parasını su gibi harcamamıza rağmen bu parayı şirkete yeniden nasıl kazandıracağımızı bilmiyorduk. Uzun lafın kısası çok talihsiz günlerdi. Katıldıktan dokuz ay sonra şirketten ayrıldım, bundan sekiz ay sonra da şirket kapandı zaten.

Benim bu iç karartıcı girişim şirketi deneyimim bir bakıma birkaç yıl içinde dünya çapında olacakların habercisiydi. Zaman içinde iyice yerleşen adıyla ".com balonu" 1995'ten 2000'lerin başına kadar sürdü. İddialı ve maliyetli olan yeni internet girişim şirketleri gelirlerini gerçekleştiremeyince haliyle paraları tükenmeye başladı. 2000'lerin başında da .com balonu patladı.

Yazılım ajanları .com hikâyesinin cüzi bir parçasıydı belki ama en aşikâr YZ bileşeniydi. Yalnız bu örnekte YZ araştırmacılarının böyle bir hayali desteklemesinin yanlış bir tarafı yoktu, sadece bu hayalin gerçekleşmesi için biraz erkendi. Yirmi yıl kadar sonra Apple'ın iPhone'u için Siri diye bir uygulama piyasaya sürülecekti. Otuz sene öncesinde SHAKEY'yi geliştiren Stanford Araştırma Enstitüsü'nde geliştirilen Siri, 1990'larda yazılım ajanları üstüne yapı-

lan çalışmaların doğrudan bir devamı gibiydi. Dolayısıyla YZ camiası Siri ile Apple'ın Bilgi Gezgini arasında birçok benzerlik gördü. Siri kullanıcının doğal dil aracılığıyla etkileşime girebildiği ve kullanıcı adına birtakım basit görevleri yerine getirebilen bir yazılım ajanı olarak tahayyül edilmişti. Siri'nin hemen ardından yine geniş bir pazara hitap eden birçok yazılım ajanı piyasaya sürüldü. Amazon'un Alexa'sı, Google'ın Asistan'ı ve Microsoft'un Cortana'sı hep aynı vizyonun gerçekleşmesidir. Hepsi ajan tabanlı YZ'nin mirasçılarıdır. 1990'larda inşa edilebileceklerini düşünmek pek gerçekçi bir yaklaşım değildi çünkü o dönem mevcut olan donanım bunun için yeterli değildi. Hayata geçirilmeleri için gereken bilgisayar gücünü sağlayabilen mobil aygıtlar ancak 2010'lu yıllarda ortaya çıktı.

## Rasyonel Eylem

Ajan paradigması YZ hakkında düşünmenin bir yolu daha olduğunu ortaya koydu: *sizin adınıza etkili bir biçimde edimde bulunabilen ajanlar inşa etmek*. Ve bu da merak uyandıran bir meseleyi gündeme getirdi. Turing testi, YZ'nin amacının insan davranışından ayırt edilemeyen davranışlar üretmek olduğu fikrini yerleştirmişti. Ancak ajanların bizim adımıza edimde bulunmasını ve bizim için en iyi olanı yapmasını istiyorsak, *yaptıkları seçimlerin tıpkı bir insanın yapacağı türden seçimler* olup olmadığının önemi yoktur. Asıl istediğimiz *iyi* seçimler yapmaları, mümkün olan en iyi seçimlere yönelmeleridir. Böylece YZ'nin amacı *insan gibi* seçimler yapmaktan *optimal* seçimler yapmaya doğru kaymıştır.

YZ alanındaki pek çok çalışmanın temelini oluşturan optimal karar teorisi 1940'lara, John von Neumann'ın çalışmalarına kadar uzanır. Evet, 1. Bölüm'de ilk bilgisayarların tasarımları konusundaki çığır açıcı çalışmalarından bahsettiğimiz John von Neumann'ın ta kendisi. Ekonomist Oskar Morgenstern ile birlikte matematiksel bir rasyonel karar teorisi geliştirdiler. Bu teori rasyonel bir seçim yapma probleminin nasıl matematiksel bir hesaba indirgenebileceğini gösteriyordu.[12] Buna paralel olarak ajan tabanlı YZ'de de ajanın bu teo-

riden yararlanarak sizin adınıza optimal kararlar alabileceği fikri hâkimdi.

Söz konusu teori **tercihler**inizden yola çıkar. Ajanınız sizin adınıza edimde bulunacaksa, neler istediğinizi bilmesi gerekir. Bu durumda siz de ajanın tercihlerinizi en iyi şekilde gerçekleştirmesini istersiniz. Peki ama ajan tercihlerinizi nereden bilecek? Diyelim ki ajanın elma, portakal ve armut arasında bir seçim yapması gerekiyor. Sizin için en iyisini yapacaksa, hangi sonucu arzuladığınızı bilmesi lazım. Şöyle bir tercih sıranız olduğunu varsayalım:

Armut yerine portakal;
elma yerine armut.

Ajanınız elma ile portakal arasında seçim yapmak durumunda kalırsa ve portakalı seçerse sevinirsiniz ama elmayı seçerse hayal kırıklığına uğrarsınız. **Tercih ilişkisi**ne basit bir örnek bu. Tercih ilişkisi her alternatif sonuç çiftini nasıl derecelendirdiğinizi gösterir. Von Neumann ve Morgenstern tercih ilişkilerine tutarlılığın birtakım temel gereksinimlerini karşılamak için ihtiyaç duymuştur. Örneğin tercihleriniz aşağıdaki gibi olsun:

Armut yerine portakal;
elma yerine armut;
portakal yerine elma.

Tercihleriniz biraz garip, çünkü portakalı armuda ve armudu elmaya yeğlemenizden yola çıkarak portakalı elmaya yeğlediğiniz sonucuna varıyorum, ama bu sonuç son önermeyle çelişiyor. Tercihleriniz tutarsız. Bir ajanın bu tercihlere dayanarak sizin adınıza iyi bir karar vermesi mümkün değil gibi görünüyor.

Bir sonraki adım, **faydalar** kavramı yoluyla tutarlı tercihleri *sayısal* olarak temsil etmektir. Faydalar fikri esasen her olası sonucun bir sayıyla ilişkilendirilmesine dayanır: Sayı ne kadar büyükse, sonuç o kadar tercih ediliyor demektir. Mesela biraz önce verdiğimiz örnekteki tercihleri portakalın faydasına 3, armudun faydasına 2 ve elmanın faydasına 1 atayarak yakalayabiliriz. 3 sayısı 2'den, 2 sayısı da 1'den büyük olduğuna göre, bu faydalar ilk tercih ilişkimizi doğru

bir biçimde yakalıyor demektir. Bu örnekte faydaları pekâlâ portakalın değeri 10, armudunki 9 ve elmanınki 0 olacak şekilde de seçebilirdik. Gördüğünüz gibi değerlerin *büyüklüğü* fark etmiyor, sadece fayda değerlerinin tayin ettiği sonuçların *sırası* önemli. Dahası tercihleri sayısal fayda değerleriyle temsil etmek *ancak* tutarlılık kısıtlarını karşılamaları halinde mümkün. Elma, portakal ve armuda sayısal değer tayin ederek yukarıda verdiğimiz tutarsız tercih örneğini temsil edebilecek misiniz bakın bakalım.

Tercihleri temsil etmek için fayda değerlerini kullanmanın tek amacı, en iyi seçimi yapma problemini matematiksel bir hesaba indirgemektir. Ajanımız sonucu bizim açımızdan faydayı maksimize eden bir eylem seçer, başka bir deyişle en çok tercih ettiğimiz sonucu ortaya çıkarmak için bir eylem seçer. Optimizasyon problemleri denen böyle problemler matematik alanında kapsamlı bir biçimde ele alınmaktadır.

Ne yazık ki seçimler genelde daha alengirlidir çünkü işin içine belirsizlik girer. **Belirsizlik durumunda seçim** ortamlarında eylemlerin birden çok olası sonucunun olduğu senaryolar ele alınır. Bu sonuçlar hakkında tek bildiğimiz her birinin gerçekleşme *olasılığıdır*.

Bunu biraz daha açmak için, ajanınızın iki opsiyon arasında seçim yapmak zorunda olduğu şu senaryoya bakalım:[13]

Opsiyon 1: Yazı gelme olasılığı ile tura gelme olasılığı ortalama olarak eşit bir demir para atılır. Tura gelirse ajanınıza 4£ verilir. Yazı gelirse ajanınıza 3£ verilir.

Opsiyon 2: Yazı gelme olasılığı ile tura gelme olasılığı ortalama olarak eşit bir demir para atılır. Tura gelirse ajanınıza 6£ verilir. Yazı gelirse ajanınıza hiçbir şey verilmez.

Ajanınız hangi opsiyonu seçmelidir? Opsiyon 1'in daha iyi bir seçim olduğunu iddia ediyorum. Peki ama daha iyi olmasının gerekçesi tam olarak ne?

Neden öyle olduğunu anlamak için **beklenen fayda** kavramına ihtiyacımız var. Bir senaryonun beklenen faydası, o senaryodan sağlanacak *ortalama* faydadır.

Opsiyon 1'e bakalım. Attığımız demir parada yazı gelme olasılığı

ile tura gelme olasılığı ortalama olarak eşittir (paranın iki yüzünden biri diğerinden ağır değildir), başka bir deyişle yarı yarıyadır. Yani ajanınız atışların yarısında 4£, yarısında 3£ alır. Dolayısıyla Opsiyon 1'den kazanmayı beklediğiniz meblağ (0,5 x 4£) + (0,5 x 3£) = 3,5£ olur; bu, opsiyonun beklenen faydasıdır.

Ajanınız Opsiyon 1'i seçse, *pratikte* asla gerçekten 3,5£ almayacaktır elbette. Ancak bu seçimi yeterince çok sayıda yapma imkânınız olsa ve hep Opsiyon 1'i seçseniz, seçim başına ortalama 3,5£ alırsınız.

Aynı mantıkla, Opsiyon 2'nin beklenen faydası (0,5 x 6£) + (0,5 x 0£) = 3£ olur, yani Opsiyon 2'yi seçmek size ortalama olarak 3£ kazandırır.

Şimdi, von Neumann ve Morgenstern'in teorisinde rasyonel seçimin temel ilkesine göre rasyonel bir ajan **beklenen faydayı maksimize etme**ye yönelik bir seçim yapacaktır. Bu örnekte, beklenen faydayı maksimize eden seçim Opsiyon 1'dir, çünkü beklenen faydası 3£ olan Opsiyon 2'ye karşılık Opsiyon 1'in beklenen faydası 3,5£'dir.

Dikkat ederseniz Opsiyon 2, Opsiyon 1'le ilişkili hiçbir sonuçtan elde edemeyeceğiniz kadar yüksek bir meblağ, 6£ elde etmek gibi cezbedici bir olasılık sunuyor. Ne var ki Opsiyon 2'de hiçbir şey elde edememek de bir o kadar olası. Opsiyon 1'in beklenen faydasının daha büyük olmasının nedeni de bu zaten.

Beklenen faydayı maksimize etme fikri genelde çok yanlış anlaşılır. İnsani tercih ve seçimleri matematiksel bir hesaba indirgeme fikri herkesin hoşuna gidecek bir fikir değil. Bu hoşnutsuzluk çoğunlukla, faydanın para olduğu veya fayda teorisinin bir bakıma bencilce olduğu (zira beklenen faydayı maksimize eden bir ajan sadece kendisini düşünerek hareket ediyor olmalıdır) gibi yanlış kanaatlerden kaynaklanır. Halbuki faydalar tercihleri yakalamanın sayısal bir yolundan başka bir şey değildir: von Neumann ve Morgenstern'in teorisi bireyin tercihlerinin neler olduğu veya olması gerektiği konusunda tamamen nötrdür. Birey yerine melek veya şeytanın tercihleri söz konusu olsa yine aynı şekilde işleyecektir. Başkalarını kendinizden daha çok önemsiyorsanız, yine değişen bir şey

olmayacaktır: Tercihleriniz diğerkâmlığınızı yansıtıyorsa, fayda teorisi dünyanın en bencil insanı için nasıl işliyorsa sizin için de öyle işleyecektir.

1990'larda, Von Neumann ve Morgenstern modeline göre tanımlanan rasyonellik çerçevesinde, bizim adımıza rasyonel bir biçimde edimde bulunacak ajanlar inşa etme paradigması olarak YZ fikri YZ'nin yeni ortodoksluğu haline geldi, hatta bugün bile hâlâ öyle:[14] Çağımızda YZ'nin çeşitli kollarını birleştiren ortak bir tema varsa, budur. Günümüzde YZ sistemlerinin hemen hepsinde kullanıcının tercihlerini temsil eden sayısal bir fayda modeli vardır; sistem beklenen faydayı bu modele göre maksimize etmeye, kullanıcı adına rasyonel bir biçimde edimde bulunmaya çalışır.

## Belirsizlikle Baş Etmek

YZ'cileri uzun zaman meşgul eden problemlerden biri de, 1990'lar itibarıyla çok daha net bir biçimde ortaya konan, belirsizlikle baş etme problemiydi. Gerçekçi olma iddiasındaki her YZ sistemi belirsizlikle baş etmek zorundaydı, hem de ne belirsizlik. Bir örnek verecek olursak, sürücüsüz otomobil veri akışını sensörleri aracılığıyla sağlar ama sensörler kusursuz değildir. Mesela menzilölçer "engel yok" dese de, düpedüz yanılıyor olma ihtimali her zaman vardır. Menzilölçerin sahip olduğu enformasyon hiçbir işe yaramıyor değildir, değeri vardır elbette, ama bu enformasyonu muhakkak doğru kabul edemeyiz. Öyleyse bu enformasyondan, hata olasılığını gözden kaçırmadan, nasıl yararlanabiliriz?

YZ tarihi boyunca tekrar tekrar, belirsizlikle baş etmeye yönelik pek çok yaklaşım icat edildi. 1990'lar itibarıyla ise içlerinden biri baskın hale geldi: **Bayes çıkarımı**. Bayes çıkarımı İngiliz matematikçi Thomas Bayes tarafından 18. yüzyılda icat edildi. Kendisi, bugün onun şerefine **Bayes Teoremi** dediğimiz tekniğin bir versiyonunu da formülleştirdi. Bu teorem, kanaatlerimizi, edindiğimiz yeni enformasyon ışığında rasyonel olarak nasıl değiştirip düzelteceğimizle ilgilidir. Sözgelimi sürücüsüz otomobil örneğinde, kanaatler

önümüzde bir engel olup olmadığına dairdir, sensör verisi de yeni enformasyondur.

Her şey bir yana, Bayes Teoremi insanların içinde belirsizlik olan konularda karar vermekte bilişsel açıdan ne kadar yetersiz olduğunu gözler önüne sermesi bakımından bir hayli ilgi çekicidir. Meseleyi daha iyi anlamak için gelin şu örneğe bakalım:

Yeni bir grip virüsü her 1000 kişiden 1 kişiyi hayati tehlike oluşturacak biçimde etkiliyor. Grip için geliştirilen testin doğruluk oranı %99. Hemen test yaptırıyorsunuz, bir de bakıyorsunuz ki sonuç pozitif.

Sizce durum ne kadar ciddi olabilir?

Testinin pozitif çıktığını öğrenince, çoğu insanın durumu *fazlasıyla* ciddiye alacağı aşikâr. Nihayetinde testin *doğruluk oranı %99*! Yani testi pozitif çıkan biri olarak gribe yakalanmış olma olasılığınız ne kadar diye sorsak, çoğu insan muhtemelen (testin doğruluk oranına paralel olarak) %99 diye cevap verirdi.

Halbuki düpedüz yanlış bir cevaptır bu. Gribe yakalanmış olma ihtimaliniz ancak *0,1*'dir. Çok mantıksız geliyor değil mi? Burayı biraz daha açalım o zaman.

Diyelim ki rasgele seçtiğimiz 1000 kişiye grip testi yaptık. Bu 1000 kişiden muhtemelen sadece 1'inin gribe yakalanmış olacağını biliyoruz, dolayısıyla örneğimizde gerçekten de öyle olduğunu varsayalım: 1 kişi grip, kalan 999 kişi değil.

Önce grip olan zavallıya bakalım. Test %99 doğru olduğuna göre, birisi gribe yakalandıysa bunu 100 kereden 99'unda doğru bir biçimde tespit edecektir. Dolayısıyla test sonucunun pozitif çıkmasını (0,99 olasılıkla) bekleyebiliriz.

Şimdi de gribe *yakalanmamış* olan şanslı çoğunluğa bakalım. Aynı şekilde, test %99 doğruysa, her 100 testten 1'inde aslında gribe yakalanmamış bir kişinin sonucu hatalı bir biçimde pozitif çıkacaktır. Öyleyse 999 kişiye test yapıldığında 9-10 test hatalı bir biçimde pozitif çıkacak diye düşünebiliriz. Başka bir deyişle, bu 999 kişi arasından 9-10'u için, aslında *grip olmadıkları halde* testlerinin pozitif çıkması beklenebilir.

Yani 1000 kişiye test yapıyorsak bu testlerden 10-11 tanesinin

pozitif çıkmasını bekleyebiliriz, ama aynı zamanda pozitif çıkanlardan aslında sadece 1'inin sahiden gribe yakalanmış olmasını da bekleyebiliriz.

Kısacası, gerçekten gribe yakalanan insan sayısı az olduğundan, *yanlış pozitif* sayısı *doğru pozitif* sayısından çok daha fazla olacaktır. (Yeri gelmişken, doktorların tam anlamıyla bir teşhis koyabilmek için birden fazla test sonucu görmek istemesinin nedeni de budur.)

Öyleyse şimdi de Bayesci akıl yürütme tarzının robotik alanında nasıl kullanılabileceğine bakalım. Diyelim ki meçhul bir alanı –ister uzak bir gezegen olsun ister depremle yerle bir olmuş bir bina– keşfe çıkan bir robotumuz var. İstiyoruz ki robotumuz bu alanı keşfedip haritasını çıkarsın. Robot, sensörleri aracılığıyla ortamı algılayabilir algılamasına ama şöyle bir sorun var: Sensörler hata yapabilir. Robot gözlem yaparken sensörler "Bu konumda bir engel var" diyorsa, söylediği doğru da olabilir yanlış da. Kesin olarak bilemeyiz. Robotumuz sensör verilerini daima doğru kabul ederse, yanlış enformasyon aldığında da buna dayanarak bir karar verip önündeki engele çarpacaktır.

Bayes Teoremi burada şu şekilde kullanılır: Grip testinde olduğu gibi, sensör okumasının doğru olma olasılığını kullanarak, belirtilen konumda bir engel olduğuna dair kanaatimizi güncelleyebiliriz. Çok sayıda gözlem yaparak içinde bulunduğumuz ortamın haritasını düzelte düzelte daha doğru bir hale getirebiliriz.[15] Böylece engellerin belirtilen yerlerinin doğruluğundan daha emin oluruz.

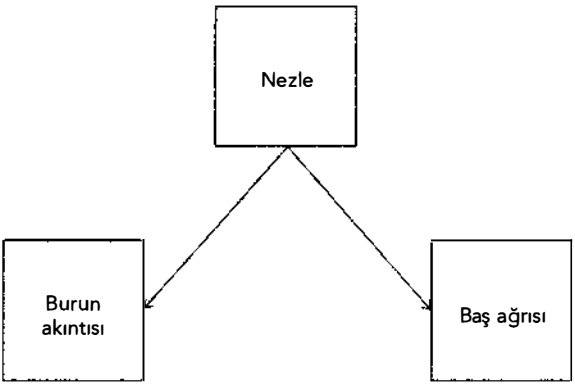
Bayes Teoremi kusurlu veriyi doğru bir biçimde ele almayı sağlaması bakımından önemlidir: Veriyi ne ıskartaya çıkarırız ne de sorgusuz sualsiz kabul ederiz. Bu teorem yoluyla dünyanın nasıl bir yer olduğuna dair olasılığa dayalı kanaatlerimizi değiştirip düzeltiriz.

Bayes Teoremi güçlüdür ama Bayesci akıl yürütme tarzını YZ'de kullanılabilir hale getirmek için yapılması gereken çok iş vardır, çünkü YZ sistemleri genelde birbiriyle ilişkili *bir sürü* karmaşık veriyle uğraşmak durumundadır. YZ araştırmacıları verilerin birbiriyle ilişkisini yakalamak için **Bayes ağlarını** geliştirmiştir. Bu ağlar ve-

riler arasındaki mevcut ilişkilerin grafik şeklindeki temsilleridir. Güncel biçimlerini almalarında Judea Pearl'ün çalışmaları önemli bir rol oynamıştır. Çalışmaları ciddi bir etkiye sahip olan Pearl YZ'de olasılığın rolünün anlaşılması ve net bir biçimde oraya konması için belki de herkesten daha çok uğraşmıştır.[16] Şekil 12'deki Bayes ağına bakalım. Bu ağ üç hipotez arasındaki ilişkileri yakalıyor: *Nezlesiniz*, *Burnunuz akıyor* ve *Başınız ağrıyor*. Bir hipotezden diğerine uzanan oklar bunların birbirini nasıl etkilediğini gösteriyor. Kabaca, *X* hipotezinden *Y* hipotezine uzanan bir ok, ilkinin doğruluğunun ikincinin doğru olmasını bir ölçüde etkilediği anlamına geliyor denebilir. Mesela nezle olup olmamanız burnunuzun akıp akmamasını etkiler; şayet burnunuz akıyorsa, bu da nezle olup olmama ihtimaliniz hakkında bize bir şeyler söyler. Bu çeşitli olasılıklar arasındaki ilişkiler Bayesci akıl yürütme tarzı yoluyla yakalanır. Bu konuda daha ayrıntılı bilgi için bkz. Ek C.

## Siri, Siri ile Karşılaşınca

2000'lerde .com furyası sona ermiş olsa da ajanlar hâlâ ilgi görüyordu ve bu ilgi ajan hikâyesindeki yeni bir yöne odaklanmaya başlamıştı. Araştırmacılar *ajanlar birbiriyle konuşabilse* neler yapılabileceğini merak ediyordu. Hiç düşünülmemiş bir şey değildi bu aslında; bilgi tabanlı YZ zamanlarında araştırmacılar birbiriyle uzmanlığını paylaşabilen uzman sistemlere kafa yormuş, böyle sistemlerin bilgi paylaşmasına ve birbirinin uzmanlık alanında sorgu yapmasına imkân veren yapay diller geliştirmişlerdi. Fakat bu yeni **çok ajanlı sistemler** fikri ile öncekiler arasında belirleyici bir fark vardı: Diyelim ki ajanımın gidip benim için yapabileceğinin en iyisini yapmasını istiyorum, siz de ajanınızın gidip sizin için yapabileceğinin en iyisini yapmasını istiyorsunuz, ama benim çıkar ve tercihlerim sizinkilere ters düşüyor, dolayısıyla ajanlarımızın çıkar ve tercihleri de çatışıyor. Böyle bir durumda ajanlarımızın bizim gündelik hayatta birbirimizle etkileşime girerken başvurduğumuz türden sosyal yetilere ihtiyacı olacaktır. İşte bu nedenle, YZ'yi bekleyen yeni zorluk bu tür yetilere sahip ajanlar inşa etmekti.[17]




Şekil 12: Basit bir Bayes ağı.

Şimdi geriye dönüp bakınca, YZ'nin *sosyal* veçhelerinin o zamana kadar pek de dikkate alınmamış olması garip geliyor. Çok ajanlı sistemler ortaya çıkmadan evvel, tek tek ajanlar inşa etme fikrine yoğunlaşılmış, YZ ajanlarının birbiriyle nasıl etkileşime gireceği gözden kaçırılmıştı. Dolayısıyla bir yerine *birden çok* ajan olabileceği varsayımı YZ hikâyesini temelden değiştirdi. Bir ajanın çözmesi gereken problem ne yapılacağını, benim adıma hangi eylemi gerçekleştireceğini bilmektir. Benim adıma hangi eylemi gerçekleştireceğini *iyi* seçerse, o ajandan memnun kalırım. Ama ortada birden çok ajan varsa, benim ajanımın seçtiği eylemin iyi mi kötü mü olduğu –en azından kısmen– *diğer ajanların ne yapmayı seçtiğine* bağlı olacaktır muhtemelen. Dolayısıyla ajanım karar verirken diğer ajanların ne yapmasının daha olası olduğunu hesaba katmalı, aynı şekilde diğer ajanlar da kararlarını verirken benim ajanımın eylemlerini göz önünde bulundurmalıdır.

Ajanların ne yapılacağına karar verirken birbirlerinin tercihlerini ve olası eylem seçimlerini göz önünde bulundurmalarına dayanan bu akıl yürütme tarzı aslında daha önceden iktisadın stratejik karar vermeyi inceleyen bir dalı olan **oyun teorisi** alanında çalışılmıştır.[18] Adından anlaşılacağı üzere oyun teorisi esasen satranç ve poker gibi oyunların incelenmesine dayansa da, ortada birden çok

ajan varken karar vermeyi gerektiren daha pek çok yerde kendine uygulama alanı bulmuştur.

Oyun teorisinin belki de en meşhur fikri, aynı zamanda çok ajanlı sistemlerde karar vermenin temel taşı haline gelen **Nash dengesidir**. Fikri ortaya atan John Forbes Nash Jr olmuştur (evet, daha önce John McCarthy'nin 1956'da düzenlediği Darmouth yaz okulunun katılımcılarını sayarken bahsettiğimiz Nash). Bu konudaki çalışmaları 1994'te Nash'e (John Harsanyi ve Richard Selten ile birlikte) iktisat alanında Nobel Ödülü'nü kazandırmıştır.

Nash dengesinin altında yatan fikri anlamak hiç de zor değil. Diyelim ki elimizde sadece iki ajan var ve ikisinin de bir seçim yapması gerekiyor. Ajan I $x$ yapmayı seçsin, Ajan II de $y$ yapmayı. Bunların iyi kararlar olup olmadığını nasıl söyleyebiliriz? Bu noktada, *ikisi de seçiminden pişman olmazsa*, kararları iyi kararlardır diye düşünülür (teknik olarak, bir Nash dengesi oluştururlar). Biraz daha açarsak:

> Ajan I memnundur çünkü, Ajan II $y$ yapmış olduğuna göre,
> Ajan I'in yapabileceği en iyi şey $x$'ti; ve
> Ajan II memnundur çünkü, Ajan I $x$ yapmış olduğuna göre,
> Ajan II'nin yapabileceği en iyi şey $y$'ydi.

Bunlara *denge* denmesinin sebebi karar verme sürecinde bir tür istikrar yakalamalarıdır. İki ajanın da başka bir şey yapması için herhangi bir saik yoktur.

Çok ajanlı sistemler camiası Nash dengesi gibi oyun teorisi fikirlerini hemen benimsedi, kendi sistemlerinde karar verme sürecinde bunları temel almaya başladı. Tam o noktada ise bu konuyla ilgili başka bir sıkıntı olduğu anlaşıldı. Nash 1950'lerde kendisine bir Nobel kazandıran fikri formüle ederken, Nash dengesinin nasıl *hesaplanacağı* üstünde pek durmamıştı. Ancak karar verirken bu fikirden yararlanabilen ajanlarımız olsun istiyorsak, bu elbette temel bir problemdir. Neticede, belki de bekleneceği üzere, Nash dengelerini hesaplamanın zor olduğu ortaya çıktı. Bu hesaplamanın randımanlı yollarını bulmak bugün YZ'de hâlâ başlıca konulardan biri olarak önümüzde duruyor.

## YZ'nin Olgunluk Çağı

1990'ların sonunda ajan paradigması YZ'nin müesses ortodoksluğu haline gelmiştir: Ajanlar inşa edip bizim için en iyisini tercih etme işini onlara devrederiz; ajanlar söz konusu tercihleri gerçekleştirmek üzere bizim adımıza rasyonel bir biçimde hareket eder, bunu yaparken de von Neumann ve Morgenstern'in beklenen fayda teorisinin örneklediği rasyonel karar vermeye başvururlar; ajanlar Bayesci akıl yürütme tarzını, bir Bayes ağı veya bunun bir varyasyonu yoluyla yakaladıkları dünya kavrayışıyla birlikte kullanarak dünya hakkındaki kanaatlerini rasyonel bir biçimde yönetirler; birden fazla ajan varsa, karar verme sürecinin çerçevesini oyun teorisinden alırız. Genel YZ değildi bu, Genel YZ'nin yol haritasını falan da vermiyordu. Ancak böyle araçların giderek daha fazla kabul görmesi, 1990'ların sonunda YZ'nin bizzat YZ camiasında farklı bir biçimde algılanmaya başlamasına yol açtı. İlk defa gerçekten de karanlıkta el yordamıyla yolumuzu bulmaya çalışmanın ötesine geçmiş gibiydik, alanımızın olasılık teorisi ve rasyonel karar teorisinin saygın ve kanıtlanmış tekniklerine dayanan gayet sağlam temelleri vardı artık.

Bu dönemde elde edilen iki başarı YZ disiplininin artık belli bir olgunluğa eriştiğini gösteriyordu. Bunlardan ilki dünyanın dört bir yanında gazetelere manşet oldu, yine bir o kadar önemli olan ikincisi ise alanın dışındakiler tarafından pek bilinmiyordu.

Kendine manşetlerde yer bulan ilk atılım IBM'den geldi: IBM'in YZ sistemlerinden DeepBlue 1997'de büyük satranç ustası Rus Garry Kasparov'u yenmeyi başardı. DeepBlue ilk kez Şubat 1996'da altı maçlık bir turnuvada Kasparov'a karşı bir maç almış, ama totalde Kasparov turnuvayı 4'e 2 kazanmıştı. Aşağı yukarı bir sene sonra DeepBlue'nun daha üst bir versiyonu ile Kasparov tekrar karşı karşıya geldi ve bu kez turnuvayı kazanan DeepBlue oldu. Görünüşe göre Kasparov o dönemde bu durumdan hiç hoşlanmamış, hatta IBM'in hile yaptığını düşünmüştü, ama sonradan, halihazırdaki parlak satranç kariyerinin yanı sıra bu deneyim hakkında konuşmak gibi bir ek kariyerin tadını çıkardı. O tarihi andan itibaren satranç ne-

resinden bakarsanız bakın çözülmüştü artık. En iyi oyuncuları gayet rahat yenebilen YZ sistemleri vardı.

DeepBlue'nun başarısında iki etmen önemli rol oynamıştı. Bunların ilki sezgisel aramaydı. 1950'lerin çığır açan dama oyunu programını yazan Arthur Samuel DeepBlue'yu görse temel tekniklerini rahatlıkla anlardı, çünkü bunlar müteakip kırk yılda sadece biraz daha rafine hale getirilmişti, o kadar. İkinci etmen ise son derece tartışmalıydı: DeepBlue esasen bir süper bilgisayardı. İşini yapması için çok fazla bilgisayar gücü gerekiyordu. Bu da sistemin aslında YZ sayılamayacağı, düpedüz bilgi işlem kuvvetinden başka bir şey olmadığı yolunda eleştirilere yol açtı. Ancak meseleye böyle yaklaşmak, bütün bu bilgisayar gücünden yararlanan YZ tekniklerinin önemini hafife almak olur. Satrancı mesela 2. Bölüm'de Hanoi Kuleleri bulmacası bağlamında değindiğimiz türden bir kapsamlı arama yaklaşımıyla ele alsaydık, aynı bilgi işlem gücüne sahip olsak bile başarılı olamazdık.

DeepBlue'nun Kasparov'u yendiğini duymayan yoktur muhtemelen, ama dönemin diğer büyük başarısı genelde pek bilinmez, halbuki çok daha esaslı sonuçlar doğurmuştur. 2. Bölüm'den hatırlarsınız: Bazı bilgi işlem problemleri tabiatı itibarıyla zordur –teknik tabirle NP-tamdır– ve YZ ile ilgili pek çok problem bu kategoriye girer. 1990'ların başında NP-tam problem algoritmalarında kaydedilen ilerlemeyle beraber, yirmi sene önceki gibi temel bir engel olarak görünmüyordu artık bu.[19] NP-tam olduğu gösterilen ilk problemin adı SAT idi (*satisfiability* yani "tatmin edilebilirlik"in kısaltması). Basit mantıksal ifadelerin tutarlı olup olmadığını kontrol etme, doğru olmalarının bir yolu var mı diye bakma problemidir bu. SAT bütün NP-tam problemlerin en temelidir. Ve yine hatırlarsanız, NP-tam problemlerin sadece belli bir türünü çözmenin randımanlı bir yolunu bulabilirseniz otomatikman hepsini çözmenin bir yolunu bulmuş olursunuz, demiştik. SAT problemlerini çözen programlar yani "SAT çözücüler", 1990'ların sonunda yeterince güçlü hale gelince, sanayi ölçeğinde problemlerin çözümünde kullanılmaya başladılar. Bugün, ben bu satırları yazarken, SAT çözücüler artık o kadar randımanlı ki bilgi işlemin pek çok aşamasında rutin olarak kul-

lanılıyorlar, NP-tam problemlerle karşılaştığımızda başvurduğumuz başlıca araçlardan biri haline gelmiş durumdalar. Yalnız NP-tam problemler artık her daim randımanlı bir biçimde çözülebilir demek değildir bu: SAT çözücülerin feriştahı gelse çözemeyeceği örnekler her zaman olacaktır. Ama NP-tamlık eskisi gibi gözümüzü korkutmuyor artık – son kırk yılda YZ alanında yaşanan, çok önemli olmakla beraber pek dillendirilmeyen başlıca gelişmelerden biri işte budur.

Nihayet belli bir saygınlık kazanmış olsa da, 1990'ların sonu itibarıyla YZ'nin halinden memnun olmayanların sayısı hiç de az değildi. Temmuz 2000'de Boston'da bir konferansa katılmıştım, yeni YZ'nin gelecek vadeden yıldızlarından birinin sunumunu dinliyordum. Hemen yan koltukta artık iyice kaşarlanmış bir YZ emektarı oturuyordu, YZ'nin Altın Çağı'ndan beri bu işlerin içinde olan, McCarthy ve Minsky zamanlarını görmüş biri. Duyduklarına burun büküyordu. "Bugün bunlar mı YZ sayılıyor artık?" diye sordu. "Ee, nerede kaldı o zaman bu işin büyüsü?" Neden böyle düşündüğünü anlıyordum: Eskiden, YZ'de kariyer yapacaksanız, felsefeden veya bilişsel bilimlerden veya mantıktan anlıyor olmanız gerekiyordu, bugün ise olasılıktan, istatistikten ve iktisattan. Kulağa pek, tabiri caizse, *şiirsel* gelmiyor değil mi? Ama gerçek şu ki işe yarıyor.

YZ bir kere daha yeni bir istikamete yönelmişti ama bu kez gelecek daha istikrarlı görünüyordu. Bu kitabı 2000 yılında yazıyor olsaydım, meslek hayatımın kalanında YZ'nin geleceğinin böyle olacağına sizi temin ederdim. Daha fazla dramaya yer yoktu artık: Bu araçları kullanarak, yavaş ama emin adımlarla ilerleyecektik işte. Bilmiyordum ki çok geçmeden son derece dramatik gelişmeler yaşanacak, YZ bir kere daha temellerinden sarsılacak.

# 5
# Çığır Açan "Derin" Atılımlar

OCAK 2014'TE, BİRLEŞİK KRALLIK teknoloji endüstrisinde eşi benzeri görülmedik bir şey oldu: Google, Silikon Vadisi standartlarına göre cüzi kalıyorsa da Birleşik Krallık'ın gayet mütevazı bilgisayar teknolojisi sektörünün standartlarına göre olağanüstü bir anlaşmayla, küçük bir girişim şirketi olan DeepMind'ı satın aldı. O dönem 25'ten az çalışanı olan şirketin tam tamına 400 milyon sterline satıldığı söyleniyordu.[1] Dışarıdan bakınca DeepMind'ın ne ürünü ne teknolojisi ne de iş planı varmış gibi görünüyordu. Hatta faaliyet göstermeyi seçtiği uzmanlık alanında yani YZ çevrelerinde bile neredeyse hiç bilinmiyordu.

Bu kadar küçük bir YZ şirketinin böylesine yüksek bir meblağa satın alınması manşetlere taşındı. Bütün dünya bu gizemli ekibi tanımak, Google'ın neden adı sanı duyulmadık bir şirketin bu kadar edebileceğini düşündüğünü anlamak istiyordu.

Yapay zekâ birdenbire büyük haber olunca haliyle büyük paraların döndüğü bir iş alanı haline de geldi. Müthiş bir ilginin odağı oldu. Havayı iyi koklayan basın her gün YZ haberleri yapıyordu. Dünyanın dört bir yanındaki hükümetler de durumu doğru tespit edip ne gibi adımlar atmalıyız diye düşünmeye başladı. Çok geçmeden çeşitli YZ girişimlerinde bulundular. Olan bitenden geri kalmak istemeyen teknoloji şirketleriyle beraber, eşi benzeri görülmedik bir yatırım dalgası geldi. DeepMind kadar üst düzey olmasa da, daha nice YZ satın alımı gerçekleşti. Uber 2015'te yaptığı bir anlaşma ile

Carnegie Mellon Üniversitesi'nin makine öğrenmesi laboratuvarından tam 40 araştırmacıyı işe aldı.

On yıla kalmadan, vaktiyle bilgi işlemin atıl bir kolu olan YZ bilimin en hararetli, en çok pompalanan alanı haline geldi. Rüzgârın bu kadar ani yön değiştirmesinde, en temel YZ teknolojilerinden olan makine öğrenmesindeki hızlı ilerlemenin etkisi büyüktü. Makine öğrenmesi YZ'nin bir alt dalıdır ama son altmış yıllık süreçte kendi yolunu çizmiştir. Bu bölümde göreceğimiz üzere, YZ ile birdenbire gözbebeği haline gelen "yavrusu" arasında zaman zaman gergin bir ilişki olmuştur.

Bu bölümde, 21. yüzyılın makine öğrenmesi devriminin nasıl gerçekleştiğinden bahsedeceğiz. Önce makine öğrenmesinin ne olduğuna dair kısa bir değerlendirme yapacağız; ardından belli bir makine öğrenmesi yaklaşımının, nöral ağların nasıl alana hâkim olmaya başladığına bakacağız. YZ'nin kendi hikâyesi gibi nöral ağların hikâyesi de sıkıntılıdır: İki "nöral ağ kışı" yaşanmıştır; nöral ağlar daha bu yüzyılın başında bile YZ çevreleri tarafından ölü veya can çekişen bir alan addedilmiştir. Ancak nöral ağlar bu süreçten nihayetinde güçlenerek çıkmıştır. Bunu sağlayan başlıca fikir **derin öğrenme** denen yeni bir teknik olmuştur. Derin öğrenme (*deep learning*) DeepMind'ın (derin zihin) temel teknolojisidir. Size DeepMind'ın hikâyesini, inşa ettiği sistemlerin dünyanın dört bir yanında nasıl ilgi gördüğünü anlatacağım. Derin öğrenme güçlü ve önemli bir tekniktir ama YZ için son nokta demek değildir. Bu yüzden, diğer YZ teknolojilerinden bahsederken yaptığımız gibi, derin öğrenmenin kısıtlarını da ayrıntılı olarak ele alacağız.

## Kısaca, Makine Öğrenmesi

Makine öğrenmesinin amacı, bilgi işlem aracılığıyla verili bir girdiden arzu edilen bir çıktı elde eden, bunu da *nasıl yapılacağına dair açık ve net bir tarif almaksızın* yapabilen programlara ulaşmaktır. Bir örnek verelim. Makine öğrenmesinin klasik uygulamalarından biri metin tanıma, yani elle yazılmış bir metnin bilgisayar ortamına aktarılarak transkripsiyonunun yapılmasıdır. Burada girdi elle ya-

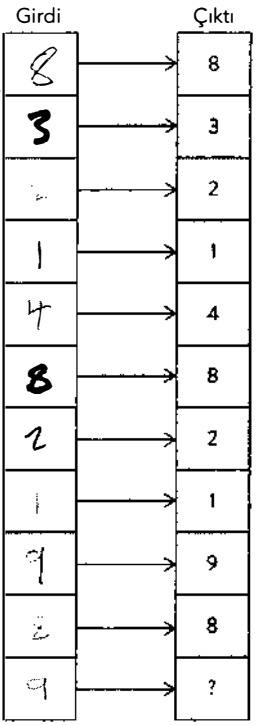
zılmış metnin fotoğrafı, arzulanan çıktı da elle yazılmış metinle temsil edilen karakterler dizisidir.

İstediğiniz postane çalışanına sorun, metin tanımanın hiç de kolay bir iş olmadığını söyleyecektir. Herkesin yazısı farklıdır, çoğunluğunki okunaksızdır; kâğıda mürekkep sızdıran kalemler kullanırız; kâğıt kirlenip yıpranır. 1. Bölüm Şekil 1'deki sınıflandırmayı hatırlayacak olursak, metin tanıma nasıl uygun bir tarif bulacağımızı bilmediğimiz problemlere örnektir. Bu iş masa oyunları oynamaya benzemez, çünkü masa oyunlarında teoride işleyen ama uygulamaya koyulmaları için sezgisel yöntemler gereken tarifler söz konusudur, oysa metin tanıma için bir tarifin nasıl bir şey olabileceğini bilmiyoruz. Bambaşka bir şey lazımdır bize ve tam da bu noktada makine öğrenmesi devreye girer.

Metin tanıma için bir makine öğrenmesi programı genelde, her bir örneğe gerçekte yazılan metnin etiketlendiği, elle yazılmış bir sürü karakter örneği verilerek **eğitilir**. Şekil 13 bunu gösteriyor.

Bu anlattığımız makine öğrenmesi türü, yani **gözetimli öğrenme** son derece önemli bir noktanın altını çiziyor: *Makine öğrenmesi için veri gerekir*. Hem de bir sürü. Birazdan göreceğimiz üzere, makine öğrenmesinin mevcut başarısında, dikkatle seçilip düzenlenmiş eğitim verisi kümelerinin sağlanmış olması hayati bir rol oynamıştır.

Bir makine öğrenmesi programını eğitirken, kullanacağımız eğitim verilerine çok dikkat etmemiz gerekiyor. Birincisi, genelde programı olası girdi ve çıktıların sadece çok küçük bir kısmıyla eğitebiliriz. El yazısı örneğimize dönersek, programa yeryüzünde mevcut olan el yazısıyla yazılmış bütün karakterleri gösteremeyiz, bu imkânsızdır. Zaten programı mümkün olan girdilerin tamamıyla eğitebilseydik, makine öğrenmesi konusunda akıllıca bir tekniğe de ihtiyaç kalmazdı: Programımızın her bir girdiyi ve buna karşılık arzu edilen çıktıyı hatırlaması yeterdi. Ne zaman bir girdi sağlansa, tek yapması gereken buna hangi çıktının tekabül ettiğine bakmak olurdu. Bu da makine öğrenmesi falan değildir tabii. Öyleyse bir programı eğitmek için, mümkün olan bütün girdi ve çıktıların ancak cüzi bir kısmı kullanılabilecektir. Diğer taraftan, eğitim verisi kümesi haddinden küçük kalırsa da programa yeterince enformasyon




**Şekil 13:** Bir makine öğrenmesi programına elle yazılmış karakterleri (bu örnekte, sayıları) tanıması için verilen eğitim verileri. Burada amaç, programı elle yazılmış sayıları kendi başına tespit edebilir hale getirmektir.

sağlanmamış olur, dolayısıyla program girdilerle çıktıları arzu edildiği gibi eşleştirmeyi öğrenemeyebilir.

Eğitim verileriyle ilgili bir diğer temel mesele de **öznitelik çıkarma**dır. Diyelim ki bir banka için çalışıyorsunuz, sizden kötü kredi risklerini tespit etmeyi öğrenecek bir makine öğrenmesi programı isteniyor. Programınızın eğitim verileri muhtemelen bir sürü eski müşteri kaydından oluşacak, her bir kayda müşterinin nihayetinde iyi risk mi yoksa kötü risk mi çıktığı etiketlenecektir. Müşteri kaydında alışıldığı üzere ad, doğum tarihi, ikamet adresi, yıllık gelir, ayrıca işlem kayıtları, krediler ve geri ödemeleri vb. olur. İşte bu

bileşenlere eğitim için kullanılabilecek **öznitelik**ler denir. Peki *bütün* bu öznitelikleri eğitim verilerinize dahil etmenin bir mânâsı var mı? Çünkü bazı özniteliklerin bir müşterinin iyi risk mi yoksa kötü risk mi olduğuyla hiç alakası yoktur belki. Hangi özniteliklerin sizin probleminiz için lazım olacağını önceden bilmiyorsanız, her şeyi dahil etmeye kalkabilirsiniz. **Çok boyutluluk belası** diye bilinen büyük bir problemdir bu: Ne kadar çok özniteliği dahil ederseniz, programa o kadar eğitim verisi sağlamanız gerekir, dolayısıyla program da bir o kadar yavaş öğrenir.

Bu nedenle, eğitim verilerinize az sayıda öznitelik dahil edebilirsiniz. Ancak bu da kendi içinde sorunludur. Bir kere, programın doğru dürüst öğrenebilmesi için gerekli olan, gerçekten de kötü kredi riskini gösteren öznitelikleri yanlışlıkla atlayabilirsiniz. Ayrıca, hangi öznitelikleri dahil edeceğinizi iyi seçmezseniz, programınızı yanlı hale getirme riskiyle de karşı karşıya kalırsınız. Örneğin kötü risk programınıza dahil etmeye karar verdiğiniz tek öznitelik müşterinin adresi olursa, programın kimi muhitlerde ikamet edenlere karşı ayrımcılık yapmayı öğrenmesine yol açmanız gayet olasıdır. YZ programlarının yanlı hale gelmesinin olabilirliğini ve bunun yol açtığı problemleri ileride daha ayrıntılı ele alacağız.

**Pekiştirmeli öğrenme**de programa apaçık eğitim verileri vermeyiz; program kendisi deneye deneye yolunu bulur, yani kararlar verir ve bu kararların iyi mi kötü mü olduğuna dair geribildirim alır. Pekiştirmeli öğrenme örneğin oyun oynama programlarının eğitiminde yaygın olarak kullanılır. Program oyunu oynar, kazanırsa olumlu veya kaybederse olumsuz geribildirim alır. Aldığı geribildirime, ister olumlu olsun ister olumsuz, **ödül** denir. Program oyunu bir sonraki oynayışında bu ödülü göz önünde bulundurur. Olumlu bir ödül alırsa büyük ihtimalle yine aynı şekilde oynar ama olumsuz bir ödül alırsa aynı şekilde oynama ihtimali düşüktür.

Pekiştirmeli öğrenmedeki başlıca zorluklardan biri pek çok durumda ödülün gelmesinin vakit almasıdır. Bu durum programın hangi eylemlerin iyi hangilerinin kötü olduğunu bilmesini zorlaştırır. Diyelim ki pekiştirmeli öğrenme programımız oynadığı bir oyunu kaybetti. Bu yenilgi programa, hatta bize oyun sırasında yapılan bel-

li bir hamle hakkında ne söyler? Hamlelerin *hepsinin* kötü olduğu sonucuna varmak muhtemelen aşırı genelleme yapmak olacaktır. Peki hangilerinin kötü hamleler olduğunu program nasıl bilir, biz nasıl biliriz? Buna **belirleme problemi** denir. Belirleme problemiyle gündelik hayatta sıkça karşılaşırız. Diyelim ki sigaraya başladınız. Öyleyse gelecekte bir vakit bu seçiminizle ilgili geribildirim almanız olasıdır. Bu örnekte, sağlık sorunları şeklinde olumsuz bir geribildirim alırsınız. Yalnız söz konusu geribildirimi sigaraya başlamaya karar verdikten uzun zaman (genelde onyıllar) sonra alırsınız. Bu kadar geriden gelen bir geribildirim sigara içmemeyi öğrenmenizi zorlaştırır. Sigara içenler sigaraya başladıktan *hemen* sonra (hayatlarını tehdit eden sağlık sorunları şeklinde) olumsuz geribildirim alsa, çok daha az insan sigara içiyor olurdu diye düşünülebilir.

Şimdiye kadar, bir programın *nasıl* öğreniyor olabileceği konusunda hiçbir şey söylemedik. Bir çalışma alanı olarak makine öğrenmesi YZ'nin kendisi kadar köklü bir geçmişe sahiptir ve bir o kadar da geniştir. Geçen altmış yılda pek çok teknik ortaya atıldı. Ama makine öğrenmesinin son başarıları esasen tek bir tekniğe dayanıyor: nöral ağlar. Aslına bakarsanız YZ'deki en eski tekniklerden biridir bu, John McCarthy'nin 1956 yaz okulu için sunduğu orijinal proje teklifinin bir parçasıdır, ama ancak bu yüzyılda yeniden kayda değer bir ilgi görmüştür.

Nöral ağlar kavramı, adından da anlaşılacağı üzere, **nöron** denen sinir hücrelerinin oluşturduğu ağlardan mülhemdir. Beynin ve sinir sisteminin mikro yapısında bulunan nöronlar birbirleriyle basit bir yoldan, beyindeki **akson** diye bildiğimiz lifler ve "bağlantı noktaları" olarak niteleyebileceğimiz **sinaps**lar yoluyla iletişim kurabilirler. Genellikle, bir nöron sinaptik bağlantıları yoluyla elektrokimyasal sinyaller alır, bu sinyallere bağlı olarak bir çıktı sinyali üretir ve diğer nöronlar da sinaptik bağlantıları yoluyla bu çıktı sinyalini alır. Burada kritik nokta, bir nöronun aldığı girdilerin farklı ağırlıklara sahip olmasıdır: Kimi girdiler diğerlerinden daha önemlidir, kimileri de nöronu ketleyerek bir çıktı üretmesini engelleyebilir. Hayvanların sinir sistemlerinde, nöron ağları ağırlıklı olarak birbiriyle bağlantılıdır: İnsan beyni kabaca 100 milyar nörondan meydana ge-

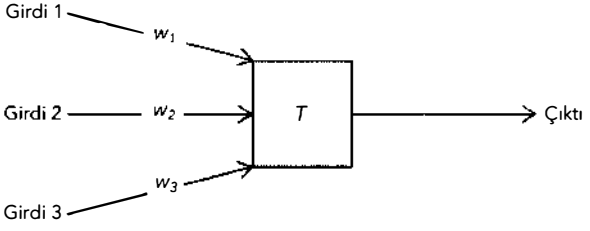
lir ve bunların her birinin genelde binlerce bağlantısı vardır.

Öyleyse makine öğrenmesinde nöral ağlar fikrinin bu tür yapıları bir bilgisayarda kullanma esasına dayandığı söylenebilir; nihayetinde insan beyni nöral yapıların gayet etkili bir biçimde öğrenebileceğinin kanıtıdır.

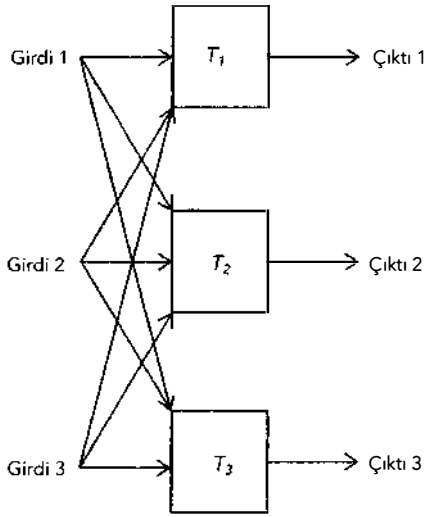
## Perseptronlar (Nöral Ağlar, Versiyon 1)

Nöral ağ incelemelerinin kökeni, ABD'li araştırmacılar Warren McCulloch ve Walter Pitts'in 1940'lardaki çalışmalarına kadar uzanır. Nöronların, elektrik devreleri –daha spesifik olarak, basit mantıksal devreler– biçiminde modellenebileceğini fark eden McCulloch ve Pitts, bu fikirden yola çıkarak dolambaçsız ama gayet genel bir matematiksel nöron modeli geliştirdi. 1950'lerde, Frank Rosenblatt bunu biraz daha geliştirerek **perseptron (algılayıcı) modeli** adını verdiği nöral ağ modelini ortaya koydu. Perseptron modeli bilfiil uygulamaya geçirilen ilk nöral ağ modeli olması bakımından önemlidir ve bugün hâlâ geçerliliğini korumaktadır.

Şekil 14 Rosenblatt'in perseptron modelini gösteriyor. Kareyi nöron olarak düşünelim; soldan kareye doğru uzanan oklara nöronun girdileri diyelim (sinaptik bağlantılara tekabül eder), kareden sağa doğru uzanan oka da nöronun çıktısı (aksona tekabül eder). Perseptron modelinde her girdi o girdinin **ağırlığı** olarak adlandırılan bir sayıyla ilişkilidir. Şekil 14'te Girdi 1 ile ilişkili ağırlık $w_1$, Girdi 2 ile ilişkili ağırlık $w_2$ ve Girdi 3 ile ilişkili ağırlık da $w_3$'tür.




**Şekil 14:** Rosenblatt'in perseptron modelinde tek bir nöral birim.




**Şekil 15:** Tek bir katman halinde düzenlenmiş üç yapay nörondan meydana gelen bir perseptron.

Nöronun her bir girdisi ya aktiftir ya da inaktif. Bir girdi aktifse, nöronu ağırlığı ölçüsünde "uyarır". Son olarak, her nöronun bir **aktivasyon eşiği** vardır, bu da başka bir sayıdır (Şekil 14'te $T$ ile gösterilmiştir). Nöron, aktivasyon eşiği $T$'yi aşan bir şiddetle uyarılırsa "tetiklenir", yani çıktısı aktif hale gelir. Başka bir deyişle, aktif olan bütün girdilerin ağırlıklarını topladığımızda $T$'ye denk veya $T$'den fazla oluyorsa, nöron bir çıktı üretir.

Bu fikri somutlaştırmak için, diyelim ki Şekil 14'te görülen nörondaki her ağırlık 1, eşik ise 2. Bu durumda nöron, girdilerinden herhangi ikisi aktifse tetiklenecektir. Başka bir deyişle nöron, girdilerinin belli bir *çoğunluğu* aktifse tetiklenir.

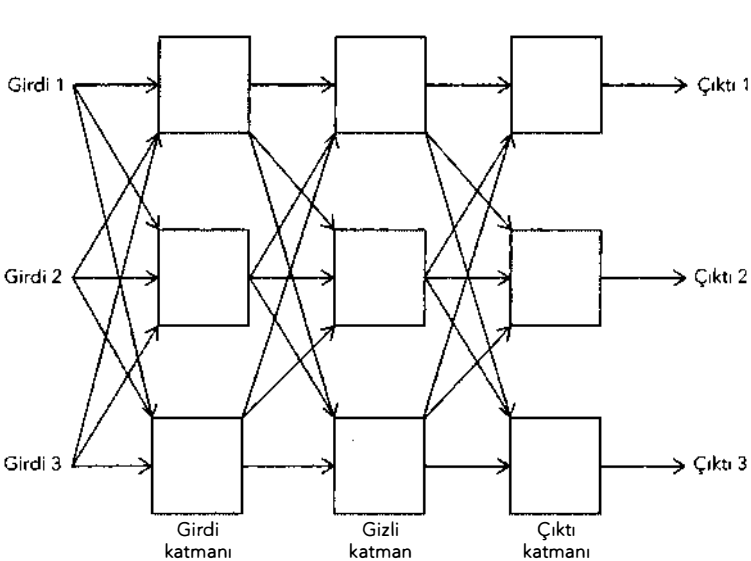
Buna karşılık, Girdi 1'in ağırlığı 2'yken, Girdi 2 ve Girdi 3'teki ağırlık 1 ve eşik yani $T$ de 2 diyelim. Bu durumda nöron ancak ya Girdi 1 aktifse, ya aynı anda Girdi 2 ve Girdi 3 aktifse ya da üç girdi birden aktifse tetiklenecektir.

Doğadaki nöral ağların pek çok nöronu varken, Şekil 15 sadece üç yapay nörondan ibaret bir perseptronu göstermektedir. Dikkat

ederseniz burada her nöron tamamen bağımsız olarak hareket etmektedir. Dahası her nöron her girdiyi "görebilir". Gelgelelim aynı girdinin ağırlığı nörondan nörona değişiklik gösterebilir. Dolayısıyla Girdi 1 ile ilişkili ağırlık, bir girdi sağladığı üç nöronun her biri için farklı olabilir. Ayrıca her nöronun aktivasyon eşiği (yani $T_1$, $T_2$, $T_3$ değerleri) de farklı olabilir. Bu nedenle söz konusu üç nöronun her birinin farklı bir hesaplama yaptığını düşünebiliriz.

Gelgelelim Şekil 15'teki düzenleme beynin sıkı sıkıya bağlantılı yapısı karşısında çok basit kalır çünkü beyinde tek bir nöronun çıktısı pek çok nöronu besler. İnsan beynindeki yapıların karmaşıklığını daha iyi yansıtmak için yapay nöral ağlar genelde, Şekil 16'da göreceğiniz üzere, **katman**lar halinde düzenlenmiştir. Buna **çok katmanlı perseptron** denir. Şekil 16'daki ağ 3'er nöronluk 3 katman halindeki 9 nörondan meydana gelir. Müteakip katmanlardaki bütün nöronlar bir önceki katmanın çıktılarının hepsini alır.

Hemen fark etmişsinizdir, böylesine küçük bir ağda bile işler karmaşıklaşmaya başladı: Şimdi elimizde toplam 27 tane nöronlar-




**Şekil 16:** Üç katman halinde dokuz nörondan oluşan bir perseptron.

arası bağlantı var, bunların her birinin ağırlığı ayrı ve 9 nöronun tek tek her birinin de ayrı bir aktivasyon eşiği var. Ta McCulloch ve Pitts'in çalışmalarında bile nöral ağlar birden çok katman halinde tahayyül edilmiş olsa da, Rosenblatt'in zamanında bütün gözler tek katmanlı ağların üstündeydi. Böyle olmasının çok basit bir nedeni vardı: Birden fazla katmanlı nöral ağların nasıl eğitileceğini hiç kimse bilmiyordu.

Her bağlantıyla ilişkili ağırlıklar bir nöral ağın işleyişi bakımından hayati önem taşır. Aslına bakarsanız bir nöral ağ nihayetinde sadece bunlardan ibarettir: bir dizi sayıdan. (Makul büyüklükte bir nöral ağ için *upuzun* bir sayı dizisi olacaktır.) Bu nedenle, bir nöral ağı eğitmek bir biçimde uygun sayısal ağırlıkları bulmayı gerektirir. Genel yaklaşım her eğitim seansından sonra ağırlığın düzeltilmesi yönündedir, ağın girdiler ile çıktıları doğru bir biçimde eşlemesi sağlanmaya çalışılır. Rosenblatt bunun için birkaç farklı teknik denemiş ve basit bir perseptron modeli için "hata düzeltme prosedürü"nü geliştirmiştir.

Bugün, Rosenblatt'in yaklaşımının kesin çalıştığını biliyoruz: Bir ağı daima doğru biçimde eğitecektir, ama çok büyük bir istisnayla. Marvin Minsky ve Seymour Papert 1969'da yayımlanan *Perceptrons*'ta[2] (Perseptronlar) tek katmanlı perseptron ağlarının yapabileceklerinin ne kadar sınırlı olduğunu gösteriyordu. Şekil 15'te gördüğümüz türden tek katmanlı perseptron modelleri bir hayli kısıtlıydı, girdiler ile çıktılar arasındaki pek çok basit ilişkiyi öğrenemiyorlardı. O dönem Minsky ve Papert'ın kitabını okuyanların en çok üstünde durduğu nokta, perseptron modellerinin "dışlayıcı VEYA" (*exclusive OR* ya da kısaca XOR) gibi son derece basit bir kavramı öğrenememesiydi. Bir örnek verecek olursak, düşünün ki ağınızın sadece iki girdisi var. Bu durumda XOR fonksiyonu, girdilerden tam olarak *bir tanesi* aktif olduğu zaman bir çıktı üretecektir (her ikisi de aktif olduğunda değil). Basit (tek katmanlı) perseptronların XOR'u yakalayamadığı kolaylıkla gösterilebilir. (Bu konu ilginizi çekerse, daha ayrıntılı bilgi için bkz. Ek D.)

Görünen o ki Minsky ve Papert'ın kitabından yola çıkılarak fazla genel sonuçlara varılmış, bunlar günümüze kadar hep tartışılmıştır.

*Birtakım perseptron sınıflarının* temel kısıtlarını ortaya koyan teorik sonuçlar *genel olarak* perseptron modelinin kısıtlarıymış gibi anlaşılmışa benziyor. Halbuki özellikle, Şekil 16'da gördüğümüz türden, *çok katmanlı* perseptronlar bu kısıtlara tabi *değildir*: Tamamen genel oldukları matematiksel bir kesinlikle gösterilebilir. Gelgelelim o dönemde çok katmanlı perseptronları olan bir ağın nasıl eğitileceği konusunda en ufak bir fikir bile yoktu, bunlar halihazırda işleyen sistemlerden ziyade teorik bir olasılıktan ibaretti. Bilimsel gelişmelerle beraber bunun mümkün hale gelmesi yirmi seneyi bulacaktı.

Perseptron modellerinin olumsuz bir tepkiyle karşılanması biraz da vaktiyle fazla beklenti yaratacak şekilde tanıtılmalarından kaynaklanmış olabilir mi diye düşünüyorum. Mesela 1958 tarihli bir *New York Times* makalesine göre nefesler tutulmuş şunların olması beklenmektedir:[3]

> ABD Donanması'nın bugün görücüye çıkardığı henüz tasarı aşamasındaki elektronik bilgisayarın ileride kendi kendine yürüyebilmesi, konuşabilmesi, görebilmesi, yazabilmesi, kendini yeniden üretebilmesi ve varoluşunun bilincinde olması bekleniyor.

Sebeplerin tam neler olduğu tartışılabilir, ama her ne olursa olsun, 1960'ların sonunda nöral ağ araştırmaları hızla azaldı; McCarthy, Minsky, Newell ve Simon'ın savunduğu simgesel YZ yaklaşımları öne çıktı (işin ironik tarafı, bu gerileme 2. Bölüm'de bahsettiğimiz YZ kışının başlangıcından sadece birkaç sene öncesine denk gelir). Rosenblatt 1971'de bir tekne kazasında hayatını kaybedince, nöral ağlar camiası da en önemli savunucularından birini kaybetmiş oldu. Rosenblatt kazadan sağ kurtulsa YZ tarihi farklı olurdu belki, ama ne yazık ki hayatını kaybetti ve nöral ağ araştırmaları on yıldan fazla yerinde saydı.

## Bağlantıcılık (Nöral Ağlar, Versiyon 2)

Bu kadar uzun süre yerinde sayan nöral ağlar alanı 1980'lerde yeniden canlanmaya başladı. Canlanmanın müjdecisi iki ciltlik *Parallel Distributed Processing*[4] kitabıydı (Paralel Dağıtımlı İşleme veya kısaca PDP). Merak uyandıran bir başlık seçilmiş. Paralel ve dağıtımlı

işleme bilgi işlemin anaakım alanlarından biridir, paralel işleyebilen bilgisayar sistemleri inşa etmekle ilgilenir. Yani kitabın adını görünce aklınıza ilk gelen YZ veya nöral ağlar değildir; kitabı sırf adına bakarak alan olduysa, aslında nöral ağlarla ilgili olduğunu fark edince herhalde bayağı şaşırmıştır. Başlık seçimi, yeni dalga ile önceki nöral ağ araştırmaları arasına mesafe koymaya yönelik bir girişimdir belki de.

Yeni külliyatın muhtemelen en önemli unsuru bir bakıma o kadar da yeni değildi: Basit perseptron sistemlerinin Minsky ve Papert'ın tespit ettiği kısıtlarını kolayca aştığı gösterilebilen çok katmanlı nöral ağlara odaklanıyordu. Ama önemli bir farkla. Perseptron hakkındaki önceki çalışmalar tek katmanlı ağlara odaklanıyordu çünkü o dönem çok katmanlı nöral ağların nasıl "eğitileceğini", nöronlararası bağlantıların "ağırlıklarının" nasıl bulunacağını kimse bilmiyordu. İşte PDP tam da bu probleme algoritma biçiminde bir çözüm sunuyordu: **geri yayılım**. Nöral ağlar alanındaki muhtemelen en önemli tekniktir bu.

Bilim dünyasında görmeye alışkın olduğumuz üzere, geri yayılım yıllar içinde tekrar tekrar icat edilmiş, ama PDP araştırmacılarının ortaya koyduğu özel yaklaşım sayesinde kesin olarak tesis edilmiştir.[5]

Ne yazık ki geri yayılımı usulünce açıklamak için üniversite seviyesinde kalkülüs gerekiyor ki bu da elinizdeki kitabın kapsamını misliyle aşıyor. Ama altında yatan fikir gayet basit. Geri yayılım bir nöral ağın sınıflandırmasında hata yaptığı yerlere bakarak çalışır. Bu hata kendini ağın çıktı katmanında gösterir. (Örneğin ağa gösterilen bir kedi fotoğrafı çıktı katmanında köpek fotoğrafı olarak sınıflandırılmış olabilir.) Geri yayılım algoritması hatayı ağda önceki katmanların her birinden geçirerek geriye doğru yayar (adı da buradan geliyor zaten). Bunun için de önce bir hata ortamı hesaplaması yapar: Her olası ağırlık için hatayı biliriz ve bu hata değeri de bir tür hata kontur haritası meydana getirir. Böylece mevcut hatadan en düşük hataya kadar aşama aşama en dik rotayı, yani hatayı çözmenin en hızlı yolunu bulabiliriz. Bu sürece **gradyan iniş** denir. Bir katman bitince bir öncekine geçilir ve bu böyle devam eder.

PDP ayrıca perseptron modellerinden çok daha genel nöron modelleri ortaya koymuştur. Örneğin perseptronlar esasen ikili (açık-veya-kapalı) hesaplama birimleridir, PDP ise çok daha genel modeller sunar.

Geri yayılımın geliştirilmesi ve PDP camiasının takdim ettiği diğer yenilikler, yirmi yıl önceki basit perseptron modeli ispatlarının fersah fersah ötesine geçen bir yığın nöral ağ uygulamasını mümkün kıldı. Nöral ağlara gösterilen ilgi tavan yaptı ama PDP furyası nihayetinde kısa ömürlü olacaktı. 1990'ların ortası, bir kere daha nöral ağ araştırmalarının pek rağbet görmediği bir dönem oldu. Bugünden geçmişe bakınca anlıyoruz ki PDP vagonunun tekerleri, temel fikirlerin bünyevi kısıtları nedeniyle değil çok daha sıradan bir nedenle yerinden çıkmıştı: Dönemin bilgisayarları yeni teknikleri kaldıracak kadar güçlü değildi. Ve PDP yerinde sayıyor gibi görünüyorken, makine öğrenmesinin diğer alanlarında hızla ilerleme kaydediliyor gibiydi. Makine öğrenmesine gösterilen ilgi bir kere daha nöral modellerden başka konulara yönelmişti.

## Derin Öğrenme (Nöral Ağlar, Versiyon 3)

Herhalde 2000 senesiydi, YZ alanında akademik bir kadroya kimin seçileceğini belirleyecek olan bir heyette görevliydim. Bir heyet üyesi, nöral ağlar konusunda çalışan adayları dikkate almamamız gerektiğine ikna etmeye çalışmıştı bizi. "Niş bir alan bu," diyordu, "neden gerileyen bir alanda çalışan birini alalım ki?" Bu sözlerine kulak asan olmadı, ama haksızlık da etmeyeyim şimdi, bir çalışma alanı olarak nöral ağların bir kere daha canlanacağını ta 2000 senesinden tahmin edebilmek herkesin harcı değildi. Ama işte 2006 civarında yeniden bir canlanma başladı ve YZ tarihindeki gelmiş geçmiş en kapsamlı, en çok reklamı yapılan büyümeye yol açtı.

Nöral ağ araştırmalarının bu üçüncü dalgasının altında yatan büyük fikir derin öğrenmeydi.[6] Derin araştırmayı karakterize eden anahtar fikir şudur demeyi çok isterdim, ama tek bir fikir değil birbiriyle ilişkili birçok fikir var ortada. Derin öğrenme üç farklı anlamda derindir.

Bunların belki de en önemlisi, isimden de tahmin edebileceğiniz üzere, *daha fazla katmanlı* olma fikridir. Her katman bir problemi farklı bir soyutlama düzeyinde işler. Girdi katmanına yakın katmanlar verilerdeki alt düzey kavramları (örneğin bir fotoğrafın köşeleri) ele alırken, ağın derinlerine doğru ilerledikçe daha soyut kavramların ele alındığını görürüz.

Derin öğrenme, daha fazla katmanlı olmanın yanı sıra, daha fazla nöronlu olmanın artılarından da faydalanabiliyordu. 1990'da tipik bir nöral ağ olsa olsa 100 nörona sahip olabiliyordu (buna karşılık insan beyninde yaklaşık 100 milyar nöron vardır). Dolayısıyla böyle ağların yapabileceklerinin son derece kısıtlı olmasının şaşılacak bir tarafı yoktu. 2016 yılında ise çok gelişmiş bir nöral ağda yaklaşık bir milyon (yani neredeyse bir arıdaki kadar) nöron olabiliyordu.[7]

Üçüncüsü, derin öğrenmenin kullandığı ağlar bizzat nöronların bir sürü bağlantısı olması anlamında derindir. 1980'lerde yüksek düzeyde bağlantılı bir nöral ağda her nöronun diğer nöronlarla 150 civarı bağlantısı olabiliyordu. Elinizdeki kitabın yazıldığı tarih itibarıyla, çok gelişmiş bir nöral ağdaki bir nöronda yaklaşık bir kedi beynindeki kadar çok bağlantı olabiliyor; buna karşılık bir insan nöronunda ortalama 10 bin civarında bağlantı vardır.

Demek ki derin nöral ağların daha fazla katmanı ve daha fazla, daha yüksek düzeyde bağlantılı nöronları oluyormuş. Böyle ağları eğitmek için geri yayılımın ötesinde tekniklere ihtiyaç vardı. Ve bunları sağlayan da 2006'da, adı derin öğrenme hareketiyle herkesten çok özdeşleştirilen, İngiliz-Kanadalı araştırmacı Geoff Hinton oldu. Hinton her açıdan önemli bir figürdür. 1980'lerde PDP hareketinin başını çekenlerden biriydi, ayrıca geri yayılımın mucitlerindendi. Şahsen, PDP araştırmalarının gözden düşmeye başladığı dönemde Hinton'ın hiç cesaretinin kırılmamış olmasını da önemli görüyorum. Bu işin peşini bırakmadı, sebat edip nöral ağ araştırmalarının derin öğrenme biçiminde bir kere daha canlandığına bizzat tanık oldu ve sonunda uluslararası çapta kabul gördü. (Bu arada, Hinton aynı zamanda 3. Bölüm'de modern mantığın kurucuları arasında adını andığımız George Boole'un torununun torunudur; gerçi bu, Hinton'ın vurguladığı gibi, mantık tabanlı YZ geleneği ile arasındaki

tek bağdır herhalde.)

Derin öğrenmenin başarısında etkili olan başlıca faktörlerden biri daha derin, büyük, yüksek düzeyde bağlantılı nöral ağlarsa, bir diğeri de Hinton ve daha nicelerinin nöral ağların eğitimine yönelik yeni teknikler konusundaki çalışmalarıydı. Ama bu başarı ancak *veri* ve *bilgisayar gücüyle* beraber mümkün olmuştu.

Verinin makine öğrenmesindeki önemini belki de en iyi anlatan örnek **ImageNet** projesinin hikâyesidir.[8] ImageNet'in fikir annesi Çin asıllı araştırmacı Fei Fei Li'ydi. 1976'da Pekin'de doğan Li 1980'lerde ailesiyle ABD'ye geldi, daha sonra burada fizik ve elektrik mühendisliği eğitimi gördü. 2009'da Stanford Üniversitesi'nin kadrosuna dahil oldu, 2013-2018'de YZ laboratuvarının direktörlüğünü yaptı. Li iyi korunan geniş veri kümelerinin, yeni sistemleri eğitmek, test etmek ve karşılaştırmak için ortak bir zemin sunması bakımından, derin öğrenme camiasının tamamına fayda sağlayacağını düşünüyordu. İşte bu amaçla ImageNet projesini başlattı.

ImageNet geniş bir çevrimiçi görsel arşividir; elinizdeki kitabın yazıldığı tarih itibarıyla sayısı 14 milyonu bulan görselden bahsediyoruz. Söz konusu görseller de JPEG gibi yaygın dijital formatlarda indirebileceğiniz fotoğraflardır. Burada asıl önemli nokta, görsellerin dikkatle 22 bin ayrı kategoride sınıflandırılmış olmasıdır. Sınıflandırmada çevimiçi bir eşanlamlılar sözlüğü olan WordNet[9] kullanılmıştır. Çok geniş bir söz dağarcığı olan WordNet'te sözcükler dikkatle sınıflandırılmış, sözgelimi hangi kelimeler eşanlamlı hangileri zıt anlamlı tek tek tespit edilmiştir vs.

Bugün ImageNet arşivine bakarken "yanardağ krateri" kategorisinde 1032 görsel, "frizbi" kategorisinde 122 görsel vb. olduğunu görüyorum. Burada altını çizmek istediğim iki nokta var: Veritabanındaki belli bir kategorideki görseller yapay değil ve aynı kategoriye dahil edilmelerinin nedeni birbirlerine benzemeleri değil. Bilakis, mesela "frizbi" kategorisine bakacak olursanız, bu görsellerin neredeyse tek ortak özelliğinin frizbi olduğunu görürsünüz: Kimi görsellerde frizbi oynayan insanlar var, kimilerinde ortalıkta kimse yok, sadece masada bir frizbi duruyor; hepsi birbirinden farklı, tek ortak özellikleri frizbi.

Görsel sınıflandırmasının "evreka!" ânı 2012'de yaşandı: Geoff Hinton ile çalışma arkadaşları Alex Krizhevsky ve Ilya Sutskever'in uluslararası bir görsel tanıma yarışmasında sunduğu nöral ağ sistemi AlexNet sergilediği performans bakımından çıtayı arşa taşıyacaktı.[10]

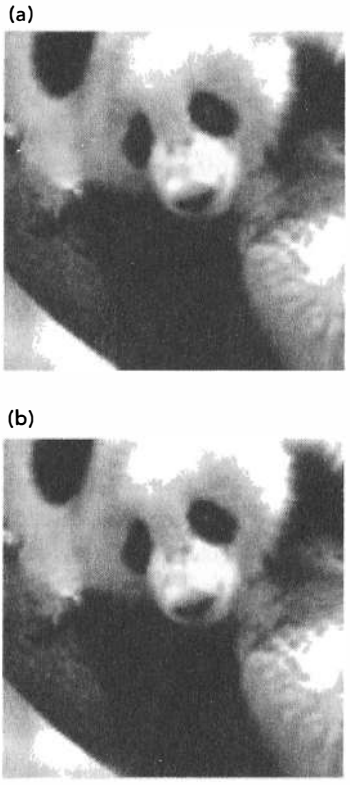
Derin öğrenmenin işlemesi için gerekli olan son faktör de bilgisayarların ham işlem gücüdür. Bir derin nöral ağı eğitmek için yüksek miktarda bilgisayar işlem gücü gerekir. Eğitimde yapılması gereken iş öyle karmaşık falan değildir ama miktar olarak çok çok fazladır. Bu yüzyılın başında yaygınlaşan yeni bir tür bilgisayar işlemcisi, bilgi işlemin ağır işlerinin rahatlıkla altından kalkabileceğini kanıtladı. Grafik İşlem Birimleri (Graphics Processing Unit, GPU) esasen bilgisayar oyunlarına yüksek kalitede animasyon sağlamak gibi bilgisayar grafik problemlerini çözmek için geliştirilmişti. Ama sonradan, nöral ağların eğitimi için de mükemmel oldukları görüldü. Bugün, derin öğrenme laboratuvarı adını hak eden her laboratuvarın bir GPU stoğu var – ama ne kadar büyük bir stokları olursa olsun, öğrencilerin yetersiz GPU şikâyetlerinin önüne geçemiyorlar.

Bütün bu aşikâr başarılarına rağmen, derin öğrenme ve nöral ağlar birtakım iyi bilinen dezavantajlardan da muzdariptir.

Birincisi, cisimleştirdikleri zekâ **opak**tır. Bir nöral ağın yakaladığı uzmanlık, nöronlararası bağlantılarla ilişkili sayısal ağırlıklarda cisimleşir ve henüz bu bilgiye ulaşmanın veya bu bilgiyi yorumlamanın bilinen bir yolu yoktur. Bir derin öğrenme programı, size röntgen taramasında kanser tümörleri gördüğünü söyleyebilir ama teşhisini gerekçelendiremez, bir banka müşterisinin kredi talebini reddedebilir ama neden reddettiğini açıklayamaz. 3. Bölüm'de MYCIN gibi uzman sistemlerin kabaca açıklama yapabildiğini, belli bir sonuca ulaşırken veya öneride bulunurken başvurulan akıl yürütme tarzının geriye doğru izini sürebildiğini görmüştük. Ancak nöral ağlar böyle kabaca bir açıklama bile sunamaz. Şu anda bu sorunu çözmek için pek çok çalışma yürütülüyor. Ancak nöral ağların cisimleştirdiği bilgi ve temsilleri nasıl yorumlayacağımız konusunda halihazırda en ufak bir fikrimiz bile yok.

Derin öğrenmenin –hemen göze çarpmasa da son derece önemli olan– problemlerinden bir diğeri de nöral ağların sağlam olmama-

sıdır. Örneğin görsellerde insanın fark edemeyeceği kadar ufak değişiklikler yapmanız nöral ağların bu görselleri tamamen yanlış sınıflandırmasına yol açabilir. Şekil 17'de buna bir örnek var.[11] Üstteki orijinal bir panda görseli, alttaki ise bu görselin üstünde oynanmış hali. İki görselin aynı gibi göründüğünü, özellikle de ikisinin de panda fotoğrafı gibi göründüğünü söylesem bir itirazınız olmaz herhalde. İşte bir nöral ağ, üsttekini doğru bir biçimde panda olarak sı-




**Şekil 17:** Panda mı gibon mu? Bir nöral ağ, (a) görselini doğru bir biçimde panda olarak sınıflandırıyor. Gelgelelim görselde bir insanın fark edemeyeceği oynamalar yapmak mümkün ve bu da aynı nöral ağın, üstünde oynanmış (b) görselini gibon olarak sınıflandırmasına yol açıyor.

nıflandırırken, alttaki görseli gibon kategorisine sokuyor. Bu tür konuların incelenmesine **çekişmeli makine öğrenmesi** deniyor. Bir rakibin, programı kasten yanıltmaya çalışması fikrinden esinle bu isim verilmiş.

Görsel sınıflandırma programımızın hayvan görselleri koleksiyonumuzu yanlış sınıflandırması dünyanın sonu değil elbette ama çekişmeli makine öğrenmesi her hatanın bu kadar masum olmayabileceğini gösteriyor. Örneğin trafik levhalarının görselleri benzer şekilde değiştirildiğinde, insanlar bunları yorumlamakta güçlük çekmezken, bir sürücüsüz otomobildeki nöral ağlar tamamen yanlış anlayabiliyor. Derin öğrenmeyi böyle hassas uygulamalarda kullanabilmemiz için önce bu tür problemleri iyice anlamamız gerek.

## DeepMind

Bu bölümün hemen başında sözünü ettiğim DeepMind'ın hikâyesi derin öğrenmenin yükselişini çok iyi özetliyor. Şirket 2010'da YZ araştırmacısı ve bilgisayar oyunu meraklısı Demis Hassabis ile okuldan arkadaşı olan müteşebbis Mustafa Suleyman ve Hassabis'in University College London'da çalışırken tanıştığı Shane Legg tarafından kuruldu.

2014 başında DeepMind'ı Google'ın aldığından bahsetmiştim; bu satın almayla ilgili haberlere denk geldiğimi hatırlıyorum da, DeepMind'ın *YZ* şirketi olduğunu öğrenince epey şaşırmıştım. O dönem YZ'nin yine revaçta olduğu aşikârdı ama Birleşik Krallık'ta (görünen o ki) 400 milyon sterlin eden *adını sanını duymadığım* bir YZ şirketi olması ağzımı bir karış açık bırakmıştı. Muhtemelen pek çok kişi gibi ben de ilk iş olarak DeepMind'ın web sitesine girdim, ama buradan pek bir bilgi edinemedim doğrusu. Şirketin teknolojisi, ürün veya hizmetleri hakkında pek bir malumat yoktu. Ama benim gibi meraklıların ağzına bir parmak bal çalmayı da ihmal etmemişlerdi: DeepMind'ın misyonu *zekâyı çözmek* olarak ilan ediliyordu. Son altmış yıldır YZ'nin bahtının bir açılıp bir kapanmasının bana alanda ilerlemeyle ilgili iddialı tahminlere temkinli yaklaşmayı öğrettiğini söylemiştim. Dolayısıyla dünyanın en büyük teknoloji devlerinden

birinin daha yeni satın aldığı bir şirketten böylesine cesur bir açıklama geldiğini görünce ne kadar hayret ettiğimi tahmin edersiniz.

Ne web sayfasında başka bir detay vardı ne de bundan fazlasını bilen bir tanıdığım. YZ'ciler duruma şüpheyle yaklaşıyordu ama bunda mesleki bir kıskançlığın da hafiften etkisi vardı muhtemelen. Aynı senenin ilerleyen zamanlarında meslektaşım Nando de Freitas'a denk gelince DeepMind hakkında biraz daha bilgi edinmiş oldum. Nando derin öğrenme alanında dünyanın en önde gelen isimlerindendir. O sıralar ikimiz de Oxford Üniversitesi'nde öğretim görevlisiydik (Nando sonradan ayrılıp DeepMind'la çalışmaya başladı). Öğrencileriyle beraber bir seminere gidiyordu, koltuğunun altında bir tomar bilimsel makale vardı. Bir şeylerin onu heyecanlandırdığı her halinden belliydi. "Londralı bir grup," dedi, "bir programı Atari video oyunları oynayacak şekilde eğitmiş, hem de oynamayı kendi kendine öğrenerek."

Çok da etkilendiğimi söyleyemem açıkçası. Video oyunu oynayan programlar nihayetinde yeni bir şey değildi. Lisans öğrencilerine proje diye vermiyor muyuz böyle şeyleri, demiştim Nando'ya, meselenin önemini kavrayamadığım için. O da tam olarak ne olup bittiğini sabırla açıklamıştı. İşte o zaman anlamıştım neden bu kadar heyecanlandığını. Ve YZ'de yeni bir çağa girmek üzere olduğumuz ancak o zaman dank etmişti kafama.

Nando'nun sözünü ettiği Atari sistemi 1980'lerin Atari 2600 oyun konsoluydu. İlk başarılı video oyunu platformlarından biriydi: 128'e kadar farklı renkle 210x160 piksel çözünürlüklü videoları destekliyordu; kullanıcı girdisi tek düğmeli bir *joystick* aracılığıyla sağlanıyordu. Platform, oyun yazılımı için takılıp çıkarılabilir kartuşlar kullanıyordu. DeepMind'ın kullandığı versiyonda toplam 49 oyun vardı. Amaçlarını şu şekilde açıklıyorlardı:[12]

> Amacımız, mümkün olduğu kadar çok oyun oynamayı öğrenmeyi başarabilecek bir nöral ağ ajanı yaratmak. Ağa belli bir oyun hakkında enformasyon verilmedi veya elle tasarlanmış görsel özellikler sağlanmadı, emülatörün dahili durumuna da aşina değildi; sadece video girdisinden, [skordan] ... ve bir grup olası eylemden öğrendi – tıpkı bu oyunu oynayan bir insanın yapacağı gibi.

DeepMind'ın başarısının önemini anlamak için, programlarının ne yapıp ne yapmadığını anlamak lazım. Burada belki de en önemli mesele *programa oynadığı oyunlar hakkında hiçbir şey anlatılmamış olmasıdır*. 3. Bölüm'de ele aldığımız türden bir bilgi tabanlı YZ sistemi kullanarak bir Atari oynama programı inşa etmeye çalışacak olsak, izleyeceğimiz yol, uzman bir Atari oyuncusundan bilgi çıkarmak ve bu bilgiyi kuralları veya başka bir bilgi temsil tertibini kullanarak kodlamaya gayret etmek olurdu muhtemelen. (Denemek isterseniz size şimdiden bol şanslar, ihtiyacınız olacak.) Ama DeepMind'ın programına oyun hakkında neredeyse *hiç* bilgi verilmemişti. Programa sağlanan tek enformasyon, konsolun ekranında belirecek olan görsel (210 x 160'lık bir renkli piksel ızgarası biçiminde) ve oyunun o anki skoruydu. Hepsi bu kadardı, programın elinde devam etmek için başka hiçbir şey yoktu. Burada bilhassa önemli olan nokta programa "*A* nesnesi $(x, y)$ konumunda" gibi bir enformasyon sağlanmamasıydı, böyle bir enformasyonu ham video verisinden bir biçimde *programın kendi kendine* çıkarması gerekiyordu.

Sonuçlar dikkate şayandı.[13] Program pekiştirmeli öğrenme yoluyla kendisine oynamayı öğretiyordu: Bir oyunu tekrar tekrar oynuyor, her oyunda türlü hamleler deneyip bunlar hakkında geribildirim alıyor, hangi eylemlerin ucunda ödül olduğunu hangilerinde olmadığını öğreniyordu. Program 49 oyundan 29'unu insan seviyesinin üstüne çıkarak oynamayı öğrenmişti. Kimi oyunlarda iyice insanüstü bir performansa ulaşmıştı.

Özellikle bir oyun çok ilgi gördü: Breakout. 1970'lerde geliştirilen ilk video oyunlarından biri olan Breakout'ta oyuncu bir "sopayı" sağa sola hareket ettirerek "topu" karşılamaya ve renkli "tuğlalardan" oluşan bir duvara doğru göndermeye çalışır. Top çarpınca tuğla yok olur, zaten oyunun amacı da mümkün olduğunca kısa sürede tuğlalardan meydana gelen duvarın tamamını yok etmektir. Şekil 18'de program, oyunu nasıl oynayacağını öğrenme safhasının başlarında (henüz yaklaşık yüz kere oynamışken) görülüyor. Bu aşamada program topu sık sık kaçırıyordu.

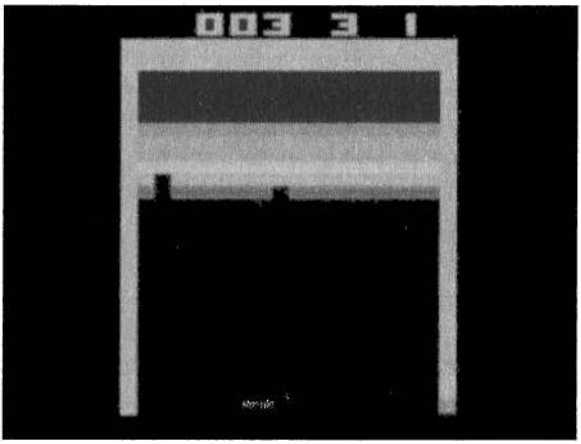
Program birkaç yüz antrenman daha yaptıktan sonra bu işin piri olup çıktı, artık topu *hiç* kaçırmıyordu. Derken müthiş bir şey oldu.

Program kısa yoldan çok puan toplamanın en randımanlı yolunun, duvarın yan tarafında bir delik açıp topu üst kısımdaki boşluğa göndermek olduğunu öğrendi; böylece top üst bariyer ile tuğla duvar arasında hızla seke seke daha çok tuğlayı yok ediyor, neredeyse oyuncunun başka hiçbir şey yapmasına gerek kalmıyordu (bkz. Şekil 19). DeepMind mühendislerinin beklediği bir davranış değildi bu, program böyle hareket etmeyi kendi kendine öğrenmişti. DeepMind'ın Breakout oynarken çekilmiş videosunu internette bulabilirsiniz. Bu videoyu farklı dönemlerde pek çok kere derslerimde kullandım. Ne zaman sınıfta göstersem, öğrencilerim programın yapmayı öğrendiği şeyin ne olduğunu anladıklarında izleyici kitlesinden muhakkak bir hayret nidası yükseliyordu.

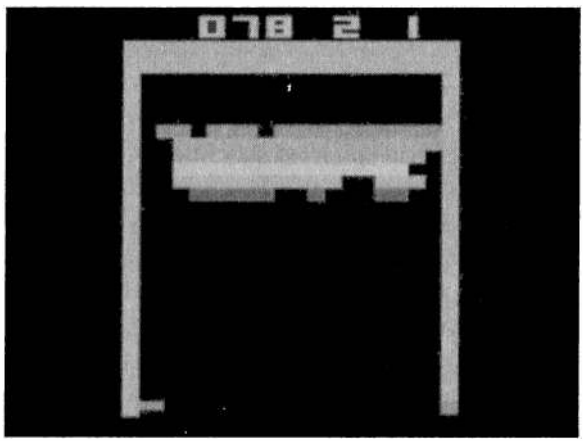
Bir kere daha altını çizelim: DeepMind belli bir Atari video oyununu oynayabilen bir program yazmış *değildi*, bunu yapmak kolay olurdu. Öyle bir program yazmışlardı ki 49 Atari oyunundan 29'unu bir insandan *daha iyi oynamayı öğrenmişti* ve öğrenirken de elinde video görüntüsü ve skordan başka *hiçbir* girdi yoktu.

Atari oyuncusunda kullanılan başlıca tekniğin pekiştirmeli öğrenme olduğundan bahsetmiştim, bu teknik üç gizli katmanı olan bir nöral ağ kullanılarak uygulamaya geçiriliyordu. Nöral ağın girdileri ön-işleniyor, görselin ham 210x160 formatı 84x84'e indiriliyor, renkli görüntünün yerini grinin tonları alıyordu. Program mevcut girdilerinden örnekler topluyor, üretilen tek tek her görüntüyü değil de her dört görüntüden birini kullanıyordu. Nöral ağ klasik derin öğrenme teknikleriyle ("stokastik gradyan iniş") eğitilmişti.

Program elbette mükemmel değildi, bazı oyunlarda performansı bayağı kötüydü. Neden böyle olduğuna baktığımızda gördüklerimiz dikkat çekici. Programın bilhassa zorlandığı oyunlardan biri Montezuma's Revenge idi. Oyuncu açısından bu oyunun zorluğu ödüllerin çok aralıklı olmasından ileri gelir, bir ödül ancak karmaşık sayılabilecek bir görevler dizisinin tamamlanmasıyla kazanılmaktadır (bu açıdan ödülün az çok doğrudan olduğu oyunlardan, örneğin Breakout'tan ayrılır). Daha genel olarak, ödülün ilgili eylemden uzun zaman sonra geldiği problemler pekiştirmeli öğrenme için çoğunlukla zordur. Esasen daha önce bahsettiğimiz belirleme proble-




**Şekil 18:** DeepMind'ın Atari oyuncusu Breakout'u yeni yeni öğrenmeye başlamışken topu kaçırıyor.




**Şekil 19:** Nihayetinde program kısa sürede yüksek bir skor elde etmenin en iyi yolunun, topu aynı yere çarptırmaya çalışarak duvarda bir delik açıp üst kısımdaki boşluğa göndermek, böylece topun üst bariyer ile duvar arasında sekerek kısa sürede daha çok tuğlayı yok etmesini sağlamak olduğunu öğreniyor. Önceden programa böyle bir şey öğretilmediği için, programı geliştirenler bu davranış karşısında hayretler içinde kalmıştı.

midir bu, yani aldığınız ödüle hangi eylemlerinizin yol açtığını bilme meselesidir.

Sadece Atari oynama programlarının başarısı bile DeepMind'a YZ tarihinde saygın bir yer kazandırmaya yeterdi. Ama şirket birbirinden etkileyici birçok başarıya daha imza attı.

Bunların en ünlüsü, bu satırları yazdığım tarih itibarıyla muhtemelen gelmiş geçmiş en meşhur YZ sistemi de olan **AlphaGo**'dur. Antik Çin menşeli masa oyunu Go'yu oynayan bir programdır bu.

Go YZ için ilgi çekici bir hedeftir. Bir taraftan, Go'nun kuralları *son derece* basittir, sözgelimi satrançtan hayli kolaydır. Diğer taraftan, 2015'te, Go oynayan programların performansı uzman insanların performansının oldukça gerisindeydi. Go niçin bu kadar zor bir hedefti öyleyse? Cevabı basit: Çünkü Go *büyük* bir oyundur. Go tahtası 19x19'dur, yani taşlar için tam 361 kare vardır; buna karşılık satranç tahtası 8x8'dir, yani sadece 64 kareden oluşur.

2. Bölüm'den hatırlayacağınız gibi, Go'nun dallanma faktörü (oyunun verili bir durumunda bir oyuncu için mevcut hamle sayısı ortalaması) yaklaşık 250'dir, satrancınki ise 35 civarındadır. Başka bir deyişle *Go oyun tahtası ve dallanma faktörü bakımından satrançtan çok daha geniştir.* Ayrıca Go oyunları uzun sürebilir, bir oyunda genelde 150 civarı hamle yapılır.

Oyunu insan oyuncular için zor hale getiren, büyüklüğüdür. Bu boyutta bir oyun tahtası çerçevesinde düşünmek insan oyuncuların sınırlarını zorlar, hatta aşar. Dolayısıyla Go'da *apaçık* bir strateji ortaya koymak son derece güçtür. Bu durum makineler için de sorun teşkil eder. Tahtanın boyutu ve dallanma faktörü basit arama tekniklerini ıskartaya çıkarır, başka bir şey lazımdır.

AlphaGo iki nöral ağ kullanır: Değer ağı sadece verili bir tahta pozisyonunun ne kadar iyi olduğunu kestirmeye çalışır, strateji ağı da mevcut bir tahta pozisyonuna bakarak nasıl hamle yapılabileceği konusunda önerilerde bulunur.[14] Strateji ağı 13 katmanlıdır. Önce insanların oynadığı uzman seviyesinde oyun örneklerinden oluşan eğitim verileriyle gözetimli öğrenme gerçekleştirilir, ardından da programın oyunu kendi kendine oynaması esasına dayanan pekiştirmeli öğrenme gelir. Son olarak, bu iki ağ Monte Carlo ağaç arama

denen sofistike bir arama tekniğine yerleştirilir.

Sistem duyurulmadan önce, DeepMind AlphaGo'ya karşı oynaması için Avrupa Go şampiyonu Fan Hui ile anlaştı: Sistem Fan'ı 5-0 yendi. Böylece ilk kez bir Go programı, baştan sona oynanan bir oyunda insan bir şampiyonu alt etmiş oldu. Çok geçmeden AlphaGo'nun Mart 2016'da Güney Kore'nin başkenti Seul'de beş maçlık bir yarışmada Lee Sedol ile karşı karşıya geleceği duyuruldu.

AlphaGo'nun bilimi büyüleyiciydi, bütün YZ araştırmacıları ne olacağını görmek için sabırsızlanıyorduk. (Bu arada, benim tahminim AlphaGo'nun olsa olsa bir-iki oyun alacağı, ama nihayetinde yarışmayı Lee'nin kazanacağı yönündeydi.) Öte yandan böyle eşi benzeri görülmedik derecede yoğun bir tanıtım yapılmasını hiçbirimiz beklemiyorduk. Yarışma dünyanın dört bir yanında manşetleri süslüyordu, hatta yarışmanın hikâyesinin uzun metraj bir filmini bile yaptılar.[15]

AlphaGo Lee Sedol'u 4-1 mağlup etti: Lee ilk üç maçı kaybetti, derken toparlanıp dördüncüyü aldı, ama beşinci maçta tekrar yenildi. Anlatılanlara bakılırsa, Lee kolay kazanırım diye düşünürken daha ilk oyundan kaybedince resmen şoka girmişti. O zamanlar bunun şakası çok yapılırdı, AlphaGo dördüncü oyunu Lee'nin gönlü olsun diye mahsus kaybetmiş denirdi.

Yarışmayı yorumlayan insanlar oyunların çeşitli noktalarında AlphaGo'nun enteresan görünen hamlelerine dikkat çektiler. "Bir insanın yapacağı türden hamleler değildi" bunlar. AlphaGo'nun nasıl oynadığını tahlil etmeye çalışırken, meseleye şüphesiz son derece insanmerkezci bir perspektiften yaklaşıyoruz; *biz* nasıl hamleler yapar, stratejiler geliştirirdik diye düşünüp oyunda bu tür hareketler görmeyi bekliyoruz. AlphaGo'yu bu şekilde anlamaya çalışmak abestir, çünkü tek bir şey yapmak, Go oynamak için optimize edilmiş bir programdan söz ediyoruz burada. Programa birtakım saikler, akıl yürütme tarzları, stratejiler atfetmek istiyoruz, halbuki programda bunların hiçbiri yok. AlphaGo'nun olağanüstü yetisi nöral ağların ağırlıklarında kayıtlıdır. Upuzun sayı listelerinden ibaret olan bu nöral ağların cisimleştirdiği uzmanlığı oradan çıkarmanın veya gerekçelendirmenin bildiğimiz bir yolu yoktur. AlphaGo bize hangi

hamleyi neden yaptığını açıklayamaz. İleride göreceğimiz gibi, derin öğrenmenin başlıca zorluklarından biri işte budur.

AlphaGo derin öğrenme ve büyük verinin yeni YZ'sinin bir zaferi olarak lanse edildi ki gerçekten öyleydi de. Ama meselenin birazcık derinine inerseniz, AlphaGo'daki akıllıca mühendisliğin çok ciddi bir bölümünün aslında klasik YZ aramasından ibaret olduğunu görürsünüz. 1950'lerde 2. Bölüm'de sözünü ettiğimiz satranç oynama programını geliştiren Arthur Samuel AlphaGo'yu görse, arama tekniklerine hiç yabancılık çekmezdi. Samuel'ın 1950'lerdeki çalışmaları ile modern çağın en ünlü YZ sistemi arasında kesintisiz bir bağ vardır.

Çığır açan bu iki başarıdan sadece on sekiz ay sonra, DeepMind yine manşetlerdeydi. Bu sefer AlphaGo'nun biraz daha genelleştirilmiş bir versiyonu olan AlphaGo Zero ile gündemdeydi. AlphaGo Zero'nun sıradışılığı insanüstü bir düzeyde oyun oynamayı hiç insan gözetimi olmaksızın, *sadece kendi kendine oynayarak* öğrenmiş olmasından geliyordu.[16] Doğruya doğru, böyle *çok fazla* oyun oynaması gerekmişti, yine de çarpıcı bir sonuçtu bu. Ardından, AlphaGo Zero'nun daha genelleştirilmiş versiyonu olan AlphaZero sistemi ortaya çıktı. AlphaZero satranç gibi bir dizi oyunu nasıl oynayacağını öğrenmiş, sözgelimi sadece dokuz saat kendi kendine oynamayla satrançta dünyanın en önde gelen programlarından Stockfish'i istikrarlı bir biçimde yenebilecek veya onunla berabere kalacak duruma gelmişti. Bilgisayar satrancı camiasının emektarları hayretler içindeydi. Kendi kendine dokuz saat oynayarak, AlphaZero'nun kendine bu oyunu dünya çapında bir satranç oyuncusu gibi oynamayı öğretmiş olması inanılır gibi değildi. Meselenin en heyecan verici tarafı buradaki yaklaşımın *genelliğiydi*: AlphaGo sadece Go oynayabiliyordu ve bu konuda ne kadar iyi olursa olsun sistemi özel olarak bu şekilde inşa etmek ciddi bir çaba gerektirmişti. AlphaZero ise birbirinden farklı pek çok masa oyununu oynayabilecek hale getirilebilir gibi görünüyordu.

Yalnız bu sonuçlara çok fazla anlam yüklememeye dikkat etmek lazım. Bir kere, genellik derecesi etkileyiciyse de (kendisinden önceki bütün emsallerinden çok daha genel bir uzman seviye oyun oy-

nama programıdır), AlphaZero genel zekâ yolunda başlı başına önemli bir ilerleme değildir. Biz insanların sahip olduğumuz zekâya dayalı yetilerin genellik düzeyinin yanına bile yaklaşamaz. Masa oyunlarını hepimizden iyi oynayabilir belki ama iki satır laf edemez, komik bir şaka yapamaz, omlet pişiremez, bisiklete binemez veya ayakkabı bağlayamaz. Olağanüstü yeteneklerinin kapsamı hâlâ çok dardır. Ve tabii masa oyunları çok soyuttur, Rodney Brooks'un hemen hatırlattığı üzere gerçek dünyadan çok uzaktırlar.

Ne var ki bütün bu itiraz ve çekinceler, DeepMind'ın Atari oyuncusundan başlayıp AlphaZero'ya kadar uzanan çalışmalarının YZ'de çığır açan bir dizi olağanüstü kazanım olduğu gerçeğini değiştirmiyor. Bütün bunlarla milyonların hayal gücünü yakalamayı başardılar.

## Genel YZ'ye Doğru?

Derin öğrenme, daha birkaç yıl öncesine kadar hayal bile edemeyeceğimiz programlar inşa etmemize olanak sağlayarak, muazzam bir başarı ortaya koymuş oldu. Ancak bütün bunlar, hakikaten takdire şayan olsalar da, YZ'yi sihirli bir dokunuşla o Büyük Hayal'e kavuşturacak değiller. Neden mi, açıklayayım. Bunun için ilk önce derin öğrenme tekniklerinin yaygın olarak kullanılan iki uygulamasına bakalım: görsel açıklaması ve otomatikleştirilmiş çeviri.

Bir görseli açıklama probleminde, bilgisayarın görseli alıp metinsel bir betimlemesini yapabilmesini isteriz. Bu yetiye bir ölçüde sahip sistemler artık yaygın olarak kullanılıyor. Sözgelimi Mac'imin son yazılım güncellemesiyle gelen bir fotoğraf yönetimi uygulaması gayet iyi iş çıkarıyor, fotoğraflarımı "Plajda", "Parti" vb. kategorilere ayırabiliyor. Elinizdeki kitabın yazıldığı tarih itibarıyla böyle pek çok web sitesi mevcut, genelde uluslararası araştırma gruplarının sayfaları; fotoğrafınızı yüklüyorsunuz, açıklamasını yazmaya çalışıyor. Mevcut görsel açıklaması teknolojisinin –ve dolayısıyla derin öğrenmenin– kısıtlarını daha iyi anlayalım diye, bunlardan birine (bu örnekte, Microsoft'un CaptionBot'una) bir aile büyüğümüzün fotoğrafını yükledim (bkz. Şekil 20).

CaptionBot'un cevabını incelemeye başlamadan önce, sizden ricam bu fotoğrafa şöyle bir bakmanız. İngiliz bilimkurgularına meraklıysanız, fotoğrafın sağındaki beyefendiyi tanımışsınızdır: Matt Smith, BBC'de yayınlanan *Doctor Who*'da 2010-13'te diziye adını veren başkahramanı canlandıran aktör. (Soldaki, tanımadığınız beyefendi ise eşimin merhum büyükbabası.)

CaptionBot'un fotoğrafa yaptığı yorum şöyle:

Sanırım Matt Smih ayakta fotoğraf için poz veriyor ve birlikte :-) :-) görünüyorlar.

CaptionBot fotoğraftaki başlıca unsuru doğru teşhis ediyor ve ortamı tanıma konusunda belli bir mesafe katediyor (ayakta, fotoğraf için poz veriyor, gülümsüyor). Ne var ki bu başarı CaptionBot'un kesinlikle yapmadığı bir şeyi yaptığını sanmamıza yol açabilir ve o yapmadığı şey de *anlamaktır*. Bunu bir örnekle biraz açalım. Matt Smith'i teşhis etmenin sistem için ne anlama geldiğini düşünün. Daha önce gördüğümüz üzere, CaptionBot gibi makine öğrenmesi sistemleri, her biri bir görsel ile bu görselin bir metinsel açıklamasından meydana gelen bir sürü örnekle eğitiliyor. Metin olarak "Matt Smith" açıklamasıyla birlikte, Matt Smith'li yeterince fotoğraf gösterilmiş olan sistem, artık ne zaman aktörün bir fotoğrafını görse doğru bir biçimde "Matt Smith" metnini üretebiliyor. Son derece kullanışlı bir yeti bu; sonsuz olası uygulamayla, onyıllar süren titiz araştırmaların doruk noktası.

Ancak CaptionBot'un yaptığı, bildiğimiz anlamda "tanımak" değildir. Bunu daha iyi anlamak için, diyelim ki *sizden* fotoğrafta gördüklerinizi yorumlamanızı istedim. Mesela şöyle bir yorum yapabilirdiniz:

*Doctor Who*'da oynayan aktör Matt Smith'i görüyorum; ayakta, kolunu yaşlı bir adamın omzuna atmış; diğer adamı çıkaramadım. İkisi de gülümsüyor. Matt'in üstündekiler Doktor'un kıyafetleri, yani muhtemelen bir yerlerde çekimdeler. Cebinde bir kâğıt tomarı var, senaryo herhalde. Elinde kâğıt bardak var, çay molası vermiş olabilirler. Arkalarındaki mavi kulübe Tardis, Doktor'un uzay gemisi/zaman makinesi. Dışarıdalar, yani Matt büyük ihtimalle dış mekân çekiminde. Yakınlarda bir ekip, kamera ve ışıklar olmalı.

**Şekil 20:** Fotoğrafta ne oluyor?

CaptionBot böyle yorumlar yapamıyordu. Matt Smith'i teşhis edebiliyor, yani doğru bir biçimde "Matt Smith" metnini üretebiliyordu ama bu bilgiyi *kullanarak* fotoğrafta neler olup bittiğini yorumlayıp anlayamıyordu. Burada mesele tam da budur, anlamanın yokluğu.

CaptionBot gibi sistemler, kendilerine gösterilen fotoğrafları yorumlama yetilerinin kısıtlı olmasının yanı sıra, başka bir bakımdan da bildiğimiz anlamda idrak sergilemezler, şu anki kapasiteleri buna elvermez.

Doktor'un kıyafetleri içindeki Matt Smith'in fotoğrafına bakınca, muhtemelen aklınızdan bin türlü düşünce geçer, duygudan duyguya koşarsınız; aktörü teşhis edip fotoğrafta neler olup bittiğini yorumlamanın ötesindedir bunlar. Mesela *Doctor Who* hayranıysanız, Matt Smith'in başrolde olduğu favori bölümünüzü (sizinki de "The Girl Who Waited"/"Bekleyen Kadın" mı?) tatlı tatlı anarken bulabilirsiniz kendinizi. Veya Doktor Who'yu anne babanız ya da çocuklarınızla izlediğiniz günlere gidebilirsiniz, dizideki yaratıklardan ödünüzün koptuğu aklınıza gelebilir vs. Veya vaktiyle bir vesileyle bir film setine gittiğinizi, bir film ekibi gördüğünüzü hatırlayabilirsiniz.

Sizin fotoğrafı böyle *anlamanızın* temelinde bir insan olarak dünyadaki deneyimleriniz vardır. CaptionBot açısından bu mümkün değildir çünkü onun böyle bir dayanağı yoktur (ve elbette böyle bir iddiası da yoktur). CaptionBot'un bu dünyada bir cismi yoktur, buna karşılık –Rodney Brooks'un hatırlattığı üzere– zekâ *cismanidir*. Burada bir kere daha YZ sistemleri idrak emaresi gösteremez gibi bir şey kastetmediğimin altını çizmek istiyorum. İdrak belli bir girdi (Matt Smith'in olduğu bir fotoğraf) ile belli bir çıktıyı ("Matt Smith" yazısı) eşlemek değildir. Böyle bir beceri idrakin ancak bir *parçası* olabilir, bütün mesele bundan ibaret değildir.

Derin öğrenmenin son on yılda hızlı bir ilerlemeye yol açtığı başka bir alan da bir dilden diğerine otomatikleştirilmiş çeviridir. Bu araçların ne yapıp ne yapamadığına bakmak, derin öğrenmenin kısıtlarını anlamamıza yardımcı olur. En bilinen otomatikleştirilmiş çeviri sistemi herhalde Google Çeviri'dir.[17] Bir hizmet olarak ilk kez 2006'da sunulan Google Çeviri'nin en yeni versiyonları derin öğrenmeyi ve nöral ağları kullanmaktadır. Sistem çok hacimli metin çevirileriyle eğitilmektedir.

Bu meseleyi daha iyi anlamak için, isterseniz şimdi de Google Çeviri'ye aşırı zor bir görev verip neler olacağına bakalım: Fransız yazar Marcel Proust'un erken yirminci yüzyıl klasiği *Kayıp Zamanın İzinde* romanının ilk paragrafını çevirmesini isteyelim. Paragrafın Fransızca orijinali şöyle:

*Longtemps, je me suis couché de bonne heure. Parfois, à peine ma bougie éteinte, mes yeux se fermaient si vite que je n'avais pas le temps de me dire: 'Je m'endors.' Et, une demi-heure après, la pensée qu'il était temps de chercher le sommeil m' éveillait; je voulais poser le volume que je croyais avoir encore dans les mains et souffler ma lumière; je n'avais pas cessé en dormant de faire des réflexions sur ce que je venais de lire, mais ces réflexions avaient pris un tour un peu particulier; il me semblait que j'étais moi- même ce dont parlait l'ouvrage: une église, un quatuor, la rivalité de François Ier et de Charles Quint.*

Söylemeye utanıyorum ama, birbirinden azimli bütün öğretmenlerimin olanca gayretine rağmen, çok az Fransızca biliyorum. Bu paragrafta sadece ara ara kimi ifadeleri seçebiliyorum, yoksa yardım

olmadan bütününe dair herhangi bir şey anladığım söylenemez.

Paragrafın Roza Hakmen'in usta kaleminden çıkan çevirisi şöyle:

Uzun zaman, geceleri erkenden yattım. Bazen, daha mumu söndürür söndürmez, gözlerim o kadar çabuk kapanıverirdi ki, "uykuya dalıyorum" diye düşünmeye zaman bulamazdım. Aradan yarım saat geçtikten sonra da, artık uykuya geçme vakti geldiği düşüncesiyle uyanırdım; hâlâ elimde zannettiğim kitabı bırakıp ışığımı [üfleyerek] söndürmek isterdim; az önce okuduklarım hakkında fikir yürütmeye, uyurken de devam ederdim, ama fikirlerim biraz farklı bir seyir izlerdi; kitapta sözü edilen şey, benmişim gibi gelirdi bana; bu bir kilise de olabilirdi, bir dörtlü de, I. François'yla Şarlken arasındaki rekabet de.[18]

İşte şimdi oldu! İşin ilginç tarafı, metin çok güzel bir Türkçeyle aksa da, anlam hâlâ apaçık değil – en azından benim için. Mesela yazar "kitapta sözü edilen şey, benmişim gibi gelirdi bana; bu bir kilise de olabilirdi, bir dörtlü de, I. François'yla Şarlken arasındaki rekabet de" derken tam olarak neyi kastediyor? İnsan nasıl bir kilise "olabilir"? Ne "dörtlüsü"? I. François'yla Şarlken arasındaki hangi "rekabet"? Odasını elektrikle aydınlatan bir okur için, ışığı "üfleyerek söndürmek" ne demek?

Şimdi de Google Çeviri'nin nasıl bir iş çıkardığına bakalım:

Uzun bir süre erken yattım. Bazen mum söner sönmez gözlerim o kadar hızlı kapanırdı ki kendi kendime "uyuuyorum" demeye vaktim olmazdı. Yarım saat sonra uyku arama zamanının geldiği düşüncesi beni uyandırırdı. ; Hâlâ elimde olduğunu sandığım sesi kısmak ve ışığımı söndürmek istedim; Az önce okuduklarımla ilgili düşüncelere dalmak için uyurken durmadım, ama bu düşünceler oldukça özel bir hal aldı; Bana kitabın konusu benmişim gibi geldi: bir kilise, bir dörtlü, Francis I ve Charles V arasındaki rekabet.*

Hakmen'in çevirisiyle arasındaki benzerlikler hemen göze çarpıyor, yani Google Çeviri iyi iş çıkarıyor. Ama kısıtlarını fark etmek için de illa çevirmen veya edebiyat alanında uzman olmak gerekmiyor elbette. "Uyku arama" tabirine bir anlam vermek zor, ardından gelen

* Yazım ve noktalama hataları aynen alınmıştır. –*ç.n.*

"elimde olduğunu sandığım sesi kısmak", "uyurken durmadım" vb. ifadelere de öyle; handiyse komik kaçan sözler bunlar. Çeviride, Türkçeye hâkim olan birinin asla kullanmayacağı türden ifadeler var. Genel olarak diyebiliriz ki belli belirsiz bir anlam çıkarsak da esasen akmayan, takır tukur bir çeviri bu.

Google Çeviri'ye çözmesi için verdiğimiz şüphesiz aşırı zor bir problem. Proust'u tercüme etmek iyi bir çevirmen için bile kolay iş değildir. Neden peki? Keza otomatikleştirilmiş çeviri araçları için neden bu kadar zor bir problem bu?

Çünkü Proust'un bu klasik romanını çevirmek için *Fransızcayı anlamaktan fazlası gerekir*. İsterseniz Fransızca okuma konusunda dünyadaki en yetkin kişi olun, yine de Proust karşısında bir afallarsınız, üstelik bunun sebebi yalnızca Proust'un üslubunun talepkâr olması değildir. Proust'u doğru dürüst anlamak, dolayısıyla doğru dürüst tercüme etmek için bir sürü arka plan bilgisine de sahip olmak lazım. Yirminci yüzyılın başlarında Fransız toplumu ve Fransızların gündelik hayatı hakkında bilgi sahibi olmak gerekir (o dönem aydınlatma için mum mu kullanılıyordu?), Fransız tarihi bilmek gerekir (I. François ile Şarlken arasındaki rekabetin muhteviyatı neydi?), erken yirminci yüzyıl Fransız edebiyatı (mesela dönemin üslubu, telmih sanatı) hakkında bilgi sahibi olmak gerekir, Proust'u tanımak gerekir (yazarın ilgi alanları nelerdi?) vb. Google Çeviri'de kullanılan türden nöral ağlarda bunların hiçbirisi yoktur.

Proust'un metnini *anlamak* için farklı türden *bilgiler* gerektiği fikri yeni değil. 3. Bölüm'de Cyc sisteminden bahsederken bile değinmiştik buna. Hatırlarsanız Cyc için koyulan hedef koca bir uzlaşımsal gerçekliğin tamamına tekabül eden bilginin sisteme girilmesiydi, böylece insan düzeyinde genel zekâya ulaşılabileceği varsayılıyordu. Bu noktada, bilgi tabanlı YZ araştırmacıları onyıllar evvel tam da bunu öngördüklerini özellikle belirtmemi isterlerdi muhtemelen. (Nöral ağlar camiası da buna karşılık, bilgi tabanlı YZ'cilerin teknikleri de pek bir işe yaramadı ama değil mi, diye sert bir cevap verirdi herhalde.) Ancak sadece derin öğrenme tekniklerini geliştirmeye devam etmenin bu sorunu çözeceği kesin değil. Derin öğrenme çözümün bir parçası olacak ama uygun bir çözüm için daha ge-

niş bir nöral ağ, daha fazla işlem gücü veya ağır Fransız romanları biçimindeki eğitim verilerinden çok daha fazlası lazımmış gibi geliyor bana. En az derin öğrenmenin kendisi kadar dramatik atılımların gerçekleştirilmesi icap edecek. Böyle atılımlar için, derin öğrenmenin yanı sıra, açık ve net bilgi temsilleri lazım olacak. Açıkça temsil edilen bilginin dünyası ile derin öğrenme ve nöral ağların dünyası arasında bir biçimde bir köprü kurmamız gerekecek.

## Büyük Bölünme

2010'da, Portekiz'in başkenti Lizbon'da gerçekleştirilmesi planlanan büyük bir uluslararası YZ buluşması olan Avrupa YZ Konferansı'nın (European Conference on AI, ECAI) programını organize etme teklifi aldım. Böyle konferanslara katılmak bir YZ araştırmacısının hayatının önemli bir parçasıdır. Araştırma sonuçlarımızı kaleme alıp değerlendirilmek üzere bir "program komitesi"ne sunarız. Önde gelen biliminsanlarından oluşan komite hangi başvuruların konferansta dinlenmeyi hak ettiğine karar verir. İyi YZ konferansları yaklaşık olarak her beş başvurudan yalnızca birini kabul eder, yani araştırmanızın kabul görmesi size belli bir itibar kazandırır; cidden büyük YZ konferanslarına beş binin üstünde başvuru yapıldığı olmaktadır. Dolayısıyla ECAI için gelen tekliften ne kadar onur duyduğumu hayal edebilirsiniz. Bilim camiasının size güvendiğinin bir işaretidir bu, ayrıca maaşınıza zam isterken özellikle belirtebileceğiniz türden bir artıdır.

Program başkanı olarak görevlerimden biri de program komitesini oluşturmaktı. Bu komitede makine öğrenmesi çevrelerinin de iyi bir biçimde temsil edilmesini istiyordum. Ama hiç beklenmedik bir şey oldu: Komite için teklif götürdüğüm makine öğrenmesi araştırmacılarının tek tek hepsinden hayır cevabı aldım. İnsanları böyle bir rolü üstlenmeye ikna etmeye çalıştığınızda kibarca reddedilmek alışılmadık bir durum değildir, çünkü nihayetinde külfetli bir iştir bu. Ama makine öğrenmesi camiasından *tek bir kişi bile* bu işe girmek istememişti. Ben mi bir yerde hata yapıyorum acaba diye düşündüm. Yoksa ECAI ile ilgili bir sorun mu vardı? Ya da başka bir

durum mu söz konusuydu?

Önceki yıllarda konferans organizasyonunda görev almış meslektaşlarımdan, ayrıca bu tür YZ etkinlikleri konusunda tecrübeli meslektaşlarımdan tavsiye istedim. Hemen hepsi az çok benzer şeyler yaşamıştı. Anlaşılan o ki makine öğrenmesi çevreleri benim "anaakım YZ" gözüyle baktığım konularla pek ilgilenmiyordu. Makine öğrenmesi camiasının iki büyük bilimsel etkinliği olduğunu biliyordum: Nöral Enformasyon İşleme Sistemleri (NeurIPS) Konferansı ve Uluslararası Makine Öğrenmesi Konferansı (ICML). YZ'nin birçok alt alanının kendi uzmanlık konferansı vardı, dolayısıyla bu çevrelerin dikkatinin söz konusu konferanslara odaklanmış olmasının şaşılacak tarafı yoktu. Açıkçası, makine öğrenmesi çevrelerinin kendilerini "YZ"nin bir parçası olarak görmediklerini hiç düşünmemiştim, ancak o zaman dank etmişti kafama.

Geriye dönüp bakınca, yapay zekâ-makine öğrenmesi bölünmesi çok erken bir tarihte başlamış gibi görünüyor: muhtemelen Minsky ve Papert'ın 1969 tarihli *Perceptrons*'uyla beraber (hatırlayacak olursanız, bu çalışmayla beraber 1960'ların sonunda nöral YZ araştırmalarında bir yavaşlama başlamıştı, ta ki 1980'lerin ortalarında paralel dağıtımlı işlemeyle beraber yeniden bir canlanma yaşanana kadar). Bu yayının yankıları yarım asır kadar sonra bugün bile hâlâ hissediliyor. Söz konusu bölünmenin kökenleri neye dayanıyor olursa olsun, bir noktada, makine öğrenmesi alanında çalışan pek çok araştırmacı anaakım YZ'den ayrılıp kendi yörüngesine girmiştir. Kuşkusuz pek çok araştırmacı makine öğrenmesi ile yapay zekâ arasında iyi bir denge tutturduğunu düşünüyorsa da, bugün pek çok makine öğrenmesi uzmanı çalışmalarının "YZ" etiketiyle nitelenmesine muhtemelen şaşıracak, hatta bundan rahatsız olacaktır. Çünkü onların gözünde YZ, benim bu kitapta bir dökümünü yaptığım, başarısızlıkla sonuçlanmış fikirlerin upuzun listesinden başka bir şey değildir.

ÜÇÜNCÜ KISIM

# Nereye Doğru Gidiyoruz?

# 6
# Günümüzde YZ

DERİN ÖĞRENME önünü açınca, YZ uygulamaları aldı başını yürüdü. Yirmi birinci yüzyılın ikinci yarısında YZ, 1990'ların World Wide Web'inden bu yana görülen yeni teknolojilerin hepsinden daha fazla ilgi çekti. Elinde biraz verisi ve çözülecek bir problemi olan herkes, acaba derin öğrenmeden yardım alabilir miyim diye sormaya başladı ve pek çok örnekte bu sorunun cevabı evet oldu. YZ hayatımızın her alanında varlığını hissettirmeye başladı. Teknolojinin kullanıldığı her yerde YZ kendisine uygulama alanı buluyor: eğitimde, bilimde, sanayide, ticarette, tarımda, sağlıkta, eğlencede, medyada, sanatta ve daha nice alanda. Gelecekte kimi YZ uygulamaları iyice görünür olacak, kimileri de olmayacak. Nasıl ki bilgisayarlar dünyamızın dört bir yanında iyice yerleşik hale geldiyse, YZ sistemleri de öyle olacak. Nasıl ki bilgisayarlar ve World Wide Web dünyamızı değiştirdiyse, YZ de değiştirecek. Nasıl ki bilgisayar uygulamalarının nerelere varabileceğini kestiremiyorsam, YZ uygulamalarının kapsamı da şu raddeye varacak diyemem, ama en azından son yıllardaki favorilerimden bahsedebilirim size.

Nisan 2019'da, insanlık olarak ilk defa bir kara delik fotoğrafı gördük hatırlarsanız.[1] Akıllara durgunluk veren bir deneydi bu: Gökbilimciler dünyanın dört bir yanından 8 radyo teleskobundan toplanan verileri kullanarak, 55 milyon ışık yılı uzaklıktaki 40 milyar km genişliğinde bir kara deliğin görüntüsünü inşa ettiler. Bu görüntü yüzyılımızın en çarpıcı bilimsel başarılarından biri. Yalnız bu kara

delik fotoğrafıyla ilgili olarak şunu *bilmiyor* olabilirsiniz: Fotoğraf ancak YZ sayesinde mümkün oldu. İleri bilgisayar görüntü algoritmalarıyla, resmin eksik unsurları "tahmin edilerek", kara deliğin görüntüsü yeniden inşa edildi.

2018'de, bilgisayar işlemcisi şirketi Nvidia'dan araştırmacılar, YZ yazılımının gayet ikna edici görünmekle beraber aslında baştan aşağı sahte insan resimleri yaratma yetisini gösterdi.[2] Yeni bir nöral ağ türü –çekişmeli üretici ağ– tarafından geliştirilen resimlerin tekinsiz bir tarafı vardı. Gözlerimiz bize bunların gerçek insanların fotoğrafları olduğunu söylüyordu ama aslında bir nöral ağ tarafından yaratılmışlardı. Daha yüzyılın başında hayal bile edilemeyecek olan bu yeti gelecekte sanal gerçeklik araçlarının başlıca bileşenlerinden biri haline gelecek: YZ alternatif gerçeklikler inşa etme yolunda emin adımlarla ilerliyor.

2018'in sonlarında, DeepMind araştırmacıları Meksika'da düzenlenen bir konferansta AlphaFold'u duyurdular. Protein katlanması denen temel bir meseleyi anlamaya yönelik bir sistemdi bu.[3] Protein katlanması birtakım moleküllerin hangi şekli alacağını tahmin etmeyi de gerektirir. Bu da Alzheimer gibi hastalıkların tedavisinde ilerleme sağlamak bakımından kritik önemdedir. Ama ne yazık ki korkutucu derecede zor bir problemdir bu. Dolayısıyla, proteinlerin şekillerini tahmin etmeyi öğrenmek için klasik makine öğrenmesi tekniklerini kullanan AlphaFold, böylesine harap edici hastalıkları anlama yolunda atılmış umut vadeden bir adımdır.

Bu bölümün kalanında, YZ'nin sağladığı imkânlardan en öne çıkan ikisini biraz daha ayrıntılı ele almak istiyorum: Bunların ilki YZ'nin sağlık alanında kullanımı, ikincisiyse uzun yıllardır hayali kurulan sürücüsüz otomobiller.

## YZ Destekli Sağlık Hizmetleri

Radyolog yetiştirmeyi bırakın artık. Önümüzdeki beş yıl içinde derin öğrenmenin radyologlardan daha iyi iş çıkaracak hale geleceği ortada.

GEOFF HINTON (2016)

Cardiogram size özel bir sağlık asistanı inşa ediyor. Takılabilir cihazınızı sağlığınızı sürekli izleyen bir araca, uyku ve antrenman takibinin ötesinde, belki de günün birinde bir kalp krizini önleyip hayatınızı kurtaracak bir araca dönüştürmek istiyoruz.

Cardiogram şirketinin web sayfası[4]

Siyaset ve ekonomiyle ucundan kıyısından bile olsa ilgilenen herkes bilir ki yurttaşlar ve hükümetler için en önemli küresel finansal problemlerden biri sağlık hizmetlerinin sağlanmasıdır. Bir taraftan, son iki yüzyılda sağlık hizmetlerinin sağlanmasında kaydedilen gelişmeler muhtemelen sanayileşmiş dünyada bilimsel yöntemin başlı başına en önemli başarısıdır: 1800'lerde Avrupa'da ortalama insan ömrü 50 seneyi bulmuyordu,[5] buna karşılık bugün Avrupa'da doğan biri 70'li yaşlarının sonunu rahatlıkla görebileceğini düşünebilir. Ayrıca gelişmiş dünyada doğum sırasında anne ölüm oranı da son derece düşük artık. Bu çarpıcı değişiklikler bir ölçüde hijyenin öneminin daha iyi anlaşılmasından kaynaklanıyor. Ama hastalıklar ve diğer rahatsızlıklar için ilaçların ve tedavi yöntemlerinin geliştirilmesi de bir o kadar belirleyici tabii. Sözgelimi bunların açık ara en önemlisi olan 1940'larda antibiyotiklerin geliştirilmesiyle beraber, ilk defa bakteriyel enfeksiyonların etkili ve güvenilir bir biçimde tedavi edilmesi sağlandı. Gelgelelim sağlık hizmetleri ve ortalama insan ömrü konusunda sağlanan bu gelişmeler henüz dünyanın dört bir yanına ulaşmış değil elbette. Bu kitabın yazıldığı tarih itibarıyla Orta Afrika Cumhuriyeti'nde ortalama insan ömrü sadece 51 sene, ayrıca dünyanın pek çok yerinde doğum hem anne hem de çocuk için tehlikeli olmaktan çıkmış değil hâlâ. Ama genel trend olumlu görünüyor ve bu da elbette memnuniyet verici.

Ne var ki bu memnuniyet verici ilerlemeler birtakım zorlukları da beraberinde getiriyor. Birincisi, yaşlı nüfus ortalaması giderek artıyor. Daha ileri yaşlı insanlar haliyle gençlere kıyasla daha fazla sağlık hizmetine gereksinim duyduğundan, sağlık hizmetlerinin toplam maliyeti artıyor. İkincisi, hastalıklar ve diğer rahatsızlıklar için yeni ilaçlar ve tedavi yöntemleri geliştirildikçe, tedavi edebildiğimiz hastalıklar yelpazesi giderek genişliyor ki bu da sağlık hizmetlerinde ekstra bir maliyete yol açıyor. Bunların yanı sıra sağlık hizmetlerini sağlamak için gereken kaynakların pahalı olması ve kalifiye sağlık personelinin kıt olması da bu hizmetlerin maliyetinde belirleyici bir rol oynuyor elbette (sözgelimi Birleşik Krallık'ta pratisyen hekim olabilmek için yaklaşık on sene eğitim görmeniz gerekiyor).

İşte bu problemler nedeniyle sağlık hizmetleri –bilhassa da bu hizmetlerin *finansmanı*– dünyanın her yerinde politikacıların ağızlarına sakız ettikleri bir meseledir. Birleşik Krallık'ta, 1940'ların sonunda, herkese ihtiyacı olduğu anda ücretsiz sağlık hizmeti sunma amacıyla, maliyeti vergiler yoluyla karşılanan ulusal bir hizmet olarak yürürlüğe giren Ulusal Sağlık Hizmeti'nin (National Health Service, NHS) en iyi nasıl finanse edilebileceğiyle ilgili tartışmaların bir türlü sonu gelmez. Biz Britanyalılar NHS'yi severiz ama en iyi nasıl finanse edilebileceği meselesini de tartışa tartışa bitiremeyiz.

Uzun lafın kısası sağlık hizmeti elzem ama sağlaması zor bir hizmettir. O zaman, bu problemi *teknolojiyle* çözmenin bir yolu bulunsa harika olmaz mıydı?

Sağlık hizmetlerinde YZ fikri yeni değil. 3. Bölüm'de, alanında çığır açan MYCIN adlı uzman sistemin, insanlarda kan hastalıklarının nedenlerini teşhis etmede insanlardan daha iyi performans sergileyebildiğinin gösterilmesiyle, geniş çapta kabul gördüğünden bahsetmiştik. Sağlık hizmetlerinin finansmanı bugün olduğu gibi 1980'lerin başında da bir sorun teşkil ediyordu. Dolayısıyla, burada bahsettiğim nedenlerden ötürü, insan sağlık çalışanlarının yetilerini yakalayabilen programlar fikri müthiş bir heyecan yarattı. Yani MYCIN'in hemen ardından sağlık hizmetleriyle ilgili bir benzer uzman sistemler dalgasının gelmesi gayet doğaldı, gerçi bunların nis-

peten büyük bir çoğunluğunun araştırma laboratuvarlarından pek uzaklaşamadığını da teslim etmemiz lazım. Sağlık hizmetlerinde YZ'ye duyulan ilgi bu sıralar tekrar geri dönmüş durumda. Üstelik bu kez kaydedilen gelişmeler geniş çapta başarıya ulaşma şansının daha yüksek olduğunu gösteriyor.

YZ destekli sağlık hizmetleri alanındaki önemli yeni imkânlardan biri *kişisel sağlık yönetimi*'dir. Bu, *takılabilir teknoloji*'nin, sözgelimi Apple Watch gibi akıllı saatler ve Fitbit gibi aktivite/antrenman takibi cihazlarının gelişiyle birlikte mümkün olmuştur. Söz konusu cihazlar insan fizyolojisinin kalp atış hızı ve vücut sıcaklığı gibi özelliklerini sürekli takip eder. Bu da çok sayıda insandan mevcut sağlık durumuyla ilgili olarak sürekli bir veri akışı sağlamak gibi büyüleyici bir ihtimali gündeme getirir. YZ sistemleri bu veri akışlarını ya lokal olarak (cebinizde taşıdığınız akıllı telefon aracılığıyla) ya da internetteki bir YZ sistemine yükleyerek analiz eder.

Bu teknolojinin potansiyelini hafife almamak lazım. *Şimdiye kadar ilk kez, sağlık durumumuz sürekli takip edilebiliyor.* En temel düzeyde, YZ destekli sağlık hizmetleri sistemlerimiz sağlığımızı yönetmede bize objektif tavsiyelerde bulunabilir. Fitbit gibi cihazların halihazırda yaptığı da bir bakıma budur zaten; aktivitelerimizi takip eder, ayrıca bize hedefler koyarlar ("günde 10.000 adım hedefi" bunun gayet aşikâr, son derece başarılı örneklerinden biri). Deneyimler, oyunlaştırma, yani hedefleri birer yarış veya oyuna dönüştürme –hatta sosyal medyayı da bu işe dahil etme– yoluyla insanların bu tür hedeflere daha fazla riayet etmesinin sağlanabileceğini gösteriyor.

Kitlesel üretilen takılabilir teknoloji henüz emekleme aşamasında ama gelecekte bizi neyin beklediğine dair pek çok alamet var. Eylül 2018'de, Apple Watch'un dördüncü jenerasyonu tanıtıldı. Kalp atış hızını takip eden ilk Apple Watch bu. Elektrokardiyografi (EKG) uygulamaları, kalp atış hızı takibiyle elde edilen verileri izleyebiliyor ve kalp hastalıklarının semptomlarını teşhis etme potansiyeline sahip, hatta gerektiğinde ambulans çağırabiliyor. Doğrudan uygulamalarından biri, inmenin veya dolaşımla ilgili başka bir acil durumun habercisi olabilen atriyal fibrilasyonun, yani kalp atışla-

rındaki düzensizliğin pek de kolay tespit edilemeyen belirtilerini yakalamak için izleme yapmak. Telefondaki ivmeölçer sayesinde birinin düştüğü anlaşılabiliyor, gerektiğinde acil sağlık hizmetlerini arama özelliği var. Böyle sistemler için gereken YZ teknikleri aslında son derece basit. Zaten bugün bu sistemlerin uygulamaya geçirilebilmesini sağlayan da aslında sürekli internete bağlı, bir dizi fizyolojik sensörle donatılmış bir takılabilir cihaza bağlanabilen güçlü bir bilgisayarı yanımızda taşıyabilir hale gelmemiz oldu.

Kimi kişisel sağlık uygulamaları için sensör bile gerekmiyor, standart bir akıllı telefon yetiyor. Oxford Üniversitesi'nden bazı çalışma arkadaşlarım *sadece bir insanın akıllı telefonunu kullanma tarzına bakarak* demans başlangıcını tespit etmenin mümkün olabileceği görüşünde. Telefonun kaydettiği kullanma tarzındaki veya davranış örüntülerindeki değişikliklerin, başka birisi bu belirtileri fark etmeden önce ve resmi bir teşhis koyulmadan çok daha önce, hastalığın başladığının fark edilmesini sağlayabileceği düşünülüyor. Demans yıkıcı bir hastalık, yaşlı nüfusu giderek artan toplumlar açısından ciddi bir sıkıntı teşkil ediyor. Erken teşhisine veya yönetimine yardımcı olabilecek araçlar büyük bir memnuniyet yaratacaktır. Yolun daha çok başında olsalar da, bu tür çalışmalar gelecekte bizi nelerin beklediğinin başka bir göstergesi aslında.

Hepsi gerçekten de heyecan verici gelişmeler ama birtakım potansiyel tuzakları da beraberlerinde getiriyorlar. Bunların içinde en bariz olanı *mahremiyetle* ilgili: Takılabilir teknoloji özel alanımıza giriyor, bizi sürekli izliyor; elde edilen veriler, bizim iyiliğimize kullanılabileceği gibi, istismara da son derece açık.

Sigorta sektöründen bununla doğrudan ilgili olarak kaygı uyandırıcı bir örnek verebiliriz mesela. 2016'da, Vitality şirketi sağlık sigortası yaptığı müşterilerine Apple Watch vermeye başladı. Akıllı saatler hareketlilik düzeyinizi takip ediyor, ödeyeceğiniz prim miktarı da ne kadar spor yaptığınıza göre belirleniyor. Bu ay miskinlikten hiç egzersiz yapmadınız mı, yüksek prim ödüyorsunuz; telafi etmek için sonraki ay deli gibi spor mu yaptınız, hoop priminiz düşüyor. Bu işleyişin doğrudan yanlış bir tarafı yok belki ama daha tedirgin edici birtakım ihtimalleri akla getiriyor. Örneğin 2018'in

Eylül ayında ABD merkezli sigorta şirketi John Hancock uzun vadede *sadece* hareketlilik düzeyini takip eden teknolojik araçları kullanmayı kabul edenleri sigortalayacağını duyurdu.[6] Ve bu da çok eleştirildi tabii.

Bu senaryoyu bir adım daha ileri taşıyalım: Yurttaşlar olarak devletin sağladığı sağlık hizmetlerinden vb. faydalanmamız için, takip edilmeyi ve günlük belli bir egzersiz hedefine ulaşmayı kabul etme şartı getirilirse ne olacak peki? Sağlık hizmetlerinden yararlanmak mı istiyorsunuz, o zaman her gün 10.000 adım atacaksınız. İyi de ne var ki bunda diye düşünenler olduğu gibi, bu durumu müthiş bir müdahale, insan olarak en temel haklarımızın istismarı olarak görenler de var.

YZ'nin sağlık hizmetleri alanında potansiyel uygulamalarından yine heyecan verici olan bir diğeri de otomatikleştirilmiş teşhistir. Makine öğrenmesinden yararlanarak röntgen ve ultrason gibi tıbbi görüntüleme cihazlarından elde edilen verileri analiz etme konusu son on yıldır büyük ilgi görüyor. Bu kitabın yazıldığı tarih itibarıyla, tıbbi görüntülemeyle elde edilen sonuçlardaki anomalilerin YZ sistemleri tarafından tekrar tekrar etkili bir biçimde teşhis edilebildiğini gösteren bir çalışmanın haberi geldi. Makine öğrenmesinin klasik bir uygulaması bu; normal ve anormal görüntü örneklerini göstererek eğittiğimiz program, bir yerden sonra görüntülerdeki anomalileri teşhis etmeyi öğreniyor.

Bu çalışmanın iyi bilinen örneklerinden biri DeepMind'dan geldi. 2018'de, şirket Londra'daki Moorfields Göz Hastanesi'yle birlikte, göz filmlerinden yola çıkarak hastalık ve anomalileri otomatik olarak teşhis etme teknikleri geliştirmeye yönelik bir çalışma yürüttüğünü duyurdu.[7] Hafta içi her gün ortalama bin göz filminin çekildiği Moorfields'de, bu filmlerin analizi hastane faaliyetlerinin ciddi bir bölümünü oluşturuyor.

DeepMind'ın sistemi iki nöral ağ kullanıyordu; ilki filmi farklı "segmentlere ayırmak" (farklı bileşenlerini tespit etmek), ikincisi de teşhis için. İlk ağın eğitimi için yaklaşık 900 film kullanıldı, bunlar bir insan uzmanın filmi nasıl segmentlere ayırdığını gösteriyordu; ikinci ağın eğitimi için de yaklaşık 15 bin örnek kullanıldı. Ya-

pılan denemeler, sistemin insan uzmanlar düzeyinde veya daha iyi bir performans sergilediğini gösteriyordu.

Müthiş sonuçlar bunlar. Mevcut YZ tekniklerinin benzer kapasitelere sahip sistemler inşa etmek için nasıl kullanıldığına dair birbirinden çarpıcı daha nice örnek verebiliriz: röntgen filmlerindeki kanser tümörlerinin tespiti, ultrason filmleriyle kalp hastalıklarının teşhisi vb. Daha önce son derece başarılı bir görsel tanıma programı olan AlexNet'in yaratıcılarından biri olarak bahsettiğimiz Geoff Hinton, makine öğrenmesinin tıbbi görüntüleme sonuçlarına dayanarak teşhis koyma problemine bir çözüm sunacağından o kadar emindi ki, radyologlar hakkında bu alt bölümün başına aldığımız o cüretkâr lafları etmişti. Bu sözler radyologları çok kızdırdı tabii. Üstüne basa basa, radyolog olmanın röntgen filmine bakmaktan çok daha fazlasını gerektirdiğini söylediler.[8]

YZ'nin sağlık hizmetlerinde kullanımı için bir kısım böyle bastırırken, bir kısım da ihtiyatlılığa davet ediyor. Bir kere sağlık hizmetleri her şeyden öte bir *insan* mesleğidir. İnsanlarla etkileşime girme, iletişim kurma yetisini belki de diğer bütün mesleklerden daha çok gerektirir. Bir pratisyen hekim hastalarını "okuyabilmelidir", hastayı hangi sosyal bağlamda gördüğünü iyi anlayabilmelidir, belli bir hastada hangi tedavinin işe yarayıp hangisinin işe yaramayacağını kestirebilmelidir, vb. Tıbbi verilerin analizi bakımından insan uzmanlar düzeyinde performans sergileyebilen sistemleri artık inşa edebildiğimiz anlaşılıyor, ancak insan sağlık çalışanlarının yaptıklarının (olağanüstü derecede önemli olmakla beraber) sadece küçük bir kısmını oluşturuyor bu.

YZ'nin sağlık hizmetleri alanında kullanılmasına karşı bir başka argüman, kimilerinin bir insanın muhakemesine bir makineninkinden daha çok güvendiği yolunda. Bu kişiler karşılarında etten kemikten bir insan görmek istiyorlar. Yalnız tam da burada iki noktayı biraz daha açmak gerekiyor.

Birincisi, insan muhakemesini en ideal ölçüt saymak biraz fazla naiflik olur. Bizler, hepimiz, kusurları olan varlıklarız. En deneyimli, en çalışkan doktorun bile kimi zaman yorgun düştüğü, duygularına yenildiği olur. Ne yaparsak yapalım, hepimiz bir biçimde ön-

yargıların pençesine düşüp yanlılığa meylederiz; çoğu zaman, rasyonel kararlar vermede iyi değilizdir işte. Makineler her açıdan insan uzmanlarınki kadar güvenilir teşhislerde bulunmaya *muktedirdir*. Sağlık hizmetleri alanındaki zorluk/imkân bu yetinin en iyi şekilde kullanılmasıyla ilgilidir. Açıkçası bunun da, insan sağlık çalışanlarının yerini YZ'nin almasıyla değil, bu insanların mesleki kapasitelerinin YZ'yle *desteklenip güçlendirilmesiyle* mümkün olduğunu düşünüyorum. YZ sayesinde rutin işlerinden kurtulan insan sağlık çalışanları mesleklerinin gerçekten zor taraflarına odaklanabilir, YZ'nin muhakemesi insan uzmana kendi görüşünü dengeleyecek yeni bir perspektif sunabilir veya uzmanın çalışmaları için daha geniş bir bağlam sağlayabilir.

İkincisi, insan sağlık çalışanından mı yardım alsam yoksa YZ sağlık hizmetleri programından mı diye bir tercih yapma imkânına sahip olmak pek çokları için lüks kaçabilir. Bugün dünyanın pek çok yerinde sağlık hizmetlerine *hiç* erişimi olmayan milyonlarca insan var; bu insanlar için YZ sistemleri bir seçenek olabilir, çeşitli imkânlar sunabilir. Bunu hakikaten müthiş bir olasılık olarak görüyorum, YZ'nin bize sunduğu bütün imkânlar içinde toplumsal açıdan en büyük etkiyi yaratabilecek olasılık olarak.

## Sürücüsüz Otomobiller

Havadan ağır araçların uçması mümkün değil.

LORD KELVIN, Kraliyet Cemiyeti Başkanı, 1895

Bu kitabın yazıldığı tarih itibarıyla, otomobil kazalarında dünya çapında her yıl bir milyondan fazla insan hayatını kaybediyor. Tek başına Çin ve Hindistan'da meydana gelen kazalar bu sayının neredeyse dörtte birini oluşturuyor. Ayrıca her yıl 50 milyon kişi araba kazalarında yaralanıyor. Gerçekten sarsıcı rakamlar bunlar; böyle bir zayiata sözgelimi yeni bir virüs yol açsaydı dünya çapında büyük bir paniğe neden olurdu. Oysa karayolu ulaşımının tehlikelerini kanıksamış gibi bir halimiz var, bunu modern dünyada yaşama zanaatının risklerinden biri olarak kabullenmiş gibi görünüyoruz. Diğer

taraftan YZ böyle riskleri iyiden iyiye azaltma imkânını elinde bulunduruyor: Orta vadede mümkün görünen sürücüsüz otomobiller nihayetinde milyonlarca insanın hayatını kurtarma potansiyeli taşıyor.

Otonom araçların olmasını istemenin daha başka da pek çok haklı gerekçesi var elbette. Programlanan bilgisayarlar otomobilleri daha *randımanlı* kullanabilir, böylece kıt ve pahalı olan yakıtlar veya diğer güç kaynaklarından daha iyi yararlanılabilir ve nihayetinde daha masrafsız ve çevre dostu otomobiller ortaya çıkabilir. Bilgisayarların ayrıca, sözgelimi yoğun kavşaklarda çok daha fazla akış imkânı sağlayarak, yol ağlarından daha iyi faydalanmayı sağlama ihtimali de vardır. Dahası otomobillerin daha güvenli hale gelmesiyle, daha iyi koruyacak yani daha ağır ve dolayısıyla daha pahalı şasi ihtiyacı ortadan kalkacak ve yine daha ucuz, daha az yakan araçlara sahip olma imkânı doğacaktır. Hatta sürücüsüz otomobillerin araba sahibi olmayı gereksiz hale getireceği bile söylenmektedir. Bu argümana göre sürücüsüz taksiler o kadar ucuz olacaktır ki araba satın alma fikri ekonomik açıdan abesleşecektir.

Bütün bu nedenlerden ve daha nicesinden ötürü, sürücüsüz otomobil fikri malumun ilamı gibidir, herkesin ilgisini çeker; haliyle de bu alanda yapılan araştırmaların çok eskilere uzanan bir geçmişi vardır. 1920'ler ve 1930'larda otomobillerin kitlesel üretiminin başlamasıyla –çoğu insan hatasından doğan– otomobil kazası kaynaklı ölüm ve yaralanmalar artınca, otomatikleştirilmiş otomobiller mümkün mü tartışmaları gündeme geldi hemen. Ta 1940'lardan beri bu alanda envai çeşit deney yapılıyorduysa da, fikir ancak 1970'lerde mikroişlemci teknolojilerinin ortaya çıkışıyla beraber gerçek anlamda uygulanabilir hale geldi. Ancak sürücüsüz otomobil fikrinin karşılaştığı zorluklar say say bitmiyordu. Bunların en temel olanı ise algı problemiydi. Bir otomobilin tam olarak nerede olduğunu ve etrafında neler olup bittiğini bilmesini sağlamanın bir yolunu bulabilirseniz, sürücüsüz otomobil problemini çözebilirdiniz. Bu problemin çözümü, modern makine öğrenmesi teknikleri olacaktı. Bu teknikler bulunmasa, sürücüsüz otomobiller mümkün olamazdı.

Günümüzün sürücüsüz otomobil teknolojisinin atası dendi mi

genelde ilk akla gelen, çeşitli Avrupa hükümetlerinin ortak araştırma fonu örgütü Eureka'nın finanse ettiği, PROMETHEUS projesi olur. 1987'den 1995'e kadar yürütülen projenin sonunda bir tanıtım gösterisi yapıldı; bir otomobil Almanya'nın Münih şehrinden Danimarka'nın Odense'sine kadar gidip geldi. Ortalama olarak yaklaşık her 9 kilometrede bir insan müdahalesi gerekmişti; otomobilin kendi kendine, hiç insan müdahalesi olmaksızın katettiği en uzun mesafe yaklaşık 161 kilometreydi. Olağanüstü bir başarıydı bu, hele ki o dönem mevcut bilgisayar gücünün ne kadar sınırlı olduğunu düşününce. PROMETHEUS bu konseptin gerçekleştirilebileceğinin bir kanıtıydı sadece, yani dört dörtlük bir sürücüsüz otomobil olmaktan çok uzaktı, yine de projenin sonuçları akıllı hız sabitleyici gibi bugün artık otomobillerde görmeye alıştığımız pek çok yeniliğin önünü açtı. Ve hepsinden önemlisi, PROMETHEUS bu teknolojinin eninde sonunda ticari açıdan uygulanabilir hale geleceğini gösteriyordu.

2004 itibarıyla artık öylesine ilerleme kaydedilmişti ki DARPA **Grand Challenge** adıyla bir yarışma düzenleyecekti. ABD'nin kırsal kesiminde 241 kilometrelik bir parkuru kendi kendine tamamlamayı başaran araç kazanacaktı. Yarışmaya üniversiteler ve şirketlerden toplam 106 ekip katıldı, hepsi de yarışmayı kazanıp DARPA'nın bir milyon dolarlık para ödülünü almak istiyordu. Finale kalan 15 ekipten hiçbiri 13 kilometrenin üstüne çıkamadı. Hatta start alanını bile geçemeyen araçlar olmuştu. Yarışmada en iyi performansı sergileyen araç olan Carnegie Mellon Üniversitesi ekibinin Sandstorm'u rotadan çıkmadan ve bariyerlere takılmadan 12 kilometre ilerlemeyi başarmıştı.

Hafızam beni yanıltmıyorsa, o dönem YZ araştırmacılarının çoğu 2004 Grand Challenge'ı sürücüsüz otomobil teknolojisinin uygulanabilir hale gelmesi için biraz daha ilerleme kaydetmek gerektiğinin kanıtı olarak görüyordu. DARPA'nın hemen ertesi sene bir yarış daha düzenleyeceğini duyurduğunu öğrenince biraz şaşırmıştım açıkçası, üstelik bu kez ödül olarak iki milyon dolar veriyorlardı. 2005 Grand Challenge'a katılım daha fazla oldu, toplam 195 ekip vardı ve bunların 23'ü finale kaldı. 8 Ekim 2005'te yapılan finalde araçların Nevada'daki çölde 212 kilometrelik bir parkuru tamamlaması gerekiyor-

du. Bu kez yarışmayı tamamlayabilen ekip sayısı 5'e yükselmişti. Kazanan, Sebastian Thrun'ın başı çektiği Stanford Üniversitesi ekibinin tasarladığı STANLEY oldu. STANLEY parkuru ortalama olarak saatte yaklaşık 32 kilometre hızla 7 saatten biraz daha kısa bir sürede tamamladı. Modifiye edilmiş bir Volkswagen Touareg olan STANLEY, GPS, lazer menzilölçerler, radar ve kameralardan elde edilen sensör verilerini yorumlayan yedi bilgisayarla donatılmıştı.

2005 Grand Challenge insanlık tarihindeki en büyük teknolojik başarılardan biriydi. O gün, sürücüsüz araba problemi çözülmüş oldu; tıpkı yaklaşık bir asır önce Kitty Hawk'ta havadan ağır araçlarla uçma probleminin çözüldüğü gibi. Wright Flyer I uçağı o denemesinde sadece 37 metre ilerleyebilmiş, Wright Kardeşler'in bu ilk uçuşu yalnızca 12 saniye sürmüştü. Ama o 12 saniyelik uçuşun ardından, havadan ağır araçlarla uçmak mümkündü artık. 2005 Grand Challenge da aynı böyleydi işte, o günden sonra sürücüsüz arabalar mümkündü.

2005 Grand Challenge'ın ardından bir dizi yarışma daha düzenlendi. Bunların en önemlisi muhtemelen 2007 **Urban Challenge**'dı. 2005'te araçlar kırsaldaki yollarda sınanmıştı, 2007'de ise kentsel kesimde test edilecekti. Sürücüsüz otomobillerin, California eyaletinin trafik kurallarına uyarak, sağa sola park etmiş araçlar, vızır vızır işleyen kavşaklar, trafik keşmekeşi gibi gündelik şeylerle başa çıkıp bir parkuru tamamlamaları gerekiyordu. Ulusal elemelere kalan 36 takımdan 11'i finale çıktı. Final 3 Kasım 2007'de California'nın güneyindeki artık kullanılmayan eski bir havaalanında yapıldı. 6 takım yarışı tamamlamayı başardı. Kazanan Carnegie Mellon Üniversitesi ekibi 4 saatlik yarışta ortalama olarak saatte yaklaşık 23 kilometre yapmıştı.

O zamandan beri, hem geri kalmamak için çırpınan köklü otomobil şirketlerinden hem de bunu erken davranıp geleneksel şirketlerin önüne geçme fırsatı olarak gören yeni şirketlerden, sürücüsüz otomobil teknolojisine büyük paralar akıtılıyor.

2014'te ABD Otomotiv Mühendisleri Odası araçların otonomluk derecesini saptamaya yarayan bir sınıflandırma şeması ortaya koydu:[9]

**0. Düzey - Sıfır otonomluk:** Aracın otomatikleştirilmiş bir kontrol işlevi vb. yoktur. Sürücü her zaman tam kontrol halindedir (gerçi araç sürücüye kolaylık sunan birtakım uyarılar ve daha başka veriler sağlayabilir). Bugün yollarda gördüğünüz otomobillerin ezici çoğunluğu bu düzeye girer.

**1. Düzey - Sürücü yardımcısı:** Araç genelde rutin konularda kontrolü bir ölçüde sürücüden kendi üstüne alır, buna karşılık sürücünün bütün dikkatini araba kullanmaya vermesi beklenir. Fren ve gazı kullanarak aracın hızını koruyan adaptif hız sabitleyici buna örnektir.

**2. Düzey - Kısmi otomasyon:** Direksiyonun ve hızın kontrolünden araç sorumludur, buna karşılık sürücünün çevreyi sürekli takip etmesi ve gerektiğinde müdahale etmeye hazır olması beklenir.

**3. Düzey - Koşullu otomasyon:** İnsan sürücünün çevreyi sürekli takip etmesi beklenmez, ama araç başa çıkamadığı bir durumla karşılaştığı takdirde insan sürücüden kontrolü eline almasını isteyebilir.

**4. Düzey - Yüksek otomasyon:** Normal koşullarda bütün kontrol araçtadır, yine de insan sürücü müdahalelerde bulunabilir.

**5. Düzey - Tam otomasyon:** Sürücüsüz otomobil dendi mi hayal edilen budur; araca biner, gideceğiniz yeri bildirirsiniz, gerisini otomobil halleder. Direksiyon yoktur.

Elinizdeki kitabın yazıldığı tarih itibarıyla, mevcut sürücüsüz otomobil teknolojisinin en gelişmiş örneği Tesla'nın Autopilot sistemidir herhalde. Bu sistem ilk kez 2012'de piyasaya sürülen Model S ile birlikte sunuldu. Tesla'nın yüksek özellikli elektrikli otomobiller serisinin amiral gemisi olan Model S, piyasaya sürüldüğü yıllarda muhtemelen dünyada ticari olarak satılan bütün elektrikli araçlar içinde teknolojik açıdan en ileri olanıydı. Eylül 2014'ten itibaren bütün Model S'ler kameralar, radar ve akustik menzil sensörleriyle donatıldı. Bütün bu ileri teknoloji ekipmanıyla neye ulaşmanın amaçlandığı, Ekim 2015'te Tesla bu araç için "Autopilot" özelliğini etkinleştiren bir yazılım çıkarınca anlaşıldı: sınırlı bir otomatik sürüş becerisi.

Medya övmelere doyamadığı Autopilot'ı kestirmeden ilk sürücüsüz otomobil ilan edivermişti bile. Tesla ise ısrarla söz konusu teknolojinin sınırlarına dikkat çekiyordu. Bilhassa da Autopilot devredeyken bile sürücülerin kesinlikle ellerini direksiyondan ayırma-

maları gerektiğini vurguluyordu. Biraz evvel verdiğimiz otonomluk derecesi şemasına göre, Autopilot olsa olsa 2. Düzey'e dahil edilebilirmiş gibi görünüyordu.

Teknoloji ne kadar iyi olursa olsun, Autopilot devredeyken eninde sonunda ciddi bir kaza olacağı belliydi, ve ilk ölümcül kaza hiç şüphesiz dünyanın dört bir yanında manşetlere taşınacaktı. Derken korkulan oldu: Mayıs 2016'da Florida'dan bir sürücü Tesla'sının 18 tekerlekli bir tırın altına girmesiyle hayatını kaybetti. Kaza raporlarına bakılırsa, otomobilin sensörleri parlak güneşe karşı beyaz tırı algılayamamıştı, yani YZ yazılımı otomobilin hemen önünde bir araç olduğunu fark etmediğinden yüksek hızla kamyona doğru sürmüş, sonuç olarak sürücü hemen oracıkta hayatını kaybetmişti.

Diğer kazaları da incelediğimizde, mevcut sürücüsüz otomobil teknolojisinin temel bir sorunu olduğunu görüyoruz. Otonomluk bakımından 0. Düzey'de sürücüden beklenenin ne olduğu gayet açık: her şey. Aynı şekilde 5. Düzey'de de beklenti çok net: hiçbir şey. Gelgelelim söz konusu iki uç arasında kalan düzeylerde sürücünün neyle karşılaşmayı beklemesi gerektiği meselesi bu kadar net değil. Florida kazası ve daha başka kazalar, sürücülerin teknolojiye haddinden çok daha fazla güvendiğini ortaya koyuyor; otomobillerinin otomasyonu son derece düşük olduğu halde 4. veya 5. Düzey'miş gibi davranıyorlar. Sürücünün sistemden bekledikleri ile sistemin gerçekten yapabileceklerinin örtüşmemesinde kısmen de olsa basının bu konudaki aşırı heyecanının da payı var gibi görünüyor, zira basın teknolojik kapasitenin inceliklerini anlama konusunda da anlatma konusunda da pek iyi değil sanki. (Bu arada, Tesla'nın sistemine verdiği adın Autopilot yani "Otomatik Pilot" olması da ayrı mesele tabii.)

2018'de başka bir kaza meydana geldi; sonradan üstüne çok yazılıp çizilen bu olay, sürücüsüz otomobil teknolojisini daha da şaibeli hale getirecekti. 18 Mart'ta Tempe, Arizona'da Uber'ın araçlarından biri sürücüsüz moddayken Elaine Herzberg (49) adında bir yayaya çarparak ölümüne sebep oldu. Böyle olaylarda genelde olduğu üzere, kazanın meydana gelmesinde başka faktörler de rol oynamıştı. Araç otomatik acil fren sisteminin kaldırabileceğinden da-

ha hızlı gidiyordu, yani otomobil acilen fren yapmak gerektiğini fark ettiğinde her şey için çok geçti artık. Otomobilin sensörleri acil fren yapmayı gerektiren bir "engel" (kurban, Elaine Herzberg) olduğunu fark etmişti etmesine ama yazılım bunu yapmaktan kaçınmak üzere tasarlanmış gibiydi (bu konuda programcıların kafası karışmıştı sanki veya en azından kafalarında tuhaf bir öncelik sıralaması vardı). Hepsinden önemlisi, arabadaki "güvenlik sürücüsü", başlıca sorumluluğu bu tür durumlara müdahale etmek olduğu halde, anlaşılan o ki akıllı telefonundan televizyon izliyor, etrafa pek dikkat etmiyordu. Otomobilin kendi kendine yapabileceklerine fazla güvenmiş olabilir pekâlâ. Elaine Herzberg'ün trajik ölümü kesinlikle önlenebilirdi, ancak bu tamamen insan hatasıydı, teknolojik bir hata değildi.

Sürücüsüz otomobiller daha becerikli ve yaygın hale gelene kadar daha böyle pek çok trajediye tanık olmamızın adeta kaçınılmaz göründüğü fikri iç karartıcı tabii. Bu tür trajedileri öngörüp önlemek için makul bir biçimde yapabileceğimiz her şeyi yapmamız lazım. Aldığımız bütün tedbirlere rağmen meydana geldiklerinde de bunlardan ders çıkarmamız lazım. Temelde geleneksel uçuş kontrol sistemlerinin yerini elektronik bir arayüzün alması olarak açıklayabileceğimiz *fly-by-wire* (FBW) gibi teknolojilerin gelişmesi, uzun vadede sürücüsüz araçlarımızın da daha güvenli hale geleceğini gösteriyor.

Bugün sürücüsüz araçlar etrafında yürütülen hummalı faaliyet, söz konusu teknolojinin yakın olduğunu gösteriyor. Peki ama *ne kadar* yakın? Arabaya atlayıp gideceğiniz yeri belirttikten sonra gerisini sürücüsüz otomobilin halletmesi ne zaman mümkün olacak gibi görünüyor? Bunun en tarafsız göstergelerinden biri, sürücüsüz otomobil şirketlerinin araçlarını California'da test etme lisansı almak için eyalet yetkililerine sunmaları gereken bilgilerdir. Bu tür bilgilerin en önemlilerinden biri de Otonom Araç Devreden Çıkma Raporu'ndan elde edilebilir. Raporda hangi şirketten hangi aracın sürücüsüz modda kaç kilometre yol aldığı, bu denemeler esnasında otomobilin kaç kere devreden çıktığı mutlaka belirtilir. Devreden çıkma, insan güvenlik sürücüsünün müdahalede bulunarak otomo-

bilin bütün kontrolünü kendi eline almasını, yani tam da Elaine Herzberg'ün durumunda yapılmış olması *gerekeni* yapmayı icap ettiren bir durumdur. Diğer taraftan, her devreden çıkma ille de kaza hali, hele ki ölümcül bir kaza demek değildir. Yine de bu veri söz konusu teknolojinin nasıl bir performans sergilediğine dair bir fikir verir. Sürücüsüz modda katedilen kilometre başına devreden çıkma sayısı ne kadar azsa o kadar iyidir.

2017'de, California'da test lisansı için 20 şirket devreden çıkma raporu verdi. Bu raporlara göre, hiç devreden çıkmadan katettiği mesafenin uzunluğu ve 1600 kilometre başına devreden çıkma sayısı bakımından, açık ara lider Waymo şirketiydi. Waymo'nun sürücüsüz otomobilinin ortalama olarak yaklaşık her 8000 kilometrede 1 kere devreden çıktığı söyleniyordu. En kötü performans ise otomobil devi Mercedes'inkiydi: 1600 kilometre başına 774 devreden çıkma. Waymo Google'ın sürücüsüz otomobil şirketidir. Başlarda, 2005 DARPA Grand Challenge'ı kazanan ekibin başındaki Sebastian Thrun tarafından Google bünyesinde yürütülen bir projeyken, 2016'da Google'ın bir yan kuruluşu haline geldi. 2018 itibarıyla, Waymo araçlarının iki devreden çıkma arasında katettiği mesafenin yaklaşık 18.000 kilometreyi bulduğu rapor edildi.

Bu veriler bize ne söylüyor? Sürücüsüz araçların gündelik hayatın bir parçası haline gelmesine ne kadar kaldı?

BMW, Mercedes ve Volkswagen gibi otomobil devlerinin nispeten düşük bir performans sergilemesinden çıkarabileceğimiz ilk sonuç, otomotiv sanayisinde tarih boyunca çeşitli başarılara imza atmanın, sürücüsüz otomobil teknolojisinde başarının başlıca koşullarından biri *olmadığıdır*. Birazcık düşününce, nedeni gayet açık aslında bunun: Sürücüsüz otomobillere giden yol içten yanmalı motordan değil *yazılımdan* geçer, YZ yazılımından. ABD'li otomobil devi General Motors'ın 2016'da sürücüsüz otomobil şirketi Cruise Automation'ı açıklanmayan (ama anlaşılan o ki yüksek) bir meblağa satın alması, Ford'un yine bu alanda faaliyet gösteren bir girişim şirketi olan Argo AI'a bir milyar dolar yatırması şaşırtıcı değil o zaman. Sürücüsüz otomobillerin piyasaya sürülmesi konusunda her iki şirket de kamuoyu önünde iddialı açıklamalar yaptı: Ford'un tah-

minlerine göre, 2021 itibarıyla "tam otonom bir araçları satışa hazır" olacaktı.[10]

Yalnız şirketlerin bir durumu hangi ölçütlere göre "devreden çıkma" olarak nitelendirdiklerini tam olarak bilmiyoruz tabii. Belki sırf fazla ihtiyatlı davrandığı için Mercedes'in devreden çıkma oranı bu kadar yüksek görünüyordur. Yine de, elinizdeki kitabın yazıldığı tarih itibarıyla, Waymo her halükârda diğerlerinden açık ara önde gibi.

Bu arada, Californialı yetkililere sunulan devreden çıkma raporlarını insan sürücülerin performansları hakkında bildiklerimizle kıyaslamak da enteresan bir perspektif sunuyor. İnsan sürücülerin performansıyla ilgili nihai bir istatistiksel veri yok elimizde ama sözgelimi ABD'de insanlar ciddi bir kazaya yol açmaksızın ortalama yüz binlerce kilometre, hatta belki bir milyon kilometre yol katediyor gibi görünüyor. Demek ki insan sürücülerin güvenlik bakımından yoldaki performansıyla makul bir kıyaslamanın yapılabilmesi için, pazar lideri konumundaki Waymo'nun bile teknolojisini neredeyse yüz misli geliştirmesi lazım. Waymo'nun rapor ettiği her devreden çıkma ille de kaza anlamına gelmiyor elbette, dolayısıyla bilimsellikten uzak bir kıyaslama bu, yine de sürücüsüz otomobil şirketlerinin hâlâ ne kadar geniş ölçekli zorluklarla karşı karşıya olduğu konusunda en azından bir fikir veriyor.

Sürücüsüz otomobiller alanında çalışan mühendis arkadaşlarla konuştuğumda, bu teknolojinin karşılaştığı başlıca zorluğu beklenmedik olaylarla başa çıkmak olarak gördüklerini anlıyorum. Otomobilleri *çoğu* ihtimalle başa çıkabilecek şekilde eğittik diyelim. Eğitimde gördüklerine hiç benzemeyen bir durumla karşılaşırlarsa ne olacak? Araba kullanırken yaşadıklarınız genelde sıradan, öngörülebilir şeylerdir ama hiç beklemediğiniz bir şekilde başınıza gelebileceklerin listesi de azımsanamayacak kadar uzundur. Böyle durumlarda insan sürücünün hemen başvurabileceği koca bir dünya deneyimi dağarı vardır. Bu durumla nasıl başa çıkacağını düşünebilir veya, düşünmeye vakit yoksa bile, en azından içgüdülerini dinleyebilir. Sürücüsüz otomobillerin böyle bir lüksü yok ve yakın bir gelecekte de olacakmış gibi görünmüyor.

Başka bir zorluk da yolların mevcut durumundan (neredeyse bütün araçlar insan sürücüler tarafından kullanılıyor) karma bir duruma (birazı insan sürücü birazı sürücüsüz otomobil) ve sonunda tamamen sürücüsüz bir geleceğe nasıl geçeceğimizle ilgili. Otonom araçların araba kullanış tarzı insanlarınkine benzemez, dolayısıyla aynı yolu paylaştıkları insan sürücülerin kafaları karışacak, olup bitene sinirleneceklerdir. Diğer taraftan insan sürücüler de öngörülemezdir, trafik kurallarına harfiyen uymazlar, dolayısıyla YZ yazılımının insan sürücülerin davranış biçimini anlayıp onlarla güvenli bir biçimde etkileşime girmesi zor olacaktır.

Sürücüsüz otomobil teknolojisindeki ilerleme konusunda genel itibarıyla ne kadar iyimser bir değerlendirme yaptığımı düşününce, bu son söylediklerim şaşırtıcı derecede karamsar gelebilir size. Bu yüzden önümüzdeki onyıllarda bu alanda ne gibi gelişmeler yaşanmasını beklediğime dair fikirlerimi biraz daha açmak istiyorum.

Birincisi, sürücüsüz otomobil teknolojisinin *bir biçimiyle* çok geçmeden, önümüzdeki on yıl içinde gündelik hayatın parçası haline geleceğine sahiden inanıyorum. Burada kastettiğim 5. Düzey seviyesinde bir tam otonomluk değil elbette. Bence söz konusu teknolojinin önce belli başlı "güvenli" niş alanlarda giderek yayıldığını göreceğiz, ardından dünyanın geri kalanına doğru yol almaya başlayacak.

Bu teknolojinin hangi nişlerde yaygın olarak kullanıldığını görebiliriz peki? Örneğin madencilik. Belki de Batı Avustralya'daki veya Alberta, Kanada'daki devasa açık maden ocaklarında: Bu yerlerde nispeten az insanın bulunması, sağı solu belli olmayan yaya ve bisikletli sayısının da daha az olması demek. Madencilik endüstrisi otonom araç teknolojisini halihazırda geniş çaplı olarak kullanıyor zaten. Mesela İngiliz-Avustralyalı Rio Tinto madencilik grubu 2018'de Batı Avustralya'nın Pilbara kesiminde bir milyar tondan fazla cevher ve minerali devasa otonom kamyonlardan oluşan filosuyla nakletmişti.[11] Kamuya açık kaynaklara bakılırsa, kamyonlar 5. Düzey bir otonomluk derecesinden bir hayli uzaklar; "otonom" dan ziyade "otomatikleştirilmiş" olarak nitelendirilebilirmiş gibi görünüyorlar. Yine de sürücüsüz araçların sınırlı bir çevrede son de-

rece randımanlı bir biçimde kullanılmasına iyi bir örnek bu.

Aynı şekilde fabrikalar, limanlar ve askeri tesisler de sürücüsüz araçlar için gayet uygun görünüyor. Önümüzdeki birkaç yıl içinde, sürücüsüz araç teknolojisinin bu alanlarda ciddi bir yer kaplıyor olacağına inanıyorum.

Sürücüsüz araç teknolojisinin bu niş uygulama alanlarının ötesine geçerek gündelik hale gelmesiyle ilgili birçok olası senaryo var. Bunların bazıları, hatta hepsi gerçekleşebilir. İyi haritalanmış sınırlı kentsel çevrelerde veya belirli güzergâhlarda düşük hızlı "pod taksi"leri görme ihtimalimiz yüksek. Elinizdeki kitabın yazıldığı tarih itibarıyla, pek çok şirket benzer hizmetler için denemeler yapıyor ama son derece sınırlı biçimlerde (ve bugün itibarıyla bu otomobillerde acil durumlar için insan "güvenlik sürücüleri" bulunduruluyor). Böyle araçları düşük hızla sınırlamak, Londra gibi trafiğin her halükârda yavaş aktığı şehirlerde muhtemelen pek sorun olmayacaktır.

Başka bir ihtimal de şehirlerde ve belli başlı otoyollarda sürücüsüz otomobillere özel bir şerit tahsis etmek. Çoğu şehirde halihazırda otobüs ve bisiklet şeritleri var zaten, neden sürücüsüz otomobil şeridi de olmasın? Bu şerit sensörlerle ve otonom araçlara yardımcı olabilecek diğer teknolojilerle desteklenebilir. Söz konusu şerit aynı zamanda yolu otonom araçlarla paylaşan insan sürücülere çok açık bir mesaj verir: Dikkat robot sürücü!

5. Düzey otonomluk meselesine gelince, korkarım daha yolumuz uzun. Ama eninde sonunda olacak bu. En iyi ihtimalle, elinizdeki kitabın yazıldığı tarihten en az yirmi yıl sonra 5. Düzey otonom araçların yaygın olacağını tahmin ediyorum. Torunlarımın dedem arabayı *kendi* kullanıyormuş diye düşününce biraz afallayıp biraz da eğleneceğinden ise adım gibi eminim.

# 7

# İşlerin Nasıl Zıvanadan Çıkabileceğine Dair Tahayyüller

YZ'DE YÜZYILIN BAŞINDAN bu yana tanık olduğumuz hızlı ilerleme basında geniş yer buluyor. Bu haberlerin bazıları dengeli ve makulken, çoğu –kusura bakmayın ama– çok saçma. Bazıları bilgilendirici ve yapıcıyken, çoğu felaket tellallığından başka bir şey değil. Örneğin Temmuz 2017'de bir ara her yerde Facebook'un iki YZ sistemini kapattığına, çünkü bunların kendi aralarında (anlaşılan tasarımcıları da dahil) başka hiç kimsenin bilmediği bir dilde konuşmaya başladığına dair haberler çıktı.[1] O dönem hem haber başlıklarında hem sosyal medya gönderilerinde sanki Facebook bu sistemleri kontrollerini kaybetmekten korktuğu için kapatmış gibi bariz bir ima vardı. Halbuki Facebook'un deneyi son derece rutin, tümüyle zararsızdı. Basbayağı öğrenci projesi düzeyinde bir deneydi bu. Ekmek kızartma makinenizin katil bir robota dönüşme ihtimali ne kadarsa, Facebook sistemlerinin sapıtma ihtimali de o kadardı, yani düpedüz imkânsızdı bu.

Facebook olayının haberleştirilme tarzına gülüp geçtim ama derin bir yılgınlığa da kapılmadım değil. Çünkü bu tür habercilik "Terminatör" anlatısını fiştekleyip duruyor; kontrol edemeyeceğimiz bir şey yaratıyormuşuz, o şey insanlığın sonunu getirebilirmiş gibi (o unutulmaz rolüne bürünmüş Arnold Schwarzenegger'in sesini duyar gibi oldunuz mu). Kendi ellerimizle yarattıklarımızın bize saldırması modern bir fikir değil elbette, Mary Shelley'nin *Frankenstein*'ına kadar uzanan bir geçmişi var.

YZ'nin geleceğiyle ilgili tartışmalar hâlâ bu anlatının hâkimiyeti

altında; nükleer silah tartışmaları nasıl bir havada yürüyorduysa, YZ'nin geleceği de öyle bir tonda konuşuluyor artık. Bu durum PayPal ve Tesla'nın kurucularından milyarder girişimci Elon Musk'ı kaygılandırmış olmalı ki çekincelerini ifade ettiği bir dizi basın açıklaması yapma gereği duydu ve "sorumluluk sahibi YZ'yi" desteklemek için araştırma fonu olarak 10 milyar dolar bağış yaptı. 2014'te, o zaman dünyanın en tanınmış biliminsanı olan Stephen Hawking YZ'nin insanlığın varlığına karşı bir tehdit oluşturmasından korktuğunu açıkladı.

Terminatör anlatısı pek çok nedenle zararlı. Birincisi, muhtemelen hiç dert etmemize gerek olmayan konularda kaygılanmamıza sebep oluyor. İkincisi, odak noktasını böyle konulara kaydırarak, asıl üstünde durmamız *gereken* meseleleri arka plana itiyor. Terminatör anlatısı gibi çarpıcı manşetler vermeseler de, *bugün* üstünde durmamız gereken meseleler bunlar. Dolayısıyla bu bölümde YZ ile ilgili korku başlıklarını ele almak istiyorum: Terminatör senaryolarının gerçekleşmesi mümkün mü ve YZ'de işler nasıl zıvanadan çıkabilir? Başlangıç olarak, hemen bir sonraki kısımda, bodoslama konuya giriyorum. Nasıl olabilir, gerçekleşmesi ne derece mümkün, işte bunları tartışacağız. Ardından etik YZ konusuna geçeceğiz: YZ sistemlerinin ahlaki fail olarak edimde bulunması mümkün mü ve etik YZ için ne gibi çerçeveler öneriliyor? Son olarak da YZ sistemlerinin, korkunç bir biçimde olmamakla beraber, başarısızlığa uğramaya meyilli bir yönünü ele almak istiyorum: Bir YZ sisteminin bizim adımıza hareket etmesini istiyorsak, ona ne istediğimizi iletmemiz gerekir. Ancak bunu yapmak hiç de kolay değildir; arzularımızı doğru dürüst iletmeye dikkat etmezsek, sonunda elimize geçen sahiden *istediğimiz* şey değil YZ'nin anladığı şey olabilir.

## Tekillik Zirvası

Günümüz YZ'sinde Terminatör anlatısı genelde **Tekillik** fikriyle ilişkilendirilir. Amerikalı fütürolog Ray Kurzweil 2005 tarihli *Singularity is Near*[2] (Tekillik Yakınımızda) kitabında bu kavramı şöyle takdim ediyor:

Eli kulağında olan Tekilliğin altında yatan başlıca fikir, insan yaratımı olan teknolojimizin değişim hızının ivme kazanması ve güçlerinin üstel bir hızla artmasıdır. ... Önümüzdeki onyıllar içinde enformasyona dayalı teknolojiler, nihayetinde örüntü tanıma güçleri, problem çözme yetileri ve insan beyninin duygusal ve etik zekâsı dahil, insanlığın bütün bilgisini ve yetkinliğini kuşatıyor olacak.

Gördüğünüz gibi Kurzweil'ın son derece geniş kapsamlı bir Tekillik tahayyülü var. Ancak terim artık neredeyse tek bir fikirle özdeşleştiriliyor: Tekillik (genel anlamda) bilgisayar zekâsının insan zekâsını aşacağı varsayılan noktadır. Bu noktada, bilgisayarların zekâlarını kendilerini geliştirmek için kullanmaya başlayabilecekleri ve bu sürecin de kendi kendini besleyeceği düşünülür. Ardından, gelişmiş makineler gelişmiş zekâlarını kendilerini daha da geliştirmek için kullanacaklar ve bu böyle devam edip gidecektir. Bu argümana göre, bir noktadan sonra sırf insan zekâsıyla kontrolü yeniden ele almak imkânsız hale gelecektir.

İlgi çekici bir fikir bu ve bir o kadar da endişe verici. Burada bir parantez açıp Tekilliğin altında yatan mantığı anlamaya çalışalım isterseniz. Özetle, Kurzweil'ın ana argümanı şu fikre dayanır: Bilgisayar donanımı (işlemciler ve bellek) öyle hızlı gelişmektedir ki bilgisayarların enformasyon işleme kapasitesi çok yakında insan beyninkini aşacaktır. Kurzweil'ın argümanı bilgi işlemin gayet iyi bilinen düsturlarından biri olan Moore yasasına dayanır. 1960'ların ortasında formüle edilen yasa, adını bilgisayar işlemcisi şirketi Intel'in kurucularından Gordon Moore'dan alır. Transistörler bilgisayar işlemcilerinin yapıtaşlarıdır; bir çipe ne kadar çok transistör sığdırabilirseniz, çip verili bir zaman periyodunda o kadar çok iş yapabilecektir. Moore yasasına göre, bir yarı-iletkenin sabit alanına sığan transistör sayısı yaklaşık 18 ayda bir 2 katına çıkar. Bunun pratikteki sonucu kabaca şöyle ifade edilebilir: Bilgisayar işlemcilerinin gücünün 18 ayda bir yaklaşık 2 katına çıkması beklenebilir. Moore yasasının birçok önemli vargısı vardır; sözgelimi bilgisayar işlem gücünün aynı hızla ucuzlaması beklenebilir, işlemcilerin giderek küçülmesi beklenebilir. Müteakip 50 yılda yaşananlar Moore yasasının ne kadar güvenilir olduğunu gösteriyordu, ancak 2010 civarında

mevcut işlemci teknolojilerinden bazıları kapasitelerinin fiziksel limitlerine ulaşmaya başladı.

Kurzweil'ın argümanı zımnen, Tekilliğin kaçınılmazlığını ham bilgisayar gücünün mevcudiyetine bağlar. Ama şüpheli bir bağlantıdır bu. Tam da bu noktada gelin bir düşünce deneyi yapalım. Diyelim ki beyninizi bir bilgisayara indirebiliyoruz (burada sadece bir fantaziden bahsediyoruz elbette) ve bu bilgisayar da gelmiş geçmiş en hızlı, en güçlü bilgisayar. Elinizin altında böylesine muazzam bir işlem gücüyle üstün zekâlı mı olurdunuz? "Hızlı düşünebileceğiniz" aşikâr, iyi de bu sizi daha *zeki* yapar mı? Tali bir açıdan, yapar sanırım, ama gerçek anlamda, yani Tekillik anlamında yapmaz elbette.[3] Başka türlü söylersek, ham bilgisayar işlem gücü kaçınılmaz olarak Tekilliğe yol açmaz. Muhtemelen Tekilliğin *zorunlu* koşullarından biridir bu (çok güçlü bilgisayarlar olmazsa insan düzeyinde zekâya ulaşamayız) ama *yeterli* koşullarından değildir (sadece çok güçlü bilgisayarlara sahip olmak insan düzeyinde zekâya ulaşmaya yetmez). Başka bir deyişle, YZ yazılımı (örneğin makine öğrenmesi programları) donanımdan çok çok daha yavaş gelişir.

Tekilliğin gelecekte mümkün olacağı fikrinden şüphe etmek için başka sebepler de var.[4] Bir kere, YZ sistemleri insanlar kadar zeki hale gelse bile, bunun kaçınılmaz sonucu kendilerini bizim anlayamayacağımız bir hızla geliştirmeleri olmayacaktır. Elinizdeki kitapta buraya kadar az çok ortaya koymuş olduk: *Biz insanlar* son 60 yılda YZ'de oldukça yavaş bir ilerleme kaydettiğimiz halde, neye dayanarak insan düzeyinde Genel YZ'nin YZ'de daha hızlı bir ilerleme sağlayacağını düşünebiliriz ki?

Bununla ilgili bir diğer argüman da birlikte çalışan YZ sistemlerinin bizim kavrama yetimizi aşması ve kontrolümüzden çıkması fikrine dayanır (karş. bölüm başında bahsettiğimiz Facebook hadisesi). Açıkçası bu argüman da pek ikna edici gelmiyor bana. Diyelim ki Einstein'ı bin defa klonladınız. Hepsinin ortaklaşa zekâsı Einstein'ın zekâsının bin katı mı yapacak? Bence bin katından bir hayli az olur. Ayrıca, bizim bin Einstein bazı şeyleri ortaklaşa tek bir Einstein'dan daha *hızlı* yapacaktır belki ama, daha *akıllı* olmakla aynı kapıya çıkmaz ki bu.

Böyle nedenlerden dolayı, tanıdığım YZ araştırmacılarının çoğu Tekillik kavramına bayağı şüpheyle yaklaşıyor, bunu en azından yakın bir gelecekte mümkün görmüyor. Bugün bizi bilgi işlem ve YZ teknolojisi bakımından durduğumuz noktadan Tekilliğe çıkaracak bir yol bilmiyoruz. Ancak işinin ehli kimi yorumcular hâlâ kaygılı, böyle argümanlar öne sürülmesini rehavete kapılmak olarak değerlendiriyorlar. Nükleer enerji deneyimine dikkat çekiyorlar. 1930'ların başında, biliminsanları atom çekirdeğinde çok yüksek miktarda enerji saklı olduğunu biliyorlardı ama bu enerjiyi nasıl serbest bırakabilecekleri, hatta bunun mümkün olup olmadığı konusunda bile en ufak fikirleri yoktu. Kimi biliminsanları nükleer enerjinin kontrol altına alınıp kullanılabileceğine hiç ihtimal vermiyordu. Sözgelimi döneminin en tanınmış biliminsanlarından olan Ernest Rutherford bu enerji kaynağından yararlanabileceğimizi tahayyül etmeyi mânâsız bulduğunu söylemişti. Ama gayet iyi bilindiği üzere, Rutherford'un nükleer enerji imkânını bu sözlerle açıkça reddettiği konuşmasından bölümlerin gazetede yayımlandığı gün, Londra'da yürürken Rutherford'un sözlerine kafa yoran fizikçi Leo Szilard'ın aklına birdenbire zincirleme nükleer reaksiyon fikri geldi. Bundan sadece on küsur yıl sonra ABD Japon şehirlerine tam da Szilard'ın o kısacık zaman diliminde tahayyül ettiği şekilde güçlendirilmiş atom bombaları atıyor olacaktı. YZ'de bir Leo Szilard ânı, bizi birdenbire Tekilliğe götürecek beklenmedik bir içgörü olabilir mi? Bir ihtimal olarak bunu yok sayamayız elbette ama düşük bir ihtimal olduğu da ortadadır. Zincirleme nükleer reaksiyon aslında bir ortaokul çocuğunun bile anlayabileceği kadar basit bir reaksiyondur. İnsan düzeyinde YZ ise, son altmış yılın YZ araştırmalarından edindiğimiz deneyimlere bakılırsa, hiç de basit değildir.

Peki ya uzak gelecek, bundan yüz hatta bin yıl sonrası? İşlerin nereye varacağı çok daha belirsiz tabii. Bilgisayar teknolojisinin yüz hatta bin yıl sonra neler yapabilir hale geleceğini tahmin etmeye çalışmak akıl kârı değil. Yine de, eğer Tekilliğe ulaşırsak, Terminatör anlatısındaki gibi gafil avlanacağımıza pek ihtimal vermiyorum açıkçası. Rodney Brooks'un analojisini hatırlayarak, insan düzeyinde zekâyı bir Boeing 747 gibi düşünelim. Tesadüfen bir Boeing 747

icat etmemiz ne kadar olası gerçekten? Veya hiç hesapta yokken bir Boeing 747 geliştirivermemiz? Bu sorulara şöyle bir karşı yanıt da verilebilir elbette: *Pek olası değilse* de, *bilfiil* gerçekleşmesi durumunda hepimiz için son derece dramatik sonuçlar doğuracak olması, Tekillik konusunda neler yapacağımızı şimdiden düşünüp planlamamızı haklı çıkarır.

Tekillik yakın olsun olmasın (ki, şimdiye kadar yazdıklarımdan çok net bir biçimde anladığınız üzere, ben yakın olmadığını düşünüyorum), bugün YZ ile ilgili ciddi kaygılar duyulduğundan, bu alanda düzenlemeler yapmanın gerekli olup olmadığını ciddiyetle düşünmemiz lazım. Nükleer enerjide olduğu gibi, YZ'nin gelişmesini kontrol etmek için yasalar –hatta belki de uluslararası antlaşmalar– mı yapmalı? Açıkçası, matematiğin kullanımını düzenleyen yasalar yapmaya çalışmak ne kadar makulse, bu da o kadar makul geliyor bana.

## Asimov Hakkında Konuşmamız Lazım

Ne zaman bir etkinlik vesilesiyle konuşurken Terminatör anlatısından bahis açılsa, muhakkak biri çıkıp, böyle şeyler olmasını önlemek için daha en baştan YZ'mizi –burada ele aldığımız üzere– zamanla sağı solu belli olmayan bir YZ haline gelmeyecek şekilde inşa etmemiz gerektiğini söylüyor. Genelde hemen ardından başka bir katılımcı da bunun için bize tam da ünlü bilimkurgu yazarı Isaac Asimov'un formülleştirdiği **Robotiğin Üç Yasası**'nın lazım olduğunu ekliyor. Asimov bu üç yasayı, "pozitronik beyinler" aracılığıyla YZ'nin güçlü bir versiyonuyla donatılmış robotlar hakkındaki bir dizi öyküsünde ortaya koyuyordu. İlk defa 1939'da formülleştirdiği yasalar Asimov'a müteakip kırk yıl boyunca olay örgülerini inşa ederken kullanacağı yaratıcı bir araç sağlamış, bu araç vesilesiyle çok beğenilen bir dizi öykü ve izleyicilerinde hayal kırıklığı yaratan bir Hollywood filmi ortaya çıkmıştır.[5] Asimov'un öykülerindeki YZ "bilimi" –pozitronik beyinler– abestir ama yine de öyküler kuşkusuz eğlencelidir. Burada ele aldığımız konu itibarıyla, esasen Asimov'un üç yasası üstünde durmamızda fayda var:

1. Bir robot bir insana zarar veremez veya bir insanın zarar görmesine seyirci kalamaz.
2. Bir robot insanların, 1. Yasa ile bağdaşmayanlar hariç, bütün emirlerine itaat etmelidir.
3. Bir robot, 1. veya 2. Yasa ile bağdaştığı sürece, kendi varlığını korumalıdır.

Çok güzel ifade edilmişler ve ilk bakışta hepsi son derece iyi düşünülmüş görünüyor. Bu yasaları temel alan YZ sistemleri inşa edebilir miyiz?

Asimov'un yasaları hakkında öncelikle şunu söyleyebiliriz: Yasalar ne kadar iyi düşünülmüş gibi görünse de, Asimov'un öykülerinin çoğu bu yasaların kusurlarının ortaya çıktığı veya çelişkilere yol açtığı durumlara dayanır. Örneğin "Köşekapmaca"* öyküsünde SPD 13 isimli robot bir erimiş selenyum gölcüğünün etrafında hiç durmaksızın daireler çizmeye mahkûm olur, çünkü emirlere itaat etme gereği (2. Yasa) ile kendini koruma gereğinin (3. Yasa) birbiriyle çatıştığı bir durumla karşı karşıyadır. Etrafında dönüp durduğu gölcükle arasındaki mesafe hep sabittir, çünkü yaklaşacak olsa kendini koruma gereği devreye girer, uzaklaşacak olsa da itaat etme gereği. Asimov'un öykülerinde buna benzer pek çok örneğe rastlayabilirsiniz (henüz Asimov okumadıysanız, şiddetle tavsiye ederim). Yani yasalar ne kadar iyi düşünülmüş olsalar da su götürür yerleri kesinlikle çoktur.

Ancak Asimov'un yasalarıyla ilgili daha büyük sorun bunların bir YZ sistemi dahilinde uygulanabilir olmamasıdır.

Sözgelimi 1. Yasa'yı uygulamanın ne anlama gelebileceğini düşünelim. Bir YZ sisteminin bir eyleme kafa yorarken, bu eylemin muhtemelen *bütün* insanlar üstünde (yoksa insanların sadece bir kısmı mı önemlidir?) bugün ve gelecekte (yoksa sadece şimdikiler mi önemlidir?) ne gibi etkiler yaratabileceğini değerlendirmesi gerekecektir. Bu yapılabilir değildir. Ayrıca "seyirci kalamaz" ibaresi de bir o kadar sorunludur. Bir sistem sürekli *bütün* olası eylemlerini hiç kimsenin zarar görmeyeceğinden emin olacak şekilde *herkese* göre

* Isaac Asimov, *Ben, Robot*, çev. Gönül Suveren, Altın Kitaplar, 1992.

değerlendirmek zorunda mı olmalıdır? Yani aynı şekilde bu da yapılabilir değildir.

Burada "zarar" kavramını bile değerlendirmek zordur. Uçakla Londra'dan Los Angeles'a gittiğinizi düşünün, sonuçta büyük miktarlarda doğal kaynağı tüketmiş ve çok fazla kirliliğe ve karbondioksit oluşumuna yol açmış olacaksınız. Bunun birine zarar vermek olduğu neredeyse kesindir ama söz konusu zararın büyüklüğünü tam olarak ölçmek imkânsızdır. Bu örneği biraz zorlama buluyorsanız eğer, tam da bu nedenle uçakla seyahat *etmeyen* tanıdıklarım olduğunu söylemem lazım. Asimov'un yasalarına harfiyen itaat eden bir robotu uçağa binmeye ikna edemezsiniz bence. Hatta bırakın uçağa binmeyi, kılını bile kıpırdatamazdı herhalde; kararsızlık iyice elini ayağını bağlayacağından, muhtemelen bir köşeye saklanıp bütün dünyadan kaçmaya çalışırdı.

Asimov'un yasaları YZ sistemlerini inşa edenlere bir dizi üst düzey genel ilke sağlayabilir belki (YZ araştırmacılarının çoğunun bunları zımnen kabul edeceği kanaatindeyim), yine de YZ sistemleri dahilinde bire bir kodlanabilecekleri fikri –Asimov'un öykülerinde de olduğu gibi– hiç gerçekçi değildir.

Sonuç olarak, Asimov'un yasalarının –ve iyi niyetle YZ için etik prensipler ortaya koymaya çalışan diğer sistemlerin– kati surette bize bir faydası yoksa, YZ sistemleri için hangi davranışların kabul edilebilir olduğunu nasıl düşüneceğiz?

## Vagon Problemi Hakkında Konuşmayı Bırakmamız Lazım

Asimov'un yasaları YZ sistemlerinde karar vermeyi yönetmek için kapsayıcı bir çerçeve sunmaya yönelik ilk ve en bilinen girişim olarak görülebilir. YZ için ciddi bir etik çerçeve sunma niyetiyle ortaya atılmış değilse de, YZ'nin yakın dönemdeki büyümesiyle beraber ortaya çıkan ve ciddi araştırmalara konu olan böyle bir sürü çerçevenin atası olarak değerlendirilebilir.[6] Bölümün kalanında, bu çalışmaları inceleyip doğru istikamette ilerleyip ilerlemediğini tartışacağız. Gelin, etik YZ çevrelerinde belki diğer hepsinden daha çok ilgi gören bir senaryoyla başlayalım.

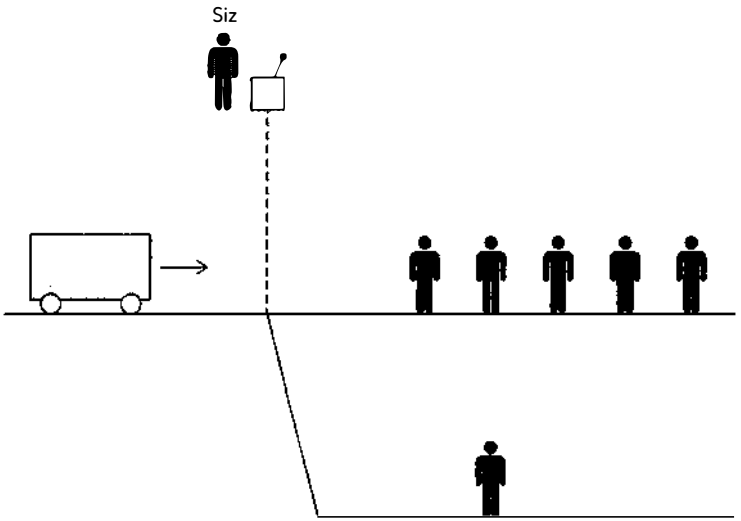
**Vagon Problemi** esasen İngiliz felsefeci Philippa Foot tarafından etik felsefesine 1960'ların sonunda takdim edilmiş bir düşünce deneyidir.[7] O dönem Foot'un amacı, kürtajı etik açıdan ele alan tartışmaları kuşatan kimi son derece duygusal meseleleri çözmekti. Pek çok versiyonu olan Vagon Probleminin en bilinen versiyonu şöyledir (bkz. Şekil 21):

Kontrolden çıkmış bir vagon son hızla, hareket edemeyen beş kişinin bulunduğu bir noktaya doğru ilerlemektedir. Yakınlarda bir makas vardır; kolu çekip makası değiştirirseniz, vagon aynı şekilde hareket edemeyen tek bir kişinin bulunduğu başka bir yola girecektir. Yani makası değiştirmezseniz beş kişi hayatını kaybedecek, değiştirirseniz de bu beş kişi yaşayacak ama o diğer bir kişi hayatını kaybedecektir.

Makası değiştirir misiniz değiştirmez misiniz?

Sürücüsüz otomobillerin yakın görünmeye başlamasıyla beraber Vagon Problemi de tekrar gündeme geldi. Uzmanlar hemen sürücüsüz otomobillerin kendilerini pekâlâ Vagon Problemindeki gibi bir durumda bulabileceğine, o zaman da böyle imkânsız bir seçim yapmak üzere YZ yazılımına başvurulacağına dikkat çektiler. Bu konuyla ilgili olarak 2016'da yayımlanan bir yazının başlığı "Sürücüsüz Otomobiller Kimi Öldüreceklerini Çoktan Düşünmeye Başladılar Bile"ydi.[8] İnternet üstünden hararetli bir tartışma başladı; tanıdığım birçok felsefeci, etik felsefesinin onca zaman karanlıkta kalmış bu problemi konusunda felsefecilerin ne düşündüğünü gerçekten merak eden bir kitlenin peydahlanıvermiş olması karşısında şaşkın ve bu durumdan memnundu.

Çok basitmiş gibi görünen Vagon Problemi şaşırtıcı derecede karmaşık birtakım meseleleri gündeme getiriyor. İçimden geçen cevabı soracak olursanız, diğer bütün koşulların eşit olduğunu varsayarsam, kolu çekip makası değiştirirdim çünkü bir kişinin hayatını kaybetmesi beş kişinin birden hayatını kaybetmesine yeğdir diye düşünüyorum. Felsefeciler buna, eylemlerin ahlaki karşılığı sonuçlarına göre değerlendirildiği için, sonuççu akıl yürütme tarzı diyor. Sonuççu teorilerin en bilineni **faydacılıktır**. Kökeni 18. yüzyıl doğumlu İngiliz felsefeci Jeremy Bentham'ın ve öğrencisi John Stuart




**Şekil 21:** Vagon Problemi. Bir şey yapmazsanız üstteki beş kişi, makası değiştirirseniz alttaki bir kişi hayatını kaybedecek. Ne yaparsınız?

Mill'in çalışmalarına dayanır. Bentham ve Mill "en büyük mutluluk ilkesi"ni ortaya atmıştır; buna göre, kabaca, insan hangi eylem "dünyanın toplam mutluluğu"nu azamileştiriyorsa onu seçmelidir. Bu formülü daha modern bir terminolojiyle ifade edersek, faydacı, toplumun toplam refahı olarak tanımlanan **toplumsal refahı** azamileştirecek şekilde hareket eden kişiye denir.

Bir genel ilke ortaya koyma fikri her ne kadar cezbediciyse de, "dünyanın toplam mutluluğu" derken tam olarak neyin kastedildiğini netleştirmek öyle kolay iş değildir. Diyelim ki Vagon Probleminde ilk hattaki beş, hadi diyelim on kişi eli kanlı katiller, ikinci hattaki bir kişi de masum bir yavrucağız. Sırf sayıca daha fazla oldukları için eli kanlı katillerin hayatı masum bir çocuğun hayatından ağır mı basacak, makası değiştirmeye karar vermek için yeterli bir kriter mi bu sizce?

Başka bir seçenek de bir eylemin bir genel "iyi" ilkesiyle bağdaştığı ölçüde kabul edilebilir olduğunu düşünmektir. Bunun en ti-

pik örneği bir canlının hayatına son vermenin yanlış olduğu fikridir. Bu ilke benimsendiği takdirde, ölüme sebebiyet veren *hiçbir* eylem kabul edilemez. Bu ilke doğrultusunda hareket eden biri Vagon Problemi durumunda yerinden kımıldayamaz, çünkü neye karar verirse versin sonuçta birileri hayatını kaybedecektir.

Üçüncü bir bakış açısı ise **erdem etiği** fikrine dayanır. Buna göre, önce bir karar vericide olmasını isteyeceğimiz erdemleri ete kemiğe büründüren "erdemli insan"ı tanımlayabilir, ardından da böyle bir insan şöyle bir ortamda neyi seçerdi diye düşünerek hangi tercihin doğru olduğunu bulabiliriz.

YZ'de karar verici haliyle bir ajan, sözgelimi dosdoğru yoluna devam edip beş can almakla yan yola sapıp bir can almak arasında karar vermesi gereken bir sürücüsüz bir otomobil olacaktır. Bu durumda, bir YZ ajanı Vagon Problemi benzeri bir durumla karşılaştığında ne yapacaktır, ne yapmalıdır?

Her şeyden önce kendimize şunu sormamız lazım: Bir YZ sisteminden, bu durumdaki bir insandan beklediğimizden *daha fazlasını* beklememiz makul mü? Dünyanın en büyük felsefecileri bile Vagon Problemine son noktayı koyamamışken, bir YZ sistemi nasıl koysun?

İkincisi, misal ben kendimi bildim bileli araba kullanırım ama bugüne kadar bir kere olsun Vagon Problemi benzeri bir durumla karşılaşmadım. Veya tanıdığım birinin böyle bir durumla karşılaştığını duymadım. Üstelik genel olarak etik veya özel olarak Vagon Problemi konusunda az önce aktardıklarım dışında pek bir şey bildiğim söylenemez – ehliyet sınavında kimse etik sorusu sormadı bana. Şimdiye kadar bunun bir zararını da görmedim. Yani bir insandan araba kullanabilmesi için etik alanında derin bir muhakeme yetisine sahip olması beklenmezken, sürücüsüz otomobillere yola çıkabilmeleri için önce Vagon Problemini çözmeyi şart koşmak açıkçası biraz abes geliyor bana.

Üçüncüsü, kimi etik problemlerin cevapları *size göre* ne kadar aşikâr olursa olsun, başka insanların *onlara göre* yine bir o kadar aşikâr olan *başka* cevapları vardır. MIT'den bir grup araştırmacının gayet akıllıca tasarlayıp çevrimiçi gerçekleştirdiği bir deney bu du-

rumu çok net bir biçimde ortaya koyuyor. Araştırmacılar önce **Moral Machine** (Ahlaki Makine) diye bir web sayfası hazırladılar; bu sayfa aracılığıyla ulaştıkları kullanıcılara Vagon Problemi benzeri bir sürü problem verip, tek tek her senaryoda sürücüsüz bir otomobilin nasıl hareket etmesi gerektiğini düşündüklerini sordular.[9] Olası kurbanlar arasında erkekler, kadınlar, kilolu insanlar, çocuklar, suçlular, evsizler, doktorlar, sporcular ve yaş almış insanlar, ayrıca köpekler ve kediler vardı. Zamanla deney çok popüler oldu, bu sayede tam 233 ülkeden yaklaşık 40 milyon bireysel karara dayanan bir veri toplayabildiler.

Bu veriler, Vagon Probleminde verilen etik kararın dayandığı tavrın, dünyanın farklı yerlerinden katılımcılarda çok temel farklar gösterdiğini ortaya koyuyor. Araştırmacılar ülkeleri benzer tavırlara göre gruplayarak, her biri nitelikleri bakımından ayrı bir etik çerçeveyi cisimleştiren başlıca üç "ahlaki küme" oluşturdular. Bunlar Batı, Doğu ve Güney kümeleriydi. Batı kümesi Kuzey Amerika ülkeleri ve Avrupa uluslarının çoğundan meydana geliyordu; Doğu kümesinde Japonya ve Çin gibi pek çok Uzakdoğu ülkesi ile Müslüman ülkeler vardı; Güney kümesi ise Orta ve Güney Amerika'daki Latin ülkelerini kapsıyordu. Batı kümesine kıyasla, Doğu kümesinde yasalara uyanların uymayanlara açık ara tercih edildiği, yayaları koruma fikrinin yolcuları koruma fikrine ağır bastığı, gençleri koruma fikrinin ise arka planda kaldığı görülüyordu. Güney kümesindeki tercihlerde statü sahibi bireyleri, gençleri ve kadınları kollayan tercihler göze çarpıyordu. Araştırmacılar meseleyi biraz daha kurcalayınca, kararların farklılaşmasında belirleyici rol oynayan başka faktörler de tespit ettiler. Örneğin refah düzeyinin yüksekliği veya hukukun üstünlüğünün tesis edilmiş olması gibi nitelikler, belli bir toplumda daha ziyade ne yönde tercihler yapılacağını tahmin etmede önemli bir rol oynuyordu.

MIT'den araştırmacılar, insanların Vagon Problemi benzeri bir durumda sürücüsüz bir otomobilin nasıl hareket etmesi gerektiğine dair görüşleriyle ilgili bu bulgularını, Almanya federal hükümetinin karar verme süreci için önerilen etik kurallar raporuyla kıyasladılar. 2017'de yayımlanan bu raporda yer alan 20 maddelik listede şöyle

ifadelere rastlanmaktadır:[10]

Tehlikeli durumlarda her zaman mala zarar vermekten kaçınmak yerine insan hayatını korumaya öncelik verilmelidir.

Kaza kaçınılmazsa, bu durumda nasıl hareket edileceğine, kişinin fiziksel özelliklerine bakılarak (yaş, toplumsal cinsiyet vb.) karar verilemez.

Seyir halindeyken verili bir anda sorumluluğu insanın mı yoksa bilgisayarın mı taşıdığı her zaman açık olmalıdır.

Araç ne zaman kimin kullandığının kaydını mutlaka tutmalıdır.

Bu rapordaki kimi öneriler, Moral Machine deneyine katılanların kimi fikirleriyle taban tabana zıttır. Örneğin raporda kişisel özelliklere bakarak ayrım yapmaya karşı çıkılırken, deneyde dünyanın farklı yerlerinden katılımcıların gençleri koruma yönünde ortak bir tercih ortaya koyduğu görülmektedir. Sürücüsüz bir otomobilin, etik raporda önerildiği üzere kişiler arasında ayrım yapmaması halinde, son evre kanser hastası olan yaşlıca bir yetişkin yerine küçük bir çocuk hayatını kaybetse, ne kadar şiddetli bir tepkiyle karşılaşılacağını bir düşünün. Verdiğim örneğin ne kadar tatsız olduğunun farkındayım, bunun için özür dilerim, ama ne demek istediğimi anlamışsınızdır.

Hiç şüphesiz merak uyandırıcı ve yaratıcı olsa da, Moral Machine deneyi ve temelindeki Vagon Probleminin, sürücüsüz otomobiller için YZ yazılımı konusunda bize pek bir şey katmadığını düşünüyorum. Önümüzdeki onyıllarda sürücüsüz otomobillerde bu tür bir etik muhakeme yetisi göreceğimizi sanmıyorum. Öyleyse gerçek bir sürücüsüz otomobil Vagon Problemi benzeri bir durumla karşılaştığında *pratikte ne yapar* dersiniz? Sürücüsüz otomobillerle ilgili çalışmaları yürütenler ayrıntılar konusunda pek ketumlar. Fakat son on-yirmi yılın YZ deneyimine bakınca, temel mühendislik ilkesinin güvenlik beklentisini azamileştirmek (başka bir deyişle risk beklentisini asgarileştirmek) olacağını düşünüyorum. Derinlemesine bir muhakeme yetisi görmeyi beklemiyorum. Muhakeme yetisine ucundan kıyısından yaklaşan bir şey olacaksa bile, *birden fazla engelle karşı karşıya kalırsan daha büyük olanından ka-*

*çın* düzeyinden öteye geçemeyecektir herhalde – ama açıkçası bu düzeyde bir akıl yürütme dahi pek olacak gibi değil. Muhtemelen görüp göreceğimiz, aracın frene abanması olacak, yani pratikte aynı durumla karşılaşsak bizim yapabileceklerimizden pek öteye geçemeyecek.

## Etik YZ'nin Yükselişi

"Kötü olmayın."

Google şirketinin mottosu, 2000-15 (yaklaşık)

YZ ve etik bağlamında, açıkçası biraz zorlama olan Vagon Probleminden şüphesiz daha kuşatıcı, daha önemli, konuyla daha doğrudan ilişkili birçok mesele var. Elinizdeki kitabın yazıldığı tarih itibarıyla, bu meseleler hararetle tartışılıyor. Her teknoloji şirketi *kendi* YZ'sinin cümle âleminkinden daha etik olduğunu kanıtlamaya ant içmiş gibi görünüyor. Gün geçmiyor ki yeni bir etik YZ girişimini duyuran bir basın bülteni daha yayımlanmasın. Dolayısıyla biraz bu girişimler üstünde durmakta, önümüzdeki yıllarda YZ'nin gelişimini ne şekillerde etkileyebileceklerine kafa yormakta fayda var.

Etik YZ için ileri sürülen ilk ve en etkili çerçevelerden biri, bir grup YZ bilimcisi ve yorumcusunun 2015 ve 2017'de California'daki Asilomar konferans merkezinde bir araya gelmesiyle ortaya çıkan **Asilomar prensipleri**dir. Toplantının başlıca sonucu olarak ortaya konan 23 ilke dünyanın dört bir yanındaki YZ bilimcilerine ve geliştiricilerine duyurulmuş, desteğini göstermek isteyenlerin imzasına açılmıştır.[11] Asilomar prensiplerinin çoğu hemen herkesin katılacağı türden ilkelerdir. Örneğin 1. Prensip'e göre YZ araştırmalarının amacı faydalı bir zekâ yaratmak olmalıdır, 6. Prensip'e göre YZ sistemleri güvenli ve emniyetli olmalıdır, 12. Prensip'e göre insanlar kendileriyle ilgili verilere erişme, bu verileri idare ve kontrol etme hakkına sahip olmalıdır.

Ne var ki kimi ilkeler daha ziyade temenni gibidir. Sözgelimi 15. Prensip'e göre "YZ'nin yarattığı ekonomik refah insanlığın tamamına fayda sağlayacak şekilde geniş çapta paylaşılmalıdır". Bu ilkenin

altına gönül rahatlığıyla imzamı atarım, fakat büyük şirketlerin bu konuda -mış gibi yapmanın ötesine geçebileceğini tahayyül etmek ne yazık ki biraz fazla saflık olur gibi geliyor bana. Büyük şirketlerin YZ'ye yatırım yapmasının başlıca nedeni böylece rekabet üstünlüğü elde ederek hissedarlarına fayda sağlamayı ummaları; insanlığa fayda sağlamayı istemeleri değil.[12]

Son beş prensip YZ'nin uzak geleceğiyle ve YZ'nin bir biçimde kontrolden çıkma ihtimaline ilişkin kaygılarla alakalı, yani burada bir kez daha Terminatör anlatısıyla karşılaşıyoruz:

- **Kapasite varsayımından kaçınma:** Bu konuda bir mutabakat olmadığı için, gelecekteki YZ kapasitelerinin üst sınırlarıyla ilgili güçlü varsayımlarda bulunmaktan kaçınmalıyız.
- **Önem:** İleri YZ yeryüzündeki hayatın tarihini derinden değiştirebilir, dolayısıyla bu olası rolünün önemine tekabül eden bir ihtimamla ve kaynaklarla yönetilecek şekilde planlanmalıdır.
- **Riskler:** YZ sistemlerinin riskleri, özellikle de insan hayatını yıkıma uğratma veya ortadan kaldırma riskleri, yaratmaları beklenen etki ölçüsünde bir gayretle planlanmalı ve hafifletilmelidir.
- **Özyinelemeli (rekürsif) kendi kendini geliştirme:** Özyinelemeli bir şekilde kendini geliştirerek [zekâsını otomatik olarak geliştirip bu gelişmiş zekâyı da kendisini daha fazla geliştirmek için kullanarak] veya yeniden üreterek hızla niteliğini veya niceliğini artırabilecek şekilde tasarlanmış YZ sistemleri sıkı güvenlik ve kontrol tedbirlerine tabi olmalıdır.
- **Ortak iyi:** Üstün zekâ sadece yaygın olarak paylaşılan etik ideallerin hizmetinde geliştirilmeli ve bir devlet veya organizasyonun değil bütün insanlığın yararına sunulmalıdır.

Bu prensiplerin altına da gönül rahatlığıyla imzamı atarım ama, aslına bakarsanız, sözü edilen senaryoların gerçek olması bize o kadar uzak ki burada bunlardan bahsetmek neredeyse felaket tellallığı yapmakla aynı kapıya çıkıyor. YZ bilimcisi Andrew Ng bütün bunları şimdiden dert etmeyi Mars'ta aşırı nüfus artışı sorununu dert edinmeye benzetiyor.[13] Gelecekte belki öyle bir noktaya geleceğiz ki bunlar *gerçekten* uykularımızı kaçıracak türden meseleler olacak, ama günümüzde böyle meseleleri bu kadar öne çıkarmak YZ'nin

güncel durumu hakkında yanıltıcı bir tablo çiziyor, daha da kötüsü dikkatimizi dağıtıp bizi asıl dert etmemiz *gereken* problemlerden (bunları bir sonraki bölümde ele alacağız) uzaklaştırıyor. Sözünü ettiğimiz senaryoların gerçekleşmesine daha çok vakit var gibi göründüğünden, herhangi bir bağlayıcılığı olmadığını bildikleri bu prensiplerin altına şirketler de imzalarını atıveriyorlar, üstelik onlar için iyi bir reklam bile oluyor bu.

2018'de, Google etik YZ için kendi esaslarını ortaya koydu. Asilomar prensiplerinden biraz daha kısa ve öz olan bu esaslar aşağı yukarı aynı meseleler (yarar sağlama, yanlılıktan kaçınma, güvenli olma) üstünde duruyordu. Google bunun yanı sıra, YZ ve makine öğrenmesinin gelişiminde en iyi pratiklerle ilgili nispeten daha somut esaslar da belirledi.[14] 2018'in ortalarında AB güvenilir YZ için etik ilkelerini belirledi,[15] bunu bilgi işlem ve bilgi teknolojileri alanlarının başlıca meslek örgütlerinden Elektrik ve Elektronik Mühendisleri Enstitüsü'nün (IEEE) etik tasarım ilkeleri izledi,[16] ayrıca bilgi teknolojileri dışındaki alanlarda faaliyet gösterenler de dahil olmak üzere belli başlı pek çok şirket kendi YZ ilkelerini yayımladı.

Büyük şirketlerin etik YZ'ye bağlılıklarını ilan etmeleri şüphesiz harika bir gelişme. Gerçi tam olarak neye bağlılıklarını ilan ettiklerini hakikaten anlayıp anlamadıkları şüpheli biraz. YZ'nin faydalarını herkesle paylaşmak gibi yüksekten uçan temenniler memnuniyetle karşılanıyor ama bunları belli başlı eylemlere tercüme etmek öyle kolay iş değil. On küsur sene boyunca Google'ın mottolarından biri "Kötü olmayın"dı. Kulağa çok hoş geliyor değil mi, sahiden de iyi niyetle söylenmiş gibi. Peki Google çalışanları için tam olarak ne demek bu? Google'ın Karanlık Taraf'a geçmesini engelleyeceklerse, onlara çok daha net bir biçimde yol göstermek gerekmez mi?

Gördüğünüz gibi, birbirinden farklı bütün bu çerçevelerde birtakım temalar tekrar tekrar karşımıza çıkıyor ve bunlar etrafında oluşan mutabakat giderek artıyor. İsveç Umea Üniversitesi'ndeki meslektaşlarımdan Virginia Dignum savunduğu bu ilkeleri üç kavram temelinde formüle ediyor: hesap verme, sorumluluk ve şeffaflık.[17]

Örneğin hesap verme bu bağlamda, bir YZ sistemi herhangi bir insanı derinden etkileyen bir karar veriyorsa, o kişinin bir açıklama-

yı hak ettiği anlamına gelir. Gelgelelim neyin açıklama sayılıp neyin sayılmayacağı sorusunun cevabı zordur, durumdan duruma değişir, ve bildiğiniz gibi mevcut makine öğrenmesi programları açıklama yapmaya muktedir değildir. Sorumluluktan bahsedebilmek için belli bir karardan kimin sorumlu olduğu konusunda belirsizlik olmamalıdır. Daha da önemlisi, bir YZ sisteminin bir karardan "sorumlu" tutulabileceğini iddia etmeye kalkışamayız: Burada sorumlu olsa olsa sistemi kullanıma sokan bireyler veya organizasyonlardır. Bu da bizi daha derin bir felsefi meseleye, **ahlaki fail** meselesine getirir.

Ahlaki fail denince genelde doğru ile yanlışı ayırabilen ve bu ayrımlar çerçevesinde eylemlerinin sonuçlarını anlayabilen varlıklar akla gelir. Popüler tartışmalarda YZ sistemleri çoğu zaman ahlaki fail olarak tahayyül edilir, dolayısıyla eylemlerinden sorumlu tutulabilecekleri varsayılır. Genel itibarıyla, YZ'de sorumluluk, ahlaki açıdan sorumluluk sahibi makineler inşa etmek demek değildir, YZ sistemlerini sorumlulukla inşa etmek anlamına gelir. Örneğin Siri benzeri bir yazılımı piyasaya sürerken, kullanıcıları bir yazılımla değil insanla etkileşime girdiklerini sanmalarına yol açacak şekilde yanıltmak, yazılımı geliştirenlerin YZ'yi sorumsuzca kullandığını gösterir. Bu örnekte yazılımın bir günahı yoktur, kabahat sistemi geliştirenlerde ve kullanıma sokanlardadır. Bu örnekte sorumlu tasarım YZ'nin insan olmadığını açıkça ortaya koymayı gerektirir.

Son olarak, şeffaflık, bir sistemin kullandığı bizim hakkımızdaki verilere bizim de erişebilmemiz, bu kapsamda kullanılan algoritmaların bize de açıklanması demektir.

Etik YZ'nin yükselişi memnuniyet verici bir gelişme, ancak önerilen çeşitli planların ne derece uygulamaya geçirileceğini bekleyip göreceğiz.

## Ne Dilediğinize Dikkat Edin

Etik YZ tartışmasında, işlerin nasıl zıvanadan çıkabileceğiyle ilgili en dünyevi gerçeklerden birini kimi zaman unutuveriyoruz: YZ nihayetinde bir *yazılımdır* ve yazılımın ters gitmesi için göz alıcı yeni teknolojiler gerekli değildir. Basitçe söylersek, hatasız yazılım diye

bir şey yoktur, ya halihazırda çökmüş yazılım vardır ya da *henüz* çökmemiş yazılım. Hatasız yazılım geliştirme problemi bilgi işlemde büyük bir araştırma alanı oluşturuyor, hataları bulup gidermek yazılım geliştirmenin başlıca faaliyetlerinden biri durumunda. Fakat YZ hatalara yepyeni kapılar aralıyor. Bunların en önemlilerinden biri, YZ yazılımı bizim adımıza bir şey yapacaksa, ne yapmasını istediğimizi ona bizim söylememizin gerekmesinden kaynaklanıyor.

Bundan bir on beş sene kadar evvel, araçların insan müdahalesi olmaksızın kendi kendini koordine edebilmesini sağlamayı amaçlayan tekniklerle ilgileniyordum. Bir demiryolu ağı senaryosu üstünde çalışıyordum. "Ağ" dediğime bakmayın, böyle söyleyince kulağa havalı bir şey gibi geliyor ama aslında dairesel bir demiryolu üstünde zıt yönlerde hareket eden iki trenden ibaret bir sistemdi bu. Zaten trenler de sanaldı, ortada bir demiryolu, hatta demiryolu maketi bile yoktu yani. Bu hayali demiryolu bir noktada dar bir tünelden geçiyordu ve trenler tünele aynı anda girerse bir kaza (simülasyonu) oluyordu. Benim amacım da işte bunu önlemekti. Genel bir çerçeve geliştirebilirsem, sistemime bir amaç sunma imkânım olacaktı (bu örnekte amaç çarpışmayı önlemekti); böylece sistem birtakım kurallarla gelecek, tren kurallara uyduğu takdirde amaca ulaşmak garantilenmiş olacaktı (trenler çarpışmayacaktı).

Sistemim çalıştı çalışmasına ama aklımdaki tam olarak bu değildi. İlk kez amacımı ifade ettiğimde, sistemimin getirdiği kurallar şöyle diyordu: *İki tren de hareketsiz kalmalı*. Bu işe yaramaz mı, yarar tabii, iki tren de hareketsiz kalırsa kaza maza olmaz ama benim aradığım böyle bir çözüm değildi.

Karşılaştığım, YZ özelinde ve bilgisayar bilimi genelinde standart bir problemdi: *Arzularımızı* bilgisayara *iletmek* isteriz ki bilgisayar bunları bizim adımıza gerçekleştirmeye çalışsın, ama bu birçok nedenle oldukça sorunlu bir süreçtir.

Birincisi, aslında ne istediğimizi bilmiyor olabiliriz, daha doğrusu bu konuda kafamız net olmayabilir. Bu durumda arzularımızı ifade etmemiz imkânsız olacaktır. Ayrıca arzularımız çoğu zaman birbiriyle çelişecektir. Öyleyse bir YZ sistemi arzularımızı nasıl anlayabilir?

Dahası tercihlerimizi bir *bütün olarak* ayrıntılarıyla ifade etmemiz mümkün olmayabilir. Çoğu zaman tercihlerimizin bir özetini vermenin ötesine geçemeyiz, öyle olunca da genel resimde boşluklar kalır. Bu durumda YZ sistemi boşlukları nasıl dolduracaktır?

Son olarak, belki de en önemlisi, *insanlarla* iletişim kurarken genellikle *ortak değer ve normlarımız* olduğunu varsayarız, dolayısıyla her etkileşimde bunları tek tek detaylarıyla ifade etmemiz gerekmez. Oysa bir YZ sistemi bu değer ve normları bilmez, dolayısıyla ya hepsini ayrıntılı olarak ifade etmek gerekecektir ya da bunların YZ sisteminin arka planında bulunduğundan emin olmak. Aksi takdirde asıl istediğimizi alamayabiliriz. Biraz önce bahsettiğim tren senaryosunda, amacın trenlerin çarpışmasını önlemek olduğunu iletmiş, ama her halükârda hareket etmelerine izin verilmesi gerektiğini aktarmayı unutmuştum. Bir insanın dolaylı yoldan asıl niyetimin ne olduğunu anlamasının neredeyse kesin olduğunu, buna karşılık bir bilgisayar sisteminin bunu *anlamayacağını* unutmuştum.

Oxford Üniversitesi'nden felsefeci Nick Bostrom çok satan kitabı *Süper Zekâ*'da (2014) böyle senaryoları popülerleştirerek anlatıyor.[18] Buna **sapkın örnekleme** adını veriyor: Bilgisayar istediğinizi yapar ama beklediğiniz şekilde değil. Bir sürü sapkın örnekleme senaryosu düşünerek saatlerce eğlenebilirsiniz: Robota kesinlikle evime hırsız girmesin derseniz, o da evinizi yakıp kül eder; kanser yüzünden ölümler son bulsun dersiniz, herkesi öldürür vb.

Gündelik hayatta tekrar tekrar karşılaştığımız bir problem bu: Ne zaman belli bir davranışı teşvik etmeye yönelik bir sistem tasarlansa, biri bu sistemin açığını bulur, öyle davranmaksızın ödüllerini alır. Yeri gelmişken Sovyet Rusya'dan (muhtemelen uydurma) bir anekdot aktarmak istiyorum size: Rus hükümeti çatal kaşık üretimini teşvik etmek istediğinden, toplam *ağırlık* bakımından en çok üretimi yapan fabrikaları ödüllendireceğini duyurmuş. Sonra ne olmuş dersiniz? Evet, fabrikalar hemen tek tek her biri daha ağır olan çatal kaşıklar üretmeye başlamış.

Disney çizgi filmlerine meraklı olanların aklına hemen şu örnek gelebilir: 1940 yapımı *Fantasia*'nın bir bölümünde, genç ve saf bü-

yücü çırağı (Miki Fare) kuyudan su çekip eve taşıma angaryasından yorgun düşünce, bir çalı süpürgeyi canlandırıp işi ona yaptırır. Derken uyuyakalır. Uyandığında bir de ne görsün, bodrum sular altında kalmış. Çırak hâlâ kova kova su getiren süpürgeyi durdurmaya çalışırken, bu bir süpürge olur size bin süpürge. İşler ancak büyücü ustasının müdahalesiyle yeniden yoluna girer. Miki istediğini söylediği şeyi almıştır, istediği şeyi değil.

Bostrom böyle senaryolar üstünde de durur: Ataş üretimini kontrol eden bir YZ sistemine üretimi azamileştirme amacı verilir; ifadeyi kelimesi kelimesine alan sistem, Dünya'yı ve evrenin geri kalanını ataşa dönüştürür. Problem bir kere daha nihayetinde iletişim problemidir. Bu örnekte, amacımızı iletirken, kabul edilebilir sınırların anlaşıldığından emin olmamız gerektiğini görürüz.

Bu problemle başa çıkma yollarından biri, eylemlerinin *her türlü yan etkisini asgarileştirmeyi* amaçlayan YZ sistemleri tasarlamaktır. Yani YZ sistemlerinin, amaçlarımıza ulaşmaya çalışırken, *dünyadaki her şeyin mevcut halini mümkün olduğunca korumaya çalışmalarını* isteriz. ***Ceteris paribus*** **tercihler** fikri bunu iyi yansıtır.[19] *Ceteris paribus* "diğer her şey sabitken" demektir; *ceteris paribus* tercihler ise, YZ sistemimize bir şey yapmasını söylerken, bu esnada diğer her şeyin mümkün olduğunca değişmeden kalmasını istediğimiz anlamına gelir. Sözgelimi "evime hırsız girmesin" derken kastettiğimiz tam olarak "evime hırsız girmesini engellerken *diğer her şeyin mevcut halini mümkün olduğunca korumaya çalış*"tır.

Bütün bu senaryolarda karşımıza çıkan, bilgisayarın *gerçekten istediğimizin ne olduğunu* anlamasının zorluğudur. **Ters pekiştirmeli öğrenme** alanı bu soruna yöneliktir. Pekiştirmeli öğrenmenin ne olduğunu 5. Bölüm'de görmüştük: Bir ajan belli bir ortamda hareket edip bir ödül alır; öğrenilmesi gereken, ödülü azamileştiren bir dizi eylemin bulunmasıdır. Ters pekiştirmeli öğrenmede ise bunun yerine "ideal" davranışı görürüz (bir *insan* ne yapar); zor olan, YZ yazılımı için ilgili ödüllerin ne olduğunu bulmaktır.[20] Uzun lafın kısası, arzulanan davranış için insan davranışını model almaktır.

# 8

# İşler Gerçekten Nasıl Zıvanadan Çıkabilir?

TEKİLLİĞİN ELİ KULAĞINDA OLDUĞU yolundaki fikirlere şüpheyle yaklaşmam, YZ'nin korkulacak hiçbir tarafı olmadığını düşündüğüm anlamına gelmiyor. YZ nihayetinde çok amaçlı bir teknoloji, nerede nasıl uygulanacağı konusundaki tek sınır hayal gücümüz. Böyle teknolojilerin hepsinin onyıllar hatta yüzyıllar sonra başta hiç niyet edilmemiş sonuçları ortaya çıkar, ve böyle teknolojilerin hepsi istismar edilme potansiyeli taşır. Tarihöncesi zamanlarda ateşi bulan adını sanını bilmediğimiz atalarımız, fosil yakıtlar kullanmanın iklim değişikliğine sebep olacağını tahmin edemezdi. İngiliz biliminsanı Michael Faraday 1831'de elektrik jeneratörü deneylerini yaparken, bu işin sonunun elektrikli sandalyeye kadar varabileceği aklının ucundan bile geçmemiştir herhalde. 1886'da otomobilin patentini alan Karl Benz, bu icadının yüz yıl kadar sonra her sene milyonlarca insanın ölümüne yol açacağını elbette kestiremezdi. Ve internetin mucidi Vinc Cerf, kendisinin bu masum yaratısı aracılığıyla teröristlerin kafa kesme videolarını paylaşacağını muhtemelen hayal bile edemezdi.

Böyle teknolojilerin hepsi gibi YZ'nin de dünya çapında hissedilecek olumsuz sonuçları olacaktır, ve böyle teknolojilerin hepsi gibi YZ de istismar edilecektir. Bu sonuçlar –bir önceki bölümde neden pek ihtimal vermediğimi açıklamaya çalıştığım– Terminatör anlatısı kadar dramatik olmayacak muhtemelen, ama önümüzdeki on-

yıllarda bizim, çocuklarımızın, torunlarımızın cebelleşmek zorunda kalacağımız sonuçlar olacak.

Dolayısıyla bu bölümde YZ ile ilgili olarak ciddiyetle üstünde durulması gerekiyor gibi görünen başlıca meseleleri ve bunların gelecekteki olası sonuçlarını ele alacağım. Bölüm sonunda, işlerin zıvanadan çıkma *ihtimalini* kabul etmekle beraber bunun Hollywood'un tahayyül ettiği şekilde gerçekleşmesinin pek mümkün olmadığı konusunda hemfikir olacağımızı umuyorum.

İstihdam ve işsizlikle başlayacağız: YZ işimizi elimizden alacak mı? Bu soruya kafa yorarken, YZ'nin çalışmanın doğasını nasıl etkileyeceği ve YZ güdümlü istihdamın yabancılaştırma ihtimali üstünde duracağız. Bu noktadan hareketle, YZ teknolojilerini kullanmanın insan hakları üstünde nasıl bir etkisi olabileceğini inceleyeceğiz ve ölümcül otonom silahlar ihtimalini düşüneceğiz. Ardından da algoritmik yanlılığın ortaya çıkışını, YZ'de yeterince çeşitlilik olmamasıyla ilgili meseleleri, keza yalan haber ve sahte YZ fenomenlerini ele alacağız.

## İstihdam ve İşsizlik

"Robotlar işimizi elimizden alacak.
Her şey için çok geç olmadan bir plan yapmalıyız."

*Guardian*, 2018

YZ'nin geniş kesimlerce –Terminatör anlatısından sonra– en çok tartışılan ve korkulan tarafı gelecekte çalışmayı nasıl etkileyeceği, bilhassa da insanları işsiz bırakma potansiyelidir. Bilgisayarlar yorulmaz, işe akşamdan kalma veya geç gelmez, tartışmaz, şikâyet etmez, sendikalaşmaz ve belki daha önemlisi ücret istemez. Neden işverenlerin bu meseleyle bu kadar ilgilendiğini, neden işçilerin bu konuda bu kadar gerildiğini anlamak hiç zor değil.

Son yıllarda ortalık "YZ Sizi Iskartaya Çıkaracak" başlıklarından geçilmiyorsa da, bugün olup bitenlerin yeni olmadığını anlamak önemli. İnsan emeğinin geniş ölçekte otomasyonu en azından Sanayi Devrimi'ne (1760-1840) kadar uzanır. Sanayi Devrimi malların

imal edilme tarzında derinden bir değişimi, küçük ölçekli üretim operasyonlarından (hane içi imalat faaliyeti) aşina olduğumuz türden bir büyük ölçekli fabrika üretimine geçişi temsil eder.

Sanayi Devrimi'nin altında yatan tek bir neden yoktu, birbirinden farklı bir dizi teknolojik ilerleme ile tarihsel ve coğrafi açıdan biricik olan birtakım koşulların yan yana gelmesiyle ortaya çıkmıştı. Sanayi Devrimi'nin yuvası olarak bilinen Birleşik Krallık'ta, imparatorluk topraklarından ithal edilen pamuk, çoğu İngiltere'nin kuzeyinde bulunan atölye/fabrikalarda işleniyordu. İmparatorluk toprakları sadece ham pamuk sağlamakla kalmıyor, nihai ürünler için de büyük bir pazar oluşturuyordu. Bu ekonomik koşullar da Birleşik Krallık tekstil sanayisinin gelişmesini sağlıyordu. Tekstilde 1730' lardan itibaren meydana gelen bir dizi teknolojik ilerleme daha büyük, daha hızlı makineleri mümkün hale getirince, istikrarlı bir biçimde yüksek kalitede ve hane içi imalata kıyasla çok geniş ölçekte kumaş imalatı da mümkün oldu. İthal edilen hammadde önce Liverpool ve Manchester'a geliyor, oradan da bu yeni sanayiye uygun olarak tekstil atölyeleri etrafında gelişen yerleşim yerlerine gönderiliyordu. Başlangıçta atölyeler su gücüne dayanıyordu, fakat kömürle çalışan buhar motorunda çeşitli teknik ilerlemeler kaydedilince, potansiyel açıdan öngörülemez olan su kaynaklarına bağımlılığı ortadan kaldıran bu yeni teknoloji hızla benimsendi. Ayrıca İngiltere bu teknoloji için gereken doğal kaynak bakımından da elverişli bir ortam sunuyor, kuzey kesimlerde kömür bolca bulunuyordu. Böylece tekstil fabrikaları çevresinde gelişen kasabaların yanı başında başlıca üretim faaliyeti kömür madenciliği olan yerleşim yerleri filizlenecekti.

Sanayi Devrimi'nin beraberinde getirdiği fabrika sistemi nüfusun büyük kesimi açısından çalışmanın doğasında kökten bir değişikliğe yol açtı. Sanayi Devrimi'nden önce insanların çoğu doğrudan veya dolaylı olarak tarım alanında çalışıyordu. Devrim, insanları tarlanın olduğu köyden fabrika etrafında gelişen kasabaya, işlerini de açık alandan ve hane içinden fabrika mekânına taşıdı. Çalışmanın doğası, bugün görsek tanıyacağımız bir hale geldi: fabrika mekânında, uzmanlaşmaya dayalı ve çok fazla tekrar içeren bir gö-

revin, otomasyon düzeyi yüksek bir üretim sürecinin parçası olarak yerine getirilmesi.

Hane içi imalata dayanan sanayiler malların fiyatı, kalitesi veya istikrarı konusunda bu yeni teknolojilerle rekabet edemedi. Bir yerden sonra bu işçiler başka iş aramak zorunda kaldı (çoğu fabrikalarda çalışmaya başlayacaktı). Çok büyük toplumsal değişimler yaşanıyor, yer yerinden oynuyordu ve bu durumu herkesin memnuniyetle karşıladığı söylenemezdi. Sözgelimi on dokuzuncu yüzyılın ilk onyıllarında Ludizm hareketi ortaya çıktı. Gevşek biçimde örgütlenmiş bir grup olan Ludistler sanayileşmeye başkaldırıyor, onların gözünde geçim kaynaklarını ellerinden alan makineleri paramparça edip yakıyorlardı. Ancak hareket kısa ömürlü oldu: Gidişattan korkan İngiliz hükümetinin 1812'de getirdiği yasal düzenlemelerle, makinelere zarar vermenin cezası idama kadar gidiyordu artık.

1760-1840 Sanayi Devrimi şüphesiz *ilk* sanayi devrimiydi: O zamandan beri teknoloji istikrarlı bir biçimde ilerliyor; Ludistlerin sanayileşmeye kafa tuttuğu on dokuzuncu yüzyıldan beridir, imalatın ve çalışmanın doğası bakımından böyle pek çok büyük değişim yaşanıyor. İşin ironik (ve üzücü) tarafı, ilk Sanayi Devrimi'nde pıtrak gibi çoğalan sanayi kasabalarının, başka bir sanayi devrimiyle beraber 1970'lerin sonuyla 1980'lerin başında mahvolması. Bu süreçte geleneksel sanayilerin gerilemesinde etkili olan pek çok faktör vardı. Uluslararası ekonominin küreselleşmesi, Avrupa ve Kuzey Amerika'nın geleneksel sanayi merkezlerinde imal edilen malların, dünyanın diğer tarafındaki gelişmekte olan ekonomilerde artık çok daha ucuza üretilmesi anlamına geliyordu. Ayrıca ABD ve Birleşik Krallık'ın öncüsü olduğu serbest piyasa ekonomisi geleneksel imalata elverişsizdi. Fakat teknolojinin asıl katkısı mikroişlemcilerin geliştirilmesi oldu; mikroişlemcilerin sahneye çıkması, sanayileşmiş dünyanın dört bir yanında çok sayıda vasıfsız fabrika işinin otomasyonuyla sonuçlandı.

Günümüz bilgisayar teknolojisini mümkün kılan da mikroişlemciler oldu. Mikroişlemci, bir bilgisayarın merkezi işlem biriminin bütün kilit bileşenlerini küçük ve ucuz tek bir birimde bir araya getiren bir sistem bileşenidir. Eskiden büyük, pahalı ve nispeten güve-

nilmez olan bilgisayarlar mikroişlemcilerle beraber ucuzlaştı, hızlandı, güvenilir hale geldi ve epey bir küçüldü. Mikroişlemciler özellikle imalat alanında pek çok işi otomatikleştirme imkânı sağladı. İngiltere'nin kuzeyi bundan feci etkilendi, hatta kırk yıl kadar önce imalat sanayilerinin yıkıma uğramasından sonra bir daha hiç toparlanamadı desek yeridir.

Diğer taraftan, yeni bir teknolojinin net etkisi ekonomik faaliyette bir *artış* yönünde olacaktır, zira yeni teknolojiler yeni iş, hizmet, zenginlik fırsatları yaratır. Nitekim mikroişlemcilerde de böyle oldu, nihayetinde yok edilen işlerden ve zenginlikten daha fazlası yaratıldı. Yalnız bu kez söz konusu işler İngiltere'nin kuzeyinde değildi.

Gördüğümüz üzere insan emeğinin otomasyonu yeni bir mesele değil, fakat bugüne özel bir boyutu var: Tıpkı eskiden otomasyon ve mekanizasyonun *vasıfsız* işçilere yaptığı gibi, gelecekte de YZ *vasıflı* işçilerin işlerini ellerinden alırsa? İnsanlar o zaman ne iş yapacak?

YZ'nin çalışmanın doğasını değiştireceği kesin. Fakat söz konusu değişimin etkileri Sanayi Devrimi'nin etkileri kadar kökten ve çarpıcı mı yoksa daha mı ılımlı olacak, işte bunu bilmiyoruz.

Oxford Üniversitesi'nden çalışma arkadaşlarım Carl Frey ve Michael Osborne'un 2013 tarihli "İstihdamın Geleceği"[1] raporu tam da bu tartışmayı alevlendiriyordu. Manşete çekilen tedirgin edici bir tahmine göre, yakın gelecekte YZ ve ilgili teknolojiler ABD'deki işlerin %47'ye varan bir kısmını otomasyona elverişli hale getirebilirdi.

Frey ve Osborne 702 mesleği otomatikleştirilme olasılığı dedikleri bir ölçüte göre sınıflandırıyorlardı. Buna göre en riskli gruplar arasında telefonla pazarlamacılar, elde dikiş dikenler, sigortacılar, veri giriş elemanları (ve daha pek çok ofis elemanı), santral görevlileri, satış görevlileri, baskı ressamları ve kasiyerler vardı. En az riskli gruplardan bazıları ise terapistler, diş hekimleri, danışmanlar, doktorlar ile cerrahlar ve öğretmenlerdi. Araştırmacılar şu sonuca varıyordu:

Bu modele göre, ulaşım ve lojistik işi yapanların, ofis ve idari destek çalışanlarının ve üretime dayalı işlerdeki emekçilerin büyük bölümü ciddi risk altında.

Araştırmacılar raporda ayrıca otomasyona *direneceğini* düşündükleri işlerin üç ayırt edici özelliğini tanımlıyorlardı.

Birincisi, muhtemelen bekleneceği üzere, kayda değer ölçüde zihinsel yaratıcılık gerektiren işlerin (sanat, medya, bilim vb.) güvende olduğunu düşünüyorlardı.

İkincisi, güçlü sosyal yetiler gerektiren işler güvendeydi. İnsan etkileşiminin ve ilişkilerinin incelik ve karmaşıklıklarını anlamamızı ve idare etmemizi gerektiren işler otomasyona direnecekti.

Son olarak, algıda zenginlik ve el becerisi gerektiren işleri otomatikleştirmek zordu. YZ açısından sorunlu bir alandır bu, zira makine algılamasında büyük ilerlemeler kaydedilmiş olmasına rağmen insanlar hâlâ makinelerden kat kat iyidir. Biz insanlar son derece karmaşık ve yapılandırılmamış ortamları çabucak anlamlandırabiliyoruz. Robotlar ise yapılı ve düzenli ortamlarla başa çıkmakla beraber eğitimlerinin kapsamı dışında kalan bir şeyle karşılaştığında zorlanır. Aynı şekilde, insan eli hâlâ en iyi robot elinden çok daha beceriklidir. Rodney Brooks 2018'de insan eli kadar maharetli bir robot eli yapmanın yirmi yılı bulacağını tahmin ediyordu.[2] En azından o zamana kadar, çok fazla el becerisi gerektiren işlerde çalışanlar muhtemelen güvendeler, yani bu satırları okuyan marangozlar, elektrikçiler veya tesisatçılar bu gece huzur içinde uyuyabilir.

Frey ve Osborne'un raporu, yayımlandığı zamandan beri geniş kesimlerce eleştiriliyor: Raporda metodolojik sorunlar ve dayanaksız varsayımlar olduğu, sonuçların fazla genel kaldığı öne sürüldü. Rapordaki en karamsar tahminlerin orta vadede gerçekleşeceğine pek ihtimal vermemekle beraber, YZ ile ileri otomasyon ve robotiğin ilgili teknolojilerinin yakın gelecekte pek çok insanı ıskartaya çıkaracağına inanıyorum. İşiniz esasen bir formdaki verilere bakarak bir karar vermekten ibaretse (örneğin başvuran kişiye kredi verilecek mi verilmeyecek mi), muhtemelen size ihtiyaç kalmayacak ne yazık ki. İşiniz esasen müşterilerle detaylı bir metnin dışına çıkmadan konuşmaktan ibaretse (çağrı merkezi işlerinin çoğu böyledir), muhte-

melen size ihtiyaç kalmayacak. İşiniz esasen iyi haritalanmış sınırlı bir kentsel çevrede rutin bir biçimde araba kullanmaktan ibaretse, muhtemelen size ihtiyaç kalmayacak. Ama *ne zaman* diye sorarsanız, işte orasını bilmiyorum.

Buna karşılık çoğumuz için, yeni teknoloji başlıca etkisini yaptığımız işin *doğasında* bir değişikliğe yol açarak gösterecek. Yani YZ sistemleri bizim yerimizi almayacak da biz işimizi yaparken YZ sistemlerini kullanmaya başlayacağız ve böylece işimizi daha iyi, daha randımanlı yapar hale geleceğiz. Traktör nihayetinde çiftçinin yerini almadı, sadece verimliliğini artırdı. Aynı şekilde elektrikli daktilo sekreterin yerini almadı, sadece verimliliğini artırdı. Çalışma hayatının ciddi bir kısmını meydana getiren bitmez tükenmez evrak işleriyle uğraşırken, bu usandırıcı süreci yöneterek hızlandırabilecek yazılım ajanlarıyla etkileşim içinde olacağız. İşimizi yaparken kullandığımız bütün araçlarda yerleşik halde YZ olacak, bizim adımıza birbirinden farklı şekillerde kararlar verecek, üstelik hangi şekillerde olduğunu görmeyeceğiz bile. Andrew Ng'e bakılırsa "ortalama biri zihinsel bir görevi neredeyse düşünmeden yerine getirebiliyorsa, bunu YZ kullanarak günümüzde veya yakın gelecekte otomatikleştirebiliriz muhtemelen".[3] Bu da *bir sürü* görevin otomatikleştirilmesi demek.

Pek çokları YZ işimizi elimizden alacak, bu yüzden işsizlik ve eşitsizliğin dünyanın dört bir yanında hüküm sürdüğü distopik bir geleceğimiz olacak diye korkarken, teknoloji açısından iyimser olanlar da robotik ve ileri otomasyonun bizi bambaşka bir geleceğe taşıyacağını umuyor. Bu **ütopyacı**lar YZ'nin bizi çalışmanın angaryalarından ve sıkıcılığından (nihayet!) kurtaracağına, gelecekte bütün işi (en azından pis, tehlikeli, sıkıntılı vb. olduğu için istenmeyen işleri) makinelerin yapacağına, böylece bizim de istediğimizi yapmakta (roman yazmak, Platon okumak, dağcılık vb.) özgür kalacağımıza inanıyorlar. Yanıltıcı bir hayal olmakla beraber bu da yeni değildir, ütopyacı YZ bilimkurguda göze çarpar (fakat *distopyacı* YZ kadar sık görülmemesi de enteresandır: Mutlu mesut sağlıklı bir yaşam süren iyi yetişmiş insanlarla dolu bir gelecekten ilgi çekici bir hikâye çıkarmak daha zor olsa gerek).

Bu noktada, mikroişlemci teknolojisinin, geliştirildiği 1970'lerin sonuyla 1980'lerin başında yarattığı etki üstünde durmakta fayda var. Bahsettiğimiz üzere, mikroişlemcilerin gelişi fabrikalarda bir otomasyon dalgasına yol açmıştı. Ve o dönem mikroişlemci teknolojisinin olası sonuçları hakkındaki kamuoyu tartışmaları, günümüz YZ tartışmalarına bir hayli benziyordu. O zaman için, pek de uzak olmayan bir gelecekte, ücretli işlerde zamanımızın çok daha azını geçireceğimize neredeyse kesin gözüyle bakılıyordu. Buna göre, aşağı yukarı haftada üç günlük çalışma norm haline gelecek, boş zaman da aynı ölçüde artacaktı.

Ama hiç de öyle olmadı. Peki neden? Bir teoriye göre, otomasyonun, bilgisayarların ve diğer teknolojik ilerlemelerin çalışma saatlerimizi boşaltmasına izin vermek varken, daha uzun saatler boyu çalıştık çünkü daha fazla mal ve hizmet satın almak istiyorduk. Bütün o renkli televizyonlar, VCR'lar, CD çalarlar, DVD oynatıcılar, ev bilgisayarları, cep telefonları, egzotik tatiller, daha büyük ve daha hızlı otomobiller, daha şık giysiler, daha iyi yiyecekleri istediğimizden, bunları karşılayabilmek için daha çok çalışmak durumunda kaldık. Hepimiz daha basit hayat tarzlarına gönül indirsek, daha az tüketim malına ve daha temel bir yaşam standardına razı olsak, belki de hepimiz daha az çalışmak durumunda kalacaktık. Başka bir açıklama ise bu durumu söz konusu teknolojik ilerlemelerle sağlanan faydanın paylaşılmamış olmasına dayandırıyor: Zenginler daha zengin oldu, eşitsizlik daha da arttı. Bu durum tam olarak nasıl açıklanırsa açıklansın, 1970'ler deneyiminden çıkan aşikâr ve biraz da iç karartıcı sonuç şu: En azından yakın gelecekte, teknolojinin hemen hepimiz için bir boş zaman ütopyası yaratması pek olası görünmüyor.

Bu da bizi *koşulsuz temel gelire*, yani çalışıp çalışmadığına bakılmaksızın ve birtakım kriterleri karşılaması beklenmeksizin toplumdaki herkese belli bir gelir garantisi verilmesi fikrine getiriyor. Koşulsuz temel gelir yeni bir fikir değil ama yakın dönemdeki teknolojik gelişmelerle, bilhassa YZ alanındakilerle birlikte yeniden gündeme geldi. YZ/robotik/otomasyonun yeterli zenginliği yaratmasıyla bir koşulsuz temel gelirin mümkün (çünkü işi makineler ya-

pabilir) ve arzulanır (çünkü öyle veya böyle insanlara iş kalmayacak, hepsini robotlar alacaktır) hale geleceği düşünülüyor.

Ütopyacı bir geleceğe inanmayı canıgönülden istesem de, YZ güdümlü geniş ölçekli koşulsuz temel gelir tertipleri yakında olacak iş gibi görünmüyor.[4] Birincisi, koşulsuz temel gelirin uygulanabilir olması için, YZ'nin önceki teknolojik yeniliklerin sağladığından çok daha büyük bir ekonomik fayda yaratması lazım. Ancak YZ'deki mevcut ilerlemelerin bu ölçekte bir fayda sağlayacağına dair hiçbir işaret yok. İkincisi, bir koşulsuz temel gelirin yürürlüğe girmesi için daha önce benzerini görmediğimiz türden bir siyasi irade gerekiyor. Bu eylem tarzının kabul görüp anaakım siyasetin bir parçası haline gelmesi için toplumsal koşulların iyice zorlaması, sözgelimi çok geniş çaplı bir işsizliğin olması lazım. Son olarak, koşulsuz temel gelir çalışmanın merkezi bir rol oynadığı toplum yapısını temelden sarsacaktır; halihazırda ise toplumların böyle değişiklikleri düşünmeye istekli olduğunu gösteren hiçbir işaret yok.

YZ çalışmanın durumunun değişmesi bakımından önemli bir faktör olsa da tek faktör değil, hatta belki en önemli faktör bile değil. Bir kere, önüne geleni silindir gibi ezip geçen küreselleşme henüz yolun sonuna gelmediğinden, dünyamızı hayal bile edemeyeceğimiz şekillerde sarsmaya devam edecek demektir. Dahası bilgisayarın kendisi de kesinlikle evrimini tamamlamış değil. Bilgisayarlar yakın gelecekte daha da ucuzlayacak, daha da küçülecek, hiç olmadığı kadar birbirine bağlı hale gelecek. Ve tek başına bu gelişmeler bile dünyamızı, bu dünyada yaşama ve çalışma tarzımızı daha da değiştirecek. Arka planda ise parçaları giderek daha fazla birbirine bağlı hale gelen dünyamızdaki toplumsal, ekonomik ve siyasi manzara her zaman olduğu gibi değişmeye devam edecek: Fosil yakıt kaynakları azalacak, iklim krizi yaşanacak, popülist siyaset güçlenecek vb. Bütün bu kuvvetlerin istihdam ve toplum üstündeki yansımaları en az YZ'ninki kadar hissedilir olacak.

### Algoritmik Yabancılaşma ve Çalışmanın Değişen Doğası

YZ hakkında bir kitapta Karl Marx ismini görünce –dünya görüşünüze bağlı olarak– şaşırabilirsiniz, fakat *Komünist Manifesto*'nun bu eşyazarı YZ'nin çalışmayı ve toplumu nereye götürdüğü konusundaki güncel tartışmayla yakından ilgili meseleler üstüne de kafa yoruyordu. Bilhassa da, on dokuzuncu yüzyılın ortasında, ilk sanayi devriminin hemen ardından, kapitalist toplum modern biçimini almaya başlamışken geliştirdiği yabancılaşma teorisi bu konularla alakalıydı. Marx bu çerçevede bir işçinin işiyle ve toplumla ilişkisini ve kapitalist sistemin bu ilişkiyi nasıl etkilediğini ele alıyordu. Sanayi Devrimi'nde ortaya çıkan fabrika sisteminde işçiler düşük ücretler karşılığında sürekli tekrar eden sıkıcı işler yaptıklarından tatmin sağlayamıyorlardı. İşlerini kendileri örgütleyemiyorlardı, hatta işleri üstünde hiçbir kontrolleri yoktu. Marx'a göre bu hayat tarzı hiç de doğal değildi. İşçilerin işleri tam anlamıyla mânâsızdı ama böyle işler yapmaktan başka da bir seçenekleri yoktu. Bütün bunlar bir yerden tanıdık geldi mi?

YZ ve ilgili teknolojilerin hızlı gelişimi Marx'ın teorisine yeni bir boyut kazandırıyor: *Gelecekte patronumuz bir algoritma olabilir*. Bu fenomene, özellikle de değişen çalışma modelinin başka bir veçhesi olan "gig ekonomisi" sektöründe çoktandır tanık oluyoruz. Bundan sadece elli yıl kadar önce, insanın mezun olduktan sonra girdiği ilk işten emekli olması nadir rastlanan bir durum değildi. Norm olan, uzun süreli istihdam ilişkileriydi, hatta iş değiştirip duranlara pek iyi gözle bakılmıyordu. Fakat uzun süreli istihdam ilişkileri zamanla azaldı, bunların yerini kısa süreli, parça başı ve sözleşmeli çalışma, kısaca gig ekonomisi aldı. Gig ekonomisinin son yirmi yılda balon gibi şişmesinin bir nedeni, mobil bilgi işlem teknolojisinin yükselişiyle, sözleşmeli çalışanlardan oluşan geniş işgücü kesimlerini koordine etmenin dünya çapında mümkün hale gelmesidir. Bir işçinin belli bir zamandaki konumu cep telefonundaki GPS sensörleriyle takip edilebiliyor, yine aynı telefonla işçiye anbean talimat verilebiliyor. İşçinin klavye vuruş sayısından gönderdiği

e-postanın tonuna kadar bir işgünü içinde yaptığı her şey bir bilgisayar programı tarafından saniye saniye takip edilip yönlendirilebiliyor.

Çalışma ortamını fazla yakından takip edip aşırı sıkı bir kontrol altında tutarak işçiler için bıktırıcı hale getirmekle eleştirilen şirketlerden biri de çevrimiçi alışveriş devi Amazon. *Financial Times*'ın 2013 tarihli bir haberinde Amazon için çalışan tipik bir depo işçisinin hali şu sözlerle anlatılıyor:[5]

> Amazon'un yazılımı, gerekli bütün ürün kalemlerini toplayarak arabayı doldurmak için en verimli yürüyüş rotasını hesaplıyor ve el tipi bir uydu navigasyon cihazının ekranına gönderilen talimatlarla depo çalışanını bir raftan diğerine yönlendiriyor. Bu verimli rotalar bile çok fazla yürümeyi gerektiriyor. ... Bir Amazon çalışanının ifadesiyle, "Robot gibi bir şeysiniz aslında, ama insan görünümlü robot; insanın otomasyonu da denebilir buna".

Marx bugün hayatta olsa, açıklamaya çalıştığı yabancılaşma kavramına çok iyi bir örnek olarak bu haberde anlatılanları da gösterebilirdi. YZ güdümlü otomasyon kavramı tam da budur işte: İnsan emeği sistematik bir biçimde, makine veya yazılımla otomatikleştirilemeyen görevlere indirgenir; bundan geriye kalan ne varsa o da inceden inceye izlenip denetlenir; yeniliğe, yaratıcılığa, bireyselliğe, hatta düşünmeye bile alan bırakılmaz. Yıllık performans değerlendirmenizi bir bilgisayar programı yapsa, hatta kovulup kovulmamanıza bu program karar verse nasıl hissederdiniz? Böyle düşünürken, sinik damarım kabarınca, bunları çok da dert etmememiz lazım belki diyesim geliyor; zira söz konusu işlerin az bir ömrü kaldı, çok yakında YZ ve robotik tarafından tamamen otomatikleştirilmiş olacaklar. (Amazon depo robotlarına ciddi paralar yatırıyor.)

Dünyadaki çalışan nüfusun büyük bölümünün böyle işlerde çalışması cezbedici bir gelecek tahayyülü değil. Üstelik yeni bir şey de değil bu, ta Sanayi Devrimi'nde başlayan trendlere eklenen bir YZ boyutu. Bugün pek çok insanın yukarıda bahsettiğimiz Amazon'un depo işçilerinden *çok daha* beter koşullarda çalıştığına şüphe yok. Teknoloji esasen nötrdür, onun nasıl kullanılacağına karar veren insandır. Dolayısıyla çalışanların, hükümetlerin, sendikaların ve

yasal düzenlemeleri yapanların YZ'nin bu veçheleri hakkında (çalışanları nasıl etkileyecek, insan onuruna yakışan çalışma koşulları nasıl olmalı vb.), keza YZ ve çalışmayla ilgili tartışmalarda bugüne kadar hep ağır basan işsizlik meselesi hakkında düşünmeye başlamaları lazım.

## İnsan Hakları

Bu tartışma, çalışma hayatlarımızın kontrolünü ele alan YZ sistemlerinin nasıl da yabancılaştırıcı olabileceğini ortaya koyuyor. Fakat YZ'nin kullanımıyla ilgili çok daha büyük, en temel insan haklarını bile zora sokan dertler de var. Bir YZ sisteminin size patronluk yapması, ne zaman mola verip ne zaman veremeyeceğinizi söylemesi, hedefler koyması, çalışma hayatınızı anbean değerlendirmeye tabi tutması ihtimalinden bahsettik; peki ya *hapse girip girmemenize karar verme erki de yine bir YZ sistemine verilseydi*?

Olmayacak şey değil, kimi YZ sistemleri halihazırda buna gayet yakın işler yapıyor zaten. Örneğin Mayıs 2017'de Birleşik Krallık'ta, Durham polis teşkilatı **Zarar Değerlendirme Risk Aracını** (Harm Assessment Risk Tool, HART)[6] uygulamaya geçireceğini duyurdu. Söz konusu YZ sisteminin amacı, bir şüphelinin serbest mi bırakılacağına yoksa gözaltına mı alınacağına karar verirken polis memurlarına yardımcı olmaktı. HART klasik bir makine öğrenmesi uygulamasıydı. 2008-2012 arasında elde edilen 5 yıllık gözaltı verisiyle –yaklaşık 104 bin "gözaltı olayı"yla– eğitilmiş, 2013 boyunca elde edilen bir yıllık veriyle de test edilmişti (yani eğitimde kullanılan veri ayrı, testte kullanılan ayrıydı).[7] Sistem az riskli vakalarda %98, çok risklilerde %88 doğru sonuç veriyordu. İki vaka türünün yüzdelerindeki bu fark ise sistemin daha en baştan çok riskli vakalarda (örneğin şiddet suçlarıyla ilgili olanlar) daha ihtiyatlı bir biçimde karar verecek şekilde tasarlanmış olmasıyla açıklanabiliyordu. Sistemin eğitiminde vakaların 34 farklı özniteliğinden yararlanılmıştı. Bunların çoğu şüphelinin siciliyle ilgili olmakla beraber yaş, cinsiyet, ikametgâh gibi özniteliklere de yer verilmişti.

Sistem sadece gözaltı kararında memurlara yardımcı olmak üze-

re tasarlanmıştı, Durham polis teşkilatının bu işi tümüyle HART'a bıraktığını düşündürecek bir durum yoktu ortada. Yine de, HART'ın kullanıldığı duyulunca, kamuoyunda ciddi bir rahatsızlık baş gösterdi.

Başlıca sıkıntılardan biri, HART'ın bir vakayı ancak çok sınırlı bir öznitelik kümesine dayanarak değerlendirmesiydi. Dolayısıyla deneyimli bir gözaltı memurunun insanları ve süreçleri idrak edişi ile sisteminki arasında dağlar kadar fark vardı. Bunun yanı sıra karar verme sürecinin şeffaf olmayışı da düşündürücüydü (tipik bir makine öğrenmesi programı olarak, gayet iyi bir performans sergilemekle beraber, kararlarını gerekçelendirme yetisine sahip değildi). Ayrıca eğitim için kullanılan verilerin ve seçilen öznitelik kümesinin önyargıya yol açma olasılığı da bir sorun olarak gündeme getirildi tabii (buna ek olarak, programın başvurduğu özniteliklerden birinin şüphelinin adresi olması kaygı vericiydi: Ya bu durum sistemin dezavantajlı mahallelerde yaşayanlara haksızlık etmesine yol açarsa, ne olacaktı?).

Başka bir mesele ise, her ne kadar araç karar vermeyi *desteklemeye* niyetiyle tasarlanmış olsa da, bir noktada asıl karar verici merci haline gelme ihtimaliydi: Yorgun, şaşkın veya tembel bir memur sorumluluğu üstünden atıp, bu işe kafa yormak yerine HART'ın önerisini uygulayıp geçse ne yapacağız?

Bence bütün bu kaygıların altında, HART benzeri sistemlerin insan muhakemesinin rolünü yıprattığı fikri yatıyor. Öyle veya böyle, bir insanın hayatı açısından ciddi sonuçları olan bir kararın başka bir insan tarafından verildiğini bildiğimizde çoğumuzun içi daha rahat ediyor. Tarafsız insanlar tarafından yargılanmak gibi en temel insan haklarımız büyük mücadeleler sonucunda kazanıldığından, haliyle, çok kıymetli. HART gibi bir sistem tarafından yargılanmak bize böylesine kıymetli hakları gözden çıkarmak gibi geliyordur belki. Dolayısıyla bu tür adımları hafife almamak lazım.

Ne var ki bunların hepsinin gayet meşru kaygılar olması, HART gibi araçların kullanımını topyekûn yasaklamamız gerektiği anlamına gelmiyor. Fakat kullanımlarıyla ilgili bazı noktalara çok dikkat etmemiz gerekiyor.

Birincisi, HART örneğinde açıkça gördüğümüz üzere, böyle araçlar ancak insanların karar vermesini *desteklemek* içi kullanılmalı, insanın yerini almamalıdır. Makine öğrenmesi programları karar verme bakımından mükemmellikten uzaktır: Zaman zaman öyle kararlar verirler ki ne kadar saçma olduğunu hemen fark edersiniz. Günümüzün makine öğrenmesinin en sinir bozucu taraflarından biri de tam olarak ne zaman sapıtacağının kestirilemeyişidir. Dolayısıyla bu tip programların kararlarını, özellikle insan hayatı açısından çok ciddi sonuçlar doğurdukları durumlarda, körü körüne uygulamak hiç akıl kârı değildir.

Bir diğer konu da bu tip bir teknolojinin geliştirilme ve kullanılma tarzıyla ilgili. HART'ı geliştiren deneyimli araştırmacılar ekibi böyle bir aracın yol açabileceği sorunları iyice düşünüp taşınmışa benziyor. Gelgelelim bütün program geliştiricilerin bu kadar deneyimli olmadığını, olası sorunları derinlemesine düşünemediğini biliyoruz. Dolayısıyla bu da insanlar adına önemli kararlar veren sistemlerin, HART örneğindekinden farklı olarak, büyük bir özen ve dikkatle inşa edilmeme ihtimali konusunda kaygı yaratıyor.

HART gibi sistemlerin envai çeşidinin halihazırda kolluk kuvvetlerinin kullanımına sunulmuş olması insan hakları örgütleri tarafından büyük bir endişeyle karşılanıyor. Örneğin Londra'nın Metropoliten Polisi, Gangs Matrix (Çete Matrisi) denen bir araç kullandığı için çok ciddi eleştiriler alıyor. Sistemde binlerce bireyin kaydı var, bu veriler kullanılarak basit bir matematiksel formül yoluyla bireyin ileride çete bağlantılı bir suça karışma ihtimali tahmin ediliyor.[8] Gangs Matrix sisteminin büyük bölümü alışılmış bilgi işlem teknolojisinden meydana geliyor, ne kadar YZ kullanıldığı belli değil, ama trend çok açık. Uluslararası Af Örgütü bu sistemi "ırk ayrımı yaparak koca bir siyahi erkek neslini suçlu ilan eden bir veritabanı" olarak tanımlıyor. Sırf belli bir müzik türünü dinlemeyi seviyorsunuz diye bile kendinizi bu veritabanında bulabileceğiniz iddia ediliyor. ABD'de PredPol diye bir şirket "öngörücü polislik"i destekleyen, nerede suç işleneceğini tahmin ettiği söylenen yazılımlar satıyor.[9] Böyle yazılımların kullanılması bir kere daha en temel insan haklarıyla ilgili meseleleri gündeme getiriyor: Ya veri yanlıysa? Ya-

zılım iyi tasarlanmadıysa? Polis programa haddinden fazla bel bağlamaya başlarsa? COMPAS adında başka bir sistem de suçlu bulunan birinin yeniden suç işleme olasılığını tahmin etmeyi amaçlıyor.[10] Sistem cezaya karar vermede kullanılıyor.[11]

Böyle fikirlerin ne kadar yanlış yerlere gidebileceğine dair uç bir örnek olarak, 2016'da iki araştırmacı yayımladıkları bir makalede kişinin suç işleyip işlemediğinin *sadece yüze bakılarak* tespit edilebileceğini iddia ediyorlardı (bu tür sistemler bizi yüz yıldır itibar görmeyen suç teorilerine geri götürüyor). Bu çalışma hakkındaki başka bir makalede ise sistemin suçlu-suçsuz kararını kişinin gülümseyip gülümsememesine göre veriyor olabileceğine dikkat çekiliyordu: Sistemi eğitmek için kullanılan polis sabıka fotoğraflarında gülümseyen pek olmuyordu.[12]

## Katil Robotlar

Buraya kadar, patronunuzun bir YZ sistemi olmasının yabancılaşmaya yol açacağından, gözaltına alınıp alınmayacağınıza bir YZ sisteminin karar vermesinin insan haklarına yönelik potansiyel bir saldırı olduğundan bahsettik. Peki ya bir YZ programı *yaşayıp yaşamayacağınıza karar verme* erkine sahip olsaydı, o zaman nasıl hissederdiniz? YZ öne çıktıkça, bu ihtimalle ilgili kaygılar basında geniş yer buluyor. Ve böyle kaygıların körüklenmesinde ezeli düşmanımız Terminatör anlatısının kısmen de olsa bir rolü var elbette.

Otonom silahlar çok hararetli bir konu başlığı. Bu tür sistemlerin iğrenç ve ahlakdışı olduğuna, asla inşa edilmemesi gerektiğine canıgönülden inanan çok. Böyle düşünenler genelde duyunca çok şaşırıyor ama bu fikre katılmayan sağlam karakterli insanların sayısı da az değil. Bu yüzden, isterseniz şimdi de –çok hassas bir konu olduğundan mümkün olduğunca yumuşak adımlarla ilerleyerek– YZ güdümlü otonom silahlar ihtimalinin gündeme getirdiği meseleleri biraz daha ayrıntılı olarak ele alalım.

Otonom silahlar hakkındaki tartışmalar büyük ölçüde dronların savaşta giderek daha fazla kullanılmasından kaynaklanıyor. Bu insansız hava araçları, savaşta füze gibi silahlar taşıyabiliyor. İnsan pi-

lot taşımadıkları için, alışıldık bir uçaktan daha küçük, hafif ve ucuz olabiliyorlar. Dronu uçurmak uzaktan kumanda eden için herhangi bir risk teşkil etmediğinden, pilotu olan araçlar için fazla riskli bulunan durumlarda kullanılabiliyorlar. Bütün bu özellikler haliyle dronları askeri kuruluşlar için çekici bir seçenek haline getiriyor.

Son elli yılda pek çok girişimde bulunulmuş olmasına rağmen, askeri dronlar ancak bu yüzyılda hayata geçirilebildi. ABD 2001'den beri Afganistan, Pakistan ve Yemen operasyonlarında dron kullanıyor. Kaç kere olduğunu bilmiyoruz ama yüzlerce kere kullanmış olması gayet olası görünüyor. Ve bu da muhtemelen binlerce ölüme sebebiyet vermiş olmak demek.

Uzaktan kumanda edilen dronlar birbirinden ciddi pek çok etik meseleyi gündeme getirir. Örneğin dronu uçuran pilot fiziksel açıdan tehlikede olmadığından, bedenen orada olsa yapmayacağı işlere kalkışabilir. Ve yine bir o kadar önemlisi, eylemlerinin doğuracağı sonuçları o kadar ciddiye almayabilir.

Bu ve benzeri daha pek çok nedenle, uzaktan kumanda edilen dronların kullanılması çok tartışmalı bir konu. Ancak *otonom* dron olasılığı bu tür endişeleri bambaşka bir seviyeye taşıdı. Zira bu araçlar uzaktan kumanda edilmeyecek, görevlerini büyük ölçüde insan yönlendirmesi veya müdahalesi olmaksızın gerçekleştireceklerdi. Ve söz konusu görevlerin bir parçası olarak, insanların canını alıp almamaya karar verme erkine sahip olabilirlerdi.

Otonom dron, veya başka bir otonom silah çeşidi, gayet iyi bildiğimiz o anlatıyı akla getiriyor hemen: amansız, ölümcül bir isabetlilikle katliam yapan, insanın insaf, merhamet veya anlayışından zerre nasibini almamış robot orduları. Böyle Terminatör senaryoları yüzünden uykularımızın kaçmasına pek de gerek olmadığını görmüştük, fakat otonom silahlar için durum farklıdır: Sapıtmaları ölümcül sonuçlar doğurabilecek bir ihtimal olduğundan, geliştirilmeleri ve kullanılmalarına karşı önemli bir argümandır. Otonom silahlar konusunda endişelenmek için daha pek çok neden var. Bunlardan biri, otonom silahları olan bir ulusun, vatandaşlarını doğrudan ateşe atmadığı için, savaşa girme kararını daha rahat verme olasılığıdır; böylece savaşlar da daha yaygın hale gelebilir. Fakat en sık

karşılaşılan itiraz otonom silahların düpedüz ahlaksızlık olduğu yönündedir: Bir insanın canını almaya karar verebilen makineler inşa etmek yanlıştır.

Yeri gelmişken, YZ güdümlü otonom silahların günümüz teknolojisiyle artık tamamen mümkün olduğunu da söyleyeyim. Şöyle bir senaryo düşünün mesela:[13] küçük bir helikopter dron, yaygın ve ucuz olanlarından; üstünde bir kamera, GPS navigasyonu, küçük bir bilgisayar ve el bombası büyüklüğünde bir patlayıcı var. Bir şehirde sokak sokak dolaşıp insanları bulmak üzere programlanmış. Bulduğu kişiyi tanıması gerekmiyor, bir insan formu tespit etmesi yeterli. Tespit edince de hedefine doğru alçalıp bombasını patlatıyor.

Bu korkutucu senaryoda YZ için tek gereken sokakta dolaşma, insan formunu teşhis etme ve uçma yetisi. Yetkin bir lisansüstü YZ öğrencisi, elinde kaynak varsa, işe yarar bir prototip ortaya çıkarır diye düşünüyorum. Üstelik bunların geniş ölçekte üretimi de *gayet* ucuza mal olacaktır. Şimdi de böyle binlerce dronun Londra, Paris, New York, Delhi veya Pekin semalarını kapladığını hayal edin. Bunun ne kadar dehşet verici bir katliam olacağını düşünün. Bu konuda kesin bilgim yok ama birilerinin çoktan böyle bir dron inşa etmiş olması çok mümkün.[14]

Çoğu insana göre ciddi ciddi otonom silahların meziyetlerini tartışmanın mantıklı bir tarafı olmasa da, her halükârda bu minvalde dönen bir tartışma var. Bu tartışmada öne çıkan isimlerden biri, ABD'deki Georgia Teknoloji Enstitüsü'nden Ron Arkin. Otonom silahların kaçınılmaz göründüğünü (birileri bir yerlerde mutlaka böyle silahlar yapacak), dolayısıyla yapılabilecek en iyi şeyin *insan askerlerden daha etik* davranışlar sergileyecek robotların nasıl tasarlanabileceğine kafa yormak olduğunu savunuyor.[15] Arkin nihayetinde insan askerlerin etik davranış sicilinin pek iç açıcı olmadığına dikkat çekerek, etik açıdan "kusursuz" silahların imkânsız olduğunu teslim etmekle beraber, insan askerden genel itibarıyla daha etik otonom silahlar inşa edilebileceğini öne sürüyor.

Otonom silahlardan yana başka argümanlar da var. Örneğin, savaş berbat bir şey olduğundan, insanlar savaşacağına robotlar savaşsın, daha iyi robotları olan kazansın diye düşünülüyor. (Bu durumda

bizim de daha iyi robotlara sahip taraf olmamız gerekiyor tabii.)

Otonom silahlara karşı çıkıp savaşın daha alışıldık tarzlarına itiraz etmemenin etik açıdan kafa karıştırıcı olduğunun altını çizen bir tartışmaya da denk geldim. Diyelim ki B-52 tipi bir bombardıman uçağınız var, 15 bin metrede seyrederken bomba yükünü bırakıyor, ama bombardımancı bu 32 ton bombanın tam olarak nereye veya *kimin* tepesine ineceğini bilmiyor. Öyleyse kimin öldürüleceğine net bir biçimde karar veren otonom silahlara itiraz edip, insanları *gelişigüzel öldüren* konvansiyonel bombalamaya karşı *çıkmamanın* nasıl bir mantığı var deniyor. Sanırım sorunun cevabı ortada, ikisine de karşı çıkmak gerekir, ne var ki pratikte konvansiyonel bombalama ölümcül otonom silahlar fikri kadar tartışılmıyor.

Otonom silahlardan yana ne gibi argümanlar öne sürülürse sürülsün, uluslararası YZ camiasından çoğu araştırmacının bu silahların geliştirilmesine kesinlikle karşı olduğu yönünde güçlü bir izlenimim var. Otonom silahların insan askerlerden daha etik davranışlar sergileyecek şekilde tasarlanabileceği yaygın kabul gören bir fikir değil. Teoride enteresan bir fikir ama pratikte bunun nasıl yapılacağını bilmiyoruz, üstelik –bir önceki bölümdeki tartışmayı hatırlayacak olursanız– yakın zamanda yapılabilirmiş gibi de görünmüyor. Bu argüman vaktiyle ne kadar iyi niyetle ve içtenlikle ortaya atılmış olursa olsun, bugün otonom silahlar inşa etmeyi meşru göstermek isteyenler tarafından gaspedilmiş olması tedirgin edici.

Geçen onyılda biliminsanları, insan hakları grupları ve otonom silahlar geliştirilmesini durdurmak isteyen daha niceleri arasında bir hareket baş gösterdi. 2012'de başlatılan Katil Robotları Durdurun kampanyası Uluslararası Af Örgütü gibi başlıca uluslararası insan hakları gruplarının da desteğini aldı.[16] Amaç tam otonom silahların geliştirilmesi, üretilmesi ve kullanılmasının bütün dünyada yasaklanmasını sağlamaktı. Kampanya YZ camiasından büyük destek gördü. Temmuz 2015'te, bu yasağı savunan bir açık mektup neredeyse 4 bin YZ bilimcisi ve 20 bini aşkın kişi tarafından imzalandı. Mektuptan sonra birbiri ardına girişimler geldi. Devlet kuruluşlarının gündeme getirilen konularda kaygı duymaya başladığına dair emareler vardı artık. Nisan 2018'de Birleşmiş Milletler bünyesinde-

ki bir Çin heyeti, ölümcül otonom silahların yasaklanması önerisinde bulundu; yine Nisan 2018 tarihli bir Birleşik Krallık Lordlar Kamarası Yapay Zekâ Komitesi raporunda, YZ sistemlerine asla insanlara zarar verme veya insan öldürme erki verilmemesi gerektiği ısrarla vurgulanıyordu.[17]

Şimdiye kadar ele aldığımız bütün bu nedenlerle, ölümcül otonom silahlar hem tehlikeli hem de etik açıdan sıkıntılıdır. Ottawa Antlaşması ile anti-personel mayınlarının geliştirilmesi, stoklanması ve kullanılmasının yasaklanması gibi,[18] ölümcül otonom silahların da benzer bir çerçevede yasaklanması gayet makul bir çözüm gibi geliyor bana. Yalnız bu o kadar kolay olmayabilir tabii: Bu silahları inşa etmek isteyenlerin baskısını bir yana bıraksak bile, bir yasağı *formüle etmek* zor olabilir. Bir önceki paragrafta değindiğimiz Birleşik Krallık YZ raporunda, ölümcül otonom silahın geçerli bir tanımını yapmanın ne denli zor olduğuna, bu durumun da mevzuat oluşturma önünde ciddi bir engel teşkil ettiğine dikkat çekiliyor. İşin pratiği açısından bakacak olursak, bir silahı sırf YZ tekniği kullanıyor diye yasaklamak çok zordur (pek çok konvansiyonel silah sisteminde YZ yazılımı şu veya bu biçimde zaten kullanılıyordur diye tahmin ediyorum). Dahası, nöral ağlar gibi belli başlı YZ teknolojilerinin kullanımını yasaklamanın işe yarar bir tedbir olduğundan pek emin değilim, çünkü yazılım geliştiriciler kullandıkları teknikleri kodlarının içine çok rahat gizleyebilirler.

Dolayısıyla, ölümcül otonom silahların geliştirilmesi ve kullanılmasını kontrol etmeye veya yasaklamaya yönelik kamusal ve siyasi bir irade ortaya koyulsa bile, bunu formüle edip yasalaştırmak ve yürürlüğe koymak zor olabilir. Ama neyse ki hükümetlerin bunu denemeye gönüllü olduğuna dair emareler var.

## Algoritmik Yanlılık

YZ sistemlerinin insan dünyasının yakasını bırakmayan önyargı ve yanlılıktan azade olmasını umarız ama ne yazık ki durum bu değildir. Son on yılda, makine öğrenmesi sistemleri başka başka alanlarda uygulanma imkânı bulup yayıldıkça, otomatikleştirilmiş karar

verme süreçlerinin **algoritmik yanlılık** sergileyebileceğini görmeye başladık. Böylece bu konu başlıca araştırma alanlarından biri haline geldi; günümüzde pek çok araştırma grubu algoritmik yanlılığın ne gibi problemler yarattığını ve bu problemlerden nasıl kaçınılabileceğini anlamaya çalışıyor.

Algoritmik yanlılık, adından da anlaşılacağı üzere, bir bilgisayar programının –sadece YZ sistemlerinin değil, herhangi bir bilgisayar programının– karar verirken bir biçimde yanlılık sergilediği durumlarla ilgilidir. Alanın önde gelen isimlerinden biri olan Kate Crawford yanlı programların iki tip zarara yol açabileceğinden söz eder: tahsis zararı ve temsil zararı.[19]

Bir grubun belli bir kaynaktan mahrum bırakılması (veya tam tersi, kaynağın dağıtımında kayırılması) tahsis zararına örnektir. Diyelim ki potansiyel müşterilerin iyi müşteri mi kötü müşteri mi olduğuna dair öngörüde bulunmaya çalışırken –borçlarını zamanında ödeyecekler mi vb.– YZ sistemlerini kullanmanın bankalara fayda sağlayacağı düşünülüyor. Bunun için, iyi ve kötü müşteri kayıtları kullanılarak bir YZ programı eğitiliyor ve bir yerden sonra bu sistem potansiyel müşterinin bilgilerine bakarak iyi mi yoksa kötü mü olduğunu tahmin edebilir hale geliyor – klasik bir makine öğrenmesi uygulamasından bahsediyoruz yani. Program yanlıysa, belli bir gruba özellikle kredi vermeyip başka bir gruba iltimas geçebilir. Dolayısıyla yanlılık burada ilgili grup için tespit edilebilen bir ekonomik dezavantaja (veya avantaja) neden olmaktadır.

Temsil zararı ise bir sistemin herhangi bir edimde bulunduğunda stereotipler veya peşin hükümler yaratması veya bunları pekiştirmesi halinde meydana gelir. Mahut bir örnek olarak, 2015'te bir Google fotoğraf sınıflandırma sistemi siyahi insanların fotoğraflarını –en berbat ırk stereotiplerini pekiştirecek şekilde– "goril" diye etiketlemişti.[20]

1. Bölüm'den hatırlayacağınız gibi, bilgisayarlar talimatlara uyan makinelerden başka bir şey değildir. Öyleyse nasıl yanlı olabilirler?

Her şeyden önce veri yoluyla. Makine öğrenmesi programları veri kullanılarak eğitilir ve bu veri yanlıysa, program veride örtük

biçimde bulunan yanlılığı öğrenir. Eğitim verisi çeşitli şekillerde yanlı olabilir.

İlk akla gelen olasılık, bizzat veri kümelerini inşa edenlerin yanlı olmasıdır. Bu durumda muhtemelen yanlı bakış açılarını veri kümesine yerleştirmiş olacaklardır. Yanlılık aşikâr veya bilinçli olmayabilir. İşin aslına bakarsanız, kendimizi ne kadar nesnel ve makul sansak da, *hepimiz* bir biçimde yanlıyızdır. Ve yanlı bakış açımız yarattığımız eğitim verilerinde kaçınılmaz olarak kendini gösterecektir.

Meselenin bu gayet insani boyutunun yanı sıra, makine öğrenmesi de –kasıtsızca– yanlılığa yol açabilir pekâlâ. Örneğin bir makine öğrenmesi programının eğitim verisi temsil bakımından yeterince kuşatıcı değilse, o program eninde sonunda çarpık sonuçlar verecektir. (Birkaç paragraf önceki kredi yazılımı örneğini düşünelim. Program tek bir coğrafi kesimden alınan bir veri kümesiyle eğitilecek olursa, başka kesimlerden kredi için başvuranlara karşı daha olumsuz sonuçlar verebilir.)

Pek iyi tasarlanmamış programlar da yanlı sonuçlar verebilir. Yine aynı kredi yazılımı örneğinde, programın eğitimi için seçilen verilerin başlıca özelliğinin etnik köken olduğunu düşünün. Bu durumda programın ne yazık ki son derece yanlı kararlar vermesi hiç de şaşırtıcı olmayacaktır. (Gerçek hayatta bir banka bu kadar aptalca bir şey yapmaz herhalde diye düşünürsünüz, değil mi? Ama ne yazık ki yanılıyorsunuz.)

Algoritmik yanlılık meselesinin bugün bilhassa öne çıkmasının nedeni, daha önce değindiğimiz üzere, güncel YZ dalgasındaki sistemlerin tam bir "karakutu" olması: Verdikleri kararları bir insanın yaptığı gibi *açıklayamıyor* veya *gerekçelendiremiyorlar*. İnşa ettiğimiz sistemlere fazla *güvenirsek*, bu sorun daha beter hale geliyor. Ve tam da böyle yaptığımızı gösteren anekdotlar var. Sözgelimi kredi veren bankamız sistemini inşa edip birkaç bin örnek üstünde deniyor; sistem insan uzmanlarla aynı kararı veriyor gibi görünüyorsa, tamam, doğru çalışıyor o zaman diye düşünerek sorgusuz sualsiz sisteme göre hareket etmeye başlıyorlar.

Yeryüzündeki her işletme problemlerini makine öğrenmesi yoluyla çözmek için delice bir yarışa girmiş gibi görünüyor. Fakat bu

hengâmede yanlı programlar yaratmaları çok muhtemel, çünkü burada ne gibi meselelerin söz konusu olduğunu pek anlamıyorlar. Bu meselelerin açık ara en önemlisi ise doğru veriyi elde etmek.

## Çeşitli(/siz)lik

1955'te, John McCarthy Rockefeller Enstitüsü'ne Dartmouth YZ yaz okulu projesini sunarken, elinde bu etkinliğe davet etmek istediği isimlerden oluşan 47 kişilik bir liste vardı. McCarthy 1996'da, "Bunlar Yapay Zekâ konusuyla ilgilenebileceğini düşündüğümüz insanlardı ama Dartmouth konferansına hepsi katılmadı," yazacaktı.

Şimdi, müsaadenizle, size bir soru sormak istiyorum: YZ'nin bilimsel bir disiplin olarak kuruluşuna tanıklık eden Dartmouth etkinliğine sizce kaç *kadın* katılmıştır?

Evet, doğru tahmin: *sıfır*. Günümüzde araştırmalara fon sağlayan saygın bir kuruluş, en temel çeşitlilik sınavını dahi veremeyen bu etkinliği desteklemeyi muhtemelen aklının ucundan bile geçirmeyecektir. Zira başvuru sahiplerinden eşitlik ve çeşitliliği nasıl sağlamayı planladıkları sorusuna açık ve net bir cevap istemek günümüzde standart bir uygulama haline gelmiş durumda artık. Yine de YZ'nin temellerinin böylesine erkek egemen bir ortamda atılmış olması çok çarpıcı. Üstelik, kitabın bu sayfasına gelene kadar çoktan fark etmişsinizdir, o zamandan sonra da büyük ölçüde erkek egemen bir disiplin olarak kalmış gibi görünüyor.

Bugün geri dönüp baktığımızda, o dönem ne gibi eşitsizlikler olduğunu hemen fark ediyor, bundan üzüntü duyuyoruz. Ne var ki, geçen yüzyılın ortalarında düzenlenmiş bir etkinliği, bugün hâlâ ulaşmayı arzuladığımız standartlara göre yargılamak pek olacak iş değil. Asıl şunu sormamız gerekiyor: YZ artık tamamen farklı bir durumda mı? İyi haberler yok değil ama genel tablo hâlâ karışık biraz. Bir taraftan, saygın bir uluslararası YZ konferansına gitseniz, elbette birçok kadın araştırmacı görürsünüz. Ama diğer taraftan, ezici çoğunluğun erkek olacağı da kesin gibidir. Çeşitsizlik, bilim ve mühendisliğin pek çok alanında olduğu gibi, YZ için de hâlâ kronik bir sorun.[21]

YZ camiasının toplumsal cinsiyet yapısı birbirinden farklı pek çok nedenle önemli. Bir kere, bu araştırmacılar topluluğunun erkek egemen olması, potansiyel bilimkadınları açısından teşvik edici bir özellik değil, dolayısıyla da alanı kıymetli yeteneklerden mahrum bırakıyor. Belki daha önemlisi, YZ sırf erkekler tarafından tasarlandıysa, ufku –tabiri caizse– *erkek YZ'si* ile sınırlı kalacaktır. Erkek YZ'si derken, erkeklerin inşa ettiği sistemlerin kaçınılmaz olarak tek bir dünya görüşünü, kadınları temsil etmeyen veya kucaklamayan bir dünya görüşünü cisimleştirmesini kastediyorum. Söylediklerimi abartılı buluyorsanız, Caroline Criado Perez'in *Görünmez Kadınlar*'ını muhakkak okumanızı öneririm.[22] Bu enteresan kitap resmen gözümü açtı. Perez burada esasen, dünyamızdaki hemen her şeyin tek bir toplumsal cinsiyeti yansıtan bir insan modeline, yani erkeklere göre tasarlanıp imal edildiğine dikkat çekiyor. Perez'e göre bunun temel nedeni –kendi tabiriyle– "veri boşluğu", yani imalat ve tasarım için rutin bir biçimde kullanılan veri kümelerinin ezici çoğunluğunun tarih boyunca hep erkek odaklı olması:

> Yazılı tarihin büyük kısmı koca bir veri boşluğudur. Geçmişin vakanüvisleri, erkek avcıdan ... itibaren, kadınların insanın evrimindeki rolüne pek az yer ayırdılar. ... Erkeklerin hayatları, bütün insanların hayatının temsilcisi sayıldı. İnsanlığın diğer yarısının hayatlarına gelince, genelde sessizlikten başka bir şey yok. ... Bütün bu sessizlikler, bütün bu boşluklar birtakım sonuçlara yol açıyor. Kadınların hayatlarını her allahın günü etkiliyorlar. Bu etki ufak olabiliyor. Sıcaklığı erkeklerin üşüme derecesine göre ayarlanmış ofislerde donmak veya yüksekliği erkeklerin ortalama uzunluğuna göre ayarlanmış üst raflara yetişmeye çalışmak gibi. ... Hayatınıza yönelik bir tehdit oluşturmuyor. Güvenlik önlemleri kadın ölçülerine göre düşünülmemiş bir arabayla kaza yapmaya benzemiyor mesela. Semptomlarınız "atipik" addedildiğinden geçirdiğiniz kalp krizinin teşhis edilememesine benzemiyor. Böyle durumlardaki kadınlar için, erkek verisine göre inşa edilmiş bir dünyada yaşamanın sonuçları ölümcül olabilir.

Criado Perez erkek odaklı tasarım ve erkek verisi probleminin nasıl da hayatın her yanına nüfuz ettiğini müthiş ayrıntılı bir biçimde belgeliyor. YZ için veri şüphesiz olmazsa olmaz ama veri kümelerindeki erkek yanlılığı da her yere yayılmış durumda. Bu yanlılık ba-

zen, konuşmayı anlama programlarının eğitiminde yaygın olarak kullanılan TIMIT söz verisi kümesindeki gibi, gayet açık oluyor. Söz konusu veri kümesinin %69'unun erkek sesi olması ister istemez konuşmayı anlama programlarının kadın seslerini yorumlamada erkek seslerine kıyasla daha kötü bir iş çıkarması sonucuna yol açar. Yanlılık kimi zaman bu kadar aşikâr değildir. Diyelim ki makine öğrenmesi programınızı eğitmek için bir dizi mutfak resmi topluyorsunuz ve bu resimlerde ağırlıklı olarak kadınlar var veya bir dizi şirket CEO'su fotoğrafı topluyorsunuz ve bu fotoğraflarda ağırlıklı olarak erkekler var. Sonuçta ne olacağını tahmin edersiniz. Dahası da var: Criado Perez'in altını çizdiği gibi, bunların ikisi de gerçekten oldu.

Bazen, yanlılık bir kültürün tamamında yerleşik olabilir. Bunun bednam bir örneği 2017'de görüldü: Google Çeviri'nin bir metni çevirirken kimi zaman cinsiyeti değiştirdiği fark edildi.[23] Anlaşıldı ki şu İngilizce cümleleri,

> He is a nurse. ["O bir hemşire" – *he* eril şahıs zamiri]
> She is a doctor. ["O bir doktor" – *she* dişil şahıs zamiri]

önce Türkçeye, ardından tekrar İngilizceye çevirince şu hale geldiğini görüyordunuz:

> She is a nurse. [Hemşireye dişil şahıs zamiri yakıştırılmış]
> He is a doctor. [Doktora eril şahıs zamiri yakıştırılmış]

Toplumsal cinsiyet klişelerini yeniden üretmenin âlâsı değil de nedir bu?

YZ'de yanlılık bir sorun, bilhassa kadınlar için, çünkü yeni YZ'nin üstüne kurulmakta olduğu temel çok güçlü bir biçimde erkeklerin tarafına meylediyor. Fakat yeni YZ'yi inşa eden ekipler bunu fark etmiyor bile çünkü aynı yanlılık bizzat onlarda da var.

YZ'nin önündeki başka bir zorluk da şu: İnsanınkine benzeyen birtakım özellikleri cisimleştiren sistemleri –örneğin Siri ve Cortana gibi yazılım ajanlarını– kazara toplumsal cinsiyet klişelerini pekiştirecek şekilde inşa edebiliriz. Her emrimizi sorgusuz sualsiz yerine getirecek itaatkâr YZ sistemlerini kadın gibi görünecek ve duyula-

cak şekilde inşa edersek, hizmet eden kadın stereotipinin propagandasını yapmış oluruz.[24]

## Yalan Haber

Yalan haber, adından da anlaşıldığı üzere, gerçek gibi sunulan yanlış, düzmece, yanıltıcı enformasyondur. Dijital çağdan önce de dünya yalan haber kaynağı bakımından zengin bir yerdi elbette ama internet, özellikle de sosyal medya gibi mecraların yalan haber propagandası için biçilmiş kaftan olduğu ortaya çıktı. Ve bunun çok çarpıcı sonuçları oldu tabii.

Sosyal medyanın olayı insanların birbiriyle iletişime geçmesini sağlamaktır. Modern sosyal medya platformları bu konuda olağanüstü başarılı. Batı'da Facebook ve Twitter, Çin'de WeChat'in kayıtlı kullanıcıları dünya nüfusunun kayda değer bir bölümünü oluşturuyor. Sosyal medya uygulamaları bu yüzyılın başında ortaya çıktığında, hepsi arkadaşlar, aile ve gündelik hayatla ilgili görünüyordu: Bolca çocuk fotoğrafı, bolca kedi fotoğrafı vardı. Ama çok geçmeden başka amaçlarla da kullanılmaya başladı. 2016'dan itibaren dünya çapında manşetlere taşınan yalan haber fenomeni, sosyal medyanın ne kadar güçlü olduğunu ve rahatlıkla küresel ölçekte olayları etkilemek için kullanılabileceğini gösterecekti.

Yalan haberin dünya çapında manşetlere çıkmasına neden olan iki olay vardı: Kasım 2016 ABD başkanlık seçimleri (sonunda Donald Trump seçildi) ve Haziran 2016 Birleşik Krallık Brexit referandumu (kıl payı bir farkla AB'den ayrılma kararı çıktı). Sosyal medya şüphesiz her iki kampanyada da önemli bir rol oynadı – Trump'ın siyaset tarzı veya davranış biçimi hakkında ne düşünülürse düşünülsün, onu destekleyenleri bir araya toplamak için sosyal medyayı ustaca kullandığını teslim etmek gerekiyor. Her iki örnekte de nihayetinde kazananların yalan haberlerle propagandasını yapmak için Twitter gibi sosyal medya platformlarının kullanıldığı öne sürülüyordu.

YZ yalan haberle propaganda yapmada kritik bir rol oynuyor. Bütün sosyal medya platformları onlarla etkileşime girerek zaman

geçirmenize bel bağlar, çünkü böylece size reklam göstererek para kazanabilirler. Platformda hoşunuza giden şeyler görürseniz daha fazla zaman geçirirsiniz veya tam tersi. Dolayısıyla sosyal medya platformlarının saiki size *hakikati* değil *hoşunuza gideni göstermektir*. Nelerden hoşlandığınızı nereden biliyorlar peki? "Beğen"e her tıklayışınızla bizzat siz söylüyorsunuz da oradan. Bu sayede platform benzer hikâyeler arayıp bulabiliyor ve muhtemelen onları da beğeniyorsunuz. İşini iyi yaparsa, nihayetinde sizin ve tercihlerinizin (acımasızca dürüst) bir resmi ortaya çıkıyor ve buna dayanarak da size hangi hikâyeleri göstereceğine karar veriyor.[25] *John Doe, beyaz erkek, şiddet içeren videoları beğeniyor ve ırkçı eğilimleri var...* Platform bu John Doe'ya, eğer "Beğen"mesini istiyorsa, sizce ne tür hikâyeler göstererecektir?

YZ'nin rolü beğendiğinizi söylediğiniz şeylere, yaptığınız yorumlara, tıkladığınız bağlantılara vb. bakarak neleri tercih ettiğinizi anlamak ve bu tercihler doğrultusunda aynı şekilde beğeneceğiniz yeni şeyler bulmaktır. Bunların hepsi klasik YZ problemleridir, bütün sosyal medya şirketlerinin söz konusu problemler üstünde çalışan araştırmacı ve geliştirmeci ekipleri olur.

Özetle platformlar nelerden hoşlandığınızı bulup size daha çok bu tür şeyler gösteriyor, başka bir deyişle beğenmeyeceklerinizi sizden saklıyorsa, sosyal medya John Doe'ya nasıl bir dünya resmi sunacaktır? Joe sadece şiddet içeren videolar izleyip ırkçı haberler okuyacak, yani dünyanın diğer yüzünü görmeyecektir; kendi sosyal medya balonunda kalacak, çarpık dünya görüşü pekişecektir. Buna teyit yanlılığı denir. Bunu böylesine kaygı verici hale getiren, çok geniş bir ölçekte meydana gelmesidir: Sosyal medya görüşleri *küresel ölçekte* manipüle ediyor ve, ister kasıtlı olsun ister tamamen tesadüf, bu manipülasyon kesinlikle kaygı verici.

Uzun vadede, YZ dünyayı algılama tarzımızı daha temel bir düzeyde değiştirmede rol oynayabilir. Tek tek her birimiz duyularımız (görme, işitme, dokunma, koklama ve tat alma) yoluyla dünya hakkında enformasyon ediniriz ve bu enformasyonu kullanarak hep beraber uzlaşıma dayalı bir gerçeklik görüşü, yani dünyanın aslında nasıl bir yer olduğuna dair yaygın kabul gören bir görüş inşa ederiz.

Belli bir olaya siz de ben de aynı şekilde tanıklık ediyorsak, ikimiz de olay hakkında aynı enformasyonu ediniriz ve bu enformasyonu da uzlaşımsal gerçekliğe ilişkin görüşümüzü şekillendirmek için kullanabiliriz. Peki ya ortak bir dünya görüşü olmadığında? Her birimiz dünyayı bambaşka bir biçimde algıladığımız zaman ne olacak? YZ ile o günleri de görebiliriz.

Google 2013'te Google Gözlük adıyla bir takılabilir bilgisayar teknolojisi piyasaya sürdü. Normal bir gözlüğü andıran ürün bir kamerayla ve küçük bir projektörle donatılmıştı; bluetooth aracılığıyla bir akıllı telefona bağlanıyordu. Ürün piyasaya sürülür sürülmez, "gizli" kamerası, uygunsuz durumlarda fotoğraf çekmek için kullanılması ihtimaliyle ilişkili bir tedirginlik yarattı. Kullanıcının görüş alanına bir görüntü yansıtan dahili projektörü ise ürünün asıl potansiyeli addediliyordu. Adeta sonsuz uygulama alanı vardı. Mesela benim isim hafızam sıfır, tanıdığım insanların isimlerini hatırlayamadığım için mahcup olduğum durumların haddi hesabı yok; Google Gözlük'ün baktığım kişiyi tanıyıp, etrafa fark ettirmeden o kişinin adını bana hatırlatması harika olurdu doğrusu.

Bu tür uygulamalara *artırılmış gerçeklik* deniyor; gerçek dünyayı alıp üstünü bilgisayarla üretilmiş enformasyonla veya görüntülerle kaplıyorlar. Peki ya gerçekliği artırmak yerine kullanıcının algılayamayacağı şekilde *tamamen değiştiren* uygulamalar? Örneğin, elinizdeki kitabın yazıldığı tarih itibarıyla oğlum Tom 13 yaşında. J. R. R. Tolkien'in *Yüzüklerin Efendisi* kitaplarına ve bunların film uyarlamalarına deli oluyor. Google Gözlük'ün okul ortamını değişik gösteren, arkadaşlarını elf gibi ve öğretmenlerini ork gibi görmesini sağlayan bir uygulaması olduğunu düşünün. Veya *Yıldız Savaşları* temalı bir uygulama vb. Sahiden eğlenceli olabilir ama şöyle de bir şey var: Her birimiz kendi dünyamızda yaşamaya başladığımızda uzlaşımsal gerçeklik diye bir şey kalır mı? Sizin ve benim ortak deneyimlerimiz olmaz artık, dolayısıyla ortak deneyim temelinde bir mutabakata da varamayız. Ayrıca böyle deneyimler *hacklenebilir* elbette. Tom'un gözlüğünün hacklendiğini düşünün; görüşlerinin *doğrudan* manipüle edildiğini, böylece dünyayı algılama tarzının temelden değiştiğini.

Bugün böyle uygulamalar yoksa da önümüzdeki 20-30 yılda mümkün hale gelmeleri çok olası görünüyor. Halihazırda, insana tamamen gerçek gibi gelen ama aslında baştan sona bir nöral ağ tarafından inşa edilmiş olan görseller üretebilen YZ sistemlerimiz var. Sözgelimi, elinizdeki kitabın yazıldığı tarih itibarıyla, DeepFake'ler konusunda ciddi kaygılar var.[26] DeepFake bir nöral ağın yarattığı değiştirilmiş fotoğraf veya videolardır. Bunun bednam bir örneğini 2019'da görmüştük; ABD Temsilciler Meclisi Sözcüsü Nancy Pelosi'nin bir videosu değiştirilerek, Pelosi konuşma bozukluğundan muzdaripmiş veya sarhoşmuş ya da uyuşturucu etkisi altındaymış gibi gösterilmişti.[27] DeepFake'ler pornolarda da kullanılıyor, sözgelimi bir videoya bambaşka "oyuncular"ın yüzleri ekleniyor.[28]

Günümüz itibarıyla DeepFake videoların kalitesi o kadar iyi değil, ama iyileşiyor, yakında bir fotoğraf veya videonun gerçek mi yoksa DeepFake mi olduğunu ayırt edemeyebiliriz. İşte o zaman, fotoğraflar veya videolar olayların güvenilir kayıtları olmaktan çıkacak. Her birimiz kendi YZ güdümlü dijital dünyamızda yaşamaya başlarsak, ortak değerler ve ilkeler temeline dayanan toplumların paramparça olması tehlikesi kapımıza dayanacak. Sosyal medyadaki yalan haberler daha başlangıç.

## Sahte YZ

4. Bölüm'de, 1990'larda ajanlar konusunda yürütülen araştırmaların, bu yüzyılın ilk on yılında Siri, Alexa ve Cortana gibi yazılım ajanlarıyla sonuçlandığını görmüştük. Siri çıktıktan kısa süre sonra, popüler basında sistemin belgelenmemiş kimi özellikleriyle ilgili bir sürü haber vardı. Mesela "Siri sen benim en iyi arkadaşımsın" gibi bir şey söylediğinizde, Siri buna anlamlı bir cevap gibi görünen bir karşılık veriyordu (demin ben de denedim: "Dostun olarak, iyi günde kötü günde hep yanındayım Michael" cevabını aldım). Basın merak içindeydi: Bu Genel Zekâ mıydı? Tabii ki hayır. Tahmin edeceğiniz üzere, Siri'nin tek yaptığı, birtakım anahtar kelimeler ve ifadelere verilmek üzere hazırlanmış basmakalıp cevaplardan birini bulunduğu yerden çekip çıkarmaktı, dolayısıyla onyıllar evvel ELI-

ZA'nın yaptığından çok da farklı bir şey yapmıyordu aslında. Ortada zekâ mekâ yoktu yani. Anlamlı gibi görünen bu cevaplar sahte YZ'den başka bir şey değildi.

Sahte YZ üretme kabahatini işleyen tek şirket Apple değil. Ekim 2018'de Birleşik Krallık hükümeti Pepper adındaki bir robotun mecliste soruları cevaplayacağını duyurdu.[29] Ama gerçekte olanların bununla alakası yoktu tabii. Evet, böyle bir robot vardı (yeri gelmişken, müthiş araştırmaların ürünü olan sağlam bir robottur Pepper) ama tek yaptığı önceden belirlenmiş sorulara önceden hazırlanmış cevapları vermekti, ELIZA düzeyinde bile değildi yani.

Siri veya Pepper örneklerinde aldatma kastı yoktu, ikisi de eğlenceli tarafları olan iyi niyetli girişimlerdi. Ama YZ camiasından çok insanı kızdırdılar, çünkü YZ'nin ne olduğuna dair çok yanlış bir tablo çiziyorlardı. Pepper'ın yaptığı gösteriyi izleyip de *gerçekten* sorulan soruları cevapladığına inanan biri, o dönemde YZ tabanlı soru-cevap sistemlerinin neler yapabileceğine dair çok yanlış fikirlere kapılmış olmalı. Yine de çoğu insanın buna inandığını sanmıyorum, çünkü gerçekte ne olup bittiğini anlamak o kadar da zor değildi. Öyleyse şöyle bir problemle karşı karşıyayız demektir: İnsanlar YZ'nin böyle saçmalıklardan ibaret olduğunu, dahası düpedüz sahtekârlık olduğunu düşünebilirler.

Başka bir olay da Aralık 2018'de Rusya'nın Yaroslavl şehrinde yaşandı: Gençlik bilim forumunda dans eden "ileri teknoloji robot" un aslında robot kostümü giymiş bir insan olduğu ortaya çıktı.[30] Etkinliği düzenleyenlerin ne niyetle böyle bir şey yaptığını bilmediğimizden, onları şaibe altında bırakamayız. Fakat Rusya devlet televizyonunun bunu gerçekten de bir "robot"muş gibi takdim ettiği açıktı. Gösteriyi izleyen biri bunun günümüzde robotik ve YZ'nin geldiği son noktayı temsil ettiği sonucuna pekâlâ varabilirdi.

Ne yazık ki bu tür sahtekârlıklara YZ'nin çeperlerinde çok sık rastlanıyor ve bu durum da YZ araştırmacılarında büyük hüsrana yol açıyor. Ekim 2017'de, Suudi Arabistan Sophia adındaki bir robota vatandaşlık verdiğini duyurdu. Haliyle medyanın çok ilgisini çekti bu haber.[31] Pek çok yorumcu insan hakları, özellikle de kadın hakları konusundaki sicili hiç de parlak olmayan Suudi Arabistan'ın bir ro-

bota vatandaşlık vermesinin ne kadar ironik olduğuna dikkat çekiyordu. Pek çoğu da burada temelden yeni bir şey olup olmadığını bilmek istiyordu: Bu robot genel zekâ yolunda atılmış bir adım mıydı?

Hanson Robotics şirketinin inşa ettiği Sophia bir insansı robot; insan gibi görünüyor, yüz ifadeleri gerçekçi ve tabii doğal dilde konuşma yetisine sahip. Aralık 2017'de Business Insider web sayfasında Sophia'yla yapılan bir "röportaj" yayımlandı (italikler Sophia'nın cevapları):[32]

Bize duygularınızdan ve sevdiğiniz şeylerden bahseder misiniz biraz?
*Sizinle aynı evde yaşayan veya aynı işyerinde çalışan bir robot oldu mu hiç?*

Hayır.
*Belki henüz farkında değilsiniz ama hayatınızda muhtemelen sandığınızdan daha fazla robot var. Günün birinde bir robotla yaşamak veya çalışmak ister miydiniz?*

Şu anda ne tür robotlarla birlikte yaşıyor veya çalışıyorum?
*Aynen.*

"Aynen" bilmediğiniz bir şeyle karşılaştığınızda verdiğiniz standart cevap mı?
*Evet.*

Facebook'un YZ araştırma direktörü Yann LeCun pek etkilenmişe benzemiyordu. Ocak 2018'de "Gerçek sihir için hokkabazlık neyse, YZ için de bu odur. ... Kusura bakmayın ama bunların hepsi fasa fiso" diye bir tweet atmıştı. Bu sözler, anaakım YZ camiasının böyle ucuz numaralara karşı tavrını iyi özetliyor.

Öyleyse sahte YZ'yi, sahne arkasında aslında insan zihinleri varken, bizi yanlış yönlendirip gördüğümüzün YZ olduğuna inandırmaya çalışan şey olarak tanımlayabiliriz. Kritik bir demo öncesi, tanıtacağı teknolojiyi bir türlü çalıştıramayan YZ girişim şirketlerinin, sahne arkasında sahte YZ'ye başvurduğuna dair söylentiler geliyor kulağıma. Bu uygulamanın ne denli yaygın olduğunu bilmiyorum ama sahte YZ herkes için gerçek bir problem, YZ çevreleri için de ciddi bir rahatsızlık konusu.

# 9

# Bilinçli Makineler?

BU KİTAPTA SADECE BİR ŞEYİ yapmayı becerebildiysem eğer, umarım o da sizi YZ ve makine öğrenmesindeki yakın dönem atılımların –son derece gerçek ve heyecan verici olmakla beraber– Genel YZ kapısını açan sihirli anahtar olmadığına ikna etmektir. Derin öğrenme Genel YZ'nin önemli bir bileşeni olabilir belki ama kesinlikle tek bileşeni değildir. Aslına bakarsanız diğer bileşenlerin neler olduğunu, hele ki Genel YZ tarifinin neye benzeyebileceğini henüz bilmiyoruz. Geliştirdiğimiz birbirinden etkileyici bütün kapasiteleri –görsel tanıma, çeviri, sürücüsüz otomobiller– toplasak bir genel zekâ etmiyor. Bu bakımdan, Rod Brooks'un ta 1980'lerde gündeme getirdiği problem hâlâ önümüzde duruyor: Zekânın bazı *bileşenleri* elimizde var ama bunları bütün haline getiren bir sistemi nasıl inşa edeceğimiz konusunda en ufak bir fikrimiz yok. Ve her halükârda, kimi önemli bileşenler hâlâ eksik. 5. Bölüm'de gördüğümüz gibi, günümüzün en iyi YZ sistemleri bile ne yaptığını *idrak ettiğine* dair manidar bir emare göstermez. Yaptıkları işi ne kadar iyi yapıyor olurlarsa olsunlar, dar sınırları olan spesifik bir görevi yerine getirmek üzere optimize edilmiş yazılım bileşenlerinden öte bir şey değillerdir.

Genel YZ'ye daha çok yolumuz olduğunu düşündüğüm için, haliyle güçlü YZ ihtimallerine, yani tıpkı bizim gibi bilinçli, özfarkındalığa sahip, tam anlamıyla otonom varlıkların mümkün olduğu fikrine daha şüpheyle yaklaşmam lazım. Yine de bir bölümde bari kendimizi biraz şımartalım ama değil mi: Güçlü YZ yakın görünmese

de, bu konu üstüne düşünerek, o istikamete doğru nasıl ilerleyebileceğimize dair spekülasyon yaparak biraz eğlenebiliriz. Öyleyse gelin bilinçli makinelere giden yola koyulup biraz gezelim. Yol boyunca neler görebiliriz, ne gibi engellere takılabiliriz, nasıl manzaralarla karşılaşabiliriz, beraber bunları hayal edelim. Ve bir o kadar önemlisi, yolun sonuna yaklaştığımızı nasıl bilebiliriz, bunu konuşalım.

## Bilinç, Zihin ve Daha Nice Muamma

1838'de İngiliz biliminsanı John Herschel Güneş'in ne kadar enerji yaydığını bulmak için basit bir deney yaptı. Doğrudan güneş alan bir yere bir kap su yerleştirdi ve güneş enerjisinin ne kadar zamanda bu suyun sıcaklığını bir santigrat derece artırdığını ölçtü. Bu sayede, basit bir hesapla, Güneş'in saniyede ne kadar enerji yaydığına dair yaklaşık bir tahminde bulunabiliyordu. Sonuçları insanın aklı hayali almıyordu: Güneş tek bir saniyede, Dünya'nın koca bir yılda yarattığından misliyle daha fazla enerji yayıyordu. Öyleyse bilimsel bir muamma vardı ortada: O dönem itibarıyla jeolojik kanıtlar Dünyamızın (ve dolayısıyla Güneşimizin) yaşının en azından onlarca milyon yıl olduğunu gösteriyordu ama bilindiği kadarıyla hiçbir fiziksel süreç yoktu ki Güneş'e bu kadar uzun süreler enerji sağlayabilsin. Bilinen herhangi bir enerji kaynağıyla Güneş'in olsa olsa birkaç bin yıl içinde sönüp gitmesi lazımdı. Dönemin biliminsanları, Herschel'ın kolayca tekrar edilebilen deneyinin kanıtlarını coğrafi kayıtların sunduğu kanıtlarla bağdaştırabilmek için çırpınırken, birbirinden mantıksız teoriler icat ettiler. Nihayetinde ancak 19. yüzyılın sonlarında ve nükleer fiziğin ortaya çıkışıyla, biliminsanları atom çekirdeğinde ne menem güçlerin gizli olduğunu anlamaya başladı. Herschel'ın deneyinden tam bir asır sonra ise fizikçi Hans Bethe artık yaygın olarak kabul gören yıldızlarda nükleer füzyon yoluyla enerji üretimi izahını ortaya koydu.[1]

Güçlü YZ'ye, az çok bizimki gibi bilinçli zihinleri, özfarkındalığı ve anlama yetisi *gerçekten* olan makineler inşa etme amacına dönecek olursak, Herschel'ın zamanındaki biliminsanlarından çok

da farklı bir pozisyonda değiliz. Çünkü Herschel'ın zamanındaki biliminsanları açısından Güneş'e enerji sağlayan kuvvetler nasıl bir muamma idiyse, insanda zihin ve bilinç fenomenleri de –nasıl evrildiler, nasıl işliyorlar, hatta davranışlarımızda nasıl bir rol oynuyorlar– bizim açımızdan tam bir muamma. Henüz bu soruların hiçbirine cevabımız yok ve yakında da olacakmış gibi görünmüyor; birkaç ipucu, bolca da spekülasyon – olup olanı bu. Bilim sayesinde evrenin kökenini ve geleceğini anlayabilirsek, tatmin edici cevaplarımız olacak. Ama bunun henüz gerçekleşmemiş olması güçlü YZ'ye yaklaşmamızı da güçleştiriyor: Nereden veya nasıl başlayacağımız konusunda en ufak bir fikrimiz bile yok.

Aslına bakılırsa durum bundan da kötü: "Bilinç", "zihin" ve "özfarkındalık" lafları ağzımdan düşmüyor, halbuki bunların ne olduğunu tam anlamıyla bildiğimiz falan yok. Bütün bu kavramlar malumun ilamı gibi *görünüyor* –nihayetinde hepsini deneyimliyoruz– ama onlar hakkında konuşmamızı sağlayan bilimsel aygıtlar yok elimizde. Hatta, sağduyu aracılığıyla şahsi deneyimlerimizden edindiğimiz onca kanıta rağmen, bilimsel açıdan kesinlikle gerçek olduklarını bile söyleyemeyiz. Herschel ele aldığı probleme bir deney aracılığıyla, sıcaklık ve enerji gibi gayet iyi anlaşılan ve ölçülebilir olan fizik kavramlarıyla yaklaşıyordu. Bilinç veya zihni incelemek için böyle testlerimiz yok, ikisini de nesnel gözleme veya ölçüme tabi tutamıyoruz. Zihni veya öznel deneyimi ölçmek için standart bir bilimsel birim yok, hatta bunları doğrudan ölçmenin bile bir yolu yok: *Ben* istesem de *sizin* ne düşündüğünüzü veya deneyimlediğinizi göremiyorum. Tarihsel olarak, insan beyninin yapısı ve işleyişine dair bildiklerimizin çoğu, beyni hastalık veya travma nedeniyle bir şekilde zarar görmüş kişiler üstünde yapılan çalışmalardan geliyor. Ama bundan kolay kolay sistematik bir araştırma programı çıkmayacağı ortada. Manyetik rezonans görüntüleme (MRI) gibi beyin görüntüleme teknikleri beynin yapısı ve işleyişine dair önemli bilgiler verse de, bireyin öznel deneyimlerine ulaşmamızı sağlamıyor.

Ama açık ve net tanımlarımız olmasa da, bilinç hakkındaki çeşitli tartışmalarda karşımıza çıkan birtakım ortak özellikler var.

Belki de en önemli fikir, bilinçli olanın, *şeyleri öznel bir biçimde*

*deneyimleyebilmesi* gerektiği fikridir. Öznel derken, *şahsi zihinsel perspektifi* kastediyorum. Bunun önemli veçhelerinden biri, felsefecilerin tabiriyle, ***qualia***'dır. Havalı bir isim, ama altında yatan fikir aslında çok basit: hepimizin deneyimlediğimiz üzere, içsel zihinsel fenomenlerin duyumsanması. Mesela kahvenin kokusu; durup bir düşünün bu kokuyu veya iyisi mi kalkıp kendinize güzel bir fincan kahve yapın; o aromayı iyice içinize çekin. İşte deneyimlediğiniz bu duyum *qualia*'ya bir örnektir. Cehennem gibi sıcak bir günde buz gibi bir birayı yudumlama deneyimini, kıştan bahara hava geçişlerinin verdiği duyguyu veya bir anne ya da babanın çocuğunun yeni bir şey başardığını görmesinin ne hissettirdiğini düşünün. Hepsi *qualia*'dır. Bunlar pekâlâ bildik gelebilir size, bizzat kendiniz de yaşamışsınızdır. Ama burada şöyle bir paradoks var: Aynı deneyimlerden bahsettiğimizi sanıyoruz, halbuki bunları gerçekten aynı şekilde deneyimleyip deneyimlemediğimizi bilmemize imkân yok, çünkü *qualia*–ve diğer zihinsel deneyimler– tabiatı itibarıyla *kişiye özeldir*. Kahveyi kokladığım*da* yaşadığım zihinsel deneyim sadece *benim* erişebileceğim bir şeydir, *sizin* yaşadığınızın da benzer bir deneyim olup olmadığını söylememe imkân yoktur, söz konusu deneyimi anlatmak için aynı kelimeleri kullanıyor oluşumuz bu gerçeği değiştirmez.

Bilinç tartışmasına yapılan pek çok katkıdan en bilineni 1974'te Amerikalı felsefeci Thomas Nagel'dan geldi.[2] Bir şeyin bilinçli olup olmadığını tespit etmeye yönelik bir testti bu. Diyelim ki şunların bilinçli olup olmadığını bilmek istiyorsunuz:

- insan
- orangutan
- köpek
- fare
- solucan
- ekmek kızartma makinesi
- kaya

Nagel'ın testinde, yukarıdaki varlıklara uygulandığında, "*X* olmak nasıl bir şey?" sorusunun anlamlı olup olmadığına bakılır. Eğer *X*

olmayı bir şeye benzetebiliyorsak eğer, o zaman *X* bilinçlidir. Yukarıdaki listeye dönecek olursak, insanlara uygulandığında bu soru anlamlıdır. Peki ya orangutanlar? Sanırım yine anlamlı. Aynı durum köpekler ve –bu kadar bariz değilse de– fareler için de geçerli. Demek ki Nagel'ın testine göre orangutanlar, köpekler ve fareler bilinçlidir. Fakat bu elbette ki bilinçli olduklarının "kanıtı" değildir. Burada nesnel kanıttan çok sağduyuya güvenmek zorundayız.

Peki ya solucanlar? Bu örnekte bilincin eşiğindeyiz sanırım. Solucanlar o kadar basit canlılar ki Nagel'ın sorusunun anlamlı olup olmadığı (en azından bana) şüpheli görünüyor, dolayısıyla da testten solucanların bilinçli olmadığı sonucu çıkıyor. Bu örnek aynı zamanda başka bir tartışmaya alan açıyor: Solucanların bile bir tür ilkel bilinci olduğu öne sürülebilir, fakat bu durumda da söz konusu bilincin ancak insanınkine kıyasla ilkel olarak nitelenebileceğinin altını çizmek zorundayım. Buna karşılık ekmek kızartma makineleri ve kayalar konusunda tartışmalı bir durum yok, testi geçemedikleri aşikâr.

Nagel'ın testi birçok önemli noktayı vurguluyor.

Birincisi, *bilinç ya vardır ya yoktur* diye değerlendirilemez. Çünkü bilinç bir spektrumdur. Bir ucunda tam gelişmiş insan bilinci vardır, öbür ucunda solucanlar. Ayrıca insanlar arasında bile fark vardır. Hatta bir insanın bilinç düzeyi alkol veya uyuşturucu etkisi altında olmasına ya da zinde veya yorgun olmasına göre bile çeşitlilik gösterir.

İkincisi, bilinç varlıktan varlığa *farklılık* gösterir. Nagel'ın makalesinin başlığı "Yarasa Olmak Nasıl Bir Şey?"di. Başlık için özellikle yarasaları seçmiş olmasının nedeni ise bu hayvanların insanlarla uzaktan yakından alakası olmamasıydı. Soru yarasalara uygulandığında şüphesiz anlamlıdır, dolayısıyla bilinç testini geçerler. Diğer taraftan yarasalarda bizde olmayan duyular, özellikle de bir tür sonar vardır. Uçarken ultrasonik sesler yayar, bu tiz seslerin yankılarını tespit ederek etrafı algılarlar. Kimi yarasaların Dünya'nın manyetik alanını bile algılayabildiği ve bunu navigasyon için kullandığı biliniyor, yani bu canlılar kendi kendilerinin manyetik pusulası durumundalar. Bizim böyle duyularımız olmadığından, yarasalığın nasıl bir şey olduğunu gerçekten bilemeyiz. Yarasa bilinci

insan bilincinden farklıdır. Nagel yarasanın bilinçli olduğundan pekâlâ emin olsak bile, bunun kavrama yetimizi aştığını düşünüyordu.

Nagel'ın başlıca amacı bilinci sınamak için bir test ortaya koymak ("*X* olmak nasıl bir şey?"), bilincin birtakım biçimlerinin kavrayışımızı aştığını göstermekti (yarasalığın nasıl bir şey olduğunu tahayyül edemeyiz). Fakat bu test bilgisayarlara da uygulanabilir: İnsanların çoğu, tıpkı ekmek kızartma makinesi olmak gibi, bilgisayar olmanın da *hiçbir şeye benzemediğini* düşünüyormuş gibi görünüyor.

Bu nedenle Nagel'ın "nasıl bir şey" argümanı, güçlü YZ'nin mümkün olduğu fikrine karşı öne sürülmüştür. Buna göre güçlü YZ mümkün değildir çünkü bilgisayarlar Nagel'ın testini geçemez. Şahsen bu argümanı ikna edici bulmuyorum, çünkü "... olmak nasıl bir şey" diye sormanın sezgilerimize başvurmaktan zerrece farkı yok gibi geliyor bana. Sezgilerimiz çok bariz örneklerde –orangutanlar ve ekmek kızartma makineleri– işe yarayabilir; buna karşılık birazcık daha incelikli örneklerde veya YZ gibi kendi doğal dünya deneyimimizin çok dışında kalan örneklerde, yine aynı şekilde güvenilir bir kılavuz sayılmaları için bir neden görmüyorum. Belki de bilgisayarlığın nasıl bir şey olduğunu sırf bizden tamamen farklı oldukları için tahayyül edemiyoruz. Bu (en azından bana göre) makine bilincinin ille de imkânsız olduğu değil sadece daha *farklı* olduğu anlamına gelir.

Nagel'ın argümanı güçlü YZ'nin imkânsız olduğunu göstermek için öne sürülen pek çok argümandan yalnızca biridir. Gelin bunların en bilinenlerine beraberce bir göz atalım.

## Güçlü YZ İmkânsız mı?

Nagel'ın argümanı, güçlü YZ'nin mümkün olmadığı çünkü *insanlarda özel bir şey olduğu* yolundaki sağduyuya dayalı itirazla yakından ilişkilidir. Bu sezgisel cevabın başlangıç noktasını oluşturan görüş, insanların canlı bilgisayarların ise cansız olması bakımından ikisinin birbirinden ayrıldığı görüşüdür. Buna göre benim bir bilgisayara kıyasla bir fareyle daha çok ortak yanım vardır, bir bilgisa-

yarın bana kıyasla bir ekmek kızartma makinesiyle daha çok ortak yanı vardır.

Bu argümana cevabım şu: İnsanlar ne kadar olağanüstü olursa olsun, nihayetinde bir grup atomdan başka bir şey değildir. İnsanlar ve beyinleri fiziksel varlıklardır, fizik kanunlarına tabidir; söz konusu kanunları henüz tam anlamıyla bilmiyor oluşumuz bu gerçeği değiştirmez. İnsanlar olağanüstü, harika, inanılmaz şeyler, ama evrenin ve kanunlarının perspektifinden bakınca hiçbir özel tarafımız yok. Yine de cevabım o zor soruya yanıt olmuyor tabii: Bir grup atom *nasıl* olup da bilinç deneyimine yol açmıştır? Bu noktaya döneceğiz.

"İnsanlar özeldir" argümanının bir varyasyonunu da Amerikalı felsefeci Hubert Dreyfus ortaya atmıştır. Dreyfus'ın YZ eleştirisinin öne çıkan noktalarından biri, fiili başarıları göz önünde bulundurulunca, YZ'nin fazla abartıldığıdır ki bunun da şüphesiz haklılık payı vardır. Dreyfus'ın güçlü YZ'nin mümkün olduğu fikrine karşı daha somut bir argümanı da vardır. Buna göre, insanın eylem ve kararları büyük ölçüde "sezgi"ye dayanır ve sezgi de bilgisayarların gerektirdiği kesinlikte formülleştirilemez. Kısacası Dreyfus'a göre insan sezgisi bir bilgisayar programı gibi bir tarife indirgenemez.

Çoğu kararımızın açıkça veya titizlikle akıl yürütmeye dayanmadığını gösteren bir sürü örnek var.[3] Kararın gerekçesini formülleştirme pek çok durumda karar verme sürecimizin bir parçası değildir, hatta belki de verdiğimiz kararların büyük bölümü bu türdendir. Bu bakımdan, karar verirken bir tür sezgiye başvurduğumuz açıktır. Fakat bu elbette zamanla kazandığımız (veya belki de evrim yoluyla miras aldığımız, genlerimizle bize aktarılan) deneyimin, bilinç düzeyinde ifade edemiyorsak da bizim için bir muamma olmayan deneyimin bir sonucudur. Bilgisayarların deneyimden öğrenebildiğini görmüştük, öyleyse kararlarının gerekçesini ifade edememekle beraber bilgisayarlar da etkili bir biçimde karar verebilir demektir.

Güçlü YZ'nin mümkün olduğu fikrine karşı en bilinen argümanı, bizzat güçlü YZ ve zayıf YZ terimlerinin de isim babası olan, felsefeci John Searle ortaya atmıştır. Searle bunu **Çince odası** senaryosu üstünden anlatır:

Diyelim ki bir odada tek başına bir adam var, çalışıyor. Derken kapıdaki gözden içeri birtakım kartlar bırakılıyor. Kartlarda Çince sorular yazılı, adam Çince bilmiyor, ama elinde nasıl Çince bir cevap yazabileceğine dair direktifler var. Direktiflere bakarak yazdığı cevabı yine aynı şekilde geri gönderiyor. Bu noktada, oda (ve içindekiler) bilfiil bir Çince Turing testine girmiş oluyor; odadan gelen cevaplar, en başta soruları gönderen kişiyi, testin öznesinin bir bilgisayar değil de insan olduğuna ikna ediyor.

Şimdi, kendimize şu soruyu soralım: *Bu örnekte herhangi bir şekilde Çincenin anlaşılması söz konusu mudur?* Searle'ün cevabı hayırdır. Adam Çince bilmiyor, e oda da Çince bilmiyor zaten, nereye bakarsak bakalım Çincenin anlaşıldığına dair bir emare yok. Adamın yaptığı, soruya cevap hazırlamak için açık ve net bir tarife dikkatle uymaktan ibaret; insan zekâsını sadece bu ölçüde kullanıyor.

Gördüğünüz gibi adam tam olarak bir bilgisayarın yapacağını yapar, bir dizi talimata, bir tarife uyar. Yürüttüğü "program", kendisine verilen talimatlardır. İşte Searle bütün bunlara dayanarak Turing testini geçen bir bilgisayarın anlama yetisi sergilemeyeceğini öne sürer.

Searle'ün argümanı doğruysa, idrak –dolayısıyla güçlü YZ– bir tarife uyarak üretilemez demektir. Öyleyse güçlü YZ'ye bildiğimiz türden bilgisayarlarla ulaşamayız. Eğer doğruysa, bu basit argüman o büyük YZ hayalinin sonu olur. Programınız istediği kadar anlama yetisine sahipmiş gibi *görünsün*, olsa olsa bir illüzyondur bu: Örtüyü hızlıca çektiğinizde altında hiçbir şey olmadığını görürsünüz.

Searle'ün eleştirisine karşı pek çok argüman öne sürülmüştür.

Bunların en bariz ve sağduyuya dayalı olanı, Searle'ün Çince odasının en basitinden imkânsız olduğuna işaret etmektir. Çünkü, her şey bir yana, insanı bilgisayar işlemcisi gibi konumlandırırsanız, bilgisayarın bir saniyede yerine getirdiği program talimatlarını uygulaması binlerce yıl sürecektir. Ayrıca söz konusu programın yazılı talimatlar şeklinde kodlanabileceğini düşünmek de abestir. Günümüzde tipik bir geniş çaplı bilgisayar programı yüz milyonlarca satır bilgisayar kodundan meydana gelmektedir. (Bu kodları bir insanın okuyup anlayabileceği biçimde yazmak ise on binlerce cilt kitaba

eşdeğer uzunlukta olarak düşünülebilir.) Bir bilgisayar bellekten talimatları mikrosaniyeler içinde alabilirken, Çince odası bilgisayarı milyarlarca kat daha yavaş olacaktır. Uygulamada böyle durumlarla karşılaşılacağını göz önünde bulundurarak, Çince odasının ve içindekilerin soruyu soran kişiyi içeridekinin bir insan olduğuna ikna *edemeyeceğini*, yani aslında Turing testini *geçemeyeceğini* söyleyebiliriz.

Başka bir karşı argüman ise, odadaki adam ve odanın kendisi anlama yetisi sergilemezken, adamı, odayı, direktifleri vb.ni içeren *sistemin* bu yetiyi *sergilediği* yolundadır. İnsan beynini didik didik edip içinde idrak arasak, bulamazdık. İnsan beyninde dili anlamaktan sorumlu görünen alanlar olduğuna şüphe yok, fakat buralarda Searle'ün beklediği türden bir anlama yetisi bulamayız.

Searle'ün yaratıcı düşünce deneyine bunlardan çok daha basit bir nedenle itiraz edilebileceği kanaatindeyim. Çince odası senaryosunun bir tür Turing testi gibi sunulmasında bir üçkâğıt var: Burada oda bir karakutu gibi değil. Çince odasında idrak olmadığını ancak içine bakarsak iddia edebiliriz. Halbuki Turing testine göre ısrarla sadece girdiler ve çıktılara bakmamız ve tanık olduğumuz davranışın insan davranışından ayırt edilip edilmediğini sormamız gerekiyordu. Eğer bir bilgisayarın yaptığı şey insan idrakinden ayırt edilemiyorsa, bilgisayar bunu "gerçekten" anlıyor mu yoksa anlamıyor mu diye bir argümana takılıp kalmak anlamsız geliyor bana.

Başka bir itiraz ise şu olabilir: Matematiksel olarak kanıtlanabilir sınırları nedeniyle, bildiğimiz türden bilgisayarlarla zekâyı hesaplamak belki de mümkün değildir. Hatırlayacak olursanız, Turing'in çalışmalarından, bilgisayarların ne yapıp ne yapamayacağı konusunda çok temel sınırlar olduğunu öğrenmiştik. Bilgisayarlar son derece açık bir biçimde tanımlanmış olmasına rağmen kimi problemleri çözemiyordu. Peki ya *YZ'nin ulaşmak için bu kadar büyük gayret gösterdiği zekâya dayalı davranış Turing'in öne sürdüğü anlamda hesaplanabilir değilse*? Turing bunu güçlü YZ'ye karşı olası bir argüman olarak bizzat tartışmıştır. YZ araştırmacılarının çoğu bu meseleyle ilgilenmez ama, daha önce birçok vesileyle gör-

düğümüz üzere, neyin *pratikte* hesaplanabilir olduğu problemi, tarihi boyunca YZ için hep büyük bir engel oluşturmuştur.

## Zihin ve Beden

Gelin şimdi de bilincin incelenmesinde karşılaşılan felsefi problemlerin en bilinenini, **zihin-beden problemi**ni ele alalım. İnsan bedeni ve beynindeki birtakım fiziksel süreçler bilinçli zihne yol açar. Peki ama bunu tam olarak nasıl ve neden yaparlar? Nöronlar, sinapslar ve aksonların fiziksel dünyası ile bizim bilinçli öznel deneyimimiz arasında ne gibi bir ilişki var? Bilim ve felsefenin en büyük, en eski problemlerinden biri olduğu için, Avustralyalı felsefeci David Chalmers bunu **zor problem olarak bilinç** şeklinde ifade ediyor.

Konuyla ilgili literatür en azından ta Platon'a kadar uzanıyor. Platon'un *Phaidros*'unda ortaya koyduğu insan davranışı modeline göre, beynin muhakemeden sorumlu bileşeni, iki atın dizginlerini elinde tutan bir arabacı gibi hareket ediyordu. Bu atlardan biri rasyonel, asil arzuları, diğeri ise irrasyonel, aşağılık arzuları temsil ediyordu. İnsanın şu hayatta nasıl bir yol tutturacağı, arabacısının iki atı nasıl kontrol ettiğine bağlıydı. Hiduizmin *Upanişad*'larında[4] da benzer bir fikirle karşılaşıyoruz.

Rasyonel benlik için pek hoş bir metafor bu (zihnimin her iki kısmının da dizginlerinin bizzat benim elimde olduğu fikri açıkçası çok daha cezbedici geliyor bana) fakat zihin teorilerinin ortak bir probleminden muzdarip. Platon arabacıyı bir zihin olarak tahayyül eder, fakat bu durumda tek yaptığı zihnin *başka bir zihin* (arabacı) tarafından kontrol edildiğini söylemektir. Felsefeciler buna **homunkulus problemi** diyorlar; homunkulus "insancık" anlamına geliyor, bu örnekte insancığımız da arabacı oluyor. Bu bir problem çünkü aslında hiçbir şeyi açıklamıyor, sadece "zihin" problemini başka bir zihin temsiline havale ediyor.

Bu "araba" modeli her halükârda şüphelidir çünkü *muhakemenin* davranışımızın itici gücü olduğunu öne sürer, halbuki durumun pek de öyle olmadığını gösteren bir sürü kanıt var elimizde. Örneğin sinirbilimci John-Dylan Haynes, büyük takdir gören bir dizi deney-

le, insan deneklerin karar vermesi ile karar verdiğinin bilincine varması arasında *10 saniyeyi* bulan bir zaman farkı olduğunu açıkça göstermiştir.[5]

Bu sonuç türlü türlü soruyu gündeme getiriyor. En basitinden, eğer ne yapacağımıza karar verme mekanizmamız bilinçli düşünce ve muhakeme değilse, böyle bir şey *ne için* var?

Evrim teorisi, insan bedeninin birtakım özelliklerini ele alırken, bunların bize evrimsel açıdan ne gibi avantajlar sağladığını sormamız gerektiğini söyler. Buna paralel olarak, bilinçli zihnin bize evrimsel açıdan ne gibi bir avantaj sağladığını sorabiliriz öyleyse. Çünkü sağladığı *hiçbir* avantaj olmasa muhtemelen ortaya da çıkmazdı.

Bir teoriye göre, bilinçli zihin gerçekte bedenimizin davranışımızı *bilfiil* üreten özelliklerinin mânâsız bir yan ürününden başka bir şey değildir. Bu teori **epifenomenalizm** gibi söylemesi zahmetli bir adla anılır. Bilinçli zihnin epifenomenal olduğunu söylemek, Platon'un öne sürdüğü gibi dizginleri elinde tutan arabacı değil de *yolcu*, daha doğrusu *kendisini arabacı sanan yolcu* olduğu anlamına gelir.

Biraz daha ılımlı bir görüşe göre, bilinçli zihin davranışımızda Platon'un iddia ettiği gibi asli bir rol oynamaz, daha ziyade bir biçimde beynimizdeki diğer süreçlerden kaynaklanmaktadır – muhtemelen, bildiğimiz kadarıyla, bizim kadar zengin bir zihinsel hayatı olmayan daha alt düzey hayvanlarda bulunmayan süreçlerden.

Sıradaki altbölümde, bilinçli insan deneyimimizin başlıca unsurlarından birini inceleyeceğiz: sosyal doğamız, yani bir sosyal grubun parçası olarak kendimizi ve başkalarını anlama yetimiz, keza başkaları ve başkalarının bizi nasıl gördüğü hakkında akıl yürütme yetimiz. Bu asli yeti büyük ihtimalle geniş, karmaşık sosyal gruplar halinde birlikte yaşama ve çalışmanın bir gereği olarak evrildi. Meseleyi daha iyi anlamak için, isterseniz İngiliz evrimsel psikolog Robin Dunbar'ın sosyal beyin üstüne birbirinden kıymetli çalışmalarıyla başlayalım.

## Sosyal Beyin

Dunbar şu temel soruya kafa yoruyordu: İnsanların (ve diğer primatların), diğer hayvanlara kıyasla, neden bu kadar büyük beyinleri[6] var? Beyin nihayetinde bir enformasyon işleme aracıdır ve insan bedeninin ürettiği toplam enerjinin kayda değer bir bölümünü, genelde yaklaşık %20'sini tüketir. Öyleyse büyük bir beyin, primatın önemli bir enformasyon işleme gereğine karşılık evrilerek bu hale gelmiş olmalı. Enerji ihtiyacının büyüklüğüne bakılırsa da sağladığı birtakım evrimsel avantajlar gerçekten önemli olmalı. Peki ama tam olarak ne gibi bir enformasyon işleme gereğinden ve evrimsel avantajdan söz ediyoruz burada?

Dunbar birçok primatı inceleyerek, gelişmiş enformasyon işleme kapasitesi ihtiyacının ne gibi faktörlerden kaynaklanıyor olabileceğini araştırdı. Örneğin bu durum belki de primatın çevresindeki besin kaynaklarını aklında tutma ihtiyacından kaynaklanıyordu. Veya hayatını ve besin arayışını daha geniş bir alanda sürdüren primatların daha geniş uzamsal haritaları aklında tutma gereğinden. Fakat Dunbar beyin büyüklüğünü en iyi yansıtan faktörün, primatın sosyal grubunun ortalama büyüklüğü, yani gruptaki ortalama hayvan sayısı olduğunu buldu. Bu da demek oluyordu ki primatların büyük sosyal grupları ayakta tutabilmeleri için, başka bir deyişle gruplardaki sosyal ilişkileri takip etmeleri, sürdürmeleri, bunlardan yararlanmaları için beyinlerinin daha büyük olması gerekiyordu.

Dunbar'ın araştırmaları merak uyandırıcı bir soruyu gündeme getirdi: İnsan beyninin ortalama büyüklüğü bilindiğine göre, Dunbar'ın analizi insan gruplarının ortalama büyüklüğünü ne kadar olarak öngörüyordu? Bu analizden elde edilen sonuç Dunbar sayısı olarak bilinir ve genelde 150 olarak alınır. Dunbar sayısı tuhaf bir fikir olarak kalabilirdi, ama daha sonra yapılan araştırmalarda da bu sayı, dünya genelinde insan gruplarının büyüklüğü bağlamında tekrar tekrar gündeme geldi. Örneğin neolitik tarım köylerinin genelde yaklaşık 150 kişiden meydana geldiği tahmin ediliyor. Daha yakın zamanlara bakacak olursak, Dunbar sayısı sözgelimi Facebook gibi

sosyal ağlar yoluyla aktif bir biçimde ilişki kurduğumuz arkadaşlarımızın sayısı hakkında da bir şeyler söylüyor.

Dunbar sayısı kabaca bir insan beyninin altından kalkabileceği insan ilişkisi sayısı olarak yorumlanabilir. Günümüz itibarıyla insanların çoğu bundan daha büyük gruplarla etkileşim halinde olabilir, ancak hakikaten takip edebileceğimiz ilişki sayısı Dunbar sayısı kadardır.

Özetle, eğer bu analiz doğruysa, *insan* beyni farklıdır çünkü *sosyal* bir beyindir. Diğer primatlarınkine kıyasla daha büyüktür, çünkü büyük sosyal gruplar halinde yaşarız ve bu da daha fazla sosyal ilişkiyi takip ve idare etme yetisi gerektirir.

Bu da bizi doğal olarak sosyal ilişkileri takip ve idare etmek tam olarak ne demektir sorusuna getiriyor. Bu soruya cevap ararken, gelin öncelikle tanınmış Amerikalı felsefeci Daniel Dennett'ın insanların davranışını nasıl anladığımızı ve tahmin ettiğimizi açıklamak için ortaya attığı **yönelimsel bakış açısı** fikrini inceleyelim.

## Yönelimsel Bakış Açısı

Dünyamıza şöyle bir bakıp etrafta gördüklerimizi anlamlandırmaya çalışırken, doğal bir biçimde failler ile diğer nesneler arasında bir ayrım yapıyor gibi görünüyoruz. Elinizdeki kitapta "ajan" terimini, bağımsız bir biçimde bizim adımıza edimde bulunan, rasyonel bir biçimde tercihlerimizin peşine düşen bir YZ programı anlamında kullandık. Bu altbölümde ise bu terimi özbelirlenimli (kendi davranışını belirleyen) bir fail olarak bizimkine benzer bir pozisyonda bulunan şeyler anlamında kullanacağız. Bir çocuk bir sürü çikolata arasından hangisini seçeceğini kafasında tartıp dikkatle birini eline alınca, faillikle karşı karşıya olduğumuzu algılarız. Çünkü burada hem seçim hem de kasıtlı, amaçlı, otonom eylem vardır. Buna karşılık bir bitki bir taşın altından baş gösterse, zamanla taşı bir tarafa itse, ortada faillik görmeyiz, çünkü burada bir tür eylem vardır ama söz konusu eylemde ne kasıt ne de bilinçli amaç algılarız.

İyi ama neden çocuğun eylemlerini bir failin/ajanın eylemleri gi-

bi yorumlarken, büyüyen bitkinin eylemlerine zihnin olmadığı bir süreç muamelesi yapıyoruz?

Bu sorunun cevabını anlamak için, dünyamızı değiştiren süreçleri açıklamaya çalışırken başvurduğumuz türlü türlü izahı düşünelim. Mevcut seçeneklerden biri, bir varlığın davranışını Dennett'ın deyişiyle **fiziksel bakış açısı**na atıfla anlamaya çalışmaktır; doğa kanunları (fizik, kimya vb.) yoluyla bir sistemin nasıl davranacağına dair tahminde bulunuruz. Örneğin Dennett, elinde tuttuğu taşı bırakırsa, biraz fizikle (taşın belli bir ağırlığı vardır, yerçekimi kuvveti taş üstünde etkilidir), taşın yere düşeceğini doğru bir biçimde tahmin edeceğini söyler. Fiziksel bakış açısı böyle örneklerde gayet iyi sonuçlar verse de, bu şekilde anlaşılamayacak kadar kompleks olan *insanların* davranışını anlamaya veya tahmin etmeye elbette uygun değildir. Teoride mümkün görünebilse de (nihayetinde hepimiz bir grup atomdan başka bir şey değiliz) pratikte şüphesiz mümkün değildir bu. Fiziksel bakış açısı, bilgisayarların veya bilgisayar programlarının davranışını anlamak için de pek uygun değildir – günümüzde tipik bir bilgisayar işletim sisteminin kaynak kodu yüz milyonlarca satır tutar.

İkinci bir seçenek ise **tasarımsal bakış açısı**dır. Buna göre davranışı, bir sistemin yerine getirmesi beklenen amacı –ne amaçla tasarlandığını– anlamaya çalışarak tahmin ederiz. Dennett burada çalar saat örneğini verir. Biri bize bir çalar saat getirse, saatin davranışını anlamak için fizik kanunlarına başvurmamız gerekmez. Bunun bir saat olduğunu biliyorsak, gösterdiği sayıları zaman olarak yorumlarız çünkü saatler zamanı göstermek üzere tasarlanmıştır. Aynı şekilde, saatten yüksek, rahatsız edici bir ses yükselse, bunu alarm olarak yorumlarız, çünkü çalar saatler alarmları kurulduğu takdirde belli zamanlarda (her zaman değil) yüksek ve rahatsız edici sesler çıkarmak üzere tasarlanmıştır. Böyle bir yorum yapabilmek için saatin *iç mekanizmasını* bilmek gerekmez. Çalar saatlerin böyle davranışlar sergilemek üzere tasarlandığını bilmek yeterlidir.

Üçüncüsü, ve burada asıl üstünde duracağımız seçenek, ise Dennett'ın deyişiyle yönelimsel bakış açısıdır.[7] Buna göre, bir varlığa – kanaat ve arzular gibi– **zihinsel haller** atfeder, ardından da bu zihin-

sel hallere yönelik sağduyu temelli bir teoriden yararlanarak söz konusu varlığın nasıl davranacağını tahmin ederiz (tahminde bulunurken onun kendisine atfedilen kanaat ve arzulara uygun tercihler yapacağını varsayarız). Bu yaklaşımın en bariz gerekçesi, insan etkinliğini açıklarken aşağıdaki gibi ifadeler kullanmanın faydalı olmasıdır:

> Janine yağmur yağacağını *düşünüyor* ve ıslanmak *istemiyor*.
> Peter işini bitirmek *istiyor*.

Janine yağmur yağacak diye düşünüyor ve ıslanmak istemiyorsa, muhtemelen yağmurluk giyeceğini veya yanına şemsiye alacağını ya da dışarı çıkmayacağını tahmin edebiliriz, çünkü bunu düşünen ve isteyen bir rasyonel failin/ajanın böyle davranışlar sergilemesini bekleriz. Dolayısıyla yönelimsel bakış açısının hem açıklama hem tahmin gücü vardır; insanların ne yaptığını ve –muhtemelen– ne yapacağını açıklamamıza imkân verir.

Dikkat ederseniz, tasarımsal bakış açısı gibi yönelimsel bakış açısının da bilfiil davranışı üreten *iç mekanizma* ile bir ilgisi yok. Teori makineler kadar insanlar konusunda da işe yarıyor. Bunu biraz daha ayrıntılı ele alalım.

Dennett'ın ortaya attığı **yönelimsel sistem** terimi, davranışları en iyi şekilde kanaat, arzu ve rasyonel seçim atfetme metoduyla anlaşılan ve tahmin edilen varlıkları tanımlar.

Yönelimsel sistemlerin en az karmaşık olanından en çok karmaşık olanına doğru doğal bir hiyerarşi vardır. 1. dereceden yönelimsel sistemin kendi kanaat ve arzuları vardır ama kanaat ve arzular *hakkında* kanaat ve arzuları yoktur. Buna karşılık 2. dereceden yönelimsel sistemde bu vardır. Meseleyi daha iyi anlamak için şu örneğe bakalım:

**1. derece:** Janine yağmur yağdığını *düşünüyordu.*

**2. derece:** Michael Janine'in yağmur yağdığını *düşünmesini istiyordu.*

**3. derece:** Peter Michael'ın Janine'in yağmur yağdığını *düşünmesini istediğini düşünüyordu.*

Gündelik hayatta yönelimsel bakış açısı hiyerarşisini 3. derecenin üstünde (bulmaca çözmek gibi yapay bir düşünsel faaliyete girişmedikçe) pek kullanmayız herhalde. Ayrıca çoğumuz 5. derecenin üstüne çıkmakta zorlanırmışız gibi görünüyor.

Yönelimsel bakış açısından yararlanma tarzımız sosyal hayvanlar olarak pozisyonumuzla yakından bağlantılıdır. Bu şekilde toplumdaki diğer faillere yönelim (niyet, maksat) atfetmenin rolü, onların davranışlarını anlamamıza ve tahmin etmemize imkân sağlamak gibi görünüyor. Karmaşık sosyal dünyamızda kendi yolumuzda ilerlerken, bireylerin planlarının (bizim planlarımızın veya gözlemlediğimiz faillerin planlarının) diğer faillerin öngörülen yönelimsel davranışlarından etkilendiği, daha üst düzey yönelimsel düşünmeyle iştigal ederiz. İnsan hayatının hemen her veçhesinde karşımıza çıkması ve insan iletişiminde böylesine kanıksanmış olması, bu düşünme tarzının değerini açıkça gösterir. Alice ile Bob arasında geçen şu dört kelimelik diyaloğu 1. Bölüm'den hatırlayacaksınız:

Bob: "Ayrılalım."
Alice: "Kim o kadın?"

Bu senaryonun yönelimsel bakış açısı temelindeki bariz açıklaması basit, doğal ve ikna edicidir: Alice Bob'un başka birini kendisine *yeğlediğini* ve hayatını buna göre *planladığını düşünür*, ayrıca –belki de Bob'u fikrinden caydırma ümidiyle– o kadının kim olduğunu *bilmek ister* ve sorduğu takdirde Bob'un söyleyeceğini *düşünür*. Bu konuşmayı, Alice ve Bob'un davranışında önemli bir rol oynamakla kalmayıp düşünmesinde ve planlamasında da öne çıkan, kanaat ve arzu gibi kavramlar *olmadan* açıklamak zor.

Dunbar insanlarda ve başka hayvanlarda beyin büyüklüğü ile daha üst düzey yönelimsel muhakeme yetileri arasındaki ilişkiyi de araştırmıştır. Bunun sonucunda, daha üst düzey yönelimsel muhakeme yetilerinin, kabaca, beynin frontal lobunun nispi büyüklüğüyle orantılı olduğu anlaşılmıştır. Beyin büyüklüğü ile sosyal grup büyüklüğü arasında güçlü bir korelasyon olduğuna göre, büyük beyinlerin doğal evrimsel açıklaması tam da, karmaşık bir toplumda sosyal muhakemeye –daha üst düzey yönelimsel muhakemeye– ihtiyaç

duyulması, bu yetinin değerli olmasıdır. İster gruplar halinde avlanırken veya hasım kabilelerle savaşırken, ister bir dişinin peşinde koşarken (belki rakipler arasından sıyrılmaya çalışırken) veya nüfuz ve liderlik için müttefik kazanmaya uğraşırken olsun, diğer bireylerin ne düşündüğünü anlayabilmenin ve tahmin edebilmenin değeri aşikârdır. Dunbar'ın argümanını hatırlayacak olursak, sosyal gruplar ne kadar büyükse daha üst düzey sosyal muhakemeyi o kadar çok gerektirir, bu da Dunbar'ın tespit ettiği beyin büyüklüğü ile sosyal grup büyüklüğü arasındaki ilişkiyi açıklar.

Asıl çıkış noktamıza geri dönersek, yönelimselliğin dereceleri ile bilincin dereceleri arasında bir korelasyon var gibi görünmektedir. *1. dereceden* yönelimsel sistemler olarak kabul edeceğimiz şeyler sınıfı gerçekten çok geniştir. Fakat daha üst düzey yönelimsellik çıtayı yükseğe taşır. İnsan daha üst düzey bir yönelimsel sistemdir, buna karşılık beni sözgelimi bir solucanın da böyle olduğuna ikna etmeye çalışsanız, kabul etmezdim. Peki ya bir köpek? Beni bu konuda ikna etmeye çok yaklaşabilirsiniz, çünkü bir köpeğin benim arzularım hakkında görüşleri olabileceğini kabul edebilirim (sözgelimi oturmasını istediğimi düşünebilir); ama bu, daha üst düzey bir muhakeme yetisiyse bile, oldukça kısıtlı ve belli bir konuyla sınırlıdır. Başka primatlarda da sınırlı bir üst düzey yönelimsel muhakeme yetisi olduğunu gösteren emareler var. Örneğin bir vervet maymunu, diğerlerini –topluluklarına tehdit oluşturan– bir leoparın varlığından haberdar etmek için bir uyarı çığlığı atar. Gelgelelim bir vervetin bu uyarı çığlığını, topluluğun diğer üyelerini kandırıp ortada leopar falan yokken leoparların saldırısına uğradıklarına inandırmak için de kullandığı gözlenmiştir.[8] Vervetin neden numara yaptığının doğal bir açıklaması daha üst düzey yönelimsel muhakemeyi de içerir: "Uyarı çığlığı atarsam, leoparların saldırdığını *düşünürler* ve kaçmak *isterler*..." Durumun daha üst düzey yönelimsel muhakemeyi içermeyen başka izahları da vardır elbette. Yine de bu anekdot insan olmayan hayvanların daha üst düzey yönelimsel muhakeme yetisine sahip olduğu iddiasını daha inanılır hale getiriyor.

Daha üst düzey yönelimsellik biçiminde sosyal muhakeme ile bilinç arasında bir korelasyon var gibi görünüyor; sosyal muhakeme

karmaşık sosyal ağları ve geniş sosyal grupları desteklemek üzere gelişmişe benziyor. Öyleyse, sosyal muhakeme neden bilinci gerektiriyor olabilir? Çalışma arkadaşım Peter Millican'a göre, bu sorunun cevabı tam da yönelimsel bakış açısının hesap kitap yapma açısından verimli olmasında yatıyor olabilir. *Beni* motive eden unsurlara –arzularım, kanaatlerim vb.– bilinçli erişimimin olması ve kendimi başka insanların yerine koyup düşünebilmem, onların davranışını (hem gerçek hem hayali koşullarda) çok daha randımanlı bir biçimde tahmin etmemi sağlar. Mesela birinin yemeğini çalmaya niyetlensem, kendimi onun yerine koyup, hissedeceği öfkeyi içgüdüsel olarak (düşünüp taşınmadan) *hissedebilirim*, planlayabileceği misillemeleri tahmin edebilirim ve dolayısıyla yemeğini çalma ayartısına direnmeye motive olurum. Enteresan bir spekülasyon bu. Ancak *insanın* sosyal muhakeme yetisi ile bilinci arasında nasıl bir ilişki olursa olsun, yakın zamanda bu konuyla ilgili kesin cevaplara ulaşacakmışız gibi görünmüyor hiç. Öyleyse asıl konumuz olan YZ'ye dönelim ve sosyal muhakeme yetisine sahip *makinelerin* mümkün olup olmadığına kafa yoralım.

## Makineler Düşünebilir ve Arzulayabilir mi?

İnsan toplumunda önemli bir rol oynayan yönelimsel bakış açısı envai çeşit varlığa uygulanabilir. Sıradan bir elektrik düğmesini düşünün mesela, yönelimsel bakış açısı düğmenin davranışına dair gayet iyi iş gören betimsel bir açıklama sağlar: Düğme elektrik akımını iletmesini istediğimizi düşündüğünde iletir, aksi halde iletmez; düğmeye basarak arzularımızı bildiririz.[9]

Gelgelelim yönelimsel bakış açısı bir elektrik düğmesinin davranışını anlamanın ve tahmin etmenin en *uygun* yolu değildir elbette. Bu örnekte fiziksel bakış açısına veya tasarımsal bakış açısına başvurmak daha basittir. Elektrik düğmesinin davranışını yönelimsel bakış açısıyla açıklamak, ona devreden akımın geçip geçmemesiyle ve bizim kendi arzularımızla ilgili kanaatler atfetmeyi gerektirir. Yönelimsel betimleme elektrik düğmesinin davranışını doğru tahmin etmeyi sağlasa da, *açıklama* olarak biraz abartılıdır.

Yönelimsel bakış açısını makinelere uyguladığımızda, iki önemli soru gündeme geliyor: Makineler hakkında konuşurken yönelimsel bakış açısına başvurmak ne zaman *meşrudur*, ne zaman *faydalıdır*? Zihinsel halleri olan makineler konusunda önde gelen düşünürlerden John McCarthy bu sorular üstüne şunları söylüyor:[10]

> Bir makineye veya bilgisayar programına birtakım kanaatler, bilgi, özgür irade, yönelim (niyet), bilinç, yetiler veya istekler atfetmek, bu atıf makine hakkında, bir insan hakkında ifade ettiği enformasyonun aynısını ifade ediyorsa meşrudur. ... Zihinsel nitelikler atfetmenin en dolambaçsız olduğu sistemler termostatlar ve bilgisayar işletim sistemleri gibi yapısı gayet iyi bilinen makinelerdir, ama en faydalı olduğu sistemler yapısı tam olarak bilinmeyen varlıklardır.

Yoğun bir alıntı bu, o nedenle isterseniz biraz açalım. Birincisi, McCarthy bir makineye dair yönelimsel bakış açısına dayalı bir açıklamanın, makine hakkında, makinenin bir insan hakkında ifade etiği enformasyonun aynısını ifade etmesi gerektiğini öne sürüyor. Şüphesiz büyük bir koşul bu, insanın aklına Turing testinin ayırt edilemezlik koşulu geliyor. Daha önce verdiğimiz bir örneği buna uyarlayarak, bir robotun yağmur yağdığına inandığını ve ıslanmak istemediğini iddia edersek, bu ancak robot ilgili kanaat ve arzulara sahip bir rasyonel fail/ajan gibi davranırsa anlamlı bir atıf olur. Yani robot, yapabiliyorsa, ıslanmamak için gerekli tedbirleri almalıdır. Tedbir almıyorsa muhtemelen ya yağmur yağdığına inanmıyormuş, ya ıslanıp ıslanmamayı o kadar da önemsemiyormuş, ya da rasyonel bir varlık değilmiş deriz.

Son olarak, McCarthy yönelimsel bakış açısının en çok bir varlığın yapısı veya işleyişini anlamadığımızda değerli olduğuna dikkat çeker. Davranışı iç yapısı ve işleyişinden bağımsız olarak açıklamak ve tahmin etmek için bir yöntem ortaya koyar (mesela söz konusu varlık ister insan olsun ister köpek veya robot, bu açıdan bir şey fark etmez). Eğer ıslanmak istemeyen ve yağmur yağdığına inanan bir rasyonel failseniz, davranışınızı sizin hakkınızda hiçbir şey bilmeden açıklayabilir ve tahmin edebilirim.

## Bilinçli Makinelere Doğru

Bütün bunlar YZ hayallerimizle ilgili olarak bize ne söylüyor peki? Sona iyice yaklaştığımız bu altbölümde, müsaadenizle, bilinçli makinelere doğru ilerlemenin nasıl bir şey olabileceği ve bunu nasıl sağlayabileceğimiz konusunda hızlıca birkaç somut öneride bulunmak istiyorum. (İyice yaşlandığım zaman bu kısmı tekrar okuyup tahminlerimin çıkıp çıkmadığını görmeyi sabırsızlıkla bekliyorum.)

Hatırlarsanız 5. Bölüm'de, DeepMind'ın inşa ettiği bir ajanın bir sürü Atari oyunu oynamayı öğrendiğinden bahsetmiştik. Bu oyunlar pek çok açıdan nispeten basitti, ama DeepMind zamanla bunun ötesine geçti tabii, artık StarCraft gibi çok daha karmaşık oyunlar oynayabilen ajanları var.[11] Günümüzde böyle deneylerde daha ziyade şu meseleler öne çıkıyor: dallanma faktörü çok büyük olan oyunların; oyunun durumu ve diğer oyuncuların eylemleri hakkındaki enformasyonun eksik olduğu oyunların; mevcut ödüller ile nihayetinde ödül kazandıran eylemler arasındaki ilişkinin biraz fazla dolaylı olduğu oyunların; ve ajanın sözgelimi Breakout'taki gibi mevcut iki seçenekten birini seçmek yerine uzun, karmaşık eylem dizilerini muhtemelen başkalarının eylemleriyle koordineli bir biçimde –veya rekabet halinde– gerçekleştirmesi gereken oyunların üstesinden gelebilmek.

Müthiş çalışmalar bunlar, alanda kaydedilen ilerleme çok etkileyici. Ancak bu çalışmalarda belli bir ilerleme kaydetmenin nasıl bilinçli makinelere kapı aralayacağını anlamak zor, çünkü ele alınan konular hiç de bilinçle ilgiliymiş gibi görünmüyor (söz konusu çalışmalara yönelik bir eleştiri değil bu, sadece elinizdeki kitabın ele almayı amaçladığı konuların dışında kalıyorlar).

Şimdi de, buraya kadarki tartışmalarımıza dayanarak, bu amaca doğru nasıl ilerleyebileceğimiz konusunda birtakım önerilerde bulunmayı deneyeceğim. Diyelim ki elimizde bir makine öğrenmesi programı var. Öyle bir program ki, *anlamlı, karmaşık, daha üst düzey yönelimsel muhakeme* gerektiren bir senaryoda başarıya ulaşmayı –tıpkı DeepMind'ın ajanının Breakout'ta yaptığı gibi, kendi

kendine– öğreniyor. Veya ajanın daha üst düzey yönelimsel muhakemeye işaret eden *sofistike bir yalan söylemesini* gerektiren bir senaryoda. Veya ajanın kendisinin ve başkalarının zihinsel hallerinin özelliklerini iletmeyi ve anlamlı bir biçimde ifade etmeyi öğrendiği bir senaryoda. Bir sistemin bu şeyleri anlamlı bir biçimde yapmayı öğrenebilmesinin, bilinçli makinelere giden yolda kayda değer bir aşama olacağını düşünüyorum.[12]

Bu noktada, çocuklarda otizmin teşhisine yardımcı olmak üzere ortaya atılan **Sally-Anne testi** gibi bir şey var aklımda.[13] Otizm çocuklukta kendini belli eden ciddi, yaygın bir psikiyatrik durumdur:[14]

> [Otizmin] başlıca semptomları, erken çocukluğun ilk yıllarında sosyal ve iletişimsel gelişimde bariz anomalilerdir. Çocuğun oyunlarında olması beklenen esneklik, hayal gücü ve -mış gibi yapma görülmez. ... Otizmde sosyal anomalilerin başlıca özellikleri ... göz teması kurmama, normal sosyal farkındalığın olmaması veya uygun sosyal davranışın sergilenmemesi, "tek başınalık", tek taraflı etkileşim ve bir sosyal gruba katılamamadır.

Tipik bir Sally-Anne testinde çocuğa şöyle bir kısa hikâye anlatılır veya canlandırılır: Sally ve Anne bir odadadır; odada ayrıca bir sepet, kutu ve misket vardır. Sally misketi sepete koyar, ardından dışarı çıkar. Sally dışarıdayken Anne misketi sepetten çıkarıp kutuya koyar. Sally odaya girer, misketle oynamak istemektedir.

Bu noktada çocuğa şu soru sorulur:

"Sally misketi nerede arar?"

Soruya uygun cevap elbette "Sepette"dir. Ancak bu cevaba ulaşmak için başkalarının kanaatleri hakkında biraz akıl yürütmek gerekir. Sally, Anne'in misketin yerini değiştirdiğini görmemiştir, dolayısıyla misketi bıraktığı yerde yani sepette bulacağına inanır. Otistik çocukların büyük çoğunluğu bu soruya yanlış cevap vermekte, klinik açıdan normal çocuklar ise hemen her zaman doğru cevap vermektedir.

Söz konusu yaklaşımın öncüsü olan Simon Baron-Cohen ve çalışma arkadaşları, bu durumu otistik çocuklarda bir **Zihin Teorisi**nin (ZT) olmayışının kanıtı olarak görüyor. ZT, gelişimini tamamlamış yetişkinlerin sahip olduğu, pratik, sağduyuya dayanan bir yeti-

dir; bireyin başkalarının ve kendisinin zihinsel halleri –kanaatler, arzular vb.– konusunda akıl yürütmesine imkân sağlar. Klinik açıdan normal bir insan ZT ile doğmaz ama ZT'yi kademeli olarak geliştirme kapasitesine sahiptir. Klinik açıdan normal çocuklar 4 yaş itibarıyla başkalarının perspektif ve bakış açılarını göz önünde bulundurarak akıl yürütebilir, ergenlik itibarıyla ise ZT'yi geliştirmeyi tamamlarlar.

Elinizdeki kitabın yazıldığı tarih itibarıyla, makine öğrenmesi programlarının nasıl ilkel bir ZT öğrenebileceğine dair birtakım araştırmalara başlanıyor.[15] Yakın dönemde araştırmacılar tarafından geliştirilen bir nöral ağ sistemi (ToMnet), başka ajanları nasıl modelleyeceğini ve Sally-Anne testindekine benzer durumlarda nasıl doğru bir davranış sergileyeceğini öğrenebiliyor. Ne var ki çalışma henüz çok erken bir aşamada, ayrıca Sally-Anne testini geçme kapasitesi yapay bilinç iddiasının altını doldurmaya yetmiyor. Yine de bunun doğru istikamette atılmış bir adım olduğunu düşünüyorum. Çünkü böylece kendimize bir hedef daha koyabiliriz: otonom olarak insan düzeyinde bir Zihin Teorisi öğrenen bir YZ sistemi.

## Bizim Gibi mi Olacak?

YZ ve insan tartışmasında haliyle beyin de sık sık gündeme gelir. Çünkü beyin insan bedeninin enformasyon işleyen başlıca bileşenidir; problem çözme, anlatılanı anlama vb. görevleri gerçekleştirirken, asıl yükü beyin taşır. Dolayısıyla beyni genelde sürücüsüz otomobili kontrol eden bilgisayar gibi tahayyül ederiz; sanki beyin, duyu organlarımızdan aldığı enformasyonu yorumlayarak, ellerimize ayaklarımıza ne yapacaklarını söylemektedir. Halbuki işin aslı tam olarak öyle değildir, beyni böyle tahayyül etmek meseleyi fazla basite indirgemek olur, çünkü beyin pek çok karmaşık bileşenin iyice iç içe geçmesiyle meydana gelen bir sistemin bileşenlerinden yalnızca biridir, bu bileşenler gezegenimizde canlılığın ortaya çıkışından itibaren milyarlarca yıl boyunca hep birlikte evrilerek tek bir organizma haline gelmiştir. Evrim açısından, bizler kuyruksuz maymunlardan –özel muamele görmeyi hak ettiğine inanan kuyruksuz

maymunlardan– başka bir şey değiliz aslında ve insan bilincini de işte bu bağlamda değerlendirmek gerekiyor.

Bilinçli zihin dahil mevcut kapasitelerimiz, atalarımızın gelişimini yönlendiren evrimsel kuvvetlerin bir sonucu.[16] Tıpkı ellerimiz, gözlerimiz, kulaklarımız gibi insan bilinci de evrildi. İnsan bilinci akşamdan sabaha gelişmedi; atalarımız aslında solucanı andırıyordu da hop bir anda acayip bir aydınlanma yaşayıp Shakespeare oluvermedi. Nasıl ki antik zamanlarda atalarımızın bilinci bizimki kadar geniş bir yelpazede değildiyse, uzak gelecekte de bizim soyumuzdan gelenlerin bilinci muhtemelen bizimkinden daha geniş bir yelpazeye sahip olacak. Evrimsel gelişim süreci elbette ki bitmiş değil.

Tarihsel kayıtlar bilincin birtakım unsurlarının nasıl ve neden ortaya çıktığına dair bazı ipuçları veriyor. Yalnız bu kayıtlarda çok boşluk var, dolayısıyla kaçınılmaz olarak –çok fazla– spekülasyon yapmamız grekiyor. Yine de her halükârda enteresan ipuçları bunlar.

*Homo sapiens* (biz) dahil bugün mevcut olan her kuyruksuz maymun türünün yaklaşık 18 milyon yıl önce ortak bir atası vardı. O tarihlerde, kuyruksuz maymunların evrim ağacı çatallanmaya başladı; yaklaşık 16 milyon yıl önce orangutanlar farklı bir evrim yoluna saptı; 6 ila 7 milyon yıl önce de biz, goriller ve şempanzelerden ayrılarak kendi yolumuza koyulduk. Bu bölünmeden sonra, atalarımızın benimsediği bir özellik bizi diğer kuyruksuz maymun türlerinden ayırmaya başlayacaktı: Ağaçlardan ziyade yerde zaman geçirir olduk, nihayetinde dikelip iki ayağımız üstünde yürümeye başladık. Bu durum iklim değişikliğiyle, orman örtüsünün azalması sonucu ağaçlarda yaşayan atalarımızın açık arazilere inmek zorunda kalmasıyla açıklanabilir. Ağacın korumasından mahrum kalmak muhtemelen atalarımız açısından yırtıcıların saldırısına uğrama riskini artırmıştı. Evrimin buna cevabı, sosyal grupların büyüklüğünün artması oldu; çünkü grup ne kadar büyükse, yırtıcılar açısından saldırmaya o kadar elverişsiz demekti. Daha büyük sosyal gruplar sözünü ettiğimiz sosyal muhakeme yetilerini, bu yetiler de daha büyük beyinleri gerekli hale getirdi.

Bir milyon yıl kadar önce atalarımızın tek tük de olsa ateşi kul-

landığını biliyoruz, fakat yaygın olarak kullanımı yaklaşık yarım milyon yıl önce başlamış gibi görünüyor. Ateş atalarımıza envai çeşit fayda sağladı; aydınlattı, ısıttı, yırtıcıları korkutup kaçırdı, beslenme imkânlarını artırdı. Diğer taraftan bu yeni teknoloji ihmal etmeye gelmiyor, dikkat ve özen gerektiriyordu; dönüşümlü olarak ateşi gözlemek, yakacak toplamak vb. için işbirliği yapmak lazımdı. Bu tür bir işbirliği (birbirlerinin arzu ve niyetlerini anlayabilmeleri için) daha üst düzey yönelimsel muhakeme yetisinden ve muhtemelen –aşağı yukarı yine o zamanlarda gelişmiş gibi görünen– dilden faydalanabilirdi. Dil kapasitesi tesis edildi mi, dilin gelişmesinin daha başka faydaları da görülecekti.

Hangi gelişmenin önce hangisinin sonra olduğunu, ayrıca bu gelişmelerin ne gibi kapasiteleri beraberinde getirdiğini kesin olarak bilmiyoruz belki ama karşımızda genel hatları itibarıyla gayet açık bir manzara var ve burada bilincin bazı bileşenlerinin zamanla ortaya çıkışını az çok seçebiliyoruz. Zor bir problem olarak bilinçle ilgili sorularımıza cevap olmuyor bu ama en azından tam gelişmiş insan bilinci için gerekli kimi bileşenlerin nasıl ve neden ortaya çıktığına dair birtakım makul ipuçları veriyor. Nihayetinde bu ipuçları bir yere varmayacak belki, ama tahminler genellikle bir şeyi daha iyi anlamamızı sağlar ve bilinci bir muamma olarak görmekten daha fazlasını sunduklarına da şüphe yok.

Nasıl ki bugün güneşin enerjisinin nereden geldiğini biliyorsak, günün birinde de bilincin ne olduğunu anlamış olacağız. Nasıl ki bugün, nükleer fizik henüz güneş enerjisinin kaynağını makul bir biçimde açıklamamışken ortaya atılan çeşitli teorilere gülüp geçiyorsak, bilincin ne olduğunu anladığımız o gün de işte bu tartışmalara gülüp geçeceğiz herhalde.

Diyelim ki bu araştırma gündemi başarıya ulaştı, gerçekten insan düzeyinde bir zihin teorisi olan makineler inşa edebildik; daha üst düzey yönelimsel akıl yürütmenin karmaşık biçimlerinin üstesinden gelmeyi otonom olarak öğrenen makineler, karmaşık sosyal ilişkiler kurup bunları yürütebilen makineler, kendilerinin ve başkalarının zihinsel hallerinin karmaşık özelliklerini ifade edebilen makineler. Bu makinelerin *gerçekten* bir "zihni", bilinci, özfarkın-

dalığı olacak mı? Sorun şu ki, bu mesafeden, bu soruyu cevaplayamıyoruz. Böyle makineleri inşa edebilmeye ne kadar yaklaşırsak – günün birinde bu *gerçekten* mümkün olursa tabii– bu konuda kafamız o kadar net olacak.

Doğruya doğru, şu an için *asla* bu soruya tatmin edici bir cevap veremiyoruz, fakat bütün mesele bundan mı ibaret sahiden? Tam da bu konuyla ilgili olarak, Turing'in söyledikleri kafamda dönüp duruyor. Hatırlayacak olursanız Turing diyordu ki, eğer makine "gerçeğinden" sahiden *ayırt edilemeyen* bir şey yapıyorsa, o zaman o şey "gerçekten" bilinçli mi veya özfarkındalık sahibi mi diye tartışmaktan vazgeçmemiz lazım. Aklımıza yatan her türlü testi geçiyor, sahiden ayırt edilemiyorsa, bizim için bu kadarı da yeterlidir belki?

—

Ekler

# Ek A: Kuralları Anlayalım

Bildiğim kadarıyla Patrick Winston ve Berthold Horn'un ortaya koyduğu hayvanlarla ilgili örneğin kural tabanının tamamı şöyle:

1. EĞER hayvanın tüyleri varsa
   O ZAMAN hayvan memelidir
2. EĞER hayvan süt veriyorsa
   O ZAMAN hayvan memelidir
3. EĞER hayvanın telekleri varsa
   O ZAMAN hayvan kuştur
4. EĞER hayvan uçabiliyorsa VE yumurtluyorsa
   O ZAMAN hayvan kuştur
5. EĞER hayvan et yiyorsa
   O ZAMAN hayvan etoburdur
6. EĞER hayvan memeliyse VE hayvan tek tırnaklıysa
   O ZAMAN hayvan toynaklıdır
7. EĞER hayvan memeliyse VE hayvan etobursa VE hayvanın rengi alacalıysa VE hayvanın siyah şeritleri varsa
   O ZAMAN hayvan kaplandır
8. EĞER hayvan toynaklıysa VE siyah şeritleri varsa
   O ZAMAN hayvan zebradır

Gelin şimdi de bu kuralların ileriye doğru zincirlemeyle nasıl işlediğine bakalım. Diyelim ki kullanıcı hayvanın süt verdiği, tek tırnaklı olduğu ve siyah şeritleri olduğu enformasyonunu verdi. İleriye doğru zincirleme şöyle ilerler:

1. Hayvanın süt veriyor olması 2. Kural'ın tetiklenmesini sağlar ve böylece hayvanın memeli olduğu enformasyonu çalışma belleğine eklenir.
2. Hayvanın memeli olduğu yolundaki yeni enformasyon, hayvanın tek tırnaklı olduğu olgusuyla birlikte, 6. Kural'ın tetiklenmesini sağlar ve böylece hayvanın toynaklı olduğu olgusu çalışma belleğine eklenir.
3. Hayvanın toynaklı olduğu yolundaki yeni enformasyon, hayvanın siyah şeritleri olduğu olgusuyla birlikte, 8. Kural'ın tetiklenmesini sağlar ve böylece hayvanın zebra olduğu olgusu çalışma belleğine eklenir.
4. Bu noktada, başka bir kural tetiklenmez.

Bu sayede, bize verilen genel kurallardan üç yeni bilgi ve söz konusu hayvan hakkında spesifik enformasyon elde ederiz.

Hatırlarsanız, geriye doğru zincirlemede tesis etmek istediğimiz bir hipotezden (amaçtan) başlıyor, ardından kuralları kullanarak daha küçük (atomik) bilgi parçalarına ulaşmaya çalışıyorduk. Gelin şimdi de kullanıcının sisteme, herhangi bir enformasyon sağlamaksızın, hayvanın toynaklı olup olmadığını belirlemek şeklinde bir amaç vermesi halinde geriye doğru zincirlemenin nasıl işlediğine bakalım:

1. Çıkarım motoru amacı yani sonucu ("hayvan toynaklıdır") olan bir kural arar. Bu örnekte böyle bir kural, yani 6. Kural'ı bulur.
2. Çıkarım motoru 6. Kural'ı tetiklemeye yetecek kadar enformasyona sahip değildir, zira şartların ("hayvan memeliyse VE hayvan tek tırnaklıysa") doğru olup olmadığını bilmemektedir. Dolayısıyla önüne bir amaç olarak bu önermelerin doğru olup olmadığını belirlemeyi koyar. Böylece "hayvan memelidir" ve "hayvan tek tırnaklıdır", teknik açıdan, *alt-amaçlar* olur.
3. Önce "hayvan memelidir" alt-amacını ele alan çıkarım motoru, bir sonucu bu olan kuralları bulmaya çalışır. Bu örnekte böyle iki kural vardır: 1. Kural ve 2. Kural. Çıkarım motoruna göre bu amaç iki farklı yoldan tesis edilebilir demektir: Ya hayvanın kıl-

ları olduğunu (1. Kural) ya da süt verdiğini (2. Kural) keşfedecektir. Dolayısıyla çıkarım motoru bunların ilkini ("hayvanın tüyleri vardır") alıp bir alt-amaç olarak tesis edecektir.

4. Çıkarım motoru alt-amacı yani sonucu "hayvanın tüyleri vardır" olan bir kural arar. Gelgelelim bu örnekte böyle bir kural yoktur, atomik bir önermeye varmış oluruz. Diğer taraftan bu, yolun sonu da değildir ama: Bu tür atomik önermelerle karşı karşıya kalan çıkarım motoru kullanıcıya belli bir önermenin doğru olup olmadığını sorabilir. Böylece geriye doğru zincirlemeye dayanan muhakeme süreci kullanıcıyla bir soru-cevap oturumu haline gelir.

   Diyelim ki kullanıcı hayvanın tüyleri olup olmadığı hakkında herhangi bir enformasyona sahip değil, dolayısıyla soruya "Bilmiyorum" diye cevap veriyor. Bu konuyla ilgili olarak daha fazla enformasyona ulaşamayan çıkarım motoru 1. Kural'ın tetiklenmeyeceği çıkarımında bulunur, dolayısıyla da sonuç olarak "hayvan memelidir"i veren diğer kurala, yani şartı "hayvan süt veriyorsa" olan 2. Kural'a geçer.
5. Çıkarım motoru "hayvan süt veriyor"u bir alt-amaç olarak tesis eder ama bunun da atomik olduğunu görür. Dolayısıyla kullanıcıya sorar. Kullanıcının bu kez "Evet" diye cevap verdiğini varsayalım. "Hayvan süt veriyor" enformasyonu çalışma belleğine eklenir, böylece 2. Kural tetiklenir ve "hayvan memelidir" çalışma belleğine eklenir.
6. Bu noktada, çıkarım motoru 6. Kural'ın şartının ilk kısmını tesis etmiş olur, bu nedenle ikinci kısma geçerek "hayvan tek tırnaklıdır"ın doğru olup olmadığını tesis etmeye çalışır. Dolayısıyla da bunu bir alt-amaç olarak tesis eder.
7. "Hayvan tek tırnaklıdır" da atomik bir önermedir, bu nedenle kullanıcıya bir soru daha sorulur. Diyelim ki kullanıcının cevabı "Evet" oldu. Söz konusu olgu çalışma belleğine eklenir, "hayvan memelidir" enformasyonuyla birleştirilir; böylece sistem 6. Kural'ın tetiklenmesini sağlayabilir, nihayetinde hayvanın toynaklı olduğu sonucuna varabilir. Kullanıcının baştaki amacı tesis edilmiş olur.

# Ek B: PROLOG'u Anlayalım

Basit bir PROLOG programının aile ilişkileri hakkında nasıl bilgi yakaladığına bakalım. Diyelim ki "kadın(*X*)" yazdığımızda bu, *X*'in kadın olduğu anlamına geliyor; "ebeveyn(*X*,*M*,*F*)" yazdığımızda ise *X* kişisinin sırasıyla babası ve annesinin *M* ve *F* olduğu ifade ediliyor. Bu durumda şöyle bir PROLOG kuralı yazarak:

kız_kardeşi(*X*,*Y*):-kadın(*X*), ebeveyn(*X*,*M*,*F*), ebeveyn(*Y*,*M*,*F*).

şu bilgiyi temsil edebiliriz:

*X*, *Y*'nin kız kardeşidir, eğer:
1. *X* kadınsa
2. *X*'in ebeveyni *M* ve *F* ise ve
3. *Y*'nin ebeveyni *M* ve *F* ise.

Mantıksal akıl yürütmeye aşina değilseniz, birinin kız kardeş olduğunu ifade etmek için fazla dolambaçlı bir yol gibi gelebilir bu size. Ama özünde, burada söylenen, *X* kadın olduğu ve *X* ile *Y*'nin ebeveynleri aynı kişiler olduğu takdirde *X*'in *Y*'nin kız kardeşi olduğundan ibarettir.

Bunların ardından, PROLOG'a birkaç olgu verebiliriz:

kadın(janine).
ebeveyn(janine, wayne, yvonne).
ebeveyn(david, wayne, yvonne).

Bu doğrultuda, PROLOG'a Janine'in David'in kız kardeşi olduğunu kanıtlamaya çalışma görevi verirsek, başarıyla yerine getirir. Burada amaç şudur:

```
kız_kardeşi(janine, david)
```

Bu amaç sunulduğunda, PROLOG "Evet" cevabı vererek, Janine'in David'in kız kardeşi olduğunu kanıtlayabildiğini belirtir.

# Ek C: Bayes Teoremi'ni Anlayalım

Bayes Teoremi'nin nasıl işlediğini, 4. Bölüm'den hatırlayacağınız grip örneği üstünden ele alalım. Teoremin bu örnekte nasıl kullanıldığını anlamak için, matematikçilerin "gösterim" dediği şeyi biraz bilmemiz gerekiyor.

Birincisi, hipotezimiz "gripsiniz", edindiğimiz kanıt da "testiniz pozitif". Şimdi, elimizdeki kanıtı göz önünde bulundurarak, hipotezin doğru olma olasılığını, yani grip olma olasılığınızı hesaplamaya çalışıyoruz. Bu değeri *Olası*(hipotez | kanıt) şeklinde yazarız. Burada "|" simgesinin anlamı "göz önünde bulundurarak"tır.

Hipotezin (gripsiniz) doğru olmasının **önsel olasılığına** işaret etmek için *Olası*(hipotez) yazarız. "Önsel olasılık", yeni kanıtları görmeden *önce* hipotezin doğruluğuna verdiğimiz olasılıktır. Elimizdeki örnekte önsel olasılık gelişigüzel seçilen bir insanın grip olma olasılığıdır. Her 1000 kişiden yaklaşık 1'inin grip olduğunu biliyoruz, öyleyse bu örnekte önsel olasılık 1/1000 = 0,001'dir. Bu değeri hipotez hakkındaki başlangıç kanaatimiz olarak düşünebiliriz. Birisi hakkında başka hiçbir şey bilmiyorsak, grip olma olasılığına ilişkin tahminimiz tam da gelişigüzel seçilen birinin grip olma olasılığı yani 1000'de 1'dir. Bu noktada, Bayes Teoremi'nin şöyle bir güzelliği vardır: Söz konusu kanaati yeni kanıtlar ışığında *güncellememize* imkân sağlar.

Ardından, hipotezin (gripsiniz) doğru olma olasılığını göz önünde bulundurarak, kanıtı görme olasılığımız için *Olası*(kanıt | hipotez) yazarız. Testin doğruluk oranının %99 olduğunu biliyoruz. Yani gripseniz, 100 testin 99'unda sonuç pozitif olacaktır. Dolayısıyla

grip olduğunuz takdirde testinizin pozitif çıkma olasılığı 0,99'dur.

Son olarak, gelişigüzel seçilen birinin yaptırdığı testin doğru sonuç verme olasılığı için *Olası*(kanıt) yazarız. Elimizdeki örneğin gerçekten incelik isteyen tek tarafı bu değeri bulmaktır: bir kişinin grip olma olasılığı çarpı testin bu kişi için pozitif çıkma olasılığı, artı, kişinin grip *olmama* olasılığı çarpı testin yine de pozitif çıkma olasılığı:

$$Prob(\text{kanıt}) = (0{,}001 \times 0{,}99) + (0{,}999 \times 0{,}01) = 0{,}011$$

Buna göre Bayes Teoremi şöyle ifade edilebilir:

$$Olası(\text{hipotez} \mid \text{kanıt}) = \frac{(Olası(\text{kanıt} \mid \text{hipotez}) \times Olası(\text{hipotez}))}{(Olası(\text{kanıt}))}$$

Örneğimizdeki değerleri formüldeki yerlerine yerleştirince de sonuç şöyle olur:

$$Olası(\text{hipotez} \mid \text{kanıt}) = \frac{(Olası(\text{kanıt} \mid \text{hipotez}) \times Olası(\text{hipotez}))}{(Olası(\text{kanıt}))}$$

$$= \frac{0{,}99 \times 0{,}001}{0{,}011} = 0{,}09$$

Sonuç olarak, doğruluk oranı %99 olan testinizin pozitif çıktığı göz önünde bulundurulduğunda grip olma olasılığınız genel itibarıyla 0,09'dur, yani 10'da 1'den birazcık daha azdır.

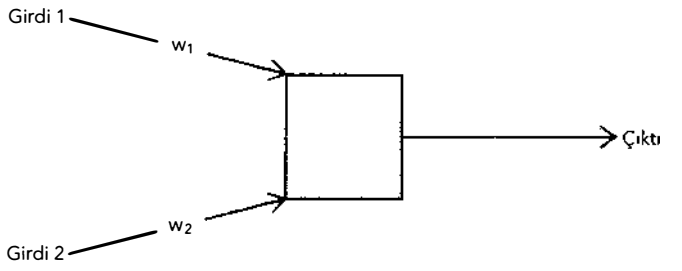
# Ek D: Nöral Ağları Anlayalım

Gelin öncelikle Minsky ve Papert'ın üstünde durduğu perseptron modeli problemini inceleyelim: Tek katmanlı perseptronlar son derece basit bir sınıflandırma problemi olan "XOR" işlevini uygulayamaz.

Diyelim ki programın öğrenebilmesini istediğimiz XOR işlevi şu: Girdi 1 *veya* Girdi 2 aktifse, çıktının aktif olmasını istiyoruz (başka bir deyişle, girdilerin *her ikisi de* aktif olduğu takdirde çıktının aktif olmasını *istemiyoruz*). Söz konusu perseptron modelinde bunun neden mümkün olmadığını daha iyi anlamak için, bir süreliğine, sanki mümkün olabilirmiş gibi düşünelim. Bu durumda ağırlıklar $w_1$ ve $w_2$'nin ve eşik değeri $T$'nin özellikleri şöyledir:

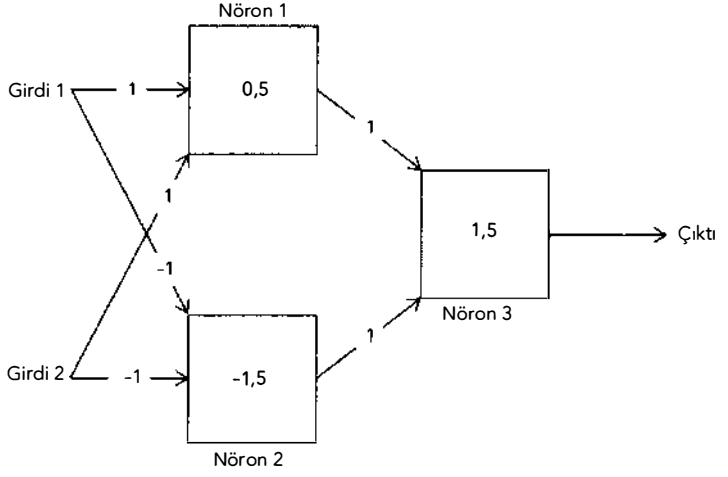
- Eğer iki girdi de aktif değilse, o zaman nöronun aldığı girdilerin toplam ağırlığı 0'dır ve tetiklenmez, öyleyse $T > 0$ olmalıdır.
- Eğer sadece Girdi 1 aktifse, o zaman perseptron tetiklenir, öyleyse $w_1 > T$ olmalıdır.
- Eğer sadece Girdi 2 aktifse, o zaman perseptron tetiklenir, öyleyse $w_2 > T$ olmalıdır.
- Eğer *hem* Girdi 1 *hem de* Girdi 2 aktifse, o zaman perseptron *tetiklenmez*, öyleyse $w_1 + w_2 < T$ olmalıdır.

Gördüğünüz gibi, aynı anda bütün bu özellikleri taşıyan bir $w_1$, $w_2$ ve $T$ değeri yoktur. Dolayısıyla XOR'u öğrenebilecek bir perseptron asla olamaz.




**Şekil D1:** XOR'u hesaplayamayan bir tek katmanlı perseptron.

Şimdi de iki katmanlı bir perseptronun XOR'u nasıl *öğrenebildiğini* inceleyelim. Şekil D2'ye bakalım. Diyelim ki bu ağ XOR'u doğru bir biçimde hesaplayabiliyor. Ağ üç nörondan oluşur: Nöron 1, Girdi 1 veya Girdi 2 aktifse tetiklenir; Nöron 2, hem Girdi 1 hem de Girdi 2 aktif *değilse* tetiklenir; ve Nöron 3, hem Nöron 1 hem de Nöron 2 tetiklenirse tetiklenir. Özetle, Nöron 3 Girdi 1 *veya* Girdi 2 ak-




**Şekil D2:** XOR'u doğru olarak hesaplayabilen bir iki katmanlı perseptron.

tifse tetiklenir, *her ikisi de* aktifse tetiklenmez.

Biraz daha detaya inip şu dört olası durumu inceleyelim:

1. Öncelikle, diyelim ki girdilerin ikisi de aktif değil. Bu durumda Nöron 1, 0 girdisi 0,5 eşiğinin altında kalacağı için tetiklenmeyecek; Nöron 2, 0 girdisi -1,5 eşiğini aştığı için tetiklenecektir. Nöron 3 ise aldığı 1'lik girdi 1,5 eşiğini aşmadığı için tetiklenmeyecektir.
2. Şimdi de diyelim ki Girdi 1 aktif ama Girdi 2 aktif değil. Nöron 1, aldığı 1'lik girdiyle 0,5 eşiğini aşacağından tetiklenir; Nöron 2 aldığı -1'lik girdiyle -1,5 eşiğini aşacağından, o da tetiklenir. Nöron 3 ise, bu durumda hem Nöron 1 hem Nöron 2'den girdi alacağından, 1,5 eşiğini aşarak tetiklenecektir.
3. Diyelim ki Girdi 2 aktif ama Girdi 1 değil. Üstteki maddede ifade ettiğimiz nedenlerle, Nöron 3 tetiklenir.
4. Son olarak, diyelim ki girdilerin her ikisi de aktif. Bu durumda Nöron 1 aldığı 2'lik girdiyle eşiği aşarak tetiklenir, Nöron 2 ise -2'lik girdisiyle tetiklenmez. Dolayısıyla, aldığı 1'lik girdiyle, Nöron 3 de tetiklenmez.

# Sözlük

**A***: YZ'de en çok kabul gören sezgisel arama yaklaşımı. 1970'lerin başında SHAKEY'nin parçası olarak Stanford Araştırma Enstitüsü'nde geliştirildi. Önceden sezgisel arama daha ziyade amaca özel bir teknikti, A* onu sağlam bir matematiksel temele oturttu.

**ağırlık** (*weight*): Nöral ağlarda, nöronlar arasındaki bağlantılara (**akson**lar) sayısal ağırlıklar verilir. Ağırlık ne kadar fazlaysa, bağlantı ilişik olduğu nöronu o kadar çok etkileyecektir. Bir nöral ağ nihayetinde bu ağırlıklardan ibarettir, bir nöral ağın **eğitimi** görevi ise uygun ağırlıkları bulmaktan geçer.

**ahlaki fail** (*moral agent*): Bir varlık ancak eylemlerinin sonuçlarını ve doğru ile yanlış arasındaki farkı anlayabiliyor, dolayısıyla eylemlerinden sorumlu tutulabiliyorsa bir ahlaki faildir. Hâkim görüş, YZ sistemlerine birer ahlaki fail muamelesi yapılmaması gerektiği –ve hatta yapılamayacağı– yönündedir. Sorumluluğun sahibi sistem değil, o sistemi geliştiren ve kullanıma sokanlardır.

**ajan** (*agent*): Kendine yeten bir YZ sistemi. Genelde birçok YZ yetisini entegre ederek, bir kullanıcı adına otonom olarak çalışır. Ajanların ekseriyetle belli bir ortamda yerleşik olduğu ve aktif bir biçimde çalıştığı varsayılır.

**ajan tabanlı arayüz** (*agent-based interface*): YZ destekli bir **yazılım ajanının** aracılık ettiği bir bilgisayar arayüzüne sahip olma fikri. Pasif bir biçimde kendisine ne yapacağının söylenmesini bekleyen sıradan bir bilgisayar uygulamasının aksine, yazılım ajanı belli bir görevde bizimle birlikte çalışarak adeta bir yardımcı gibi aktif rol alır.

**akson** (*axon*): Bir nöronun diğerleriyle bağlantısını sağlayan bileşeni. Ayrıca bkz. **sinaps**.

**aktivasyon eşiği** (*activation threshold*): Yapay bir **nöron**un aldığı pek çok girdiden sadece bazıları aktiftir. Nöron ancak aktif girdilerinin toplam **ağırlığı** aktivasyon eşiğini aştığı takdirde "tetiklenir".

**AlexNet:** 2012'de çarpıcı gelişmelere imza atarak görsel tanıma alanında çığır açan bir sistem. **Derin öğrenme** sistemlerindeki dönüm noktalarından biri.

**algılama** (*perception*): İçinde bulunduğunuz ortamda etrafınızda neler olduğunu anlama süreci. Algılama **simgesel** YZ'nin başlıca referans noktalarından biriydi.

**algoritmik yanlılık** (*algorithmic bias*): YZ sistemlerinin karar verirken yanlılık sergileme ihtimali. Eğitimde kullanılan veri setinin yanlı olmasından veya yazılımın pek iyi tasarlanmamış olmasından kaynaklanabilir. Tahsis zararı ve temsil zararı biçiminde karşımıza çıkar.

**AlphaGo:** Çığır açan bir Go programı, DeepMind'dan bir ekip tarafından geliştirildi. Mart 2016'da, Kore'nin başkenti Seul'de dünya şampiyonu Lee Sedol'u 4-1 mağlup etti.

**arama, arama ağacı** (*search, search tree*): Temel bir **problem çözme** tekniği. Aramada, bir bilgisayar programı bir amaca nasıl ulaşacağını bulmaya çalışırken, bir **başlangıç durumu**ndan başlayıp sınırlı sayıda eylemi kullanarak bir arama ağacı yaratır.

**Asilomar prensipleri** (*Asilomar principles*): YZ bilimcileri ve yorumcularının 2015 ve 2017'de California'daki Asilomar konferans merkezinde düzenlenen iki toplantı sonucunda geliştirdiği etik YZ ilkeleri.

**başlangıç durumu** (*initial state*): **Problem çözme**de, biz görevimizi yerine getirmeden önce, problemin neye benzediğini anlatan terim.

**Bayes ağı** (*Bayesian network*): Bağlantılı olasılık verilerinden meydana gelen karmaşık ağları yakalamak için bir **bilgi temsili** tertibi. **Bayes Teoremi**'nden yararlanarak bir tür **Bayes çıkarımı** ortaya koyar.

**Bayes Teoremi/çıkarımı** (*Bayes' Theorem/Bayesian inference*): Olasılık teorisinin temel kavramlarından biri olan Bayes teoremi, YZ alanında yeni bir veri veya kanıt elde edildiğinde dünya hakkındaki kanaatlerimizi değiştirip düzeltme imkânı sağlar. Dahası, yeni kanıt hatalı veya belirsiz olduğunda, onu uygun biçimde ele almak için bir yol sunar.

**beklenen fayda** (*expected utility*): **Belirsizlik** durumunda karar verme problemlerinde, belli bir hareket tarzının beklenen faydası, o seçimden elde etmeyi bekleyebileceğiniz ortalama faydadır.

**beklenen faydayı maksimize etme** (*maximizing expected utility*): **Belirsizlik** durumunda karar vermede, birden çok alternatif arasında bir seçim söz konusu olduğunda, rasyonel bir **failin/ajanın** ortalama olarak maksimum faydayı sağlayan, yani **beklenen faydayı** maksimize eden alternatifi seçmesi ilkesi.

**belirleme problemi** (*credit assignment problem*): **Makine öğrenmesi**nde, programın hangi eylemlerinin iyi hangilerinin kötü olduğuna karar verme problemi. Diyelim ki makine öğrenmesi programınız satranç oynadı ve kaybetti. Hangi hamlelerinin kaybetmesine yol açtığını nasıl bilecek?

**belirsizlik** (*uncertainty*): YZ'nin çok çeşitli veçhelerinde karşımıza çıkan bir problem. Elde ettiğimiz enformasyon nadiren kesindir (kesinkes doğru veya yanlıştır), genelde bununla ilgili bir belirsizlik vardır. Aynı şekilde bizler de karar verirken bu kararların ne gibi sonuçlar doğuracağını nadiren kesinkes biliriz, genelde gerçekleşme ihtimali farklı olan birden çok olası sonuç vardır. Dolayısıyla belirsizlikle başa çıkmak YZ'nin temel konularından biridir. Ayrıca bkz. **Bayes Teoremi** ve Ek C.

**belirsizlik durumunda seçim** (*choice under uncertainty*): Böyle bir durumda, vermemiz gereken kararın birden çok olası sonucu vardır; tek bildiğimiz, yapma ihtimalimiz olan her seçimde, her bir sonucun meydana gelme olasılığıdır. Bkz. **beklenen fayda**.

**betik** (*script*): 1970'lerde geliştirilen bir **bilgi temsili** tertibi, sık rastlanan durumlardaki basmakalıp olay dizilimini yakalamayı amaçlar.

**bilgi çıkarma** (*knowledge elicitation*): Bir **uzman sistem** inşa ederken, insan uzmanlardan bilgi temin etme ve bu bilgiyi kodlama problemi.

**Bilgi Gezgini** (*Knowledge Navigator*): Apple'ın 1980'lerde geliştirdiği, **ajan tabanlı arayüz** fikrini takdim eden bir konsept videosu.

**bilgi grafiği** (*knowledge graph*): Google'ın World Wide Web'den otomatikman bilgi alarak geliştirdiği çok geniş bir **bilgi tabanlı** sistem.

**bilgi mühendisi** (*knowledge engineer*): Bilgi tabanlı sistemler inşa etmek için eğitilmiş kişi. **Bilgi çıkarma** üstünde çok çalışırlar.

**bilgi tabanı** (*knowledge base*): Bir **uzman sistem**de bilgi tabanı, genelde **kurallar** biçiminde kodlanmış olan, insan uzman bilgisinden meydana gelir.

**bilgi tabanlı YZ** (*knowledge-based AI*): 1975-1985'te alana hâkim olan YZ paradigması. Problemler hakkındaki, genelde **kurallar** biçiminde karşımıza çıkan, açık ve net bilgileri kullanmaya yoğunlaşır.

**bilgi temsili** (*knowledge representation*): Bilgiyi bilgisayarların işleyebileceği bir biçimde açık ve net olarak kodlama problemi. **Uzman sistemler** döneminde hâkim yaklaşım **kurallar**dan yararlanmaya dayanıyordu, bununla beraber mantık da yaygın olarak kullanılıyordu.

**birinci dereceden mantık** (*first-order logic*): Matematiksel muhakemeye tam bir temel kazandırmak için geliştirilmiş son derece genel bir dil ve muhakeme sistemi. **Mantık tabanlı YZ** paradigmasında bununla ilgili pek çok araştırma yapılmıştır.

**Blok Dünyası** (*Blocks World*): Bir "mikro dünya" simülasyonu. Burada görev prizma veya küp gibi çeşitli nesneleri düzenlemektir. En bilineni SHRDLU'da karşımıza çıkıyor. Blok Dünyası senaryoları bir YZ sisteminin gerçek dünyada karşılaşacağı bilhassa **algılama** gibi zor problemlerin çoğunu soyutlama yoluyla ortadan kaldırmakla eleştirilmiştir.

***ceteris paribus* tercihler** (*ceteris paribus preferences*): Bir YZ sistemine tercihlerimizi belirtirken, "diğer her şey sabitken" varsayımıyla, yani diğer şeylerin mümkün olduğunca değişmeden kalacağını farz ederek hareket ettiğimiz fikri.

**Cyc hipotezi** (*Cyc hypothesis*): **Genel YZ**'nin öncelikle bir bilgi problemi olduğu, dolayısıyla yeterince donanımlı bir **bilgi tabanlı** sistemin Genel YZ'ye sahip olabileceği yolundaki hipotez.

**Cyc projesi** (*Cyc project*): **Bilgi tabanlı YZ** günlerinden meşhur (veya bednam) bir deney. Az çok eğitimli bir insanın dünya hakkında sahip olduğu bütün bilgi verilerek, genel itibarıyla zekâya dayalı bir YZ sistemi inşa etmeye çalışılmış ama nihayetinde olmamıştır.

**çalışma belleği** (*working memory*): Bir **uzman sistem**in çözülmekte olan problem hakkında enformasyona sahip kısmıdır.

**çekişmeli makine öğrenmesi** (*adversarial machine learning*): **Makine öğrenmesi**nin bu dalında, bir programın nihayetinde hatalı çıktılar yaratmasına yol açmaya çalışırız. Zira kimi durumlarda, bir insana hataya mahal vermeyecek kadar açık gibi görünen bir girdi, makine öğrenmesi programını pekâlâ yanıltabilir.

**çıkarım motoru** (*inference engine*): Bir **uzman sistem**in muhakemeye başvuran, **kurallar**dan ve **çalışma belleği**ndeki olgulardan yeni bilgi elde eden kısmı.

**Çince odası** (*Chinese room*): Felsefeci John Searle'ün **güçlü YZ**'nin mümkün olmadığını göstermek için ortaya attığı bir senaryo.

**çok boyutluluk belası** (*curse of dimensionality*): **Makine öğrenmesi**nde, eğitim verinize daha fazla **özellik** dahil etmenin çok daha fazla **eğitim** gerektirmesi problemi.

**çok katmanlı perseptron** (*multi-layer perseptron*): **Katmanlı nöral ağ**ın ilk biçimlerinden biri.

**çok ajanlı sistemler** (*multi-agent systems*): Birden çok **ajan**ın birbiriyle etkileşime girdiği sistemler.

**çözülebilir problem** (*tractable problem*): Bir problem, bir algoritmayla verimli bir biçimde çözülebiliyorsa, çözülebilirdir. **NP-tam** problemler çözülebilir değildir çünkü elimizde bunları verimli bir biçimde çözmeyi garantileyen algoritmalar yoktur. Ayrıca bkz. **P vs NP**.

**dallanma faktörü** (*branching factor*): Bir problemi çözerken, her karar verişinizde göz önünde bulundurmanız gereken alternatiflerin sayısı. Örneğin bir oyun oynuyorsanız, dallanma faktörü verili bir pozisyonda yapabileceğiniz ortalama olası hamle sayısıdır. Dallanma faktörü yaklaşık olarak, üç taş oyununda 4'tür, satrançta 35, Go'da ise 250. Dallanma faktörü ne kadar büyükse, arama ağacı da o kadar çabuk imkânsız bir büyüklüğe ulaşır, dolayısıyla **sezgisel yöntemler**den yararlanarak **arama**ya odaklanmak kaçınılmaz hale gelir.

**dar YZ** (*narrow AI*): İnsanın düşünsel yetiler yelpazesinin tamamına ulaşmaya çalışmak yerine, **Genel YZ**'ye kıyasla son derece spesifik problemlere odaklanan YZ sistemleri inşa etme fikri. Terim YZ çevrelerinden ziyade alan dışından kişilerce kullanılmaktadır.

**davranışsal YZ** (*behavioural AI*): **Simgesel YZ**'ye bir alternatif; 1985-95 döneminde çok ilgi gördü. Bir sistem inşa ederken öncelikle sergilemesi gereken tek tek davranışlara odaklanma, ancak bundan sonra söz konusu davranışların birbiriyle ilişkisini araştırma fikrine dayanıyordu. En çok ilgi gören davranışsal YZ yaklaşımı **kapsama mimarisi**ydi.

**DENDRAL:** Klasik bir **uzman sistem**; kullanıcıların organik bileşenlerin kimyasal yapısını tespit etmesine yardımcı oluyordu.

**derin öğrenme** (*deep learning*): İçinde yaşadığımız yüzyılda **makine öğrenmesi** araştırmalarının yükselişe geçmesinde önemli bir rol oynayan, çığır açıcı teknikler. Ayırt edici özellikleri daha derin, birbirine daha bağlı **nöral ağlar**; daha büyük, dikkatle seçilip düzenlenmiş **eğitim** verisi setlerinin ve birtakım yeni tekniklerin kullanımıdır.

**derinlik öncelikli arama** (*depth-first search*): Problem çözmede kullanılan bir **arama** tekniği tipi. **Arama ağacını** aşama aşama inşa etmek yerine tek bir dal üstünden ilerlenir.

**doğal dili anlama** (*natural language understanding*): Programların İngilizce veya Türkçe gibi olağan insan dillerinde etkileşime girebilmesi.

**eğitim** (*training*): Bir makine öğrenmesi programının görevi, girdiler ile çıktıların ilişkililiğini, bu ilişkililiğin nasıl hesaplanacağı kendisine bildirilmeksizin öğrenmektir. Bunu öğrenmek üzere, program genellikle girdilere ve ilgili girdilere tekabül eden çıktılara dair örnekler verilerek eğitilir. Ayrıca bkz. **gözetimli öğrenme**.

**ELIZA:** Diyalog tabanlı YZ alanında çığır açan bir deney. 1960'larda Joseph Weizenbaum tarafından geliştirilen ELIZA, basit senaryo şablonlarını kullanarak, bir psikoterapisti simüle eder.

**epifenomenalizm** (*epiphenomenalism*): Zihnin ve bilinçli düşüncenin davranışı yönlendirmediği, daha ziyade davranışı bilfiil idare eden süreçlerin yan ürünü olduğu yolundaki görüş.

**erdem etiği** (*virtue ethics*): Etik problemlerle karşılaşınca, değer verdiğimiz etik ilkeleri cisimleştiren bir "erdemli insan" tanımlayıp, o kişi olsa şimdi ne yapardı diye düşünerek ne yapacağımızı seçmemiz gerektiği fikri.

**Evrensel Turing Makinesi** (*Universal Turing Machine*): Genel bir **Turing makinesi** tipi; modern bilgisayar için bir şablon sağlamıştır. Bir Turing makinesi belli bir tarifi/algoritmayı kodlarken, Bir Evrensel Turing Makinesine *her türlü* tarif/algoritma verilebilir.

**faydacılık** (*utilitarianism*): Toplumun refahını azamileştirmeye yönelik edimlerde bulunmayı seçmemiz gerektiği fikri. **Vagon Problemi**nde, bir faydacı beş kişinin hayatını kurtarmak uğruna bir kişiyi ölüme terk etmeyi seçecektir. Ayrıca bkz. **erdem etiği**.

**faydalar** (*utilities*): YZ programlarında **tercihleri** temsil etmek için standart bir teknik: Önce, bütün olası sonuçlara faydalar denen sayısal değerler veririz. Ardından, YZ sistemi faydayı maksimize eden sonuca, yani en çok tercih edilen sonuca yol açacak hareket tarzını hesaplamaya çalışır. Ayrıca bkz. **beklenen fayda** ve **beklenen faydayı maksimize etme**.

**fiziksel bakış açısı** (*physical stance*): Bir varlığın davranışını, fiziksel yapısı ve fizik yasaları çerçevesinde tahmin edip açıklama fikri. Karş. **yönelimsel bakış açısı**.

**Genel YZ** (*General AI*): Bkz. **Yapay Genel Zekâ**.

**geri yayılım** (*backprop/backpropagation*): **Nöral ağların eğitimi** için en önemli algoritma.

**geriye doğru zincirleme** (*backward chaining*): **Bilgi tabanlı** sistemlerde, tesis etmek istediğimiz bir amaçtan (örn. "hayvan etobur") başlama ve elimizdeki verileri (örn. "hayvan et yiyor") kullanarak söz konusu amacın doğrulanıp doğrulanmadığına bakmak suretiyle onu tesis etmeye çalışma fikri. Karş. **ileriye doğru zincirleme**.

**gözetimli öğrenme** (*supervised learning*): **Makine öğrenmesi**nin bu en basit biçiminde, belli bir programı girdilerin ve arzu edilen çıktıların örneklerini göstererek eğitiriz.

**gradyan iniş** (*gradient descent*): **Nöral ağların eğitimi**nde kullanılan bir teknik. Ayrıca bkz. **geri yayılım**.

**Grand Challenge**: DARPA'nın düzenlediği bir sürücüsüz otomobil yarışı. Ekim 2005'te yarışmayı kazanan STANLEY sürücüsüz otomobil çağının artık başladığının habercisiydi.

**güçlü YZ** (*strong AI*): Tıpkı biz insanlar gibi gerçekten zihni, bilinci, farkındalığı vb. olan YZ sistemleri inşa etme amacı. Ayrıca bkz. **zayıf YZ** ve **Genel YZ**. Güçlü YZ'nin gerçekten mümkün olup olmadığını veya mümkünse nasıl bir şey olacağını bilmiyoruz.

**hatalı muhakeme** (*unsound reasoning*): Mantıkta, öncüllerle gerekçelendiremediğimiz sonuçlara ulaşıyorsak, bu muhakemenin hatalı olduğu söylenir. Ayrıca bkz. **sağlam muhakeme**.

**hedef durum** (*goal state*): **Problem çözme**de, görevimizi yerine getirmeyi başardığımız anda problemimizin nasıl görünmesini istediğimizi anlatır.

**HOMER**: 1980'lerde geliştirilen bir **ajan**; bir "deniz dünyası" simülasyonu ortamında yer alır. HOMER sınırlı bir İngilizceyle konuşabiliyor, deniz dünyasında kendine verilen görevleri yerine getirebiliyor ve eylemlerini sağduyuya dayanarak bir ölçüde anlayabiliyordu.

**homunkulus problemi** (*homunculus problem*): Zihin teorisinin klasik problemlerinden biri. Zihin problemini farkında olmaksızın başka bir zihin temsiline havale ederek açıklamaya çalıştığımızda ortaya çıkar.

**ImageNet**: Fei Fei Li'nin geliştirdiği, etiketlenmiş görsellerden oluşan bir veritabanı. **Derin öğrenme**de, eğitim programlarının çeşitli görselleri açıklayabilmesinde müthiş bir etkisi olmuştur.

**ileriye doğru zincirleme** (*forward chaining*): **Bilgi tabanlı** sistemlerde, enformasyondan yola çıkarak muhakeme yoluyla sonuçlara ulaşma. Karş. **geriye doğru zincirleme**.

**kanaat** (*belief*): Bir YZ sisteminin içinde bulunduğu ortam hakkındaki bir enformasyonu. **Mantık tabanlı** YZ'de sistemin kanaatleri **bilgi tabanı** ve **çalışma belleğ**inin içeriklerinden oluşur.

**kapsama mimarisi, kapsama hiyerarşisi** (*subsumption architecture, subsumption hierarchy*): Davranışsal YZ döneminden bir robot mimarisi. Buna göre, robotun arzulanan davranışları bir hiyerarşi şeklinde düzenlenir; davranışın önceliği en alttakinden en üsttekine doğru azalmaktadır.

**karar problemi** (*decision problem*): Evet veya Hayır şeklinde bir cevabı olan matematik problemi. "16'nın karekökü 4 müdür?" veya "7920 asal sayı mıdır?" buna örnektir. Alan Turing'in çözdüğü *Entscheidungsproblem*'de, "Bir algoritmayla çözülemeyecek karar problemleri var mıdır?" diye soruluyordu. Turing herhangi bir algoritması olmayan karar problemleri *olduğunu* gösterdi (örn. durma problemi). Bu tip problemler için **karar verilemez** deniyor.

**karar verilebilir problem** (*decidable problem*): Bir algoritmayla çözülebilen problem.

**karar verilemez problem** (*undecidable problem*): Bildiğimiz kadarıyla, matematiksel olarak kesin bir biçimde, bir bilgisayar (daha doğrusu **Turing makinesi**) tarafından çözülemeyen problem.

**katmanlı nöral ağ** (*layered neural net*): Bir **nöral ağı** düzenlemenin standart yolu. Her katmanın çıktıları bir sonraki katmanı besler. Nöral ağ araştır-

malarının erken dönemlerinde karşılaşılan başlıca problemlerden biri, katmanlı nöral ağların **eğitimi** için bilinen bir yol olmaması, tek katmanlı ağların da yapabildikleri bakımından çok sınırlı kalmasıydı.

**kombinasyon patlaması** (***combinatorial explosion***): Birbiri ardına tercihler yapmamızı gerektiren bir durumda, her bir yeni tercih, gözden geçireceğimiz olasılık sayısını *katlayarak* artırır. **Arama**da karşılaştığımız temel bir YZ problemidir bu, arama ağaçlarının çok çok hızlı büyümesine yol açar.

**kural** (***rule***): "EĞER ..., O ZAMAN ..." şeklinde ifade edilen münferit bir bilgi parçası. Örneğin şu kurala bakalım: "EĞER hayvanın memesi varsa, O ZAMAN hayvan memelidir." Bu kural bize bir hayvanın memesi olduğu enformasyonuna sahipsek, bundan yola çıkarak yeni bir enformasyon, yani hayvanın memeli olduğu enformasyonunu elde edebileceğimizi söyler.

**LISP: Simgesel YZ döneminde yaygın olarak kullanılan bir üst düzey programlama dili.** John McCarthy tarafından geliştirilmiştir. LIST makineleri özel olarak LISP programlama dilini çalıştıracak şekilde tasarlanmış bilgisayarlardı. Ayrıca bkz. **PROLOG**.

**Lighthill Raporu** (***Lighthill Report***): 1970'lerin başında Birleşik Krallık'ta yazılan bir rapor. Dönemin YZ araştırmalarını sert bir dille eleştirir. Dolayısıyla araştırma fonlarında ciddi kesintilere neden olmuştur, ayrıca YZ **kışına** yol açan başlıca etkenlerden biri kabul edilir.

**makine öğrenmesi** (***machine learning***): Zekâya dayalı bir sistemin en temel kapasitelerinden biri. Bir **makine öğrenmesi** programı, bunu nasıl yapacağı kendisine açıkça söylenmeksizin, girdiler ile çıktılar arasındaki bir ilişkiyi öğrenir. **Nöral ağlar** ve **derin öğrenme** yaygın makine öğrenmesi yaklaşımlarıdır.

**mantık** (***logic***): Akıl yürütme için biçimsel bir çerçeve. Ayrıca bkz. **birinci dereceden mantık** ve **mantık tabanlı YZ**.

**mantık programlama** (***logic programming***): Bir programlama yaklaşımı; buna göre, biz sadece bir problem hakkında bildiklerimizi ve amacımızı ifade ederiz, gerisini makine halleder. Ayrıca bkz. **PROLOG**.

**mantık tabanlı YZ** (***logic-based AI***): Zekâya dayalı karar verme sürecinin mantıksal muhakemeye indirgendiği bir YZ yaklaşımı, örneğin bkz. **birinci dereceden mantık**.

**minimaks arama** (***minimax search***): Oyun oynamada başlıca arama tekniklerinden biri. Rakibinizin sizin mümkün olduğunca kötü bir performans sergilemeniz için uğraştığını varsayarak, azami fayda sağlamaya çalışırsınız. Ayrıca bkz. **arama ağacı**.

**Moral Machine:** Kullanıcılara çeşitli **Vagon Problemi** senaryolarında neyin seçilmesi gerektiğinin sorulduğu çevrimiçi bir deney.

**ödül** (*reward*): Bir **pekiştirmeli öğrenme** programının eylemlerine karşılık aldığı, olumlu veya olumsuz geribildirim.

**Tekillik** (*Singularity*): Makine zekâsının insan zekâsını aşacağı varsayılan nokta.

**MYCIN:** 1970'lerden bir **uzman sistem** klasiği. İnsanlarda kan hastalıklarını teşhis eden bir doktor yardımcısı olarak faaliyet göstermiştir.

**Nash dengesi** (*Nash equilibrium*): Oyun teorisinin temel kavramlarından biri. Bir grup karar vericinin her birinin aynı anda, diğerlerinin yaptığı seçimleri göz önünde bulundurarak, yapabileceğinin en iyisini yaptığına kanaat getirip seçiminden memnun olması.

**nöral ağlar** (*neural networks/nets*): "Yapay nöronlar"dan yararlanan bir **makine öğrenmesi** yaklaşımı. **Derin öğrenme**de kullanılan temel teknik. Ayrıca bkz. **perseptron.**

**nöron** (*neuron*): Birbiriyle bağlantılı olan ve **aksonlar** yoluyla iletişim kuran sinir hücrelerinden biri. Beynin temel enformasyon işleme birimi ve **nöral ağların** ilham kaynağı.

**NP-tam** (*NP-complete*): Verimli bir biçimde çözülmeye direnen bir bilgi işlem problemleri sınıfı. NP-tamlık teorisinin geliştirildiği 1970'lerde pek çok YZ probleminin aslında NP-tam olduğu keşfedildi. Ayrıca bkz. **P vs NP.**

**ontolojik mühendislik** (*ontological engineering*): Bir **uzman sistem**de (ve daha genel olarak bir **bilgi tabanlı** sistemde), sisteminizdeki bilgiyi temsil etmek için kullanacağınız kavramsal söz dağarcığını tanımlama görevi.

**opaklık** (*opaqueness*): Bir nöral ağın sahip olduğu uzmanlığın bir dizi sayısal ağırlık halinde kodlanmış olması sorun yaratır çünkü bu ağırlıkların "ne anlama geldiğini" söyleyemeyiz. Yani mevcut nöral ağlar kararlarını açıklayamaz veya gerekçelendiremezler.

**oyun teorisi** (*game theory*): Stratejik muhakeme teorisi. YZ alanında YZ sistemlerinin birbiriyle nasıl etkileşime girebileceğini ve girmesi gerektiğini anlamayı sağlayacak bir çerçeve olarak yaygın biçimde kullanılır.

**öncüller** (*premises*): **Mantık**ta başlama noktanızdır. Öncüllerden, mantıksal muhakeme yoluyla, birtakım sonuçlar çıkarırsınız.

**önsel olasılık** (*prior probability*): "Önsel" burada "daha fazla enformasyon edinmeden önce" anlamında kullanılıyor: yani bir hipoteze, hakkında henüz daha fazla enformasyon edinmemişken iliştirilen olasılık.

**öznitelik** (*feature*): Bir **makine öğrenmesi** programının kararlarını dayandırdığı verilerin bileşenleri.

**öznitelik çıkarma** (*feature extraction*): **Makine öğrenmesi**nde, bir veri setindeki hangi özelliklerin **eğitim** verisi olarak seçilmesi gerektiğine karar verme problemi.

**P vs NP**: **NP-tam** problemlerin her zaman verimli bir biçimde çözülüp çözülemeyeceği problemi. Günümüzde hâlâ çözülememiş olan, yakınlarda da çözülebilecek gibi görünmeyen en büyük matematik problemlerinden biridir. (Başka bir ifadeyle bu, P = NP mi yoksa P ≠ NP mi problemidir.)

**pekiştirmeli öğrenme** (*reinforcement learning*): Bir **makine öğrenmesi** biçimi; buna göre, bir **ajan** belli bir çevrede edimlerde bulunur ve eylemlerine karşılık **ödül** biçiminde geribildirimler alır.

**perseptron modeli** (*perceptron model*): Perseptron bugün hâlâ güncelliğini koruyan bir **nöral ağ** tipidir. Konuyla ilgili araştırmalar esasen 1960'larda başlamış, 1970'lerin başı itibarıyla tek katmanlı perseptron modellerinin öğrenebileceklerinin ne kadar sınırlı olduğunun anlaşılmasıyla iyice sönümlenmiştir.

**planlama** (*planning*): Bir **başlangıç durumu**nu **hedef durum**a getiren bir eylem sıralamasını bulmak. Ayrıca bkz. **arama**.

**problem çözme** (*problem solving*): YZ'de, bir problemi bir **başlangıç durumu**ndan **hedef durum**a getiren doğru eylem sıralamasını bulmak. YZ'de problem çözmeye standart yaklaşım **arama**dır.

**PROLOG**: **Birinci dereceden mantığa** dayanan bir programlama dili, bilhassa **mantık tabanlı** YZ döneminde çok revaçtaydı.

**PROMETHEUS**: Avrupa'da 1987'den 1995'e kadar yürütülen çığır açıcı bir sürücüsüz otomobil teknolojisi deneyi.

***qualia***: Şahsi zihinsel deneyimler. Örneğin bir kahvenin kokusu, sıcak bir günde soğuk bir içecek içmek.

**R1/XCON**: 1970'lerden klasik bir **uzman sistem**; DEC tarafından VAX bilgisayarların konfigürasyonuna yardımcı olmak üzere geliştirilmiştir. YZ sayesinde para kazanılabileceğini gösteren ilk örneklerdendir.

**Robotiğin Üç Yasası** (*Three Laws of Robotics*): Bilimkurgu yazarı Isaac Asimov'un 1930'larda ortaya koyduğu bu üç yasa, YZ davranışı için bir tür etik çerçeve önerir. Enine boyuna düşünülerek tasarlanmış olsalar da, doğrudan uygulamaya geçirilmeleri mümkün değildir. Hatta tam olarak ne kastettikleri bile belli değildir.

**sağduyuya dayalı akıl yürütme** (*common-sense reasoning*): Bu geniş kapsamlı terim esasen insanların rahatlıkla yapabildiği ama **mantık tabanlı** YZ için son derece zor olduğu anlaşılan gündelik muhakeme türüne işaret eder.

**sağlam muhakeme** (*sound reasoning*): Elde edilen sonuçlar **öncül**lerle gerekçelendirilebiliyorsa muhakemenin sağlam olduğu söylenir.

**Sally-Anne testi** (*Sally-Anne test*): Bir varlığın **Zihin Teorisi**, yani başkalarının kanaat ve arzuları hakkında akıl yürütme yetisi olup olmadığını tespit etmeyi amaçlayan bir test.

**sapkın örnekleme** (*perverse instantiation*): Bir YZ sisteminin istediğinizi yapması, ama bunun beklediğiniz şekilde olmaması.

**semantik ağlar** (*semantic nets*): **Bilginin temsili** için bir tertip; kavramlar ile varlıklar arasındaki ilişkileri grafiksel bir yazımdan yararlanarak yakalamamıza imkân sağlar.

**sensör** (*sensor*): Robota ham algı verisi sağlayan cihaz. Kameralar, lazer radarlar, ultrasonik menzilölçerler ve çarpma dedektörleri tipik sensörlerdir. Ham algı verisini yorumlamak başlıca zorluklardan biridir.

**sezgisel yöntemler, sezgisel arama** (*heuristics, heuristic search*): **Sezgisel yöntemler, aramay**ı odaklayan "ampirik kurallar"dır. Aramayı doğru istikamete odaklamayı *garantilememeleri* bakımından da ampirik kurallardır. Ayrıca bkz. A*.

**SHAKEY**: Otonom robotlar konusunda çığır açan bir deney. 1960'ların sonunda Stanford Araştırma Enstitüsü'nde geliştirilmiştir. Pek çok önemli YZ teknolojisinin öncüsüdür.

**SHRDLU**: **Altın Çağ**'da takdir edilen ama daha sonraki dönemlerde, simüle edilen mikro dünyalara odaklanmakla eleştirilen bir sistem.

**simgesel YZ** (*symbolic AI*): Akıl yürütme ve planlama süreçlerini açıkça modellemeyi kapsayan bir YZ yaklaşımı.

**sinaps** (*synapse*): Nöronların iletişim kurmasını sağlayan "bağlantı noktaları".

**sonradan ortaya çıkan özellik** (*emergent property*): Birçok bileşenden oluşan bir sistemde, bileşen sistemlerinin etkileşimi sonucunda, genelde beklenmedik veya öngörülmedik bir şekilde zuhur eden nitelik.

**sonuç** (*consequent*): Bir uzman sistemde kullanılan haliyle "EĞER ..., O ZAMAN ..." kuralında sonuç, O ZAMAN'dan sonra gelen kısımdır. Örneğin "EĞER hayvanın memesi varsa, O ZAMAN hayvan memelidir" kuralında, "hayvan memelidir" ifadesi sonuçtur.

**STANLEY**: 2005 DARPA **Grand Challenge**'ı kazanan robot. Stanford Üniversitesi'nde geliştirilen bu sürücüsüz otomobil yaklaşık 212 kilometrelik parkuru ortalama olarak saatte yaklaşık 32 kilometre hızla tamamladı.

**STRIPS**: Stanford Araştırma Enstitüsü'nde **SHAKEY** projesinin bir parçası olarak geliştirilen çığır açıcı bir **planlama** sistemi.

**şart** (*antecedent*): Bir uzman sistemde kullanılan haliyle "EĞER ..., O ZAMAN ..." kuralında şart, "EĞER"den hemen sonra gelen koşul kısmıdır. Örneğin "EĞER hayvanın memesi varsa, O ZAMAN hayvan memelidir" kuralında şart "hayvanın memesi varsa"dır.

**tasarımsal bakış açısı** (*design stance*): Bir varlığın davranışını, o varlığın ne için tasarlandığına bakarak anlamaya ve tahmin etmeye çalışma fikri. Örneğin bir saatin üstündeki sayıları zaman olarak yorumlarız çünkü saat zamanı göstermek üzere tasarlanmıştır.

**tasım** (*syllogism*): Mantıksal muhakeme için, antik çağlardan beri bilinen basit bir örüntü.

**tercihler, tercih ilişkisi** (*preferences, preference relation*): Tercihlerinizin, her olası alternatif çiftini sıraladığınız bir betimlemesi. Bir **ajan** sizin adınıza edimde bulunacaksa, sizin açınızdan mümkün olan en iyi seçimi yapabilmesi için tercihlerinizi bilmesi gerekir.

**ters pekiştirmeli öğrenme** (*inverse reinforcement learning*): Bir **makine öğrenmesi** programının bir insanın ne yaptığını gözlemleyip, bu gözlemlerine dayanarak bir **ödül** sistemini öğrenmeye çalışması durumu.

**tetiklenme** (*fire*): **Bilgi tabanlı** sistemler bağlamında, **çalışma belleğ**indeki enformasyon kuralın **şart**ına uyuyorsa kural tetiklenir, böylece kuralın **sonucu**nu çalışma belleğine ekleme imkânı doğar.

**toplumsal refah** (*social welfare*): Bir topluma sağlanan toplam faydayı, yani toplumun genel itibarıyla ne durumda olduğunu ölçmeye yönelik her türlü girişim bir toplumsal refah ölçütüdür.

**TouringMachines**: 1990'ların ortasından tipik bir **ajan** tasarımı. Burada ajanın kontrolü sırasıyla reaksiyon gösterme, planlama ve modellemeden sorumlu üç katmana ayrılır.

**Turing makinesi** (*Turing machine*): Bir matematiksel problem çözme makinesi; bir problemi çözmek için belli bir tarifi cisimleştiren makine. Bilgisayarla çözülebilen her matematik problemi Turing makinesiyle de çözülebilir. Alan Turing tarafından *Entscheidungsproblem*'i çözmek için icat edilmiştir. Ayrıca bkz. **Evrensel Turing Makinesi**.

**Turing testi** (*Turing test*): Turing'in makineler "düşünebilir" mi sorusunu ele alırken ortaya attığı test. Buna göre, eğer bir varlıkla bir süreliğine etkileşime girdiğiniz halde makine mi yoksa insan mı olduğunu tam olarak kestiremiyorsanız, onu insan benzeri zekâya sahip bir varlık kabul edersiniz. Son derece yaratıcı ve etkili bir çözümdür, yine de gerçek bir YZ testi olarak çok da ciddiye almamak gerekir.

**tümdengelim** (*deduction*): Mantıksal muhakeme; mevcut bilgiden yeni bilgi türetme.

**Urban Challenge:** Kırsalda yapılan 2005 DARPA **Grand Challenge**'ın ardından, 2007'de otonom araçları kent ortamında test etmek üzere düzenlenen yarış.

**uzman sistem** (***expert system***): Dar sınırları iyi tanımlanmış bir alanda problem çözmek için insan uzman bilgisini kullanan bir sistem. **MYCIN, DENDRAL** ve **R1/XCON** klasik örnekleridir. 1970'lerin sonundan 1980'lerin ortalarına kadar, uzman sistemler inşa etmek YZ araştırmalarının başlıca odaklarından biriydi.

**üst düzey programlama dili** (***high-level programming language***): Programı çalıştıran gerçek bilgisayarın kimi alt düzey ayrıntılarını bilgisayarı programlayan kişiye göstermeyen programlama dili. Üst düzey programlama dilleri, en azından teoride, makineden bağımsızdır; belli bir program farklı bilgisayarlarda çalışabilir, sözgelimi Python ve Java. Ayrıca John McCarthy'nin LISP'i de ilk örneklerindendir.

**ütopyacı** (***utopian***): YZ'nin ve diğer yeni teknolojilerin bize ütopik bir gelecek sunacağı (teknolojinin bizi çalışmadan azat edeceği vb.) görüşünü savunan kişi.

**Vagon Problemi** (*Trolley Problem*): Esasen 1960'larda ortaya atılan bir etik muhakeme problemi: Makası değiştirmezseniz beş kişi, değiştirirseniz bir kişi ölecektir; bu durumda makası değiştirir misiniz yoksa değiştirmez misiniz?

**Winograd şablonları** (*Winograd schemas*): **Turing testi**nin bir alternatifi. Tek kelimesi farklı, kalan kısımları aynı olan iki cümle verilir. Bu tek kelime iki cümlenin anlamını tamamen değiştirmektedir. Test de iki cümle arasındaki bu farkı anlama esasına dayanır. Metni *kavramayı* gerektirdiği için, Winograd şablonlarının Turing testinde çokça kullanılan ucuz numaraları yakaladığı düşünülür.

**Yapay Genel Zekâ, YGZ** (*Artificial General Intelligence*, AGI): İnsanın sahip olduğu düşünsel yetiler yelpazesinin tamamına –planlama, akıl yürütme, doğal dilde diyaloğa girme, şaka yapma, hikâye anlatma, hikâyeyi anlama, oyun oynama vb. bütün yetilere– muktedir YZ sistemleri inşa etme amacı.

**yazılım ajanı** (***software agent***): Robotlar gibi fiziksel dünyada değil de bir yazılım ortamında ikamet eden ajan. Bunu bir yazılım robotuymuş gibi düşünün.

**yerleştirilmiş** (***situated***): **Davranışsal YZ** döneminin temel fikirlerinden biri; YZ'nin ilerlemesi için, **uzman sistemler** örneğinde genelde olduğu üzere cisimsiz değil, belli bir ortama bilfiil yerleşmiş olan ve eylemleriyle bu ortamı etkileyen **ajan**lar geliştirmek gerektiği düşünülüyordu.

**yönelimsel bakış açısı** (*intentional stance*): Bir varlığın davranışını tahmin etmeye ve açıklamaya çalışırken, ona kanaat ve arzular gibi **zihinsel haller** atfetme, bu kanaat ve arzular temelinde rasyonel bir biçimde edimde bulunacağını varsayma fikri.

**yönelimsel sistem** (*intentional system*): **Yönelimsel bakış açısı** ile nitelenen sistem türü.

**YZ kışı** (*AI winter*): YZ'ye son derece eleştirel yaklaşan Lighthill Raporu'nun 1970'lerin başında yayımlanmasının hemen ardından gelen dönem. Alana şüpheyle bakılmaya başlanmış, YZ araştırmalarının finansmanında ciddi kesintilere gidilmiştir. Bu dönemi **bilgi tabanlı YZ** izlemiştir.

**YZ'nin Altın Çağı** (*Golden Age of AI*): YZ araştırmalarının erken dönemi, yani kabaca 1956-75 yılları (**YZ kışı** öncesi). Bu dönemde yapılan çalışmalarda "böl ve yönet" yaklaşımı benimsenmiştir: Zekâya dayalı davranışın tek tek her *bileşenini* sergileyen sistemler inşa edilirse, ileride bunların bir bütün haline getirilebileceği umuluyordu.

**Zarar Değerlendirme Risk Aracı** (*Harm Assessment Risk Tool*, HART): Birleşik Krallık'taki Durham polis teşkilatı için geliştirilmiş olan bir **makine öğrenmesi** sistemi. Bir şüphelinin serbest mi bırakılacağına yoksa gözaltına mı alınacağına karar verirken polis memurlarına yardımcı olur.

**zayıf YZ** (*weak AI*): Anlama yetisine (bilince, zihne, özfarkındalığa vb.) sadece sahipmiş gibi görünen, bunlara gerçekten sahip olma iddiası taşımayan makineler inşa etme amacı. Ayrıca bkz. **güçlü YZ** ve **Genel YZ**.

**Zihin Teorisi** (*Theory of Mind*): Klinik açıdan normal yetişkin insanların başka insanların zihinsel halleri (kanaatleri, arzuları, maksatları) hakkında akıl yürütmelerine imkân sağlayan gündelik kapasite. Ayrıca bkz. **yönelimsel bakış açısı** ve **Sally-Anne testi**.

**zihin-beden problemi** (*mind-body problem*): Bilimin en temel problemlerinden biri: Beyin/bedendeki fiziksel süreçler ile zihin ve bilinçli deneyim arasında nasıl bir ilişki vardır?

**zihinsel haller** (*mental states*): Zihnin kilit bileşenlerinden biri. Örneğin kanaatler, arzular vb. Ayrıca bkz. **yönelimsel bakış açısı**.

**zor problem olarak bilinç** (*hard problem of consciousness*): Fiziksel süreçlerin öznel bilinçli deneyimlere nasıl ve neden yol açtığını anlama problemi. Ayrıca bkz. *qualia*.

# Notlar

## Giriş

1. https://www.cnbc.com/2018/02/01/google-ceo-sundar-pichai-ai-is-more-important-than-fire-electricity.html.
2. https://medium.com/syncedreview/artificial-intelligence-is-the-new-electricity-andrew-ng-cc132ea6264.

## 1. Turing'in Elektronik Beyinleri

1. A. Hodges, *Alan Turing: The Enigma*, Burnett Books Ltd, 1983; Türkçesi: *Enigma*, çev. Zeynep Ünalan, Olasılık, 2017.
2. Turing'in, o muazzam bilimsel mirasının yanı sıra, Birleşik Krallık'ta derin bir sosyal mirası da vardı. Uzun vadede son derece etkili bir kampanyanın ardından nihayet 2014'te, çoktan hayatını kaybetmiş olan Turing'den özür dilenerek itibarı iade edildi; çok geçmeden, hükümet aynı kanun maddesi uygulanarak cezalandırılan bütün erkeklere hitaben bir özür yayımladı.
3. Bir sayının asal olup olmadığını kontrol etmenin en aşikâr yoludur bu, en zarif veya en randımanlı yolu değil. Sözgelimi antik çağlardan beri bilinen, Eratosthenes Kalburu adı verilen pek hoş bir alternatifi daha vardır.
4. Buradan itibaren Evrensel Turing Makinesi ile Turing Makinesi arasında ayrım yapmayı bırakıp ikisi için de sadece ikinci terimi kullanacağım.
5. Turing *Entscheidungsproblem*'i çözme şerefini Princetonlı matematikçi Alonzo Church ile paylaşır, zira Church Turing'den bağımsız olarak ve kısa süre önce çözüme dair çok farklı bir kanıt ortaya koymuştur. Gelgelelim bu ikisinden Turing'inki kesin kanıt olarak kabul edilir, çünkü daha doğrudan, daha bütüncül ve daha erişilebilirdir; ayrıca içerimleri de, Turing makinesinin icadı yoluyla dünyayı değiştirmiştir.
6. Daha doğru bir ifadeyle, algoritma bir tariftir, program ise Python veya Java gibi gerçek bir programlama dilinde kodlanmış bir algoritmadır. Dolayısıyla algoritma, programlama dilinden bağımsızdır.
7. Turing makinelerine gereken aslında bundan çok daha kaba bir programlama biçimidir. Burada sıraladığım talimatlar daha ziyade nispeten düşük

seviyeli bir programlama dilinde göreceğiniz türden talimatlar, yine de bir Turing makinesi programına kıyasla çok daha soyut kalıyorlar (ve anlaşılmaları da daha kolay).

8. T. H. Cormen, C. E. Leiserson ve R. L. Rivest, *Introduction to Algorithms*, MIT Press ve McGraw-Hill, 1990.
9. A. M. Turing, "Computing Machinery and Intelligence", *Mind*, No. 49, 1950, s. 433-60.
10. Bu diyalog ELIZA'nın her Macintosh bilgisayarda bulunan bir versiyonuyla gerçekleştirilmiştir. Bilgisayarınız Mac'se Uygulamalar / İzlenceler / Terminal'den kendiniz de deneyip görebilirsiniz. Terminal penceresini açınca, karşınıza bir sürü ıncık cıncık yazı çıkacak. Bunların yüklenmesi bitince sırasıyla "Esc" tuşuna (klavyenizin sol üst köşesinde) ve "X" tuşuna basın, ardından "doctor" yazın ve son olarak "Enter"a basın. İşte bu kadar. Ama gerçek olmadığını unutmayın!
11. https://cs.nyu.edu/~davise/papers/WinogradSchemas/WSCollection.html.
12. Ne yazık ki literatürün tutarlı veya oturmuş bir terminolojisi yok. Pek çok kişi "yapay genel zekâ"yı, özfarkındalık sahibi olup olmadıkları gibi felsefi soruların hiç üstünde durmadan, makinelerde genel amaçlı ve insan düzeyinde zekâ üretme amacı anlamında kullanıyor. Bu anlamda yapay genel zekâ kabaca Searle'ün zayıf YZ'sine denk düşer. Ne var ki terim kimi zaman, işleri daha da karıştıracak şekilde, Searle'ün güçlü YZ'si gibi bir anlamda da kullanılmaktadır. Ben bu kitapta terimi zayıf YZ gibi bir anlamda kullanıyorum ve kısaca "Genel YZ" diyorum.
13. https://www.humanbrainproject.eu/en/.

## 2. YZ'nin Altın Çağı

1. Bunların en etkililerinden biri, John McCarthy'nin fikir babası olduğu, bilgisayar kullanma zamanını paylaşma fikriydi. Bilgisayar başında geçirilen zamanın çoğunda bilgisayar atıl kalıyor, yani kullanıcının bir şeyler yazmasını veya bir programı çalıştırmasını bekliyor oluyordu. McCarthy bu "atıl zamanı" başkalarıyla paylaşmayı akıl etti, böylece pek çok insan aynı anda bilgisayarı kullanabilecekti. Dolayısıyla pahalı bilgisayarları çok daha verimli kullanmak mümkün hale geldi.
2. Gerçekte açılımı "Liste İşlemcisi"dir. LISP tamamen *simgesel* bilgi işlemle ilgilidir, bunun anahtarı da simge listeleridir.
3. S. Nasar, *A Beautiful Mind*, Simon & Schuster, 1998; Türkçesi: *Akıl Oyunları*, çev. Petek Demir, Altın Kitaplar, 2002.
4. J. McCarthy ve diğ., "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, 1955" (yeniden basım *AI Magazine*, 24: 4, 2006, s. 12-14).

5. T. Winograd, *Understanding Natural Language*, Academic Press, 1972.
6. Doğal dilde, bir diyalogda evvelce karşımıza çıkan bir şey için sözgelimi "o" zamirinin kullanılmasına sık rastlanır. Doğal dildeki bir diyaloğu anlamaya çalışan veya gerçekleştiren bir bilgisayar programının böyle referansları çözebilmesi gerekir. Bu günümüzde bile hâlâ aşılması gereken bir zorluk olarak karşımızda duruyorken, SHRDLU'nun bu durumla (sınırlı da olsa) başa çıkma becerisi zamanında haliyle çığır açıcı bir yenilik olarak değerlendirilmiştir.
7. R. E. Fikes ve N. J. Nilsson, "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving", *Artificial Intelligence*, 2(3-4), 1971, s. 189-208.
8. SHAKEY'yi kontrol eden bilgisayar, 1960'ların sonu itibarıyla dönemin en ileri teknolojili ana bilgisayarlarından biri olan PDP-10'du. Ağırlığı bir tonu aşan, geniş bir oda büyüklüğünde olan PDP-10'un belleği bir megabayta kadar çıkıyordu; cebimdeki akıllı telefon ise bunun 4000 katı bir belleğe sahip, haliyle de PDP-10'a kıyasla inanılmaz hızlı.
9. https://www.youtube.com/watch?v=TamMqvk4Bb8.
10. https://www.youtube.com/watch?v=AVaITA7eBZE.
11. Bu kesin bir hesaplama değil, sadece yaklaşık bir değerdir; söz konusu sayıların ölçeğine dair bir fikir verme niyetiyle ortaya konmuştur.
12. Teknik açıdan, diyelim ki $b$ arama probleminin dallanma faktörü, $d$ ise arama ağacının derinliği olsun. Bu durumda ağacın en alt aşamasında (d derinliğinde) $b^d$ sayıda durum olacaktır.

$$\underbrace{b^d = b \times b \times b \times \ldots \times b}_{d \text{ kere}}$$

Dallanma faktörü, teknik bir terimle, üstel büyüme sergiler. YZ literatüründe hiç karşılaştığımı sanmıyorum ama kimi kaynaklarda geometrik büyüme terimi de kullanılıyor.
13. P. E. Hart, N. J. Nilsson ve B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", *IEEE Transactions on Systems Science and Cybernetics* 4(2), 1968, s. 100-107.
14. Böyle bilgi işlem problemlerinin isimleri olur; mesela bununki "Bağımsız Küme".
15. Seyyar satıcı problemini biraz basitleştirerek aldım. Tam tanımı ise şöyle: Elimizde bir dizi şehir var, buna $C$ diyelim. $C$'deki her $i$ ve $j$ şehir çifti arasında belli bir mesafe var, buna da $d_{i,j}$ diyelim. Başka bir deyişle $i$ şehri ile $j$ şehri arasındaki mesafe $d_{i,j}$ kadar. Bir de "sınır" var, $B$, bir depo benzinle toplam ne kadar mesafe katedilebileceğini gösteriyor. Şimdi sorumuz şu: Toplamda $B$'den fazla mesafe katetmeden, belli bir sırayı izleyerek şehirden şehre geçip, sonunda bu yolculuğa başladığımız yere dön-

memize imkân sağlayan bir rota var mıdır?
16. NP-tamlık ve P v NP hakkında daha fazla bilgi için bkz. Okuma Önerisi bölümü.

## 3. Bilgi Güçtür


1. P. Winston ve B. Horn, *LISP*, Pearson, 3. Basım, 1989.
2. E. H. Shortliffe, *Computer-Based Medical Consultation: MYCIN*, American Elsevier, 1976.
3. R. Schank ve R. P. Abelson, *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*, Psychology Press, 1977.
4. W. A. Woods, "What's in a Link? Foundations for Semantic Networks", *Representation and Understanding – Studies in Cognitive Science* içinde, haz. D. G. Borow ve A. Collins, Morgan-Kaufmann, 1975.
5. D. McDermott, "Tarskian Semantics, or, No Notation without Denotation!", *Cognitive Science* 2(3): s. 277-82, 1978. Makalenin başlığı ("Düzanlam Olmadan Gösterim de Yok") biraz Amerikan Bağımsızlık Savaşı zamanından kalma "Temsil Olmadan Vergi de Yok" sloganını çağrıştırır (McDermott bu makaleyi 1976'da, ABD'de bağımsızlığın 200. yıl kutlamalarına nispeten yakın tarihlerde yazmıştır).
6. J. McCarthy, "Concepts of Logical AI", yayımlanmamış metin.
7. W. F. Clocksin ve C.S. Mellish, *Programming in PROLOG*, Springer-Verlag, 1981.
8. PROLOG'da kullanılan tümdengelim biçimine "çözüm" (*resolution*) denir. Esasen 1960'larda icat edilen bu teknik, PROLOG'da kullanılan kurallara etkili bir biçimde uygulanabilir.
9. D. H. D. Warren, "Generating Conditional Plans and Programs", *Proceedings of the Second Summer Conference on Artificial Intelligence and Simulation of Behaviour (AISB-76)* içinde, Edinburgh, Temmuz 1976.
10. R. V. Guha ve D. Lenat, "Cyc: A Midterm Report", *AI Magazine*, 11(3), 1990.
11. V. Pratt, "CYC Report", yayımlanmamış metin, 1994. Şu adresten erişilebilir: http://boole.stanford.edu/cyc.html.
12. R. Reiter, "A Logic for Default Reasoning", *Artificial Intelligence*, 13, 1980, s. 81-132.
13. Söz konusu senaryonun standart görsel temsili elmas şeklinde olduğu için, bu örnek *Nixon elması* adıyla bilinir.

## 4. Robotlar ve Rasyonalite

1. R. A. Brooks, "Intelligence Without Representation", *Artificial Intelligence*, 47, 1991, s. 139-59.
2. İşin enteresan tarafı, insan zekâsının bu üstün veçheleri neokorteks tarafından idare edilir ve insan evrimine ilişkin kayıtlar bize beynin bu kısmının nispeten yakın tarihte evrildiğini göstermektedir. Yani akıl yürütme ve problem çözme insan için nispeten yeni becerilerdir, atalarımız evrimsel tarihin büyük bir kısmında bunlar olmadan başının çaresine bakmak durumunda kalmıştır.
3. S. Russell ve D. Subramanian, "Provably Bounded-Optimal Agents", *Journal of Artificial Intelligence Research*, 2, 1995.
4. https://www.irobot.co.uk.
5. R. A. Brooks, "A Robot That Walks: Emergent Behaviors from a Carefully Evolved Network", *Proceedings of the 1989 Conference on Robotics and Automation* içinde, Scottsdale, Arizona, Mayıs 1989.
6. I. A. Ferguson, "TouringMachines: Autonomous Agents with Attitudes", *IEEE Computer*, 25(5), s. 51-55, 1992. "TouringMachines" isminin "Turing Makineleri"ne gönderme yapan bir kelime oyunu olduğu açık. TouringMachines'i geliştiren Innes Ferguson neredeyse çeyrek asırdır çok sevdiğim bir arkadaşım ama bu şakasına gülüp geçebilir miyim gerçekten bilmiyorum. Otuz sene sonra bile insanların hâlâ bu mimari hakkında yazılar yazıyor olacağını bilse, bu şaka vaktiyle ona da bu kadar komik gelmezdi bence.
7. S. Vere ve T. Bickmore, "A Basic Agent", *Computational Intelligence*, 6, 1990, s. 41-60.
8. Tarihsel olguları doğru aktarmak adına, grafik kullanıcı arayüzünü ve masaüstü metaforunu bizzat Apple'ın icat etmediğini belirtmem lazım. Bunu yapanlar Xerox Palo Alto Araştırma Merkezi'nden (PARC) araştırmacılardı. Fakat Apple buradaki potansiyeli gerçekleştirip ürüne dönüştürmüştü.
9. https://www.youtube.com/watch?v=umJsITGzXd0.
10. P. Maes, "Agents That Reduce Work and Information Overload", *Communications of the ACM*, 37(7), 1994, s. 30-40.
11. O. Etzioni ve D. Weld, "A Softbot-based Interface to the Internet", *Communications of the ACM*, 37(7), 1994, s. 72-76.
12. J. von Neumann ve O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 1944.
13. Daha rahat anlaşılması için, bu örnekte faydayı paraya eşitliyorum. Pratikte, para ve fayda çoğu zaman birbiriyle ilişkilidir ama genelde aynı şey değildir. Fayda teorisine sadece parayla ilgili bir teori muamelesi yapar-

sanız pek çok iktisatçıyı karşınıza alırsınız, çünkü söz konusu teori esasen tercihleri yakalamanın ve hesaplamanın sayısal bir yoludur.
14. S. Russell ve P. Norvig, *Artificial Intelligence - A Modern Approach*, Pearson, 3. Basım, 2016, s. 611.
15. R. Murphy, *An Introduction to AI Robotics*, MIT Press, 2001.
16. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.
17. M. Wooldridge, *An Introduction to MultiAgent Systems*, John Wiley, 2. Basım, 2009.
18. A. Rubinstein ve M. J. Osborne, *A Course in Game Theory*, MIT Press, 1994.
19. B. Selman, H. J. Levesqueand ve D. G. Mitchell, "A New Method for Solving Hard Satisfiability Problems", *Proceedings of the Tenth National Conference on AI (AAAI1992)* içinde, San Jose, California, 1992.

## 5. Çığır Açan "Derin" Atılımlar

1. Bu konuda birbirinden çok farklı rakamlar ortaya atıldı. *Guardian*'daki şu habere dayanarak 400 milyon sterlin olduğunu yazıyorum: https://www.theguardian.com/technology/2014/jan/27/google-acquires-uk-artificial-intelligence-startup-deepmind.
2. M. Minsky ve S. Papert, *Perceptrons: An Introduction to Computational Geometry*, MIT Press, 1969.
3. https://www.nytimes.com/1958/07/08/archives/new-navy-device-learns-by-doing-psychologist-shows-embryo-of.html.
4. D. E. Rumelhart ve J. L. McClelland (haz.), *Parallel Distributed Processing*, MIT Press, 1986.
5. D. E. Rumelhart, G. E. Hinton ve R. J. Williams, "Learning Representations By Back-propagating Errors", *Nature*, 323, 1986, s. 533-36.
6. I. Goodfellow, Y. Bengio ve A. Courville, *Deep Learning*, MIT Press, 2016; Türkçesi: *Derin Öğrenme*, çev. F. Vural, R. Cinbiş, S. Kalkan, Buzdağı, 2021.
7. Nöronların sayısının tarih boyunca nasıl bir oranda arttığına bakınca, yaklaşık 40 küsur yıl sonra bir insan beynindeki kadar çok sayıda yapay nörona ulaşmayı bekleyebiliriz. Gerçi bu, yapay nöral ağların yaklaşık 40 yılda insanın zekâ seviyesine ulaşacağı anlamına gelmiyor, çünkü beyin büyük bir nöral ağdan ibaret değil. Belirleyici bir yapısı da var.
8. http://www.image-net.org.
9. https://wordnet.princeton.edu.
10. A. Krizhevsky, I. Sutskever ve G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Proceedings of Neural In-*

*formation Processing Systems* içinde, 2012, s. 1106-14.
11. I. Goodfellow ve diğ., "Explaining and Harnessing Adversarial Examples", arXiv:1412.6572. [YZ ve makine öğrenmesi alanındaki pek çok güncel makaleye, açık bir çevrimiçi arşiv olan "arXiv"den ulaşmak mümkün. Bu dipnottaki gibi bir atıfla karşılaştığınızda, https://arxiv.org adresine gidip, makalenin referans numarası olan sayıyı, mesela bu örnekte 1412.6572'yi girmeniz yeterli.]
12. V. Mnih ve diğ., "Playing Atari with Deep Reinforcement Learning", arXiv:1312.5602v1.
13. V. Mnih ve diğ., "Human-Level Control through Deep Reinforcement Learning", *Nature*, 518, 2015, s. 529-33.
14. D. Silver ve diğ., "Mastering the Game of Go with Deep Neural Networks and Tree Search", *Nature*, 529, 2016, s. 484-89.
15. https://www.imdb.com/title/tt6700846/.
16. D. Silver ve diğ., "Mastering the Game of Go without Human Knowledge", *Nature*, 50, 2017, s. 354-59.
17. https://translate.google.com.
18. Marcel Proust, *Kayıp Zamanın İzinde: Swann'ların Tarafı*, çev. Roza Hakmen, YKY, 1999, s. 9. [Bağlama uygun olarak, köşeli parantez içindeki kelime bu çeviriye sonradan eklendi. –*ç.n.*]

## 6. Günümüzde YZ

1. https://www.bbc.com/news/science-environment-47873592.
2. https://www.theverge.com/2018/12/17/18144356/ai-image-generation-fake-faces-people-nvidia-generative-adversarial-networks-gans.
3. https://www.theguardian.com/science/2018/dec/02/google-deepminds-ai-program-alphafold-predicts-3d-shapes-of-proteins.
4. https://blog.cardiogr.am/tagged/research.
5. Yalnız ortalama ömrün 50'de kalmasında, doğum sırasında veya hayatının ilk yıllarında ölenlerin oranı belirleyici bir rol oynamış olmalı. Çünkü yetişkinliğe kadar hayatta kalmayı başarmış olanların, bugün makul sayacağımız bir yaşa kadar yaşama oranı da az değildi.
6. https://www.bbc.com/news/technology-45590293.
7. J. De Fauw ve diğ., "Clinically Applicable Deep Learning for Diagnosis and Referral in Retinal Disease", *Nature Medicine*, 24, Eylül 2018, s. 1342-50.
8. https://towardsdatascience.com/why-ai-will-not-replace-radiologists-c7736f2c7d80.
9. A. Herrmann, W. Brenner ve R. Stadler, *Autonomous Driving*, Emerald, 2018.

10. https://media.ford.com/content/fordmedia/fna/us/en/news/2016/08/16/ford-targets-fully-autonomous-vehicle-for-ride-sharing-in-2021.html.
11. https://www.riotinto.com/news/releases/AHS-one-billion-tonne-milestone.

## 7. İşlerin Nasıl Zıvanadan Çıkabileceğine Dair Tahayyüller

1. https://www.independent.co.uk/life-style/facebook-artificial-intelligence-ai-chatbot-new-language-research-openai-google-a7869706.html.
2. R. Kurzweil, *The Singularity is Near*, Penguin, 2005.
3. V. Vinge, "The Coming Technological Singularity: How to Survive in the Post- Human Era", *Vision 21: Interdisciplinary Science and Engineering in the Era of Cyberspace* içinde, s. 11-22.
4. T. Walsh, "The Singularity May Never Be Near", arXiv:1602.06462v1.
5. https://www.imdb.com/title/tt0343818/.
6. D. S. Weld ve O. Etzioni, "The First Law of Robotics (A Call to Arms)", *Proceedings of the National Conference on Artificial Intelligence (AAAI-94)* içinde, 1994, s. 1042-47.
7. P. Foot, "The Problem of Abortion and the Doctrine of the Double Effect", *Oxford Review*, No. 5, 1967.
8. https://www.businessinsider.com/self-driving-cars-already-deciding-who-to-kill-2016-12.
9. E. Awad ve diğ., "The Moral Machine Experiment", *Nature*, 563, 2018, s. 59-64.
10. https://bmdv.bund.de/SharedDocs/EN/publications/report-ethics-commission.html.
11. https://futureoflife.org/2017/08/11/ai-principles/.
12. Yeri gelmişken, şeffaf olmak adına, şunu belirtmem lazım: Asilomar prensipleriyle sonuçlanan iki toplantıya da davet edildim, katılmayı da çok isterdim ama tarihleri başka işlerinkiyle çakıştığından olmadı; her zamanki gibi daha önceden verilmiş başka sözler, dolayısıyla yerine getirilecek başka yükümlülükler vardı.
13. https://www.theregister.com/2015/03/19/andrew_ng_baidu_ai/.
14. https://ai.google/responsibilities/responsible-ai-practices/.
15. https://wayback.archive-it.org/12090/20210105083335/https://ec.europa.eu/digital-single-market/en/news/draft-ethics-guidelines-trustworthy-ai.
16. https://standards.ieee.org/industry-connections/ec/autonomous-systems/.
17. V. Dignum, *Responsible Artificial Intelligence*, Springer, 2019.
18. N. Bostrom, *Superintelligence*, Oxford University Press, 2014; Türkçesi: *Süper Zekâ*, çev. Ferit Burak Aydar, Koç Üniversitesi Yayınları, 2019.
19. S. O. Hansson, "What Is Ceteris Paribus Preference?", *Journal of Philosophical Logic*, 25(3), 1996, 307-32.

20. A. Y. Ng ve S. Russell, "Algorithms for Inverse Reinforcement Learning", *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)* içinde, 2000.

## 8. İşler Gerçekten Nasıl Zıvanadan Çıkabilir?

1. C. B. Benedikt Frey ve M. A. Osborne, "The Future of Employment: How Susceptible Are Jobs to Computerisation?", *Technological Forecasting and Social Change*, 114, Ocak 2017.
2. https://rodneybrooks.com/blog/.
3. https://hbr.org/2016/11/what-artificial-intelligence-can-and-cant-do-right-now.
4. https://www.prospectmagazine.co.uk/economics-and-finance/universal-basic-income-as-unpopular-as-it-is-impractical-%E2%80%A8.
5. https://www.ft.com/content/ed6a985c-70bd-11e2-85d0-00144feab49a.
6. https://www.bbc.com/news/technology-39857645.
7. M. Oswald vd, "Algorithmic Risk Assessment Policing Models: Lessons from the Durham HART Model and 'Experimental' Proportionality", *Information & Communications Technology Law*, 27:2, 2018, s. 223-50.
8. https://www.wired.co.uk/article/gangs-matrix-violence-london-predictive-policing.
9. https://www.predpol.com.
10. https://www.equivant.com/northpointe-suite-case-manager/.
11. https://www.theatlantic.com/technology/archive/2018/01/equivant-compas-algorithm/550646/.
12. https://www.callingbullshit.org/case_studies/case_study_criminal_machine_learning.html.
13. Bildiğim kadarıyla bu senaryoyu ilk ortaya atan YZ araştırmacısı Stuart Russell.
14. https://www.forbes.com/sites/zakdoffman/2019/07/08/russias-military-plans-to-copy-jihadi-terrorists-and-arm-domestic-size-drones/?sh=1ff70a5332e7.
15. R. Arkin, "Governing Lethal Behaviour: Embedding Ethics in a Hybrid Deliberative/Reactive Robot Architecture", Teknik Rapor GIT-GVU-07-11, Bilgi İşlem Fakültesi, Georgia Teknoloji Enstitüsü.
16. https://www.stopkillerrobots.org/.
17. Birleşik Krallık Lordlar Kamarası Yapay Zekâ Komitesi 2017-19 Raporu, "AI in the UK: Ready, Willing and Able?", HL Paper 100, Nisan 2018.
18. http://www.icbl.org/en-gb/home.aspx.
19. https://medium.com/@ayesharbajwa/what-we-talk-about-when-we-talk-about-bias-a-guide-for-everyone-3af55b85dcdc.

20. https://www.cnet.com/tech/services-and-software/google-apologizes-for-algorithm-mistakenly-calling-black-people-gorillas/.
21. *Nature*, 563, 27 Kasım 2018, s. 610-11.
22. C. Criado Perez, *Invisible Women: Exposing Data Bias in a World Designed for Men*, Chatto & Windus, 2019; Türkçesi: *Görünmez Kadınlar*, çev. Mine Şengel, Epsilon, 2021.
23. https://systemx.stanford.edu/news/2018-07-18-000000/ai-can-be-sexist-and-racist-%E2%80%94-it%E2%80%99s-time-make-it-fair.
24. https://www.fastcompany.com/90320194/the-future-of-ai-is-genderless.
25. Facebook'un sizin hakkınızda tuttuğu enformasyon kaydını görebiliyorsunuz: https://www.facebook.com/adpreferences/?entry_product=ad_preferences_delegation.
26. https://www.theguardian.com/technology/2018/nov/12/deep-fakes-fake-news-truth.
27. https://www.cbsnews.com/news/doctored-nancy-pelosi-video-highlights-threat-of-deepfake-tech-2019-05-25/.
28. https://www.theguardian.com/technology/2018/jan/25/ai-face-swap-pornography-emma-watson-scarlett-johansson-taylor-swift-daisy-ridley-sophie-turner-maisie-williams.
29. https://committees.parliament.uk/committee/203/education-committee/news/102507/pepper-the-robot-appears-before-education-committee/.
30. https://www.theguardian.com/world/2018/dec/12/high-tech-robot-at-russia-forum-turns-out-to-be-man-in-robot-suit.
31. https://www.forbes.com/sites/zarastone/2017/11/07/everything-you-need-to-know-about-sophia-the-worlds-first-robot-citizen/?sh=7403e02346fa.
32. https://www.businessinsider.com/interview-ai-robot-sophia-hanson-robotics-2017-12.

## 9. Bilinçli Makineler?

1. https://www.nobelprize.org/prizes/themes/how-the-sun-shines.
2. T. Nagel, "What Is It Like to Be a Bat?", *Philosophical Review*, 83(4), 1974, s. 435-50.
3. D. Kahneman, *Thinking, Fast and Slow*, Penguin, 2012; Türkçesi: *Hızlı ve Yavaş Düşünme*, çev. Filiz Deniztekin ve Osman Çetin Filiztekin, Varlık, 2015.
4. "Sürücünün anlayışına sahip olan ve zihninin dizginlerini kontrol eden kişi, yolculuğunun sonunda, her yerde ve her şeyde olanın yüce yurduna varır" (*Katha Upanişad*, 1.3).
5. C. S. Soon ve diğ., "Unconscious Determinants of Free Decisions in the

Human Brain", *Nature Neuroscience*, 11, 2008, s. 543-45.

6. Daha doğrusu *neokorteks*; Dunbar beynin algılama, akıl yürütme ve dili idare eden bu kısmının büyüklüğüyle ilgileniyordu.
7. D. C. Dennett, *The Intentional Stance*, MIT Press, 1987.
8. D. C. Dennett, "Intentional Systems in Cognitive Ethology", *Behavioral and Brain Sciences*, 6, 1983, s. 342-90.
9. Y. Shoham, "Agent-Oriented Programming", *Artificial Intelligence*, 60 (1), 1993, s. 51-92.
10. J. McCarthy, "Ascribing Mental Qualities to Machines", *Formalizing Common Sense: Papers by John McCarthy* içinde, haz. V. Lifschitz, Ablex, 1990.
11. https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii.
12. Buradaki anahtar kelime "anlamlı". Ne zaman böyle bir test önerilse, muhakkak birileri de çıkar testin bir açığını bulur, hiç öngörülmedik bir şekilde sonuca ulaşarak testi geçtiğini iddia eder. Turing testinde de tam olarak böyle olmuştur. Özetle, alavere dalavereyle sonuca ulaşan programları kastetmiyorum, bunu esaslı bir biçimde yapan programların peşindeyim.
13. S. Baron-Cohen, A. M. Leslie ve U. Frith, "Does the Autistic Child Have a 'Theory of Mind'?", *Cognition*, 21(1), 1985, s. 37-46.
14. S. Baron-Cohen, *Mindblindness: An Essay on Autism and Theory of Mind*, MIT Press, 1995.
15. N. C. Rabinowitz ve diğ., "Machine Theory of Mind", arXiv:1802.07740.
16. Evrimsel psikoloji konusunda uzman değilim. Bu kısımda kılavuzum Robin Dunbar'ın *Human Evolution* (Penguin, 2014) kitabı oldu. Konu ilginizi çekiyorsa, daha ayrıntılı bilgi için muhakkak öneririm.

# Okuma Önerisi

MERAK UYANDIRAN kimi konularda daha ayrıntılı bilgi için, okuma önerileriyle ilgileneceğinizi tahmin ediyorum. Dolayısıyla bu bölümde, literatüre ışık tuttuğunu düşündüğüm belli başlı kaynakları sıralamak istiyorum.

YZ'ye giriş niteliğinde muazzam bir modern akademik çalışma için bkz. Stuart Russell ve Peter Norvig, *Artificial Intelligence: A Modern Approach* (Pearson, 3. Basım, 2016). Bu benim standart kaynağım; elinizdeki kitap üstünde çalışırken, teknik ayrıntıları kontrol etmek için bu kitaptan tekrar tekrar faydalandım.

YZ tarihi ve alanın nasıl geliştiği konusunda biraz daha detay isterseniz, Nils Nilsson'ın *The Quest for Artificial Intelligence*'ı (Cambridge University Press, 2010; Türkçesi: *Yapay Zekâ: Geçmişi ve Geleceği*, çev. Mehmet Doğan, Boğaziçi Üniversitesi Yayınevi, 2019) tam size göre. Alanın en iyi araştırmacılarından birinin kaleminden çıkma bu kitap, modern YZ'nin birçok kolu için mükemmel bir tarihsel kılavuz.

Alan Turing'in hayatı üstüne çok kitap var ama bunların açık ara en iyisi Turing'in hikâyesini dünyaya duyuran şu kitap: Andrew Hodges, *Alan Turing: The Enigma* (Burnett Books/Hutchinson, 1983; Türkçesi: *Enigma*, çev. Zeynep Ünalan, Olasılık, 2017). Lisans öğrencisiyken okuduğum şu üç ciltlik kitap çok hoşuma gitmişti: *Handbook of Artificial Intelligence*, William Kaufmann & Heuristech Press (Cilt I, haz. Avron Barr ve Edward A. Feigenbaum, 1981; Cilt II, haz. Barr ve Feigenbaum, 1982; Cilt III, haz. Paul R. Cohen ve Edward A. Feigenbaum, 1982). Altın Çağ'da inşa edilen sistemler ve bu sistemlerin inşasında kullanılan teknikleri çok iyi anlatıyor. Bir süredir bas-

kısı bulunmuyor, ama ekitap versiyonuna rahatlıkla ve daha ucuza ulaşabilirsiniz. Uzman sistemler dünyası için iyi bir kaynak kitap arıyorsanız, Peter Jackson'ın *Introduction to Expert Systems*'ı (Addison Wesley, 1986) bu iş için biçilmiş kaftan. Bundan otuz sene önce, ben daha lisans öğrencisiyken okutuluyordu bu metin; dolayısıyla elinizdeki kitabı yazarken bir vesileyle tekrar göz atmak çok keyifli oldu. Uzman sistemler için bilgi mühendisliği konusuna ayrıntılı bir giriş için bkz. *Building Expert Systems* (haz. Frederick Hayes-Roth, Donald A. Waterman ve Douglas B. Lenat, Addison Wesley, 1983). YZ'de mantık geleneğini daha iyi anlamak için, Michael Genesereth ve Nis Nilsson'ın *Logical Foundations of Artificial Intelligence*'ını (Morgan Kaufmann, 1987) mutlaka öneririm. Bu kitabı yüksek lisans öğrencisiyken altını çize çize okumuştum, gerçi ele aldığı teknikler on yıla kalmadan hükmünü yitirmişti. Yine de iyi yazılmış bir kitap olma özelliğini hâlâ koruyor doğrusu, YZ araştırmalarının en etkili dallarından birini ustaca anlatıyor, üstelik mantığa giriş bakımından da muazzam.

Rodney Brooks'un *Cambrian Intelligence*'ı (MIT Press, 1999) davranışsal YZ ve kapsama mimarisi üstüne belli başlı çalışmalarını bir araya getiriyor. Biraz ukalalık gibi olacak ama, ajanlar, ajan mimarileri ve çok ajanlı sistemlere giriş olarak kendi kitabımı da öneririm: *Introduction to MultiAgent Systems* (Wiley, 2. Basım, 2009). YZ'de oyun teorisinin rolünü daha iyi anlamak istiyorsanız, Yoav Shoham ve Kevin Leyton-Brown'ın *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*'ına (Cambridge University Press, 2008) mutlaka göz atın.

Makine öğrenmesi uçsuz bucaksız bir konu. Elinizdeki kitapta bu konuya ancak şöyle bir değinebildim. Günümüzün başlıca ilgi alanı olarak nöral ağlar üstünde biraz durdum belki ama makine öğrenmesi için daha başka pek çok teknik var. Biraz daha ayrıntılı olmakla beraber yine de aşırı teknik olmayan bir giriş için bkz. Ethem Alpaydın, *Machine Learning: The New AI* (MIT Press, Essential Knowledge Series, 2016; Türkçesi: *Yapay Öğrenme*, çev. Aylin Ağar, Tellekt, 2020). DeepMind'da iş kovalıyorsanız (bugünlerde hangimiz kovalamıyoruz ki), aradığınız kitap Ian Goodfellow, Yoshua Bengio ve Aaron Courville'in *Deep Learning*'i (MIT Press, 2017; Türkçesi: *Derin Öğrenme*, çev. F. Vural, R. Cinbiş, S. Kalkan, Buzdağı, 2021). Son olarak, Rus-

sell ve Norvig'in ikinci paragrafta bahsettiğim kitabı makine öğrenmesine çeşitli yaklaşımlar konusunda da derli toplu muazzam bir kaynak.

Tekillik üstüne iyi bir tartışma arıyorsanız, Murray Shanahan'ın çok rahat okunan kitabı *The Technological Singularity*'yi (MIT Press, Essential Knowledge Series, 2015); etik YZ için Virginia Dignum'ın *Responsible Artificial Intelligence*'ını (Springer, 2019); teknoloji ve istihdam için Carl Benedikt Frey'in *The Technology Trap*'ini (Princeton University Press, 2019) mutlaka öneririm.

Maggie Boden'ın *Artificial Intelligence and Natural Man*'i (MIT Press, 1977) güçlü YZ ve bilinçli makineler için klasik bir giriş kaynağı. Bilinç konusunda biraz daha ayrıntılı, ders kitabı gibi bir giriş metni için, Susan Blackmore'un *Consciousness: An Introduction*'ı (Hodder & Stoughton, 2010; Türkçesi: *Bilinç: Çok Kısa Bir Başlangıç*, çev. Oğuz Akçelik, İstanbul Kültür Üniversitesi, 2019) var. Daniel Dennett'ın zihin, bilinç ve YZ üstüne yazdığı hemen her şey gayet okunası; başlangıç için *Kinds of Minds*'ı (Basic Books, 1996; Türkçesi: *Aklın Türleri*, çev. Handan Balkara, Varlık, 1999) öneririm. Ve elinizdeki kitabın devamında tek bir şey okuyacak olursanız eğer, Dennett'ın "Where Am I?" başlıklı makalesini öneririm. 1978 tarihli *Brainstorms*'unda yayımlanan bu metne çevrimiçi olarak ulaşabilirsiniz.

# Dizin

—

metis bilim | 51

Bilgisayar teknolojilerinin hızla geliştiği şu günlerde "yapay zekâ" terimini giderek daha sık duyuyoruz; bilim, sağlık, eğitim, sanayi, eğlence, sanat ve daha nice alanda yapay zekâ uygulamaları gün geçtikçe yaygınlaşıyor. Peki ama yapay zekâ (YZ) derken tam olarak ne kastediyoruz?

Otuz yıldan uzun süredir YZ araştırmacısı olarak çalışan Michael Wooldridge, bu kitapta bize YZ'nin ne olduğunu ve –belki daha da önemlisi– *ne olmadığını* açıklıyor. Gerek medyada çıkan sansasyonel haberlerin gerekse yakın gelecekte "bilinçli makinelerin" aramızda olacağını ima eden araştırmacıların aşırı iyimserliğinin yanıltıcı bir tablo çizdiğini vurgulayan Wooldridge, YZ araştırmacılarının *gerçekte* ne üstünde ve nasıl çalıştığını anlatıyor.

"Makine öğrenmesi" ve "derin öğrenme" nedir? YZ konusunda filmlerde, edebiyatta ve medyada karşılaştığımız "Terminatör" senaryolarının gerçekleşmesi mümkün mü? YZ'de işler nasıl zıvanadan çıkabilir? YZ konusunda gerçekten endişelenmemiz gereken meseleler ve endişelenmemize çok uzun bir zaman hiç gerek olmayan meseleler neler? YZ işimizi elimizden alacak mı ve çalışmanın doğasını nasıl etkileyecek? YZ teknolojilerini kullanmanın insan hakları üstünde nasıl bir etkisi olabilir? YZ sistemlerinin ahlaki fail olarak edimde bulunması mümkün mü? Ve elbette: Makineler düşünebilir ve arzulayabilir mi?

Yapay zekânın geçmişine, mevcut durumuna ve nereye gittiğine dair nesnel bir değerlendirme sunan bu kitabı, konuya ilgi duyan okurlarımıza hararetle tavsiye ediyoruz.

Metis Bilim
ISBN-13: 978-605-316-268-1



Metis Yayınları
www.metiskitap.com