BİLGİSAYAR: "EUREKA, EUREKA"

HASAN DAVULCU*

Burada teorem ispatlayabilen bilgisayar programlarının incelenmesine bir giriş yapmak istiyoruz. Bu tip programlara "teorem ispatlayıcı" diyoruz.

Yüksek hızlarda çalışan dijital bilgisayarların gelişmesine kadar ilgi görmeyen otomatik teorem ispatlama 1930'ların başlarında Herbrand'ın çalışmalarıyla doğmuştur. Daha sonraki çalışmalar 1950'lerin sonlarında Newell, Simon, Shaw, Gelernter tarafından keşfe yarayan taktikler (heuristic) üzerine kurulmuştur. 1960'larda çalışmaların ağırlığı sözdizim kaidelerine ait farklı metodlar üzerine kaymış ve 'resolution' tipi sistemler üzerinde yoğunlaşmıştır. 1970'lerde yeniden taktiksel yaklaşımlar benimsenmiş ve insan tarafından sağlanan, konuya bağlı bilgilerin makineye yüklenmesi üzerinde durulmuştur. Buradaki resolution kelimesi konuya bağlı çok az bilgi ve sadece birkaç tane sentaks'la (dizin) ilgili taktik kullanan çok yönlü kullanım alanlarına sahip teorem ispatlayıcıları ile eşleştirilmiştir.

Birçok yönüyle teorem ispatlama ve problem çözme birbirlerine benzemektedir. Teorem ispatlama yapay zeka (YZ) çalışmaları içerisinde de tarihsel bir geçmişe sahiptir. YZ alanındaki ilk çalışmalarda teorem ispatlama için kullanılabilecek taktikler (heuristics) araştırılmıştır.

YZ araştırmalarının ilk kilometre taşlarından bir tanesi, Newell, Simon, ve Shaw'un Mantık Teorisi (The Logic Theorist)'dir.

Mantık Teorisi, RAND Corporation Johnniac bilgisayarlarında, Carnegie-Mellon Üniversitesinden Newell ve Simon'la birlikte çalışan programcı Cliff Shaw tarafından hazırlanmıştır. İlk liste işlemcilerinden bir tanesi olan, IPL-I (Information Processing Language I) programlama dilinde yazılmıştı. Problem konusunu mantık önermeleriyle sınırlayan Newell ve Simon çok basit yazılım ve donanım kullanarak başarılı bir program hazırlamışlardı. Newell ve Simon kendilerine, Russel ve Whitehead'in Principa Mathematica adlı kitaplarındaki teorem ve önermelerin ispatlarını hedef almışlardı.

Aşağıdaki 5 aksiyomla

1. $(PVP) \Longrightarrow P$
2. $P \Longrightarrow (QVP)$
3. $(PVQ) \Longrightarrow (QVP)$
4. $[PV(QVR)] \Longrightarrow [QV(PVR)]$
5. $(P \Longrightarrow Q) \Longrightarrow [(RVP) \Longrightarrow (RVQ)]$

başlayan Mantık Teorisi

$$\begin{aligned}
(P \Longrightarrow \sim Q) &\Longrightarrow (\sim P \Longrightarrow Q) \\
\sim (PVQ) &\Longrightarrow \sim P \\
[HV(QVR)] &\Longrightarrow [(PVQ)VR] \\
[(\sim Q \Longrightarrow P) &\Longrightarrow (P \Longrightarrow Q)]
\end{aligned}$$

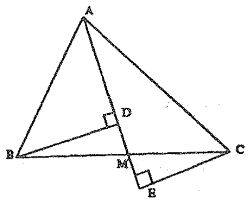
önermelerini ispatlamayı başarmıştı. İspatladığı teoremleri aksiyomlarıyla birleştiren Mantık Teorisi Principia Mathematica'nın 2. bölümündeki ilk 52 teoremin 38'ini ispatlayabilmişti. Hatta, Mantık Teorisi bazı teoremleri ilk ispatlarından daha zarif şekilde ispatlamıştı.

Zamanla, otomatik teorem ispatlayıcılar mantık, düzlem geometrisi, grup teorisi, küme teorisi, türev, integral, analiz ve topolojiyle ilgili teoremleri ispatlamışlar ve program doğrulama, program sentezleme, konuşma dillerini anlama, robot planlama ve veri tabanı oluşturma ve düzeltme gibi uygulama dallarında kullanılmışlardır.

Bir diğer kilometre taşı ise problem çözme formatıyla oluşturulan ve lise geometrisinde beklenmedik başarı gösteren Gelertner'in Geometri Makinasıdır. Aşağıda Gelertner'in makinesinin ispatlayabileceği tipik bir geometri teoreminin ispatını bulacaksınız. Sunuluşunu basitleştirmek amacıyla makinenin ispatındaki bazı adımları vermiyoruz.

TEOREM: Bir üçgenin iki köşesi, bu köşeleri birleştiren kenarın kenar ortayına eşit uzaklıktadırlar. (Şekil 1)

* Kıbrıslı bir okurumuz.



Şekil 1.

VERİLER: $\overline{BM} = \overline{MC}$, $\overline{BD}, \overline{AM} \perp \overline{CE} \perp \overline{ME}$.

HEDEF: $\overline{BD} = \overline{EC}$.

ÇÖZÜM	SEBEP
$\sphericalangle DMB = \sphericalangle EMC$	İç ters açılar
$\sphericalangle BDM = \sphericalangle CEM$	dik açılar
$\overline{BM} = \overline{MC}$	veri
CEM bir üçgendir	diagram üzerine yapılmış kabul
$\Delta CEM \cong \Delta BDM$	kenar-açı-açı
$\overline{BD} = \overline{EC}$	eş üçgenlerin benzeşen kenarları

Makine geriye doğru uslamlamayla şöyle ilerler. Hedefi:

G1 $\overline{BD} = \overline{EC}$ olarak belirler.

Dolayısıyla bu çeşit bir hedefin yakalanabilmesi için oluşturulmuş listesine bakar ve diğerlerinin arasından, iki doğru parçasının eşit olduğunu göstermek için bunların eş üçgenlerin mütabık kenarları olduğunu göstermesinin yeterli olduğunu görür. $BD, \Delta BDM$'nin EC de ΔCEM'nin içersinde olduğu için yardımcı hedefi,

G2 $\Delta CEM \cong \Delta BDM$

seçer ve iki üçgenin eş olduğunu göstermesi için gerekenlerin sıralandığı listeye başvurduğunda;

a) Kenar-kenar-kenar, b) Kenar-açı-kenar, c) Kenar-açı-açı, yardımcı hedefini

G3 CEM ve BCM üçgenleri için "kenar-kenar-kenar" olarak tanımlar. Epey uzun bir çabadan sonra başaramayınca hedef olarak

G4 Kenar-açı-açı seçer. Bunu kanıtlayabilmenin pek çok yolu vardır. Bunlardan bir tanesi üç yardımcı hedef içeren:

G6 $\overline{BM} = \overline{MC}$

G7 $\sphericalangle DMB = \sphericalangle EMC$

G8 $\sphericalangle BDM = \sphericalangle CEM$'dir.

Makine G 6'yı verilerin içerisinde bulur. G7 ve G8'i çözmek için, iki açının eşit olma durumlarını içeren listesine başvurur ve orada, diğerlerinin arasında. "İç ters açılar eşittir" ve "tüm dik açılar eşittir" ilkelerini bulur. $\sphericalangle DMB$ ve $\sphericalangle EMC$'nin iç ters açılar ve $\sphericalangle BDM$ ve $\sphericalangle CEM$'nin dik açılar olduğunu tespit ettiği için yardımcı hedefleri G6-G8'i elde etmiş oluyor ve dolayısıyla G5, G2 ve G1. İspatın ikinci adımında makine aşağıdaki yardımcı hedeflerden herhangi birini seçebilirdi:

G2.1 $\Delta CEA \cong \Delta BDA$

G2.2 $\Delta CEA \cong \Delta BDA$

G2.3 $\Delta CEM \cong \Delta BDA$

G2 $\Delta CEM \cong \Delta BDM$

Fakat, hafızasında durumun "genel" bir şeklini oluşturduğu için (Şekil 1'in temsili bir gösterimi) makine, "ölçümlerle", yardımcı hedefler G2.1-G2.3 doğru olmadığı tespit ediyor ve böylece çalışma zamanı da azımsanmayacak ölçüde azalmaktadır.

Yukarıdaki, yanlış yardımcı hedeflerin filtrelenmesi genelleştirilerek, otomatik teorem ispatlama birçok alanlarında kullanılmaktadır. Örnek olarak, Grup Teorisinde yanlış bir yardımcı hedef, bilinen gruplar üzerinde (örneğin Klein 4 grup) yapılan testler sonucunda elenebilmektedir.

Bilgisayar matematiğinde bir diğer ilgi çekici gelişme ise Lenat'ın son zamanlarda matematiksel keşif konusundaki çalışmalarıdır. Lenat elementer kümeler üzerine bir kısım gerçeklikler ve bazı elementer keşfe yarayan taktikler kullanarak Sayılar Teorisinde kavramlar yakalamaya çalışan bir program hazırlamaktadır. Belli sonuçlar almıştır bile: Program, sayı kavramı, çarpma, bölme ve asal sayı gibi kavramları keşfetmiş ve herhangi bir sayının asal çarpanlara ayrılışının tekliğini farketmiştir.

Yukardaki girişimi, otomatik teorem ispatlama tekniği ile birleştirmek için planlar yapılmaktadır. Bu şekilde ispatlayıcı, keşfedici sistemin tahminlerini ayıracak ve kendisi için ele alabileceği önermeler oluşturacaktır.

Bilgisayar tarafından çözülmüş, uzun zamandır çözülemeyen bir problem "10. dereceden bir projektif düzlem var mı?" sorusudur.

Bir projektif düzlem 'nokta'lardan oluşur. Bu noktaların belli bazı kümeleri 'doğru'ları oluşturur. N. dereceden bir projektif düzleminin tam olark $N^2 + N + 1$ nokta ve bir o kadar da doğrusu vardır. Bu nokta ve doğruların aşağıdaki aksiyomları sağlaması gerekmektedir:

1- Her doğruda N+1 nokta vardır.

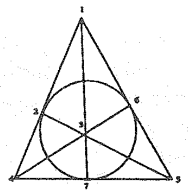
2- Her nokta N+1 doğru üzerinde bulunur.

3- Herhangi iki doğru sadece bir noktada çakışabilir.

4- Herhangi iki noktadan tek bir doğru geçer.

Şekil 2 ikinci dereceden bir projektif

düzlemi göstermektedir.



Şekil 2

Şekilde 6 7 noktalarından geçen 'doğru' bir çemberle gösterilmiştir. Bu düzlem şekil 3 teki, bilgisayarların da kullanabileceği bir matrisle ifade edilebilir. Bu matrisin anlamı şöyle: matrisini sütunları projektif düzlemin noktalarını, satırları ise doğrularını gösteriyor. Matrisin j inci elemanının 1 olması ise i inci doğrunun ij inci noktadan geçtiğini belirtiyor. Örneğin şekil 3 teki ilk satır 1, 2, 4 noktalarından geçen doğruyu veriyor.

10 . dereceden bir pojektif düzlem varsa bunun bir matris gösterimi olacaktır. Bir süper bilgisayar böyle bir özel matris araştırması yaptı. Tüm durumları tek tek inceledi ve böyle bir matris bulamadı. Böylece 10. dereceden bir pojektif düzlem olamayacağını ispatlamış oldu.

1	1	0	1	0	0	0
0	1	1	0	1	0	0
0	0	1	1	0	1	0
0	0	0	1	1	0	1
1	0	0	0	1	1	0
0	1	0	0	0	1	1
1	0	1	0	0	0	1

Şekil 3

Dört-renk probleminin çözümlerine ve 10. dereceden projektif düzlemlerin olamayacağına inandığımızda [1]'deki çevirimizde tanımladığımız anlamda bilgisayar destekli ispatlara inanıyoruz demektir. Matematikçilerin yıllar yılı her iki probleme de yaptıkları çalışmalardan bir sonuç alamamaları, geleneksel metodların bir ispat için yetersiz olduğunu sergilemiştir. Uğraşıldıkça, bir ispat elde edebilmek için bakılması gereken durumlar sayısının kağıt üzerinde gösterilebilecek olanlardan çok daha yüksek olduğu ortaya çıktı.

Gelin bilgisayarın projektif düzlem probleminin çözümündeki rolünü inceleyelim. Projektif düzlem sabit sayıda nokta ve doğrulardan oluşan soyut bir geometrik nesnedir. (yukarıya bakınız!) En basit şekliyle, herhangi bir üçgende 3 nokta, 3 doğruyla birleştirilmiştir. Bu düzlemin derecesi 1'dir. Her projektif düzlemin kendi geometrisi vardır. Örnek olarak, ikinci basit düzlem (derecesi 2), 7 noktanın 7 doğru ile birleştirilmesinden oluşur. Fakat bu düzlem bile acaiptir. Düzlemi geleneksel Euclid geometrisinde bir parça kağıt üzerine çizmeye çalışırsanız göreceksiniz ki doğrulardan biri çember olmuştur. (Şekil 2). Her projektif düzlem, 0 ve 1 rakamlarından oluşan özel bir matris (incidence matrix) ile tanımlanabilir. Bu matris bazı kolay kontrol edilebilen şartları sağlar. Örnek olarak, 10. dereceden projektif düzlem için, matrisin 111 sütun ve 111 satıdan oluşması ve her satırda ve her sütunda 11 bir ve 100 sıfır olması gerekir. Ve-burası önemli nokta-eğer yukardaki tüm şartları sağlayan böyle bir matris yoksa, 10. dereceden bir pojektif düzlem de yoktur.

Böylece, projektif düzlem için olan araştırmamız onun matrisinin araştırmasına indirgenmiştir. Fakat, bir süperbilgisayar için bile 0 ve 1 rakamlarından oluşan tüm 111×111 matrislerinin taranması imkansız bir iştir.

Montreal'daki Concordia Universitesi'nden Lam, Thiel ve Swiercz, teorik argümanlarla farklı durumların sayısını bilgisayarın inceleyebileceği bir sayıya düşürdüler ve son araştırma için gereken programı yazdılar. Bilgisayar 'Backtrack search' diye isimlendirilen bir algoritmayla bazı kısımları boş olan bir matrisi, bizim matrisimizin şartlarını sağlayacak şekilde doldurmaya çalıştı. Eğer program bir tek durumda bile başarsaydı, 10. dereceden bir projektif düzlem bulmuş olacaktı. Fakat, Cray-1A süperbilgisayarları araştırmasını hiçbir 'incidence' matris bulamadan tamamladı. Ondan sonra üç araştırmacı 10. dereceden bir projektif düzlemin olmadığı kanısına vardılar.

Böyle bir ispatla karşılaştığımızda ilk akla gelen soru sanıyoruz şu olacaktır: Herhangi bir donanım veya yazılım hatası olmadığını nereden anlayabiliriz?

Olası hataların pek çok kaynağı vardır: Yanlış girdiler, yazılım hataları, derleyicideki hatalar ve donanım hataları (örneğin bilgisayarın hafızasındaki bitlerin rastgele değişmesi). Eğer olabilirse, bazı hataların olduğunu kabul etmeliyiz, Lam burada: 'Bu bir deneysel sonuçtur,' yazıyor, her zaman hata olasılığı vardır' Fakat inandırıcı edici bir biçimde devam ediyor ve ekliyor '10. dereceden bir pojektif düzlemin varlık olasılığı çok küçüktür."

Bu ispatın doğruluğunun gösterilmesinin sebepleri açık olmalıdır. Belki tüm bilgisayar yazılımlarını kontrol edebiliriz (derleyiciler ve işletim sistemleri de dahil) Fakat program çalışırken rastgele hataların ve donanım hatalarının yokluğunu nasıl garanti edebiliriz?

Hataların yokluğunu ispatlayamasak bile, sonucun doğruluğunu dolaylı yoldan sağlayabiliriz. Örnek olarak, bağımsız bir şekilde, farklı yazılım ve donanımlar kullanarak (ki bu gerçekte dört-renk problemi için Kanada'daki Manitoba Üniversitesinden Allaire tarafından yapılmıştır). Başka bir test ise şudur. Basit ama mantıgın temel prensiplerinden birisi der ki: yanlış bir önermeden yola çıkıp her şeyin ispatlanması olasıdır. Eğer dört-renk teoremi yanlışsa, iki doğru olduğuna inanıyoruz, biz onu kullanarak diğer teoremleri ispatlayabiliriz. Ve bunlardan bir tanesi matematiksel olarak ispatlanmış bir doğrulukla çelişkiye düşebilir. Şimdiye kadar böyle bir çelişkinin yakalananmaması, bilgisayarın sonucuna denk düşmektedir.

Tabi ki, bilgisayarların kullanılması belli bir belirsizliği getirmektedir. Lam'ın dediği gibi: 'Fizikçiler bazı belirsizliklerle yaşamayı öğrendiler, biz matematikçilerin de kesin doğrulanamayan bazı ıspatlarla yaşamayı öğrenmemiz gerekiyor'.

Burada en genel soru, ispatın uzunluğudur. Bazı matematiksel önermelerin en kısa ispatları bile bir insanın kontrol etmesi için çok uzun olabilir. Gerçekten bu kategoriye girebilecek ilginç veya önemli bir teorem var mı bilmiyoruz. Dört-renk teoreminde olduğu gibi bazı teoremlerin kağıt üzerinde ispatı olmayabilir. Appel ve Haken bu görüşü paylaşmaktalar: 'Dört-renk teoremi için yaptığımız ispat, sadece teorik methodlarla teorik matematikte yapılabileceklerin sınırlı olduğunu göstermektedir'.

[1]'deki Hardy'nin benzetmesini biraz ilerletecek olursak, bilgisayar destekli bir ispat, uzak bir gezegendeki bir zirvenin fotoğrafının bir uzay aracıyla dünyaya geçilmesi olabilir. Bizim gözlemcimiz bunu güvenilmez bir elektronik görüntü, ikinci el bir kanıt olarak reddedebilir. Fakat farketmelidir ki zirve çıplak gözle görünmediği halde gerçekte varolabilir.

Matematikçiler bir bilgisayar destekli ispatla karşılaştıklarında aynı duruma düşmektedirler. İspatın kabulünü, bir diğer matematikçi kendilerinin de kontrol edebildiği daha kısa bir ispatla gelene kadar erteleyebilirler. Fakat böyle bir ispat imkansız olabilir. Böylece, eğer 'dolaylı' bir kanıtı tanımazlarsa, sadece geleneksel olmayan yöntemlerle erişilebilecek matematiksel doğruluklara set çekme riskini göze alırlar.

Manin'e göre, ispat kavramı içersinde gizli bir zaman kavramı taşımaktadır. Diyorki: 'Bir "ispat" günümüzün matematiksel doğruları için bir ispattır. Eğer örnek olarak küme teorisine yeni bir aksiyom ekleyecek olursak ve bu kabul görürse o zaman bizim ispat anlayışımız daha da gelişecektir.'

Bilgisayarların kullanılması da, zaman içersinde, kabul gören bu matematiksel değerlerden birisi olabilir. Hatta Amerikalı matematikçi John Miller'in yarı şaka kehanet ettiği gibi; iki nesil sonra herhangi bir şey bilgisayarla kontrol edilmedikçe, ispatlanmış sayılmayacatır.

KAYNAKLAR

[1] Bilgisayarlar ispat peşinde, Matematik Dünyası, Cilt 1, Sayı 5.

[2] Dört Renk Problemi için bkz. Matematik Dünyası Cilt: 1, Sayı:1.

[3] Aruro Sangalli, 'The Burden of Proof is on Computer', New Scients, Science 23 February, 1991.

[4] Morris W. Firebaugh, Artifical Intelligence, PWS-KENT Publishing Company, 1988, Boston.

[5] W. W. Bledsoe, Non-Resolution Theorem Proving. Artificial Intelligence Magazine. Vol.9, 1977.

[6] The Computer Science and Engineering Research Study (COSERS), edited by Bruce W. Adren, 1980, The MIT Press.

[7] Philp C. Jackson, Jr., Introduction to Artificial Intelligince, 1.ed., New York, 1974.