

CEP ÜNİVERSİTESİ

Yapay Zekâ

JEAN-PAUL HATON
MARIE-CHRISTINE HATON

İletişim Yayınları • PRESSES UNIVERSITAIRES DE FRANCE

CEP ÜNİVERSİTESİ

Yapay Zekâ

L'intelligence artificielle

JEAN-PAUL HATON / MARIE-CHRISTINE HATON

Çevirenler

AYŞEN EKMEKÇİ-ALEV TÜRKER

İletişim Yayınları • *PRESSES UNIVERSITAIRES DE FRANCE*

İletişim Yayınları • PRESSES UNIVERSITAIRES DE FRANCE

CEP ÜNİVERSİTESİ

İletişim Yayıncılık A.Ş. Adına Sahibi: Murat Belge
Genel Yayın Yönetmeni: Fahri Aral
Yayın Yönetmeni: Erkan Kayılı
Yayın Danışmanı: Ahmet İnsel

Yayın Kurulu:
Fahri Aral, Murat Belge, Tanıl Bora, Murat Güftekingil, Ahmet İnsel, Erkan Kayılı, Ümit Kıvanç, Tuğrul Paşaoğlu, Mete Tunçay.

Görsel Tasarım: Ümit Kıvanç
Dizgi ve Sayfa Düzeni: Hüsnü Abbas - İsmail Abbas
Baskı: Şefik Matbaası (iç) / Seda Matbaası (kapak)
Dağıtım: Hür Basın Dağıtım A.Ş.

İletişim Yayıncılık A.Ş. · Cep Üniversitesi 25 · ISBN 975-470-112-1

1. Basım · İletişim Yayınları, Nisan 1991.

Nisan 1989 tarihli 1. baskısından çevrilmiştir.

© Que sais-je?, Presses Universitaires de France, 1989
108, Boulevard Saint-Germain, 75006, Paris-France
© İletişim Yayıncılık A.Ş., 1991
Klodfarer Cad. İletişim Han. No:7 34400
Cağaloğlu-İSTANBUL. Tel: 516 22 60 · 61 · 62

Önsöz

Günümüzde bilgi bir yandan en önemli değer haline gelirken diğer yandan da artan bir hızla gelişiyor, çeşitleniyor. Ama katlanarak büyüyen bilgi üretiminden yararlanmak, özellikle gündelik yaşam kaygılarının baskısı altında, zorlaşıyor. Her şeye rağmen bilgiye ulaşma çabasını sürdürenler için de imkânlar pek fazla değil.

Ayrıca, özellikle Türkiye gibi ülkelerde bir konuda kendini geliştirmek ya da sırf merakını gidermek için herhangi bir konuyu öğrenmek isteyenlerin şansı çok az. Üniversitelerimiz, toplumumuzun yetişkin bölümüne katkıda bulunmak için gerekli imkânlardan yoksun.

Cep Üniversitesi kitapları işte bu olumsuz ortamda, evlerinde kendilerini yetiştirmek, otobüste, vapurda, trende harcanan zamandan kendileri için yararlanmak isteyenlere sunulmak üzere hazırlandı.

20. yüzyıl Fransız kültür hayatının en önemli ürünlerinden olan, bugün yaklaşık 3000 kitaplık dev bir dizi oluşturan "Que sais-je" (Ne Biliyorum?) dizisini İletişim Yayınları Türkçe'ye kazandırıyor. İletişim'in Cep Üniversitesi, bu büyük diziden seçilmiş,

Türkiyeli okurlar için özellikle ilgi çekici olabilecek eserlerin yanısıra, Avrupa'nın başka yayınevlerinin benzer bir çerçevede yayımladığı kitapları da içeriyor.

Ayrıca, Türkiye'nin siyaset, kültür, ekonomi hayatıyla ilgili konularda özel olarak bu dizi için yazılmış telif eserler "üniversite"nin "öğrenim programını" tamamlayacak.

Cep Üniversitesi'nin her kitabı alanının öndegelen bir uzmanı tarafından yazıldı. Kitaplar, hem konuya ilk kez eğilen kişilere hem de bilgisini derinleştirmek isteyenlere seslenebilecek bir kapsam ve derinlikte. Bilginin yeterli ve anlaşılır olması, temel kıstas. Cep Üniversitesi kitaplarını lise ve üniversite öğrencileri yardımcı ders kitabı olarak kullanabilecek; öğretmenler, öğretim üyeleri ve araştırmacılar bu kitaplardan kaynak olarak yararlanabilecek; gazeteciler yoğun iş temposu içinde çabuk bilgilenme ihtiyaçlarını Cep Üniversitesi'nden karşılayabilecek; çalıştığı meslek dalında bilgisini geliştirmek isteyen, evinde, kendi programlayabileceği bir mesleki eğitim imkânına kavuşacak; ayrıca, herhangi bir nedenle herhangi bir konuyu merak eden herkes, kolay okunur, kolay taşınır, ucuz bir kaynağı Cep Üniversitesi'nden temin edebilecek.

Cep Üniversitesi kitapları sık aralıklarla yayımlandıkça, benzersiz bir genel kültür kitaplığı oluşturacak. İnsan Hakları'ndan Genetik'e, Kanser'den Ortak Pazar'a, Alkolizm'den Kapitalizm'e, İstatistik'den Cinsellik'e kadar uzanan geniş bir bilgi alanında hem zahmetsiz hem verimli bir gezinti için ideal "mekân", Cep Üniversitesi.

İletişim
Yayınları

İçindekiler

I. BÖLÜM

Yapay Zekâya Giriş 7
- Tanım 7
- Kısa Tarihçe 8
- YZ'nin Temel Özellikleri 10
- YZ'nin Alanları 12

II. BÖLÜM

Problemlerin Çözümlenmesi 16
- Yapay Zekâ'da Problemlerin Çözümlenmesinin Yeri 16
- Problem Durumları Aralığı (Espace) 22
- Oyun Ağaçlarının Özel Durumu 38

III. BÖLÜM

Bilgilerin Tanıtımı ve Usavurma 41
- Giriş 41
- Mantıksal Tanıtımlar 42
- Anlambilimsel Şebekeler 52
- Prodüksiyon Kuralları 54
- Yöntemsel Tanıtımlar 63
- Biçimlendirilmiş Nesnelerle Tanıtımlar 64

IV. BÖLÜM

Uzman Sistemler 70
- Giriş 70
- Uzman Sistem Nedir? 71
- Uzman Sistemlerin Uygulanması 73
- Bir Uzman Sistemin Gelişimi 78
- Sonuç 82

V. BÖLÜM
Anadilin İşlenmesi 84
Giriş 84
Yazılı Anadil 85
Konuşmanın İşlenmesi 90
VI. BÖLÜM
Robotbilim 98
Giriş 98
Planların Üretilmesi 100
Görme 105
VII. BÖLÜM
Bilanço ve Perspektifler 113
Giriş 113
Logiciel Görünümler 113
Malzeme Görünümleri 116
Gelişmiş Temalar 118
BİBLİYOGRAFYA 125

BİRİNCİ BÖLÜM
YAPAY ZEKÂYA GİRİŞ

I. Tanım

Zaten oldukça tartışmalı olan Yapay Zekâ (YZ) terimi birçok karışıklığa ve yanlış yorumlara yolaçmaktadır. Zekâ kavramının kendisinin karmaşık ve göreceli olması nedeniyle YZ'nin bilimsel yönteminin ne olduğunu tanımlamak kolay değildir.

YZ birbirini bütünleyen iki bakış açısına göre ele alınabilir:

- Biri, bilgisayarın bir model veya teoriyi test etmek için oluşturum aracı olarak kullanıldığı, zekânın çalışma biçimlerinin incelenmesini kapsar; bu bakış açısı tanımaya dair bir tür yöntem ortaya koyar;
- Daha pragmatik olan diğeri, bilgisayarı, genellikle insan zekâsına özgü kapasitelerle donatmak üzere sarfedilen çabalarla ilgilidir; bilgilerin edinilmesi, algılama (görme, işitme), düşünme, karar verme vs. Daha sıkça karşılaşılan, bu ikinci bakış açısıdır. Bu bakış bir bilgisayar programı aracılığı ile zekice tutumları yarıştırmaktan oluşur, bununla birlikte insanın sözkonusu işleyişinin aynısını yansıtmaz.

İnsana özgü mekanizmalar hakkında daha iyi bir bilgi, enformatik sistemlerin performanslarını arttırmaya elverdiği ölçüde sözkonusu iki yaklaşım oldukça birbirini tamamlayıcı özelliktedir. YZ, tekniklerini kullandığı enformatiğin ileri bir dalı olarak ortaya çıkar; buna karşılık enformatik yavaş

yavaş YZ'nin bilgilerini kendine maleder. Bu YZ'nin çok disiplinli olma özelliğinden bir şey götürmez; YZ bu kitabın tümünde de görüleceği gibi birçok bilim ve tekniğin kesişme noktasıdır.

YZ '80'li yıllarda, bu alanda uzmanlaşmış kuruluşların ve geniş kapsamlı araştırma-geliştirme projelerinin yürüttüğü uygulamalar sayesinde ekonomik planda olduğu kadar, dünyadaki binlerce araştırmacı ve öğretim görevlisi aracılığı ile akademik planda da bir gerçeklik haline gelmiştir.

II. Kısa Tarihçe

1. Doğuş - YZ Birleşik Devletler Dartmouth'da düzenlenen ve bir grup yüksek düzeyli bilim adamını biraraya getiren bir konferansın tarihi olan 1956'da doğmuştur. Bu konferansta J. McCarthy, M. Minsky, C. Shannon, A. Newell ve H. Simon zekâ ile donatılmış bilgisayar programlarını gerçekleştirme olasılığını araştırmayı önermişlerdir. Bu vesile ile uydurulan *Artificial Intelligence* terimi kusurlarına ve ta başından beri Newell ve Simon gibi bazı araştırmacıların karşı koymalarına rağmen süregelmiştir.

İlk YZ programları (bir Mantık teoremleri uygulayıcısı, *Logic Theorist* ve bir satranç oyunu programı; her ikisi de Newell ve Simon'ındır) ve LISP (YZ'nin kılavuz dili) de bu dönemde ortaya çıkmıştır (bak. Yedinci Bölüm)

Aslında bu "resmî" ortaya çıkıştan daha öncelere, ta Antikite'ye dek uzatılabilen bir hazırlık dönemi vardır, çünkü insanın "zeki" makinelere ilgisi çok eskiye dayanır. Bilimsel bir planda ise bir yandan otomatlar üzerindeki teorik ve pratik gelişmeler, diğer yandan matematiksel mantık çalışmaları YZ'nin ilk temellerini oluşturmuştur.

İlk olarak Babbage (ve "analitik" makinesi, 1842), Turing (ve evrensel makinesi, 1936) ve beynin edinilen bilgileri işlemesinde karşı tepki olaylarının önemini ortaya çıkaran sibernetikten (1943'e doğru, özellikle Wiener ile doğan) bahsedilebilir. İkinci olarak 1930'a doğru, çözümün algoritmik olmadığı (yani çözümün belli bir sıra ile gerçekleşen sınırlı bir işlemler dizisi şeklinde ifade edilmediği) problemlerin varlığı gerçeğinin kanıtlanması ile sonuçlanan çözümsüzlük ve eksik kalma konusundaki önemli çalışmalar Leibniz, Boole, Hilbert, Gödel ve Church'e aittir.

Gerçek YZ'nin doğuşuna yolaçan katalizör 1943'e doğru bilgisayarların ortaya çıkmasıdır. Bu dönemden itibaren bazı öncüler bu makineleri biraz zekâ ile donatma meselesini ele aldılar. Böylelikle 1950'de matematikçi A. Turing bir makinenin zeki olup olmadığına karar verme olanağı tanıyan bir test ortaya koydu. Temel kuralı tartışmalı olmakla birlikte bu test, bu dönemden itibaren bilgisayarların "zekâ"sına verilen önemi gayet iyi göstermektedir.

2. Gelişmenin Önemli Evreleri - YZ'nin tarihi çok şematik olarak dört evreye bölünebilir:

- Karşılaşılan sorunlardaki büyük güçlükleri henüz tam olarak tartamamış olan araştırmacıların abartılı bir iyimserliği ile geçen '50'li yıllar YZ'nin *doğuş*una tanık olmuştur.

- Sonraki on yıl çok sayıdaki proje ve sağlanan bazı önemli sonuçlarla disiplinde gerçek bir *hareketliliğe* tanık olmuştur: "heuristique" araştırma algoritmaları ve hatta GPS, genel problem çözümleme sistemi ya da MACSYMA, matematikte formel işlem sistemi vs.

- '70'li yıllar bilgilerin temsil edilmesi ve usavurmada, uzman sistemler (Dördüncü Bölüm), ana-

dilin anlaşılması (Beşinci Bölüm) ya da ileri robot bilimi konularında YZ'nin temellerinin ortaya konmasına yolaçan çalışmalarda bir *patlama*ya tanık olmuştur.

- '80'ler adı geçen alanlarda çarpıcı pratik uygulamalar ve buna paralel olarak sanayileşmiş ülkelerin büyük bir bölümünde gözü yükseklerde olan projeler tarafından araştırmaya yönelik çabalarda kaydadeğer bir artış ile, *YZ'nin ekonomik hayata girişi*'ni temsil ederler.

III. YZ'nin Temel Özellikleri

YZ'nin ulaştığı alanlar sayısızdır. Bununla birlikte, bütün YZ sistemleri arasında bu disiplinin temel özelliklerini daha iyi sınırlandırma olanağı sağlayan bir ortak noktalar bütünü vardır.

Bir YZ programının özelliği her şeyden önce, "klasik" enformatikte de olduğu gibi, sayılardan çok *sembolik enformasyonlar*'ı kullanmasıdır. Bu enformasyonlar, kavramları, kuralları, nesneleri düşünürken bir insanın gözönünde bulundurduğu olguların aynılarını temsil ederler. Bu, sayısal işlem yöntemlerinin kullanılmasını dışlamaz, ancak sonuçların işlenmesi genellikle sembole dayalı olarak yapılır. Buna göre tıbbi teşhise yarayan bir sistem 38,2 °C sayısal değerinden hastanın hafif ateşli olduğu sonucunu çıkaracaktır.

YZ'de bulunan ikinci bir kavram da klasik algoritma metodlarının tersine "heuristique" metodlar kavramıdır. Program yapmak için belli bir programlama dilinde kodlanmış olan bir algoritma, verilmiş bir problemi çözmeye yönelik işlemler dizisinin eksiksiz betimlemesinden oluşur. "Heuristique", determinist olmayan yolları izleyen ve başarının garantili olmadığı ancak "çalıştığı" zaman genellik-

le işlem süresinde büyük tasarruf sağlayan bir çözüm metodudur. Başarısızlık durumunda tekrar geriye dönmek ve bir başka çözüm yolunu denemek gerekiyor. Formel integral türünde bir problemin insan tarafından çözümlenmesi genellikle şöyle bir yol ortaya koyar: İntegralin tipine göre bir değişken değişikliği denenir, sonra gerekirse bir başkası veya ilk deneme, problemin sadeleşmesini sağlamadıysa bölümler halinde bir başka integrasyon denenir. Bir problemin çözümünün "heuristique" araştırması ikinci bölümde görüleceği gibi genellikle sadece en çok umut vaadedenleri gözönünde bulundurmak üzere çözüm yollarının tamamının "kısaltılmasından" ibarettir.

YZ'de "heuristique" kullanımı çok sık görülür. Geleneksel algoritma metodları ile çözülemeyen iki problem sınıfını ele alma olanağı verir:

- İnsanın çeşitli etkinlik alanlarında çok yaygın olan ve bilinen algoritmik çözümü olmayan problemler: algılama, karar verme, anlayış;
- Eldeki enformatik olanaklar için algoritmik çözümü büyük karmaşıklık yaratan problemler. Tipik örneği oyunlar, özellikle de satrançtır.

YZ'nin bir başka özelliği de işlenen verilerin ve bilgilerin *eksik, yanlış* ve hatta *çelişkili* olduğu durumlarla yetinmesidir. Bir insan için örneğin tıbta, iş idaresinde, bankacılık sektöründe vb.'de böyle gündelik durumların sözkonusu olduğunu belirtmek gerekir. YZ çok çetin olan bu problemi tamamen çözmese de etkili metodlar geliştirir, özellikle yaklaşık veya monoton olmayan işlem vb. gibi özel işlem teknikleri biçiminde.

Önemli bir YZ kavramı da bilgidir. Başlangıçta araştırmacıların problemlerin çözümlenmesi için genel metodlar bulmaya çalışmalarına karşın, günümüzün YZ'si az sayıda ve iyice sınırlanmış alan-

larda bu alanlara özgü çok miktardaki bilgiye dayanarak problemleri çözebilen sistemleri anlamakla "yetiniyor". Bir kere daha bu yöntemin kesinlikle insanda var olduğunu belirtmek gerekiyor; kendi etkinlik alanında çok fazla bilgi ve deneyime sahip gerçek bir uzman. Bu bilgiler bir programlama dilinde prosedürler biçiminde kodlandıkları zaman, doğal biçimlerine daha yakın olarak, YZ sistemine deklaratif bir biçimde katkıda bulunurlar. Bilgi tabanlı sistemler, özellikle de uzman sistemler, günümüzde YZ yapısının önemli bir parçasını oluştururlar. Taban prensipleri dördüncü bölümde anlatılacaktır.

Son olarak, YZ özü bakımından *çokdisiplinli*'dir. Bir sistemin gerçekleştirilmesi gelişmiş enformatik tekniklerine geniş ölçüde başvurur (özgül gereksinimleri dillerin ve özel makinelerin gelişmesine de muhtaçtır). Ama YZ köklerini başka disiplinlerden de alır: mantık ve bilişsel psikoloji (bilgilerin ve kanıtlamaların gösterilme temelleri için), dilbilim (konuşma ve yazı dilinin işlenmesi için), ergonomi, felsefe ve mutlaka birgün sinirbilim ve biyoloji.

IV. YZ'nin Alanları

YZ'ye özgü bahsettiğimiz özellikler bir uygulama alanları kümesinde biraraya gelirler. Onları kısaca tanıtacağız.

1. Otomatik Teorem Kanıtlaması - Bu etkinlik YZ'nin başından beri vardır. Özellikle geometri veya cebirde, şimdilik hiçbir makine kanıtlamamış olsa da, henüz bilinmeyen teoremler alanında kayda değer başarılar elde edilmiştir. Bu çalışmalar ortaya koydukları genel işlem ve çıkarsama metodları açısından önemlidirler.

2. Yazılı "Anadilin İşlenmesi - Bir makineye

anadilinde başvurabilmek özellikle enformatik uzmanı olmayan kişilerin ellerinin altında büyük bir yaygınlıkla kullanılması için önemli bir yarar sağlayacaktır. Anadilin işlenmesi alanı aslında değişik etkinlikleri kapsar: insan-makine diyaloğu bağlamında cümlelerin açıklanması ve üretilmesi, veri tabanlarına ya da servislere giriş (örneğin Minitel yoluyla), yardımla çeviri vb.

Otomatik çeviride ilk başarısızlıklar ne "sözcüğü sözcüğüne" çevirinin, ne cümle yapısının gramer çözümlemesinin anadili anlamada yeterli olmadığını ortaya koydular. Aslında dil, bağlam, araokuyucular vb. üzerine enformasyonları ve çeşitli ve çok sayıdaki bilgiyi (sözcük dağarcığı, söz dizimi, anlambilim, pragmatik) etkili bir usavurmada işin içine katmak gerekir.

3. Sözün Otomatik Olarak İşlenmesi - Bu alan iki görünümden oluşur: Tanıma (makineye konuşmak ve anlamasını sağlamak) ve sentez (makineyi konuşturmak). Yazılı anadilin işlenmesi ile benzerlikler sunar. Burada yine, makine tarafından bir cümlenin anlaşılması, birçok bilgiyi devreye sokan bir süreçtir. Fakat, benzeşen sinyali bozan gürültüler (adeta mikrofondan çıkar gibi sesli sinyal) ve büyük akustik değişkenliği (okuyucuya ya da fonetik ve sesbilimsel bağlama göre) işleme zorluğunu çoğaltırlar.

Deneysel insan-makine diyaloğu sistemleri laboratuvarda çalışırlar. Bununla birlikte, problemin olağanüstü zorluğu nedeniyle, ticari aletler, öğrenimden sonra bir ya da birkaç okuyucu tarafından telaffuz edilen özel kelimelerle sınırlı bir sözcük dağarcığını bilmekle yetinirler. Akustik biçimlerin bu karşılaştırması, biçimlerin tanınması tekniğini ortaya çıkarır.

4. Görüntülerin Yorumlanması ve Bilgisa-

yar Tarafından Görme - Görme bir makinenin çevresini farketmeye yönelik başka bir olanaktır. Aynen işitsel algılama durumunda olduğu gibi görme probleminin basitleştirilmesi, basit görsel formların algoritmik form tanıma metodları yardımıyla tanımlanmasından ibarettir: örneğin bir metnin baskı karakteri, bir sahnenin yerleri iyi saptanmış eşyaları, mikroskop altında sitolojik bir resmin üzerindeki hücreler ya da kromozomlar, bir uydu resmi üzerindeki özel bölgeler vs. Bununla birlikte, bir sahnenin ya da bir görüntünün gerçekten anlaşılması, bağlamsal bir işlemi yürütmek için düşünülen alana özgü bir bilgi bütününe dayanarak bu metodların ötesine gitmeyi gerektirir: tıbbi teşhis amacı ile radyolojik görüntülerin açıklanması, basılı ya da elyazısı bir metnin anlaşılması, üretim zinciri üzerindeki bir nesnenin kontrolü, özerk bir aracın kılavuzluğu vb.'de olduğu gibi. Bu son etkinlikler özellikle robotbilime aittir. Görüntülerin işlenmesi YZ'nin endüstriyel alanlardaki ilk uygulamalarından biridir.

*5. **Robotbilim*** - Robotbilimin geniş alanı, özellikle sanayide, iş otomatizasyonu etkinliklerini kapsar (fabrikasyon, yönetim, tamir). Varolan robotların büyük bir bölümü (örneğin bir montaj hattındaki kaynakçı ya da boyacı robotlar), işleri sırayla durmadan tekrarlarlar ve zekâları yoktur. Buna karşılık yeni kuşak robotlar, giderek çevrelerini algılamaya (özellikle görüntü ile) ve hareketlerini planlamaya yönelik zekâ yeteneği ile donatılmışlardır (bir görevi sonuca ulaştırmak için, bir stratejinin ya da bir eylem planının oluşturulması, bir görevin tamamlanması sırasında ortaya çıkan bir durum ya da zorluk karşısında dinamik olarak yeniden plan yapmak). Böylece YZ, yarının sanayisinin önemli bir bileşeni olan robotbilimde sürekli artan

bir rol oynar.

6. Oyunlar - Oyunlar (satranç, dama, go, tavla), araştırmacılar için YZ'nin ta başından beri tercih edilen bir alan oluşturmuştur. Bu çalışmaların amacı oyunla ilgili olmaktan öteye gitmiştir, çünkü "heuristique" stratejileri kullanarak genel çözüm arama kavramlarına bir kere daha dokunulmuştur. Sistemler genellikle olumlu başarılara ulaşmıştır, henüz hiçbir program, büyük bir satranç ya da go ustasını egale etmeyi başaramamış olsa da. Bu alana yönelik çalışmalardaki gelişmeler, YZ'nin evrimini yeterince açıklıyor: İlk sistemler, bilgisayarın tek kuvveti üzerine kurulmuştur: Kısıtlı bir zamanda çok sayıdaki çözüm yolunu gözönünde bulundurma becerisi. Bu arada, bu problemlerin katışımsal patlaması öyledir ki bu patlamayı önlemek için "heuristique"in kullanımı kısa zamanda kaçınılmaz olmuştur. Daha sonra amaca ulaşmak için (rakibi yenmeyi bilmek) geliştirilmiş hareketler planı kavramı devreye sokulmuştur. Bu son gelişme bir kez daha, incelenen oyuna özgü bilgilere başvurmuştur.

7. Uzman Sistemler - YZ etkinliklerinin büyük bir bölümünün işlenen alan hakkındaki bilgilere başvurduğunu gördük. Uzman sistemler bilgi tabanlı sistemlerin en çarpıcı örneklerinden birini oluştururlar. Göreceli olarak son zamanlarda ('70'li yılların başı) ortaya çıkmış olan bu sistemler kısıtlı alanlarda farklı iş türlerindeki (teşhis, tavsiye, planlama, anlama) uzman kişilerin performanslarına ulaşma yeteneğine sahiptirler. Usavurumu sırasında uzman bir sistem tarafından kullanılan bilgiler önce uzmanların yanında elde edilip şekillendirilirler ve bir bilgi tabanına geçirilirler.

İKİNCİ BÖLÜM
PROBLEMLERİN ÇÖZÜMLENMESİ

I. Yapay Zekâ'da Problemlerin Çözümlenmesinin Yeri

1. İncelenen Problem Türleri - İlk kez bir bilgisayara çözdürülmeye çalışılan problemler arasında, çeşitli oyunlar ve teoremlerin kanıtlanması önemli bir yer tutar. Bu iki alanda da insanın bilinen zihinsel performansına karşın makinelerin bellek kapasiteleri ve hesaplama çabukluklarının; onlara daha çabuk ve en azından aynı düzeyde başarılı olma olanağı vereceği düşünülmüştür.

Bu problemler incelendiğinde gerçekte şu sonuç çıkıyor: Bir çözüme ulaşmak için araştırılacak yolların sayısı o kadar çoktur ki, makinenin bu basit "yapı" özellikleri geçerli performanslara ulaşmak için son derece yetersizdir.

Burada satranç örneği açıklayıcıdır. Taşların tek yönlü oynama kuralları ile yetinerek yapılacak en iyi hamlenin aranması oyunun belli bir anında çarpıcı sayıdaki hamle, olası karşılık, karşılığa verilecek karşılığın gözden geçirilmesini gerektirecektir; bu olasılıklar bugünkü herhangi bir makine ve büyük bir olasılıkla geleceğin makineleri için de çok fazladır. Buna karşılık, becerikli oyuncunun yöntemi gözlemlendiğinde, yöntemin hesabın kuvvetine değil kuralların çok daha zekice kullanılmasına dayandığı görülür: oyunun durumunun taşların sayısı ve pozisyonuna göre değerlendirilmesi, kurallara uygun olası hamlelerin saptanması, merkezî ko-

numların kontrolünün aranması, taşların elverişli takası vb. gibi saldırı veya savunma stratejilerinin ortaya konması. Makineye satrançta düzeyli oynama olanağı sağlamak, onu böylesi stratejik kapasitelerle donatmadan gerçekleşemez.

Aynı şekilde yukarıda söz edilen problemlerin otomatik çözümü, zekice yöntemlere dayalı özgül yapay zekâ tekniklerinin devreye sokulmasını zorunlu kılmaktadır.

Planlama problemleri (ya da hareket planlarının geliştirilmesi) programlama ve anlama sentezleri de aynı kategoriye dahil edilebilir. Bu son nokta hakkında, yazı ve konuşma dili ve görme konusunda daha sonra tekrar konuşulacak. O halde problemlerin çözümü YZ'de merkezî bir rol oynamaktadır. Genellikle insanın kolaylıkla ya da uygun bir sürede çözemediği problemlerle uğraşılır, ancak özelliklerinin tamamen belirtilebilmeleri koşuluyla. Usavurma sırasında insan müdahalesi olmadan, özellikleri kötü belirtilmiş bir problemi çözmeye kalkışmak başarısızlığa yolaçacaktır.

2. Problem Örnekleri - Aşağıdaki örnekler temelde oyunlar ve karar verme ile ilgili alanlara aittir. Çoğunlukla problemlerin otomatik çözümü konusunda bilinen seçme örneklerdirler.

a) *Çiftçi, Kurt, Keçi ve Lahana*.- Dördü birden bir ırmağın sol yakasındadırlar. Sadece çiftçinin çalıştırdığı bir tekne onu ve muhtemelen kahramanlardan birini bir kıyıdan diğerine taşıyabiliyor. Çiftçinin yokluğunda kurt keçiyi, keçi de lahanayı yiyecektir. Hepsi kazasız belasız sağ kıyıya nasıl geçirilecektir?

b) *Kaplar*.- Su ile dolu 5 litrelik bir kabımız ve 2 litrelik boş bir kabımız var. Su bir kaptan diğerine aktarılabilmekte ya da bir kaptan dışarıya boşaltılabilmektedir.

2 litrelik kapta 1 litre su nasıl elde edilebilir?

c) *3x3 Oyunu.*- Burada sözkonusu oyun, üzerinde 1'den 8'e kadar numaralanmış 8 kare pulun olduğu 9 haneli bir tablodur. Bir pul, varsa bitişiğindeki boş haneye (9. haneye) kaydırılabilir. Belirli bir başlangıç şeklinden yola çıkarak nasıl verilen sonuç şekline ulaşılabilir? Eğer mümkünse...

d) *Sahte Para.*- Görüntüleri aynı olan on iki tane bozuk paramız var. İçlerinden birinin ağırlığı farklıdır.

"Sahte parayı" ayırdetmek için bir Roberval tartısı kullanarak yapılacak asgari tartma sayısı kaçtır?

e) *Formel İntegral Hesabı.*- Analiz tabanı fonksiyonlarının integralleri bilinmektedir.

Aşağıdaki örneği hesaplayınız:

$$\int 2 \sin(x) \cos(x)\, dx.$$

f) *"Küpler Dünyası"nda İstifler.*- Soldaki görüntüden başlayarak sağdaki istifi elde etmek için küplerin yerini değiştirin:



g) *Seyyar Satıcı.*- A şehrinden yola çıkan bir seyyar satıcı, bir şehirden iki kere geçmeden ve sonunda A'ya dönmek koşuluyla B, C,...H şehirlerine sırayla gitmek istiyor. Şehirlerin arasındaki uzaklıklar bilinmiyor.

Katedilen mesafeyi en aza indirecek şekilde

hangi güzergahı izlemelidir?

h) *Sekiz Vezir.*- Sekiz tane veziri satranç kurallarına göre alınmayacakları bir şekilde satranç tahtasına yerleştiriniz.

i) *Hanoi Kuleleri.*- Üç tane disk (bir büyük, bir orta ve bir küçük) 1 numaralı çubuğa sırayla geçirilir.

Çubuktaki disklerin en üstündekini alarak ve bir başka çubuktaki yığının en üstündeki diskin (ya da disk bulunmayan tek çubuğun üzerine) üzerine yerleştirerek bu diskleri 2 numaralı çubuğa geçiriniz.

Yığın halinde iken bir disk ancak daha büyük bir diskin üzerinde bulunabilmektedir.

3 numaralı bir ara çubuk kullanılabilir.

j) *Maymunlar ve Muzlar.*- Aç bir maymun ancak bir kasanın üzerine çıkarak erişebileceği bir salkım muzu ele geçirmek istiyor. Maymunun uygulayabileceği hareketler şunlardır: tek başına hareket etmek, kasayı iterek hareket etmek, tırmanmak, muzları yakalamak.

3. GPS, Bir "Genel Problem Çözücü" - GPS sistemi, *General Problem Solver,* 1959'da A. Newell ve W. Simon tarafından tasarlanmış olup, problemlerin çözümünde insan etkinliğinin daha iyi anlaşılmasına yönelik ilk çalışma çerçevelerinden birisidir.

Başlangıçta bu etkinliğin çeşitli alanlarda uygulanması istenmişse de pratikte uygulama alanı, bilmecelerin çözümü ve sembolik integrasyon ile sınırlı kalmıştır. Bu sınırlılık, çalışmaların başlangıcındaki yaratıcılardan daha az iyimser olan güncel görüş ile uyuşmaktadır: Güncel sistemler (uzman sistemler) kullanılan araştırma sistemleri ("heuristique" kavramı) genel olarak sınırlı bir işe yöneliktirler ve özel konularla ilgilenirler. Bu noktada prob-

lemlerin "genel çözücüsü" düşüncesinden tamamen uzaklaşırlar.

Bununla beraber YZ araştırmalarında GPS sisteminin etkisi çok büyük olmuştur. Öte yandan GPS temelli düşünceler (içinde bulunulan durum ile ulaşılacak hedef arasındaki değişikliklerin azaltılmasına yönelik araştırma ve çabalar) bazı hareket planı yaratıcılarının kaynağıdırlar.

4. Bir Problem Üzerinde Nasıl Düşünülür? - Bir problem çözümlemesinin değişik aşamalarının gösterilme teknikleri ve işlem metodu belli sayıdaki belirticiye bağlı olacaktır. Örneğin şunlar sorulabilir:

a) Çözümleme ile yükümlü sistem (yani program) problemle ilgili bildiklerinden başlayarak bir çözüm bulmak zorunda mı ya da kullanıcı ile etkileşim veya harici enformasyonlara girme sözkonusu olabilir mi? Sorun, bu harici enformasyonlara giden yolları en elverişli hale getirme tarzının araştırılmasında uygun bir dizi bulmak olacaktır. Bu sorun özellikle etkileşimli bilgi tabanlı sistemler konusunda tekrar görülecektir.

b) En uygun olmayan ancak daha kolayca ve hızlıca elde edilebilen bir çözümle yetinilebilir mi? 2g'deki seyyar satıcı örneğinde en kısa olmayan ancak "yeterince" kısa olan bir güzergah tatmin edici olabilir.

c) Düşünme sırasında karar verilen bir hareket bir kez gerçekleştirildikten sonra iptal edilebilir mi? 2c'deki 3 x 3 oyunu örneğinde her zaman bir pulun yer değiştirmesinin, yerine bir başkasının gelmesi ile dengelenebileceği açıktır. Buna karşılık dolu olmayan bir kabın içindekini önceden sıvı içeren daha büyük bir kaba boşaltmak olgusunun aksi mümkün değildir.

d) Bir hareketin, problemin o anki durumu üze-

rindeki etkisini tam olarak öngörmek mümkün müdür? 2f örneğinde verili bir görünümden yola çıkarak bir kübün yerini değiştirmek bilinen bir görünümün oluşması ile sonuçlanır. Bunun tersine yere kağıt açmak gereken bir kağıt oyununda rakiplerin oyununun en azından bir kısmı üzerinde spekülasyonlar yapılabilir.

e) Problem, çözümü daha kolay "alt-problemlere" ayrıştırılabilir mi ve *bu* alt problemler bağımsız mıdır, değil midir? Temel fonksiyonları bilinen integrallere indirgenmeye çalışılan bir integral hesabı bu duruma örnektir.

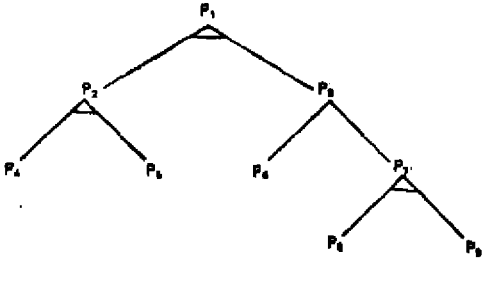
Özellikle bir problemin çözümüne ulaşmak için gereken asgari bilgiler hakkında daha başka sorular da sorulabilir.

5. Bir Problemin Çözümlenmesinin Gelişimini Gösterme Yolları -

a) *Durumların Grafiği.*- Çözüm sırasında bir problemin birbirini takibeden durumlarının aralığı (espace), her bağın bir durumu gösterdiği ve her çubuğun bir durumdan diğerine değişimi sağlayan bir geçişi gösterdiği grafik biçiminde ortaya konur.

Devir olmazsa grafik ağaç biçimini alır.

b) *VE-VEYA Problem İndirgeme Grafiği.*- Aşağıdaki grafiği ele alalım (burada ağaç şeklindedir):



Şekil 1.- Bir VE-VEYA problem çözümleme ağacı

Bu ağaç P1 başlangıç probleminin, kendileri de P4 ve P5'in ya da P6.VEYA P7'nin çözümlenmesi ile çözümlenebilecek P2 VE P3'ün çözümlenmesiyle çözümlenebileceği gerçeğini gösteriyor. Son olarak P7, P8 VE P9'un çözülmesiyle çözümlenebilir.

P1 ağacın kök-bağıdır. P4, P5, P6, P8 ve P9, bu alandaki sistem bilgileri gözönüne alındığında çözümün çabuk olduğu yaprak-bağlarıdır.

Yukarıda görülen iki gösterme türünde anahtar yapı grafiktir (ya da ağaç). Bir problemi çözmek bu yapıda ilerleyerek mümkün olacaktır.

II. Problem Durumları Aralığı (Espace)

1. Gösterme - Bir problem, içinde aşağıdakilerin yeraldığı bir üçleme (I, O, B) gibi görülebilir:

* I, ilk durumların tamamını gösterir (tekse ilk durumu);

* O, sistemin bilinen ve bir durumu bir diğeriyle değiştirme olanağı veren operatörlerinin tamamını gösterir; ve

* B, hedef-durumların tamamını gösterir.

Bir durum (ya da bir "konum"), ele alınan probleme bağlı verileri birbirine bağlayan bir nesneler bütününü ve ilişkileri yansıtan, bir veriler yapısı olarak görülecek. Satrançta bu yapı satranç tahtasını gösteren bir tabloyu ya da bir çiftler listesini (hane numarası, taş ismi) kapsayabilir. Küpler ve bir robota yaptırılacak işler alanında:

SURTABLE (A),
SURTABLE (B),
SUR (C, A)

gibi bir yüklem listesi I-2f örneğinin başlangıç durumunu tanımlayacaktır.

Problemin bir çözümü, O'ya ait olan I'nin bir elemanından B'nin bir elemanına gitmeyi sağlayan

tamamlanmış Ok operatörleri dizisinden oluşacaktır.

O halde problem çözümü aralığı (espace) bir durumlar grafiği ile gösterilmiş olacak ve problemin çözümü bir çıkış bağından bir sonuç bağına götüren bir yoldan oluşacaktır.

Çözümleme sistemi şunları kapsar:

- bir grafikle düzenlenmiş veriler yapısı bütünü;
- uygulama koşulları ve hareketleri ile biçimlendirilmiş operatörler bütünü;
- çözümleme stratejisini ortaya koyan kontrol yapısı.

Grafik grubunun çözümleme sırasında kullanılan, sistematik hafızalaması, elde bulunan hafızanın hızla doymasına yolaçabilir. Aslında çoğunlukla bir durumu hafızaya almak sadece birkaç değişiklikle sağlanan durumun yeniden kopya edilmesi anlamına gelir. Bu problem *frame problem* adıyla bilinir.

Bu zorluğun çözümü başlangıç durumunu tam olarak tanımlamak, sonra da uygulanmış olan operatörlere bağlı değişiklikleri vermektir (ya da operatörlerin kendilerinin takip sırasını). Bu şekilde yer ve "tekrar kopyalama" zamanı sorunu çözümlenmektedir. Durumlar grafiğinin güzergahında geriye dönüş *(back tracking)* örneğinde olduğu gibi operatörün hareketi iptal edilmeyecekse bu uygun bir çözümdür. Bu son durumda, etkileri yokedilmek istenilen bir önceki operatör uygulamasını tekrar bulmak için özel metodlara başvurmak gereklidir.

2. Örnekler -

a) *Çiftçi Problemi (I-2a).*- Sol yakada olmayanın sağ yakada olacağı varsayılarak, sadece sol kıyıda bulunanın bilinmesi ile problemin bir durumu karakterize edilebilir. Bu, doğru olduğu açıkça bilinmeyen her şeyin yanlış kabul edildiği kapalı alan

denilen varsayıma tekabül eder. Bu varsayım, daha gerçekçi olan uygulama alanları bilgisinin bütünsel olmadığı açık alan varsayımı ile çelişir.

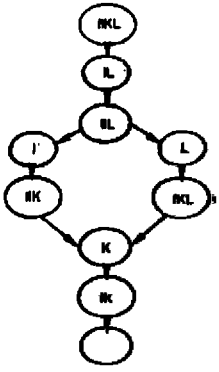
İki operatör şunlardır:

* ya çiftçiyi sol yakadan sağ yakaya yalnız veya sol yakadaki oyunculardan biri ile geçirmek

* ya da tam tersi.

Aslında farklı varlıklara uygulanabildikleri ölçüde "operatör şemaları"ndan bahsedilmektedir.

İncelenen problemin durum aralıklarının (espace) tamamı çok fazla değildir ve şekil 2'de gösterildiği gibi çizilebilir. (kısaltılmış olarak K ve L keçi ve lahanayı gösteriyor):



Şekil 2. Çiftçi probleminin çözümü

Bağların operatörler ile etiketlenmesi gerekmektedir.

Tehlike durumları (çözümdeki çıkmazlar) gösterilmemiştir.

Çözüm yedi hareket ile elde edilmiştir.

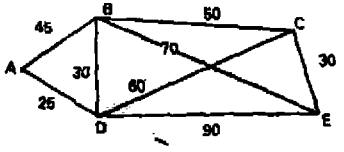
Bu şemada olduğu gibi etkileri geçersizleşen operatörlerin sıralı uygulamasından kaçınılırsa şe-

mada devir olmaz. Bu özel örnekte teorik olarak yönlendirilmiş olan şemanın iki yönde gelişebileceği saptanmaktadır; problemin simetrisinden dolayı karşıt parkurda başlangıç kıyısındaki bütün oyuncuları karşıya geçirir. Bu genelleştirilemez.

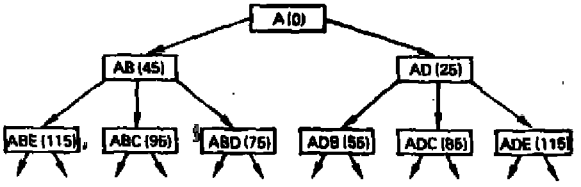
Burada araştırma aralığı göreceli olarak sınırlıdır. Buna karşılık örneklerin çoğunluğunda, tam aralığın sadece potansiyel bir varlığı vardır. Çözümleme aşaması bu aralığın sadece çözümü kapsadığı sanılan bir parçasının açıkça belirtilmesinden ibarettir ve bu en iyi aralık ve olası hesap zamanı koşullarıdır.

b) *Seyyar Satıcı Problemi (I-2g).*- Bu problem bir öncekinden şu olgu ile ayrılır; Grafiğin her aşamasına başlangıç-bağından itibaren katedilen kilometre sayısını gösteren sayısal bir değer ilave edilmelidir; buna sayısal değerli grafik denir. Çözüm-bağı her şehirden geçişi ve asgari değerlendirme ile A'ya dönüşü gösterir.

Sayıların şehirlerarası uzaklıkları gösterdiği aşağıdaki örneği ele alalım:



Potansiyel durumlar grafiği aşağıdaki gibi başlar·



Yaprak-bağlar üzerinde şu şekilde son bulur: ABCEDA(240).

Grafikte derinlemesine ilerlendiğinde bağ sayısında "katışımsal patlama" denilen bir durum gözlenir (her şehir bütün diğerlerine doğrudan bağlı olsaydı bu patlama daha çarpıcı olacaktı).

Burada sadece grafiğin tamamının açımsanması asgari kilometre ölçüsüne karşılık veren çözümün ne olduğunu söyleme olanağı verir.

Bu arada genellikle, böyle bir araştırmadaki zaman ve hafıza-aralığının sınırlı olma özelliği nedeniyle "optimum-altı" bir çözümle yetiniliyor.

c) *3 x 3 Oyunu Problemi (I-2c).*- Durum grafiği oyun üzerinde gösterilmiş özel niteliklere sahip olabilir:

* "Monoton Olmama": Grafik, etkileri iptal olan operatörlerin sıralı uygulanmasına tekabül eden devirlerden oluşur. Burada bir pulun bir devir ile yansıtılacak olan gidiş ve gelişi işaretlenmeli ve hesaba katılmamalıdır. Aksi halde çözümün algoritması sonsuz bir kıvrım haline dönüşebilir.

* "İlgili Olmama": Oyunun sökülebilir olduğu ve iki pulun yerlerinin değiştirilebileceği varsayılırsa verili bir durumun Uo başlangıç durumundan ya da Uo'nun iki pulunun yer değiştirmesinden elde edilen Uo durumundan türeyebileceğini doğrulamak kolaylaşır. Eğer çözüm-durumu Uo ile aynı ilgili bileşende bulunmuyorsa Uo'dan başlayarak sonuca gidilemez.

3. Çözümleme Ağacının Sistemli Geliştirme Teknikleri - Bir ağaca uygulanabilen bütün grafikler, burada basitçe gözden geçirilecek olan ağaç yolları teknikleridir.

O kümesi operatörlerinin tamamı ya da bir kısmının uygulanması ile tüm olası ardıllarını doğuran operasyona "bağ-geliştirilmesi" adı verilir.

a) Enlemesine araştırma *(breadth-first search)*: Bu araştırma metodu aynı derinlik seviyesindeki bağların sırayla geliştirilmesinden ve çözümün bulunamaması halinde bir sonraki derinlik seviyesinde sürecin yinelenmesinden ibarettir. Bu nedenle alternatifler p derinlik seviyesinde hafızaya alınmış bağlardan en eskisinin, p + 1 seviyesindeki bağları doğurmak için öncelikle geliştirildiği veriler yapısında bir diziye (FIFO, *first-in first-out*) yerleştirilmiştir. Başlangıç durumu geleneksel olarak sıfır derinlik seviyesine yerleştirilir.

Enlemesine araştırma algoritmik olarak nitelendirilir, çünkü ağacın belli bir sayıda bitmiş dalı olduğu sürece bir sonuç varsa bu sonucu bulmayı garanti eder.

Bu metod çok hafıza-aralığı (belli bir seviyede aynı anda hafızalanması gereken bağ sayısı üslü bir şekilde artar) ve fazla hesap çabası sarfetme sakıncasına sahiptir. Sonuç olarak bu metod bir çözüme bir çok yolla ulaşılan ve yolların özellikle uzun olduğu problemlere uygulanmamıştır.

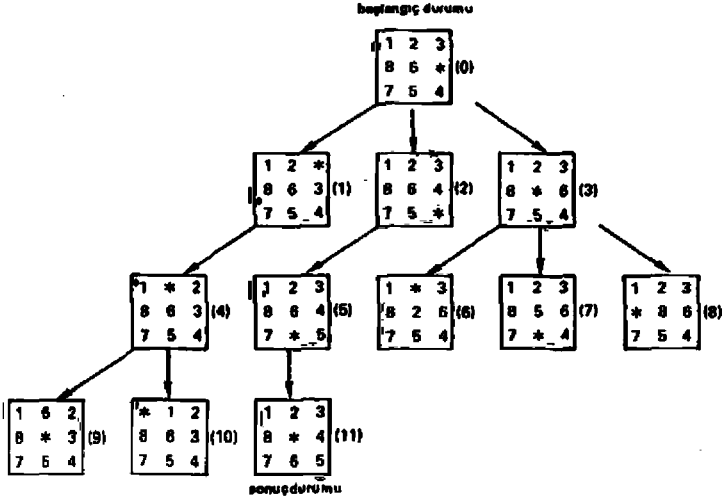
3 numaralı şekil 3 x 3 oyununda enlemesine araştırmayı gösteriyor.

Operatörlerin uygulanma düzeni yani yer olduğunda pulların yer değiştirme düzeni şöyledir: aşağıya doğru, yukarıya doğru, sağa doğru ve sola doğru. Bağlardaki sayılar geliştikleri sırayı gösterirler.

b) Derinlemesine araştırma *(depth-first search)*: Burada yöntem giderek artan derinlik seviyelerine doğru giden yolun araştırılmasından ibarettir. Başarısızlık durumunda (çıkmazla sonuçlanan kol) başlangıç bağına geri dönülür, mümkünse ardıllarından bir başkası araştırılır ve yeniden derinlemesine yol alınır. Yarıda bırakılan bağları hafızaya kaydeden verilerin yapısı bu defa, en son hafızalanan bağın öncelikle işleneceği bir yığındır (LIFO

last-in first-out).

Bu araştırma metodu, sonuçta başarının olmadığı sonsuza dek uzayan yollara götürebilir: Bu nedenle algoritmik olmadığı söylenir.



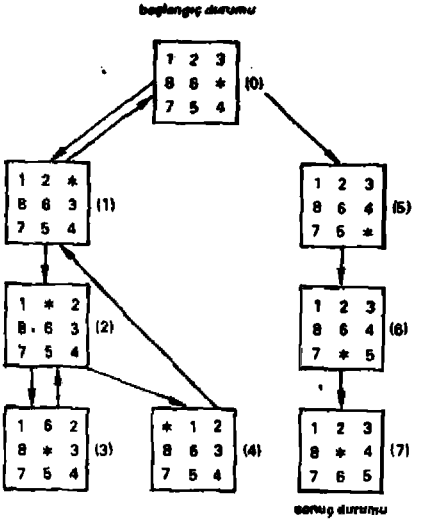
Şekil 3.- 3 x 3 oyununun enlemesine çözümlenmesi

Limit derinliği kavramı sayesinde algoritmik hale getirilir. Araştırma Pe olarak belirtilen maksimum bir limite kadar derinlemesine yapılır. Başarısızlık halinde Pe'nin değeri arttırılır ve süreç yinelenir. Geriye dönüşlü derinliğin sözkonusu olduğu söylenir.

4 numaralı şekil 3 x 3 oyununda Pe= 3 ile bu metodu gösteriyor (eğer Pe = 2 seçilseydi 2 seviyesinde gelişmenin tamamlanacağı, çözümde başarısız olunacağı, sonra da 3 seviyesine geçileceği belirtilmelidir).

Klasik olarak kullanılan belli sayıdaki başka metodlar bu kör metodlar kategorisine aittirler. Özellikle bir sonuca götürüp götürmediğini görmek

ve başarısızlık durumunda tekrar başlamak üzere, sonu belli olmayan bir yolun tamamını kateden metoddan bahsedilebilir. Bu metod bir sonuç bulmayı garanti etmez. Bu, sonuca götürmeyecek gibi duran yolları bırakarak daha az sistematik hale getirilebilir. Brüt formüllerinden ve ağırlık spektrumlarından yola çıkarak organik moleküllerin yapılarının aydınlatılmasına yönelik olan ilk "uzman sistem" DENDRAL, bu yöntemi kullanıyordu.



Şekil 4.- 3 x 3 oyununun denlemesine çözüm.

4. 'Heuristique'leri Devreye Sokan Teknikler -

a) *Genel.*- Yukarıda görülen sistematik metodlar genellikle çok etkili değildirler, çünkü çözümlenecek problemin özelliklerini hesaba katmazlar. Ayrıca bir durumlar grafiği ile şekillendirilen problem karşısında insanın araştırma biçimine uygun düşmekten uzaktırlar.

Çözümleme performansını arttırmak için YZ *"heuristique"* metodları devreye sokar. Bir çözüm bulmak her zaman garantili değildir ve bir tane bulunursa da optimum olması şart değildir. Ama daha az güç sarfederek çözüm bulmak düşünülebilir. Ayrıca çoğunlukla optimum çözümü bulmak zorunluluğu yoktur. Örneğin planlama problemlerini çözmekle yükümlü insanlar optimum çözümü değil, uygun bir zamanda elde edilecek "iyi" bir çözümü araştırırlar.

"Heuristique"leri devreye sokmak araştırma kontrolü stratejilerinin devreye sokulmasıdır. İki teknik düşünülebilir ve üst üste konulabilir:

-Verili bir durumdan yola çıkarak uygulanacak olan operatörlerin ödeme emri (ordonnancement) fonksiyonunu kullanmak;

-Bir bağın diğerlerine göre geçerlilik süresini hesaplamak için değerlendirme fonksiyonunu kullanmak.

Çözüm sürecinin her aşamasında geliştirilecek bağlar yeniden düzenlenir (bir yığında ya da dosyada sistematik olarak yerleştirilmek yerine): O halde düzenli araştırmadan sözediliyor.

b) *Bir Değerlendirme Fonksiyonunun Kullanılması.*- Düzenli araştırma belli bir ölçüye yönelik en iyi bağı geliştirmeyi seçmek için geri döner. Sonuca gitmede çok şansı olan bağlara dayanarak çok sayıdaki problem türü için sistematik metodlardan daha başarılı olarak ortaya çıkar.

Bağların değerlendirilmesi prensibi bir fiyat fonksiyonu sayesinde araştırmanın belli bir anında ortaya çıkan bütün seçenekler üzerinde tam bir düzen tanımına bağlı bulunmaktadır.

Bir n bağından geçen asgari fiyatın yol fiyatının ideal bir şekilde hesaplanabileceği n'i düşünelim. F* (n) denilen bu miktar, n'nin ilk durum olan g*

(n)'ye uzaklığına ve hedef bağı h* (n)'ye uzaklığına bağımlı olarak iki parçaya ayrılabilir, şöyle ki:

$f^*(n) = g^*(n) + h^*(n)$

Tabi ki elimizde i'yi belirtirken $f^*(i) = h^*(i)$ başlangıç durumu var.

g* fonksiyonu araştırmanın bir anında, daha önce karşılaşılmış olan ve ilk durumdan o andaki bağa götüren en iyi yolun fiyatı ile değerlendirilebilir. Bu fiyat fonksiyonu g olsun. Elimizde her zaman

$g(n) \geq g^*(n)$ vardır.

g'nin seçimi işlenen alana kuvvetlice bağlıdır.

I-2c'deki 3 x 3 oyununda g(n) için başlangıç durumundan beri oynatılmış pulların sayısı seçilebilir, bunun anlamı verili tanıma göre bağın derinlik derecesidir.

N bağı işlendiği sırada sonuç-bağına götüren yol bilinmiyorsa h* (n) değerlendirilmesi daha zordur. Bu durumda sözkonusu alanda "determinist olmayan" bir seçim yapma olanağı veren "heuristique" enformasyonlara başvurulması gerekir.

Bağları düzenlemek üzere f(n)'ye dayanan algoritma A algoritması olarak adlandırılır. Grafik bitmiş ve bir sonuç elde edilmişse A algoritması bir yol göstererek durur.

Eğer h, h*'nin bir küçülteniyse, yani eğer bütün n'ler için $h(n) < h^*(n)$ doğrulanmışsa, ve eğer çözümler varsa, bu algoritmanın sonuca giden en iyi yolu bulacağı ispatlanmıştır: Bu aynı zamanda "akla uygun" da denilen A* algoritması (A yıldız)dır.

H fonksiyonu benzer şekilde nülse (boş) ve g(n), n'nin derinlik derecesini eşitliyorsa, sonuçta akla uygunluk özelliğini doğrulayan enlemesine algoritmaya varılacağı saptanabilir. Bu sırada, gerçek "heuristique"in yokluğu çözüm araştırması süresini arttırır.

A algoritmasında değerlendirme fonksiyonlarının "heuristique" gücü doğrudan doğruya h fonksiyonunun seçimine bağlıdır. 3 x 3 oyununda, h için, son duruma göre yanlış yerleştirilmiş pulların sayısı alındığında, bir h* küçülteninin elde edileceği kesindir, fakat bu seçim kesinlikle en iyisi değildir.

Şekil 5 bu örnekte g için derinlik derecesini ve h için yanlış yerleştirilmiş pulların sayısını (parantez içindeki değerler) alan ordinatlı araştırınayı göstermektedir.

Bu örnek üzerinde sonuca varmak üzere, h için daha ziyade her pulu son hanesine götürme olanağı veren minimum yer değiştirme sayılarının toplamı alınarak ya da daha iyisi bilgiden ve bir oyun organizasyonu metodundan yararlanarak araştırmanın etkililiği arttırılabilir.

Genellikle h*'nin bir küçülteni olmayan h fonksiyonu kullanılarak ("heuristique") güç arttırılır (akla uygunluk adına). O halde algoritma daha az karmaşıktır ve daha kapsamlı problemleri çözme olanağı verir. Ayrıca böyle bir h fonksiyonu daha kolay hesaplanabilir ki bunun işlenen bağ sayısının ve işlenme süresinin azaltılması gibi çift avantajı vardır.

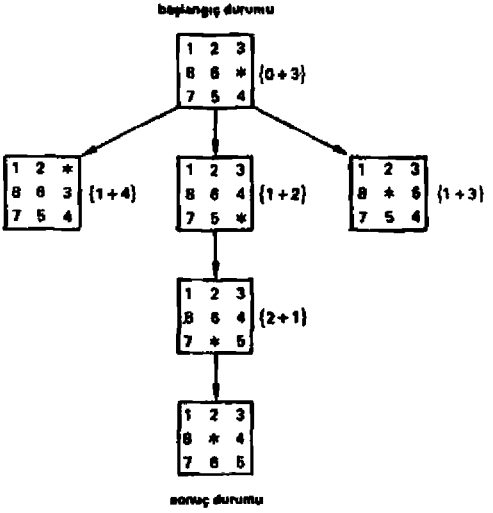
Bazı problemler için f = g + w x h biçiminde bir f fonksiyonunu akılda tutmak ilginçtir; burada w pozitif değerleri olan bir ağırlık fonksiyonudur.

Yükseltilmiş w, "heuristique" bileştireni arttırır ve fiyatı ile uğraşmadan bir yol bulma olanağı sağlar (ancak elde etme değeri ile uğraşarak).

Bunun tersine küçük w öncelikle enlemesine bileşkeni arttırır ve çözüm şemasının hiçbir bölgesinin tamamen esgeçilmemesini sağlar.

Deney, w'yi derinlik derecesinin tersine değiştirme düşüncesini esinlemektedir.

Gerçek bir durumda, optimum çözüme yaklaş-



Şekil 5.- 3 x 3 oyununun düzenli araştırma ile çözümlenmesi

ma çabasının değeri, h fonksiyonunu hesaplama çabası ile geliştirilen bağ sayısı arasında bir ortayol, aranacaktır.

Araştırma aralığını sınırlamak için başka "heuristique" metodlar da elde edilebilir. Örneğin,

- sadece belli bir değerlendirme fonksiyonuna yönelik olan en iyi bağları tutarak araştırma ağacı kısaltılabilir *(best-few search);*

- öncekilerle hiç uğraşmadan ilk olarak sonuca en yakın olan bağ ile ilgilenilebilir *(best-first search);*

- bir operatör uygulama düzeni tanımlanabilir, ki bu da operatörlere bir önem sırası kazandırır;

- bir çözüm-yoluna ait olma şansının az olduğu düşünülen bağlar elenebilir;

- söylenildiği gibi en ucuz fiyat seçilebilir ya da

yapılmamışsa en pahalı "değeri olan" seçimi yapabilirsiniz (burada *a posteriori* fiyattan bahsediliyor).

Son olarak iki yönlü araştırma gibi bazı teknikler çözümü hızlandırabilirler. Eğer tamamen biliniyorlarsa hem başlangıç durumundan hem de sonuç durumundan yola çıkarak çözüm yollarına ulaşmaya çalışılabilir. Yerleştirilmesi zor olan metod çok kullanılmaz. Tek yönlü araştırmanın etkisiz olduğu ortaya çıkarsa bu metod düşünülür. Geliştirilecek bağ sayısının azaltılacağı düşünülebilir, ancak araştırmanın sınırlarının kesişmesi olmama riski göze alınır, bu da metodun yararlılığını ortadan kaldırır.

5. Zor Problemlerin Çözümlenmesi - İncelenen problemin, anlatımında çoğunlukla üstü kapalı şekilde dile getirilen özelliklerindeki zorluklar kastediliyor. Zorluklar, bir yandan göstermenin basitleştirilmesini diğer yandan araştırma alanının kısaltılmasını içeren çifte avantaj sunarlar.

Şu halde amaç, problemin verili zorluklar bütününü karşılayan bir durumu bulmaktır. Bu gizli mesajlar ya da sözün otomatik işlemi, görme ve planlama problemlerindeki durumdur (zaman, fiyat, hammadde, elde bulunan makineler zorluklarıdır).

Bu nedenle, bir durumun tanımlanmasının dışında birbirine bağlı zorluklar listesini de edinmek gerekir.

Her bir bağın geliştirilmesi sırasında zorluk kurallarının uygulanması, olası bütün yeni zorlukların ortaya çıkmasına yolaçmaktadır. Zorluklar arasında bir karşıtlığa rastlanırsa empasa gelinmiştir. Çözüm tamsa başarı vardır, aksi takdirde araştırmayı sürdürmek üzere yeni bir bağ seçilir.

Klasik örnek (bir öğrencinin ailesine mesajı...)

Kripto çözücüsü

```
  SEND
+ MORE
------
 MONEY
```

harflerinin hepsinin değişik rakamlara karşılık geldiğini bilerek şifreyi çözünüz.

III. Problemin Alt-Problemlere İndirgenmesi

1. Tanım: VE-VEYA Grafiği . Problem indirgeme metodu, baştaki problemin, çözümün dolaysız ya da olanaksız olduğu bir alt problemler kümesine dönüştürülmesinden oluşur. Bu terminal problemler üç türdedir:

(I, {o}, B): Bir tek operatör uygulanabilir, çözümleme dolaysızdır:

(I, O, B), I ∩ B ile boş kümeden ayrılır: Problemin çözümü anlatımındadır;

(I, { }, B) ile I ∩ B = Ø (*boş küme*): Problem çözümsüzdür.

I-5'de görüldüğü gibi bu ayrıştırma bir ağaca uygun düşen VE-VEYA grafiği biçiminde dile getirilebilir. Kök, başlangıç problemine, yapraklar ise terminal problemlere karşılık gelir.

2. Örnekler -

a) *İntegral hesabı (I-2e):* Basit olan integral hesabı örneğini tekrar ele alalım:

$$\int 2 \sin(x) \cos(x)\, dx.$$

Bu problem değişik yollarla çözülebilir: Bu durumda çözüm ağacının ilk gelişmesi bir VEYA bağı ile ifade edilir. Ya da dolaysız bir trigonometrik de-

ğişim uygulanılır ve yeni bir problemle karşılaşılır:

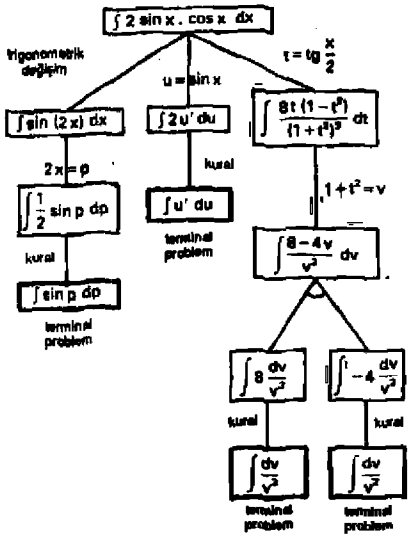
$$\int \sin(2x)\,dx,$$

ya da sin (x) = u değişkeninin değişikliğine karşılık gelen integralin $\int 2u'\,du$ formu saptanır,

ya da her zaman çözüme götürebilecek olan t = tg (x/2) değişkeninin değiştirilmesi uygulanır.

Bu son alternatifte, elde edilen rasyonel kesirin ayrıştırılması problemi bir VE bağına ayrıştırmaya götürür.

Şekil 6'daki VE-VEYA ağacı bu görüşleri toparlar.



Şekil 6.- İntegral hesabında VE-VEYA ağacı

b) *Hanoi Kuleleri (I-2i).*- Hanoi kuleleri problemi problem indirgeme metodu kullanılarak çözülebilir. Bu örnekte ayrıştırma seçeneklerinin çok olduğu ve çözüm ağacının boyunu kısaltmak için probleme bağlı "heuristique" teknikleri kullanmanın gerekli olduğu görülmektedir.

3. Bir Çözümü Araştırma Teknikleri - Burada, daha önce görülmüş tekniklere benzeyen kör teknikler üzerinde vakit kaybedilmeyecektir.

Bu, problem indirgeme bakış açısında A*'nın eşdeğerlisi AO* algoritmasıdır, ki bu h(n)'nin bazı özellikleri doğruladığı VE-VEYA grafiklerinde bir h(n) fiyat fonksiyonuna bağlı olan "heuristique" araştırmadan ileri gelir. İki temel operasyonun tekrarından oluşur:

-Kısmi en iyi çözüm grafiğini bulma olanağı sağlayan aşağıya doğru inen *(top-down)* bir grafik oluşturma işlemi;

-Geliştirilmekte olan bağın ardıllarının yeni hesaplanmış olan değerlerinden başlayarak değerleri aşağıdan yukarıya doğru gözden geçirme süreci. Gözden geçirilen bu değer böylece yükselerek yayılır.

H benzer şekilde "nül" olduğu için, enlemesine bir parkura ulaşılır.

A ve A* algoritmaları gibi, AO* da, daha önceden görüldüğü gibi, örneğin ayıklama veya grafiğin bazı bölgeleri üzerinde odaklaşma teknikleri sayesinde özel durumlara daha etkili olarak uygulanabilmesi için değişikliğe uğratılabilir.

4. Durumların Aralıkta Gösterilmeleri İle Bağlantı - Bir problemin çözümlenme gelişmesini göstermenin iki yolu arasındaki fark problemin anlatımı karşısında iki usavurma modelini ortaya çıkarıyor. Çözümleme ağacının geliştirme operatörleri kavramları değişiktir: İlkinde problemin bir du-

rumunu bir başka duruma çeviren operatörler sözkonusudur, ikincide bir problemi çözülecek yeni alt-problemlere çeviren ayrıştırma operatörleri sözkonusudur. Böylece aynı problem iki yoldan çözülebilir. Örneğin bir gösterme şeklinden diğerine geçişin mümkün olduğu kanıtlanabilir.

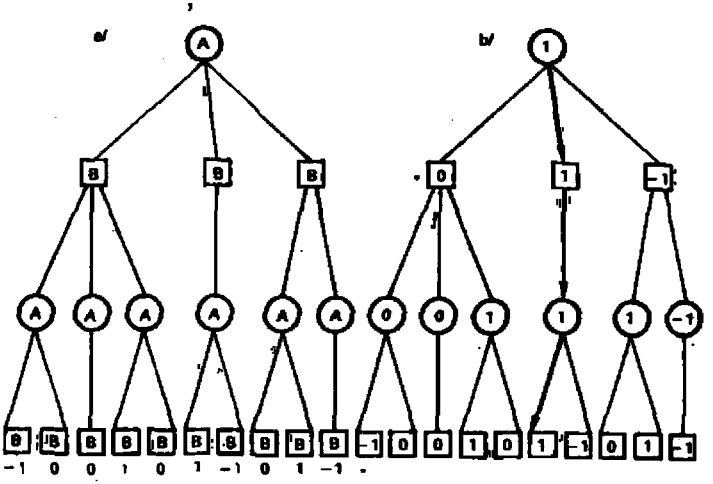
Verilen bir problem karşısındaki seçim, bir kısmı sezgisel özelliğe sahip olabilen ve diğerleri ise operatörlerin tanımlanması sorularına, değer fonksiyonları seçimine, algoritma karmaşıklığına vb.'ne bağlı olan değişik faktörlere dayanacaktır.

IV. Oyun Ağaçlarının Özel Durumu

1. Gösterme - Birbirlerinin konumlarını (satranç, dama vb.'de olduğu gibi) ve neler yapabileceklerini bilen A ve B rakiplerinin sırayla oynadıkları bir durumla karşı karşıyayız.

Verili bir durumdan yola çıkarak A oyuncusu için olası olan oyun, örneğin "oyun ağacı" denilen bir ağaç şeklinde geliştirilebilir. A yapabileceği hamleleri B'nin olası karşılıklarını ve bunlara vereceği karşılıkları gözönünde bulundurur ve bunlar seçimin kişiye bağlı olduğu bir derinlik seviyesine kadar uzatılabilirler. A'nın ve B'nin oynaması gereken durumları belirtmek için VE bağları ve VEYA bağları karşılıklı olarak nitelendirilir ve yanlış bir şekilde VE-VEYA ağacından sözedilir.

Örneğin şekil 7a'daki ağaç, oyun sırası A'da iken onun bir durumdaki (ağacın kökü) "düşünce"sini yansıtıyor. Yapraklar gelecekteki olası durumları göstermektedirler, eğer A ve B sırayla bu yönde hareket ederlerse. Karmaşık bir oyun sırasında bu konumlar nadiren oyunun son durumlarıdır. Oyuncunun bu durumların zafere götürme ihtimallerini tahmin etmesi sözkonusudur.



Şekil 7. Oyun ağacı ve "minimax" algoritma

2. Yapılacak Hamleyi Seçme Tekniği - Yapılacak en iyi hamleyi belirlemek için A, tabi ki "heuristique" yapılı olabilecek ölçütlere uygun olarak yaprak-bağların "statik" bir değerlendirmesini yapacaktır. Bu değerlendirmeden itibaren kararın algoritması, aşağıdaki kipliğe (modalité) göre ta köke kadar ağacın her bağına bir değer verilmesinden oluşur (A için en iyi hamleyi göstererek).

- A'nın, ardıllarına önceden verilmiş değerlerin azamisini oynamak zorunda olduğu bir bağa etki edilir;

- B'nin, ardıllarına daha önceden verilmiş değerlerin asgarisini oynamak zorunda olduğu bir bağa etki edilir.

Bu metod "minimax" algoritma olarak adlandırılır. Bu işlem için kök-bağı şekil 7b'de gösterildiği gibi yapraklardan birine verilmiş olan bir değerin

aşağıdan yukarıya doğru yayılmasından doğan sayısal bir değerden etkilenmiş olur.

Bu defa kökten beri hesaba katılan bu değerin izlediği yol, B'nin aynı stratejiyi uyguladığı bilinerek, A için en yararlı ya da en az zararlı hamlelerin ardarda gelmesine tekabül eder.

Bu teknikte, ağaçta yukarıya doğru çıkarken konumlara dinamik olarak etki ettirilen değerlerin konumlara doğrudan doğruya statik değerlendirme ile etki ettirilecek değerlerden daha çok bilgi kapsadıkları varsayımı uygulanmaktadır.

"Minimax" prosedür, araştırma ağacının gelişme sürecini (yetkili operatörlerin uygulanması) değerlendirme sürecinden tamamen ayırır. Canlı oyuncuya yaklaşmanın bir yolu bazı durumları hemen ekarte etmek, değişik kurallara bağlı olarak başka durumlar üzerinde yoğunlaşmak olacaktır.

Bu yol yukarı doğru çıkma sırasında katsayıların bağıntılarını hesaba katarak hızlandırılabilir ki bu ağacın her dalından geçmeden başlangıç durumunu değerlendirme olanağı sağlar. Bu "alfa-beta" algoritmasından çıkma yoludur.

ÜÇÜNCÜ BÖLÜM
BİLGİLERİN TANITIMI VE USAVURMA

I. Giriş

Buradaki bilgiden insanın bütün bilgi türleri kastediliyor: gerçek dünyayı oluşturan nesneler, olgular ve olaylar, temel nesnelerin gruplandırılmaları ve genelleştirilmelerine karşılık gelen daha geniş kavramlar, kavramlararası ilişkiler, "heuristique"-ler ve yapma-bilme stratejileri ve hatta usavurma yöntemleri. Buna, bilgişine duyulan güvene ve kendine özgü düşünme biçiminden elde edilen görüntüye bağlı "metabilgi" eklenir: "bilgiye ait bilgiden" de bahsediliyor.

Bilgi bir probleme özgü, bir alanda genelleşmiş, derin, yüzeysel, kesin, şüpheli, belirsiz, eksik vb. olabilir.

Bilgilerin tanıtımı sorunu onların bir usavurma sistemi tarafından kullanılabilecek sembolik bir biçime transkripsiyonlarıdır.

Bir tanıtma biçimi içiçe girmiş, hatta birbirine karışmış iki yönü birleştirir:

- bilgiyi tanıtmak için verilerin yapısı;
- bu enformasyonu kullanmanın ya da usavurmanın ortak metodu. Usavurma mekanizması işlenen problemle ilgili yeni enformasyonları ve bilgilerin dinamik olarak keşfedilmesini sağlar. Bu ancak açıklıkla sıralanmış enformasyonların hesaba katılabildikleri bir *bilgi tabanını* klasik veri tabanından net olarak ayrır.

Bazen deklaratif bilgiler ve yöntemsel bilgiler öne sürülmüştür. Bu ikicilik, bilginin deklaratif sunuluşu ("ne?" sorusuna bir cevap veren) ile "nasıl yapmalı" türünde bir bilgi sunuluşu arasındaki ayırımı ifade eder. Görüleceği gibi günümüzdeki eğilim bir bilginin spesifikasyonu ile kullanma mekanizmalarını aynı biçimsellik içinde birleştiren karma sunuluşlardır.

Aşağıdaki bölümler önemli tanıtma biçimlerini incelemektedir.

II. Mantıksal Tanıtımlar

Formel mantık alanındaki teorik gelişmelerden doğan bu tanıtımlar YZ'nin ilk günlerine dek uzanır. Deklaratif sunuluşun ilk örneği olarak özellikle matematik mantığını içerirler (önermeler mantığı ve birinci derecedeki yüklemler mantığı). Ama standart olmayan başka mantıklar da aynı şekilde kullanılmıştır.

1. Önermeler Mantığı -

a) *Bağıntılı Mantık Formülleri.*- Önerme terimi belli bir sentaksa göre formüle edilmiş ve verili bir alanda "doğru" ya da "yanlış" değeri alabilen iddia düşüncesini kapsar.

O halde ŞARAP (BORDEAUX), ŞARAP'ın "yüklem" olduğu ve BORDEAUX'nun onun "kanıtı" olduğu bir ilksel iddia (veya "atomik formül") örneğidir. Eğer bu iddia "bordeaux bir şaraptır" şeklinde sunulursa her zaman doğrudur.

Bir yüklemin birçok kanıtı olabilir; "Morgon'un kırmızı bir etiketi var" şeklinde yorumlanabilecek ETİKET (MORGON, KIRMIZI) da olduğu gibi.

Bu tür ilksel iddialar (— ile gösterilir) "olumsuzluk" operatörü ile reddedilebilirler ve bileşik mantık formüllerini (iyi şekillenmiş formüller de

denilir) oluşturmak için VE ($\wedge$), VEYA ($\vee$) mantık bileştirenleri, mantık içermesi ($\Rightarrow$) ya da denklik ($\Leftrightarrow$) sayesinde aralarında birleştirilirler.

Ayrıca formüller işlenen alanın elemanlarının alanın diğer elemanlarına uygulanması olan "fonksiyonları" kapsar. Örneğin değerin MÂCON olduğu SERMAYE (MÂCON'un) örneğinde SERMAYE bir fonksiyon ayırdedicidir.

İyi şekillenmiş bir mantık formülü bir bilgi parçasının sunuluşunu sağlar (doğru veya yanlış).

b) *Önermeler Mantığının Yetersizliği.*- Önermeler mantığı kendisinden gerçek dünyanın elemanlar bütünü için geçerli özellikler çıkarsanmak istenildiğinde yetersiz kalır. O halde bütün Sauternes şaraplarının beyaz şarap olduğunu söylemek ancak bütün Sauternes'leri ayrıntılayarak ve her birinin beyaz şarap olduğunu açıklayarak mümkün olabilir.

Birinci derecedeki yüklemler mantığı daha genel iddiaların gücünü kesinleştirme ve bu tür bilgileri ifade etme yollarını doğurur.

2. Birinci Derecedeki Yüklemler Mantığı -

a) *Giriş.*- Birinci derecede yüklemler mantığı, önermeler mantığının bir uzantısı gibi görülebilir. Önermeler mantığının temel özellikleriyle donatılmış olarak, ayrıca, "değişkenler" olarak adlandırılan genel elemanların girişini sağlar. Genellikle u,...v,...,z, diye kaydedilen bu değişkenler, evrensel kantifikatör $\forall$ (ne olursa olsun) ile ya da egzistansiyel kantifikatör $\exists$ (var) ile gösterilebilirler.

Şu halde ($\exists$ x) P(x), x'in doğru olduğu alan için, en azından bir x elemanı (ya da kişisi) olduğunu gösterir.

Öte yandan, ($\forall$ y) Q(y), Q tanım alanının bütün y'leri için, Q(y)'nin doğru olduğunu gösterir.

Yüklemler mantığının iyi şekillendirilmiş for-

mülleri, önceden olduğu gibi, verili bir yorum için doğru ya da yanlış değerlerini alırlar.

Açıklamalar:

1) Birinci derece kavramı, değişkenlerin kantifiye edilebileceği ancak ne yüklemlerin ne de fonksiyonların kantifiye edilemeyeceği olgusuna bağlanmaktadır.

2) Kantifiye edilmiş bir formülün değerini çözümlemek için genel bir metod bulmanın olanaksız olduğu kanıtlanır: Birinci derecedeki yüklemler mantığı "çözümlenemez"dir.

b) *İyi Şekillendirilmiş Formül Örnekleri:*

(∃y) (PAPA_NIN_YENİŞATOSU (y)∧ ETİKET (y, BEYAZ)

(∀x) (BOURGUEIL (x) ⇒ ETİKET (x, KIRMIZI)

(∀x) (CRUSTACE (x) ∧ ŞARAP (u) ∧ETİKET (u, KIRMIZI) ⇒ SALIK VERMEMEK (u, x))

c) *Mantıksal Usavurma.-* Formel mantık biçimselliğinde usavurmak, varolan bir formüller kümesinden yola çıkarak yeni formüller kanıtlama anlamına gelir. Bunun için mantıkçılar tarafından ortaya konulan bir kurallar kümesi düzenlenir. Şöyle ki:

- P ve P⇒Q formüllerinden başlayarak Q'yü türetme olanağı sağlayan *"modus ponens" kuralı.*

Bir "instanciation"un, yani bir değerin yerine bir sabit ya da bir başka değişken koyma operasyonunun gerçekleştirilmesinde yararlı olabilir. Bu operasyon, örneğin iki formülün *modus ponens* ile işlenebileceğinin anlaşılmasında yararlıdır.

İki formül şunlardır:

ARBOİS (y) ⇒ bölge (y, JURA) ve

ARBOİS (PUPILLIN),

şu sonuç çıkarılabilir;

BÖLGE (PUPILLIN, JURA)

çünkü PUPILLIN sabitinin y değişkeninin yerini

alması ARBOIS (y) ve ARBOIS (PUPILLIN)'in uyumlu hale gelmesini sağlıyor.

Uyumlu hale getirme mekanizması "birleşme" ya da denetleme olarak adlandırılır. Geneldir ve değişkenleri hesaba katan bütün bilgi tanıtma modellerinde bulunur;

- $P \Rightarrow Q$ ve $\neg Q$'dan itibaren $\neg P$'yi türetme olanağı sağlayan *"modus tollens" kuralı;*

-*Çözümleme Prensibi:* "Şartlar" biçiminde ifade edilen formüllere uygulanan bir çıkarsama kuralıdır. Çözümleme prensibi matematikçi mantıkçıların, özellikle de Herbrand'ın çalışmalarının ürünüdür. 1965'de Robinson tarafından enformatik işleme uygulanmıştır. Bu kural iki benzer-şarttan yeni bir şart (çözücü denilir) çıkarma olanağı sağlar. Bu yeni şart aşağıdaki paragrafta incelenecektir.

Saptama: Yukarıda belirtilen üç kural aynı zamanda önermeler mantığına da uygulanır,

- *tümel uzmanlaşma:* $(\forall x)$ P(x) özel bir A örneği için P(A)'yı belirtir.

d) *Çözümleme Prensibi.*

- Şart kavramı: Bir şart atomik formüllerin ayrımı (yani VEYA'larla birleşmesi) ile oluşturulur.

Örneğin:

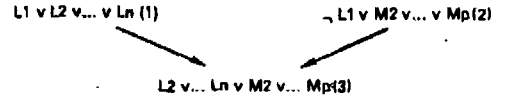
$\neg$COTESDETOUL (x) V ETİKET (x, KIRMIZI) V ETİKET (x, GRİ) bir şarttır.

İyi şekillenmiş bütün formüllerin bir şartlar kümesine dönüştürülebilecekleri kanıtlanmaktadır (Skolem algoritması). Bir önceki şart da mantık formülün dönüştürülmesinden elde edilmiştir: $(\forall x)$ (COTESDETOUL (x) $\Rightarrow$ ETİKET (x, KIRMIZI) V ETİKET (x, GRİ). Başlangıç formülü gibi, şartlar da değişkenleri kapsayabilirler.

- Çözümleme metodu:

Bu metod ilk olarak aynı sözcüğün (littéral) bir kere negatif bir kere pozitif olarak yeraldığı iki

farklı şartın (1 ve 2) araştırılmasından, daha sonra da bu ikişinden bir çözücü şart (3) çıkarılmasından oluşur. Çözücü şart, mekanizmayı başlatan iki sözcük bir kere çıkarılmak kaydıyla iki benzer şartı meydana getiren bütün sözcüklerin birleşimi ile oluşur.



- Çürütme yolu ile mantıksal sonuç çıkarma sistemleri.

Çözümleme, özellikle, belli bir X mantık formülünün diğer mantık formüllerinden oluşan bir FL kümesinin mantıksal bir sonucu olduğunu kanıtlamak için uygulanır.

Çürütme yoluyla kanıtlama metodu soyut usavurmaya benzer. Kanıtlanacak formül ya da teoremin "yadsınması"ndan oluşur; yani karşıtı —¬ X'in doğru olarak varsayılmasından, —¬ X'in, C = FL∪ (—¬ X) kümesini oluşturacak şekilde FL kümesine katılmasından ve çözümleme mekanizmasının tekrara dayalı bir şekilde gözönüne sunulmasından oluşur. Bu yöntemle yavaş yavaş C'nin bütün mantıksal sonuçlarından oluşan R(C) kümesi oluşturulur.

Eğer C başlangıç kümesinde bir çelişki varsa çözümleme mekanizması belirli bir noktada aşağıda görüldüğü gibi YANLIŞ değerli "NIL" şartını çıkaracağı iki çelişkili şartla karşılaşır:



Bu durumda FL ∧ (— x) kümesinin çelişkili olduğu ve X'in FL kümesininmantıksal bir sonucu olduğu kanıtlanacaktır.

3. Formel Mantığın Avantajları ve Sakıncaları - Formel mantık sağlam teorik temeller üzerine kuruludur ve birçok durumda olguları ve olgular arasındaki tümdengelime ait ilişkileri ifade etmenin doğal yolu gibi görünür. Teorem ispatı, bilmece çözümü vb. gibi eksiksiz verilere sahip olan kesin usavuruma, tamamıyla uyarlanmıştır.

Bu arada, yüklemlerin DOĞRU ya da YANLIŞ ikili değerlerini aldığı ve bir daha hesaba katılmadığı bu biçimselliğin kesinliği ayrıntılı değerlendirmeler yapma olanağını vermez. Ayrıca, insanın gıyabi usavurma kapasitesinin tersine, eksik enformasyon olduğu zaman bu biçimsellik karar verme olanağı sağlamaz. Bu sınırlamaları az da olsa kaldırmak için başka mantıklar devreye sokulmuştur; özellikle de monoton olmayan mantıklar, çokdeğerli mantıklar ve kiplik (modalité) mantıkları (bölüm 5'de kısaca tanımlanırlar).

Öte yandan formel mantık, kullanılan bilgiler kümesinin organizasyonu olanaklarını üretmez ki bu, bu kümenin önemli olduğu durumlardaki kullanımı sırasında bir sakınca oluşturur. Kısacası yöntemsel ya da "heuristique" özellikle bilgileri sunmak oldukça zordur.

4. Mantığın Uygulanması - Değişik sistem ya da diller mantıksal usavurmanın içinde toplanırlar, örneğin:

a) STRIPS, hareket planlayıcısı (daha sonra görülecektir);

b) PROLOG, 1972'de Marsilya'da A. Colmerauer tarafından geliştirilmiş mantık programlama dilidir ve çok sayıda uygulaması vardır. Bu dilde bilgi "Horn şartları" ençok bir negatif sözcüklü şartlar

biçiminde sunulur. Şartlar:

- "olguları" şu şekilde çevirirler:

uyum (géromé, pinot_siyah)

"siyah pinot géromé ile uyum sağlar" şeklinde yorumlanabilir.

- veya "kuralları" şu şekilde çevirirler:

aynı bölge (X,Y): -peynir bölgesi (X,U),

şarap bölgesi (Y,U).

"X ve Y'nin her ikisi U bölgesinden geldiğine göre belli bir X peyniri belli bir Y şarabı ile aynı bölgedendir" anlamına gelebilir.

Sabitleri (géromé ya da siyah_pinot gibi) küçük harflerle, değişkenleri ise (X, Y veya U gibi) büyük harflerle yazacağız. Virgül mantıkta VE anlamını taşır.

<sol taraf> : - <sağ taraf>

türünde bir şart şöyle okunur: " <sol taraf>'ı kanıtlamak için sağ taraf 'ı kanıtlamak yeterlidir."

Bir PROLOG programı tam anlamıyla bir FL şartları dizisinden oluşur ve programın özel bir uygulaması için bu şartlara FL'nin mantıksal bir sonucu olup olmadığı öğrenilmek istenilen bir şart eklenir. Program geriye dönüşün olduğu derinlemesine çalışan "destek" adlı strateji tarafından kontrol edilen bir çözümleme mekanizması sayesinde uygulamaya "açıklanır".

Örnekte peynirler ve şaraplar alanına girelim ve diyelim ki:

uyum (X,Y)

X bir peynir ve Y bir şarapken X ve Y'nin aralarında uyum olduğunu ifade ediyor.

Ve aşağıdaki PROLOG programını önerelim (burada sonraki açıklamanın daha iyi anlaşılması için şartlar numaralandırılmıştır):

(1) uyum (géromé, pinot_siyah).

(2) uyum (roquefort, châteauneuf-du-pape).

(3) uyum (laguiole, cahors)
(4) uyum (X,Y): -keçi_peyniri (X),
aynı bölge (X,Y).
(5) uyum (X,Y): pâte-molle (X), etiket (Y, kırmızı),
gövde (Y, iri)
(6) uyum (chaource, Z): -şarap-şampanya-yapısı (Z).
(7) şarap-şampanya-yapısı (bouzy).
(8) aynı bölge (X,Y): -peynir bölgesi (X,U),
şarap bölgesi (Y,U).
(9) şarap bölgesi (chassagne, bourgogne).
(10) keçi peyniri (charolles).
(11) keçi peyniri (pelardon).
(12) peynir bölgesi (charolles, bourgogne).
(13) pâte-molle (camembert).
(14) pâte-molle (géromé).
(15) etiket (gewurtztraminer, beyaz).
(16) etiket (morgon, kırmızı).
(17) etiket (çiçekli, kırmızı).
(18) gövde (morgon, iri).
(19) gövde (çiçekli, ince).

Bu program bir kere hafızaya alındıktan sonra, uygulamasına aşağıdaki sorgulama ile başlayalım:

"camembert ile hangi şarap servis edilir"

uyum (camembert, L)? şeklinde yazılacaktır. burada L herhangi bir değişkeni ifade ediyor.

PROLOG yorumcusu, soru şartını tatmin edecek L'yi bulmaya çalışırken programın şartlarını tarayacak ve cevaplayacaktır:

L = morgon

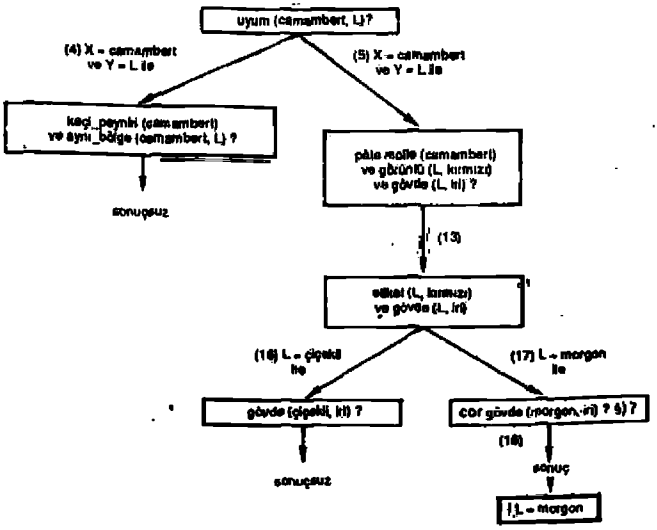
Şekil 8'deki şema bu usavurumu gösteriyor.

pişirilmiş pâte (F)?

veya

uyum (M, gewurtztraminer)?

sorularına yanıt vermeyecektir. Bu sorular yetki alanının dışındadır, ilkini pişirilmiş pâte yüklemini tanımadığı için, ikincisini gewurtstraminer ile uyu-



Şekil 8.- PROLOG'da çözümleme örneği.

şan peyniri tanımadığı ve usavurma ile bunu bulmayı başaramadığı için çözemiyor.

Son olarak keçi peyniri ile hangi şarabın servis edileceğini yeniden araştıralım. Bunun için programın uygulaması:

uyum (V,W), keçi_peyniri (V)?

ile başlatılır.

Verilen cevap şudur:

V = charolles W = chassagne.

Pélardon'un bir keçi peyniri olduğunu bilmesine karşın sistemin ona uygun bir şarap ismi veremediği görülecektir. Yanıt bu peynirden sözetmemektedir.

Başka durumlarda örneğin

etiket (X, kırmızı)?

ile bütün kırmızı şarapların araştırılmasında sistem çoğul bir yanıt verecektir:

X = morgon

X = çiçekli

c) Mantıksal çıkarsama tekniklerinin, tabanda açıkça yer almayan bir enformasyonu çıkarma olanağı sağlayan veri tabanlı dilek şeklinde ifade edilen bir problemin çözümlenmesinde, bir uygulama bulduğu tümdengelen veri tabanları. Günümüzün farklı modelleri, büyük miktarlardaki enformasyonların tek tipte şekillendirilmiş kullanımına uygulanan klasik bir veri tabanı yönetim sistemi (SGBD) ile PROLOG türünde bir tümdengelim mekanizması arasında az çok güçlü bir birleşmeye tekabül ederler.

5. Diğer Mantıklar - Mantığın kullanım alanını yaygınlaştırmak amacıyla yeni biçimsellikler araştırılmıştır. Özellikle aşağıdakileri belirtelim:

a) *Çokdeğerli mantıklar* 2'den büyük n doğruluk değerleri sayısı ile bir yüklemler mantığı yayılmasını oluştururlar. Bu bilgilerin ifadesindeki anlam belirsizliğini ya da fluluğu açıklama olanağı verir. Eğer n sonsuza doğru uzatılırsa gerçeğe yakın katsayı kavramına ulaşılır;

b) *Kipsel (modale) mantıklar* sezgisel içerme kavramını daha iyi belirtmeye yararlar. Bunun için, bir mantık formülünü kesinleştirmeye yarayan birçok kiplik devreye sokulur (olabilir, gerekli, olumsal ve imkansız gibi). Örneğin kat'i içerme:

Ancak ve ancak $\neg$ Poss ($x \wedge \neg y$) ise $x \Rightarrow y$, ancak ve ancak x ve $\neg$y'ye aynı anda sahip olmak imkansızsa x'in kat'i olarak y'yi kapsadığı anlamına gelecektir.

c) Usavurmanın belli bir anında doğru olan bir iddianın daha sonra yanlış olarak ortaya çıkabileceği *monoton olmayan mantıklar*.

Bu tür bir mantık özellikle gıyabi usavurmayı dile getirme olanağını sağlar: Tersi kanıtlanmadığı sürece, Champagne'dan gelen bir V şarabı verili olduğunda (buradaki büyük harf özel bir şarabı yani bir sabiti gösteriyor) aşağıdaki iddiaya göre bunun köpüklü şarap olduğu varsayılacaktır:

(∀x) (ŞARAPBÖLGESİ (x, CHAMPAGNE) ⇒
YAPI (x, KÖPÜKLÜ))

Usavurma sırasında, bahsedilen V şarabının "Champagne şarabı" ismini taşıdığı anlaşılırsa ve aşağıdakiler biliniyorsa:

(∀x) (CHAMPAGNE-ŞARABI (x) ⇒
ŞARAPBÖLGESİ (x, CHAMPAGNE))

(∀x) (CHAMPAGNE ŞARABI (X) ⇒
YAPI (x, KÖPÜKSÜZ)),

doğru olduğu varsayılan bu bilgiden türeyebilen bütün mantıksal sonuçlar gibi "V köpüklüdür"ün de doğruluğu ortaya konulur.

Bu mantığa dayalı bazı diller, planlar geliştirmeye yönelik PLANNER gibi, aşağıdaki türden kuralların öne sürülme olanağını sağlıyorlar:

"Eğer A belli bir örnek kategorisine aitse ve ¬P(A) doğru olarak tanınmamışsa, o halde P(A) varsayılır".

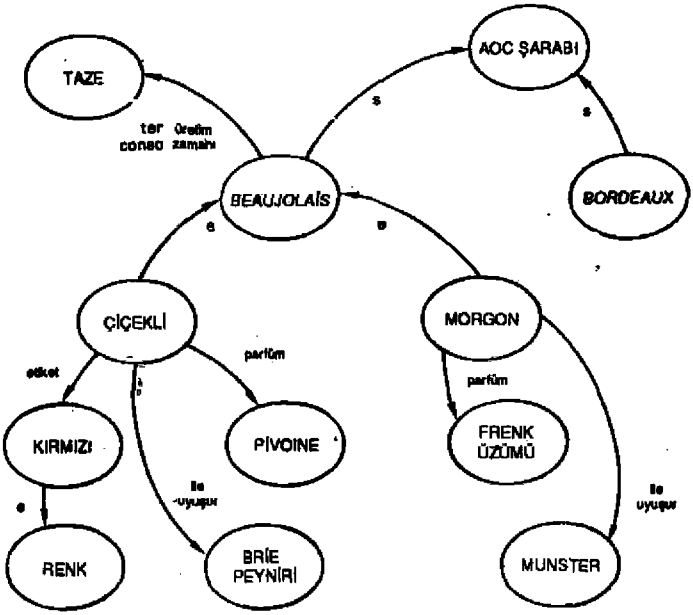
III. Anlambilimsel Şebekeler

1. Giriş - Ortak şebekeli hafıza düzenleme çalışmalarının sonucu olan anlambilimsel şebekeler ilk başta anadilin açıklanmasında uygulanmıştır. Anlambilimsel şebeke, bağların kavramları, yayların ise anlambilimsel yapılı ilişkileri (bunlar kavramlar arasındaki anlama bağlıdırlar) gösterdiği etiketli bir grafiktir.

Bağlantılar, jenerik kavramlar (s bağı, "türünde") arasındaki, bu kavramlar ile özel örnekler (e

bağı, "bir-dir") arasındaki ilişkileri ya da bir eylemi gösteren fiile dolaylı tümleçleri bağlayanlar gibi diğer ilişkileri ortaya çıkarabilirler. Şekil 9 anlambilimsel bir şebeke örneğini gösteriyor.

2. Anlambilimsel Şebekelerde Usavurma - Usavurmak, yapısal uyum sağlama mekanizması tarafından bir bilgi şebekesine konulan problemi gösteren bir şebekeyi taklit etmeye başlamaktır. Uyum nadiren kaçınılmazdır, ayrıca özelliklerin kalıtımını ortaya koymak zorunludur.



Şekil 9.- Anlambilimsel şebeke örneği

Aslında bazı bağlantı türleri, genelden özele tüm geçişler gibi, bazı bağlara, bağlı oldukları bağların özelliklerinin geçmesini sağlarlar. Önceki ör-

nekte *çiçekli beuajolais*'ye bağlı bütün özellikleri ve *kontrollü köken belirleme şarabının* özelliklerini miras alır.

Tabii ki bu tür kalıtım kuralları kullanıldığında önlemler alınmalıdır.

3. Uygulamalar - Anlambilimsel şebekeler, kavramlar arasındaki düşünce birliklerini ya da eylem fiili etrafında merkezleşmiş eylem kavramlarını açıklayan bilgileri sunmak için anadilin kavranması sistemlerinde kullanılırlar.

Önceki örneğin bir bölümünün de gösterdiği gibi kavramlar arasındaki bağlantıların hiyerarşik bir yapıda olduğu durumlarda bilgilerin sınıflandırılmasını göstermeye elverişlidirler. 1979'da bir coğrafi sit üzerinde verili bir maden filizinin varolma olasılığını değerlendirmek için Birleşik Devletler'de geliştirilmiş olan uzman PROSPECTOR sisteminde jeolojik bilgi anlambilimsel bir şebeke biçiminde gösterilmiştir.

4. Avantajlar ve Sakıncalar - Anlambilimsel şebekeler hiyerarşik kavramlar kümesinin tanıtımına iyi uyarlanmışlardır. Özelliklerin kalıtım mekanizması kaba bir usavurma biçimini sağlar. Bu düzeydeki sınırlanmalar bir şebekeyi tanıtma modeli haline getirirler ki bu model başka modellerin tümleci olarak kullanılmıştır.

Ayrıca, standartlaştırma çabalarına rağmen,e özellikle ilişki türlerinde, standart terminoloji alanında şimdilik yer almaz. Son olarak, yönteme dair bilgileri ifade etmek mümkün değildir. Bu sınırlama aşağıda incelenecek olan nesne şebekelerinde hiç yoktur.

IV. ProdüksiyonKuralları

1. Giriş - Bir prodüksiyon kuralı

eğer koşullar *şu halde* sonuçlar (katsayı)

eğer öncüller (koşullar bölümü) doğrulanmışsa sonuçlar çıkarılabileceğini (ya da bir eylemin, bir hesap yönteminin ya da afişlemenin başlatılabileceğini) gösteren biçim bilgisinin bir parçasıdır.

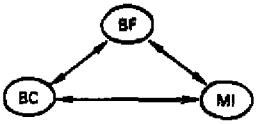
Bu örnekte kurala, ona duyulan güveni ya da hatta bu bilginin gerçekliğini gösteren bir katsayı eşlik etmektedir.

Çoğunlukla uzman sistemler bilgi tabanlarında bu biçimsellikte sunulan bir bölüme sahiptirler.

Kurallara etki ettirilen katsayılar "şüpheli" usavurma olanağı sağlarlar ve genellikle bilgi tabanının temelinde olan uzmanın deneyiminin sonucudurlar.

2. Kurallı Sistemin Mimarisi - Kurallar biçiminde ifade edilen bilgi salt deklaratif özelliktedir. Bilgi ve bilginin kullanılma mekanizması (yani usavurma) arasında tam bir ayrım vardır.

Böyle mekanizmaların büyük bir çeşitliliği olmasına karşın bunlar "çıkarsama motoru" veya "yorumlayıcı" jenerik terimi ile yeniden gruplandırılırlar.



Şekil 10. Prodüksiyon kurallı bir sistemin mimarisi

Şekil 10 prodüksiyon kurallı bir sistemin mantıksal mimarisini şematize eder.

Kurallardan oluşan bilgi tabanı BC sistemin uzun vadeli hafızası anlamına gelir. Belli bir süreklilik özelliği olan verili bir alan üzerindeki bilgileri kapsar.

Çıkarsama motoru MI, verili bir başvuru duru-

munda bilgi tabanını manipüle etmekle yükümlü algoritmik mekanizmadır.

Olgular tabanı BF, sözkonusu problem üzerindeki bilinen enformasyonları sürekli olarak kapsayan kısa vadeli bir hafızaya tekabül eder. Başvurunun başında işlenen problem hakkındaki hazırlık verilerini kapsar. Sonra BC'deki kuralların uygulanması ile veya problemin çözümüne ulaşılana kadar sistemin sorduğu olası sorulara kullanıcının verdiği yanıtlar sayesinde zenginleşir. Aynı kurallar gibi olgular da gerçeğe uygunluk katsayıları ile dolu olabilir.

3. Usavurma - Yorumlayıcı, karma bir kayıtta birleştirilebilen ileri zincirleme ve geri zincirleme şeklindeki iki taban mekanizmasına göre çalışır. Aşağıdaki c) bölümünde incelenen bir örnek bu mekanizmaları gösterecektir.

a) *İleri zincirleme* veriler tarafından yönlendirilen bir usavurmadır. Kuralları *şartlar → sonuçlar* yani *sol→sağ* yönünde gözden geçirmekten oluşur.

Usavurmanın verili bir etabında çıkarsama motoru üç aşamalı bir devir izler:

1. Öncül bölümün, olgular tabanında bilinen elemanlardan yola çıkılarak doğrulandığı kuralların ortaya çıkarılması

Bu iş için motor, kuralların değişkenleri kapsadığı durumda mantıksal tanıtım konusunda daha önceden çağrıştırılmış denetleme mekanizmasını devreye sokmalıdır. Motora 1. derece denir (mantık terminolojisine uygun olarak). Böyle bir kural değişkenlerin farkla istekleri için birçok defa kullanılır.

Bir değişkene ancak bir sabit ikame edilebildiği için (ve başka bir değişken değil) 0+ dereceden sözedilir.

Buna karşılık kabul edilen kurallar değişkenle-

ri kapsamadığı için 0 dereceden bahsedilir. Bu durumda bir kural, uygulandığı anda "öldürülür" çünkü yeni bir uygulama yeni enformasyonlar getiremez.

2. Seçenekler Arasından Bir Kuralın Seçimi.- Bu etap genel olarak "karşıtlıkların çözümlenmesi" diye adlandırılır.

Bu seçim çeşitli "heuristique" ölçütlere göre yapılabilir, örneğin şunlar kullanılarak;

- rastgele çıkarılan bir kural;
- en özgül kural (yani en büyük öncüller bölümüne sahip olan);
- o ana kadar hiç hesaba katılmamış ya da en son kullanılmış olan bir kural (değişkenli kurallar durumu);
- olgular tabanında yeni ortaya çıkan bir olguyu devreye sokan bir kural;
- en yüksek gerçeğe uygunluk katsayısına sahip olan kural;
- paralel olarak uygulanacak olan bütün kural seçenekleri vs.

Bu seçimi, içerikleri ile kurallara gönderme yapan bir üst düzeyin kuralları ile yönetmek de mümkündür. Bu durumda *metakurallar*dan sözedilmektedir.

Şöyle ki:

Eğer koşul 1 ise o halde ....'den bahseden kurallar üzerinde yoğunlaşmak

veya

Eğer koşul 2 ise o halde ....'den bahseden kuralları elemek,

metakural örnekleridir.

3. Seçilmiş Kuralın Uygulanması: Sonuçlar bölümünün kapsamındaki elemanlar BF'yi zenginleştirecektir. Bir eylem sonuçta belirtildiği zaman bu eylem başlamıştır.

Devir, hiçbir kuralın uygulanamadığı (olgular tabanının "doyurulduğu" söylenir) veya önceden saptanmış bir hedefin kanıtlanması başarılana kadar devam edecektir.

b) Geri zincirleme bir hedef tarafından yönlendirilen usavurmadır. Kuralların *sonuçlar-koşullar* yönünde kullanılmasıdır.

Motor ilk başta boş olan bir yığına yerleştirilmiş, kanıtlanacak bir başlangıç hedefinden yola çıkarak, önceki işleyiş modelininkilere benzeyen üç aşamalı bir devri izleyerek çalışır.

1. Hedef üzerinde sona eren kuralların ortaya çıkarılması (eldeki yığının en üstü);

2. Gözönüne alınacak kuralın seçilmesi (karışıklıkların çözümlenmesi);

3. Ulaşılacak yeni alt-hedefler gibi öncüllerin elemanlarının incelenmesinden ibaret olan bu kuralın uygulanması. Eğer zaten BF'de varsalar (olası bir usayatkınlık katsayısı ile doğru ya da yanlış olarak bilinirler) işlem tamamlanmıştır. Yoksalar hedefler yığınına kaydedilirler.

Devir yeni hedef olarak eldeki yığının en üstünü işleyerek yeniden başlar, bu, yığın boşalana kadar (hedefe ulaşılmıştır) ya da hiçbir kural gözönüne alınacak durumda olmayana kadar sürer.

Geri zincirleme böylece problemlerin indirgenmesi tekniğini ortaya koyar.

c) Karma kayıt: Yukarıda tanımlanan iki mekanizma aynı sorunları yanıtlamaz.

Reel birçok problemde insan veriler tarafından yönlendirilen bir yöntemi (örneğin tıbbi teşhis alanında hasta hakkındaki enformasyonlardan yola çıkarak sonuçlar çıkarmak) ve hedefler tarafından yönlendirilen bir yöntemi (bir teşhis düşüncesini kanıtlamaya çalışmak) alternatifli olarak ortaya koyar. Bu, prodüksiyon kurallı sistemlerde bir kar-

ma kayıt mekanizması olarak yansır.

Örnek olarak, bir şarap tatma seansına katıldığımızı ve tanınacak bir beyaz şarabın bir sürahide bize sunulduğunu düşünelim. Beyaz şaraplar hakkındaki bilgi tabanı aşağıda bir özeti yeralan bir kurallar kümesinden oluşmaktadır (açıklamanın anlaşılır olması için kasten sınırlanmıştır):

R1: Eğer sarı ve likörümsü ise o halde beyaz Bordeaux şarabı (kesin)

R2: Eğer bourgogne ve sert ise o halde cote_de_beaune'dur (kesin)

R3: Eğer cote_de_beaune ise ve yumuşak değilse o halde charlemagne burgonyasıdır (olası)

R4: Eğer cote_de_beaune ise ve yumuşak değilse o halde charlemagne burgonyasıdır (kesin)

R5: Eğer sek ise ve kehribar renginde değilse ve taze meyve kokulu ise o halde Alsace'dır (olası)

R6: Eğer sek ve güzel kokulu ise ve kehribar renginde değilse o halde bourgogne'dur (kesin).

Bu örnekte gerçeğe benzerlik katsayıları sembolik şekilde ifade edilmiştir. Sistemin amacı sunulan şarabın tanınmasıdır.

Tatma sonucundaki verilerden yola çıkarak bir konsültasyon ortaya koyalım. Şarap kehribar renginde değildir, güzel bir kokusu vardır, serttir, sektir ve yumuşaklığı yoktur.

Problemin bu verileri (kehribar renginde değil, güzel kokulu, sert, sek, yumuşak değil) olgu tabanına kaydedilir ve ileri zincirleme için ankraj noktaları olarak kullanılır:

-R6 uygulanabilecek tek kuraldır. R6'nın uygulanması BF'ye dahil olan bourgogne olgusunu (kesin) doğurur.

-Sistem ileri zincirleme yönünü izler. Bourgogne'un belirmesi olgusu ile artık R2'nin uygulanabileceğini ortaya çıkarır. R2 uygulanmıştır ve cote_

de_beaune (kesin) BF'ye dahildir.

-R3 şu anda seçenek olarak durur ve her zaman ileri zincirlemede olan sistem şu sonucu çıkarır: charlemagne-burgonyası (olası).

Bu usavurma durumunda olgu tabanı aşağıdakileri kapsar:

kehribar renginde değil (kesin)
güzel kokulu (kesin)
sert (kesin)
sek (kesin)
yumuşak değil (kesin)
burgonya (kesin)
cote_de-beaune (kesin)
charlemagne burgonyası (olası)

ve artık hiçbir kural uygulanamaz. İleri zincirleme süreci bloke olmuştur. Sistem ortaya çıkan verileri en iyi şekilde kullanmıştır, ancak sunulan beyaz şarabın tanınması belli bir ölçüde şüphe taşır.

Şarabın bir charlemagne_burgonyası olduğu yolundaki bu sezgiyi doğrulamak için, hedefi daha kesin bir şekilde kanıtlamayı denemek üzere motor geri zincirleme yönünde çalışacaktır.

-R4 kuralı charlemagne-burgonyası (kesin) sonucunu verir ve tek uygulanabilen kuraldır. Geriye uygulanması üç yeni hedefi ortaya çıkarır:

* cote_de_beaune (kesin) ve yumuşak değil (kesin) olgu tabanının bir parçasını oluşturur.

* çakmaktaşı tadı olgusu bilinmemektedir ve böylece ulaşılacak yeni hedef haline gelir. Sistem, çakmaktaşı tadı ile sonuçlanan kuralları araştırır ve bulamaz. Enformasyon elde etmenin tek yolu, örneğin soru sorarak, sistem kullanıcısını sorgulamaktır:

"Beyaz şarap çakmaktaşı tadında mıdır?"

-Kullanıcının "evet" dediğini varsayalım. Bu yeni bilgi olgu tabanı kapsamına eklenmiştir, bütün

R4 öncülleri doğrulanmıştır ve sistem incelenen beyaz şarabın bir charlemagne burgonyası olduğunu kesinlikle söyler.

4. Gerçeğe Uygunluk Katsayılarının Hesaba Katılması - Kurallara ve olgulara bağlı gerçeğe uygunluk katsayıları, usavurma sırasında nihai çözümün gerçeğe uygunluğu sağlanana kadar geliştirilmelidir. Farklı metodlar uygulanabilir.

Açıklama olarak katsayıları, usayatkınlık teorilerine (manipüle edilen katsayıların negatif değerler alabildiği durumda olasılık teorisinin bir uzantısı olarak görülebilir) uygun olarak işleme yöntemini vereceğiz. Bu, ilk uzman sistemlerden biri olan MYCIN'de kullanılan hesap tarzı yöntemidir.

Katsayılar -1'in "hayır olduğu kesin" ve +1'in "evet olduğu kesin" anlamına geldiği -1, +1 entervalinde değerlerini alırlar. Nül bir katsayı bilinmeyene karşılık gelir.

BC'nin R1 kuralını kapsadığını varsayalım:

eğer A ve B ise o halde C (0,7)

burada 0,7, A ve B kesin ise C sonucuna verilen değeri gösterir.

A ve B'nin, gerek önceki usavurmadan çıkarsanmış gerekse başlangıçta sağlanmış olan (örneğin belirşiz veriler) karşılıklı cA = 0,8 ve cB = 0,9 katsayıları ile önceden bilindiğini varsayalım: Aşağıdaki iki durumu inceleyelim:

a) C hâlâ bilinmiyor

O halde R1'in uygulaması şu sonucu verir:

- Öncüller bölümüne şu katsayının uygulanması

cPr = min (cA, cB) = 0,8,

- sonra C sonucuna şu katsayının uygulanması

cC = cPr * 0,7 = 0,56,

b) C 0,75 katsayısı ile BF'de daha önceden vardır.

0,75 (R1'in uygulanmasından önce) ve 0,56

(Rl'in uygulanması ile elde edilmiş) katsayılarının her ikisi de kesinliğe doğru yönelirler: C'nin gerçeğe benzerliğinin güçlendirildiği düşünülür. Aşağıdaki şekilde düzenlenirler:

cC = 0,75 + 0,56 - 0,75 * 0,56 = 0,89

Daha genel olarak, eğer iki farklı yoldan karşılıklı c1 ve c2 katsayılarıyla c sonucu çıkarılıyorsa aşağıdaki şekilde düzenlenirler:

c1 + c2 - c1 * c2 eğer c1 ve c2'nin her ikisi de pozitif ise

c1 + c2 + c1 * c2 eğer c1 ve c2'nin her ikisi de negatif ise

(c1 + c2) / (1-min (|c1| , |c2| eğer c1 * c2 < 0 ise

İki nül katsayı sıfır verir. Böyle bir sistemde -1 ve +1'in düzenlenmesi gerektiği durum (üçüncü formülün belirsizliği) bilgi tabanında anlambilimsel bir tutarsızlığı belirtir.

*5. **Avantajlar ve Sakıncalar*** - Kurallar, birinin diğerlerine başvurmadığı bağımsız bilgi parçalarıdır. Her kuralın uygulanma koşulu öncüller bölümünde yeralır.

Sonuçta, koşullara bağlı bir usavurmada ortaya konulan "heuristique" bilgiyi yansıtmanın basit ve doğal bir yolunu oluştururlar ve özellikle uzman sistemlerde yoğun olarak kullanılırlar.

Ayrıca çıkarsama motoru tarafından izlenilen yol, kullanıcıya sistemin izlediği usavurum hakkındaki açıklamaları sağlayacak açıklıkta belirtilir.

Sonuçta, katsayıların girişi kurallarda belirtilen yargıları dengeleme, belirsiz verileri hesaba katma ve saf bir ikili mantığa bağlı olmayan problemleri ele alma olanağı sağlar.

Ayrıca, kurallı bir sistem tamamen modüler olacaktır ve sonuçta kolaylıkla değiştirilecek veya tamamlanacaktır. Bununla beraber, bu bariz kolaylık ya programcı ya da geliştirilen otomatik bir sistem

tarafından kurallar kümesinin tutarlılığının kontrolüne ve devamlılığına gerek duyar.

Üstelik, büyük sistemlerde kuralları düzenlemek veya karışıklıkların çözümünde daha iyi bir verim sağlamak üzere kurallara öncelik tanımak gerekir, ki bu da zengin bilgi tabanlı artımlı (incrémentale) yapı argümanının tersine gider.

Nihayet, eğer kurallar bir uzman tarafından kullanılan "heuristique" "yüzey" bilgisini iyi tanıtırlarsa uzmanın dayandığı "derin" kavramlar hakkındaki bilgileri hesaba katma olanağı vermezler.

Sonuç olarak bilgi tabanlı sistemler prodüksiyon kurallarının zamanının çoğunluğunu kapsıyorlarsa genellikle diğer tanıtma modellerine bağlanılır.

V. Yöntemsel Tanıtımlar

1. Giriş - Biri deklaratif diğeri yöntemsel olan iki tanıtım modeli gerçekçi bir bilgi tabanında tamamen ayrıştırılamazlar, çünkü birbirini tamamlayıcı özelliklere sahiptirler. Genellikle *yöntemsel tanıtım* terimi ile klasik algoritma yöntemlerinin YZ'ye doğrudan girişi kastedilir.

2. Yöntemlerin Harekete Geçmesi - Bu harekete geçiş farklı modellere göre yapılabilir:

- Yöntemin devreye girmek zorunda olduğu anda doğrudan çağırma ile. Böylece, öncüllerin doğrulanmış olduğu bir prodüksiyon kuralının ileri zincirlemede uygulanması, kuralın sonucunda şekillenen bir yöntemin doğrudan harekete geçmesine yolaçabilir;
- bir değerin yükleme etki ettirilmesinin ortak bir yöntemin harekete geçirilmesiyle sonuçlanabildiği "frames" durumlarında olduğu gibi yöntemsel bağlılık sayesinde;

- belli bir koşulun doğrulanıp doğrulanmadığını denetlemekle ve gerektiğinde uygun özel yöntemi harekete geçirmekle yükümlü etkili "démon" bir eylemle. Genel yöntemlerden çıkan ve programı ağırlaştıracak olan özel durumlar böyle işlenir, fakat sürekli denetim pahalı ve riskli olmaktadır;

- gerçekleştirilecek olan hedefi belirleyen bir şema ile yönlendirilmiş olan çağrı ile. Karakteristik şemanın kendisine yüklenen göreve uygun şemayla eşzamanlı olduğu yöntem harekete geçirilmiştir. Bu mekanizma, problemin daha önce gördüğümüz, özgül çözümleme yöntemlerinin varolduğu alt problemlere ayrıştırılması tekniğine uygulanır. Hareket planlarının geliştirilmesine yönelik CONNIVER dilinde de vardır.

3. Avantajlar ve Sakıncalar - Kabul edilen tanıtım modeli ne olursa olsun, her zaman için bilgiyi açıklamak ve "uygulamak" üzere bir mekanizma gereklidir. Görüldüğü gibi, çekirdek bir yönteme her zaman gerek duyulur (örneğin prodüksiyonun kurallı bir sisteminin çıkarsama motoru).

Öte yandan "heuristique" yapıdaki bilgiler ve metabilgiler genellikle yöntemsel şekilde daha kolay açıklanırlar.

Sonuçta, yöntemsel bağlılık, bilginin ve insan usavurumunun yansıtılmasına daha iyi uydurulmuş karma bir tanıtım sağlar.

Buna karşılık, yöntemsel tanıtım, modülerlik ve esneklik kaybına uğrar. Bu tanıtım sorulan sorunun çözüm tarzının *a priori* bilgisini en azından kısmen kapsar ve sonuçta bir açıklama mekanizmasının ortaya çıkmasını kolaylaştırmaz.

VI. Biçimlendirilmiş Nesnelerle Tanıtımlar

1. Giriş - Elde bulunan bilgiler kümesini biçim-

lendirilmiş şekilde organize etme gerekliliği çok sayıda uygulama alanının bir sabitidir. Başta hafıza psikolojisi hakkındakiler olmak üzere çeşitli çalışmaların ardından 1974'de M. Minsky tarafından öne sürülmüş görüntü şemaları ya da prototiplerin (bundan sonra kullanacağımız Amerikan *frames* terimi) ve 1977'de R. Schank tarafından öne sürülmüş olan anadilin işlenmesine yönelik senaryoların *(scripts)* formalizmleri ortaya çıkmıştır. Belli bir etkinlik türünde manipüle edilmiş kavramsal nesnelerin biçimlendirilmesine olanak verirler.

Daha sonraları, özellikle de "yönlendirilmiş nesneler"i programlama dilleri üzerindeki çalışmaların ardından, bu biçimlendirmeler daha genel olan "nesnelerle tanıtım" kavramına dahil edilmiştir.

2. Frames ve Senaryolar - Bir *frame,* fiziksel bir nesneye, bir kavrama, tipik bir duruma ait, bilgiler kümesini biçimlendirilmiş şeklinde tekrar biraraya toplar.

Şarap kavramı şöyle de açıklanabilirdi:

<şarap *frame*'i
türü: içki
adlandırılması: ad1: (olası değerler:
AOC/VDQS/ülke şarabı/
masa şarabı
(defo: ülke şarabı)
(gerekliyse adlandırmaya
bakın)
yapı: yap: (olası değerler: sek/y
arı-sek/yumuşak/likörümsü/
keskin)
alkol derecesi: de (enterval: 9'dan 15'e)
(defo: 12)
üretici: untel (gerekirse etiket üzerindeki ismi
bulmak)
etiket:.....>

Bu *frame,* bu kavrama bağlı bir o kadar temel bilgiye uygun gelen "adlandırma" veya "prodüktör" gibi yüklemler veya haneler *(slots)* kümesinden oluşur. Her yüklem farklı yönlerde tanımlanabilir, örneğin, "olası değerler", "defolar" vb.

Bu örnek, *frame*'lerin esas özelliklerini gösterir, özellikle aşağıdakiler varsa:

- simgelere bağlı gıyabi değerler. Özel bir şarap, "ülke-şarabı" şeklinde belirtilmiş "adlandırma" simgesinin değerine sahip olacaktır. Bu değer, tesadüfen devreye girmesi sözkonusu olana kadar (gıyabi usavurma) belki de usavurmanın devamına olanak verecektir.
- olası değer entervalleri ve kümeleri.
- *eğer-gerekirse* (eğer usavurma simgenin değerini belirtiyorsa yöntem harekete geçer) veya bir yüklem için değer bulmamışsa sisteme ne yapacağını bildiren *eğer başarısızlık* gibi, yöntemler. Burada, V'de işaret edildiği gibi yöntemsel bağlanma sözkonusudur.

Usavurma sırasında bu jenerik kavramın örnekleri ortaya çıkacaktır, örneğin sınıf şarabı simgelerini taşıyan belli bir şarap. Simgelerin değerleri, bağlı usavurma mekanizması ile derece derece belirtilecektir.

İşte bir *frame* şarap örneği:

<örnek minervois-86
adlandırma: ülke-şarabı
yapı: keskin
alkol_derecesi: 14
üretici: GC à la Caunette (Hérault)
etiket: koyu-kırmızı>

Senaryolar aynı tanıtma prensibine dayanıyorlar ama tipik olaylar dizisine bağlıdırlar, nesnelerin statik tarifine değil. Bu durumda doğal olarak etkilenen eylemler kümesinden oluşacak olan "şarabın

tadımı" senaryosu böyle tanımlanabilir. Böyle bir tanıtmanın anadil metinlerinin anlaşılmasındaki yararı görülüyor.

3. Biçimlendirilmiş Nesneler - Nesnelerin formalizminde, bir uygulama alanı *özelliklerin kalıtımı* mekanizmasından doğan ve *jenerik sınıflar* ve *nesne örneklerinden* oluşan hiyerarşik bir kümeden ibarettir. Her nesne, bir *frame* gibi, simgelerle tanımlanır. Nesnede barınan enformasyonların manipülasyonunu sağlayan yöntemler, bir *metodlar* kümesini kapsar.

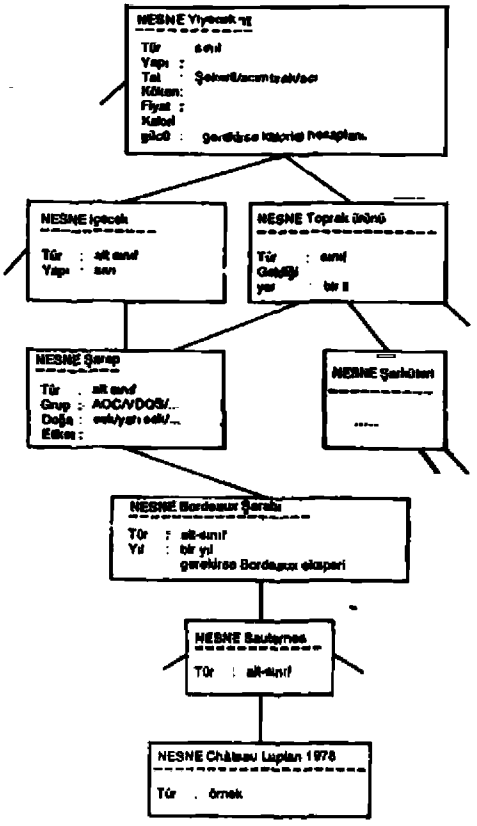
Bir nesneye veya onun simgelerinden birine bir alt-problemi çözümlemekle yükümlü bir prodüksiyon kuralları kümesi de bağlanabilir.

Bir alanın nesneleri bağımsızdırlar ve *mesajların iletilmesi* mekanizması ile aralarında ilişki kurarlar. Bir nesneye gönderilen mesaj nesnenin metodlarından biri ile sağlanan belli bir işlemi açıkça belirtir.

Şekil 11'de ,şema yiyecekler alanının bir bölümünü nesne terimleriyle göstermektedir. Özelliklerin kalıtımının *çok fazla* olabildiği görülmektedir: örneğin, alt-sınıf şarap hem içecek özelliklerini hem de Toprak-ürünü özelliklerini alır.

Bu formalizmde usavurma mekanizmaları çoktur:

- özelliklerin kalıtımı bir kavramın bilgilerinin seyirleri yönüne geçişini sağlar:
- metodlar bir nesnenin içinde çok biçimli olabilen (yöntemler, kurallar vb) lokal bir usavurma sağlarlar.
- global düzeyde, usavurma bir nesne örneğini tekrarlı bir mekanizma ile bu nesnelerin yorumlanmasına ulaşacak şekilde bir sınıf ile karşılaştırılmasından (olasılıkla belirsiz bir denetleme aracılığı ile) oluşur.



Şekil 11.- Nesnelerle tanıtım örneği.

Bu, uzman sistemlerde kullanılır ayrıca nesnelerin bir sahne görüntüsünde yeralan fiziksel nesneleri temsil ettiği görüntüde ve de sözün işlenmesinde de kullanılır.

4. Avantajlar - Nesnelerle tanıtımlar günümüzde geniş çapta kullanılıyorlar, çünkü çok sayıda

avantajları vardır:

- bir uygulama alanının biçimlendirilmesi, gerektiğinde bu alanda bir düzenleme sağlayarak;
- bir kavramla ilgili bilgi elemanları kümesinin toplanması (örneğin tıpta bir hastalığın bütün yönleri, semptomlar, işaretler, bağlı hastalıklar, tedavi yolları, yan etkileri vb.)
- deklaratif ve yöntemsel yönlerin birleştirilmesi;
- bilgilerin ve usavurumların önem sırasına konulması.

DÖRDÜNCÜ BÖLÜM
UZMAN SİSTEMLER

I. Giriş

Prodüksiyon kurallı sistemler uzman sistemlerin (US) kökeninde yeralırlar, ancak uzman sistemler şimdi usavurma ve bilgilerin temsil edilmesi konusunda başka teknikleri entegre ediyorlar. Uzman sistemlerin başarıları, kısmen, tanı, karar alma vb. gibi konularda başta çok az enformatize olan işlere anlamlı sonuçlarla yaklaşılmasını sağlamaları olgusundan kaynaklanır.

US'lerin '70'li yıllar boyunca ortaya çıkışı, ele alınan alandan bağımsız bir şekilde bir uzman tarafından çok iyi belirlenmiş bir etkinlik alanında reel bir problemi çözmek için ortaya konulan mekanizmaların incelenmesine yönelik olarak, genel problemlerin çözüm tekniklerinin (daha önce sözü edilen GPS) araştırılmasından bu yana, YZ'nin hedeflerinin gelişmesine denk düşer. Bu alandaki öncü proje DENDRAL'dir. Bu proje 1965'de E. Feigenbaum ve meslektaşları tarafından Birleşik Devletler Stanford Üniversitesi'nde bir kimyagere, organik bir bileşiğin yapısını, kitle spektrogramının ve ham kimyasal formülünün verileriyle bulması için, yardımcı olmak üzere başlatılmıştı.

Bu iddialı proje başlangıçta, en küçük bir düzeltmenin yeniden programlamada engelleyici maliyetler yarattığı çok büyük bir Fortran programına yolaçtı: Fizik-kimya alanına özgü bilgiler usavurma mekanizmalarına sıkı sıkıya bağlıydı. Dolayısıyla

projeden, bir alanın özgül bilgilerini yorumlama mekanizmalarında açık bir şekilde ayırmanın gerekli olduğu düşüncesi çıktı. Bilgi tabanlı sistemlerin ve US'lerin asıl temeli zamanla, verili bir problemi çözmek için bir bilgiler ve olgular bütününü kullanan çıkarsama motoru kavramıyla birlikte tedricen doğmuştur.

Metod, daha sonra tıp alanında ve US'lerin evriminde büyük etkisi olan (bazı kan hastalıklarının teşhisi) MYCIN (Shortliffe, 1976, Stanford Üniversitesi), moleküler genetikte planlama deneylerine yönelik MOLGEN (Stefik, Stanford Üniversitesi) ya da jeolojide, daha önce sözü edilen PROSPECTOR (Duda ve Hart, Stanford Research Institute, 1979) gibi sistemlerin deneyiyle geliştirildi.

O zamandan beri, yüzlerce US geliştirildi ve kullanıldı. İşlenen olguların derin bilgisini entegre eden *ikinci kuşak* US'lerin ortaya çıkışına çok kısa bir süreden beri tanık olunuyor.

II. Uzman Sistem Nedir?

1. Tanım - Bir uzman sistem, açık bir şekilde sağlanan ve esas olarak ele alınan alanın uzmanlarından elde edilen bir bilgiler bütünü sayesinde uzman bir kişinin kısıtlı ve sınırları iyice belirlenmiş olan bir alandaki verili bir işte gösterdiği performansa ulaşması mümkün bir *logiciel*'ler bütünüdür.

2. Uzman Bir Sistemin Temel Bileşenleri - Uzman sistem bir bilgi tabanlı sistem türü örneğidir. Üç kısımdan oluşur:

- Belki de birçok formalizm içinde, hem *sürekli olguların* ("Madiran bir kırmızı şaraptır"), hem de bir problemi çözmek için gerekli olan beceri ve uzmanlık alanının bilgiler bütününün içinde kodlan-

dığı bir sistemin uzun dönemli hafızasını oluşturan bir ya da birçok *bilgi tabanı.*

- Çözülecek probleme özgü *olguları* ya da *verileri* içeren *bir olgular tabanı* ("Bu şarap ahududu kokulu"). Dolayısıyla, problemin çözümüne varılana kadar usavurma mekanizmasıyla bulunan yeni olgularla zenginleşecek bir çalışma hafızası sözkonusudur.

Uyarı: Terminoloji yazarlara göre değişmektedir. Bazen, incelenen alana ait sürekli olguların olgular tabanına dahil olduğu düşünülür. Özellikle bir PROLOG programı içinde olgular ve kurallar arasında yapılan ayrıma dikkat çekilecektir. Biz alan üzerindeki uzun dönemli bilgi (bilgi tabanı) ile bir sistem yoklaması içinde incelenen özel probleme ait kısa dönemli bilgi (olgular tabanı) ayrımını yapmayı tercih ediyoruz;

- Bir *çıkarsama motoru* ya da *yorumlayıcı,* üçüncü bölümde anlatılana benzeyen bir devire göre, bir usavurmayı yönetmek için bilgileri ve olguları kullanmaktan sorumlu algoritma.

Bir motor belli bir işleyiş stratejisine sahiptir. En basit durumlar için strateji eksiksiz bir çözümler aralığı parkurundan ibaret olabilir. Genellikle, uygun bir zamanda bir çözüme ulaşmak için, daha iyi hazırlanmış, belki de metabilgiler bütünü üzerinde temellenmiş bir stratejinin ortaya konması gereklidir. Bu metabilgiler (özellikle çatışmaların çözümü için kullanılan metakurallar) zaten genellikle bir alan üzerindeki uzmanlığın önemli bir kısmını oluştururlar.

3. Uzman Sistemlerin Önemi - Bilgiler ile onların kullanılma mekanizması arasındaki açık ayrım, bilgi tabanlı sistemlerin kullanımı ve reel yararlanımı için pratik bir önem arzeder (dev sistemlerin artan yapımı, tabanlarının günışığına çıkarıl-

masını ve bakımının göreli kolaylığı vb).

Şu anki sistemlerin çoğunluğunun bilgilerinin en azından bir kısmı prodüksiyon kuralı biçiminde ifade edilir. Bu tanıtım modeli bir çok açıdan ilginçtir:

- bilgilerin okunabilirliği;
- kuruluş kolaylığı ve kuralların bağımsızlığı nedeniyle zaman içinde değişim kolaylığı;
- örneğin kullanılan kurallar dizisinin incelenmesi sayesinde bir usavurmayı açıklama kapasitesi.

III. Uzman Sistemlerin Uygulanması

1. Uzman Sistemlerin Tipolojisi - Uzman sistemler aşağıdaki özelliklere sahip problemlerin çözümüne yardımcı olabilirler:

- tıpkı problemin verileri gibi, belki de eksik ve şüpheli olan büyük bir bilgi yığınını ortaya koyma gerekliliği;
- alanın evrimsel ve donuk olmayan karakteri;
- çözümlemenin "heuristique" yapısı (ancak algoritmik çözümü olan bir problem için bir uzman sistem düşünmek de aynı derecede avantajlı olabilir);
- çözüm içinde sembolik hesabın sayısal hesaba üstünlüğü.

Bu problemler üç büyük kategoride sınıflanabilir:

- *Verilerin yorumlanması:* Burada, kendisinden sembolik bir yorum çıkarmak için bir veriler ve fiziksel işaretler kümesini incelemek sözkonusudur. Bu tür, özellikle teşhis problemlerini kapsar. (noktasal ya da zamansal boyutu içeren): tıbbi teşhis, arıza teşhisi, mali teşhis, vb. Bu, şu anki uzman sistemlerin önemli bir parçasını oluşturur. Ayrıca

önemli işaret yorumlama sınıflarını (çeşitli alıcılardan elde edilen biyomedikal işaretler, akustik, endüstriyel işaretler); önemli görüntü ve tahmin yorumlama sınıflarını (bazen incelenen sürecin matematiksel bir oluşturumuyla bağlantı içinde, belli bir durumun tahmin edilebilir sonuçlarının araştırılması) belirtmekte fayda var. Bu, örneğin, meteorolojinin, jeopolitiğin (dünyanın bir noktasındaki çatışmaların öngörülmesi), tarımın (ürün tahmini), ekonominin ya da para işlerinin, doğal tehlikelerin vb. durumudur.

- Planlama: Bu çok biçimli disiplin çeşitli alanlarda rastlanılan karmaşık etkinlikleri kapsar. Bu etkinlikler arasında şunları sayabiliriz:

* karar alma (kumandaya yardım, bir girişimin ya da projenin yönetimi, risklerin değerlendirilmesi vb.);

* belgeleme ve düzenleme analizi;

* görevlerin düzenlenmesi ve kaynakların tahsisi (bu problemler işlemsel araştırmanın algoritmik yöntemleriyle çözülür, ancak uzman sistemler hesap zamanında genellikle daha esnek ve daha az pahalı çözümler getirirler);

* robotbilimde hareketlerin planlanması ve prodüksiyon kontrolü.

- Kavrama: Uzman bir sistem bir şartnamenin spesifikasyonlarından yola çıkarak kavrama sürecine yardım edebilir. Birkaç örnek sayalım: VLSI entegre devreler, işletme serileri, bir mutfağın düzenlenmesi, enformatik bir sistemin (materiel ve "logiciel" biçimi, gen klonajı, yeni bir kimyasal bileşiğin yaratılması.

Son olarak uzman sistemlerin bilgisayar yardımlı öğrenim (EAO) (ya da daha genel olarak enformasyon) sistemlerinde kullanılabildiğine işaret etmek gerekir. Uzman sistemler aşağıdaki bazı

özellikleri sayesinde EAO alanını zenginleştirirler:

- Uzman bir sistemin sadece bir alıştırmanın çözümünü bilmekle kalmayıp, *öğrenene* açıklamalar verme, yanıtlarına bağlı olarak alıştırmaların düzeyini değiştirme, usavurmada ortaya konan bilgileri açıklama vb. olanağını sağlayan çözümleme metoduna sahip olması;

- Formasyon toplantıları sırasında dinamik bir şekilde değiştirilebilir bir öğrenen *modeli* ya da *profili* tanımlama olanağı;

- Çeşitli uzmanlık seviyelerinin entegrasyonu: incelenen alan üzerindeki uzmanlık, genel pedagojik uzmanlık, özel bir öğrenen modeli.

Uygulamalar çok sayıda ve çeşitlidir: tıp (bir hastalıklar sınıfı için teşhisin öğrenilmesi), endüstri (özellikle demir madenciliğinde olmak üzere, karmaşık tesisat operatörleri için yönergelerin öğrenilmesi), klasik öğretim (yabancı diller, matematik, fizik...).

2. Uygulama Alanları - Sözü edilen çeşitli US türleri çok değişik uygulama alanlarında biraraya gelirler. Aşağıda, uygulamanın çeşitli aşamalarında uzman sistemler tarafından çözülmesi istenen problemlerle birlikte bu tür alanların tam bir listesini veriyoruz:

- *Havacılık ve uzay:*

* pilotluğa yardım,

* hava trafiğinin, bir görevin planlanması,

* uyduların kontrolü: duruş düzeltimi, bataryaların yönetimi, fırlatma operasyonlarının kontrolü;

- *Tarım:*

* hastalıkların teşhisi ve bitkisel patolojide tedaviye yönelik tavsiye,

* tohumların, gübrelerin seçimine yardım,

* bir çiftliğin, ürün toplamanın yönetimi;

- *Banka, maliye ve sigorta:*

* mali yönetim,
* yönetim kontrolü ve odit,
* portföylerin yönetimi, yatırıma yardım,
* girişimlerin teşhisi,
* ödünç verme risklerinin değerlendirilmesi,
* sözleşmelerin kaleme alınması;

- *Biyoteknolojiler:*

* moleküler genetikteki deney planlarının kavranması,
* nükleik asit ayrımlarının açıklanması,
* elektroforez donlarının açıklanması;

- *Kimya:*

* yeni organik moleküllerin sentezine yardım,
* moleküllerin üç boyutlu yapısının açıklanması;

- *Hukuk, mevzuat:*

* normların incelenmesi (güvenlik, vb.)
* emeklilik dosyalarının tasfiyesi,
* karmaşık mevzuatın yönetimi,
* belgelerin kaleme alınmasına yardım (sözleşmeler, noter belgeleri, vb.),
* yasal metinlerin kullanımına yardım;

- *Elektronik:*

* elektronik devre, santral ve telefon şebekesi arızalarının teşhisi,
* VLSI devrelerin anlaşılmasına yardım,
* elektronik ve karmaşık elektrik devrelerinin (lojik sistemler, lokomotifler, demiryolları katarları, vb.) bakımı;

- *Jeoloji:*

* jeolojik ve madensel araştırmaya yardım,
* jeoteknik deneme kampanyalarının kavranması,
* açma havuçlarının açıklanması,
* açmaların yönetimi;

- *Enformatik:*

* programlamaya yardım,
* enformatik bir sistemin (materiel ve "logiciel") biçimlenmesi,
* kullanım ve bakıma yardım,
* büyük *progiciel*'lerin (mekanik ve istatistik vb.) kullanımına yardım;

- *Endüstri:*

* endüstriyel gelişime kılavuzluk,
* işletme serilerinin, CAO, kavranması,
* esnek atölyelerin düzenlenmesi, üretim yönetimi,
* denetim, arıza teşhisi, kazaların önlenmesi.

Bu tür uygulamalar çeşitli endüstriyel alanlarda bulunurlar: otomobil, tarımsal beslenme, petrokimya, demir madenciliği, nükleer endüstri vb.

- *Matematik:*

* sembolik hesap,
* matematik program kütüphanelerinin kullanımına yardım,

- *Tıp:*

* teşhise yardım,
* görüntülerin yorumlanmasına yardım (radyoloji, ekografi, vb.)
* reanimasyondaki hastaların bakımı;

- *Askerlik:*

* hedeflerin belirlenmesi ve izlenmesi (radar, sonar), denetim,
* çok sayıda enformasyonun birleştirilmesi, risklerin değerlendirilmesi,
* kumandaya yardım,
* araç kullanımına yardım (tanklar, helikopterler, avcı uçakları);

- *Öğretim ve formasyon.*

IV. Bir Uzman Sistemin Gelişimi

1. Seçim Kriterleri - Bir uzman sistem geliştirme kararı, ancak düşünülen işin toplu koşullarının dikkatli bir incelemesinden sonra alınabilir. Aşağıdaki faktörler bu tür bir geliştirmenin düşünülmesini makul kılar:

- problem sadece nicel değil aynı zamanda nitel enformasyonları da ortaya koyar;
- bilgiler sadece sağduyu bilgileri değildir; sezgisel bir karaktere de sahiptirler;
- problemi çözebilen ve yöntemlerini açıklayabilen tanınmış uzmanlar vardır;
- problem makul bir güçlüktedir ve muhtemel ya da arzu edilen algoritmik çözüme sahip değildir.

Daha sonra, ortaya konulan problemin, US türündeki çözümünün reel olarak doğrulandığından emin olmak gerekir. Sadece ekonomik kriterlere dayanan bir doğrulama yeterli değildir, çünkü uzman sistemin yatırım getirisini hesaplamak genellikle zordur. Buna karşılık, uzman sistemlere yapılan başvuru şu koşullarla doğrulanabilir:

- uzmanlık nadirse ya da sağlam temelden uzaksa;
- uzmanlık bir yerde yoğunlaşmış fakat dağınık bir şekilde kullanılıyor ise;
- kararlar "zorlayıcı" koşullarda alınmak zorundaysa.

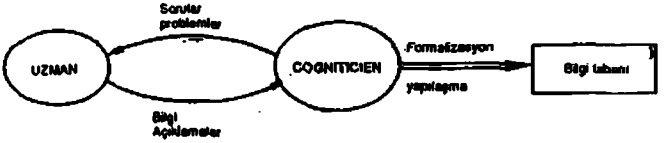
2. Aktörler - YZ'nin güncel durumu içinde, uzman bir sistemin geliştirilmesi en azından iki kişinin (ya da iki grup kişinin) müdahalesini gerektirir:

- ele alınan alanın uzmanı;
- ayrıca çok iyi bir enformatik kültüre sahip olan bir yapay zeka uzmanı (bazen *cogniticien* ya da *bilgi mühendisi* diye adlandırılır).

Uzman ve bilişselin incelenen alan için kavram-

ları, beceriyi, usavurma keşiflerini toparlayan bilgi tabanını derlemek üzere sıkı bir işbirliği içinde çalışacaktır.

Çok nazik ve esas olarak zanaatsal bir temel aşama sözkonusudur. Sembolik öğrenme aletlerinin bundan böyle bilgi tabanının kuruluşuna yardım edeceği umut edilebilir.



Şekil 12. Uzmanlık transferi süreci

3. Gelişme Aşamaları- US'ler çok özel niteliklere sahip olan ve devirleri klasik *logiciel*'lerden farklı olan *logiciel*'lerdir.

Genellikle, bir uzman sistem projesi dört aşamadan oluşur:

- İncelenen problemin sadece bir alt kümesini hesaba katan bir *maketin* ya da *kanıtlayıcının* ortaya konması. Bu sistem problemin uzman sistem teknikleriyle çözülebilir olduğunu kanıtlama ve projenin tamamı için gerekli olan kaynakları değerlendirme olanağı verir.

- Problemin bütün görünümlerini kapsayan fakat zorunlu olarak gerçek kullanım çevresine entegre edilmeyen bir bilgi tabanına sahip bir *prototipin* geliştirilmesi.

-*Nihai ürünün* çevresine entegre edilmesi. Aslında, uzman bir sistem genellikle varolan bir sisteme entegre olur: bir işletmenin enformasyon sistemi, üretim zinciri, vb. Bu entegrasyon karmaşık enformatik ve ergonomik problemlere yolaçar: başka *logiciel*'ler ya da veri tabanlarıyla biraraya gelme,

gerçek zaman görünümleri, gelecekteki US kullanıcılarıyla etkileşim, vb.

- Sistemin *işletilmesi.* Bilgiler genellikle evrimsel bir yapıda olduğu için, uzman bir sistemin bakımı ve günışığına çıkarılması özgül *logiciel*'ler tarafından sağlanan önemli operasyonlardır.

4. Aletler - Bir US genellikle, farklı işlevleri olan karmaşık ve çok biçimli bir *logiciel*'dir.

- Ortaya konulan *problemin* bir ya da birçok bilgi tabanını kullanan bir ya da birçok çıkarsama motoru tarafından çözümlenmesi.

- Başlangıçtaki ortaya koyma ve sonraki bakım aşamalarında, tam formalizm içinde insan uzmanlığının kodlanması için *cogniticien* tarafından kullanılan fonksiyonların, bilgilerin *elde edilmesi, değiştirilmesi ve günışığına çıkarılması.*

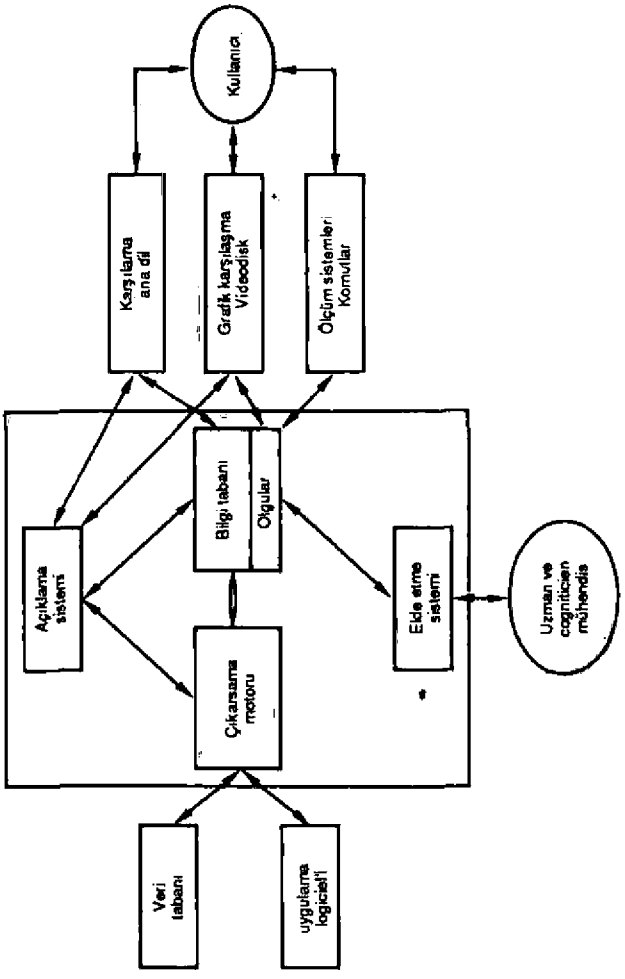
- Ortaya konulan problemi çözmek üzere US tarafından kullanılan yolun *açıklanması.* Bu, sistemin ortaya konması sırasında ya da formasyona yönelik bir US durumunda kullanımda önemlidir.

- Harici sistemlerle *bağlantı:* US'nin talebiyle bazı görevleri yerine getirmekten sorumlu *logiciel*'ler, usavurma için gerekli bazı olguları sağlayan veri tabanları, vb.

- Nihai kullanıcı ile ve gerekirse US tarafından denetlenen fiziksel süreçle *karşılaşma.* Bu işlevsellik sistemin iyi işletilmesi için önemlidir ve sistemi çok çeşitli formlar ile donatır: bir sürecin (basınç, ısı, seviyeler, vb.) parametrelerini saptayan alıcılara bağlama, grafiksel görüntüleştirme, anadilin kullanımı.

Şekil 13 uzman bir sistemin işlevselliklerine toplu bir bakışı yansıtmaktadır.

Uzman bir sistemin karmaşıklığı gözönüne alındığında, geliştirilmesi genellikle doğrudan bir programlama dilinde değil, aşağıda betimlenen işlevsel-



Şekil 13.- Uzman bir sistemin işlevsellikleri.

liklerin tümünü ya da bir kısmını sergileyen üst düzeyde bir *logiciel* araç kullanılarak yapılır; bu da, bu geliştirmeyi büyük ölçüde kolaylaştırır. Pazarda, çeşitli makine sınıfları için bu tür araçların birçok çeşidi mevcuttur; mikrobilgisayarlar, iş makineleri, sembolik makineler, merkezî bilgisayarlar.

V. Sonuç

Uzman sistemlerin kullanımı birçok faaliyet sektöründe hızla gelişti, çünkü pratikte sağladıkları faydalar açıktır. Bu arada, önemli sınırlamaları vardır:

- Bilgi tabanının bütününün "elle" oluşturulması gerekliliği ve otomatik öğrenmenin hemen hemen tamamen olmaması: Bir US şimdilik, deneyimle iyileşmez.
- Kapalı uzmanlık alanlarının sınırlılığı ve US'nin kendi yetkinlik sınırları konusundaki bilgi eksikliği;
- Uzman kişinin usavurmasını dayandırdığı, ortada görünmeyen olguların derin bilgisine karşılık, uzman sistem tarafından kullanılan bilgilerin yüzeysel görünümü;
- Uzman sistem ile kullanıcı arasındaki etkileşim araçlarının yetersizliği;
- Ortaya konulan usavurma tarzlarının sınırlılığı.

Şimdi, aşağıdaki eğilimlere sahip olan ikinci kuşak sistemler ortaya çıkmaktadır:

- Bilgilerin birçok birbirini tamamlayıcı modele göre temsil edilmesi (özellikle yapısallaşmış nesneler kavramı etrafında).
- Derin bilgilerin kullanımı (özellikle açıkça görülmeyen olguların açık ve niteliksel bir modelleştirilmesi ile bağlantı halinde); bir uzmanınkine daha

yakın usavurma yapma olanağı sağlar (örneğin geçici usavurma).

- US'nin varolan enformatik sistemlere daha iyi entegrasyonu.

- Anadilin anlaşılması, imajların işlenmesi gibi diğer YZ tekniklerinin aynı anda kullanılması.

BEŞİNCİ BÖLÜM
ANADİLİN İŞLENMESİ

I. Giriş

Fransızca ya da İngilizce gibi bir anadil insan ile makine arasındaki etkileşimin ayrıcalıklı bir aracıdır. Bir veri tabanı içinde bulunan enformasyonlara giriş her ne kadar formel, yapay bir dili gerektiriyorsa ve onu öğrenmek zorunlu ise de, anadilde (yazılı ya da sözlü) bir soru sorabilmek kullanıcı için çok büyük bir avantajdır. Hafızadaki dil gündelik dilin az çok zengin bir altkümesidir ve üstelik kullanımın veri tabanının nasıl düzenlendiğini bilmesini gerektirmez.

Ayrıca anadil önemli bir ifade gücü ve büyük bir özlülük sağlar. Çok sayıda özelliğin yanı sıra eksilti ya da adıl referans gibi olgular diyaloğa etkililik ve doğallık verir. Örnek olarak bir veri tabanına giriş diyaloğunu veriyoruz (kullanıcının soruları italik yazılmıştır, yanıtlar sisteme aittir):

S: *Layon bağ üzümleri hangi bölgeden gelmektedir?*

Y: L'Anjou

S: *Ya tavel?* (eksilti)(*)

Y: Les Côtes du Rhône

S: *Savoie şarabının rengi nedir?*

Y: Beyaz

S: *Vic-bihl'inki?* (adıl referans)

(*) Eksilti (*phénomènes d'ellipses*): Bir sözün anlaşılması için bulunması zorunlu olmayan bir ya da birkaç sözcüğün eksik söylenmesi. (ç.n.)

Y: Aynı

Bu özellikleriyle anadil dağınık enformasyonlara hızla ulaşmayı arzu eden profesyonel için olduğu kadar, rastgele bir kullanıcı için de (Minitel ile haber alma, danışma merkezleri) çekicidir.

Anadilin işlenmesi çeşitli bilgi türlerini ortaya koymayı gerektiren ve çeşitli ve tamamlayıcı yetenekleri devreye sokan bir etkinliktir:

- Dilbilim: İncelenen dilin tanımına uygun bir usulün bulunması için;
- akustik, fonetik, sesbilim ve sağdeyi: konuşma durumunda;
- mantık: açıkça görünmeyen usavurmaların matematiksel formülasyonu için;
- psikoloji: incelenen olayları formalize etme amacıyla insani bir modelden esinlenme çabası için;
- ergonomi: kullanılabilir insan-makine diyaloğu sistemleri kurmak için;
- nihayet, bütün bu yetenekleri kullanmak için yapay zekâ.

Bir makineyle ilişki kurmak için bir anadilin kullanılması aslında yapay zekânın en kuvvetli yanlarından birini temsil eder ve anadilin işlenmesinin enformatik içinde artan bir önem kazanması beklenebilir. Aşağıda, yazılı dilin temel işleyiş ilkelerini sunuyor ve konuşma diline özgü özellikleriyle konuyu bitiriyoruz.

II. Yazılı Anadil

1. Kısa Tarihçe - Yazılı dilin işlenmesine yönelik *ilk girişimler* 1946'dan itibaren otomatik tercümeyle (OT) ilgiliydi. Bilgisayar basit bir elektronik sözlük olarak algılanıyordu. 1952'de Bar-Hillel gibi öncülerle birlikte, MIT'te ilk OT konferansı yapıldı. O dönemde, OT tercüme edilen cümlenin sözcüğü

sözcüğüne çevrilmesinden ve (gramere bağlı kalarak) sözdizimsel bir reorganizasyonundan ibaretti. Sorunun bu derece küçümsenmesi kendini güçlü bir şekilde hissettirecek ve yakın bir döneme kadar OT'nin tarihini etkileyecektir.

'60'lı yıllar anadilin işlenmesinde özellikle N. Chomsky'nin formel (1957) ve transformasyonel gramerler üzerine çalışmalarıyla birlikte *presemantik dönemi* temsil eder. Dolayısıyla ilk sistemler Birleşik Devletler'de ortaya çıktı: Baseball (bir sezonun maçlarıyla ilgili bir veri tabanıyla çalışma), Student (cebir alıştırmalarının çözümü), Sir ya da Eliza (bir psikiyatr ile hastası arasındaki diyaloğun oluşturumu). Bütün bu sistemler analiz edilecek cümledeki anahtar sözcüklerin basit araştırmasına dayanırlar ve dolayısıyla son derece sınırlıdırlar. Aynı dönemde, Bar-Hillel bugün evrensel olarak kabul edilen bir anlayışla, bir cümleyi anlamadan çevirmenin olanaksızlığını ve sorunu çözmek için bir bilgiler bütününe başvurmanın gerekliliğini kabul ediyordu.

'70'li yıllar, özellikle Lunar (BBN'den Woods), SHRDLU (MIT'ten Winograd), Margie (Yale'den Schank) sistemleriyle, *sentakso-semantik dönemin* başlangıcına damgasını vurdu. Lunar sözdizimsel (syntaxique) bir analizi ve sonra sözdizimsel ağacın anlambilimsel (sémantique) bir yorumunu gerçekleştiriyor. SHRDLU analiz sırasında anlambilimsel bilgiler kullanıyor; bu sistem elde bulunan çeşitli bilgi kaynakları arasında bir ilişki kurmanın gerekliliğini ve analizin bir aşamasında ortaya çıkan belirsizlikleri başka bir aşamaya başvurarak çözmenin olanaklı olduğunu gösterme şerefine sahiptir. 1973'de Wilks de İngilizceden Fransızcaya çeviri yapan önemli bir sistem ortaya koydu.

Basit bir cümlenin bir şema ile temsil edilebildi-

ği kavramsal ilişki teorisine uygun çerçeve ve yazı düzenlerinin ortaya çıkması, aynı anda, dile getirilen ile uygulama ortamının gerçekliği arasındaki ilişkiyi dikkate alan *pragmatik dönemin* başlangıcını belirler. Bu dönem, asıl metindeki yanlışları hoşgören esnek bir analiz yapma amacıyla, elde bulunan tüm bilgi seviyelerinin karşılıklı etkileşimi ile nitelenen şu anki döneme kadar devam etti.

2. Uygulamalar - Yazılı anadil işlemi uygulamaları çok sayıdadır. Az çok gelişmiş ilerlemeler sağlayan başlıcaları arasında şunlar sayılabilir:

- otomatik tercümeye yardım (ya da BYT, bilgisayar yardımıyla tercüme);
- metin özetlerinin otomatik olarak hazırlanması;
- dokümanların otomatik olarak indekslenmesi (yüzeysel kavramaya uygun olarak anahtar kelimelerin aranması);
- metinlerin otomatik olarak üretilmesi (bir anlamı olan sözdizimsel bir formun oluşturulması);
- dokümanların hazırlanmasına yardım (hataların, yanlış kuruluşların, tutarsızlıkların bulunması ve gerektiğinde düzeltilmesi), özellikle basılmaları amacıyla;
- enformatik ya da ekonomik sistemlerin kontrolü: veri tabanlarına giriş, bir uzman sistemle ya da bir robotla diyalog, bilgisayar yardımlı öğretim (BYÖ);
- problemlerin anadilde çözümü (bilgisayarların programlanması...)

3. Bilgiler ve İşlem Seviyesi - Anadil, kullanımını etkili kılan, işleyişini ise yüksek nitelikli hale getiren, bazılarını daha önce ele almış olduğumuz ayırdedici özelliklere sahiptir. Bu özellikler sözdizimsel uygulamalardan bağımsız olarak bütün dillerde bulunur:

- eşanlamlılıklar ve eşseslilikler;
- *allotaxies* ve *homotaxies* (aynı anlama sahip farklı sözdizimsel yapılar ve tersi);
- anafor(*) (adıl referanslardan daha önce sözedildi);
- eksiltiler;
- mecazlar (sözcüklerin asıl anlamından farklı bir anlamda kullanılması);
- vb.

Anadili kullanmak çok farklı modellere uygun olarak temsil edilen ve kullanılan çeşitli bilgilere, eşgüdümlü ve "zeki" bir biçimde başvurulmasını gerektirir. Bu bilgiler aşağıdakilere bağlıdır:

- *biçimbilim* (morfoloji) ve *sözcük dağarcığı:* uygun formların tanımı ve olanaklı dış görünüş formlarının oluşturulması fiil çekimi, çoğullar, vb.);
- *sözdizim* (sentaks), bir cümlenin öğelerinin (sözcükler, sözcük grupları, dizimler) yeniden gruplanması. Bir dilin sözdizimini göstermek üzere çok sayıda formalizm ortaya atılmıştır. Bunlar ortaya çıkış sıralarıyla şunlardır:

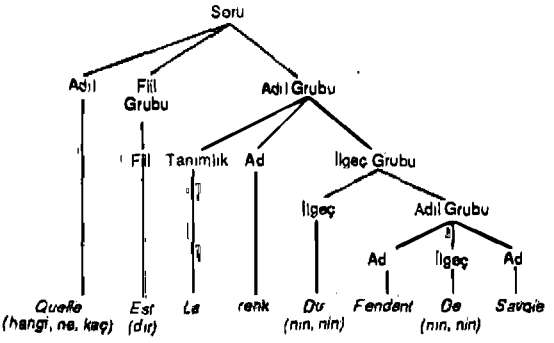
* Chomsky'nin, problemin tamamen sözdizimsel görünüşüne uygun formel ve transformasyonel kavramları;
* Fillmore'un anlambilimsel bir görüntüye sahip ad durumu gramerleri (bir fiil ile bu fiilin dahil ettiği değişik etkenler ve tümleçler arasındaki bağları ortaya koyar);
* Halliday'in, dilin fonksiyonel yönü üzerinde duran sistemik gramerleri;
* Kay'in, farklı bilgi türlerini birleştirmek için şema kavramından yararlanan fonksiyonel gramerleri.

Bir cümlenin sözdizimsel yapısı, birinci bölü-

(*) Anafor: bir etki sağlamak için bir sözcüğün tümcenin çeşitli öğeleri başında yinelenmesi (ç.n.).

mün bir diyaloğundaki bir cümleyi ele alan şekil 14'ün gösterdiği gibi, kolaylıkla bir ağaç ile gösterilebilir.

- *Anlambilim,* sözcüklerin anlamlarına ve gizli kavramlara bağlıdır. Anlambilimsel enformasyonlar çok çeşitli şekillerde gösterilebilir: sözcük dağarcığındaki sözcüklere bağlı özel nitelikler (canlı/cansız yaşayan, vb.), anlambilimsel şebekeler (kavramlar arasındaki ilişkileri göstermek için bu alanda yaygın bir biçimde kullanılır). Anlambilimsel şebekeler anlambilimsel bağların kuruluşuyla çıkarsamalar yapılmasına (dolayısıyla bir cümleyi yorumlamak için akıl yürütülmesine) olanak verir. Anlambilimsel özellikleri göstermek ve istif edilmemiş özellikleri çıkarsama yoluyla açık bir şekilde bulmak için mantıksal usavurma da aynı ölçüde kullanılır (özellikle PROLOG dilinin, fakat aynı zamanda üçüncü bölümde anlatılan anadile daha yakın mantıkların yardımıyla yüklemler mantığında);



Şekil 14.- Sözdizimsel ağaç örneği

- Uygulama ortamına özgü, zaman içinde, örneğin diyalog süresince gelişmeye elverişli bilgileri kapsayan *pragmatik*. Bu bilgiler anlambilimsel şebekeler, prodüksiyon kuralları ve özellikle prototipik betimlemeler (şemalar, senaryolar) biçiminde ifade edilebilirler.

4. Güncel Eğilimler - Anadilin işlenmesi artık endüstriyel bir etkinliktir, ama aynı zamanda bu alanda yapılan araştırmalar önemlidir, çünkü uygulamalar hâlâ çok sınırlıdır. Güncel eğilimler arasında şunlar sayılabilir:

- sözdizimsel ve anlambilimsel analizler arasındaki etkileşim;
- ana dile özgü olayları (anlam belirsizliği, şüphelilik, vb.) ele almak için mantık kullanımı;
- hataları, bilinmeyen sözcükleri ve kuruluşları kabul eden toleranslı analiz teknikleri;
- yalıtılmış cümlelerin kavranmasının ötesine giden gerçek bir konuşmaya izin veren insan-makina diyaloğu.

III. Konuşmanın İşlenmesi

1. Genel Sunuş - Bir makineyle iletişim kurmak için sözün kullanılması yazılı dilinkilere benzer avantajlar sağlamanın yanı sıra özel durumlarda da yararlar getirir:

- kullanıcının elleri ya da gözleri meşgul olduğunda;
- uzaktan iletişimde (telefon, radyo iletişimi).

İnsan ve makinenin sözlü iletişimi aşağıdaki konularda tamamlayıcı bir etkinlikler bütününü kapsar:

- istifleme ve iletme amacıyla sözün kodlanması ve azaltılması;
- okuyucunun doğrulanması (bir kişinin kimliği-

nin sesiyle onaylanması);

- sözün sentezi ("bir makineyi konuşturmak");

- esas olarak YZ tekniklerine bağlı olan, sözün tanınma-kavranması ya da RAP ("bir makinenin ses aracılığıyla anlamasını sağlamak"). Biz bundan sonra kendimizi bu yan ile sınırlayacağız.

Çok geniş olan uygulama potansiyelleri alanı iki büyük grupta toplanabilir:

- verilerin elde edilmesi: seçme (paketler, bagajlar), kalite kontrolü, şifre çözümü, sesli dikte makinesi;

- otomatik sistemlerin kumanda ve denetimi: makinelere basit emirlerle kumanda edilmesi (sanayi, motorlar), makine-aletlerin programlanması, bir robotla diyalog, servislere ulaşma (rezervasyonlar, veri bankaları, haber alma merkezleri).

Çok değişken karmaşıklıkları vardır; en gösterişlileri (dikte makineleri, diyaloglar) çözümlenmesi uzun sürecek temel sorunlar ortaya koyarlar. Güçlükler, sesli sinyalin özelliklerinden kaynaklanmaktadır:

- *Yineleme:* Bu sinyalin taşıdığı büyük miktardaki enformasyondan fazla zarar vermeksizin uygun enformasyonu çekmek gerekir.

- *Değişkenlik:* Okuyucular arasındaki (cinsiyet, aksan, vb.) ve aynı zamanda okuyucunun kendi içindeki (duygular, yorgunluk, vb.) farklılıklar sözlü bir mesajın tanınmasını çok güçleştiriyor. Ayrıca konuşmanın temel sesleri ya da ses ögeleri yakın sesler tarafından deforme oluyor (bağlamsal koartikülasyon olayları).

- *Süreklilik:* Akustik sinyal normal konuşmada esas olarak süreklidir. Ses ögeleri ya da sözcükler gibi birimlere ayrılması ("bölümlenme") insandaki karmaşık bir dil sürecinden kaynaklanır.

Bir kez daha çözümü çok sayıda bilginin kulla-

nımından geçen bir problemle karşı karşıyayız. Bu bilgilerin bazıları yazılı dil konusunda ayrıntılandırıldı. Söz için ayrı bilgiler de var:

- sesli sinyalden anlamlı parametreler çıkarmak için *akustik* bilgiler;
- ses ögelerinin özelliklerine ve bağlam içindeki değişimlerine bağlı fonetik ve sesbilimsel bilgiler;
- konuşma dilinin çok karışık "noktalama"sını düzenleyen sağdeyisel bilgiler (ses şiddeti, ritm ve ses perdesi).

İlk RAP girişimleri '40'lı yıllara kadar uzanıyor; önce otonom elektronik sistemlerin yardımıyla, sonra bilgisayarın güçlerini sınırlı bir başarıyla kullanarak. Problemin son derece güç olduğunun ve çok büyük miktarlarda bilgiyi ortaya koymanın gerekli olduğunun anlaşıldığını görmek için '60'lı yılları beklemek gerekir (otomatik tercüme ya da problem çözümü durumlarına benzer bir durum).

Daha sonra ilkeleri ve amaçları çok farklı olan iki yaklaşım ortaya çıkıyor:

- Bir okuyucu tarafından telaffuz edilen yalıtılmış sözcüklerle sınırlı bir sözcük dağarcığının tanınmasıyla kendini sınırlayan, uygulaması daha fazla olan *global* denilen bir *yaklaşım*. 1972'de Birleşik Devletler'de ilk ticari sistemi doğuran bu yaklaşımdır.
- Birimlerin tanınmasındaki yetersizlikleri bir ölçüde azaltabilmek için üst düzeydeki dilbilim seviyelerine ait bilgilere başvuran, daha temel olan, *analitik* denilen yaklaşım: sürekli doğal konuşmadaki fonetikler (sadece yalıtılmış sözcükler değil). 1971-1976 arasında Amerikan Savunma Bakanlığı'nın desteğiyle yürütülen büyük bir araştırma bu düşünceye dayanıyordu. Bundan birçok sistem doğdu; bunlardan biri başlangıç spesifikasyonlarını taşıyordu (HARPY, Carnegie-Mellon Üniversitesi'nde

gerçekleştirildi). Bununla birlikte, bugüne kadar bu ilkelere dayanan hiçbir sistem ticarileşemedi.

Bugünkü eğilim, ilerleyebilmek için aynı anda farklı modellerin kullanıldığı bu iki yaklaşım arasında daha belirsiz bir ayırım yapma yönündedir.

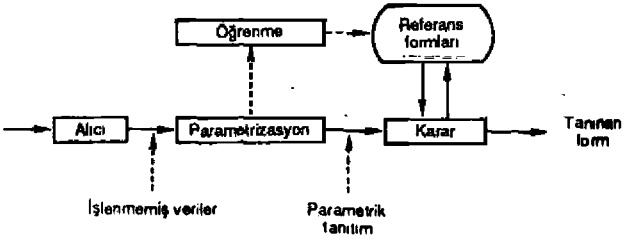
2. Sözcüklerin Tanınması - Global RAP yaklaşımında, tanıma birimi öncelikle, yalıtılmış olarak telaffüz edilen sözcüktür. Bu daha önce ortaya konan bölümleme problemlerini kuşkusuz çözer. Ayrıca, bir ya da birkaç kişinin sesinin tanınmasıyla yetinilecekse problem çok basitleşir. Böylelikle problem, bilinmeyen bir sözcüğün telaffuzuna uygun düşen akustik formu, makine tarafından daha önceden yapılan bir öğrenme aşamasıyla elde edilen ve sözcük dağarcığını oluşturan sözcüklerin akustik izleri ile global olarak karşılaştırmaktan ibaret olur.

Burada tipik formların tanınması (FT) problemi sözkonusudur. FT çok yaygın özerk bir disiplini temsil ettiği halde genellikle YZ'ye dahildir. FT'nin genel amacı çeşitli algılayıcılar tarafından biraraya getirilen formları tanımlayabilen otomatik algılama sistemlerini oluşturmaktır.

Bir FT sisteminin şekil 15'de gösterilen genel yapısı üç büyük kısımdan oluşur:

- Formu almak (örneğin konuşma için mikrofon, görüntülerin tanınması için optik okuyucular: yazılı karakterler, bir dekorun eşyaları) ve onu bir programın kullanabileceği şekilde işlemek için bir kavrama sistemi.
- Bir formdan iyi örneklenmiş bir parametreler bütünü çıkarmakla yükümlü bir parametrizasyon katı (örneğin, konuşma için frekans alanlarındaki enerji, basılı bir krakteri çizmek için siyah ya da beyaz noktalar ya da bir resimdeki bir eşyanın "iskelet"ini temsil eden çizgiler).

- Bir prototip formlar kümesine başvurarak bir formu tanıyan bir karar katı.



Şekil 15.- Bir FT (formların tanınması) sisteminin prensip şeması

Bir FT sisteminin oluşturulmasındaki kilit sorun incelenen formların değişkenliğini hesaba katma olanağı tanıyan prototip formların uygun bir tanıtımının yapılmasında yatar. İki büyük yöntem grubu kullanılır:

- İncelenen formların parametrelerinin istatistiksel olarak tanımlanmasına dayanan *istatistiksel* yöntemler. Bu yöntemler "bir sınıfa ait olmanın en yüksek olasılığı" kriterine uygun olarak, bilinmeyen bir biçimin nasıl sınıflandırılacağına dair bir karar alma olanağı verirler.

- Formların yapısıyla ve temel altbiçimlerin biraraya getirilmesi aracılığıyla yapılan betimlenmeleriyle ilgilenen *yapısal* yöntemler.

Bu iki yöntem-istatistiksel ve yapısal- birbirini tamamlar ve bazen aynı anda kullanılır.

FT'nin çok sayıda pratik uygulaması ortaya çıktı. Bu uygulamalar esas olarak iki kategoride toplanır:

- sinyallerin tanınması: söz, biyomedikal, sismik ya da endüstriyel sinyaller.

- Görüntülerin tanınması: basılı karakterlerin

optik olarak okunması, robotbilim, teledeteksiyon, biyomedikal sektör.

Sözcüklerin RAP'ta tanınması konusuna gelince, büyük bir güçlük aynı sözcüğü etkileyen ve iki sözcük arasındaki "akustik mesafe"nin hesaplanmasını nazik bir hale getiren geçici ritm ve süre değişikliklerinden kaynaklanıyor. 1968'de Sovyet ve 1971de Japon araştırmacıları tarafından önerilen, dinamik programlamanın kullanımı bu probleme en uygun çözümü getiriyor. Bu tür algoritmalar artık çok yaygınlaşmış ve orta büyüklükteki sözcük dağarcıklarının tanınmasına (birkaç yüz sözcüğe kadar) çok iyi uyum sağlamış durumdadır.

Sözcüklerin tanınması değişik yönlere doğru yayıldı:

- Özellikle rastlantısal bir değişkenlik modelizasyonunun (Markov'un modeli) yardımıyla çok okuyuculu tanıma;
- Yapay duraklamaları olmayan *sıralanmış* sözcükleri tanıma;
- Global ve analitik yaklaşımlar arasındaki ayrımın artık belirli olmadığı, büyük sözcük dağarcıklarının tanınması.

3. Cümlelerin Anlaşılması ve Sözlü Diyalog - Sürekli konuşmanın tanınmasıyla ilgilenildiğinde kısmi de olsa analitik bir yaklaşım gereklidir: Akustik formun ötesinde, konuşmayla bildirilen mesajı oluşturan dilbilimsel ögeleri yeniden bulmak sözkonusudur. Bu durumda böyle bir mesajın değişik şifreleme ve şifre çözme seviyelerine müdahale eden bilgiler daha açık bir şekilde ortaya konulmalıdır.

Temel bir ilk aşama, ayrı ayrı dilbilimsel birimler olarak sesli akustik sinyali tanımlamaktan ibarettir: ses ögeleri, heceler, vb. Bu akustik-fonetik şifre çözümü, özellikle çok-okuyucularda çok güç ve

sonraki kavrama aşamalarının bağlı olduğu bir işlemdir. Hiçbiri gerçekten tatmin edici olmayan birçok yöntem önerildi. Uzman sistemlerin tekniklerinden, uzun süredir araştırmacıların haksız olarak basit bir FT problemi olarak gördükleri bu şifre çözme işleminde kullanılan bazı bilgileri aydınlatmak için yararlanıldı.

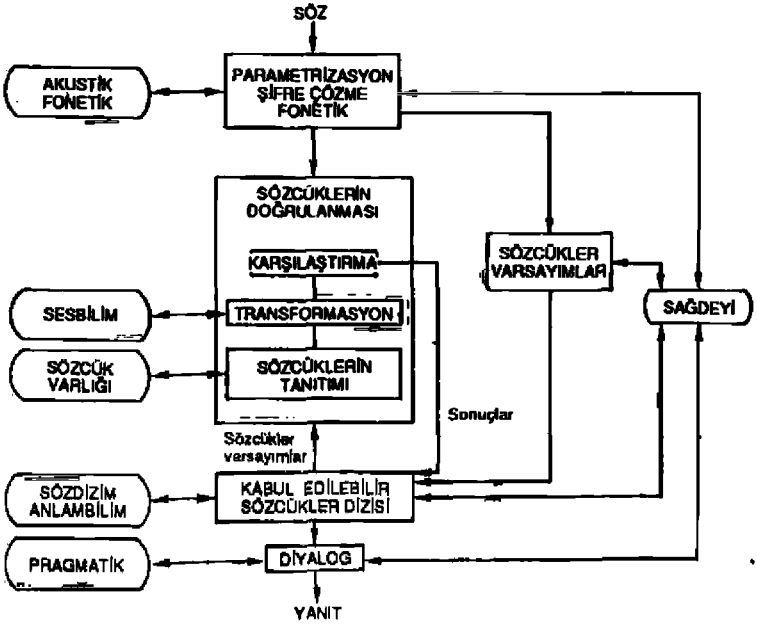
Daha genel olarak, sözün anlaşılması daha önce sözü edilen bilgiler bütününü kullanır. Bir sistemin kavrayışı, uygulamada genellikle birarada bulunan iki üç çözüme başvurabilir:

- *Bu bilgilerin* matematiksel bir model, özellikle Markovcu rastlantısal modeller içine *kör ve örtük entegrasyonu*. Prensip, çok büyük sesli veriler derlemelerinin incelenmesiyle böyle bir modelin parametrelerini düzenleyerek sözün çeşitliliğini yakalamaktır. Bu modeller şu an için elde bulunan etkili modeller arasında yeralırlar fakat aynı zamanda esneklikten uzaktırlar.

- *Belirtik bilgi tabanlı bir sistem anlayışı*.- Daha önce görülen bilgilerin tanıtımı, usavurma ve mimarlık stratejileri sorunlarını ortaya koyar. Konuşma dilinin kullanımı, sisteme entegre edilmesi önemli olan belli bir diyalog seviyesi içerir. Problemin bu yönünü gerçekten hesaba katan pek az sistem vardır, oysa bu, önemli bir araştırma yoludur. Kısacası, kavrama süreci karmaşık ve çok biçimlidir. Şekil 16 akustik-fonetik şifre çözümünden diyaloğa kadar çeşitli bileşenleri gösteriyor.

Çeşitli bilgilerin *en uygun* işbirliğine olanak tanıyan mimari anlayışı, görüldüğü gibi, anadilin kavranması için ya da başka kavrayış alanları için ortaya konulan problemlere benzeyen önemli bir problemdir: yazının görülmesi, anlaşılması. Bilgi kaynaklarının ortak bir veri yapısının aracılığıyla iletişim kurduğu *blackboard* modeli, bu tür prob-

lemler açısından ilginç özellikler sergilerler. Bu, yedinci bölümde ele alınacaktır.



Şekil 16.- Sözün anlaşılması sisteminin bileşenleri.

ALTINCI BÖLÜM
ROBOTBİLİM

I. Giriş

Robotbilim çok biçimlidir. Basit manipülatörden karar alabilen özerk robota kadar çok çeşitli karmaşıklıklara sahip otomatik sistemlerle ilgilidir. YZ ancak ileri sistemlerin kavranışında devreye girer, ama yeni kuşak robotlardaki rolü giderek daha fazla önem kazanmaktadır.

O halde, burada, dünya içinde ve üzerinde etkili bir *robot, zeki bir makine* yapmak için yürütülen çalışmalarla ilgileniyoruz. Örneğin önüne konulan hedefi gerçekleştirmek için yapılacak hareketlere karar vermek üzere çevresini yorumlaması gereken özerk bir robot sözkonusudur (çok iyi belirlenmiş tekrara dayalı görevleri yerine getirmek üzere jestleri "programlanmış" bir sanayi robotu değil).

Üstesinden gelinmesi gereken çok sayıda problem var. Bunu anlamak için, basit bir çevrede, fonksiyonu eşyaların yerini değiştirmek olan ve verili bir anda bir eşyanın yerini değiştirme görevi verilmiş bir robot düşünün.

Gerçekleştirilecek işlemlerin sırası şöyledir:

- Eşyayı bulmak: Bu, kamerasının yön saptamasını, algılanan imajın analiz edilmesini, kameranın yöneliminin düzeltilmesini gerektirir... Robot eşyayı tanıyana ve yerini belirleyene kadar bu devam eder.

- Eli hareket ettirmek: Robotun tutma organı hareketli eklemler sayesinde çeşitli derecelerde öz-

gürlüğe (olası yer değiştirmelere ve döndürmelere uygun olarak) sahip olabilen bir *effecteur*'dür.

- Eşyayı yakalamak: Robot temasın niteliğini değerlendirebilmelidir. Bu, algılayıcılar sayesinde yapılır. Örneğin bir kağıt parçası ile teması değerlendirmeyi ve onu yumuşak bir şekilde havada yakalamayı becerebilen "yapay deriler" yapılabilir.

- Eşyanın yerini değiştirmek: Bu işlem robotun boş bir yol bulmasını ve karşılaştığı güçlüklerin üstesinden gelmesini (engeller, eşyanın konumunun sapması vb.) gerektirir.

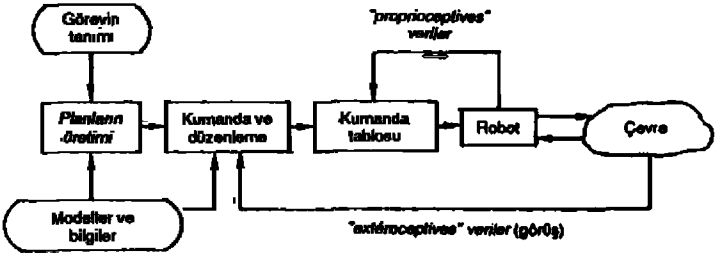
Dolayısıyla kendimizi,

- algılama
- yapılacak hareketlerin planlanması
- yerine getirme

fonksiyonlarına denk düşen bir işleyiş devri karşısında buluruz; ayrıca zorunlu olarak her fonksiyonun diğer ikisi üzerinde karşı tepkisi sözkonusudur.

YZ'nin robotbilimdeki rolü bu sistemlerin becerikliliğini arttırmaktır. YZ'nin katkılarını planlama ve görme konularında gözden geçireceğiz.

Şekil 17 *üçüncü kuşak* denilen bir robotun şemasını vermektedir.



Şekil 17.- Üçüncü kuşak bir robotun şeması

II. Planların Üretilmesi

1. Giriş - İkinci bölümde, planlama konusunda sık sık aktarılan maymun ve muz problemini anlatmıştık. İstenilen özerklik, maymunun (ya da genel olarak sistemin) dış bir müdahale olmaksızın, kendi eylem planını, yani bir başlangıç durumundan kalkıp bir hedefe ulaşma olanağı veren temel işlemlerin ardışıklığını belirlemeyi başarmasını gerektirir. Bu görevi üstlenen programa *eylem planları üreticisi* ya da *planlamacı* denir.

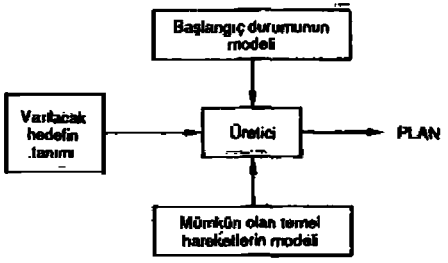
2. Uygulamalar - Planlama tekniği, robotbilim problemleri dışında, çeşitli alanlarda uygulanır:

- bir stratejinin kullanılmasını gerektiren oyunlar;
- üretim yönetimi: örneğin kumanda aletinin yönetimi (işlemlerin zaman çizelgesi, gerekli kaynaklar), bir atölyenin ya da bir işletmenin bütün işleyişinin planlanması;
- görevlerin planlanması (örneğin uzay mekiklerinin yeniden biçimlendirilmesi, bakım işlerinin planlanması, imalat dizilerinin kavranması ya da moleküler genetik ya da organik sentez deneylerinin planlanması vb.).

3. Bir Plan Üreticisinin Yapısı - Bir plan üreticisi ortaya konulan problemi çözümlemek için belirli türde bilgileri hesaba katan bir YZ sisteminin yapısına sahiptir. Şekil 18 bir plan üreticisinin temel bileşenlerini göstermektedir.

Üretici, bir durumlar alanının gelişim tabanı ilkelerine ya da problemin alt problemlere indirgenmesine (ikinci bölümde incelendiği gibi) başvuran çeşitli yöntemler kullanarak işleyebilir.

Örneğin, tıpalı ya da tıpasız şarap şişeleri, bardaklar ve içki testilerinin bulunduğu ve eylemlerin kapları boşaltmak ya da doldurmaktan ibaret oldu-



Şekil 18.- Bir plan üreticisinin yapısı

ğu bir ortamda olduğumuzu düşünelim. Bununla birlikte gerçekliğin hiç bir zaman bu kadar basit olmadığını ve algılama organlarının sınırlı karar gücü, gerçek dünyanın nesnelerinin tam bir tanım verilmesine olanak tanımayan karmaşıklığı ve robotun eylemlerindeki kusurlar gibi çok sayıda faktörün planlama işini karmaşıklaştırdığını belirtmek gerekir.

Olası temel operatörlerin üç yüklem listesiyle tanımlandığı mantıksal bir formülizmi benimsiyoruz.

- operatörün titizlikle çalışabilmesi için doğrulanması gereken önkoşulların listesi (p)
- operatörün uygulamasıyla gerçek yüklemlerin listesine eklenen yüklemleri kapsayan eklenti listeleri (a)

Operatörler şöyle betimlenebilir:

- boş bir bardağı (y), tıpası çıkarılmış ve n birim şarap içeren (0 <n <9) bir şişe (x) yardımıyla doldurmaktan ibaret olan bardak-doldurma (x, y)

p: şişe (x) ve tıpası çıkarılmış (x) ve seviye (x, n) ve büyük (n, O) ve küçük (n, 9) ve bardak (y) ve boş (y)

a: dolu (y), seviye (x, n-1)

r: boş (y), seviye (x, n)

- boş bir içki testisini (y), tıpası çıkarılmış ve n birim şarap içeren (1 <n <9) bir şişenin yardımıyla içki testisini doldurma:

p: şişe (x) ve tıpası çıkarılmış (x) ve seviye (x, n) ve büyük (n, 1) ve küçük (n, 9) ve içki testisi (y) ve boş (y)

a: dolu (y), seviye (x, n-2)

r: boş (y), seviye (x, n)

- bir şişenin (x) tıpasını çıkarmaktan ibaret olan tıpa çıkarma (x)

p: şişe (x) ve tıpalı (x)

a: tıpası çıkarılmış (x)

r: tıpalı (x)

Bardak doldurma (x, y) gibi bir operatörün uygulanabilir "monoton olmayan" bir kural gibi anlaşılabileceğini belirtelim.

İleri doğru:

x, n birim şarap içeren (0 <n <9) tıpası çıkarılmış bir şişe ise ve y boş bir bardak ise, bardak doldurulabilir. Bu eylem belli sayıdaki başlangıç olgusunu ortadan kaldırır ve bardağın dolu olmasına ve şişenin n-1 birim içermesine yolaçar.

Ya da *geriye doğru:*

B şişesinin boşaltılması, yani (B, 0) seviyesinin elde edilmesi isteniyorsa, bir y bardağı, 1 birim şarap içeren B ile doldurmaya çalışılabilir. Fakat bunun için önümüzde ulaşılacak iki alt hedef vardır: tıpası çıkarılmış (B), seviye (B, 1), bardak (y) ve boş (y).

Böylelikle bir eylem planı belirli bir hedefe ulaşmaya izin veren işlemlerin birbiri ardı sıra gelmesinden oluşur.

4. Bir Plan Üreticisinin İşleyişi - Birçok yöntem plan üretimi probleminin çözülmesine yardım eder.

a) Durumlar aralığının gelişimi

Bu yöntem aşağıdakilerden birinden oluşabilir:

- başlangıç durumunun tanımından yola çıkarak uygulanabilir operatörlerin araştırılmasını, onların uygulamaya konması ve yavaş yavaş çözüm ağacının geliştirilmesi;

- durum-hedeften yola çıkarak ona götüren operatörlerin araştırılması ve ağacın aynı şekilde geliştirilmesi;

- iki yönlü bir araştırma yapılması.

Üçüncü bölümde de görüldüğü gibi çeşitli problemler kendilerini dayatıyor.

- katışımsal patlama riski: Karmaşık problemler için "heuristique"ler tarafından yönlendirilen iyi bir çözüm araştırma stratejisinin kullanılması gerekir.

- teorik olarak grafik durumlarının istiflenmesini gerektiren hafıza-kapasitesinin önemi: Sadece başlangıç durumunun ve operatörün her uygulanışındaki ekleme ve eksiltme listelerinin tanımını koruyarak çözümlenen *frame-problem*'dır. Verili bir durumda doğru ve yanlış yüklemleri bulmak için başlangıç durumundan bu özel duruma götüren yolu izleyen bir algoritmayı izlemek yeterlidir.

b) Amaçların ve araçların analizi: Bu yöntem yinelemeli olarak aşağıdakilere döner:

- uygulamanın spesifik bir değerlendirme fonksiyonu sayesinde içinde bulunulan durum ile taşınan amaç (bir amaçlar yığınının tepesine yerleştirilmiş) arasındaki farkları hesaplamaya;

- bu farkı azaltabilecek bir operatörü seçmeye;

- bu operatörün önkoşullarını yeni alt-amaçlar olarak almaya ve onları amaçlar yığını içinde istif etmeye;

- alt-amaçlar doğrulanırsa ve üzerinde çalışılan operatör uygulanabilir hale gelirse, onu içinde bulu-

nulan duruma uygulamaya ve süreci izlemeye.

Bu algoritma daha önce sözünü ettiğimiz GPS sistemine aittir. İçinde bulunulan bir durum ile taşınan hedef arasındaki farkların, amacı betimleyen ve içinde bulunulan durum içinde doğrulanmayan yüklemlerle belirlendiği STRIPS (Fikes ve Nilsson) sistemlerde uygulama olanağı buldu.

Amaçların ve araçların analizi, bir çözüm varsa bile her zaman bu çözümün bulunmasını sağlamaz. Gerçekten, kendimizi içinde bulunulan durumda gerçekleşmiş olmayan ve hiçbir operatörün gerçekleştirilmesine olanak tanımadığı bir alt-amaç karşısında bulabiliriz. Ya da belirli bir alt-amacın gerçekleştirilmesi olgusu önceden gerçekleştirilmiş bir amacı yıkar ve bu amacın yeniden kurulması birinciyi yıkar: Karşılıklı olarak birbirine bağlı alt amaçlardan sözediyoruz. Örneğin küplerin manipülasyonu durumunda bir istiflemeyi yerine getirmek için, önceden yapılmış bir istiflemeyi bozmak ve bunun tersini yapmak gerekir.

Seçim noktaları üzerinden geriye dönüşler yoluyla işin içinden sıyrılmaya çalışılabilir: Verili bir anda birçok operatör aday olduğu zaman ya da bir önkoşulun alt-amaçlara ayrıldığı noktada. Bu girişimler bazen problemin çözümlenmesini sağlarlar.

Böyle bir amaç, bir çözüm sağladığında, katışımsal patlamayı azaltma avantajına sahiptir. Kusuru, a)'da belirtilen yöntemde olduğu gibi "aşırı-zorlama" karakteridir: Nihai planda ortaya çıktığı şekliyle olayların kronolojisini akılda tutarak problem çözülmeye çalışılır.

a) Diğer yöntemler, işlemlerin yerine getirilme sırasını belirleme anını geciktirerek, bu zorlama kusurundan kaçınırlar. Bunlar arasında özellikle STRIPS düzenlemelerinden doğanlar ya da hiyerarşik plan üreticilerinde kullanılanlar sayılabilir.

Örneği NOAH (Sacerdoti) sisteminde, başlangıçta planın gevşek terimlerle tanımlandığı yüksek bir soyutlama düzeyine yerleşilir. Daha sonra genel bir eylemin daha ayrıntılı bir eylemler şebekesine ayrıştırılmasına çalışılır ve gereksiz eylemler araştırılarak ve alt eylemlerin olası kronolojisi hesaba katılarak, elde edilen plan eleştirilir. Bu, çizgisel bir plan elde edilene kadar sürdürülür.

d) Planlama konusundaki güncel eğilimler şunlarla ilgilidir:

- "zaman" parametresinin hesaba katılması;
- daha etkili bir kontrol amacıyla insani planlama etkinliği üzerinde düşünme;
- plan bölümlerinin planlamacı tarafından öğrenilmesi; bu onun usavurmada kısaltmalar yapmasına olanak sağlar;
- zor planlama durumunda bir uzmanla ilişki kurulması;
- sapma olduğu durumda uygulama süresince planın dinamik bir şekilde yeniden ortaya konması.

III. Görme

1. Giriş - Bilgisayar tarafından görmenin amacı bir görüntünün ya da bir olayın yorumlanması ve bu yorumun çevre üzerindeki bir eylemle (eşyaların kullanılması, bir taşıtın yerinin değiştirilmesi, vb.) ilişkisinin kurulmasıdır. Bir görme sistemi her zaman, başta görüntülerin işlenmesi ve formların tanınması olmak üzere, başka tekniklere başvuran modüller içerir.

Görüntü işlemi, görüntüleri çok boyutlu sinyaller olarak görür ve onları daha uygun ya da görselleştirme, iletme ya da analize daha iyi uyarlanmış özellikler sergileyen başka görüntülere dönüştürür.

Formların tanınması, daha önce gördüğümüz

gibi, nesneleri, bu nesneler üzerinde uygulanan ölçümlerden kalkarak tanımlamaktan ibarettir. Bu, görmede bir taban etkinliğidir ve özellikle üç boyutlu işlemde, bir nesnenin algılayıcılara gösterebileceği çeşitli yönler olgusundan kaynaklanan büyük güçlükler getirir.

Numaralandırılmış görüntüler analizi '50'li yılların sonuna doğru başladı ve öncelikle çokyüzlü nesnelerden oluşan basit üç boyutlu sahnelerin (kısaca 3B) yorumlanmasıyla ilgilendi. '70'li yıllarda, özellikle "düşük seviyeli" algoritmaların (görüntünün) etkililik eksikliğini telafi etmek üzere, yorumlanacak sahneye ilişkin sembolik bilgilerin kullanımını vurgulayan YZ teknikleri ortaya kohmaya başlandı. Bölümlenme paradigmasından yola çıkan bir dizi deney gerçekleştirildi; bu paradigmaya göre görme süreci görüntünün düşük seviyedeki bir bölümlenmesi ile bölümlenmenin sonuçlarının üst seviyede bir sembolik yorumu arasındaki karşılıklı etkileşimden ibarettir. Ancak bu bakış açısı beklenen sonuçları vermedi.

"Modern" bilgisayarla görme iki anahtar düşünceye dayanır:

- paralel hesap için yeni yapıların gerekliliği;
- fiziksel sinyallerden bilişsel yorumlara kadar bir tanıtım hiyerarşisinin varlığı.

Bilgisayarla görme sanayi alanında YZ'nin ilk kullanımlarına dahil olmuştur ve sistemlerin görme kapasitesi hâlâ insanın görme kapasitesinden uzak olduğu halde, bu alandaki sistemlerin sayısı giderek artmaktadır. Temel uygulamalar otomatik denetim ve sistemlerin otomatik yönetimiyle ilgilidir.

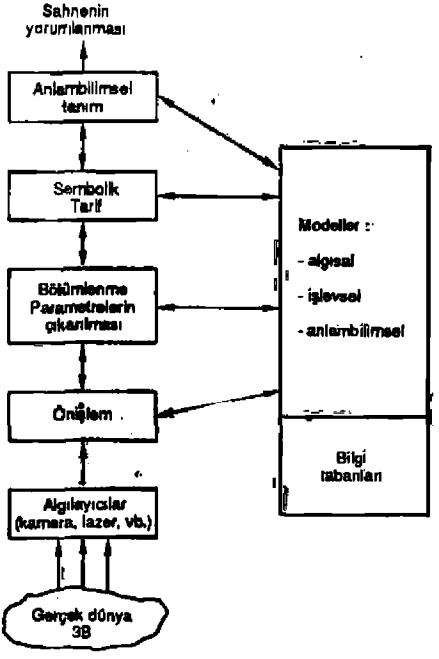
2. Temel Özellikler - Bilgisayarla görme sistemlerinin kurulması çeşitli basitleştirmeleri gerekli kılar.

Şekil 19 bir görme sisteminin genel yapısını ve-

riyor.

Görme sürecinin çeşitli aşamaları vardır.

- görüntünün önişlemi ve parametrelerin çıkarılması, nesnelerin sınırlarının yapısının, dokuların (konturlarla sınırlanmış alanlar içinde, lokal yoğunluk ya da renk değişimlerini belirler), mesafelerin (bir nesnenin gerçek formunu hesaplamak için önemli) ve nihayet nesneler ile sahneyi gözlemleyen arasındaki (hareketli robotların gelişimi çerçevesinde) hareketlerin ve dinamik etkileşimlerin belirlenmesine hizmet eden konturlar üzerine bilgiler sağlar (hızlı değişimler ve belirtici ışık yoğunlukları).



Şekil 19. Bir bilgisayarla görme sisteminin genel yapısı.

- birbirini izleyen soyutlama seviyelerinde anlambilimsel bir tanım hazırlanması.

Bu etkinliklerin bütünü çok sayıda ve çeşitli (nesne, sahne modelleri, fiziksel bilimler, vb.) bilgiye başvurur.

Yorumlama süreci lineer bir biçimde gelişmez. Çeşitli işlem seviyeleri arasındaki karmaşık etkileşimler YZ'de iki genel klasik şemaya aittir:

- Giriş verilerinden varsayımlar türetmekten ibaret olan aşağıdan yukarıya (ascendante) ya da verilerle yönlendirilen bir strateji. Sadece işlemden geçen görüntülerin türü üzerindeki zorlamalar ve uygulama ortamı üzerindeki önemli sınırlamalar bu şekilde sahnenin bir yorumuna ulaşmaya izin verir;

- Önceden belirlenmiş modellere başvurarak sahne üzerindeki *a priori* varsayımları doğuran, yukarıdan aşağıya (descendante) ya da bir sahne modeliyle yönlendirilen bir strateji. Bu varsayımlar düşük seviyelerden gelen enformasyonla ya da aşağıdan yukarıya görüntüden çıkarılan varsayımlarla karşılaştırılabilir.

Genellikle iki strateji birlikte kullanılmalıdır. Alt seviyeler ile üst seviyeler arasındaki etkileşimler de önemli bir araştırma alanı oluşturur.

3. Metodolojik Yönler - İşlemin farklı seviyeleri farklı tekniklere başvurur:

a) Görüntünün elde edilmesi: Görüntü farklı türlerde alıcılarla (video kameralar ya da CCD, lazer tarama, vb.) numaralandırılır. Böylelikle, görüntünün her noktası için ortak gri seviyesini ve gerektiğinde rengi kapsayan ve *pixel* denilen bir elemanlar matrisi tarafından temsil edilir;

b) Görüntünün ikilileştirilmesi: Bugünkü bilgisayarla görme uygulamaları her *pixel*'in siyah ya da beyaz olduğu ikili görüntüleri işler. Böyle bir görüntü, görüntünün gri seviyesinde eşiklenmesiyle elde

edilir. İstenilen hesap gücü ve hafıza kapasitesi büyük ölçüde indirgenilir; bu, özellikle aydınlatma koşulları sağlanabildiğinde (örneğin denetimi yapılacak nesneler aynı oranda aydınlatılmış bir zemin üzerine yerleştirildiğinde), ikili görüntülerin görsel denetim gibi basit uygulamalar için yaygın kullanımını açıklar;

c) Konturların bulunması: Uygun bir ekle birleştirme yoluyla, nesnelerin yerini belirleme ve onları tanıma için önemli olan konturları bulma olanağı verir. Son onbeş yıl süresince bu konu üzerine önemli bir çalışma yürütüldü.

d) Görüntünün bölgelere ayrılması: Bir bölge, belli türdeşlik, gri seviyesi, doku, form, renk vb. kriterleri sergileyen bir alana denk düşer. Bir görüntüyü bölgelere ayırmak için iki yaklaşım önerildi: Bunlardan biri konturlara dayanıyor; diğeri bir türdeşlik kriterini karşılayan *pixel*'leri yeniden gruplamaya çalışır. Bu, ya bazı *pixel*'lerden başlayarak ve benzer pixel'leri biraraya getirerek yükselen biçimde ya da büyük bölgelerden başlayarak ve onları daha küçük bölgelere ayırarak azalan biçimde yapılabilir. Bu iki strateji birbirini tamamlayıcıdır ve sonuçların doğruluğunu sağlamak için birleştirilebilirler. Gerçeklikte, konturlar için olduğu gibi, verilerin sesli olması ve determinist olmaması olgusu, görüntü, bağlam, uygulama vb. üzerine "heuristique" bilgilerin hesaba katılmasını gerekli kılar. Bilgi tabanlı sistemlerin problemler içinde bir rolü olduğu açıktır. Örneğin, uzman sistemler bu düzeyde başarıyla denenmiştir.

e) Nesnelerin tanınması: Bir nesnenin bir sahne içinde tanımlanması bir sahnenin yorumlanmasının önemli bir aşamasıdır ve sürekli konuşmanın anlaşılmasında sözcüklerin tanınması süreciyle karşılaştırılabilir. Nesnelerin (ya da nesne sınıfları-

nın) önceden kaydedilmiş modellerinin görüntüden çıkarılan parametrelerle uyum içine sokulmasından ibarettir. Modelin tanımı genellikle görüntü içinde kullanılabilen enformasyondan daha eksiksizdir, çünkü nesneler yorumlama sisteminin bakış açısından ancak kısmi olarak görülebilirler (görüntünün 2 boyutlu gerçekleştirilmesinde 3 boyutlu bir modelin gizli yönü, başka nesnelerle kısmen gizlenen yönler, nesnelerin çerçeve dışında kalan kısımları, formların çıkarılması sürecindeki hatalar vb.). Bundan dolayı, bir nesnenin tanınması ender olarak deterministtir, bunun da genel yorumlama süreci üzerinde etkisi vardır.

Nesne modelleri grafikler ve hipergrafikler gibi ilişkili veri yapıları ile tanımlanabilirler. Bu durumda, bir nesnenin tanınması iki grafik arasındaki en iyi uyumun bulunmasına gelir. Nesnelerin jenerik sınıflarını tanımlamak üzere, başka tanıtımlar da kullanılabilir, örneğin *frame*'ler.

f) Yorumlama teknikleri: Bunları aşağıdaki paragrafta geliştireceğiz.

4. Yorumlama Aşaması - Bu, bilgisayar tarafından görmenin son aşamasıdır ve bu aşamada bir görüntü ya da bir sahnenin anlambilimsel yorumu türetilir.

Katışımsal patlamadan kaçınmak için etkili araştırma stratejileri uygulamak gereklidir. Sözün anlaşılmasında da olduğu gibi en popüler "heuristique" metodlar, yorumlamanın her aşamasında sadece en iyi kısmi çözümü kapsamaktan ibaret olan *ilk önce en iyinin araştırılması*, sadece en çok umut vaadeden n çözümleri geliştirmekten ibaret olan *demet halinde araştırma* ya da usavurmanın güvenilir bir ya da birçok nokta üzerine dayandığı *güven kümeleri* tarafından yürütülen strateji (örneğin görüntüde iyi tanınan nesneler).

Tahmin-doğrulama (ya da varsayım-ve-test) paradigması aynı şekilde devreye sokulmuştur: Görüntüde özel nesnelerin bulunup ortaya çıkarılması, yani araştırmanın, nesnelerin bulunma şansının en yüksek olduğu sınırlı alanlar üzerinde odaklaştırılması sözkonusudur. Bu teknik, yukarıdan aşağı ya da aşağıdan yukarı yorumlama metodlarına bağlanabilir.

Görüntünün yorumlanması ancak eldeki bütün bilgilerin, başlangıç verilerinden çıkarılan bilgiler olduğu kadar uygulama alanı üzerindeki *a priori* bilginin de hesaba katılmasıyla başarılabilir. Örneğin bir hava fotoğrafında küçük bir kare bir araba olarak yorumlanabileceği gibi bir ev olarak da yorumlanabilir. Alan hakkındaki iyi bir bilgi, dar bir alanda yeralan bir küçük dikdörtgenin araba olma şansının daha yüksek olduğunu çıkarsama olanağı sağlar.

Kilit nokta birçok alıcıdan gelen verileri (ki bazılarının görüntüyle bir bağlantısı olmayabilir, temas algılayıcılarında olduğu gibi) entegre eden bilgi tabanlı mimariler kavramıdır ve bütün kullanılabilir bilgi kaynaklarından yararlanmaktır. *Blackboard* mimarisi burada da çok sayıdaki algılayıcıdan gelen verilerin kaynaşması ve *çokbilgi tabanlı* usavurma için ilginç bir çerçeve doğurur.

*5. **Pratik Uygulamalar*** - Tam görme sistemleri şu anda sadece laboratuvar maketleri durumundadır. Bu arada, görüntü işlemi ve bilgisayar tarafından görme, endüstriyel uygulamalarda yaygın olarak kullanılmaktadır. Günümüzde ticarileştirilmiş sistemlerin görme kapasiteleri henüz çok basittir ama alan gelişme halindedir ve yerleşik sistem sayısı düzenli olarak artmaktadır.

İki temel uygulama alanı şunlardır:

- *Denetleme,* güncel sistemlerin yüzde 90'ını

karşılıyor. Gelecek on yıl içinde ürün denetiminin yarısının otomatik olarak yapılacağını düşünmek makuldür. Bu problemin en basit envanter işlerinden kalite kontrolü için nesnelerin ya da nesne bölümlerinin incelenmesine dek uzanan değişken bir karmaşıklığı vardır. Genel olarak iki denetleme görevi sınıfı vardır: nicel denetleme (ya da görsel metroloji) ve nitel denetleme (ya da kalite kontrolü).

- *Rehberlik,* aynı şekilde değişik karmaşık görevleri kapsar: Montaj hatları üzerindeki nesnelerin tutulma ve yerleştirilme ya da kaynak işlemlerinden engellerin olduğu durumda vs. hareketli robotlara rehberlik etmeye kadar. Bu sektör gelecek yıllarda giderek artan bir önem kazanacaktır.

Birçok güncel sanayi sistemi ikili görüntüleri kullanır ki bu, özellikle aydınlatma koşulları kontrol altına alınabildiğinde, bazı basit operasyonlar için yeterlidir. Gri düzeyde görüntüleme, entegre devre maliyetlerinin azaltılması ve yöntemsel gelişmeler sayesinde, endüstri dünyasına girer. Karmaşık durumlarda nesnelerin tutulması ya da bir robotun bütün rehberliği gibi görevler için kaçınılmaz olan üç boyutlu görme de (gerek lazerli dedektörler gibi etkili alıcılar aracılığı ile gerekse iki gözlü görme sayesinde) buna katılacaktır.

YEDİNCİ BÖLÜM

BİLANÇO VE PERSPEKTİFLER

I. Giriş

YZ *logiciel*, materyel ve mimarlığı da ilgilendiren çeşitli disiplinlerdeki önemli gelişmeleri de takibeder. Bu bölüm, bu gelişmeler hakkındaki enformasyonlar bütününü toplamaktadır; şu bağlamda:

- *logiciel ve materyeller:* YZ dillerinin ve makinelerinin büyük özelliklerini vermekle ve taban *logiciel*'inin gelişimini belirtmekle yetineceğiz.
- *gelişmiş temalar:* YZ hakkındaki güncel çalışmalar gelecek vaadeden bir yollar bütününe göre yürütülmektedir. Daha önce işlenmiş olan, bilgilerin tanıtılması ve usavurma yolundaki temel problemden tekrar bahsetmeyeceğiz. Buna karşılık üç önemli tema tanıtacağız: sembolik öğrenme, bilgilerin çok-etkenli ve çok-tabanlı mimarlığı, bağlantılı modeller.

II. Logiciel Görünümler

1. Diller - YZ uygulamaları, Pascal veya C gibi klasik dillerde hatta, özellikle sistemin gerçek zamanda uygulama gerektiren bölümleri için, toplayıcı (assembleur) dilde programlanabilirler. Bu arada bazı diller özelliklerinden ötürü bilgilerin tanıtımı ya da usavurmadan kaynaklanan sorunlara daha özel olarak uyarlanırlar. Bilhassa aşağıda genel bir özetini verdiğimiz PROLOG, LISP ve "yönlendirilmiş nesne" dilleri sözkonusudur.

a) PROLOG: '70'li yılların başında Marsilya'da A. Colmerauer tarafından tasarlanmış olan ve birçok türü olan bu dili üçüncü bölümde ele almıştık. Burada sadece önemli noktaları belirteceğiz:

- Horn "şart"ları ile sınırlandırılmış olan birinci derecedeki yüklemler mantığına dayanan, yöntemsel olmayan dil;
- entegre birleştirme mekanizması;
- entegre çözümleme mekanizması (daha önceden tanımlanmış bir stratejiyi izleyerek geri dönüşlü geri zincirleme);
- bir programda şartların önem sırası;
- bir mantıksal usavurumdan doğan problemlerin uzman sistemlerin yazılımına uygulanması;

b) LISP: 1958'den itibaren tamamen J. MacCarthy tarafından tasarlanan LISP Fortran ya da COBOL çağında ortaya çıkmıştır, ama bilimsel hesabın ve yönetim enformatiğinin üstün geldiği dönemde çok az tanınmıştır. Gelişmesini YZ'nin ortaya çıkmasına ve özellikle çok sayıdaki diyalektlerinden biri ile yazılan uzman sistemlerin artışına borçludur. Büyük makineler ya da özgül makinelerden mikro bilgisayarlara ya da "LISP yonga"larına kadar bütün malzemelerde kullanılabilmektedir. LISP'in belli başlı özellikleri şunlardır:

- fonksiyon kavramına dayalı programlama, modüler ve yayılabilen *récurcivité;*
- bütün tanıtım veya usavurum modellerinden bağımsız ve sembolik işlem uygulanan "üstdillerin" gelişmesine olanak sağlayan genel dil;
- tanıtım birliği: bir program ya da bir verinin her ikisi de sembolik ifadelerdir;
- zengin programlama donanımı: karşılıklı etkileşim, tam sayfa printer, ayarlama aletleri;
- otomatik teorem kanıtlamasına, anadilin anlaşılmasına, planlar geliştirmeye, sözdizimsel anali-

ze, çıkarsama motorlarının yazılımına vs.'ye uygulanma;

- PROLOG ile birleştirme imkanı, hatta tamamlayıcı özelliklerinin LOGLISP (Birleşik Devletler) ya da LISLOG (CNET-Fransa) gibi dillerde birleştirilmesi.

c) "Yönlendirilmiş nesne" dilleri (LOO): Üçüncü bölümde belirtildiği gibi, soyut türler üzerindeki (türlere bağlı operasyonlarla tiplendirilmiş enformasyon kavramı) özelliklerin geçmesi ile birlikte anlambilimsel şebekeler ve *frame*'ler üzerindeki (simge-değer çiftleri ile deklaratif tanıtım) çalışmalardan doğmuşlardır. Aşağıdaki özelliklere sahiptirler:

- bilgilerin pasif simgelerin (deklaratif yön) ve metodların (yöntemsel yön) bağlı olduğu nesne sınıfları ve alt sınıfları şeklinde tanıtılması. Böylelikle bir nesne veriler kümesini ve bu verileri manipüle etmekle görevli yöntemleri içerir;

- mesajların nesneler arasındaki geçişi sayesinde denetlenen özelliklerin kalıtımını devreye sokan usavurum;

- gıyabi usavurma, emisyon ve varsayımların doğrulanma olanağı;

- nesne sınıflarının ve bağlantılı yöntemlerin yeniden kullanılması ile büyük *logiciel*'lerin hızlı maketlenmesinde etkinlik. Kesinlikle yeni olan LOO'da programlama verili bir problem için bir sınıflar kümesi geliştirme anlamına da gelir.

Örnek olarak bazı çalışma istasyonları için programlama çevresi olarak tutulan SMALLTALK (yaygın hale gelmiş olan bilgisayar ekranları üzerinde "çok pencereleme" kavramının başlangıcında) ya da taban dillerinden türetilmiş olan bir diller kümesi verilebilir:

* enformatikte genel olarak kullanılan diller:

Le-LISP'ten (LISP'in Fransızca diyalekti) yola çıkarak CEYX C++ veya C'den türemiş C Hedefi, PROLOG'dan türemiş OBJLOG veya ESP (Japon);

* ve özgül bilgi tanıtma dilleri (LOOPS, LOOKS, KOOL vs.).

2. Taban Logiciel'leri - YZ sistemlerinin, özellikle de uzman sistemlerin gelişmesinin genellikle programlama dillerinin üstünde yer alan *logiciel* araçların yardımıyla gerçekleştiğini görmüştük. Bu araçların gelişimi, iki görünüm ortaya koyar:

- ilki büyük bilgi tabanlarının manipülasyonuna bağlıdır: yeni tanıtım ve usavurma, doğrulama ve tabanların tutarlılığının sürdürülmesi, arka arkaya gelen versiyonların yönetimi, ayrı ayrı geliştirilmiş sistemlerin kooperasyonu. Böylece gelecekte gelişmesi gereken bilişsel ayrım atölyesi kavramına ulaşılmıştır.

- ikincisi klasik enformik görünümleri entegre ederek YZ sistemleri ile kullanıcıların enformik çevrelerindeki üstün sistemler arasında bir ilişki sağlama tasasını ortaya koyar. Aynı şekilde değişik programlama dilleri, veri tabanı yönetimi sistemleri, *logiciel* kütüphaneleri, kullanım sistemleri, telematik *logiciel*'ler vs. anında hesaba katılırlar.

III. Malzeme Görünümleri

1. Giriş - Sayısal hesaplar için tasarlanmış olan günümüz bilgisayarları sembolik işleme geniş ölçüde başvurarak YZ uygulamalarına kötü uyarlanmışlardır. Bu problem genellikle, daha önce de görüldüğü gibi spesifik YZ dilleri kullanılarak, *logiciel* çözümler yardımıyla çözümlenmeye çalışılır. Ama asıl çözüm yolu yeni makinelerin yaratılmasıdır.

Bir bilgisayarın gücünü *Mips* ya da *Mflops* (saniyede gerçekleştirilen operasyon sayısı) ile karak-

terize etmek yaygındır. Bu ölçü YZ'de çok az açıklayıcıdır, bu nedenle YZ'de genellikle *Lips* kavramı (saniyedeki mantıksal çıkarsama sayısı) terih edilir. Şu andaki performanslar on Lips'den (bir mikrobilgisayar) yüzbinlerce Lips'e kadar (YZ'ye en iyi uyarlanmış makineler) uzanırlar.

2. YZ Makineleri - Her ne kadar, özellikle üçüncü dereceden uygulamalar (bankalar, sigorta) için, bu yönde yeni bir eğilim ortaya çıkmışsa da, YZ merkez görünümlü büyük bilgisayarları çok az kullanır. Güncel gelişmelerin hemen hemen tamamı üç malzeme üzerinde gerçekleştirilirler:

- Önemli güçteki, oldukça pahalı, hem gelişmeler için hem de servis merkezleri olarak (örneğin Minitel ile vs'lere girmek için) kullanılan *süper-mini-bilgisayarlar.*

- Gerek 32 yonga güçteki mikroişlemciler gerekse özellikle etkili RİSC mimarlıkları (sınırlı bilgi oyunlu) etrafında tasarlanmış *iş makineleri.* Bu kişisel iş makineleri yüksek tanımlı grafik ekranlarına ve özellikle zengin sistemlerin geliştirilmesi için bir *logiciel* çevreye sahiptirler. Bu makineler ile yüksek dizili mikro-bilgisayarlar arasında varolan mesafe tedrici olarak azalma eğilimindedir.

- Klasik mimarlık bilgisayarları *sembolik makineler* ancak bunların, sembolik dillerin (özellikle LİSP) kullanımında etkinliklerini arttırmaya yönelik özellikleri vardır. Bu özellikler şunları kapsarlar:

* merkezî hafızalar: çok büyük boyutta, tipleştirilmiş (her kelime-hafıza, istiflenen enformasyondan başka bu enformasyonu karakterize eden bir betimleyici kapsar);

* işlem organları: hafızanın kablolu yönetimi, sembolik bilgilerin kullanımı için mikro programlar;

'* kullanıcı ile karşılıklı etkinlik: grafik araçlar ve erişilmez bir zenginlikteki *logiciel* çevre.

Teknolojik gelişme, bu değişik makine türleri arasındaki ayrımı giderek daha aza indiriyor: mikro-bilgisayarın gücünün artması, büyük kapasiteli hafıza bileşenleri, mikro-bilgisayarlar için sembolik işlemci, sembolik VLSI bileşenleri ("bir yonga üzerindeki LİSP makinası") vs.

2. Yeni Mimariler.- Yukarıda adı geçen bütün makineler Von Neumann'ın klasik mimarlığına dayanan bilgisayarlardır. Daha yüksek derecede bir paralelliği olan ve ***Mégalip***'lere ulaşacak ve onları geçecek şekilde sembollerin manipülasyonuna uyarlanan, tamamen yeni mimarilerin tasarlanması için çalışmalar sürdürülmektedir. Bu, büyük ülkelerin bir çoğunda (Japonya, Birleşik Devletler, Avrupa) yürütülen *beşinci kuşak* makineleri projelerinin hedeflerinden biridir.

Prototipler çalışmakta, ilk ticari oluşumlar ortaya çıkmaktadır ama, bu 21. yüzyıl makinelerinin yaygınlaştığını görmeden önce daha çok çaba sarfedilecektir.

IV. Gelişmiş Temalar

1. Otomatik Öğrenme - YZ sistemlerinin kuruluşunun, genellikle önemli boyuttaki bilgi tabanının yerleştirilmesini gerektirdiğini görmüştük: Bu operasyon, alanın uzmanlarının bilgisinden nazik bir çıkarma yapma aşamasını ve bu enformasyonun kullanılmasına olanak sağlayan bir formalizm içinde yapılaştırılmasını gerektirir. Bu süreç önemli çabalar istemekte ve en azından: uzman ve *cogniticien*, iki temel aktörü devreye sokmaktadır.

Bununla birlikte, şimdilik hiçbir makine insanda bulunan bir doğal davranışı bulundurma kapasi-

tesine sahip değildir: Dünya hakkındaki bilgisini çoğaltmak ve deney sayesinde yeni durumlar karşısında tavır geliştirmek.

Sistemleri *otomatik sembolik öğrenme* kapasiteleri ile donatma çabaları hâlâ araştırma alanının kapsamı içindedir. Özellikle, sistemlerin kullanımının, önerilen çözümleri onaylayan veya reddeden bir uzman kullanıcı ile güçlü bir etkileşimi gerektirdiği sektörlerde uygulamalar başlamıştır.

Kendisine yüklenilen işlevi yerine getirmesi için bir sistemin kapasitesinin iyileştirilmesi çeşitli biçimler alabilir:

- veri tabanının genişletilmesi;
- varolan bilginin geliştirilmesi ya da daha iyi biçimlendirilmesi;
- sorulan problemleri çözümlemeye (insanda çok sık olan usavurma kısaltmaları gibi, daha performanslı araştırma kontrolü stratejileri ve "heuristique"ler sayesinde), belirleyici noktalar üzerinde odaklaşmaya ve gereksiz bilgileri elemeye yönelik daha üstün yetenekler;
- sistemin yetke alanında bir artış vs.

Otomatik sembolik öğrenme konusunda farklı eğilimlerle karşılaşılıyor. En temel olanlar aşağıdakilerdir:

- Örneklerden ve karşı-örneklerden başlayarak, onlardan kurallar, yasalar, genel düzenleme şemaları çıkaracak şekilde tümevarım ile öğrenme. Aynı zamanda gözlem ile öğrenme de sözkonusu olabilir - ki bu durumda sistem gözlenen verileri sınıflandırabilmek için onların ortak noktalarını bulmak zorundadır: *Kavramsal sınıflandırma*'dan sözediliyor.
- Tümdengelim ile öğrenme: genel kurallardan yola çıkarak özgül olayların saptanması, eskilerden yola çıkarak genel yeni kuralların oluşturulması, stratejilerin geliştirilmesi vs.

- Benzeşimle öğrenme: bir problemin bilgisinin ve benzer bir problemin nasıl çözümleneceğini bulmak için bu problemin çözümlenmesini yapacak yolun kullanılması. Bu yaklaşım öncekilerle birleşebilir.

2. Nöro-Mimetik Modeller - Bilgisayar sayısal ve yinelenen görevlerde çok başarılı oluyorsa da insan beyni sembolik işlemi ilgilendiren her şeyde, gerçek zamandaki çağrışımsal araştırmada (fikirlerin, yüzlerin, sözcüklerin vs.) alt edilemez. Bağlantısal ya da nöro-mimetik yaklaşım, beyni, bir makine üzerinde taklit etmek için değil ancak tamamen yeni modeller tasarlamak için insanın işleyişinden esinlenmeye çalışmaktadır. Bu modeller son derece basit ve sayıları yüksek olan elemanter işlemciler (Eİ) arasındaki bir bağlantıya dayanırlar ("nöron-makineler", aslında beyin nöronlarından bir hayli farklıdırlar). Bu onlara çok ilginç özellikler sağlar:

- Olguların ve bilgilerin *yeri belirtilmemiş tanıtımları:* enformasyon artık, başvurulabilecek hafıza-haneleri kümesi içinde determinist biçimde istiflenmemekte, Eİ'ler arasındaki bağlantı "noktaları"na dağıtılmaktadır, ve bu, bu sistemlere eksik verilerle çalışma olanağı sağlar.

Verili bir modele göre Eİ'lerin *sağlam bir biçimlendirilmesi* (bu özellikle, harici uyarıcıları alan bir giriş katından sistemin cevabını veren çıkış katına kadar ard arda gelen katmanlar halinde). Her Eİ girişte çok sayıdaki başka Eİ'den gelen enformasyonları alır ve kendi çıkış enformasyonunu da bir başka Eİ'ye gönderir. İnsan beyninde çok yüksek düzeyde ara bağlantılar vardır (bir nöron onmilyonlarca başka nörona bağlanabilir).

- Çok sayıdaki Eİ'nin *paralel işleyişi* (örneğin bir katta). Bu yüksek paralellik derecesi insan beyninde de bulunur: yarım sanayiden az bir sürede bi-

linen bir yüz ya da kelimenin tanınma olgusu (nöronların göreceli "yavaşlığı" hesaba katıldığında bu yüz kadar işlem adımına tekabül etmektedir) ancak nöronların paralel işleyişi ile açıklanabilir.

- Örneklere dayalı bağlantı noktalarını, muhtemelen yinelemeli bir şekilde değiştirme yoluyla yeni bilgileri *öğrenme kapasitesi.* Değişik öğrenme algoritmaları önerilmiştir: hata gradyanının *retro-propagation*'u bir maliyet istatistiği fonksiyonunun minimize edilmesi vs.

1943'ten beri, Mc Culloch ve Pitts nöronları eşikli otomatlarla biçimlendirmeyi öneriyorlar. '60'lı yıllarda ortaya çıkan *perceptron* (Rosenblatt) ilk nöromimetik sistemdir. Biçimlerin tanınması problemleri için kullanılmıştır ancak Minsky ve Papert'in 1969'dan itibaren sınırlarını göstermelerinden sonra terkedilmiştir. 1975'den itibaren çağrışımsal hafızalar ve öğrenmeye muktedir yeni şebeke türleri önerilmiştir, ancak bu konudaki gerçek patlama 1982'den sonra, gizli katlı şebekeler (*perceptron*'un genelleştiricisi) Boltzmann'ın stokastik makineleri vs. gibi modellerin ortaya çıkmasıyla gerçekleşmiştir. Bu modeller biçim tanıma, optimizasyon, gerçek zamana uyarlama (örneğin bir süspansiyon motorunun kontrolü) çok alıcılı verilerin kaynaşması problemlerini çözümleme olanağı sağlarlar. Bunlar robotbilimde, süreçlerin kumandasında, görmede, sözün işlenmesi vs.de görülürler.

Günümüzdeki modellerin hiçbirisi yeterli değildir: Bu konuda çok sayıda çalışma yürütülmektedir. Her ne kadar bağlantısal modeller daha önce tanıtılmış olan evrensel çözümü oluşturmasalar da ciddi ilerlemeler umulabilir.

Şimdilik bağlantısal sistemler çoğunlukla bilgisayarlar üzerinde oluşturuluyor. Aynı zamanda çok sayıda paralel makine ticarileştirilmiştir ki bu sis-

temlerin gücünü oldukça arttırmıştır. Gelecek onyıl boyunca milyonlarca Eİ'yi yerleştirme olanağı sağlayan örnekleme ve optik makinelerinin ortaya çıkmasını beklemek gerekiyor.

3. Çok Etkenli Sistemler -

a) Giriş.- Çok etkenli sistemler özerk etkenlerin aynı kaynakları paylaştığı ya da bir çözümün araştırılmasında ortaklaşa çalıştığı özgün bir problem çözümleme yaklaşımı sunarlar. Etkenler şunlar olabilir:

- Dış ortamla doğrudan doğruya karşılıklı çalışan özerk fiziksel kendilikler. Tipik örnek, senkronizasyon, planlama ve yapım kontrolü problemleriyle birlikte çok robotlu bir çevredir (özellikle otomatik hale getirilmiş bir atölyede). Bu konu burada ele alınmayacaktır;
- Tanımlanmış bir mimaride karşılıklı çalışan enformatik kendilikler *(logiciel* modüller, muhtemelen makineler). Verili bir işte her etkenin bir bilgi tabanı ya da uzman bir yöntem olduğu çok uzmanlı sistemlerin paradigması tipiktir.

Bu çalışmalar '70'li yıllar boyunca doğmuştur, bir yandan *blackboard* modeli ile diğer yandan aktör dillerle (Hewitt, 1977), jenerik sınıflar kavramı ile birlikte LOO'larda farklı bir biçime bürünmüştür. Bunlar çok umut vaadeden ve çok-işlemcili makinelere iyi uyarlanan *dağıtılmış* YZ kavramının doğmasına yolaçmıştır. Bağlantısal modellerin taban ilkelerinden birine böyle varıldığını hatırlayalım. Sonunda bu iki araştırma akımı arasında verimli bir ortaklık umut edilebilir.

Dağıtılmış bir yaklaşımın YZ'deki yararı kolayca anlaşılmaktadır çünkü bir yandan bu yaklaşım problem indirgeme kavramına sahiptir (bir görevin karmaşıklığını onu bölerek indirgemek ve tabii ki muhtemelen her biri kendine özgü tanıtma biçimi-

ne sahip olmak kaydıyla bilgi tabanlarını makul ebattaki parçalara bölmek) diğer yandan ise bazı problemlerin dağıtılmış yapısını hesaba katar: aralıkta dağıtım, birçok uzman arasında fonksiyonların uzmanlaşması, *collégial* karar alma.

b) Etkileşim Modelleri.- Çok-etkenli sistemlerin temel bir özelliği etkenlerin *özerkliği*'dir. Buna karşılık çözümlenecek temel problemler etkenler arasındaki etkileşim ve çalışmanın toptan denetimidir. Bu problemler için çeşitli ara çözümler bulunmasına rağmen biz iki yaklaşımı inceleyeceğiz:

- LOO'lardan doğrudan doğruya gelen *mesajlarla iletişim.* Bir etken, kendine özgü bilgileri ve kontrol mekanizması olan bir nesnedir. Her etken, başka bir etkene doğru olası bir "devam etme" enformasyonu ile bir başka etkene, ona belli bir problemi bağlamak için, başvurur.

- Özellikle *blackboard* modelinde, *enformasyonları bölme.* Bu modelde, aslında karşılıklı olarak birbirini tanımayan, bir zatiyetler kümesi ortak bir veri tabanını paylaşarak çalışır. Kendilikler ya da bilgi kaynakları çok değişik yapılarda olabilir: üretim kuralı tabanı ve onun motoru, klasik yöntem vs. *Blackboard*'da belli bir koşul sağlandığı zaman her kendilik asenkron bir şekilde blackboard'a varır. O halde yeni bir varsayımı gösterebilir, onu bir başkası ile dönüştürebilir. Böylelikle yavaş yavaş genel problem çözümüne doğru ilerlenebilir. Bu model çok iyi hazırlanmış çözümleme straejilerinin ortaya konmasına olanak tanır, özellikle de bazı kendiliklerin bir meta-bilgiler kümesine dayanan strateji uzmanları olabilmeleri gerçeğine bağlı olarak.

Blackboard modelinin ilk olarak biçim yorumlama problemleri için öne sürüldüğünü görmüştük (söz, sonra görme ve farklı sinyal türleri). Daha sonra genel problemlerde karar almaya yardım etti-

ği ortaya çıktı. *Blackboard* mimarisine uygulamalar geliştirme olanağı sağlayan araçlar vardır.

BİBLİYOGRAFYA

Bonnet, A.; *L'intelligence artificielle. Promesses et réalités*, Paris, Inter-Editions, 1984.

Bonnet, A.; Haton, J.-P.; Truong-Ngoc, J.M.; *Systèmes Experts. Vers la maitrise technique*, Paris, Inter-Editions, 1986.

Clocksin, W.F.; Mellish, C.S.; *Programmer en PROLOG*, Paris, Eyrolles, 1985.

Farreny, H.; Ghallab, M.; *Eléments d'intelligence artificielle*, Paris, Hermès, 1987.

Giannesini, F.; Kanowi, H.; Pasero, R.; van Caneghem, M.; *PROLOG*, Paris, Inter-Editions, 1985.

Laurière, J.L.; *Intelligence artificielle: résolution de problèmes par l'homme et la machine*, Paris, Eyrolles, 1986.

Nilsson, N.J.; *Principles of Artificial Intelligence*, Berlin, Springer Verlag, 1980.

Rich, E.; *Artificial Intelligence*, New York, McGraw-Hill, 1983.

Quiennec, C.; *Langage d'un autre type: LISP*, Paris, Eyrolles; 1982.

Waterman, D.A.; *A guide to expert systems*, Reading, Addison-Wesley, 1986.