

BİLİMSEL BULUŞLARIN ÖĞRENEN BİLGİSAYAR PROGRAMLARI İLE MODELLENDİRİLMESİ

GLAUBER SİSTEMİ

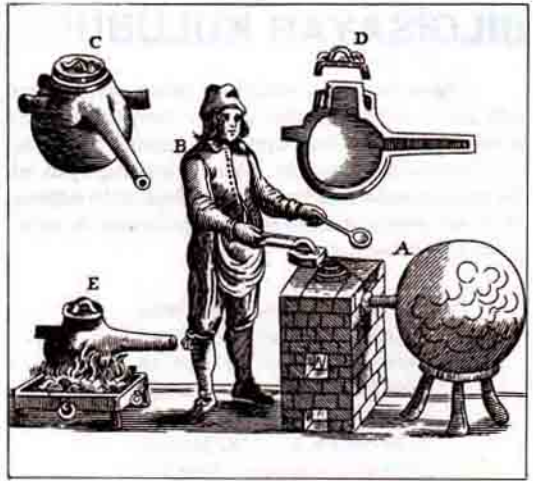
Şakir KOCABAŞ*

(Geçen sayının devamı)

Geçen sayıda da belirttiğimiz gibi GLAUBER sistemi Langley ve arkadaşları (1987) tarafından geliştirilmiş ve verilere dayanan bir sistem. Bu program ismini 17. yüzyılda yaşamış meşhur Alman kimyacı Johann R. Glauber'ın (1604-1670) adından alıyor. Glauber, kimyayı kendi kendine öğrenmiş, sonra Avrupa'yı dolaşıp çeşitli ülkelerde o zaman uygulanan kimyasal metotları öğrenmiş ve sonunda Amsterdam'a yerleşerek orada kendisine bir laboratuvar kurmuştur. Glauber, daha çok metalurji ile asit ve baz reaksiyonları ile ilgilenmiş ve asit-baz teorisini geliştirmiştir. İşte GLAUBER sistemi, Glauber'ın teorisinin bulunuşunu modellendiriyor. Programın uygulaması 17. yüzyılda bilinen kimyasal maddelerin fiziksel özelliklerinden ve basit kimyasal reaksiyonlardan hareketle, nasıl asit-baz teorisine ulaşılabilirliğini gösteriyor. Program önce bu kimyasal maddeleri, onların fiziksel özelliklerinden faydalanarak kavramsal öbeklendirme (conceptual clustering) denilen bir metotla gruplandırıyor. Sonra bu gruplandırmadan elde edilen genel bilgileri o zamanlar bilinen kimyasal reaksiyonlara uygulayarak genelleme yolu ile asit-baz teorisine ulaşıyor. Bunu nasıl yaptığını biraz sonra inceleyeceğiz. Fakat önce programın 17. yüzyıl kimya bilgisini nasıl temsil ettiğini görelim.

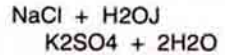
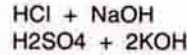
Biz bu programı, Langley ve arkadaşlarının (1987) yazmış oldukları "Scientific Discovery" kitabındaki tasvirinden hareket ederek, Prolog dilinde yazdık (Programın çalıştırılabilir bir özetini aşağıda veriyoruz). Kullandığımız temsil biçimi onların kullandıkları temsil biçiminden biraz farklı. Bu kitabı okuyanlar farkı göreceklerdir. Bu hatırlatmayı yaptıktan sonra şimdi GLAUBER programında bilginin nasıl temsil edildiğini görelim. Kimyasal reaksiyonlar şu şekilde temsil ediliyor:

ampirik (reaksiyon, [tuz-ruhu, kostik-soda], [soda-tuzu])



ampirik (reaksiyon, [zanc-yağı, kostik-potas], [kükürtlü-kostik-potas]),

Bu ifadeleri şu şekilde yorumluyoruz: "Tuzruhu ve kostik soda reaksiyonundan tuz çıkar", "zanc yağı ve kostik potas reaksiyonundan kükürtlü kostik potas çıkar". Tabii biz bunları bu gün kimyada



şeklinde ifade ediyoruz. Tabii 17. yüzyıl kimyasında bu reaksiyondan ayrıca su çıktığı bilinmiyordu. Bu yüzden reaksiyonu yukandaki şekilde anlıyorlardı.

Kimyasal maddeler hakkındaki fiziksel bilgiler ise GLAUBER programında şu şekilde temsil ediliyor:

ampirik (ekşi, tuz-ruhu).
ampirik (keskin-tatlı, kostik-soda).
ampirik (tuza-benzer, potas-tuzu).

Bu ifadeler sırasıyla "tuzruhu ekşi tattadır", "kostik soda keskin tattadır", "potas tuzu tuza benzer", anlamına geliyor. Diğer kimyasal maddeler ve bunların bilinen reaksiyonları hakkındaki bilgileri böylece GLAUBER'e yükledikten sonra programı çalıştırmaya başladığımız zaman, program bu kimyasal maddeleri özelliklerine göre sınıflandırıyor. Bu sınıflandırma tekniğine yukarıda da belirttiğimiz gibi "kavramsal öbeklendirme" deniyor. Buna göre potas tuzu (yani KCl), soda tuzu (NaCl), kükürtlü soda tuzu (Na₂SO₄) tuz tadında oldukları için GLAUBER tarafından bir sınıfta toplanıyor ve bunlara kullanıcı tarafından bir sınıf ismi, meselâ "tuz" ismi veriliyor. Aynı şekilde tuzruhu (HCl), kezzap (HNO₃) ve zanc yağı (H₂SO₄) ekşi tatta olduklarından bir sınıfta, kostik soda (NaOH), kostik potas (KOH) da bir sınıfta toplanıyor ve bunlara da "asit" ve "baz" isimleri verildiğinde GLAUBER bu sınıflandırmayı kendi bilgi tabanına şu şekilde katıyor.

formel (asit, tuz-ruhu).

* Department of Electronic and Electrical Engineering, King's College London (KQC), London WC2R, UK.

formel (baz, kostik-soda).

formel (tuz, soda-tuzu).

GLAUBER daha sonra bu genel isimleri bilinen kimyasal reaksiyonlara şu şekilde uyguluyor ve her kimyasal maddeye cins ismini vererek,

ampirik (reaksiyon [tuz-ruhu, kostik-soda], [soda-tuzu]).

ifadelerinden

hipotez (reaksiyon, [asit, baz], [tuz])

ifadesini, yani "bir baz bir asitle reaksiyona girerse tuz meydana gelir," hipotezini çıkartıyor ve bunu bilgi tabanına ilâve ediyor.

GLAUBER programı, öğrenen sistemler arasında fiziksel bilimlerde veriye dayalı buluşları modelendiren ilk sistemlerden biri. Çıkardığı sonuçlar basit olmasına rağmen kimya biliminin gelişme süreci için önemli bir soyutlamayı temsil ediyor. Ayrıca, bulunan bu hipotez yani "bir baz bir asitle reaksiyona girerse tuz meydana gelir," hipotezi kimyada çok genel bir prensip olup bir çok asit-baz reaksiyonlarını açıklayabilmekte. Bir bilimde bir hipotezin önemi, onun açıklama gücü ile orantılı olarak artıyor. Böyle açıklama gücü yüksek hipotezler ilerde daha başka araştırmalara ve buluşlara yol açabiliyor. Deneyssel metodun geliştirilmesinden sonra, kimya biliminin gelişmesi de işte böyle genel hipotezlerin bulunması sayesinde mümkün olmuştur.

Bilimsel araştırma ve buluşların bilgisayar programları ile modellendirilmesi alanındaki çalışmalar bize, bilim adamlarının buluşları hangi bilgilere dayanarak nasıl yaptıklarını, bu süreç sırasında hangi aşamalardan geçtiklerini, ne gibi metodolojik kurallara uyguladıklarını öğrenmemize yarıyor. Bu çalışmalardan elde ettiğimiz bilgilerle belki ilerde bir "Bilimsel Araştırma Metodolojisi" geliştirmemiz mümkün olabilir. Böyle bir araştırma metodolojisinin gelecekte araştırmacı bilim adamlarına sağlayabileceği faydalar ortada.

Bu çalışmaların bilim felsefesi açısından da önemi çok büyük. Çünkü bu çalışmalar, bize sadece bilimsel teorilerin geliştirilmesinde hangi aşamalardan geçtiğinin anlaşılmasını ve bunlarda uygulanan metodolojik kuralların anlaşılmasını sağlamakla kalmayıp, bilimsel teorilerin hangi tür bilgilere dayandığını ve dolayısıyla bu teorileri daha iyi anlamamızı sağlayacaktır. □

GLAUBER PROGRAMININ LİSTESİ

GLAUBER
© C. Sakir Kocabas (1987)
A Qualitative Discovery System in Chemistry
A Simplified version of GLAUBER (Langley, et al., 1987)
Classify Properties and Objects
This part of the program generates formal knowledge by conceptual clustering

classify_properties:-
empirical(P,V),

```
not(formal(C,P)),  
name_class(P),  
fail.  
classify_properties:-  
classify_objects.  
name_class(P):-  
nl, write('Please give a name to the class of objects'),  
nl, write('which have the property '),  
write(P),  
write(' -> '),  
read(C),  
assertz(formal(C,P)),  
assertz(formal(class,C)),  
write(formal(C,P)),  
write(' is added'),  
write(formal(class,C)),  
write(' is added').
```

```
classify_objects:-  
formal(C,P),  
empirical(P,Obj),  
not(formal(C,Obj)),  
assertz(formal(C,Obj)),  
write(formal(C,Obj)),  
write(' is added'),  
fail.  
classify_objects.
```

3 Find hypotheses

3.1 This part of the program finds hypotheses

```
find_hypotheses:-  
empirical(reactn,[X,Y,I2]),  
formal(C1,Y), formal(C2,Y), formal(C3,Z),  
not(tested(hypothesis(reactn,[C1,C2],[C3])),  
test(hypothesis(reactn,[C1,C2],[C3])),  
fail.  
find_hypothesis.
```

```
test(hypothesis(reactn,[C1,C2],[C3])):-  
forall((and(formal(C1,Y),formal(C2,Y),empirical(reactn,[X,Y],[C3])),formal(C3,Z)),  
assertz(hypothesis(reactn,[C1,C2],[C3])), nl,  
write(hypothesis(reactn,[C1,C2],[C3])),  
write(' is added'),  
asserta(tested(hypothesis(reactn,[C1,C2],[C3])),  
test(hypothesis(reactn,[C1,C2],[C3])).  
write('):- write('), nl.  
write('):- write('), nl, nl.
```

3.1 Logic functions

```
and(X,Y):-  
call(X), call(Y).
```

```
and(X,Y,Z):-  
call(X), call(Y), call(Z).
```

```
forall(X,Y):-  
not(X), !, fail.  
forall(X,Y):-  
not(and(call(X),not(call(Y))),
```

3.2 Chemical Data

```
empirical(reactn,[hcl,naoh],[nacl]),  
empirical(reactn,[hcl,koh],[kcl]),  
empirical(reactn,[h2so4,naoh],[na2so4]).
```

```
empirical(reactn,[zn,hcl],[zncl2]),  
empirical(reactn,[fe,h2so4],[feso4]).
```

```
empirical(salt_like,nacl),  
empirical(salt_like,kcl),  
empirical(salt_like,na2so4),  
empirical(salt_like,feso4),  
empirical(salt_like,zncl2).
```

```
empirical(tastes_bitter,naoh),  
empirical(tastes_bitter,koh).
```

```
empirical(tastes_sour,hcl),  
empirical(tastes_sour,h2so4).
```

```
empirical(has_metallic_colour,zn),  
empirical(has_metallic_colour,fe).
```

FOTOĞRAFIN DÜŞÜNDÜRDÜKLERİ



Geçen sayıda yayınladığımız yandaki resim, bir denizaltı canlısı olan paçavra balığıdır. Yumurta ve çoğalan balığın dış çevreye uyum sağlayana dek, yumurta sarısı içeren bir kesecik taşıdığı ve yenecek varlıkları ile beslendiği bilinmektedir.

Bu sayıda da alttaki fotoğrafı ilginize sunuyoruz



GLAUBER'in Prolog dilinde yazdığımız program listesi. Programı yükledikten sonra çalıştırmak için şu komutu kullanmak gerekiyor :

? classify properties

Bu komut bilgi tabanındaki kimyasal maddeleri özelliklerine göre sınıflandırıyor. Bu sınıflandırma esnasında kullanıcı, sistem tarafından bulunan sınıflara "asit", "baz", "tuz" gibi birer isim veriyor. Daha sonra şu komut veriliyor:

? find hypotheses

Bu komutla sistem hipotezleri buluyor. Bulunan hipotezlerin listesini görmek istediğiniz zaman şu komutu verebilirsiniz:

? listing (hypothesis)

KAYNAKLAR :

- Buchanan, B.G. & Feigenbaum, E.A. (1978). *Dndral and Meta-Dendral: Their Application Dimension*. Artificial Intelligence, 11, 5-24.
- Falkenhainer, B.C. and Michalski, R.S. (1986). *Integrating Quantitative and Qualitative Discovery: The ABACUS System*. Machine Learning 1, 4, 367-422.
- Friedland, P. (1979). *Knowledge-Based Experiment Design in Molecular Genetics*. IJCAI-1979, 285-287.
- Jones, R. (1986). *Generating Predictions to Aid the Scientific Discovery Process*. AAAI-86, 513-517.
- Kocabaş, S. (1988a). *Qualitative and Quantitative Reasoning in Particle Physics*. Technical Report. Department of Electronic and Electrical Engineering. King's College London. University of London.
- Kocabaş, S. (1988b). *Conflict Resolution as Discovery in Particle Physics*. Technical Report. Department of Electronic and Electrical Engineering. King's College London, University of London.

Kulkarni, D. & Simon, H.A. (1988). *The Processes of Scientific Discovery: The Strategy of Experimentation*. Cognitive Science 12, 139-175.

Langley, P., Simon, H.A., Bradshaw, G.L., Zytkow, J.M. (1987). *Scientific Discovery: Computational Explorations of the Creative Processes*. Cambridge, Mass., The MIT Press.

Lenat, D.B. (1979). *On automated scientific theory formation: a case study using the AM program*. In J.Hayes, D.Michie and L.J. Mikulich (Eds.) *Machine Intelligence 9* (Halstead, New York) 251-283.

Lenat, D.B. (1983). *EURISKO: A Program That Learns New Heuristics and Domain Concepts*. Artificial Intelligence 21, 61-98.

Nordhausen, B. & Langley, P. (1987). *Towards an Integrated Discovery System*. IJCAI-87, 198-200.

Rose, D. & Langley, P. (1986). *Chemical Discovery as Belief Revision*. Machine Learning, 1, 423-452.

Zytkow, J.M., & Simon, H.A. (1986). *A Theory of Historical Discovery: The Construction of Componential Models*. Machine Learning, 1, 107-137.

PC/Computing Dergisinde, Carol ELLISON imzası ile yayınlanan "Japonlar Neden Yazılım Gerçekleştiremiyor???" başlıklı yazının, M.Cem ŞAKI tarafından yapılan özet tercümesidir.



"Ve siz şimdi bana ABO'ym denir nükleer güce ihtiyacı olduğunu sorabilirsiniz"