

Programlama Dillerinin Prensipleri

Hafta 14 - Mantıksal Programlama

Dr. Öğr. Üyesi M. Fatih ADAK

İçerik

- Tarihsel gelişim
- Tanım
- Prolog dili ve yapısı
- Prolog bellek yönetimi
- Mantıksal konsept
- Olaylar
- Kurallar
- Önermeler
- Prolog dilinde değişken ve sabit kavramı
- Eşitlik kavramı
- Durum oluşturma
- Kararlılık
- Geriye Zincir kuralı
- Detaylı Prolog örneği

Tarihsel Gelişim

- Mantıksal programlamanın tek temsilcisi Prolog dilidir.
- 1970'li yıllarda Fransa, Aix-Marseille üniversitesinden Alain Colmerauer ve grubu tarafından tanıtılmıştır.
- Prolog Fransızca **P**rogrammation en **L**ogique kelimesinden gelmektedir.
- Mantığın doğrudan doğruya bir bilgisayar dili olarak kullanılmasını sağlar.



Alain Colmerauer (1941 - 2017)

Tanım

- Mantıksal programlamada bilgisayarın belirli bir problemi çözebilmesi için programa problem ve çözüm yoluyla ilgili bilgi verilmesi gereklidir.
- Mantıksal programlamada döngü, alt program, seçmeli yapı gibi kontrol blokları yerine bildirme esaslı (declarative) bir yapı kullanılır.



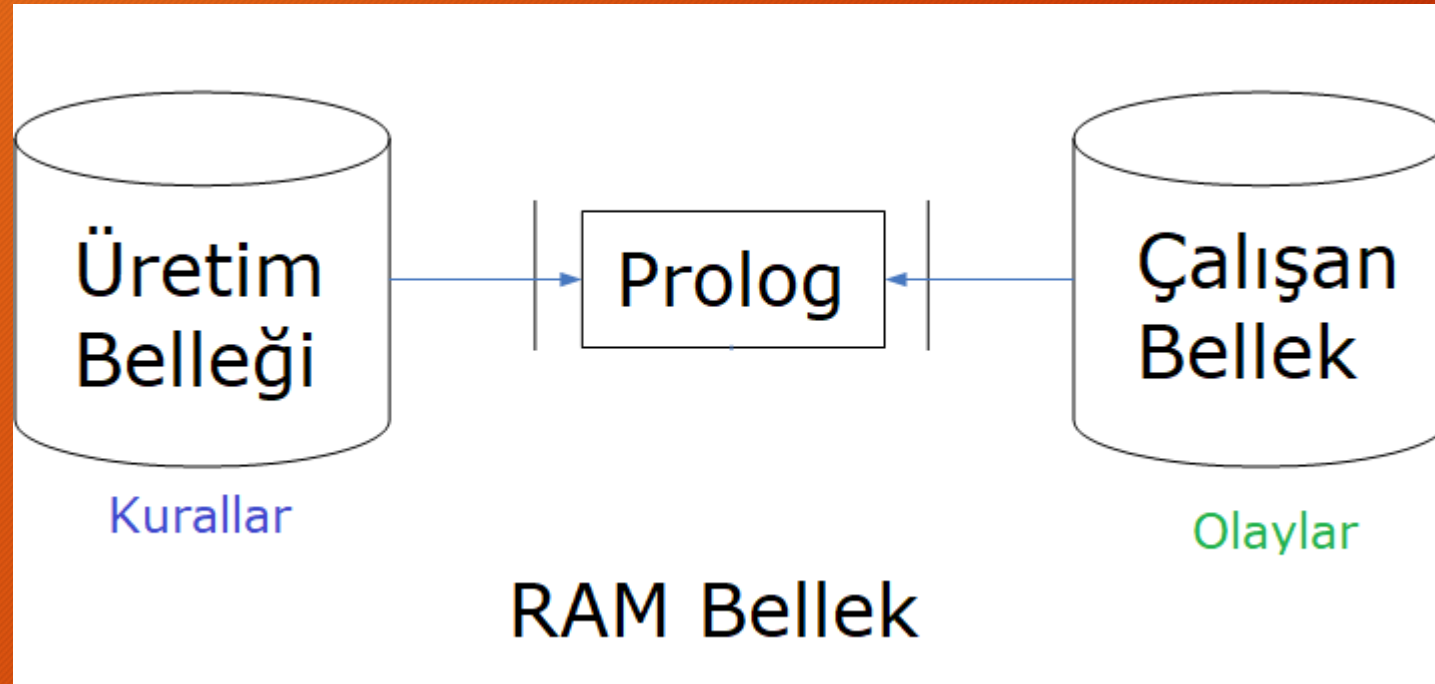
Prolog Dili ve Yapısı

- Mantıksal programlama kural tabanlı sisteme benzer.
- Kurallar basit if kontrolleri ile gerçekleşir.
- Programın gövdesini olayların gerçekleşmesi oluşturur.
- Prolog birçok kusur içermesine rağmen büyük problemleri az programlama bilgisi ile çözmeyi sağlar.

Prolog Bellek Yönetimi

- Kural tabanlı sistemlerde programlama dili, belleği iki bölgede yönetir.
 - Üretim Belleği
 - Çalışan Bellek

Prolog Bellek Yönetimi



Mantıksal Konsept

- Mantıksal programlamada bildirme esaslı yapı kullanıldığı için dili iki temel konsept oluşturur.
 - Olaylar
 - Kurallar



Olaylar

- Doğru olan durumlar olayları oluşturur.
- Mantıksal programlama, bir durumun doğru olduğu ispat etmek için kullanılır.
- Kişi ve Yiyecek değişkenleri üzerindeki kısıtlamalar

Kişi	Yiyecek
Ahmet	Döner
Ayşe	Sarma
Mehmet	Pilav
Kadir	Köfte
Hamza	Makarna
Hüma	Balık

Yemekyer(Ahmet,Döner)

Kurallar

- Basit if yapıları kuralları oluşturur.
- Kurallar ifade edilmek için önermelerden faydalanılır.
 - Atomik Önermeler
 - Bileşik Önermeler

Önermeler

- Önermeler sonuçları doğru olanlar ya da ispat edilmesi gerekenlerdir.
- Yanlış olan durumlar önerme olarak eklenmez.
- Örneğin $\text{Eksi}(X)$ ifadesi
 - X bir sayı ise onun negatif olduğunu söyleyen bir önermedir.
 - X çoğunluğu belirtiyorsa onu eksiltmeyi ifade edecektir.Her iki durumda da önerme doğrudur.

Prolog Dilinde Değişken ve Sabit Kavramı

- Büyük harf veya alt çizgi ile başlayanlar değişkendir.
- Küçük harf ya da tek tırnak içinde olanlar sabitlerdir.

Değişken: Kişi, Pilav vb.

Sabit: yemekyer(Ahmet, Pilav)

Eşitlik Kavramı

- Emir Esaslı ve Nesne Yönelimli dillerde eşitlik değişkenin içerdiği değerin aynı olma durumudur.
- Mantıksal Programlamada eşitlik her iki ifadenin aynı doğruluk tablosuna sahip olmaları şeklinde yorumlanır.

P	Q	$\neg P$	$P \rightarrow Q$	$\neg P \wedge (P \rightarrow Q)$
T	T	F	T	F
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

□

Eşitlik Kavramı

- Mantıksal olmayan dillerde boolean türündeki iki değişken x ve y eşit olması için x doğru ise y doğru olmalı ya da x yanlış ise y yanlış olmalıdır.
- Mantıksal programlamada X doğru ise Y doğru önermesinde, X yanlış ise Y yanlış olmak zorunda değildir.

Durum Oluşturma

- Murat'ın patronu Mehmet'tir.

`patron(Mehmet,Murat)`

- Murat X şirketinde çalışmaktadır.

`çalışır(Murat,X)`



$\forall a$ if *çalışır*(a, X) then *patron*(Mehmet, a)

Kararlılık

patron(Mehmet,Kemal) = patron(Mehmet,Kemal) ifadesi doğrudur. Ama Mantıksal programlama bunu yapmaz. Böyle bir çıkarımda bulunamaz.

- Mantıksal programlamanın karar verebilmesi için **Geriye Zincir Kuralı** olmalıdır.

if X then H

if H then P

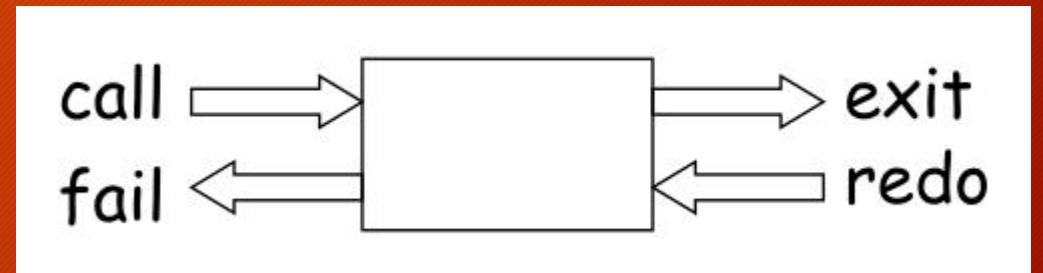
if P then Y



Y doğru ise X doğrudur. Prolog bunu hesaplayabilir.

Geriye Zincir Kuralı

- Her kısıtlamananın 4 portu vardır.
- Başarısız portlar yinele portlarını geri besler.
- Çağrım başarılı oldukça bir sonraki kısıtlamaya ilerlenecektir.
- Başarısızlık ortaya çıkarsa geriye doğru hareket eder.



Detaylı Prolog Örneği

```
kanişköpeği(Karabaş)  
avköpeği(İnceKuyruk)  
teriercinsi(Afacan)
```

```
köpek(X) ⊂ kanişköpeği(X)           //Kanişler bir köpektir  
köpek(X) ⊂ avköpeği (X)             // Av köpekleri bir köpektir  
köpek(X) ⊂ teriercinsi(X)           // Terier bir köpektir  
küçük(X) ⊂ kanişköpeği(X)          // Kaniş köpekleri küçüktür  
küçük(X) ⊂ teriercinsi(X)           // Terier cinsi küçüktür.  
büyük(X) ⊂ avköpeği(X)             // Av köpekleri büyüktür.  
evcil(X) ⊂ köpek(X)                 // Köpekler evcildir  
eviçievcil(X) ⊂ evcil(X) and küçük(X) // Küçük evciller ev içi evcildir.  
evdışievcil(X) ⊂ evcil(X) and büyük(X) // Büyük evciller ev dışı evcildir.
```

Kaynaklar

- Yumusak N., Adak M.F. *Programlama Dillerinin Prensipleri*. 1. Baskı, Seçkin Yayıncılık, 2018
- Sebesta, Robert W. *Concepts of programming languages*. 11 ed. Pearson Education Limited, 2016.
- Sethi, Ravi. *Programming languages: concepts and constructs*. Addison Wesley Longman Publishing Co., Inc., 1996.
- Watt, David A. *Programming language design concepts*. John Wiley & Sons, 2004.
- Malik, D. S., and Robert Burton. *Java programming: guided learning with early objects*. Course Technology Press, 2008.
- Waite, Mitchell, Stephen Prata, and Donald Martin. *C primer plus*. Sams, 1987.
- Hennessey, Wade L. *Common Lisp*. McGraw-Hill, Inc., 1989.
- Liang, Y. Daniel. *Introduction to Java programming: brief version*. pearson prentice hall, 2009.
- Yumusak N., Adak M.F. *C/C++ ile Veri Yapıları ve Çözümlü Uygulamalar*. 2. Baskı, Seçkin Yayıncılık, 2016