

Programlama Paradigmaları

Programlama paradigması; problemlere üretilen çözümlerin nasıl formül haline getirileceğiyle alakalı olan temel programlama stilidir.

Yazılım dünyasındaki genel programlama paradigmaları:

- ❑ Yapısal (Imperative) Programlama (Pascal, C)
- ❑ Nesne-tabanlı (Object-oriented) Programlama (Java, C++)
- ❑ Fonksiyonel (Functional) Programlama (LISP, Haskell)
- ❑ Mantıksal (Logic) Programlama (Prolog, Mercury)

1

MANTIKSAL PROGRAMLAMA (LOGIC PROGRAMMING)

2

Mantıksal Prog. (Temel Prensipler)

- ❑ Bir mantık programı, aksiyom (varsayım) kümeleri ve bir hedef cümlesinden oluşur.
- ❑ Hedef cümlesinin doğruluğu ve aksiyomların yeterli olup olmadığının tespiti için çıkarım kuralları vardır.

Algoritma = Mantık + Kontrol

3

Mantıksal Prog. (Nasıl Çalışır?)

Gerçek yaşamdaki bazı durumları anlatmak için açıklayıcı (declarative) cümleler kullanılır.

(1) Every mother loves her children.

(2) Mary is a mother and Tom is Mary's child..

4

Mantıksal Prog. (Nasıl Çalışır?)

Bazı genel anlamlandırma kuralları uygulandığında yeni çıkarımlar elde edilir.

Örnek olarak;

(1) ve (2) cümleleri bilinirse aşağıdaki yargı çıkartılabilir;

(3) Mary loves Tom.

Daha dikkatli bir gözlemlerle (1) ve (2) cümlelerinin bütün insanları ve bu insanlar arasındaki bağıntıları anlattığı farkedilebilir.

5

Mantıksal Prog. (Nasıl Çalışır?)

Bu örnek mantıksal programlamanın ana fikrini ortaya koyar.

Mantıksal programlamada kısacası nesneler üzerindeki sonsuz bağıntılar kullanılır ve bu bağıntılar ile (3) gibi yargılara ulaşılır.

Bu işlemler için "Predicate Calculus" vb. yöntemler kullanılır.

6

Mantıksal Prog.(Predicate Calculus)

Predicate Calculus "Hüküm hesapları" olarak Türkçe'ye çevrilebilir.

Hüküm hesapları bilinen gerçeklerden yeni gerçekler ortaya çıkarmak için kurallar ve aksiyomlar (varsayım) sağlar.

7

Mantıksal Prog.(Predicate Calculus)

Sabitler

"Mary" ve "Tom" gibi nesneleri belirten semboller...

Predicate Sembolleri

"loves", "mother" ve "child_of" gibi bağıntıları belirten semboller...

8

Mantıksal Prog.(Predicate Calculus)

Konuşma dili...

Mary loves Tom.

Predicate Calculus karşılığı...

loves(mary,tom)

9

Mantıksal Prog.(Predicate Calculus)

If everybody respects somebody then that person is a king.

$\forall X(\forall Y \text{ person}(Y) \Rightarrow \text{respects}(Y,X)) \Rightarrow \text{king}(X)$

10

Mantıksal Prog. (Uygulama Alanları)

- İlişkisel veritabanları
- Doğal dil işleme arayüzleri
- Uzman sistemler
- Sembolik denklem çözümü
- Teorem kanıtlama
- Prototip üretimi
- Simülasyon

11

PROLOG

12

Prolog

Yapay zekada sık kullanılan programlama dillerinden biri de Prolog dilidir. Son versiyonu Visual Prolog olarak piyasaya çıkartılmıştır.

Prolog, yüklem (ifade) mantığını kullanarak, bilgisayara çözümü aranan problem hakkında bilinen gerçekleri ve kuralları vererek, uygun bir çözüm elde edilmesini sağlar.

13

Prolog

- Geleneksel programlama dillerinde, bir programcı herhangi bir problemin nasıl çözüleceğini bilgisayara adım adım tanıtmak zorundadır.
- Oysa bir Prolog programcısının yapması gereken şey, çözüm aranan problem hakkında bilinen gerçekler ve kuralları, bunlar arasındaki ilişkileri tanımlamak, daha sonra mümkün olan bütün çözümleri bulmak görevini Prolog'a vermektir.

14

Prolog (Özellikleri)

- Bir Prolog programı, diğer dillerde olduğu gibi bir dizi komut satırından değil, doğruluğu önceden bilinen gerçeklerden ve bu gerçeklerden bilgi sağlamaya yarayan kurallardan oluşur.

- Prolog, cümlecikler (Horn Clauses) üzerine inşa edilmiştir. Cümlecikler, yüklem mantığı denilen formal sistemin bir alt kümesidir.

- Prolog'da bir Karar Motoru (Inference Engine) vardır. Bu motor, verilen bilgiyi kullanarak cevabı aranan bir problem için, mantıksal bir şekilde karar veren bir işlemdir.

15

Prolog (Özellikleri)

- Karar motorundaki Kalıp Eşleştirici (**Pattern Matcher**) sorulara uygun olan cevapları eşleştirerek önceden bilinen ve program içine kaydedilen bilgiyi geri çağırır.

- Prolog programının temelini, program akışı içinde önceden verilen gerçekler ve kurallar oluşturur.

- Sorulan sorulara mantıklı cevaplar bulmanın yanısıra, bulduğu tek bir çözümle yetinmez, başka alternatifleri de inceleyerek mümkün olan bütün çözümleri bulur.

16

Prolog (Cümleler)

Bir Prolog programcısı nesneleri, bağıntıları (relation) ve bu bağıntıların doğru olduğu kuralları tanımlar.

Bill likes dogs.

Bu cümleye bağıntının ne zaman doğru olduğunu gösteren bir kural eklenirse:

Bill likes dogs if the dogs are nice.

17

Prolog (Gerçekler)

Prolog dilinde yazılmış bazı gerçekler:

likes(bill, cindy).

likes(cindy, bill).

likes(bill, dogs).

Gerçekler nesnelerin özelliklerini belirtmek amacıyla da kullanılırlar:

"Kermit is green."

green(kermit)

"Catlin is a girl"

girl(catlin)

18

Prolog (Kurallar)

Kurallar verilen gerçeklerden yeni gerçekler türetmeye yararlar.

Cindy likes everything that Bill likes.

Catlin likes everything that is green.

Aynı kuralları Prolog dilinde yazmak gerekirse:

likes(cindy, Something):-likes(bill, Something).

likes(catlin, Something):-green(Something).

19

Prolog (Sorgular)

Prolog'da bir dizi gerçek verildiğinde; bu gerçeklerle ilgili sorular sorulabilir ve bu işleme Prolog sistemini sorgulamak denir.

Does Bill likes Cindy?

likes(bill, cindy).

Bu sorguya karşılık Prolog "yes" cevabını döndürür.

20

Prolog (Sorgular)

Dikkat edilirse Prolog dilinde gerçekler ve sorgular arasında yazım farkı yoktur.

Nesneler ("bill") küçük harfle başlar. Bu "bill"ın sabit bir nesne olduğunu gösterir.

"What" ise büyük harfle başlar ve değişken bir nesnedir.

21

Prolog (Sorgular)

Prolog bir sorguya cevap verebilmek için daima en tepeden başlayarak gerçekleri tarar. Prolog tarama işlemine taranacak gerçek kalmayana kadar devam eder.

What does Bill likes?
likes(bill,What).

What=cindy
What=dogs **2 solutions**

22

Prolog (Örnek Program)

PREDICATES

nondeterm likes(symbol, symbol)

CLAUSES

likes(ellen, tennis).
likes(john, football).
likes(tom, baseball).
likes(eric, swimming).
likes(mark, tennis).
likes(bill, Activity):-likes(tom, Activity).

GOAL

likes(bill,baseball).

23

Prolog (Değişkenler)

Prolog'da değişkenler kullanılarak genel gerçekler ve kurallar yazılabildiği gibi genel sorularda sorulabilmektedir.

Bill likes the same thing as Kim.
likes(bill,Thing):-likes(kim,Thing).

Prolog'da değişkenler büyük harfle veya "_" karakteri ile başlar.

24

Prolog (Değişkenlerin Değer Alması)

- Diğer programlama dillerinde, değişkenlerin değer almalarına imkan tanıyan atama ifadeleri Prolog'da yoktur. Prolog'u diğer programlama dillerinden ayıran en önemli özelliklerden biri de budur.
- Prolog'daki değişkenler almaları gereken değerleri atamayla değil, gerçekler veya kurallardaki sabitlere eşleştirilirken alırlar.

25

Prolog (Örnek)

PREDICATES

nondeterm sever(symbol, symbol)

CLAUSES

sever(oktay, okuma).

sever(yavuz, bilgisayar).

sever(orhan, tavla).

sever(vedat, uyuma).

sever(ismail, yuzme).

sever(ismail, okuma).

Hem okuma hem de yüzmeden kimin hoşlandığı sorusuna cevap aramak için şu sorgu kullanılır;

GOAL

sever(Sahis, okuma), sever(Sahis, yuzme).

26

Prolog (Hedefler)

Sorgulamaları hedef olarak tanımlamak daha anlamlıdır, çünkü Prolog'a bir sorgu yöneltmekle, ona yerine getirilmesi gereken bir hedef verilmiş olur.

Hedefler sever(ismail, yuzme) gibi basit olabileceği gibi sever(Sahis, okuma), sever(Sahis, yuzme) gibi daha karmaşık da olabilir. Birden fazla parçadan oluşan hedefe **Birleşik Hedef**, her bir parçaya da **alt hedef** denir.

27

Prolog (Bağlaçlar ve Ayraçlar)

Birleşik bir hedefin çözümünü bulmak için her iki alt hedefin doğru olması gerekir. Bu durumda, iki alt hedef arasında (ve) anlamına gelen ';' kullanılır. Fakat istenilen durumlarda alt hedeflerden sadece birinin doğru olması şartı da aranabilir. Bu durumda alt hedefler arasında (veya) anlamına gelen ',' kullanılması gerekir. Bu duruma **Ayırma işlemi** denilmektedir.

28

Prolog (Örnek)

PREDICATES

nondeterm araba(symbol,long,integer,symbol,long)
nondeterm kamyon(symbol,long,integer,symbol,long)
nondeterm arac(symbol,long,integer,symbol,long)

CLAUSES

araba(chrysler,130000,3,kirmizi,12000).
araba(ford,90000,4,gri,25000).
araba(datsun,8000,1,kirmizi,30000).
kamyon(ford,80000,6,mavi,8000).
kamyon(datsun,50000,5,sari,20000).
kamyon(toyota,25000,2,siyah,25000).
arac(Marka,Kilometresi,Yas,Renk,Fiyat):-
araba(Marka,Kilometresi,Yas,Renk,Fiyat);
kamyon(Marka,Kilometresi,Yas,Renk,Fiyat).

GOAL arac(Marka, Kilometresi, Kullanim_Suresi, Renk, 25000).

29

Prolog (Örnek devamı)

Bu örnekte fiyatı 25.000 dolardan az olan bir araba **veya** fiyatı 20.000 dolardan daha az olan bir kamyon var mıdır?’ şeklindeki bir soruya uygun bir hedef yazmak istersek:

arac(Marka,Kilometresi,Yas,Renk,Fiyat):-

araba(Marka, Kilometresi, Kullanim_Suresi, Renk, Fiyat), Fiyat< 25000; kamyon (Marka, Kilometresi, Kullanim_Suresi, Renk, Fiyat), Fiyat< 20000.

30

Sonuçlar

- ✓ Mantıksal programlama kendisini birçok alanda kanıtlamış bir yöntemdir. Bu kullanım alanları arasında uzman sistemler, doğal dil işleme mantıksal programlamanın kullanıldığı en zor alanlardır.
- ✓ Mantıksal programlama günümüzde Web için geliştirilen uygulamalara bilgi tabanlı tekniklerin uygulanmasını sağlamaktır.

31

Sonuçlar

- ✓ Prolog programlama dilinin uygulama alanlarını genişletici bir etken Prolog’un JAVA implementasyonlarının geliştirilmesi olmuştur.
- ✓ Prolog günümüzde en çok yapay zeka uygulamalarında kullanılmakta, bunda en çok yüklem mantığı tekniğinin güçlü etkisi gözlenmektedir.

32