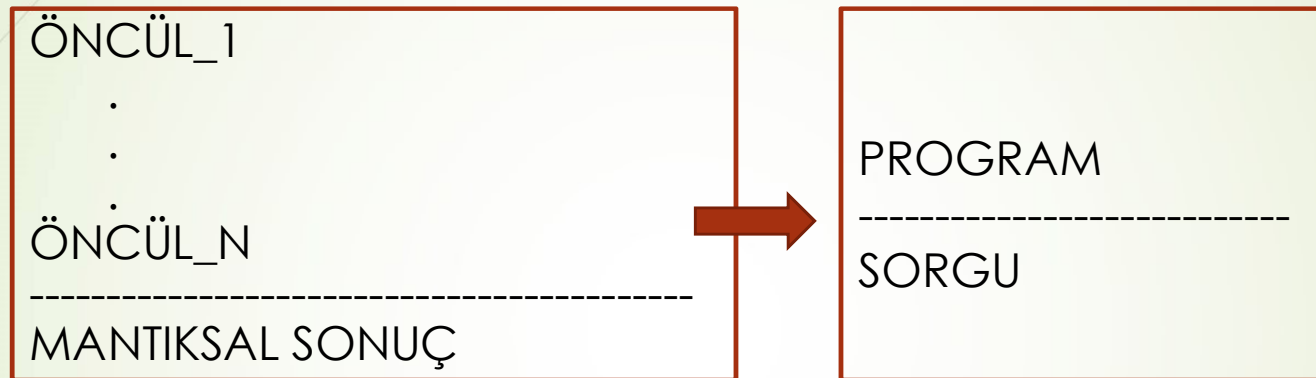




Prolog ile Mantık Programlamaya Giriş

Mantıktan Programlamaya ... (Genel)





Prolog Nedir?

- Alain Colmerauer ve Robert Kowalski tarafından yetmişli yıllarda geliştirilmiştir.
- Prolog = Programmation en Logique (Programming in Logic)
- Prolog, diğer birçok programlama dilinden farklı olarak deklaratif bir programlama dilidir.
- Geleneksel yapıdaki prosedürel programlama dillerinde programcı adım adım problemin çözüm yolunu betimlemelidir.
- Tamamen deklaratif olan programlama dillerinde, programcı problemin ne olduğunu belirtir ve gerisini programlamalama dilinin arka planındaki sisteme bırakır.
- Deklaratif bir programlama dili olan Prolog'da;
 - Programcı erişilecek bir hedefi ortaya koyar / betimler
 - Prolog sistemi bu hedefe nasıl ulaşacağını bulur.



Kullanım Alanları

- Doğal Dil İşleme
- Uzman Sistemler
- Otomatik Akıl Yürütme
- Problem Çözme
- ...

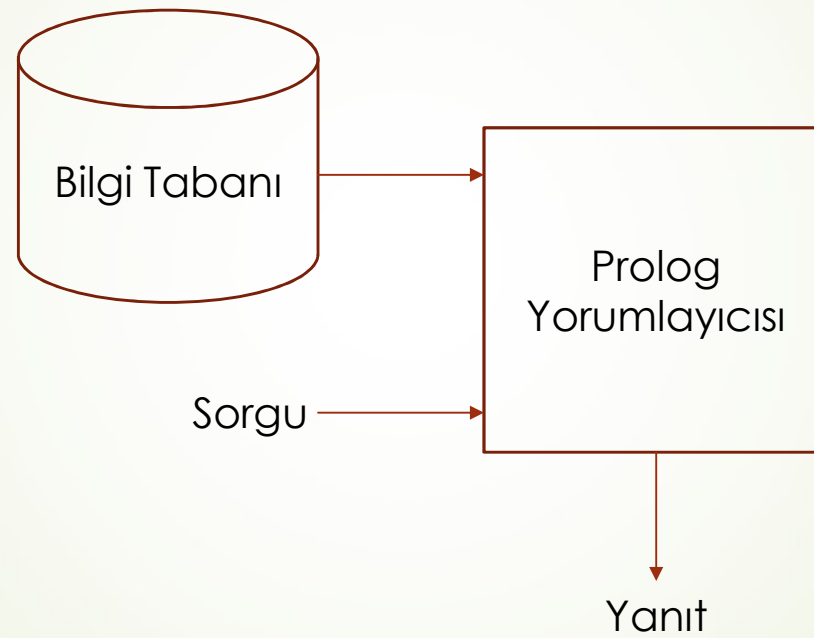
SWI-Prolog

- Açık kaynaklı ve ücretsiz bir Prolog yorumlayıcısıdır (interpreter).
- Linux, Windows ve Mac OS'ta çalışabilme imkanı sağlar.
- Tamamen ISO-Prolog uyumludur.
- Çevrimiçi sürümü
 - <http://swish.swi-prolog.org/>
- Masaüstü sürümü
 - <http://www.swi-prolog.org/download/stable>
- IDE seçenekleri
 - <http://www.swi-prolog.org/IDE.html>



SWI Prolog

Bilgi Tabanı ve Prolog Yorumlayıcısı



Bilgi Tabanı - v1

kedi(sylvester).
kedi(tom).
kedi(garfield).
sarman(garfield).
miyavla.

?- kedi(tom).
yes

?- sarman(garfield).
yes

?- sarman(tom).
no

?- tembel(garfield).
ERROR: predicate tembel/1 not defined.



Bilgi Tabanı - v1

kedi(sylvester).
kedi(tom).
kedi(garfield).
sarman(garfield).
miyavla.

?-miyavla.
yes

?-tirmala.
no



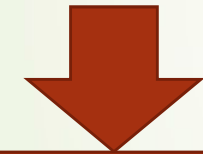
Bilgi Tabanı – v2

kedi(garfield).
lazanya_yer(garfield).

Gerçek (fact)

mutlu(garfield) :- lazanya_yer(garfield).
uyur(garfield) :- mutlu(garfield).

Kural (rule)



Baş (head)



Gövde (body)

?- kedi(garfield).
yes

?- mutlu(garfield).
yes



Tümcecik (Clause)

kedi(garfield).
lazanya_yer(garfield).

mutlu(garfield) :- lazanya_yer(garfield).
uyur(garfield) :- mutlu(garfield).

- Bu bilgi tabanında toplam 4 tümcecik vardır.
- Bunlardan ikisi gerçek, ikisi ise kural şeklinde kodlanmıştır.
- Bir cümleciğin sonuna nokta (.) konulur.



Yüklem (Predicate)

kedı(garfield).
lazanya_yer(garfield).

mutlu(garfield) :- lazanya_yer(garfield).
uyur(garfield) :- mutlu(garfield).

- Bu bilgi tabanında toplam 4 yüklem vardır. Bunlar:

kedı, lazanya_yer, mutlu, uyur



Bilgi Tabanı – v3

kedi(garfield).
tok(garfield).
iyi_uyumuş(garfield).

mutlu(garfield) :- tok(garfield), iyi_uyumuş(garfield).

- **Virgül işareti Prolog’da birleşimi (conjunction) ifade eder.**

Bilgi Tabanı – v3

kedi(garfield).
tok(garfield).
iyi_uyumuş(garfield).

mutlu(garfield) :- tok(garfield).
mutlu(garfield) :- iyi_uyumuş(garfield).
% mutlu(garfield) :- tok(garfield); iyi_uyumuş(garfield).

- **Noktalı virgül işareti Prolog'da ayrışımı (disjunction) ifade eder.**
- **Noktalı virgül ile tek bir ifade altında toplanan tümcecikler ayrık olarak da ifade edilebilir.**
- **% işareti Prolog'da yorum satırı oluşturmada kullanılır.**

Mantık ve Prolog

	Mantık	Prolog
Gerektirme (Implication)	$B \rightarrow A$	$A :- B$
Birleşme (Conjunction)	$A \wedge B$	A , B
Ayrışma (Disjunction)	$A \vee B$	$A ; B$



Bilgi Tabanı – v4

Değişken (Variable)

kedı(tom).
kedı(sylvester).
fare(jerry).
kuş(tweety).

kovalar(tom, jerry).
kovalar(sylvester, tweety).

?- kedi(X).

X=tom;

X=sylvester;

no.



Bilgi Tabanı – v4

Değişken (Variable)

kedı(tom).
kedı(sylvester).
fare(jerry).
kuş(tweety).

kovalar(tom, jerry).
kovalar(sylvester, tweety).

?- kovalar(tom, X), fare(X).

X=jerry

yes



Bilgi Tabanı – v5

Değişken (Variable)

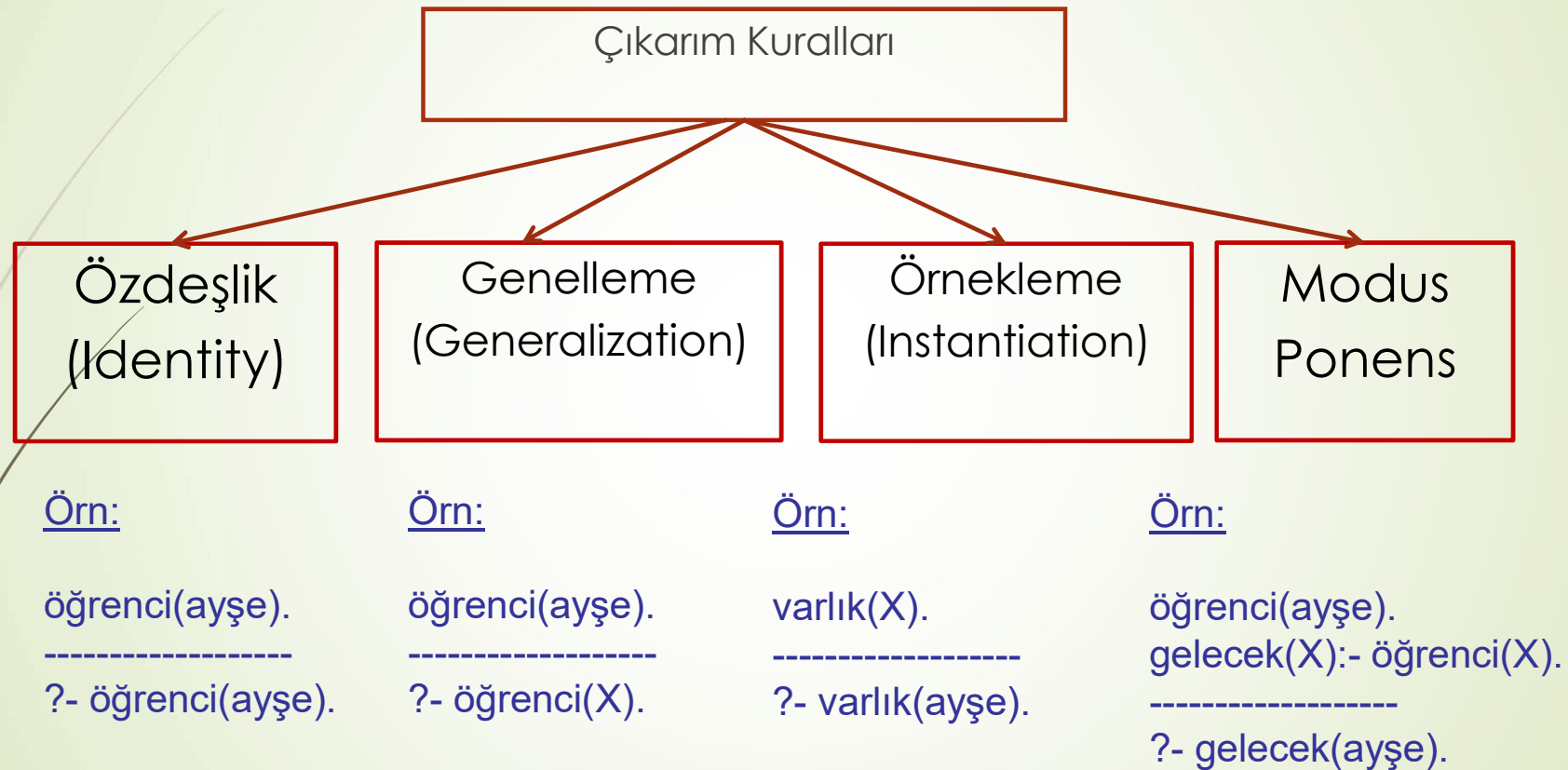
kedi(tom).
kedi(sylvester).
fare(jerry).
kuş(tweety).

kovalar(X,Y) :- kedi(X), fare(Y).
kovalar(X,Y) :- kedi(X), kuş(Y).

?- kovalar(tom, X).

X = jerry ;
X = tweety ;
no

Prolog'un Çıkarım Kuralları



Prolog'un Temel Yapıları

Temel Yapılar

Terimler

Önermeler

Sabitler

Değişkenler

Gerçekler

Kurallar

Bileşik
Terimler

Sorgular

Örn:
ayşe
öğrenci
15
33.8

Örn: X
—

Örn:
öğrenci(ayşe).

Örn:
gelecek(X):-
öğrenci(X).

Örn:
?- gelecek(ayşe).

Atomlar
Sayılar

Örn:
öğrenci(ayşe).

Atomlar ve Sayılar

- Küçük harfle başlayıp büyük harfler, küçük harfler, rakamlar veya alt çizgi gibi karakterlerle devam eden bir karakter dizisi atomdur.

garfield, sarman_kedi, müzikÇalar

- Tek tırnaklar arasına yazılan herhangi bir karakter dizisi atomdur.

'Ahmet', 'Merhaba dünya', '@\$%'

- Tamsayılar: 12, -34, 22342
- Ondalıklı sayılar: 34573.3234, 0.3435



Değişkenler

- Büyük harf veya alt çizgi karakteri ile başlayıp büyük harfler, küçük harfler, rakamlar veya alt çizgi gibi karakterlerle devam eden bir karakter dizisi değişkendir.

X, Y, Değişken, Ahmet, _X

Bileşik Terimler

- Atomlar, sayılar ve değişkenler bileşik terimleri oluşturmada kullanılan bileşenlerdir.
- Bileşik terimlerin oluşumunda bir yüklem bir dizi argümanla birleştirilir.
 - Argümanlar parantezler içine, birbirlerinden virgülle ayrılacak şekilde yazılır.
 - Yüklem mutlaka bir atom olmak zorundadır.
- Örneğin;
 - – kedi(garfield)
 - – kovalar(tom, jerry)
 - – kovalar(sylvester, Y)

Argüman Sayısı (Arity)

- Bileşik bir terimin argüman sayısına arity adı verilir.
 - **kedi(garfield)**, 1 argümanlıdır
 - **kovalar(tom,jerry)**, 2 argümanlıdır
- Bir yüklem farklı argüman sayısına sahip şekilde aynı bilgi tabanında bulunabilir. Prolog bu farklı argüman sayısına sahip yüklemeleri farklı şekilde değerlendirir.
- Yüklem adından sonra gelen «/» işareti argüman sayısını belirtir.
 - kedi/1
 - kovalar/2