

Prolog

IVAN BRATKO, *PROLOG PROGRAMMING
FOR ARTIFICIAL INTELLIGENCE*.
PEARSON EDUCATION, 2001.

Logičko programiranja

Stil programiranja koji se bitno razlikuje od imperativnog programiranja

Programming in **Log**ic – Prolog je najrasporstranjeniji jezik za ovu namenu

Prva implementacija – Univerzitet u Marseju početkom 1970tih

Dalji razvoj na Univerzitetu u Edinburgu je doveo do standardizovane verzije - Edinburški prolog

Prološki programi su deklarativni

Postoje samo dve komponente programa

- Činjenice i
- Pravila

Prološki sistem učitava program, a korisnik postavlja serije upita

Matematička logika

Iskazni račun

- Iskazna slova $p, q, r, \dots$
- Operatori $\neg, \wedge, \vee, \Rightarrow$
- Istinitosne vrednosti formula $\top, \perp$

Predikatski račun

- Skup funkcijskih simbola – $f, g, h, f_1, f_2, \dots$
- Skup relacijskih simbola – $P, Q, R, P_1, P_2, \dots$
- Operatori $\neg, \wedge, \vee, \Rightarrow, \exists, \forall$
- Koriste se i
 - Simboli promenljivih (zamena) – $X, Y, Z, X_1, \dots$
 - Simboli konstanti – $a, b, c, \dots$ – funkcijski simboli arnosti 0

Primer

```
dog(fido).  
dog(rover).  
dog(henry).  
cat(felix).  
cat(michael).  
cat(jane).  
animal(X):-dog(X).
```

SWI Prolog

```
4 ?- [pr0501].  
true.  
  
5 ?- dog(fido).  
true.  
  
6 ?- dog(jane).  
false.  
  
7 ?- animal(fido).  
true.  
  
8 ?- dog(X).  
X = fido ;  
X = rover ;  
X = henry.  
  
9 ?- animal(felix).  
false.
```

Prolog okruženje

?- -- prompt
?- [ime]. -- učitavanje programa
?-halt. -- izlazak
?-make. -- ponovno učitavanje

```
11 ?- write('Hello World'),nl,write('Welcome to Prolog'),nl.  
Hello World  
Welcome to Prolog  
true.
```

Podaci u Prologu - **termi**

Brojevi

- Celi i razlomljeni, sa opcionim znakom + ili – ispred
 - 623
 - -47
 - +5
 - 025
 - 6.43
 - -.245
 - +256.
 - 0.56

Podaci u Prologu - **termi**

Atomi

- Konstante koje nemaju numeričku vrednost
- Niz slova, cifara i donje crte koji počinje malim slovom
 - john
 - today_is_Tuesday
 - fred_jones
 - a32_BCD
- Ali ne
 - Today
 - today-is-Tuesday
 - 32abc

Podaci u Prologu - termi

Atomi

- Konstante koje nemaju numeričku vrednost
- Niz karaktera ograničenih jednostrukim navodima
 - 'Today is Tuesday'
 - 'today-is-Tuesday'
 - '32abc'
- Niz specijalnih karaktera + - * / > < = & # @
 - +++
 - >=
 - >
 - +-

Podaci u Prologu - **termi**

Promenljive

- Niz slova, cifara i donje crte koji počinje velikim slovom

- X
- Author
- Person_A
- _123A

Ali ne

- 45_ABC
- Person-A
- authorToday is Tuesday'
- 'today-is-Tuesday'
- '32abc'

Podaci u Prologu - termi

Složeni termi

- Strukturirani podaci – predikati (ili funktori).
- Jedan ili više argumenata razdvojenih zarezom
- Opšti oblik $functor(t_1, t_2, \dots, t_n)$ $n \geq 1$, gde su t_i termi
 - likes(paul,prolog)
 - read(X)
 - dog(henry)
 - cat(X)
 - >(3,2)
 - person('john smith',32,doctor,london)
- likes(dog(henry),Y)
- pred3(alpha,beta,gamma,Q)
- pred(A,B,likes(X,Y),-4,pred2(3,pred3(alpha,beta,gamma,Q)))

Podaci u Prologu - termi

Liste

- Specijalan tip strukturiranih podataka
- Neograničen broj argumenata (termova) razdvojenih zarezima i oivičeni srednjim zagradama []
 - `[dog,cat,y,mypred(A,b,c),[p,q,R],z]`
 - `[[john,28],[mary,56,teacher],robert,parent(victoria,albert),[a,b,[c,d,e],f],29]`
 - `[[portsmouth,edinburgh,london,dover],[portsmouth,london,edinburgh],[glasgow]]`
 - `[]`

Reprezentacija zasnovana na logici

Kao i u predikatskom računu, predikati su osnovna reprezentaciona struktura u prologu

Prolog može da se posmatra kao dokazivač teorema koji prihvata upite korisnika i ispituje da li su tačni li ne

Srpski	Predikatski računa	Prolog
i	$\wedge$	,
ili	$\vee$	;
samo ako	$\Leftarrow$	:-
ne	$\neg$	not

Reprezentacija zasnovana na logici

Engleski	Predikatski računa	Prolog
Everyone likes Susie	$\forall x \text{ Likes}(x, \text{Susie})$	<code>likes(X, susie). likes(Everyone, susie).</code>
George likes Kate and George likes Susie.	$\text{Likes}(\text{George}, \text{Kate})$ $\wedge \text{Likes}(\text{George}, \text{Susie})$	<code>likes(george, kate), likes(george, susie).</code>
George likes Susie if George does not like Kate	$\text{Likes}(\text{George}, \text{Susie})$ $\Leftarrow \neg \text{Likes}(\text{George}, \text{Kate})$	<code>likes(george, susie) :- not(likes(george, kate)).</code>

Hornove formule (kaluze)

Disjunkcija sa najviše jednim pozitivnim (nenegiranim) literalom

$$\neg p \vee \neg q \vee \cdots \vee \neg t \vee u \quad \longrightarrow \quad p \wedge q \wedge \cdots \wedge t \Rightarrow u \quad \longrightarrow \quad u : -p, q, \dots, t.$$

Specijalno

$$\neg p \vee \neg q \vee \cdots \vee \neg t \quad \longrightarrow \quad p \wedge q \wedge \cdots \wedge t \Rightarrow \perp \quad \longrightarrow \quad : -p, q, \dots, t.$$

Pravilo rezolucije je osnovni mehanizam zaključivanja

$$\frac{A \vee p, B \vee \neg p}{A \vee B}$$

$$\frac{A \vee p, B \vee \neg p}{A \vee B}$$

Zaključivanje

a.
b.
c: $\neg a, b$.
d: $\neg b, c$.
?-d.

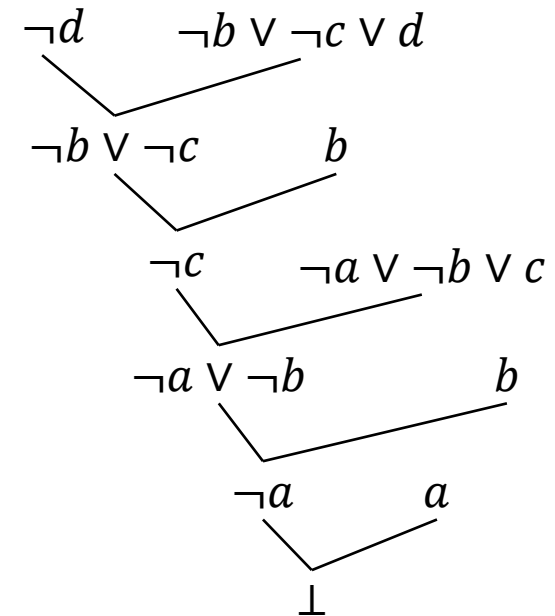
$\{a, b, a \wedge b \Rightarrow c, b \wedge c \Rightarrow d\} \vdash d$

tj.

$\{a, b, \neg a \vee \neg b \vee c, \neg b \vee \neg c \vee d\} \vdash d$

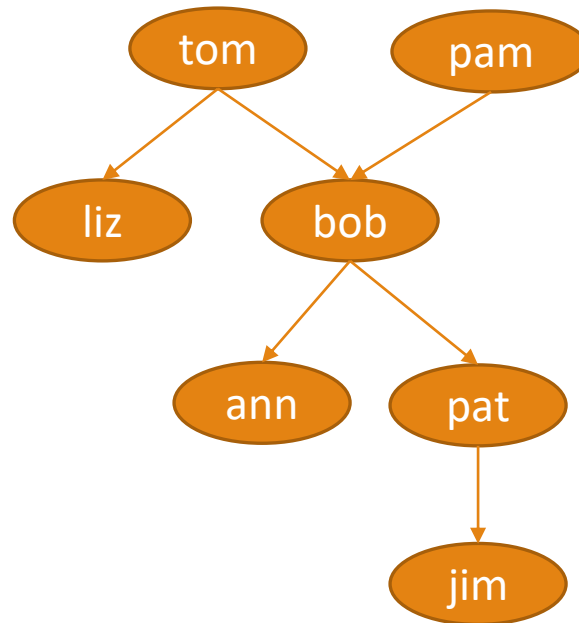
tj.

$\{a, b, \neg a \vee \neg b \vee c, \neg b \vee \neg c \vee d, \neg d\} \vdash \perp$



Definisanje činjenica

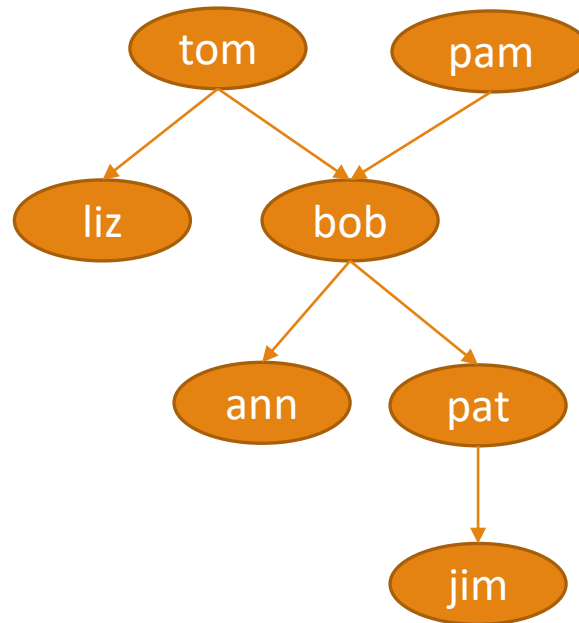
```
parent(tom,bob).  
parent(pam,bob).  
parent(tom,liz).  
parent(bob,ann).  
parent(bob,pat).  
parent(pat,jim).
```



```
14 ?- parent(bob,pat).  
true.  
  
14 ?- parent(liz,pat).  
false.  
  
15 ?- parent(X,liz).  
X = tom.  
  
16 ?- parent(bob,X).  
X = ann ;  
X = pat.  
  
17 ?- parent(X,Y).  
X = tom,  
Y = bob ;  
X = pam,  
Y = bob ;  
X = tom
```


Definisanje činjenica

```
parent(tom,bob).  
parent(pam,bob).  
parent(tom,liz).  
parent(bob,ann).  
parent(bob,pat).  
parent(pat,jim).
```



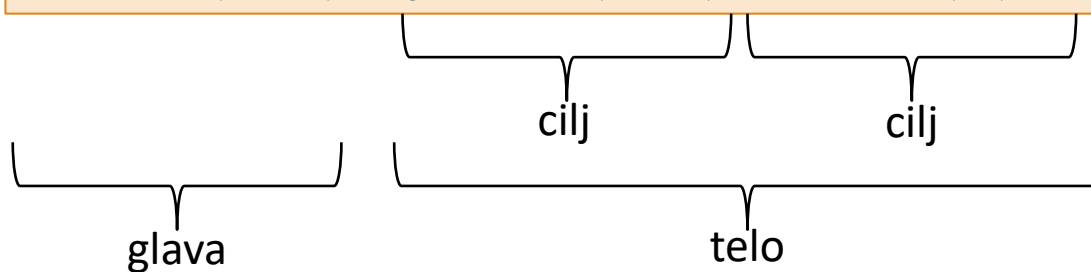
```
18 ?- parent(Y,jim),parent(X,Y).  
Y = pat,  
X = bob.  
  
19 ?- parent(tom,X),parent(X,Y).  
X = bob,  
Y = ann ;  
X = bob,  
Y = pat ;  
false.  
  
20 ?- parent(X,ann),parent(X,pat).  
X = bob.
```

Definisanje pravila

```
parent(tom,bob).  
parent(pam,bob).  
parent(tom,liz).  
parent(bob,ann).  
parent(bob,pat).  
parent(pat,jim).
```

```
female(pam).  
male(tom).  
male(bob).  
female(liz).  
female(pat).  
female(ann).  
male(jim).
```

```
mother(X,Y):-parent(X,Y),female(X).
```



```
27 ?- mother(pam,bob).  
true.
```

X = pam i Y = bob

```
mother(pam,bob):-parent(pam,bob),  
                  female(pam).
```

```
28 ?- mother(X,Y).  
X = pam,  
Y = bob ;  
X = pat,  
Y = jim.
```

Definisanje pravila

```
parent(tom,bob).  
parent(pam,bob).  
parent(tom,liz).  
parent(bob,ann).  
parent(bob,pat).  
parent(pat,jim).
```

```
female(pam).  
male(tom).  
male(bob).  
female(liz).  
female(pat).  
female(ann).  
male(jim).
```

```
mother(X,Y):-parent(X,Y),female(X).
```

```
grandparent(X,Z):-parent(X,Y),parent(Y,Z).
```

```
sister(X,Y):-parent(Z,X),parent(Z,Y),female(X).
```

```
sister(X,Y):-parent(Z,X),parent(Z,Y),female(X), X\=Y.
```

```
33 ?- sister(X,pat).  
X = ann ;  
X = pat ;  
false.
```

```
35 ?- sister(X,pat).  
X = ann ;  
false.
```

Zaključivanje u Prologu

Robot

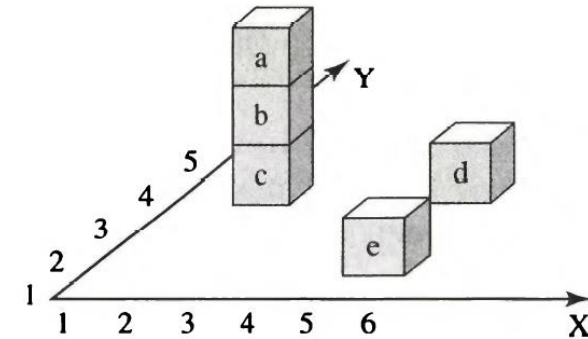
Robot u skladištu ima kameru kojom može da detektuje objekte

- Objekti su oblika kocke jedinične ivice
- Robot detektuje oznake na objektima i njihove pozicije
- Koordinate u skladištu se celi brojevi

Za objekte na slici robot može da detektuje objekte a, d i e i vodi evidenciju na čemu koji objekat stoji

```
see(a, 2, 5).  
see(d, 5, 5).  
see(e, 5, 2).
```

```
on(a, b).  
on(b, c).  
on(c, table).  
on(d, table).  
on(e, table).
```



Robot

```
see(a,2,5).  
see(d,5,5).  
see(e,5,2).
```

```
on(a,b).  
on(b,c).  
on(c,table).  
on(d,table).  
on(e,table).
```

```
2 ?- on(Block,_).  
Block = a ;  
Block = b ;  
Block = c ;  
Block = d ;  
Block = e.
```

```
3 ?- see(B1,_,Y),see(B2,_,Y).  
B1 = B2, B2 = a,  
Y = 5 ;  
B1 = a,  
Y = 5,  
B2 = d ;  
B1 = d,  
Y = 5.
```

```
4 ?- see(B1,_,Y),see(B2,_,Y),B1\=B2.  
B1 = a,  
Y = 5,  
B2 = d ;  
B1 = d,  
Y = 5,  
B2 = a ;  
false.
```

```
5 ?- on(B,_),on(_,B).  
B = b ;  
B = c ;  
false.
```

Robot

```
see(a,2,5).  
see(d,5,5).  
see(e,5,2).
```

```
on(a,b).  
on(b,c).  
on(c,table).  
on(d,table).  
on(e,table).
```

```
z(B,0):-on(B,table).  
z(B,Z):-on(B,B0),z(B0,Z0),Z is Z0+1.
```

```
zz(B,0):-on(B,table).  
zz(B,Z0+1):-on(B,B0),zz(B0,Z0).
```

```
16 ?- z(a,Z).  
Z = 2 .  
  
17 ?- z(a,Z).  
Z = 2 ;  
false.  
  
17 ?- zz(a,Z).  
Z = 0+1+1 .  
  
18 ?- zz(a,ZFormula),Z is ZFormula.  
ZFormula = 0+1+1,  
Z = 2 .
```

Unifikacija

Dva terma se mogu unifikovati

- Ako su identični
- Ako se varijablama koje se pojavljuju u termima mogu se pridružiti objekti tako da zamenom vrednosti tih varijabli, temi postanu identični

`date(D,M,2021)` i `date(D1,may,Y1)` se mogu unifikovati

- `D -> D1`
- `M -> may`
- `Y1 -> 2021`

```
19 ?- date(D,M,2021)=date(D1,may,Y1).  
D = D1,  
M = may,  
Y1 = 2021.
```

Ovi termi će postati identični i za sledeće vrednosti promenljivih

- | | |
|--------------------------|-------------------------|
| ◦ <code>D = 1</code> | <code>D = third</code> |
| ◦ <code>D1 = 1</code> | <code>D1 = third</code> |
| ◦ <code>M = may</code> | <code>M = may</code> |
| ◦ <code>Y1 = 2021</code> | <code>Y1 = 2021</code> |

```
20 ?- date(D,M,2021)=date(D1,may,Y1),  
      date(D,M,2021)=date(15,M,Y).  
D = D1, D1 = 15,  
M = may,  
Y1 = Y, Y = 2021.
```


Algoritam unifikacije

Ako su S i T konstante, tada se S i T mogu unifikovati samo ako su isti objekat

Ako je S promenljiva i T ima ma koju vrednost, tada se mogu unifikovati ako se S dodeli vrednost T.

Ako su S i T structure, one se mogu unifikovati samo ako

- S i T imaju isti predikat
- I svi njihovi argumenti se mogu unifikovati

`triangle(point(1,1), A, point(2,3)) = triangle(X, point(4,Y), point(2,Z))`

- `triangle = triangle`
- `X = point(1,1)`
- `A = point(4,Y)`
- `point(2,3) = point(2,Z)`
 - `Z = 3`

Deklarativano vs. proceduralno značenje Prološkog programa

$P:-Q,R.$ - P, Q, R imaju sintaksu termova

Deklarativno tumačenja klauze:

- P je tačno ako su Q i R tačni
- Iz Q i R sledi P

Proceduralno tumačenja klauze:

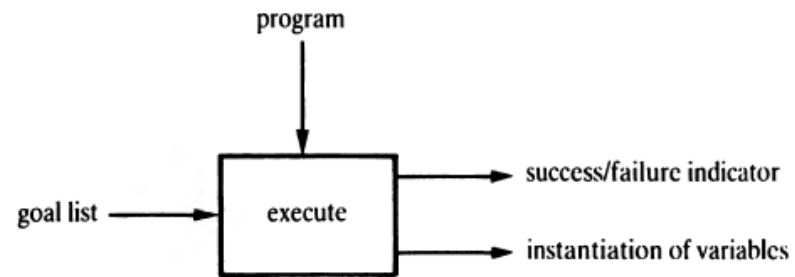
- Da bi se rešio problem P , najpre mora da se reši podproblem Q , a potom podproblem R .

U proceduralnom tumačenju ne određuju se samo ciljevi, već i redosled realizacije cilja.

Proceduralno značenje

Izlazni podaci:

- Indikator uspeha – success/failure
- Generisane vrednosti u slučaju uspešnog izvršavanja



Koraci izvršavanja

```
big(bear).  
big(elephant).  
small(cat).  
brown(bear).  
black(cat).  
gray(elephant).  
dark(Z):-black(Z).  
dark(Z):-brown(Z).
```

```
22 ?- dark(X),big(X).  
X = bear.
```

Inicijalna lista ciljeva: dark(X), big(X)

Scan (top to bottom) – dark(X)

- dark(Z):-black(Z).
- Nova lista ciljeva black(X),big(X)

Scan – black(X)

- black(cat) -> X = cat
- Nova lista ciljeva big(cat) -> fail
- Lista ciljeva black(cat),big(cat) -> fail
- Lista ciljeva black(X) -> fail
- Lista ciljeva dark(X), big(X)

Scan – dark(X)

- dark(Z):-brown(Z).
- Nova lista ciljeva brown(X),big(X)

Scan – brown(X)

- brown(bear) -> X = bear
- Nova lista ciljeva: big(bear)