

SEZNAMI, OPERATORJI IN ARITMETIKA V PROLOGU

Ivan Bratko

PROLOG

- Prolog = “čisti Prolog” + dodatki
- Čisti Prolog ~ logika
- Z dodatki naredimo logično zasnovo prologa uporabno v praksi (npr. branje in izpis)
- Dodatki:
 - “Čisti” (ne vplivajo na logični pomen programa, popravijo le zunanji izgled)
 - “Umazani” (nimajo logičnega pomena, npr. `write(X)`)
 - Nekateri dodatki: seznamski zapis, operatorski zapis, aritmetika, vhod-izhod

SEZNAMSKI ZAPIS

- Primeri seznamov:

[a, b, c, d]

[]

[ann, tennis, tom, running]

[link(a,b), link(a,c), link(b,d)]

[a, [b,c], d, [], [a,a,a], f(X,Y)]

GLAVA IN REP (HEAD AND TAIL)

- $L = [a, b, c, d]$
 - “a” je *glava* od L
 - $[b, c, d]$ je *rep* od L
- Zapis s pokončno črto:
 - $L = [\text{Head} \mid \text{Tail}]$
 - $L = [a, b, c] = [a \mid [b, c]] = [a, b \mid [c]] = [a, b, c \mid []]$

SEZNAMSKI ZAPIS JE LE KOZMETIKA

- Seznamski zapis: [Head | Tail]
- Ekvivalentno standardnemu prologovemu zapisu: .(Head, Tail)
- Pazi: “.” je funktor
- Ekvivalentna zapisa:

$$[a, b, c] = .(a, .(b, .(c, [])))$$

Drugi izraz lahko ponazorimo z drevesom, koren je prva pika

ELEMENT SEZNAMA

% member(X, L): X is member of L

member(X, [X | _]). % X appears as head of list

member(X, [_ | L]) :-

 member(X, L). % X in tail of list

RAZNI NAČINI UPORABE member/2

- ?- member(c, [a,b,c,d]). % Ali je a element danega seznama
yes
- ?- member(X, [a,b,c,d]). % Poisci katerikoli element seznama
X = a;
X = b;
...
- ?- member(a, L). % Poisci seznam L, ki vsebuje "a"
L = [a | _] ; % "a" je na zacetku L
L = [_, a | _] ; % "a" je na drugem mestu v L
L = [_, _, a | _] ; % "a" je na tretjem mestu v L
...

STIK SEZNAMOV (CONCATENATION OF LISTS)

```
% conc( L1, L2, L3): L3 is concatenation of L1 and L2
```

```
conc( [ ], L, L).                % Base case
```

```
conc( [X | L1], L2, [X | L3]) :- % Recursive case  
    conc( L1, L2, L3).
```


PRIMERI UPORABE CONC

?- conc([a,b,c], [1,2,3], L).

L = [a,b,c,1,2,3]

?- conc([a,[b,c],d], [a,[],b], L).

L = [a, [b,c], d, a, [], b]

?- conc(L1, L2, [a,b,c]).

L1 = [], L2 = [a,b,c];

L1 = [a], L2 = [b,c];

L1 = [a,b], L2 = [c];

L1 = [a,b,c], L2 = []

KAJ PA TOLE?

?- conc(L, _, _).

....

Kateri meseci so pred majem, kateri za?

?- Meseci = [jan,feb,mar,apr,may,jun,jul,aug,sep,oct,nov,dec] ,

conc(Before, [may | After], Meseci).

Before = [jan, feb,mar,apr],

After = [jun,jul,aug,sep,oct,nov,dec]

Briši glede na vzorec, vse od treh 'z' naprej

?- L1 = [a,b,z,z,c,z,z,z,d,e],
conc(L2, [z,z,z | _], L1).

% Dani seznam

% L2 je L1 do treh z

ELEMENT SEZNAMA S CONC

% member2(X, L): X is member of list L

member2(X, L) :-

conc(_, [X | _], L).

BRISANJE IZ SEZNAMA

% del(X, L, NewL)

del(X, [X | Tail], Tail).

del(X, [Y | Tail], [Y | Tail1]) :-
del(X, Tail, Tail1).

?- del(X, [a, b, c, d], L1).

...

VSTAVLJANJE V SEZNAM

% insert(X, L, NewL)

Vstavljanje je obratna operacija od brisanja.

Brisanje deluje v obe smeri: briši ali vstavi

Zato posebnega programa za insert/3 niti ne potrebujemo

VSTAVLJANJE

```
?- del( apple, L, [1,2,3] ).           % Kaj je L?
```

```
...
```

```
% insert( X, L, LongerL): Vstavi X v L na kateremkoli mestu
```

```
insert( X, List, LongerList) :-
```

```
    del( X, LongerList, List).
```

```
% member3( X, L): X je element L
```

```
member3( X, L) :-
```

```
    del( X, L, _).           % L je mogoce izbrisati iz L
```


PODSEZNAM

% sublist(List, Sublist): Sublist se pojavi kot podseznam v List

% Nariši in prepisi v prolog

sublist(S, L) :-

conc(L1, L2, L),

conc(S, L3, L2).

DEFINICIJA SEZNAMA

```
list( [ ]).
```

```
list( [ _ | Tail] ) :-  
    list( Tail).
```

```
% Generiraj sezname narascajocih dolzin
```

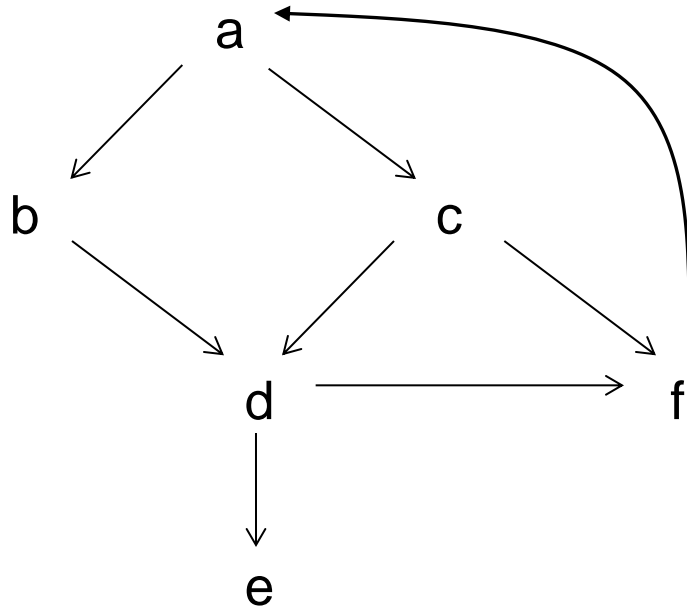
```
?- list( L).
```

```
L = [ ];
```

```
L = [ _A];
```

```
...
```

POT V GRAFU



link(a, b).

link(b, d).

link(c, f).

link(d, f).

link(a, c).

link(c, d).

link(d, e).

link(f, a).

OBSTAJA POT V GRAFU

% path(StartNode, GoalNode): path exists between the nodes

path(Node, Node).

path(Start, End) :-

link(Start, Next),

path(Next, End).

POT V GRAFU

% path(Start, Goal, Path):

% Path = list of nodes from Start to Goal

path(Start, Start, [Start]).

path(Start, Goal, [Start | Rest]) :-

link(Start, Next),

path(Next, Goal, Rest).

- Poskusi:

?- path(a, e, P).

?- path(a, c, P). % Problem: iskanje v globino zgreši rešitev

- Kako dosežemo iskanje z iterativnim poglobljanjem? To je približek iskanja v širino

?- conc(P, _, _), path(a, c, P). *% conc generira nastavke naraščajočih dolžin*

OPERATORSKI ZAPIS

OPERATORSKI ZAPIS

- Zunanja, kozmetična izboljšava
- Ekvivalentna zapisa:

$$+(*(2,a), *(b,c)) = 2*a + b*c$$

- +, * sta infiksna operatorja, vgrajena v prolog
- Kovencija v prologu: + ima višjo prioriteto kot *

UPORABNIK LAHKO UVEDE SVOJE OPERATORJE

has(peter, information).

supports(floor, table).

Lahko pišemo z operatorji:

:- op(600, xfx, has).

:- op(600, xfx, supports).

peter has information.

floor supports table.

TIPI OPERATORJEV

(1) infiksni operatorji

xfx xfy yfx

(2) prefiksna operatorja

fx fy

(3) postfiksna operatorja

xf yf

- **yfx** levo asociativni operator
- **xfy** desno asociativni
- Razlika med **fx** in **fy**?

DEKLACIJA OPERATORJA, “DIRECTIVE”

- **op(Prioriteta, Tip, Operator).**
- Deklariraj operatorje, da bosta naslednja stavka legalen zapis v prologu:

eva ima kolo in zelen avto in lep bel klobuk.

?- X = ..., Y = ..., ...

if X > Y then Z = X else Z = Y.

- Napiši interpreter v prologu za if-then-else stavek.

INTERPRETER ZA IF-THEN-ELSE

% Primerna deklaracija operatorjev

:- op(500, fx, if).

:- op(400, xfx, then).

:- op(300, xfx, else).

% Interpreter

if POGOJ then S1 else S2 :-	% To velja, če
POGOJ, S1	% velja: POGOJ in S1
;	% ali
\+ POGOJ, S2.	% velja: ne POGOJ in S2

ARITMETIKA

VGRAJENI PREDIKATI ZA ARITMETIČNE OPERACIJE

?- $X = 1 + 2$.

$X = 1 + 2$

?- X is $1 + 2$.

$X = 3$

% "is": vgrajeni predikat, izsili računanje

ARITMETIČNE OPERACIJE

- $+$, $-$, $*$, $/$, $**$ na realnih številih
- $//$, mod na celih številih
- \sin , \cos , \log , ... standardne funkcije

?- X is $2 + \sin(3.14/2)$.

$X = 2.9999996829318345$

?- A is $11/3$.

$A = 3.6666666666666665$

?- B is $11//3$.

$B = 3$

?- C is $11 \text{ mod } 3$.

$C = 2$

PRIMERJALNI VGRAJENI PREDIKATI

X > **Y**

X < **Y**

X >= **Y**

X <= **Y**

X ::= **Y**

X != **Y**

štev. vrednosti X in Y aritm. enaki

številski vrednosti neenaki

?- $315 * 3 \geq 250 * 4$.

no

?- $2+5 = 5+2$.

no

?- $2+5 ::= 5+2$.

yes

LENGTH OF LIST

$\text{length}([], 0)$.

$\text{length}([_ | L], N)$:-

$\text{length}(L, N0)$,

N is $N0 + 1$.

Ali lahko zamenjamo vrstni red ciljev v drugem stavku?

Aritmetika - „čisti“ ali „umazani“ dodatek?