

Програмирање и програмски језици

4-СМ

Јелица Васиљевић

Динамички предикати

1. Пролошки програм (или база знања) се састоји од једног или више **предиката**.
2. Сваки предикат се састоји од једне или више **клауза**
 1. Клауза може бити безусловно тачна – **чињеница**
 2. Клаузом се може представити начин извођења закључака – **правило**

```
<program>      ::= <predikat> | <program><predikat>
<predicate>    ::= <klauza> | <predicate><klauza>
<clause>       ::= <cinjenica> | <pravilo>
```

Динамички предикати

1. Могуће је мењати програм у току извршавања
 - a. Додавање нових предиката
 - 1) На почетак програма – пре свих осталих предиката
 - 2) На крај програма – након свих постојећих предиката
 - 3) У чему је разлика ?
 - b. Уклањање постојећих предиката
2. Могуће је додавати/брисати само оне предикате који су на почетку дефинисани као **динамички**
 - a. Предикат се декларише као динамички на следећи начин

:-dynamic imePredikata/brojArgumenata.

Динамички предикати

Нека је дата следећа база знања

```
:-dynamic zivotinja/1.  
zivotinja(lesi).  
zivotinja(cuki).  
zivotinja(maza).
```

Постављењем питања : „Која животиња постоји у бази“ добијамо следеће одговоре

```
?- zivotinja(X).  
X = lesi ;  
X = cuki ;  
X = maza.
```

Брисање предиката zivotinja(maza) :

```
?- retract(zivotinja(maza)).  
true.  
?- zivotinja(X).  
X = lesi ;  
X = cuki.
```

Динамички предикати

Додавање новог предиката `zivotinja(lunja)` на **почетак** програма:

```
?- asserta(zivotinja(lunja)) .  
true.  
?- zivotinja(X) .  
X = lunja ;  
X = lesi ;  
X = cuki.
```

Додавање новог предиката `zivotinja(bubi)` на **крај** програма:

```
?- assertz(zivotinja(bubi)) .  
true.  
?- zivotinja(X) .  
X = lunja ;  
X = lesi ;  
X = cuki ;  
X = bubi.
```

Динамички предикати

Брисање свих података о животињама из базе:

```
?- retractall(zivotinja(_)).  
true.  
?- zivotinja(X).  
false.
```

Листање свих података о животињама из базе:

```
?- listing(zivotinja).  
:- dynamic zivotinja/1.  
  
zivotinja(lesi).  
zivotinja(cuki).  
zivotinja(maza).  
  
true.
```

Динамички предикати

1. Написати програм којим се следећа база знања допуњава свим релацијама „већи од“

```
veci (slon, konj) .  
veci (konj, macka) .  
veci (macka, mis) .
```

Треба да се добије



```
veci (slon, konj) .  
veci (konj, macka) .  
veci (macka, mis) .  
veci (slon, macka) .  
veci (konj, mis) .  
veci (slon, mis) .
```

Динамички предикати

1. Решење :

```
:-dynamic veci/2.
```

```
veci(slon,konj).
```

```
veci(konj,macka).
```

```
veci(macka,mis).
```

```
je_veci(X,Y):-veci(X,Y).
```

```
je_veci(X,Y):-veci(X,Z),je_veci(Z,Y),not(veci(X,Y)),assertz(veci(X,Y)).
```

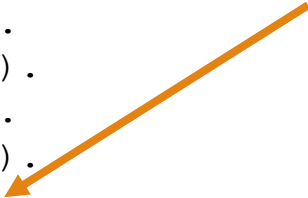
Динамички предикати

1. Резултат :

```
?- je_veci(X,Y).  
X = slon,  
Y = konj ;  
X = konj,  
Y = macka ;  
X = macka,  
Y = mis ;  
X = slon,  
Y = macka ;  
X = slon,  
Y = mis ;  
false.
```

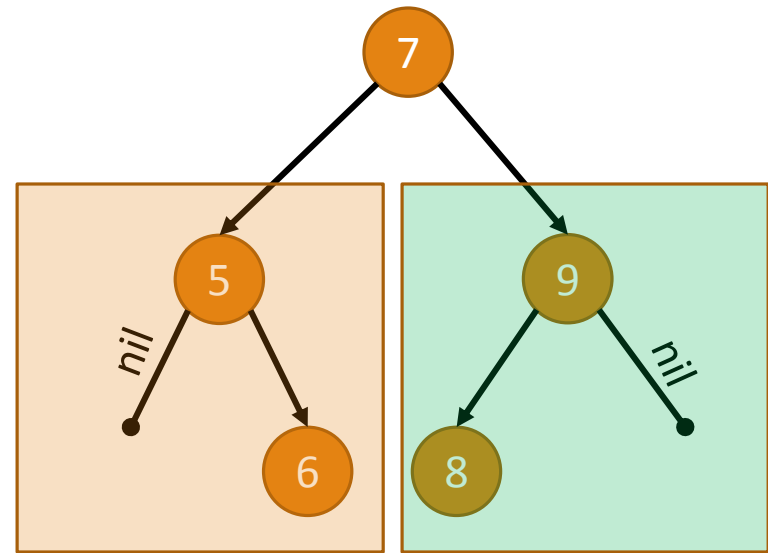
```
?- listing(veci).  
:- dynamic veci/1.  
  
:- dynamic veci/2.  
  
veci(slon, konj).  
veci(konj, macka).  
veci(macka, mis).  
veci(slon, macka).  
veci(konj, mis).  
veci(slon, mis).  
  
true.
```

Зашто се јавља
баш на овом
месту ?



Динамички предикати – Бинарна стабла

```
stablo(5,7,9).  
stablo(nil,5,6).  
stablo(nil,6,nil).  
stablo(8,9,nil).  
stablo(nil,8,nil).
```



Код претраживачких (уређених) бинарних стабала важи: Мањи од корена Већи од корена

Динамички предикати — Бинарна стабла

2. Формирати бинарно претраживачко стабло од листе бројева.

Динамички предикати — Бинарна стабла

2. Формирати бинарно претраживачко стабло од листе бројева.

```
:-dynamic stablo/3.
```

```
napravi([A|Rep]):-asserta(stablo(nil,A,nil)),ubaci(Rep).
```

```
ubaci([X|Y]):-nadjimesto(X,_),ubaci(Y).
```

```
ubaci([]).
```

Зашто је баш овако дефинисано ?

Шта ће се десити ако се овај предикат склони?

```
nadjimesto(Elem,Koren):- stablo(Levo,Koren,_),Elem<Koren,not(Levo=nil),nadjimesto(Elem,Levo).
nadjimesto(Elem,Koren):-
stablo(nil,Koren,Z),Elem<Koren,retract(stablo(nil,Koren,Z)),assertz(stablo(Elem,Koren,Z))
,assertz(stablo(nil,Elem,nil)).
```

```
nadjimesto(Elem,Koren):- stablo(_,Koren,Desno),Elem>=Koren,not(Desno=nil),nadjimesto(Elem,Desno).
nadjimesto(Elem,Koren):-
stablo(Z,Koren,nil),Elem>=Koren,retract(stablo(Z,Koren,nil)),assertz(stablo(Z,Koren,Elem)),
assertz(stablo(nil,Elem,nil)).
```

```
nadjimesto(Elem,_):- assertz(stablo(nil,Elem,nil)).
```

Динамички предикати — Бинарна стабла

2. Решење :

```
?- ubaci([10,5,15,3,7,11,18]).  
true .
```

```
?- listing(stablo).  
:- dynamic stablo/0.
```

```
:- dynamic stablo/3.
```

```
stablo(5, 10, 15).  
stablo(nil, 3, nil).  
stablo(3, 5, 7).  
stablo(nil, 7, nil).  
stablo(nil, 11, nil).  
stablo(11, 15, 18).  
stablo(nil, 18, nil).
```

```
true.
```

ИЛИ

**Којим предикатима је
омогућено исправно
извршавање програма
на ова два начина ?**

```
?-napravi([10,5,15,3,7,11,18]).  
true .
```

```
?- listing(stablo).  
:- dynamic stablo/0.
```

```
:- dynamic stablo/3.
```

```
stablo(5, 10, 15).  
stablo(nil, 3, nil).  
stablo(3, 5, 7).  
stablo(nil, 7, nil).  
stablo(nil, 11, nil).  
stablo(11, 15, 18).  
stablo(nil, 18, nil).
```

```
true.
```