Curs 4

2024-2025

Programare Logică şi Funcţională

Cuprins

Limbajul IMP

Limbajul IMP

Vom defini un limbaj care conţine:

- Expresii
  - Aritmetice — `x + 3`
  - Booleene — `x >= 7`
- Instructiuni
  - De atribuire — `x = 5`
  - Conditionale — `if(x >= 7, x =5, x = 0)`
  - De ciclare — `while(x >= 7,x = x - 1)`
- Compunerea instrutiunilor — `x=7;while(x>=0,x=x-1)`
- Blocuri de instrucţiuni — `{x=7;while(x>=0,x=x-1)}`

Limbajul IMP

Exemplu

Un program în limbajul IMP

```
{x = 10 ; sum = 0;
while(0 =< x,
      {sum = sum + x; x = x-1}
      )},sum
```

- Semantica
  după executia programului, se evaluează `sum`

Sintaxa BNF a limbajului IMP

$$
\begin{aligned}
E &::= n \mid x \\
&\quad \mid E + E \mid E - E \mid E * E \\
\\
B &::= \texttt{true} \mid \texttt{false} \\
&\quad \mid E =< E \mid E >= E \mid E == E \\
&\quad \mid \texttt{not}(B) \mid \texttt{and}(B\ ,\ B) \mid \texttt{or}(B\ ,\ B) \\
\\
C &::= \texttt{skip} \\
&\quad \mid x = E \\
&\quad \mid \texttt{if}(\ B\ ,\ C\ ,\ C\ ) \\
&\quad \mid \texttt{while}(\ B\ ,\ C\ ) \\
&\quad \mid \{\ C\ \} \mid C\ ;\ C \\
\\
P &::= \{\ C\ \},\ E
\end{aligned}
$$

Definirea limbajului IMP în Prolog

Decizii de implementare

- definim operatorii {} si ;

```
:- op(100, xf, {}).
:- op(1100, yfx, ;).  % asociativ la stanga
```

- definim un predicat pentru fiecare categorie sintactică

```
stmt(while(BE,St)) :- bexp(BE),  stmt(St).
```

- while, if, and, etc sunt functori în Prolog
  `while(true,skip)` este un termen compus
- , are semnificatia obisnuită
- pentru valori numerice folosim întregii din Prolog

```
aexp(I) :- integer(I).
```

- pentru identificatori folosim atomii din Prolog

```
aexp(X) :- atom(X).
```

Expresiile aritmetice

$$E ::= n \mid x \mid E + E \mid E - E \mid E * E$$

Prolog

```
aexp(I) :- integer(I).
aexp(X) :- atom(X).
aexp(A1 + A2) :- aexp(A1), aexp(A2).
```

Expresiile aritmetice

Exemplu

```
?- aexp(1000).
true.
?- aexp(id).
true.
?- aexp(id + 1000).
true.
?- aexp(2 + 1000).
true.
?- aexp(x * y).
true.
?- aexp(- x).
false.
```

Expresiile booleene

$$B ::= \texttt{true} \mid \texttt{false} \mid E \texttt{ =< } E \mid E \texttt{ >= } E \mid E \texttt{ == } E \mid \texttt{not}(B) \mid \texttt{and}(B \texttt{ , } B) \mid \texttt{or}(B \texttt{ , } B)$$

Prolog

```
bexp(true). bexp(false).
bexp(and(BE1,BE2)) :- bexp(BE1), bexp(BE2).

bexp(A1 =< A2) :- aexp(A1), aexp(A2).
```

Expresiile booleene

Exemplu

```
?- bexp(true).
true.
?- bexp(id).
false.
?- bexp(not(1 =< 2)).
true.
?- bexp(or(1 =< 2,true)).
true.
?- bexp(or(a =< b,true)).
true.
?- bexp(not(a)).
false.
?- bexp(!(a)).
false.
```

Instructiunile

C ::= skip
| x = E
| if(B , C , C)
| while(B , C)
| { C } | C ; C

Prolog

```
stmt(skip).
stmt(X = AE) :- atom(X), aexp(AE).
stmt(St1;St2) :- stmt(St1), stmt(St2).
stmt(if(BE,St1,St2)) :-  bexp(BE), stmt(St1), stmt(St2).
```

Instructiunile

Exemplu

```
?- stmt(id = 5).
true.
?- stmt(id = a).
true.
?- stmt(3 = 6).
false.
?- stmt(if(true, x=2;y=3, x=1;y=0)).
true.
?- stmt(while(x =< 0,skip)).
true.
?- stmt(while(x =< 0,)).
false.
?- stmt(while(x =< 0,skip)).
true .
```

Programele

$P ::= \{\ C\ \},\ E$

Prolog

```
program(St,AE) :- stmt(St), aexp(AE).
```

Exemplu

```
test0 :- program( {x = 10 ; sum = 0;
                   while(0 =< x,
                         {sum = sum + x; x = x-1}
                        )}
                   , sum).
```

?- test0.
true.

Sintaxa şi semantica

Ce defineşte un limbaj de programare?

Sintaxa şi semantica

Ce defineşte un limbaj de programare?

- Sintaxa – Simboluri de operaţie, cuvinte cheie, descriere (formală) a programelor/expresiilor bine formate

Sintaxa şi semantica

Ce defineşte un limbaj de programare?

- Sintaxa – Simboluri de operaţie, cuvinte cheie, descriere (formală) a programelor/expresiilor bine formate
- Practic – Un limbaj e definit de modul cum poate fi folosit
  - Manual de utilizare şi exemple de bune practici
  - Implementare (compilator/interpretor)
  - Instrumente ajutătoare (analizor de sintaxă, depanator)

Sintaxa şi semantica

Ce defineşte un limbaj de programare?

- Sintaxa – Simboluri de operaţie, cuvinte cheie, descriere (formală) a programelor/expresiilor bine formate
- Practic – Un limbaj e definit de modul cum poate fi folosit
  - Manual de utilizare şi exemple de bune practici
  - Implementare (compilator/interpretor)
  - Instrumente ajutătoare (analizor de sintaxă, depanator)
- Semantica – Ce înseamnă/care e comportamentul unei instrucţiuni?

Sintaxa şi semantica

Ce defineşte un limbaj de programare?

- Sintaxa – Simboluri de operaţie, cuvinte cheie, descriere (formală) a programelor/expresiilor bine formate
- Practic – Un limbaj e definit de modul cum poate fi folosit
  - Manual de utilizare şi exemple de bune practici
  - Implementare (compilator/interpretor)
  - Instrumente ajutătoare (analizor de sintaxă, depanator)
- Semantica – Ce înseamnă/care e comportamentul unei instrucţiuni?

Vom defini semantica formală a limbajului IMP în Prolog!

Pe săptămâna viitoare!