4. ARITMETICA ÎN PROLOG

Deşi PROLOG a fost proiectat în primul rând ca un limbaj de programare simbolică, el dispune de operatorii pentru cele patru operații aritmetice de bază şi operatorii unari minus şi plus, la care se adaugă operatorii mod şi div . Se folosesc simbolurile următoare:

- Operatori binari: + adunare
  - scădere
  * înmulțire
  / împărțire
  mod restul împărțirii întregi
  div împărțirea prin trunchiere
- Operatori unari: - (semnul) minus
  + (semnul) plus

Expresiile obținute cu ajutorul acestor operatori se evaluează pe baza unui set de reguli, care precizează precedența şi asociativitatea operatorilor precum şi conversiile aplicate operanzilor:

- ***Precedența*** (prioritatea) determină ordinea de efectuare a operațiilor într-o expresie cu diverşi operatori.
- ***Asociativitatea*** indică ordinea de efectuare a operațiilor într-o secvență de operații care au aceeaşi precedență.
- ***Regulile de conversie de tip*** asigură stabilirea unui tip comun pentru ambii operanzi, la fiecare operație care solicită acest lucru şi în care tipurile diferă.

Lista din tabelul următor este organizată pe categorii cu aceeaşi precedență, în ordine descrescătoare. Pentru fiecare categorie se indică asociativitatea şi tipul operanzilor şi tipul rezultatului. Trebuie remarcat că prima categorie cuprinde numai operatori unari. Aceştia nu trebuie confundați cu operatorii binari cu acelaşi simbol.

Operatori	Asociativitate	tip operand1	tip operand2	tip rezultat
+ -	de la dreapta la stânga	integer real	- -	integer real
mod div	de la stânga la dreapta	integer	integer	integer
*	de la stânga la dreapta	integer integer real real	integer real integer real	integer real real real
/	de la stânga la dreapta	integer sau real	integer sau real	real

Ordinea de evaluare determinată de precedență şi asociativitate poate fi modificată grupând operațiile cu ajutorul parantezelor.

Asupra reprezentărilor binare a datelor numerice se pot executa o serie de operații specifice de obicei prelucrărilor în limbaj de asamblare:

- ***Complementarea reprezentării binare*** : predicatul ***bitnot(operand1, X)*** are ca efect legarea la variabila X a valorii întregi a cărei reprezentare binară se obține prin complementarea bit cu bit a reprezentării binare a operand1 (complementare înseamnă înlocuirea valorii 0 cu valoarea 1 şi invers).
- ***Deplasarea la stânga a reprezentării binare*** : predicatul ***bitleft(operand1, nr_poziții, X)*** are ca efect legarea la variabila X a valorii întregi a cărei reprezentare binară se obține prin deplasarea la stânga cu nr_poziții a codului binar al operand1 (deplasarea la stânga se obține prin eliminarea a unui număr egal cu nr_poziții de biți din stânga şi inserarea în dreapta a unui număr de nr_poziții de biți egali cu zero).
- ***Deplasarea la dreapta a reprezentării binare***: predicatul ***bitright(operand1, nr_poziții, X)*** are ca efect legarea la variabila X a valorii întregi a cărei reprezentare binară se obține prin deplasarea la dreapta cu nr_poziții a codului binar al operand1 (deplasarea la dreapta se obține prin eliminarea a unui număr egal cu nr_poziții de biți din dreapta şi inserarea în stânga a unui număr de nr_poziții de biți egali cu bitul cel mai semnificativ).

- ***Şi*** : predicatul ***bitand(operand1, operand2, X)*** are ca efect legarea la variabila X a valorii întregi a cărei reprezentare binară se obține prin operația "şi" asupra reprezentărilor binare ale operanzilor.
- ***Sau*** : predicatul ***bitor(operand1, operand2, X)*** are ca efect legarea la variabila X a valorii întregi a cărei reprezentare binară se obține prin operația "sau" asupra reprezentărilor binare ale operanzilor.
- ***Sau exclusiv*** : predicatul ***bitxor(operand1, operand2, X)*** are ca efect legarea la variabila X a valorii întregi a cărei reprezentare binară se obține prin operația "sau exclusiv" asupra reprezentărilor binare ale operanzilor.

Execuția operațiilor "şi", "sau" şi "sau exclusiv" se face executând operațiile respective asupra fiecărui bit din reprezentările binare ale celor doi operanzi conform următoarelor reguli:

x	y	şi	sau	sau exclusiv
1	1	1	1	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

Ca exemplu, considerăm următoarea interogare:

```
Goal: bitand(98, 95, X)
X = 66

1 Solution
```

Valoarea 66 a fost obținută ca efect al executării operației "şi" asupra reprezentării binare (pe 16 biți) a numărului 98 şi a numărului 95:

98 = 0000000001100010
95 = 0000000001011111
0000000001000010 = 66

În prelucrările aritmetice realizate în PROLOG semnul "=" joacă un rol special. El este privit ca un operator cu doi operanzi: operand1 = operand2. Următoarele reguli determină comportarea acestui operator:

- Dacă unul din operanzi este variabilă nelegată şi celălalt este un obiect, atunci operatorul leagă variabila la obiectul respectiv.
- Dacă un operand este variabilă legată şi celălalt este un obiect, atunci operatorul testează dacă valoarea legată este aceeaşi cu obiectul respectiv.
- Dacă un operand este variabilă liberă, iar celălalt este variabilă legată, atunci operatorul leagă variabila liberă la obiectul legat de cealaltă variabilă.
- Dacă ambii operanzi sunt variabile legate, atunci operatorul testează dacă valorile legate sunt aceleaşi.

Există predicate predefinite specializate care controlează inegalitatea dintre expresiile numerice:

>	mai mare
<	mai mic
>=	mai mare egal
<=	mai mic egal
<>	diferit

Următorul tabel conţine o listă de predicate predefinite şi funcţiile aritmetice cărora le sunt asociate.

Predicat	Funcţia calculată
round(x)	rotunjeşte la cel mai apropiat întreg
trunc(x)	trunchiază
abs(x)	valoarea absolută
sqrt(x)	radical de ordinul 2
exp(x)	e^x
ln(x)	logaritm natural
log(x)	logaritm în baza 10
random(x)	leagă la x un număr aleator din [0, 1)
sin(x)	sinus; x se consideră exprimat în radiani

cos(x)	cosinus; x se consideră exprimat în radiani
tan(x)	tangenta; x se consideră exprimat în radiani
arctan(x)	arctangenta

Pentru exemplificarea prelucrărilor aritmetice, să considerăm problema calculării dobânzilor totale pentru împrumutul acordat de o bancă, în condițiile în care rambursarea se face în tranşe lunare egale. Punem în evidență două modalități de rezolvare a cestei probleme. Prima dintre ele este o soluție recursivă. În acest scop se defineşte clauza:

```
dobanda(Imprumut, Rata_curenta, Rata_anuala_dobanda, Dobanda_totala)
```

în care variabilele folosite au următoarea semnificație:

Imprumut - împrumutul acordat

Rata_curenta - rata curentă lunară

Rata_anuala_dobanda - rata anuală a dobanzii

Dobanda_totala - dobânda totala (de achitat până la rambursarea integrală a împrumutului.

Dobânda totală este suma dobânzilor calculate prin aplicarea ratei lunare a dobânzii asupra împrumutului de restituit, care se diminuează progresiv cu sumele rambursate lunar. Calcularea dobânzii încetează odată cu rambursarea integrală a împrumutului. Deci condiția de margine poate fi enunțată astfel: "pentru un împrumut de restituit egal cu zero dobânda totală este zero, indiferent de celelalte elemente". Programul care rezolvă problema recursiv este:

```
predicates
        dobanda(real,real,real,real)
clauses
        dobanda(0,_,_,0):-!.
        dobanda(Imprumut,Rata_curenta,Rata_anuala_dobanda,Dobanda_totala):
-
                Sold=Imprumut-Rata_curenta,

dobanda(Sold,Rata_curenta,Rata_anuala_dobanda,Dobanda_urmatoare),
```

```
Dobanda_totala=Dobanda_urmatoare+Imprumut*Rata_anuala_dobanda/1200.
```

În cea de a doua regulă semnificația variabilelor este următoarea:

Sold - împrumutul de restituit în luna următoare (i.e. Imprumut-Rata_curenta)

Dobanda_urmatoare - dobânda aferentă lunilor următoare

Dobanda_totala – dobânda totală este Dobanda_urmatoare + dobânda datorată pentru luna în curs, i.e Imprumut*Rata_anuala_dobanda/1200.

Cea de a doua soluție a problemei utilizează o relație matematică de calcul direct:

```
predicates
        dobanda(real,real,real,real)
clauses
        dobanda(Imprumut,Rata_curenta,Rata_anuala_dobanda,Dobanda_totala):
-
        Dobanda_totala=Imprumut*Rata_anuala_dobanda*(Imprumut+Rata_curen
        ta)/
                                                (2400*Rata_curenta).
```

Iată şi răspunsul ambelor programe la interogarea:

```
Goal : dobanda(5000000,100000,20,X)
X = 2125000
1 Solution
```