

Laborator 2

- Copiati urmatorul program Prolog intr-un fisier cu extensia.pl;

```
culoare(rosu).
culoare(galben).
culoare(alb).
frunze(alterne).
frunze(mari).
frunze(nervuri_paralele).
petale(multepetale).
petale(petalegalbene).
petale(grupate_trompeta).
fl(trandafir, rosu, alterne, multepetale).
fl(floarea_soarelui, galben, mari, petalegalbene).
fl(crin, alb, nervuri_paralele, grupate_trompeta).

floare(A, X, Y, Z):-culoare(X), frunze(Y), petale(Z),
fl(A,X,Y,Z).
```

- Setati trace –ul:

```
?-trace.
```

- Urmariti cum se efectueaza procesul de backtracking in urma apelarii:

```
?-floare(crin, X,Y,Z).
```

- Puteti elimina trace–ul astfel:

```
?-notrace.
```

- **Predicatul cut (!)**

Daca Prologul gaseste predicatul cut intro regula, nu va mai efectua backtracking. Toate deciziile luate vor ramane finale.

- Cut rosu: modifica corespondenta dintre semnificatia declarativa si cea procedurala a programului Prolog;

- Cut verde: ajuta la cresterea eficientei programului.

```
% cut verde
min1(X, Y, X):-X=<Y,! .
min1(X, Y, Y):-X>Y.
```

```
% cut rosu
min2(X, Y, X):-X=<Y,! .
min2(X, Y, Y).
```

Daca se schimba ordinea clauzelor pentru min2, solutiile rezultate pot fi gresite:

```
min2(X, Y, Y).
min2(X, Y, X):-X=<Y,! .
```

- **Predicatul fail (fail)**

Fail, intr-o conjunctie de scopuri (de obicei la sfarsit), forteaza intrarea in procesul de backtracking;

Copiat programul de mai jos intr-un fisier cu extensia pl si interogati-l asa cum este indicat. Observati efectul predicatului fail.

```
getX('X1').
getZ('Z1').
getX('X2').
getZ('Z2').
wr:-getX(X), getZ(Z), wrXZ(X, Z), fail.
wrXZ(X, Z):-write(X), write(' '), write(Z), write(' '), nl, fail.
```

Interogare:

?- wr

- nl determina programul sa sara la linie noua.
- write(X) afiseaza pe ecran valoarea cu care a fost instantiat X

Aritmetica in Prolog

•= = egalitate de termeni

$1+2 = 2+1$ no

•:= = egalitate de valori numerice a doua expresii aritmetice

$1+2 := 2+1$ yes

•\= = termeni diferiti

•X is Y: variabila X este instantiata cu valoarea Y.

X is 3+2.

Functia $f(x) = \begin{cases} 0, & x \leq 3 \\ 2, & x \in (3,6] \\ 4, & x > 6 \end{cases}$ cum ar putea fi implementata in Prolog?

Interogati cu: $f(1,Y), 2 < Y$.

Este optima implementarea voastra? Daca nu, cum ati putea sa o optimizati?