Laborator 6-7

2018-2019

Programare Logică

Operatori în Prolog

- În Prolog putem defini operatori noi astfel

  `:- op(Precedence,Type,Name).`

  Atenție! Definirea unui operator este sintactică, nu spune nimic despre semnificația sa, care trebuie definită separat.

Exemplu

```
:- op(500, xf, is_dead).

kill(marsellus,zed).
is_dead(X) :- kill(_,X).
```

Citiți mai multe despre operatori:
SWI-Prolog
Learn Prolog Now!

Calculul propozițional calsic

- În laboratorul trecut am definit limbajul logicii propoziționale clasice:

```
is_var(a). is_var(b).
:- op(630, xfy, sau).
:- op(620, xfy, si).
:- op(610, fy, nu).
:- op(640, xfy, imp).

?- X= a si nu b.
X = a si nu b.
```

- ați definit predicatul `find_vars` care determină mulțimea variabilelor unei formule:

```
?- find_vars(a imp (b imp (c sau a)), [], Vars).
Vars = [c, b, a].
```

- În acest laborator vom implementa algoritmul Davis-Putnam.

În continuare vom presupune ca formulele sunt corecte sintactic.

Practică

- Lucrați în fișierul `sat.pl`.

Exercițiul 1: scrieți un predicat `inloc_imp(X,Y)` cu următorul efect: `Y` este `X` în care toate implicatiile au fost scrise folosind `sau` și `nu`.

Amintiți-vă că $\varphi \to \psi \sim \neg\varphi \vee \psi$

Exemplu:

```
?- inloc_imp(a imp b imp c, Y).
Y = nu a sau nu b sau c
```

Forma NNF

Fie φ o formulă din calculul propozițional clasic care nu conține implicații.

- Formula φ este în forma NNF dacă negația este numai pe variabile.

  Exemple:
  $p \wedge \neg q$ este în formă NNF
  $\neg(p \vee q) \wedge r$ nu este în formă NNF

- Formula φ poate fi adusă la forma NNF folosind:
  - regulile De Morgan
  $$\neg(\varphi \vee \psi) \sim \neg\varphi \wedge \neg\psi,$$
  $$\neg(\varphi \wedge \psi) \sim \neg\varphi \vee \neg\psi,$$
  - principiului dublei negații
  $$\neg\neg\psi \sim \psi$$

Exercițiul 2: scrieți un predicat `is_nnf(X)` care să întoarcă `true` dacă `X` este în formă NNF; vom presupune ca `X` nu conține implicații.

Practică

Exercițiul 3: scrieți un predicat `nnf(X,Y)` astfel încât `Y` să fie `X` în formă NNF; vom presupune ca `X` nu conține implicații.

Exemplu:

```
?- nnf(nu nu a, Y).
Y = a .

?- nnf(a, Y).
Y = a .

?- nnf(a si nu nu b, Y).
Y = a si b .

?- nnf(nu (a sau b) si nu nu c, Y).
Y = (nu a si nu b)si c
```

FND si FNC

- Un literal este o variabilă sau negaţia unei variabile.
- O formă normală disjunctivă (FND) este o disjuncţie de conjuncţii de literali.
- O formă normală conjunctivă (FNC) este o conjuncţie de disjuncţii de literali.
- Dacă φ este o formulă în forma NNF atunci ea poate fi adusă la FNC sau FND folosind distributivitatea

$$\varphi \vee (\psi \wedge \chi) \sim (\varphi \vee \psi) \wedge (\varphi \vee \chi),$$
$$(\psi \wedge \chi) \vee \varphi \sim (\psi \vee \varphi) \wedge (\chi \vee \varphi)$$

FNC

Orice formulă poate fi adusa la FNC prin urmatoarele transformări:

1. înlocuirea implicațiilor și echivalențelor

$$\varphi \to \psi \sim \neg\varphi \vee \psi,$$
$$\varphi \leftrightarrow \psi \sim (\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi)$$

FNC

Orice formulă poate fi adusa la FNC prin urmatoarele transformări:

1. înlocuirea implicațiilor și echivalențelor

$$\varphi \to \psi \sim \neg\varphi \vee \psi,$$
$$\varphi \leftrightarrow \psi \sim (\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi)$$

2. regulile De Morgan

$$\neg(\varphi \vee \psi) \sim \neg\varphi \wedge \neg\psi,$$
$$\neg(\varphi \wedge \psi) \sim \neg\varphi \vee \neg\psi,$$

3. principiului dublei negații

$$\neg\neg\psi \sim \psi$$

4. distributivitatea

$$\varphi \vee (\psi \wedge \chi) \sim (\varphi \vee \psi) \wedge (\varphi \vee \chi),$$
$$(\psi \wedge \chi) \vee \varphi \sim (\psi \vee \varphi) \wedge (\chi \vee \varphi)$$

Exemple

1. Determinați FNC pentru formula
$(\neg p \to \neg q) \to (p \to q)$

Exemple

1. Determinați FNC pentru formula
$(\neg p \to \neg q) \to (p \to q)$
$\sim \neg(\neg p \to \neg q) \vee (p \to q)$

Exemple

1. Determinați FNC pentru formula
$(\neg p \rightarrow \neg q) \rightarrow (p \rightarrow q)$
$\sim \neg(\neg p \rightarrow \neg q) \vee (p \rightarrow q)$
$\sim \neg(p \vee \neg q) \vee (\neg p \vee q)$

Exemple

1. Determinați FNC pentru formula

$(\neg p \to \neg q) \to (p \to q)$
$\sim \neg(\neg p \to \neg q) \vee (p \to q)$
$\sim \neg(p \vee \neg q) \vee (\neg p \vee q)$
$\sim (\neg p \wedge q) \vee (\neg p \vee q)$

Exemple

1. Determinați FNC pentru formula
$(\neg p \to \neg q) \to (p \to q)$
$\sim \neg(\neg p \to \neg q) \vee (p \to q)$
$\sim \neg(p \vee \neg q) \vee (\neg p \vee q)$
$\sim (\neg p \wedge q) \vee (\neg p \vee q)$
$\sim (\neg p \vee \neg p \vee q) \wedge (q \vee \neg p \vee q)$

2. Determinați FNC pentru formula
$\neg((p \wedge q) \to q)$

Exemple

1. Determinați FNC pentru formula
$(\neg p \to \neg q) \to (p \to q)$
$\sim \neg(\neg p \to \neg q) \vee (p \to q)$
$\sim \neg(p \vee \neg q) \vee (\neg p \vee q)$
$\sim (\neg p \wedge q) \vee (\neg p \vee q)$
$\sim (\neg p \vee \neg p \vee q) \wedge (q \vee \neg p \vee q)$

2. Determinați FNC pentru formula
$\neg((p \wedge q) \to q)$
$\sim \neg(\neg(p \wedge q) \vee q)$
$\sim p \wedge q \wedge \neg q$

Practică

Exercițiul 4: scrieți un predicat `cnf(X,Y)` astfel încât `Y` să fie `X` în formă cnf.

Exemplu:

```
?- cnf(a,Y).
Y = a .

?- cnf(a imp b,Y).
Y = nu a sau b .

?- cnf(a imp b imp c,Y).
Y = nu a sau nu b sau c .

?- cnf(a imp b imp (c si d),Y).
Y = (nu a sau nu b sau c)si(nu a sau nu b sau d)
```

Practică

Exercițiul 5: scrieți un predicat `toclausal(F,L)` astfel încât `L` să fie forma clauzală a lui `F`, unde `F` este în formă cnf.

Exemplu:

```
?- toclausal((a sau nu b) si (nu c sau nu a) si d, L).
L = [[a, nu b], [nu c, nu a], [d]].
```

- Scrieți un predicat `remove_trivial(LC,L)` unde `L` și `LC` sunt forme clauzale, `L` fiind obținută prin eliminarea clauzelor triviale din `LC`.

  Exemplu:

  ```
  ?- remove_trivial([[a,b, nu a],[c],[b ,d, e, nu d]],
  L). L = [[c]]
  ```

Practică

Exercițiul 6: Combinând predicatele pe care le-ați scris până acum, scrieti un predicat `clausal_form(F,LC)` în care `F` este o formulă corectă (nu neapărat în formă cnf) iar `LC` este forma ei clauzală.

Exemplu:

```
?- clausal_form(a si (b imp (nu a sau c)), LC).
LC = [[a], [nu b, nu a, c]]
```

Practică

Exercițiul 7: Scrieți două predicate `list1(LC,X,L1)` și `list2(LC,X,L2)` în care:

- `LC` este o formă clauzală,
- `X` este o variabila care apare în `LC`,
- `L1` este lista clauzelor care contin literalul `X`,
- `L2` este lista clauzelor care contin literalul `nu X`.

Exemplu:

```
?- list1([[a],[nu b, nu a, c]],a, L1).
L1 = [[a]].

?- list2([[a],[nu b, nu a, c]],a, L2).
L2 = [[nu b, nu a, c]].
```

Practică

Exercițiul 8: Scrieți un predicat `allresolvents(LC,X, LR)`: în care `LR` este lista rezolvenților obținu cti prin aplicarea rezoluției pentru variabila `X` și clauzele din `LC`.

Exemplu:

```
?- allresolvents([[a],[nu a, c], [d], [a, b]], a, LR).
LR = [[c], [b, c]].
```

Atenție:

- pentru eliminarea duplicatelor folosiți predicatul predefinit `list_to_set/2`.
- poate fi util predicatul predefinit `findall/3`.

Practică

Exercițiul 9: Scrieți un predicat `resolution1(LV, LC, Sat)` în care `LC` este o formă clauzală,
`LV` este listă variabilelor care apar în `LC`,
`Sat` este rezultatul algoritmului Davis-Putnam având la intrare lista de clauze `LC` astfel încât variabila considerată la fiecare pas este dată de lista `LV`.
De exemplu, dacă `LV=[b,a,c]` la primul pas din algoritmul Davis-Putnam se vor considera clauzele care conțin ca literali `b` și `nu b`, la al doilea pas a și nu a, șamd.

Exemplu:

```
?- resolution1([a,b,c,d],[[a],[nu a, c], [d], [a, b]],
Sat).
Sat = [] . % lista vida de clauze
?- resolution1([a,c],[[a],[nu a, c], [nu c]], Sat).
Sat = [[]]. % lista contine numai clauza vida
```

Practică

Exercițiul 10: Combinând predicatele scrise până acum, scrieți un predicat `resolution(F, Sat)` în care `F` este o formulă corectă din calculul propozițional iar `Sat` este ieșirea algoritmului Davis-Putnam, i.e:

`Sat=[]` dacă `F` este satisfiabilă,

`Sat=[[]]` dacă `F` nu este satisfiabilă.

Exemplu:

```
?- resolution(a imp (a sau b), Sat).
Sat = [].

?- resolution(a si nu a, Sat).
Sat = [[]].
```

Succes la examen!