

Introducere în Limbajul Prolog

Cuvantul PROLOG vine de la **PRO**graming in **LOG**ic. Prologul este un limbaj *declarativ*, aceasta însemnând că un program nu descrie cum se calculează o soluție, ci constă din o mulțime de *fapte* și legături logice (*reguli*) care descriu relațiile pe care se bazează aplicația de rezolvat. În Prolog, pentru obținerea unei soluții, utilizatorul pune mai degrabă o întrebare decât să ruleze programul. Când se pune o întrebare, sistemul caută în baza de fapte și reguli pentru a găsi, prin deducție logică, răspunsul.

Cum lansăm Prologul?

Pentru dezvoltarea programelor, vom folosi implementarea SWI-PROLOG. Lansarea acestuia se face apăsând butonul Start și efectuând selecțiile Programs -> SWI Prolog -> SWI Prolog. Pe ecran va apare fereastra aplicației și prompterul ? care arată că programul așteaptă comenzi.

Comenzi uzuale

? – pwd.

Afișează directorul curent.

?-ls.

Afișează fișierele din directorul curent.

?-cd('cale').

Schimba directorul curent.

Exemplu:

?-cd('c:/windows').

Directorul curent devine c:\windows.

?-halt.

Această comandă încheie sesiunea de lucru curentă în SWI-Prolog.

Important: fiecare enunț în Prolog se termină cu caracterul punct (.).

Pentru a se afișa informații cu privire la o comandă, apelăm

?-help(ume_comanda).

Exemplu:

?-help(halt).

Faptele

Faptele reprezintă cea mai simplă formă de predicat din Prolog. Sintaxa pentru fapte este următoarea:

nume_relatie(arg1, arg2, ..., argN).

unde

nume_relatie este numele relației (al predicatului)

arg1, arg2, ... - argumentele

N reprezintă aritatea predicatului nume_relatie.

Bineînțeles, orice fapt se încheie cu punct (.). Un predicat de aritate 0 se poate defini simplu: nume_predicat.

Exemplu de fapt:
frate(dan, maria).

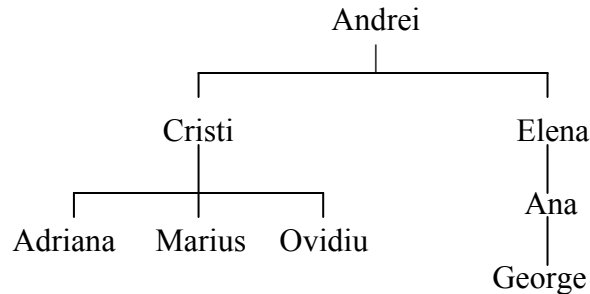
'frate' reprezintă numele relației, iar dan și maria reprezintă argumentele care fac parte din această relație. Important de observat este că atât numele relației, cât și *constantele* 'dan' și 'maria' sunt scrise începând cu literă mică.

Exemplu

Să presupunem că avem relațiile existente dintr-o familie și vrem să punem întrebări de genul "este marian înrudit cu ...?" sau "spuneți care sunt toate surorile lui marian" ș.a.m.d.

Ca argumente vom avea oameni, iar proprietățile pe care vrem să le definim vor fi tata, mama, frate, sora, Alegem trei predicate de bază, barbat, femeie și parinte și în funcție de ele vom descrie relațiile dintr-o familie.

Presupunem că avem ca exemplu următoarea familie:



În Prolog, reprezentăm aceasta sub formă de fapte astfel:

barbat(andrei).
barbat(cristi).
barbat(marius).
barbat(ovidiu).
barbat(george).

femeie(adriana).
femeie(elena).
femeie(ana).

parinte(andrei, cristi).
parinte(andrei, elena).
parinte(cristi, adriana).
parinte(cristi, marius).
parinte(cristi, ovidiu).
parinte(elena, ana).
parinte(ana, george).

Introduceți acest program într-un fișier folosind un editor text (de ex.: Notepad) și salvați-l într-un fișier cu extensia pl, de exemplu prog1.pl.

Observație: faptele construite cu același nume de relație (cum ar fi 'barbat') se afla unul lângă altul în fișierul sursă. Aceste fapte formează definiția relației respective.

Am salvat așadar programul în fișierul 'prog1.pl'; în continuare, scrieți în fereastra de Prolog

[prog1].

și apăsați tasta ENTER. Aceasta duce la încărcarea în memorie a programului. Realizați această *compilare* ori de câte ori modificați ceva în program.

Putem în acest moment formula întrebări de forma: este Andrei părintele lui Cristi? sau este Elena părintele lui Ovidiu? Întrebările se introduc la prompterul ?. Pentru a le formula pe cele de mai sus, procedăm astfel:

?- parinte(andrei, cristi).

Sistemul va consulta baza de fapte de care dispune și va răspunde cu

Yes

La întrebarea

?-parinte(elena, ovidiu).

răspunsul va fi

No

pentru că sistemul nu găsește în baza de cunoștințe nici un fapt care să corespundă acestei întrebări.

Dacă dorim să aflăm care sunt copiii lui Andrei, trebuie să introducem noțiunea de variabilă. O variabilă poate fi văzută ca o locație de memorie care primește în timpul procesului de raționament o valoare. Odata asignată, valoarea unei variabile nu mai poate fi schimbată. Numele unei variabile trebuie să înceapă cu literă mare.

?-parinte(andrei, Copil).

'Copil' reprezintă o variabilă. Sistemul va căuta în baza de fapte un fapt care să corespundă interogării și se va opri la parinte(andrei, cristi). Răspunsul la interogare va fi: Copil = cristi

Pentru a obține toți copiii lui Andrei, vom folosi o interogare mai complicată:

?-parinte(andrei, Copil), write_ln(Copil), fail.

Răspunsul va fi:

cristi

elena

No

Virgula desemnează în Prolog operatorul logic 'și', fail este o relație (predicat) predefinită care eșuează întotdeauna determinând căutarea de noi soluții, iar write_ln afișează valoarea din paranteză și trece pe rândul următor.

Cu ajutorul operatorului ',' putem construi interogări mai complicate precum:

?-parinte(andrei, Fiica), femeie(Fiica).

Acesta determină o fiică a lui Andrei; răspunsul va fi

Fiica = elena

Exercițiu: Contruiți o interogare cu ajutorul căreia să determinați nepoții lui Andrei.

Variabilele pot să apară și în cadrul faptelor dintr-un program pentru a desemna adevăruri general valabile. Dacă adăugăm la exemplul nostru faptul 'place(X, garoafe)',

aceasta înseamnă că tuturor persoanelor le plac garoafele; altfel spus, oricare ar fi X, are loc relația $\text{place}(X, \text{garoafe})$.

Regulile

Până acum am definit relațiile ca mulțimi de fapte. O relație poate fi definită și în funcție de alte relații existente în program. Pentru aceasta vom folosi construcții sintactice mai complicate numite *reguli*.

O regulă are următoarea sintaxă:

$\text{nume_relatie}(\text{arg1}, \dots, \text{argN}) :- \text{nume_relatie_1}(\dots), \dots, \text{nume_relatie_M}(\dots)$.

Exemplu: putem defini predicatul 'frate' astfel – Copil1 este frate cu Copil2 dacă Parinte este părintele lui Copil1, Parinte este părintele lui Copil2, Copil1 este diferit de Copil2 și Copil2 este de sex masculin. Această definiție va fi codificată prin următoarea regulă care va fi adăugată la programul prog1.pl:

```
frate(Copil1, Copil2) :- parinte(Parinte, Copil1), parinte(Parinte, Copil2), Copil1 \=
                        Copil2, barbat(Copil2).
```

Operatorul :- are semnificația 'dacă'.

După ce adăugați regula și reîncărcați programul (cu [prog1].), introduceți interogarea:

?- frate(ovidiu, X).

Răspunsul va fi:

X = marius

Pentru a putea urmări modul în care este rezolvată interogarea anterioară, introduceți comanda

?-trace.

Și apoi repetați interogarea

?-frate(ovidiu, X).

Pe ecran vor apărea diferite subscopuri care trebuie rezolvate de sistemul prolog pentru a obține răspunsul la interogare. Pentru a trece de la un subscop la altul, apăsați tasta ENTER.

Observații

1. Partea din stânga operatorului :- în definiția unei reguli este numită capul regulii iar cea din dreapta corpul regulii.
2. Un fapt poate fi văzut ca o regulă fără corp (fără condiție).
3. Regulile și faptele care definesc aceeași relație (au același nume de relație în cap) trebuie grupate în sursa programului.
4. În Prolog, relațiile sunt numite de obicei predicate.

Exerciții

1. Definiți relațiile fiu și fiica.
2. Modificați regula frate pentru a obține regula pentru relația sora.
3. Pornind de la programul anterior și de la un predicat $\text{cuplu_casatorit}(\text{Nume_sot}, \text{Nume_sotie})$, adăugați fapte care descriu acest predicat și definiți relațiile soacra, cumnat și ginere.