

Laborator 12a

Entitatile limbajului Prolog

Un program Prolog contine urmatoarele entitati:

- **fapte** despre obiecte si relatiile existente intre aceste obiecte;
- **reguli** despre obiecte si relatiile dintre ele, care permit deducerea (inferarea) de noi fapte pe baza celor cunoscute;
- **intrebari**, numite si **scopuri**, despre obiecte si relatiile dintre ele, la care programul raspunde pe baza faptelor si regulilor existente.

Multimea faptelor si a regulilor unui program Prolog formeaza **baza de cunostinte** Prolog.

1. Fapte

Faptele sunt predicate de ordinul intai. Ele stabilesc relatii intre obiectele universului problemei.

Exemple

papagal(coco).

iubeste(mihai,maria).

iubeste(mihai,ana).

frumoasa(ana).

bun(gelu).

Argumentele predicatelor Prolog pot fi constante, variabile sau structuri.

2. Scopuri

Obtinerea rezultatului unui program Prolog se face prin fixarea unor scopuri care pot fi adevarate sau false, in functie de continutul bazei de cunostinte Prolog. Scopurile sunt predicate pentru care se doreste aflarea valorii de adevar. Cum scopurile pot fi vazute ca intrebari, rezultatul unui program Prolog este raspunsul la o intrebare (sau la o conjunctie de intrebari).

Exemple

Considerand baza de cunostinte specificata anterior, se pot pune diverse intrebari, cum ar fi:

?- iubeste(mihai, maria).

Yes

?- papagal(coco).

Yes

?- papagal(mihai).

No

3. Constante

Constantele definesc obiecte specifice, particulare, sau relatii particulare. Exista doua tipuri de constante: **atomi** si **numere**. Atomii sunt constante simbolice care incep cu o litera mica si pot contine litere, cifre si caracterul “_”. Exista si alte caractere ce pot forma **atomi speciali**, care au o semnificatie aparte in limbaj.

Atomii pot desemna:

- obiecte constante care sunt argumentele predicatelor, de exemplu atomii mihai si maria in faptul iubeste(mihai,maria);
- predicate Prolog, fie cele definite de utilizator, fie cele predefinite in sistem; de exemplu atomul iubeste in faptul iubeste(mihai,maria);
- atomi speciali, de exemplu atomii “:-” si “?-”

Numerele pot fi intregi sau reale.

4. Variabile

Predicatele Prolog, ca admit ca argumente si obiecte generice numite **variabile**. Numele unei variabile incepe cu litera mare si poate contine litere, cifre si caracterul “_”. Numele unei variabile poate fi si simbolul “_”, ceea ce indica o *variabila anonima*. Utilizarea unei variabile anonime semnifica faptul ca nu conteaza valoarea la care se va instantia acea variabila. De exemplu, interogarea

?- iubeste(_maria).

are sensul „exista cineva care o iubeste pe Maria?”.

O variabila poate fi *instantiata* (legata) daca exista un obiect asociat acestei variabile, sau *neinstantiata* (libera) daca nu se stie inca ce obiect va desemna variabila.

La fixarea unui scop Prolog care contine variabile, acestea sunt neinstantiate iar sistemul incearca satisfacerea acestui scop cautand in baza de cunostinte o instantiere adecvata a variabilelor din scopul dat.

La incercarea de satisfacere a scopului, cautarea se face intotdeauna pornind de la inceputul bazei de cunostinte.

In cazul in care exista mai multe fapte (sau reguli) in baza de cunostinte care unifica cu intrebarea pusa, exista mai multe raspunsuri la intrebare, corespunzand mai multor solutii ale scopului fixat. Prima solutie este data de prima unificare si exista atatea solutii cate unificari diferite exista. La realizarea oricarei unificari se marcheaza faptul (sau regula) care a unificat. La obtinerea urmatoarei solutii, cautarea este reluata de la marcaj in jos in baza de cunostinte. Obtinerea primei solutii este de obicei numita **satisfacerea scopului** iar obtinerea altor solutii, **resatisfacerea scopului**.

Pentru a obtine o singura solutie, dupa satisfacerea scopului se tasteaza "Enter". Pentru a cere o resatisfacere a scopului se tasteaza ";". In cazul in care, un scop nu poate fi (re)satisfacut, sistemul raspunde No.

Example

?- iubeste(mihai, X).

X = maria;

X = ana;

No

?- iubeste(X, Y).

X = mihai

Y = maria;

X = mihai

Y = ana;

No

5. Reguli

O **regula** Prolog exprima un fapt care depinde de alte fapte si este de forma:

$$S :- S_1, S_2, \dots, S_n.$$

cu semnificatia urmatoare: S daca S_1 si S_2 si ... si S_n .

Fiecare S_i , $i = 1, n$ și S au forma faptelor Prolog, deci sunt predicate, cu argumente constante, variabile sau structuri. Faptul S care definește regula, se numește **antetul** regulii, iar S_1, S_2, \dots, S_n formează **corpul** regulii și reprezintă conjunctia de scopuri care trebuie satisfăcute pentru ca antetul regulii să fie satisfăcut.

În condițiile existenței regulilor în baza de cunoștințe Prolog, unificarea scopului se încearcă atât cu fapte din baza de cunoștințe, cât și cu antetul regulilor din baza de cunoștințe. La unificarea unui scop cu antetul unei reguli, pentru a putea satisface acest scop trebuie satisfăcută regula. Aceasta revine la a satisface toate faptele din corpul regulii, deci conjunctia de scopuri (de la stânga la dreapta). Scopurile din corpul regulii devin subscopuri a căror satisfacere se va încerca printr-un mecanism similar cu cel al satisfacerii scopului inițial.

Exemple

Fie următoarea bază de cunoștințe:

| | |
|---|-----|
| frumoasa(ana). | % 1 |
| bun(vlad). | % 2 |
| cunoaste(vlad, maria). | % 3 |
| cunoaste(vlad, ana). | % 4 |
| iubeste(mihai, maria). | % 5 |
| iubeste(X, Y) :- bun(X), cunoaste(X, Y), frumoasa(Y). | % 6 |

Enunțul (6) definește o regulă Prolog; relația $iubeste(X, Y)$, fiind definită atât printr-un fapt (5) cât și printr-o regulă (6).

Satisfacerea scopului

?- iubeste(vlad, ana).

se va face în următorul mod. Scopul unifică cu antetul regulii (6) și duce la instanțierea variabilelor din regula (6): $X = vlad$ și $Y = ana$. Pentru ca acest scop să fie îndeplinit, trebuie îndeplinită regula, deci fiecare subscop din corpul acesteia. Aceasta revine la îndeplinirea scopurilor $bun(vlad)$, care reușește prin unificare cu faptul (2), $cunoaste(vlad, ana)$, care reușește prin unificare cu faptul (4), și a scopului $frumoasa(ana)$, care reușește prin unificare cu faptul (1).

Ce se întâmplă dacă se pune întrebarea:

?- iubeste(X, Y).

Prima solutie a acestui scop este data de unificarea cu faptul (5), iar raspunsul este:

$X = \text{mihai}, Y = \text{maria}.$

Sistemul Prolog va pune un marcaj in dreptul faptului (5) care a satisfacut scopul. Urmatoarea solutie a scopului iubeste(X,Y) se obtine incepand cautarea de la acest marcaj in continuare in baza de cunostinte. Scopul unifica cu antetul regulii (6) si se vor fixa trei noi subscopuri de indeplinit, bun(X), cunoaste(X,Y) si frumoasa(Y). Scopul bun(X) este satisfacut de faptul (2) si variabila X este instantiata cu valoarea vlad, $X = \text{vlad}$. Se incearca acum satisfacerea scopului cunoaste(vlad,Y), care este satisfacut de faptul (3) si determina instantierea $Y = \text{maria}$. Se introduce in baza de cunostinte un marcaj asociat scopului cunoaste(vlad,Y), care a fost satisfacut de faptul (3). Se incearca apoi satisfacerea scopului frumoasa(maria). Acesta esueaza. In acest moment sistemul intra intr-un proces de *backtracking* in care se incearca resatisfacerea scopului anterior satisfacut, cunoaste(vlad,Y), in speranta ca o noua solutie a acestui scop va putea satisface si scopul curent care a esuat, frumoasa(Y). Resatisfacerea scopului cunoaste(vlad,Y) se face pornind cautarea de la marcajul asociat scopului in jos, deci de la faptul (3) in jos. O noua solutie a scopului cunoaste(vlad,Y) este data de faptul (4) care determina instantierea $Y = \text{ana}$. In acest moment se incearca satisfacerea scopului frumoasa(ana). Cum este vorba de un nou scop, cautarea se face de la inceputul bazei de cunostinte si scopul frumoasa(ana) este satisfacut de faptul (1). In consecinta a doua solutie a scopului iubeste(X,Y) este obtinuta si sistemul raspunde:

$X = \text{vlad}, Y = \text{ana}.$

La incercarea de resatisfacere a scopului iubeste(X,Y), printr-un mecanism similar, se observa ca nu mai exista alte solutii. In concluzie comportarea sistemului Prolog este:

?- iubeste(X, Y).

$X = \text{mihai}$

$Y = \text{maria};$

$X = \text{vlad}$

$Y = \text{ana};$

No

Observatie: Acest laborator are ca sursa cartea "Tehnici de programare Prolog pentru inteligenta artificiala", autori: A.M. Florea, S. Radu, A.H. Mogos, Editura Printech, 2007