4. Logica cu predicate de ordin I şi SWI-Prolog.

4.1 Logica cu predicate de ordin I

La începutul anilor '70, R. Kowalski a propus o formulă logică referitoare la propozițiile P_1, ... P_n de tipul:

$$\mathbf{P_1 \wedge P_2 \wedge ... P_n \rightarrow C}$$

care are, în logica cu predicate de ordinul întâi, semnificația declarativă conform căreia $\mathbf{P_1 \wedge P_2 \wedge ... P_n}$ implică concluzia **C.** Cu alte cuvinte, dacă $\mathbf{P_1}$ **şi** $\mathbf{P_2}$ **... şi** $\mathbf{P_n}$ sunt fiecare adevărate atunci şi **C** este adevărat. Formula de mai sus poate fi scrisă sub forma unei condiționări:

C dacă $\mathbf{P}_1$ şi $\mathbf{P}_2$ ... şi $\mathbf{P}_n$

şi poate fi executată ca o procedură a unui limbaj de programare recursiv, unde **C** este antetul procedurii, iar $\mathbf{P}_1$, $\mathbf{P}_2$, ... $\mathbf{P}_n$ corpul acesteia. Aşadar, pe lângă interpretarea declarativă, logică, a unei astfel de formule, expresia poate fi interpretată procedural astfel:

*pentru a executa **C** se execută $\boldsymbol{P_1}$ şi $\boldsymbol{P_2}$ ... şi $\boldsymbol{P_n}$.*

Logica cu predicate de ordinul I a fost dezvoltată pentru a da posibilitatea exprimării raționamentelor despre obiecte complexe sau clase de obiecte şi despre relațiile existente între ele.

4.1.1 Alfabetul logicii cu predicate de ordin I

Alfabetul logicii cu predicate de ordinul I conține următoarele simboluri fundamentale:

- *simboluri pentru reprezentarea constantelor*,
- *simboluri pentru reprezentarea variabilelor*,
- *simboluri pentru reprezentarea funcțiilor* – O funcție este o reprezentare ce transformă o listă de constante într-o constantă. De exemplu, funcția *tata (ion)* transformă o persoană numită *ion* într-o altă persoană care este tatăl lui *ion*;
- *simboluri pentru reprezentarea predicatelor*. Numărul de argumente *n* al predicatului conferă anumite calități ale acestuia:
    - *n=0* → predicatul este o propoziție logică;

- <u>*n=1*</u> → predicatul specifică o proprietate a argumentului;
- <u>*n=2*</u> → predicatul specifică o relație binară;
- …………………
- <u>*n=m*</u> → predicatul specifică o relație m-ară;

Numărul de argumente *n* a predicatului se mai numeşte *aritatea* acestuia.

- <u>*virgule şi paranteze*</u> *, () ;*
- <u>*conectori logici*</u> ($\sim$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$) . Conectorii logici corespund intuitiv conjuncțiilor gramaticale utilizate în limbajul curent. Avem astfel următoarea corespondență:

➢ $\sim sau \neg$, negație: not

A	**~A**
a	f
f	a

➢ $\wedge$, conjuncție: şi

A	**B**	**A$\wedge$B**
a	a	a
a	f	f
f	a	f
f	f	f

➢ $\vee$, disjuncție: sau

A	**B**	**A$\vee$B**
a	a	a
a	f	a
f	a	a
f	f	f

În logica predicatelor, disjuncția este *inclusivă*. Astfel, se spune: "*Mă duc la cules de prune sau mă duc la adunat fân*" şi prin aceasta înțelegem că doar una din variante este posibilă şi nu amândouă.

- $\rightarrow$, implicație: dacă … atunci …

A	**B**	**A→B**
a	a	a
a	f	f
f	a	a
f	f	a

În logica predicatelor, implicația **A→B** ia valoarea *f* numai dacă A are valoarea *a* şi B are valoarea *f*. Altfel, dacă A este *f*, **A→B** ia valoarea *a* indiferent de valoarea lui B. Aceste proprietăți ale conectorului implicație şi partea din logica predicatelor pe care se bazează aceste proprietăți *nu au fost întotdeauna acceptate de diverse şcoli ale logicii*.

- $\leftrightarrow$, echivalență: … dacă şi numai dacă …

A	**B**	**A↔B**
a	a	a
a	f	f
f	a	f
f	f	a

- <u>*cuantificatori logici*</u> (cel existențial($\exists$) şi cel universal ($\forall$)). În logica predicatelor validitatea generală sau parțială a unei formule este reprezentată prin cuantificatorul universal $\forall$ şi cel existențial $\exists$.

Conceptele prezentate mai sus pot fi ilustrate prin următorul exemplu. Se dau următoarele propoziții în limbaj natural:

P1: Andrei este student.

P2: Mariei îi place sportul.

P3: Mariei îi place şcoala.

P4: Lui Andrei îi plac banii.

P5: Andrei îl place pe X dacă lui X îi place şcoala.

P6: X poate învăța Y dacă X este student şi dacă lui X îi place Y.

Prin introducerea constantelor "Andrei, Maria, sport, şcoala, bani", a variabilelor X şi Y, a predicatelor "student, place, poate_invata" formăm următorul program Prolog:

C1: student (andrei).

C2: place (maria, sport).

C3: place (maria, şcoala).

C4: place (andrei, bani).

C5: place (andrei, X) :- place (X, şcoala).

C6: poate_învăța (X, Y) :- student (X), place (X, Y).

Clauzele Horn C1, C2, C3 şi C4 sunt fapte, iar C5 şi C6 sunt clauze Horn ce constituie partea procedurală a programului.

4.1.2 Construcția formulelor bine formate în logica cu predicate de ordin I

În cazul logicii cu predicate de ordinul I, predicatele sunt funcții logice de mai multe argumente, argumentele predicatelor numindu-se *termeni*.

Dacă D este un domeniu de valori, atunci un termen se defineşte astfel:

1. *O constantă este un termen cu valoare fixă aparținând domeniului D.*
2. *O variabilă este un termen ce poate primi valori diferite din domeniul D.*
3. *Dacă f este o funcție de n argumente ($f : D^n \rightarrow D$) si $t_1 ... t_n$ sunt termeni, atunci $f(t_1 ... t_n)$ este termen.*
4. *Toți termenii sunt generați prin aplicarea regulilor 1÷3.*

Dacă P este un predicat de aritate n şi $t_1 ... t_n$ sunt termeni, atunci $P(t_1 ... t_n)$ *se numeşte atom sau formulă atomică.* Nici o altă expresie nu poate fi atom.

Se numeşte literal un atom sau un atom negat.

O formulă bine formată, in logica cu predicate de ordinul I, se defineşte astfel:

1. *Un atom este o formulă bine formată.*

2. *Dacă P este o formulă bine formată atunci:* $\sim P, (\exists x)P(x), (\forall x)P(x)$ *sunt formule bine formate.*
3. *Dacă P şi Q sunt formule bine formate atunci:* $P \wedge Q, P \vee Q, P \to Q, P \leftrightarrow Q$ *sunt formule bine formate.*
4. *Orice formulă bine formată este generată prin aplicarea de un număr finit de ori a regulilor (1)÷(3).*

O reprezentare intuitivă a modului de construire a cuvintelor, deci a formelor bine formate, în limbajul logicii cu predicate de ordinul I, este prezentată in Figura 4.1.

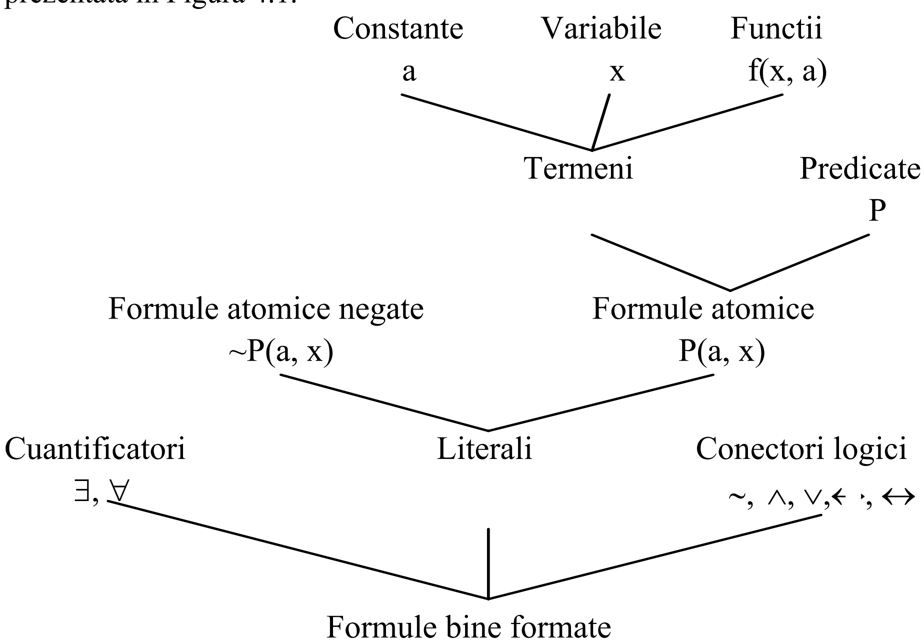

Figura 4.1 *Construcţia formulelor bine formate în logica cu predicate de ordinul I*

Exemple:

1. $(\exists x)(\exists y)(\exists z)Tata(x,y) \wedge Tata(y,z) \to Bunic(x,z)$ este o **formulă bine formată.** ***Tata(x,y), Tata(y,z)* si *Bunic(x,z)*** sunt **literali**, în acest caz literali pozitivi deci nenegaţi. **x, y, z** sunt **variabile**.

2. $(\forall x)(\exists y)(Egal(y, f(x)) \vee (\forall z)(Egal(z, f(x)) \to Egal(y,z)))$ este o **formulă bine formată. x, y** şi **z** sunt **variabile, f(x) funcție,** toate fiind considerate **termeni**. $Egal(y, f(x))$ este un **literal.**

3. $(\exists x)(Vanzator(x) \rightarrow \sim Onest(x))$ este o **formulă bine formată.** $Vanzator(x)$ si $\sim Onest(x)$ sunt **literali,** primul pozitiv şi cel de al doilea negativ.

4. $(\forall P)(P(x) \rightarrow Q(x))$ **nu** este o **formulă bine formată** deoarece cuantificatorii nu pot fi aplicaţi predicatelor. Acest lucru este posibil numai în logicile de ordin superior (logici de ordinul II).

5. $Om(\sim aristotel)$ **nu** este o **formulă bine formată** deoarece negaţia nu poate fi aplicată unei constante şi, în general, nici unui termen.

6. $(\exists x)(\exists y)Casatorit(Barbat(x), Femeie(y))$ **nu** este o **formula bine formată** deoarece argumentele predicatelor nu pot fi predicate.

Pe baza funcţiei de evaluare şi a domeniului de interpretare, formulele bine formate prezintă următoarele proprietăţi:

- O formulă bine formată este **validă (tautologie)** dacă formula este **adevărată** în orice interpretare.
- O formulă bine formată este **inconsistentă (contradicţie, nerealizabilă)** dacă formula are valoarea **fals** în orice interpretare.
- O formulă bine formată este **realizabilă (consistentă)** dacă există cel puţin o interpretare în care formula are valoarea **adevărat**.

4.2 Probleme rezolvate

1. Fie următoarele enunţuri:
   - **a)** ***Orice sportiv este puternic.***
   - **b)** ***Oricine este inteligent şi puternic va reuşi în viaţă.***
   - **c)** ***Oricine este puternic va reuşi în viaţă sau va ajunge bătăuş.***
   - **d)** ***Există un sportiv inteligent.***
   - **e)** ***Ionel este sportiv.***

Exprimând enunţurile în *logica cu predicate de ordinul I* se obţin următoarele formule bine formate:

- **a)** $(\forall x)(sportiv(x) \rightarrow puternic(x))$.
- **b)** $(\forall x)(\text{int}\,eligent(x) \wedge puternic(x) \rightarrow reuseste(x))$.
- **c)** $(\forall x)(puternic(x) \rightarrow (reuseste(x) \vee bataus(x)))$.

d) $(\exists x)(sportiv(x) \wedge \text{int}\,eligent(x))$.
e) $Sportiv('Ionel')$.

Axiomele se transformă în forma clauzală şi se obțin următoarele clauze:

a) $\sim sportiv(x) \vee puternic(x)$.
b) $\sim \text{int}\,eligent(x) \vee \sim puternic(x) \vee reuseste(x)$.
c) $\sim puternic(x) \vee reuseste(x) \vee bataus(x)$.
d) $\begin{matrix} sportiv(a) \\ \text{int}\,eligent(a) \end{matrix}$.
e) $sportiv(ionel)$.

Programul SWI-Prolog care se obține prin transformarea acestor clauze este următorul:

```
puternic(X) :- sportiv(X).
reuseste(X) :- inteligent(X), puternic(X).
sportiv(a).
inteligent(a).
sportiv(ionel).
```

2. Fie următoarele enunțuri:
   a) ***Orice număr rațional este un număr real.***
   b) ***Există un număr prim.***
   c) ***Pentru fiecare număr x există un număr y astfel încât x < y.***

Exprimând enunțurile în *logica cu predicate de ordinul I* se obțin următoarele formule bine formate:

a) $(\forall x)(rational(x) \rightarrow real(x))$.
b) $(\exists x)prim(x)$..
c) $(\forall x)(\exists y)maiMic(x, y)$.

Axiomele se transformă în forma clauzală şi se obțin următoarele clauze:

a) $\sim rational(x) \vee real(x)$.
b) $prim(a)$.
c) $maiMic(x, maiMare(x))$.

unde **maiMare(x)** este funcția care înlocuieşte variabila **y** cuantificată existențial.

Forma Prolog echivalentă a acestor clauze este:

```
real(X) :- rational(X).
prim(a).
maiMic(X, maiMare(X)).
```

unde **maiMare(X)** este o structură Prolog.

Observații:

Nu orice axiomă poate fi transformată în Prolog, astfel dintr-un anumit punct de vedere, puterea expresivă a limbajului este inferioară celei a logicii cu predicate de ordinul I.

Pe de altă parte, limbajul Prolog *oferă* o mulțime de *predicate de ordinul II*, adică predicate care acceptă ca argumente alte predicate Prolog, care nu sunt permise în logica cu predicate de ordinul I. Uneori, aceste predicate de ordinul II existente în Prolog pot fi folosite pentru a modela versiuni de programe Prolog echivalente cu o mulțime de axiome care nu au o reprezentare în clauze Horn distincte. *Acest lucru oferă limbajului Prolog o putere de calcul superioară celei din logica clasică.*

4.3 Probleme propuse

1. Se vor studia problemele rezolvate (problemele prezentate pe parcursul acestui capitol), încercând găsirea altor posibilități de soluționare a acestora. Utilizați şi alte scopuri (interogări) pentru a testa definițiile predicatelor introduse. Se atrage atenția asupra faptului că toate cunoştințele din acest capitol vor fi necesare şi în derularea celorlalte capitole.
2. Scrieți în Prolog următoarele expresii din vorbirea curentă:
    a) Elenei îi displace studiul.
    b) Cineva îl iubeşte pe Andrei.
    c) Andrei este prieten cu oricine pe care îl iubeşte, dacă şi această persoană îl iubeşte pe el.
    d) Mariana place pe oricine o admiră.
    e) Daniel se teme de oricine este mai mare decât el.
    f) Daniel este gelos pe oricine place Mariana, dacă această persoană nu este mai scundă decât Daniel.
3. Se dă o bază de date pentru reprezentarea cunoştințelor despre cărți, editurile care le publică şi magazinele care vând cărți. Faptele implicate sunt de tipul:

    vinde(Magazin, Editura) % magazinul Magazin vinde
    % cărți ale editurii Editura.

carte(Titlu, Editura)	% titlul Titlu este editat de editura % Editura
deschis(Magazin)	% dacă magazinul Magazin este % deschis
inchis(Magazin)	% dacă magazinul Magazin este % închis

Să se obțină răspunsuri la întrebări de tipul:

- Cine publică "Elemente de inteligență artificială şi Prolog"?
- Ce magazine vând cărți de la MatrixROM?
- De unde pot cumpăra "Programare în Prolog"?
- Există o editură care a publicat atât "Elemente de inteligență artificială şi Prolog" cât şi "Programare în Prolog"?
- Există două magazine care vând atât "..." cât şi "..."?
- Când sunt toate magazinele deschise/închise?