

## Capitolul 3

### Logica (calculul) cu predicate de ordinul I

După cum am punctat în finalul **Capitolului 2**, suntem nevoiți să extindem limbajul (logica) folosit(ă) până în prezent, deși rezultatele privind complexitatea (timp) a algoritmilor de testare a satisfiabilității formulelor **LP** sunt deja descurajatoare. Principalul argument este acela că **lumea reală nu poate fi modelată satisfăcător și simplu prin formule logice care utilizează doar variabile propoziționale și conectori**. De aceea principala modificare va fi de *natură sintactică*, manifestată prin adăugarea *cuantificatorilor logici* (există și pentru orice), împreună cu introducerea *variabilelor*, *constantelor*, *simbolurilor funcționale* (de aritate mai mare decât unu) și a *simbolurilor predicative*. Din acest punct de vedere, **limbajul logicii cu predicate de ordinul I** va conduce *sintactic* la o mulțime de formule, notată **LP1**, care va include strict clasa **LP**. *Semantica* pentru **LP1** va fi mai complexă, având însă la bază același concept de *structură* (*asignare*, *interpretare*) și fiind „consistentă” cu semantica **LP**. De altfel, **pentru LP1, dacă nu vom folosi concepte prezentate în mod explicit ca fiind noi, toate notațiile, noțiunile, rezultatele, etc., vor fi identice cu cele introduse pentru LP**.

#### §1. Sintaxa logicii cu predicate de ordinul I

Prin urmare, pentru a mări *puterea de exprimare*, în sensul informal precizat deja, a logicii folosite până în prezent (**LP**), vom

începe cu anumite transformări de **natură sintactică**. Mai precis, pentru a construi mulțimea de formule a logicii cu predicate de ordinul I, **LP1**, vom porni cu următoarele mulțimi de simboluri:

- $X = \{x_1, x_2, \dots\}$ : o mulțime cel mult numărabilă de **variabile funcționale**, sau, pe scurt, **variabile**.
- $P = \{P_0, P_1, \dots\}$ : o mulțime cel mult numărabilă de **simboluri predicative** (sau **predicate**, sau **relații**), cu arități. La rândul său, fiecare  $P_i$  este o mulțime cel mult numărabilă de **predicate de aritate i** ( $i \in \mathbb{N}$ ). Elementele lui  $P_0$  se mai numesc și **variabile predicative**.
- $F = \{F_0, F_1, \dots\}$ : o mulțime cel mult numărabilă de **simboluri funcționale** (sau **funcții**) cu arități, fiecare  $F_i$  fiind o mulțime cel mult numărabilă de **funcții de aritate i** ( $i \in \mathbb{N}$ ). Elementele lui  $F_0$  se numesc și **constante (funcționale)**.
- $\mathcal{C}_1 = \{\neg, \vee, \wedge\}$ : o mulțime de **conectori logici (conective logice)**, la care se pot adăuga, opțional, și alte simboluri cum ar fi  $\rightarrow$ ,  $\leftrightarrow$ , etc.
- $\mathcal{C}_2 = \{(\forall x) \mid x \in X\} \cup \{(\exists x) \mid x \in X\}$ : o mulțime de **cuantificatori (cuantori), universali**, respectiv **existențiali** ( $(\forall x)$  se citește „pentru fiecare  $x$ ”, sau „pentru oricare (orice)  $x$ ”, iar  $(\exists x)$  – „există  $x$ ”, „există măcar un  $x$ ”, etc.). Pentru început este suficient să considerăm doar cuantorii universali.

Ca și în cazul **LP**, vom porni cu alfabetul total, adică reuniunea mulțimilor precedente, împreună cu parantezele rotunde și virgula (pe scurt, **P**):

$$Alf = X \cup \left( \bigcup_{i=0}^{\infty} P_i \right) \cup \left( \bigcup_{i=0}^{\infty} F_i \right) \cup \mathbf{C}_1 \cup \mathbf{C}_2 \cup \mathbf{P}$$

Mulțimile  $\bigcup_{i=0}^{\infty} P_i$  și  $\bigcup_{i=0}^{\infty} F_i$  vor fi notate tot cu  $P$  respectiv  $F$ , atunci când nu există confuzii.

**Observație.** Prin „o mulțime cel mult numărabilă” înțelegem o mulțime numărabilă, finită sau vidă. Intuitiv, variabilele funcționale notate  $x$  vor fi *nume generice* pentru *elementele* dintr-un anumit domeniu, care va fi fixat ulterior (prin intermediul *funcției semantice*). Un simbol predicativ  $P$  de aritate  $i$  reprezintă o *relație  $i$ -ară neprecizată*, adică este un *nume generic* pentru orice funcție cu  $i$  argumente peste același domeniu, codomeniul fiind  $\{\mathbf{a}, \mathbf{f}\}$ , sau **B** (într-o *interpretare* vom avea  $P(\langle a_1, a_2, \dots, a_i \rangle) = 1$  dacă și numai dacă elementele  $a_1, a_2, \dots, a_i$  sunt în relația numită  $P$ ). Similar, un simbol funcțional  $f \in F_i$  este numele generic al oricărei funcții de  $i$  argumente, peste același domeniu și codomeniu. Pentru parantezele utilizate în scrierea cuantificatorilor ar trebui de fapt folosit un alt font (ca de altfel și pentru virgulă). ■

Mulțimea formulelor, **LP**<sub>Alf</sub> (indicele va fi pus în evidență doar atunci când lipsa sa ar putea genera confuzii) va fi definită structural, analog cu cazul **LP** (elemente mulțimilor **C**<sub>1</sub>, **C**<sub>2</sub>, **P**, exceptând virgula,

pot fi considerate ca fiind operatori pe şiruri de caractere, de aritate 1 sau 2).

**Definiția 3.1 (sintaxa LP1).** Fie  $A/f$  alfabetul fixat anterior. Atunci mulțimea **formulelor calculului cu predicate de ordinul I**,  $LP1_{A/f}$ , este dată constructiv prin:

**Baza.** Se definește mulțimea **formulelor atomice**, notată cu  $At$ , prin:

- (i)  $P_o \subseteq At$  (variabilele predicative sunt formule atomice).
- (ii) Pentru fiecare  $n \in \mathbf{N}^*$ , pentru fiecare  $P \in P_n$ , pentru fiecare  $t_1, t_2, \dots, t_n \in T$ , avem  $P(t_1, t_2, \dots, t_n) \in At$ .
- (iii) *Nimic altceva nu mai este formulă atomică.*

În cele de mai sus,  $T$  denotă mulțimea **termilor (funcționali)**, care este la rândul ei definită constructiv astfel:

**Baza.**  $X \subseteq T$  și  $F_o \subseteq T$  (variabilele și constantele sunt termi).

**Pas constructiv.** Pentru fiecare  $n \in \mathbf{N}^*$ , pentru fiecare  $f \in F_n$ , pentru fiecare  $t_1, t_2, \dots, t_n \in T$ , avem  $f(t_1, t_2, \dots, t_n) \in T$ .

Concluzionăm această etapă a definiției prin a „spune” că  $At \subseteq LP1$  (formulele atomice sunt formule).

**Pas constructiv.** Continuăm definirea lui  $LP1_{A/f}$  cu partea „formule vechi din formule noi”.

- (i) Dacă  $F \in LP1$  atunci  $(\neg F) \in LP1$ .
- (ii) Dacă  $F_1, F_2 \in LP1$  atunci  $(F_1 \wedge F_2), (F_1 \vee F_2) \in LP1$  (dacă dorim, putem introduce și  $(F_1 \rightarrow F_2), (F_1 \leftrightarrow F_2) \in LP1$ , etc.).

(iii) Dacă  $F \in \mathbf{LP1}$  atunci  $(\forall x)(F) \in \mathbf{LP1}$  (dacă dorim, punem și  $(\exists x)(F) \in \mathbf{LP1}$ ), pentru fiecare  $x \in X$ . ■

Ca și în cazul  $\mathbf{LP}$ , se definesc constructiv  $\text{subf}(F)$ , mulțimea subformulelor formulei  $F$ , și  $\text{Arb}(F)$ , arborele atașat lui  $F$  (cuantorii sunt operatori de aritate 1). Singura subformulă a unei formule atomice este ea însăși și  $\text{Arb}(F)$  va fi constituit în acest caz dintr-un simplu nod. Un term poate fi, la rândul său, privit ca un arbore (ca de altfel și orice formulă atomică), astfel încât arborele unei formule poate fi „detaliat”, dacă înlocuim fiecare nod corespunzător unui term cu arborele atașat acestuia (similar pentru o subformulă atomică). În definiția precedentă considerăm ca am pus implicit și (desigur că urmează *Nimic altceva nu mai este formulă*);

(iv) Dacă  $F \in \mathbf{LP1}$  atunci  $(F) \in \mathbf{LP1}$ .

**Definiția 3.2 (arborele atașat unui term și unei formule atomice).**

Conform definiției anterioare, arborele atașat unui term  $t \in T$ , notat  $\text{Arb}(t)$ , poate fi dat constructiv prin:

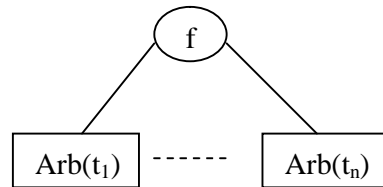
**Baza.** Dacă  $t = c \in F_0$ , atunci  $\text{Arb}(t)$  este:



Dacă  $t = x \in X$ , atunci  $\text{Arb}(t)$  este:



**Pas constructiv.** Fie  $n \in \mathbf{N}^*$ ,  $f \in F_n$ ,  $t_1, t_2, \dots, t_n \in T$ , astfel încât  $t = f(t_1, t_2, \dots, t_n)$ . Să presupunem că știm arborii atașați termilor  $t_1, t_2, \dots, t_n$ , adică  $\text{Arb}(t_1), \text{Arb}(t_2), \dots, \text{Arb}(t_n)$ . Atunci  $\text{Arb}(t)$  va fi:

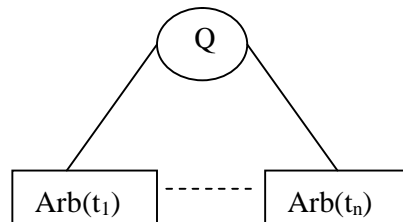


Știind  $\text{Arb}(t)$  pentru fiecare  $t \in T$ , putem acum construi  $\text{Arb}(F)$  pentru fiecare  $F \in \mathcal{A}t$ , după cum urmează.

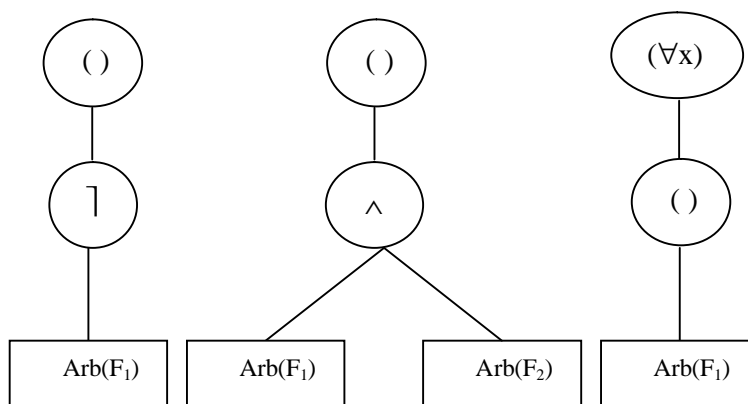
(i) Dacă  $P \in P_0$ , atunci  $\text{Arb}(P)$  este:



(ii) Fie  $n \in \mathbb{N}^*$ ,  $Q \in P_n$ ,  $t_1, t_2, \dots, t_n \in T$  astfel încât  $P = Q(t_1, t_2, \dots, t_n)$  și să presupunem că sunt cunoscuți  $\text{Arb}(t_1), \text{Arb}(t_2), \dots, \text{Arb}(t_n)$ . Atunci  $\text{Arb}(P)$  va fi:



În sfârșit, fie  $F$  o formulă oarecare. Ea poate avea una din formele:  $(\neg F_1)$ ,  $(F_1 \wedge F_2)$ ,  $(\text{Arb}((F_1 \vee F_2)))$  este similar cu cel pentru  $\wedge$  ( $(\forall x)(F_1)$  (similar,  $(\exists x)(F_1)$ ), sau  $(F_1) (\text{Arb}(F_1))$  coincide cu  $\text{Arb}((\neg F_1))$ , lipsind doar nodul etichetat cu  $\neg$ ), unde  $F_1, F_2$  sunt tot formule (oarecare). Arborele atașat lui  $F$  va avea în consecință una dintre formele:



■

**Observație.** Și arborele pentru implică, echivalent, etc. este similar cu cel pentru și. După cum se poate vedea, parantezele folosite la construcția termenilor și a formulelor atomice nu apar explicit, la fel cum parantezele pentru cuantificatori sunt considerate ca făcând parte din numele acestora. Ideea este că dacă se cunosc aritățile simbolurilor funcționale și predicative și dacă impunem în plus ca toate mulțimile de simboluri distincte care apar în  $A/f$  să fie disjuncte, atunci putem scrie  $Pt_1t_2...t_n$  în loc de  $P(t_1, t_2, ..., t_n)$ , adică putem elimina de tot aceste paranteze (care oricum trebuiau să aibă alt font decât cele din **P**). Și în cazul cuantificatorilor, în loc de  $(\forall x)(F)$  putem scrie  $\forall_x(F)$ . Astfel, putem conveni că singurele paranteze care „merită” a fi luate în considerare atunci când apar într-o formulă sunt cele din **P**. ■


În cazul lui  $\text{subf}(F)$ , poate este bine să subliniem faptul că  $\text{subf}((\forall x)(F_1)) = \{(\forall x)(F)\} \cup \{(F)\} \cup \text{subf}(F_1)$ , în rest nefiind diferențe esențiale față de cazul **LP**.

**Definiția 3.3 (apariții libere și legate ale variabilelor).** Fie  $F \in \mathbf{LP1}$  și  $x \in X$ , astfel încât  $x$  apare în  $F$ , la o poziție oarecare  $j$  (în sens textual, stânga/dreapta, ca *literă într-un cuvânt*, apariția menționată *nefiind parte a numelui unui cuantificator*  $(\forall x)$  sau  $(\exists x)$ ). Apariția fixată a lui  $x$  se numește **legată** dacă este într-o parte (subformulă)  $G$  a unei (alte) subformule a lui  $F$  de forma  $G_1 = (\forall x)(G)$  (sau  $(\exists x)(G)$ ). În restul cazurilor, apariția considerată se numește **liberă**. ■

Să mai punctăm o dată faptul că folosind reprezentarea formulelor ca arbori (precum și o definiție corespunzătoare a noțiunii de arbore), orice apariție a unei variabile într-o formulă poate fi definită formal într-un mod simplu (conform exemplelor și exercițiilor). Vom nota, pentru fiecare  $F \in \mathbf{LP1}$ , cu  $\text{free}(F)$  - mulțimea variabilelor care au apariții libere în  $F$ , și cu  $\text{leg}(F)$  - mulțimea variabilelor care au apariții legate în  $F$ . Desigur că pentru fiecare  $x \in X$ , este posibil ca  $x$  să nu apară în  $F$ , să aibă doar apariții libere, doar apariții legate, sau și apariții libere și apariții legate. Putem nota cu  $\text{var}(F) = \text{free}(F) \cup \text{leg}(F)$ . O situație nenaturală din punct de vedere semantic, dar posibilă sintactic, este aceea în care o variabilă  $x$  nu apare de loc în  $F$  (în sensul considerat), dar este prezent ca nume al unui cuantificator. Vom



conveni să notăm mulțimea acestor variabile cu  $restvar(F)$  și să includem și această mulțime în  $var(F)$ .


 **Exercițiul 3.1.** Definiți constructiv  $leg(F)$  și  $free(F)$ .

**Definiția 3.4 (închideri).** O formulă  $F \in \mathbf{LP1}$  se numește **închisă** dacă nu conține apariții libere de variabile (altfel spus,  $free(F) = \emptyset$ ). Pentru formula  $F$ , se numește **închiderea sa universală** formula  $(\forall x_1)((\forall x_2)(\dots((\forall x_k)(F))\dots)$  (notată, pentru simplitate și cu  $(\forall_*)(F)$  sau chiar  $(\forall F)$ ), unde  $\{x_1, x_2, \dots, x_k\} = free(F)$ . Analog (înlocuind  $\forall$  cu  $\exists$ ) se definește (notează) **închiderea existențială** a lui  $F$ . Se va numi **matricea** lui  $F$  (notată  $F^*$ ) acea formulă obținută din  $F$  prin ștergerea (sintactică, textuală) a tuturor cuantificatorilor  $(\forall x)$  și  $(\exists x)$ . O formulă care nu este închisă, se numește **deschisă** (o formulă în care  $var(F) = \emptyset$  se consideră a fi închisă). ■

**Definiția 3.5 (substituții).** Prin **substituție** vom înțelege o secvență finită de elemente de tipul  $[x/t]$  (numite și **substituții elementare**), unde  $x \in \mathcal{X}$ ,  $t \in \mathcal{T}$ . ■

O substituție va avea astfel forma  $\mathbf{s} = [x_1/t_1] \bullet [x_2/t_2] \bullet \dots \bullet [x_n/t_n]$  fiind practic un cuvânt peste alfabetul  $\mathbf{S} = \{[x/t] \mid x \in \mathcal{X}, t \in \mathcal{T}\}$ , adică un element al lui  $\mathbf{S}^*$  (monoidul liber generat de  $\mathbf{S}$ ). **O substituție  $\mathbf{s}$  (ca mai sus) se aplică unei formule  $\mathbf{F}$ , rezultând o formulă  $\mathbf{G}$ , notată  $(\mathbf{F})\mathbf{s}$ , care se obține din  $\mathbf{F}$  prin înlocuirea fiecărei apariții libere a variabilei  $x_1$  cu termenul  $t_1$ , apoi a fiecărei apariții libere a variabilei  $x_2$  cu  $t_2$ , etc.**

De obicei, se utilizează doar **substituții permise** pentru o formulă  $F \in \mathbf{LP1}$  dată. *Substituția elementară  $[x/t]$  este permisă pentru  $F$  (sau,  $F$  **acceptă**  $[x/t]$ ) dacă  $t$  nu conține variabile libere care au apariții legate în  $F$*  (s de mai sus va fi permisă pentru  $F$  dacă va fi permisă pentru fiecare componentă a sa,  $[x_i/t_i]$ ,  $i \in [n]$ ). Ordinea (fixată deja prin modul de scriere) aplicării substituțiilor elementare dintr-o substituție  $S$  este esențială în majoritatea cazurilor. O substituție  $S$  este **normalizată (pentru  $F$ )** dacă *ordinea de aplicare a substituțiilor elementare componente nu contează*. Mai precis,  $S$  este normalizată dacă avem  $(F)S = (F)S'$ , pentru fiecare  $S'$  care este obținută din  $S$  printr-o *permutare* a componentelor acesteia. **Substituția vidă** (ca element neutru al lui  $S^*$ ), notată  $[]$ , nu face desigur nici o transformare în formula  $F$  căreia îi este aplicată, adică avem  $(F)[] = F$ .

 **Exercițiul 3.2.** *Arătați că pentru fiecare  $F \in \mathbf{LP1}$  și fiecare substituție  $S$ , permisă pentru  $F$ , care îndeplinește condiția că ea nu conține un element de forma  $[x/t]$  astfel încât  $t$  nu conține  $x$ , există măcar o substituție  $S'$ , **echivalentă** cu  $S$  ( $S'$  este echivalentă cu  $S$ , pentru  $F$ , dacă  $(F)S = (F)S'$ ) și care este normalizată.*

După cum am mai precizat, dacă nu este nevoie de o redefinire explicită, în continuare vom folosi și alte concepte/notații sintactice introduse în **Capitolul 2** pentru **LP**, cum ar fi: *literal* (o formulă atomică sau negația acesteia), *clauză*, *clauză Horn*, *formă normală*, etc. Ca o convenție, cu litere latine mici de la sfârșitul alfabetului ( $x, y$ ,

z, ...) vom nota variabilele. Literele latine mici de la începutul alfabetului (a, b, c, ...) vor nota constantele. Literele latine mari de la mijlocul alfabetului (P, Q, R, ...) vor nota simbolurile predicative. Literele latine mici de la mijlocul alfabetului (f, g, h, ...) vor fi rezervate pentru desemnarea simbolurilor funcționale, iar cu F, G, H, ... vom nota formulele. Dacă nu am adopta astfel de convenții (dorim, totuși, să avem o anumită libertate de exprimare și să folosim și alte nume înafara celor admise de alfabetul inițial, așa cum am folosit și în cazul **LP**), am avea, posibil, cazuri în care, la o primă vedere, *nu am putea distinge între o variabilă și o constantă*. Sintactic, diferența dintre un nume de variabilă (element din  $X$ ) și un nume de constantă (element al lui  $F_0$ ) este clară doar în momentul când numele respectiv apare într-un cuantificator.

**Definiția 3.6.** Un term care nu conține variabile se numește **term de bază**. Analog, vom avea **formule de bază**, **substituții de bază**, etc. ■

Ca și pentru **LP**, putem *renunța la* anumite paranteze (nu uităm, sunt implicate doar elementele lui **P**) într-o formulă, bazându-ne pe aceeași idee (priorități acordate operatorilor, proprietăți generale ale acestora cum ar fi asociativitatea, convenții, etc). În cazul unui cuantor, avem nevoie, înafara priorității care este maximă (adică 0; standard, vom admite că  $\neg$  are prioritatea 1,  $\wedge$  - 2,  $\vee$  - 3,  $\rightarrow$  - 4,  $\leftrightarrow$  - 5, etc.; în cadrul unei secvențe continue de operatori de aceeași prioritate, parantetizarea se face *la dreapta*), și de definirea a ceea ce se numește

**domeniul sintactic** al cuantificatorului respectiv. *Domeniul sintactic pentru o apariție a unui cuantor (să spunem că aceasta este ("x)) într-o formulă F, reprezintă porțiunea continuă de text (care este o subformulă G alui F), care începe cu primul simbol de după o apariție a cuantorelui în cauză, identificat mai jos prin subliniere (parcurgem formula de la stânga la dreapta) și se termină cu un anumit simbol ulterior (să-l identificăm tot prin subliniere), conform următoarelor situații:*

- $F = \dots ("x)(G')$  ... . Nu sunt probleme, deoarece parantezele cerute de sintaxa dată de **Definiția 3.1.** nu au fost eliminate. Ca observație, parantezele subliniate de mai sus sunt (se mai numesc și) *corespondente* (definiția sintactică a unei formule nu permite existența unei paranteze deschise fără ca ea să fie închisă ulterior, neexistând ambiguități nici în privința parantezei închise care corespunde unei anumite paranteze deschise).
- $F = \dots ("x)G$ . Situația evidențiază faptul că *domeniul se extinde „până la sfârșitul formulei”*, dar condiția suplimentară este aceea că *în G nu există nici o paranteză închisă care să nu aibă corespondent o paranteză deschisă situată tot în domeniu* (nu uităm că domeniul începe oricum cu primul simbol de după ( $\forall x$ )). Nu am mai subliniat simbolurile de început și sfârșit a domeniului, deoarece ei sunt primul și respectiv ultimul simbol din G.

- $F = \dots ( \dots ( " x ) \dots ) \dots$  . Este pus în evidență cazul în care au fost eliminate parantezele inițiale de după cuantificator, dar în subformula  $G$  care urmează după acesta există o paranteză închisă care are corespondent *înainte* de apariția cuantorului al cărui domeniu îl căutăm. Ceea ce am subliniat este **prima** paranteză cu această proprietate (practic, simbolul imediat de dinaintea ei fiind ultimul care aparține domeniului).

Prioritatea cuantificatorilor fiind maximă, se găsesc mai întâi domeniile sintactice ale fiecărui cuantificator dintr-o formulă și apoi se aplică regulile de prioritate legate de conectori (acestea fiind similare de fapt cu cele din **LP**). Intuitiv, dacă privim **LP1** ca un *limbaj de programare imperativ (formulele fiind programe)*, construcțiile de tipul  $(\circ_1 x_1)(\circ_2 x_2) \dots (\circ_n x_n)$ , cu  $\circ_1, \circ_2, \dots, \circ_n \in \{\forall, \exists\}$ , reprezintă *linia de definiție a unei proceduri* (în care se specifică drept parametri *variabilele locale*). Procedurile n-au însă și un sfârșit marcat explicit, astfel încât *domeniul sintactic este corpul procedurii, delimitarea lui clară fiind desigur absolut necesară pentru a cunoaște porțiunea de text în care numele unei variabile păstrează o aceeași semnificație*. Continuând paralela, *variabilele libere din domeniul sintactic al unui cuantificator reprezintă variabilele globale*, iar *aplicarea unei substituții ar fi un apel de procedură* (deosebirea față de un program imperativ fiind aceea că valorile parametrilor, adică numele cuantificatorilor din linia de definiție, pot fi elemente *oarecare* sau *anumite* din domeniile considerate, conform *semanticii*). Să mai precizăm și faptul ca domeniile sintactice ale diverșilor cuantificatori pot fi, datorită formei sintactice a formulelor, doar **disjuncte**, adică de

tipul ... 1 ... 1 ... 2 ... 2 ... , sau **imbricate** - ... 1... 2... 2... 1... , dar **nu se pot „intersecta”**, ca în situația ... 1... 2... 1... 2... . Dacă o apariție a unei variabile  $x$  se află în interiorul mai multor domenii imbricate, desigur că ea aparține tuturor. Aparițiile diferite ale unui simbol pot avea semnificații (valori) diferite. Astfel, în ... 1... 2... 3 ...  $x_{(1)}$  ... 3 ...  $x_{(2)}$  ... 2... 1... , apariția lui  $x$  notată  $x_{(1)}$  și cea notată  $x_{(2)}$ , pot practic fi considerate ca fiind nume de variabile diferite. În ceea ce privește *parantezele adăugate*, ele pot fi fie *restaurări* ale parantezelor șterse, fie ne folosim de (iv) (**Definiția 3.1, Pas inductiv** extins).

**Exemplu.** Fie formula:

$$F = Q(x) \vee (\exists x)(\forall y)(P(f(x), z) \wedge Q(a) \vee (\forall x)R(x, z, g(x))).$$

**Semnificația simbolurilor care apar:**

- $x, y, z \in X$ . Avem  $x, y \in X$  deoarece sunt (și) nume ale unor cuantificatori, iar  $z \in X$  conform convențiilor făcute (e literă de la sfârșitul alfabetului). Variabila  $x$  apare de 4 ori în  $F$  (de la stânga la dreapta:  $x$  are o apariție liberă în  $Q(x)$ ; o apariție legată în  $P(f(x), z)$ , ea fiind în domeniul sintactic al cuantorului  $(\exists x)$ ; o altă apariție legată, pe prima poziție din  $R(x, z, g(x))$ , aici apariția fiind și în domeniul sintactic al cuantorului  $(\forall x)$ ; o a treia apariție legată, în același domeniu sintactic, este în  $R(x, z, g(x))$ , și anume în  $g(x)$ ). Variabila  $y$  *nu apare nici o dată în  $F$ , deși apare cuantorul  $(\forall y)$*  (ea apare însă în  $\text{restvar}(F)$ ). Variabila  $z$  apare de 2 ori în  $F$ , ambele apariții fiind libere (prima dată în  $P(f(x), z)$  și a doua oară în  $R(x, z, g(x))$ ). În

consecință,  $\text{free}(F) = \{x, z\}$ ,  $\text{leg}(F) = \{x\}$ ,  $\text{var}(F) = \{x, z\}$  (sau chiar  $\{x, y, z\}$  dacă acceptăm ca  $\text{restvar}(F)$  să fie introdus în  $\text{var}(F)$ ).

- $a \in F_0$  conform convențiilor.
- $Q \in P_1, P \in P_2, R \in P_3$ , din scrierea sintactică (nici convențiile nu sunt contrazise).
- $f, g \in F_1$ , din scrierea sintactică (la fel, convențiile „spun” același lucru).

**Domeniile sintactice ale cuantorilor.** Domeniul sintactic al cuantorului  $(\exists x)$  este format din (subformula)  $(\forall y)(P(f(x), z) \wedge Q(a) \vee (\forall x)R(x, z, g(x)))$ . Apoi, cel al lui  $(\forall y)$ , din  $(P(f(x), z) \wedge Q(a) \vee (\forall x)R(x, z, g(x)))$ . În sfârșit, domeniul lui  $(\forall x)$  este  $R(x, z, g(x))$ . Ținând cont și de prioritățile impuse cuantorilor, *formula corectă, total parantetizată* (conform **Definiției 3.1**), este de fapt:

$$F = (Q(x) \vee (\exists x)((\forall y)((P(f(x), z) \wedge Q(a)) \vee (\forall x)(R(x, z, g(x)))))$$

Orice amănunte legate de sintaxă ar trebui studiate numai pe această formă, practic noi neavând definiții formale pentru cazul unor formule din care lipsesc anumite paranteze.

**Mulțimea subformulelor formulei date.**

$\text{subf}(F) = \{Q(x), P(f(x), z), Q(a), R(x, z, g(x)), (\forall x)(R(x, z, g(x))), (R(x, z, g(x))), (P(f(x), z) \wedge Q(a)), ((P(f(x), z) \wedge Q(a)) \vee (\forall x)(R(x, z, g(x)))), (\forall y)((P(f(x), z) \wedge Q(a)) \vee (\forall x)(R(x, z, g(x)))))), (((P(f(x), z) \wedge Q(a)) \vee (\forall x)(R(x, z, g(x)))))), (\exists x)((\forall y)((P(f(x), z) \wedge Q(a)) \vee (\forall x)(R(x, z, g(x)))))), ((\forall y)((P(f(x), z) \wedge Q(a)) \vee (\forall x)(R(x, z, g(x)))))), F\}$ . Dintre acestea, unele sunt **formule atomice**:

$Q(x)$ ,  $P(f(x), z)$ ,  $Q(a)$  și  $R(x, z, g(x))$ . În plus,  $Q(a)$  este o **formulă de bază**. Să reamintim din nou că avem o definiție formală pentru  $\text{subf}(F)$  doar în cazul în care n-am eliminat paranteze din  $F$  și că *subformulă a lui  $F$  este orice subcuvânt al său care aparține lui LP1*.

**Termii (funcționali) care apar în formulă.** Ei sunt în ordine:  $x$ ,  $f(x)$ ,  $z$ ,  $a$ ,  $g(x)$  (de fapt, termii  $x$  și  $z$  au câte două apariții, ei fiind și variabile).

**Mulțimea subformulelor închise ale lui  $F$ .** Nu avem asemenea subformule. Totuși, prin convenția stabilită, am admis că formulele care nu conțin deloc variabile (adică formulele de bază) sunt formule închise. Singura asemenea subformulă în cazul nostru este deci  $Q(a)$ .

**Matricea formulei date.** Ea este (am eliminat și parantezele corespondente devenite inutile în urma ștergerii cuantificatorilor):

$$F^* = (Q(x) \vee ((P(f(x), z) \wedge Q(a)) \vee R(x, z, g(x)))).$$

**Închiderea universală a formulei** este:

$$F' = (\forall z)((\forall x)(Q(x) \vee (\exists x)((\forall y)((P(f(x), z) \wedge Q(a)) \vee (\forall x)(R(x, z, g(x))))))).$$

Similar se obține **închiderea existențială**.

Fie acum **substituția**  $S = [x/c] \bullet [z/g(b)] \bullet [y/x]$ . Dacă o aplicăm formulei  $F$  (inițiale, nu celei parantetizate complet) găsim succesiv:

$$\begin{aligned} (F)S &= \\ (Q(c) \vee (\exists x)(\forall y)(P(f(x), z) \wedge Q(a) \vee (\forall x)R(x, z, g(x))))[z/g(b)] \bullet [y/z] &= \\ = (Q(c) \vee (\exists x)(\forall y)(P(f(x), g(b)) \wedge Q(a) \vee (\forall x)R(x, g(b), g(x))))[y/x]. \end{aligned}$$

Ultima aplicare nu schimbă nimic deoarece  $y$  nu apare în  $F$ . Conform definiției rezultă că  $S$  **nu este permisă** pentru  $F$  (datorită faptului că în

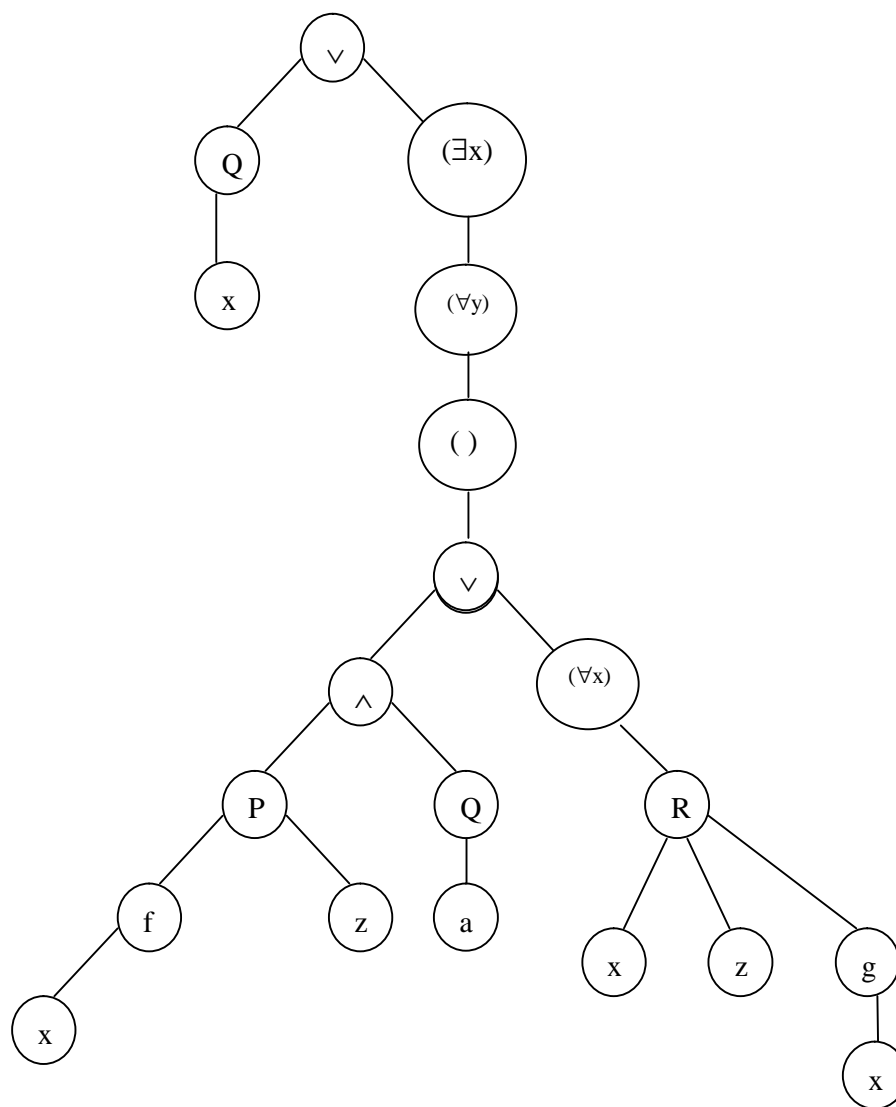


substituția elementară  $[y/x]$ , termul  $t = x$  conține o variabilă care apare legată în  $F$ ). Cum însă  $y$  este doar numele unui cuantificator, aplicarea lui  $s$  formulei  $F$  nu este dăunătoare în acest caz (o substituție nepermisă generează de obicei grave anomalii semantice). Tot din acest motiv,  $s$  este și **normalizată**.

**Arborele atașat formulei**

$$Q(x) \vee (\exists x)(\forall y)(P(f(x), z) \wedge Q(a) \vee (\forall x)R(x, z, g(x))),$$

este (din nou, trebuia de fapt să considerăm formula corectă, total parantetizată; oricum, aceasta implică doar adăugarea unor noduri interne, etichetate cu „()”):



■

Putem acum introduce și studia elementele de semantică privitoare la calculul cu predicate de ordinul I.

## §2. Semantica logicii cu predicate de ordinul I

**Înțelesul (semantica)** unei formule  $F \in \mathbf{LP1}$  va fi, la fel ca în logica propozițională, o valoare de adevăr  $0, 1 \in \mathbf{B}$ , valoare obținută într-un mod extensional. Elementul principal în definirea semanticii va rămâne noțiunea de *structură*. Deși definiția unei structuri și găsirea unei valori de adevăr pentru o formulă va depinde practic doar de simbolurile care intervin în acea formulă, vom prefera să utilizăm în continuare să ne folosim de funcțiile totale în locul celor parțiale.

**Definiția 3.7.** Se numește **structură** un cuplu  $S = \langle U_S, I_S \rangle$  în care  $U_S$  este o mulțime **nevidă** numită **univers**, iar  $I_S$  este o funcție (numită și **interpretare**)

$$I_S : X \cup P \cup F \rightarrow U_S \cup [U_S^* \rightarrow \mathbf{B}] \cup [U_S^* \rightarrow U_S],$$

care satisface condițiile:

- Dacă  $x \in X$ , atunci  $I_S(x) \in U_S$ .
- Dacă  $P \in P_n$ , atunci  $I_S(P) : U_S^n \rightarrow \mathbf{B}$ .
- Dacă  $F \in F_n$ , atunci  $I_S(F) : U_S^n \rightarrow U_S$ . ■

În cele de mai sus, ca și în restul materialului,  $[A \rightarrow B]$  desemnează mulțimea tuturor funcțiilor totale având domeniul  $A$  și codomeniul  $B$ , iar  $[A^* \rightarrow B]$  denotă mulțimea tuturor funcțiilor de oricâte argumente (inclusiv 0) peste  $A$ , cu valori în  $B$ . Prin urmare, **interpretarea (semantica) unei variabile în structura  $S$  este un element din**

univers, interpretarea unui simbol predicativ  $n$ -ar este o funcție de la  $U_S^n$  la  $\{0, 1\}$  (sau, uneori, mulțimea elementelor din  $U_S^n$  pentru care valoarea în cauză este 1), iar semantica unui simbol funcțional de aritate  $n$  este o funcție de la  $U_S^n$  la  $U_S$ . Pentru simplificarea exprimării, vom renunța la indici dacă nu există confuzii și vom nota pe  $I_S$  tot cu  $S$ . Similar cu cazul logicii propoziționale, orice structură va putea fi unic extinsă astfel încât să fie definită pentru toate elementele lui **LP1**.

**Definiția 3.8.** Pentru fiecare structură  $S = \langle U_S, I_S \rangle$ , vom numi **extensia sa imediată funcția**

$$S': X \cup P \cup F \cup T \cup \mathbf{LP1} \rightarrow U_S \cup [U_S^* \rightarrow \mathbf{B}] \cup [U_S^* \rightarrow U_S] \cup \mathbf{B},$$

dată constructiv în continuare. Pentru început, vom pune  $S'(a) = S(a) (= I_S(a))$ , pentru fiecare  $a \in X \cup P \cup F$ , ceea ce înseamnă că  $S'$  s-a definit, în particular, pentru fiecare term elementar. Fie acum orice  $t \in T$ , adică orice  $n \in \mathbf{N}^*$ , orice  $t_1, t_2, \dots, t_n \in T$  și orice  $f \in F_n$ , astfel încât  $t = f(t_1, t_2, \dots, t_n)$ . Atunci  $S'(t) = S(f)(S'(t_1), S'(t_2), \dots, S'(t_n)) (\in U_S)$ . Am încheiat astfel procesul de definire al lui  $S'$  pe  $X \cup P \cup F \cup T$  și rămâne să definim  $S'$  pe **LP1**. Vom face acest lucru în mod constructiv.

**Baza.** Fie  $F = A \in At$ . În această situație avem fie  $A = P \in P_0$  fie  $A = P(t_1, t_2, \dots, t_n)$ ,  $n \in \mathbf{N}^*$ ,  $t_1, t_2, \dots, t_n \in T$ . În primul caz  $S'$  este

deja definită ( $S'(P) = S(P) \in \mathbf{B}$ ), iar în al doilea caz punem desigur  $S'(P) = S(P)(S'(t_1), S'(t_2), \dots, S'(t_n)) \in \mathbf{B}$ .

**Pas constructiv.** Vom avea de considerat cazurile:

- (i)  $F = (\bigwedge F_1)$ . Atunci  $S'(F) = \overline{S'(F_1)}$ .
- (ii)  $F = (F_1 \wedge F_2)$ . Atunci  $S'(F) = S'(F_1) \cdot S'(F_2)$ .
- (iii)  $F = (F_1 \vee F_2)$ . Atunci  $S'(F) = S'(F_1) + S'(F_2)$ .
- (iv)  $F = (\forall x)(G)$ . Atunci  $S'(F) = 1$  dacă și numai dacă pentru fiecare  $u \in U_S$  avem  $S'_{[x/u]}(G) = 1$  unde  $S'_{[x/u]}$  este o interpretare care coincide în totalitate cu  $S'$  exceptând faptul că  $S'(x) = u$ .
- (v)  $F = (\exists x)(G)$ . Atunci  $S'(F) = 1$  dacă și numai dacă există (măcar) un element  $u \in U_S$  astfel încât  $S'_{[x/u]}(G) = 1$ . ■

Vom pune, evident, și  $S'(F) = S'((F))$ , pentru fiecare  $F \in \mathbf{LP1}$ . De asemenea, de acum înainte nu vom face nici o diferență între  $I_S$ ,  $S$ ,  $S'$ , notând valoarea de adevăr a unei formule  $F \in \mathbf{LP1}$  într-o structură dată  $S$  prin  $S(F)$  sau chiar  $F^S$  (de fapt, tehnic vorbind, am putea face acest lucru de-abia după demonstrația **Teoremei 3.1**). În mod cu totul similar vor fi notate interpretările celorlalte simboluri în structura dată:  $x^S$ ,  $c^S$ ,  $f^S$ ,  $P^S$ ,  $t^S$ , etc. Putem vorbi acum de noțiuni pe care le cunoaștem deja, cum ar fi *model*, *formule satisfiabile*, *valide*, *nesatisfiabile*, *consecință semantică*, *echivalență tare și slabă*, etc. Așa cum am mai precizat, dată o formulă  $F$  și o structură  $S$ , avem nevoie doar de valorile lui  $S$  pe simbolurile care apar în  $F$ , adică, la fel ca în cazul logicii

propoziționale, vom folosi titulatura de *structură corectă pentru o formulă* (sau o mulțime de formule), pentru a denota restricția unei structuri la simbolurile din  $Alf$  care apar într-o formulă (acesta fiind de fapt o funcție parțială pe  $X \cup P \cup F \cup T \cup \mathbf{LP1}$ ). Pot fi identificate și anumite (clase de) structuri speciale atât la nivel sintactic cât și la nivel semantic. *Sintactic*, să nu uităm că mulțimea de formule din calculul cu predicate de ordinul I depinde de alfabetul inițial  $Alf$ . Vom presupune astfel că mulțimea de variabile este numărabilă, că în mulțimea  $P$  cel puțin  $P_0, P_1, P_2$  sunt nevide, că mulțimea  $F$  conține măcar un simbol funcțional (indiferent de aritate), etc. Fără restricții de acest tip, există pericolul ca  $\mathbf{LP1}$  să fie *trivială* (de exemplu, mulțimea vidă), *banală* (poate coincide cu  $\mathbf{LP}$  sau chiar cu o subclasă a acesteia) sau *prea particulară* (putându-se exprima, de exemplu, doar relațiile unare dintre diverse obiecte). *Semantic*, putem admite existența unor **simboluri speciale de predicate sau funcții care să fie interpretate identic în orice structură**. Un exemplu este *relația de egalitate*, care poate fi considerată ca fiind reprezentată la nivel sintactic de un simbol predicativ de aritate 2 (să presupunem că  $P_2 \neq \emptyset$  și „=”  $\in P_2$ ). În momentul în care considerăm o formulă  $F$  care conține un asemenea simbol și ne interesează existența unui model, *este normal să-l căutăm doar printre structuri de forma  $S = \langle U_S, I_S \rangle$  în care  $=^S(a, b)$  este egal cu 1 atunci și numai atunci când  $a$  coincide cu  $b$  în universul dat ( $=^S : U_S \times U_S \rightarrow \mathbf{B}$ )*. Cu alte cuvinte, în asemenea situații, clasa structurilor/modelelor „admise” este restrânsă în mod forțat, ceea ce nu are întotdeauna efecte secundare favorabile. Astfel, în situația descrisă,

clasa de formule este notată  $\mathbf{LP1}_{\{=\}}$  (pe scurt,  $\mathbf{LP1}_{=}$ ) și se numește **calculul cu predicate de ordinul I cu egalitate** (punctăm iar: dacă  $F \in \mathbf{LP1}_{=}$ , atunci, dacă  $F$  conține simbolul predicativ  $= \in P_2$ , pentru a i se calcula valoarea de adevăr  $S(F)$  se iau în considerare doar structuri  $S$  în care „=” este interpretat *doar* ca fiind egalitatea pe universul respectiv). Acestea se vor numi **structuri/interpretări standard**. Similar se poate considera  $\mathbf{LP1}_{\{+, \cdot\}}$ ,  $\mathbf{LP1}_{\{=, +\}}$ , etc. Vom vedea că *problema satisfiabilității* pentru  $\mathbf{LP1}$  (notată **SAT1**) este *semidecidabilă*, iar *problema satisfiabilității* pentru  $\mathbf{LP1}_{=}$  este *nedecidabilă*. Un rol important în demonstrații îl au *structurile Herbrand*.

**Definiția 3.9 (universuri și structuri Herbrand).** Fie  $F \in \mathbf{LP1}$ . Se numește **univers Herbrand (atașat lui F)**, mulțimea  $D(F)$  definită prin:

**Baza.** În  $D(F)$  se pun toate elementele din  $F_0$  care apar în  $F$ . Dacă  $F$  nu conține nici o constantă, atunci se pune **forțat** în  $D(F)$  un element oarecare din  $F_0$  (numele rezervat standard, de obicei, este  $a$ ).

**Pas constructiv.** Fie orice  $n \in \mathbb{N}^*$ , orice  $f \in F_n$  care apare în  $F$  și termii oarecare  $t_1, t_2, \dots, t_n \in D(F)$ . Atunci  $f(t_1, t_2, \dots, t_n) \in D(F)$ .

O **structură Herbrand (pentru F)** este o structură (corectă pentru F)  $H = \langle U_H, I_H \rangle$ , în care  $U_H = D(F)$ , iar  $I_H$  satisface condiția că *interpretează fiecare term prin el însuși*. Mai exact,  $H(f(t_1, t_2, \dots, t_n)) =$

$f^H(t_1^H, t_2^H, \mathbf{K}, t_n^H) = f(t_1^H, t_2^H, \mathbf{K}, t_n^H)$ , pentru fiecare  $n \in \mathbf{N}$  și fiecare  $t = f(t_1, t_2, \dots, t_n) \in \mathcal{T}$ . ■

Se poate spune că  $D(F)$  este mulțimea termenilor de bază construiți cu simboluri din  $F$ . Într-o structură Herbrand, dacă  $f \in F_0$  atunci  $H(f) = f$  și în consecință dacă  $t$  este un term de bază avem și  $t^H = t$ . Interpretarea unei variabile este cea uzuală ( $x^H \in D(F)$  pentru fiecare  $x \in \mathcal{X}$ ), la fel ca și interpretarea simbolurilor predicative (ele vor fi funcții oarecare peste  $D(F)$  cu valori în  $\mathbf{B}$ ). **A nu se confunda  $f^H(t_1^H, t_2^H, \mathbf{K}, t_n^H)$ , care denotă aplicarea efectivă a funcției  $f^H : D(F)^n \rightarrow D(F)$  n-uplului  $\langle t_1^H, t_2^H, \mathbf{K}, t_n^H \rangle$ , cu  $f(t_1^H, t_2^H, \mathbf{K}, t_n^H)$  (valoarea aplicării anterioare), care este un term fără variabile aparținând lui  $D(F)$ , adică, în ultimă instanță, un șir de caractere.** Dacă există o structură Herbrand care este model pentru o formulă  $F$ , atunci spunem și că  **$F$  admite model Herbrand.**

**Exemplu.** Fie formula  $F = (\forall x)(P(x, f(x)) \wedge Q(g(b, z)))$ .

(i) **Să se găsească o structură  $S = \langle U_S, I_S \rangle$ , corectă pentru  $F$ , astfel încât  $S \models F$ .** Conform **Definiției 3.8**, vom avea  $S(F) = 1$  dacă și numai dacă pentru fiecare  $u \in U_S$  avem  $S_{[x/u]}(P(x, f(x)) \wedge Q(g(b, z))) = 1$ , adică dacă și numai dacă pentru fiecare  $u \in U_S$  avem  $S_{[x/u]}(P(x, f(x))) = 1$  și  $S_{[x/u]}(Q(g(b, z))) = 1$ , adică dacă și numai dacă pentru fiecare  $u \in U_S$  avem  $P^{S_{[x/u]}}(x^{S_{[x/u]}}(x^{S_{[x/u]}}), f^{S_{[x/u]}}(x^{S_{[x/u]}})) = 1$  și  $Q^{S_{[x/u]}}(g^{S_{[x/u]}}(b^{S_{[x/u]}}, z^{S_{[x/u]}})) = 1$ ,



adică dacă și numai dacă pentru fiecare  $u \in U_S$  avem  $P^S(u, f^S(u)) = 1$  și  $Q^S(g^S(b^S, z^S)) = 1$ . În acest moment, pentru obținerea structurii cerute, se pune mai întâi problema alegerii unui univers  $U_S$  corespunzător. Ar trebui să ne gândim măcar la  $D(F)$  (a fost recent construit) și la  $\mathbf{N}$ , care este o mulțime cunoscută, având numeroase proprietăți. Cum ambele mulțimi au același cardinal, vom încerca întâi cu  $U_S = \mathbf{N}$ . Apoi, într-un prim pas ar trebui să găsim o relație  $P^S : \mathbf{N} \times \mathbf{N} \rightarrow \mathbf{N}$  și o funcție  $f^S : \mathbf{N} \rightarrow \mathbf{N}$ , astfel încât pentru fiecare  $u \in U_S$  să avem  $P^S(u, f^S(u)) = 1$ . O relație binară pe  $\mathbf{N}$ , cunoscută, este *relația de ordine parțială*  $\leq$ . Astfel, dacă alegem  $P^S @ \leq$ , putem lua  $f^S$  ca fiind *funcția succesor* ( $f^S(n) = n + 1$ , pentru fiecare  $n \in \mathbf{N}$ ) și atunci avem  $P^S(u, f^S(u)) = 1$ , pentru fiecare  $u \in U_S$ . Structura deja sugerată mai trebuie completată de alegerea corespunzătoare a relației  $Q^S : \mathbf{N} \rightarrow \mathbf{N}$ , a funcției  $g^S : \mathbf{N} \times \mathbf{N} \rightarrow \mathbf{N}$  și a valorilor  $b^S \in \mathbf{N}$  și  $z^S \in \mathbf{N}$ , astfel încât să avem și  $Q^S(g^S(b^S, z^S)) = 1$ . Cum toate acestea sunt elemente precizate, există numeroase opțiuni convenabile. De exemplu, putem lua  $b^S = 5$ ,  $z^S = 10$ ,  $g^S(m, n) = m + n$  pentru fiecare  $m, n \in \mathbf{N}$  ( $g^S$  este adunarea pe  $\mathbf{N}$ ) și, în sfârșit,  $Q^S(n) = 1$  dacă și numai dacă  $n$  este număr impar. Trecerea la o structură Herbrand similară este ușoară, dacă ținem cont de faptul că există o corespondență bijectivă între  $\mathbf{N}$  (sau o submulțime finită a sa) și  $D(F)$ , oricare ar fi formula  $F$ .

(ii) Să se găsească o structură  $S = \langle U_S, I_S \rangle$ , corectă pentru  $F$ , astfel încât  $S \models F$ . Procedăm identic cu cazul precedent, doar în final luăm de exemplu  $Q^S(n) = 1$  dacă și numai dacă  $n$  este număr par.

(iii) Să se găsească universul Herbrand al lui  $F$ . Conform Definiției 3.9, începem (Baza) prin a introduce în  $D(F)$  constantele care apar în  $F$ . Prin urmare, inițial (conform convențiilor deja adoptate),  $D(F) = \{b\}$ . Rămâne să aplicăm **Pasul constructiv** cât timp este posibil, „obținând”  $D(F) = \{b, f(b), f(f(b)), \dots\} \cup \{g(b, b), g(g(b, b), b), \dots\} \cup \{f(g(b, b)), f(f(g(b, b))), \dots\} \cup \{g(f(b), b), \dots\} \cup \dots$ . Desigur că  **$D(F)$  este numărabilă**. Într-adevăr, considerând  $D(F)$  ca o mulțime de cuvinte peste alfabetul  $\mathcal{A} = \{b, g, f, (, ), ,\}$ , putem defini *lungimea*,  $l(w)$ , unui cuvânt  $w \in \mathcal{A}^*$  ca fiind egală cu numărul de caractere care îl compun (poate chiar exceptând parantezele). Atunci, pentru fiecare  $n \in \mathbf{N}$ , mulțimea cuvintelor de lungime  $n$  va fi finită și astfel  $D(F)$  va fi o reuniune numărabilă de mulțimi finite, deci numărabilă, adică aflată în corespondență bijectivă cu  $\mathbf{N}$ . Mai mult,  $D(F)$  va fi chiar **recursiv enumerabilă** ([ȚIP]), adică există un semialgoritm care, fără a avea nici o intrare, **listează** toate elementele din  $D(F)$  (de exemplu, în ordinea crescătoare a lungimii acestora). În acest mod, putem presupune că  $\mathcal{A}^*$  este total ordonată printr-o relație pe care o vom nota cu  $\prec$  și definită formal prin:  $w_1 \prec w_2$  dacă și numai dacă  $w_1$  este înaintea lui  $w_2$  în lista furnizată ca ieșire de semialgoritm anterior (putem chiar să menținem în această listă o singură apariție, de exemplu prima întâlnită în parcurgerea ei de la stânga la dreapta, a oricărui

cuvânt). În consecință, mulțimea  $D(F) \cap \mathcal{A}^*$  este și ea *total ordonată prin restricția relației  $\mathcal{D}$* . Vom prescurta acest lucru scriind  $D(F) = \{w_1, w_2, \dots\}$ , știind și că  $w_1 \mathcal{D} w_2 \mathcal{D} \dots$ . Să mai notăm faptul că  $D(F)$  este întotdeauna o mulțime *nevidă*, ea putând fi *finită în cazul în care  $F$  nu conține simboluri funcționale de aritate diferită de 0* (în restul cazurilor  $D(F)$  are aspectul de mai sus).


(iv) **Să se găsească o structură Herbrand  $H1 = \langle U_{H1}, I_{H1} \rangle$  care să fie model pentru  $F$  și o alta  $H2 = \langle U_{H2}, I_{H2} \rangle$  care să nu fie model pentru  $F$ .** Ținând cont de punctele anterioare și de **Definiția 3.9** (structurile Herbrand pentru o formulă  $F$  sunt întotdeauna corecte pentru  $F$ ), avem  $D(F) = \{w_1, w_2, \dots\}$  și putem lua  $P^{H1} = P^{H2} = \angle$ ,  $f^{H1}(w_i) = f^{H2}(w_i) = w_{i+1}$  ( $i \in \mathbb{N}$ ),  $b^{H1} = b^{H2} = b \in D(F)$  (acest lucru este obligatoriu conform definiției unei structuri Herbrand),  $z^{H1} = z^{H2} = b$  (de exemplu) și  $g^{H1}(u, v) = g^{H2}(u, v) = uv$  (aceasta însemnând concatenarea cuvintelor  $u$  și  $v$ ). În sfârșit, vom pune  $Q^{H1}(u) = 1$  dacă și numai dacă  $l(u)$  este număr par și respectiv  $Q^{H2}(u) = 1$  dacă și numai dacă  $l(u)$  este număr impar. Rezultă imediat că  $H1 \models F$  și  $H2 \not\models F$ . ■

**Teorema 3.1 (de extensie).** Pentru fiecare structură  $S = \langle U_S, I_S \rangle$ , extensia sa imediată  $S'$  dată prin **Definiția 3.8** este funcție și este unica funcție având domeniul  $X \cup P \cup F \cup T \cup \mathbf{LP1}$  și codomeniul  $U_S \cup [U_S^* \rightarrow \mathbf{B}] \cup [U_S^* \rightarrow U_S] \cup \mathbf{B}$ , care extinde  $S$  și satisface condițiile:

- (i)  $S'(\neg F_1) = \overline{S'(F_1)}$ , pentru fiecare  $F \in \mathbf{LP1}$ .
- (ii)  $S'((F_1 \wedge F_2)) = S'(F_1) \bullet S'(F_2)$ , pentru fiecare  $F_1, F_2 \in \mathbf{LP1}$ .
- (iii)  $S'((F_1 \vee F_2)) = S'(F_1) + S'(F_2)$ , pentru fiecare  $F_1, F_2 \in \mathbf{LP1}$ .
- (iv)  $S'((\forall x)(G)) = 1$  dacă și numai dacă pentru fiecare  $u \in U_S$  avem  $S'_{[x/u]}(G) = 1$  unde  $S'_{[x/u]}$  este o interpretare care coincide cu  $S'$  exceptând faptul că  $S'(x) = u$  (pentru fiecare  $x \in X$  și fiecare  $G \in \mathbf{LP1}$ ).  
Mai sus putem adăuga și
- (v)  $S'((\exists x)(G)) = 1$  dacă și numai dacă există (măcar) un element  $u \in U_S$  astfel încât  $S'_{[x/u]}(G) = 1$  (pentru fiecare  $x \in X$  și fiecare  $G \in \mathbf{LP1}$ ).

Relația  $S'((F)) = S'(F)$  o putem considera ca fiind adevărată chiar prin convenție.

**Demonstrație.** Faptul că  $S'(a) = S(a)$  pentru fiecare  $a \in X \cup P \cup F$  ( $S'$  extinde  $S$ ) este adevărat din definiție. Pentru a arăta faptul că  $S'$  este funcție și este unică având proprietățile considerate, procedăm prin inducție structurală (analog cu demonstrația corespunzătoare de la logica propozițională). Proprietatea de a fi funcție a lui  $S'$  se păstrează succesiv pentru  $T$  și  $\mathbf{LP1}$  deoarece  $S$  este funcție, unicitatea rezultând prin reducere la absurd, folosind direct relațiile (i) – (v). ■

 **Exercițiul 3.3.** Fie  $F$  o formulă din calculul cu predicate de ordinul  $I$ , cu sau fără egalitate,  $(\neg)^*F$  închiderea sa universală și  $(S^*)F$

*închiderea sa existențială. Ce se poate spune despre legătura dintre aceste formule în privința satisfiabilității (validității)?*

În secțiunile/paragrafele următoare din **Capitolul 3** vom urma un drum oarecum similar cu calea parcursă în cazul **LP**, pentru a ajunge în situația să testăm satisfiabilitatea unei formule din **LP1** utilizând rezoluția (adaptată pentru noul context). Avem nevoie de câteva rezultate preliminare.

### §3. Forme normale în LP1

Să observăm mai întâi că **LP1** poate fi considerată ca o *extensie reală* a lui **LP**, atât *la nivel sintactic* cât și *la nivel semantic*. Orice formulă de bază din **LP1** (formulă care nu conține variabile) poate fi interpretată ca o formulă propozițională (inclusiv elementele lui  $P_0$ ). Dacă formula în cauză nu conține nici apariții de conectori (și, evident, nici de cuantori), ea va putea fi considerată drept o variabilă propozițională. Conchidem că *din punct de vedere sintactic* avem într-adevăr  $\mathbf{LP} \subseteq \mathbf{LP1}$  (însă este iar nevoie de niște presupuneri suplimentare privind cardinalitatea unor mulțimi ca  $P_0, F_0, F_1$  sau  $P_1$ ). Mai mult, considerând orice structură  $S$  ca funcție semantică în sensul **LP1**, ea poate fi considerată ca o extensie a funcției corespunzătoare aplicată pentru formulele care sunt (și) elemente ale lui **LP** (iar această ultimă funcție, după cum se constată direct din definiții, reprezintă exact o structură în sensul **LP**). Vom vedea că relația dintre **LP** și **LP1** este chiar mai profundă, **LP** putând fi considerată ca o *aproximare* alui

**LP1**, la fel cum mulțimea numerelor raționale este o aproximare a mulțimii numerelor reale. Numele folosite, *logică propozițională* respectiv *logică cu predicate de ordinul I*, sunt legate de lipsa cuantificatorilor respectiv de folosirea acestora pentru a „lega” doar variabilele (nu și, de exemplu, simbolurile funcționale sau cele predicative). Cuvântul *calcul*, utilizat ca un sinonim pentru *logică* în acest context, exprimă faptul că mulțimile (de formule) considerate nu sunt „amorfe”, ci pot fi „prelucrate” într-un mod „sistematic” (vorbind despre *calculul diferențial și integral*, de exemplu). O logică (calcul) în care ar fi permisă și cuantificarea simbolurilor predicative (eventual și a celor funcționale), ar putea fi numită **logică (calcul) cu predicate de ordinul II (LP2)**, dacă ținem cont de faptul observat deja că (măcar) elementele lui  $P_0$  sunt formule atomice pentru **LP** dar „variabile” în **LP1** (analog, formulele atomice din **LP1** ar putea fi „variabile” în **LP2**). Fiecare nou tip de logică (**LP1**, **LP2**, ...), trebuie să fie în același timp o extensie și o aproximare a logicii anterioare (pe scurt, o **generalizare** sau o **logică de ordin superior**). După cum am mai subliniat de câteva ori, deși rezultatele privind *complexitatea* sau *tratabilitatea* (*algoritmică* a) unor probleme importante sunt descurajatoare chiar pentru logici de ordin inferior, căutarea unor logici „mai generale” (de ordin superior, dar și neclasice) este justificată prin *mărirea puterii de exprimare*.

**Exemplu.** Găsiți o formulă  $F \in \mathbf{LP1}_=$ , care:

- Este satisfiabilă.

- Conține un simbol funcțional binar  $f \in F_2$  și un simbol constant  $e \in F_0$ .
- Pentru fiecare  $S \subseteq F$  avem:  $\langle U_S, f^S \rangle$  este grup.

Am fi putut formula cele de mai sus astfel: *Să se găsească o formulă din calculul cu predicate de ordinul 1 cu egalitate care să exprime faptul că o anumită relație binară (numită  $f$ ) peste o mulțime nevidă oarecare, determină pe aceasta o structură de grup (al cărui element neutru este notat  $e$ ).* În acest caz, ar fi existat întrebări inevitabile, cum ar fi „Cum se poate defini formal faptul că o formulă exprimă o anumită cerință?” (acest lucru nu este posibil, deoarece măcar unul dintre elementele implicate nu poate avea o definiție formală). Formularea adoptată mai sus are avantajul că este precisă, deoarece toți termenii implicați au definiții formale (formulă din **LP1**<sub>=</sub>, formulă satisfiabilă, simbol funcțional binar, structură, grup, etc.). Pentru rezolvare va trebui să luăm în considerare și diferitele clase de axiome (diferite) prin care se identifică un grup. Deoarece formula căutată  $F$  este din **LP1**<sub>=</sub> și elementul neutru este precizat explicit, putem folosi axiomele „standard” (vom reveni la acest tip de problemă și în **Capitolul 4**):

1.  $f$  este lege de compoziție internă, binară, pe orice mulțime nevidă considerată:

$$F_1: (\forall x)(\forall y)(\exists z)(f(x, y) = z) \wedge (\forall x)(\forall y)(\forall u)(\forall v)(f(x, y) = u \wedge f(x, y) = v \rightarrow u = v).$$

2.  $f$  este asociativă:

$$F_2: (\forall x)(\forall y)(\forall z)(f(x, f(y, z)) = f(f(x, y), z)).$$

3. *f* admite element neutru (notat  $e$ ):


$$F_3: (\exists e)(\forall x)(f(x, e) = f(e, x) \wedge f(x, e) = x)).$$


4. Fiecare element  $x$  este simetrizabil în raport cu  $f$ :

$$F_4: (\forall x)(\exists x')(f(x, x') = f(x', x) \wedge f(x, x') = e)).$$

Cerințele sunt astfel imediat satisfăcute dacă luăm

$$F = F_1 \wedge F_2 \wedge F_3 \wedge F_4. \blacksquare$$

 **Exercițiul 3.4.** Considerând exemplul anterior, puteți găsi o formulă  $F$  care să satisfacă aceleași cerințe, exceptând folosirea explicită a lui  $e$ ? Dar dacă am interzice și folosirea explicită a simbolului de egalitate?

 **Exercițiul 3.5.** Să se găsească o formulă care să conțină un simbol predicativ binar  $P$  și care să exprime faptul că  $P$  este o relație (binară) antisimetrică.

**Teorema 3.2 (de substituție).** Fie  $H \in \mathbf{LP1}$ , oarecare. Fie orice  $F, G \in \mathbf{LP1}$  astfel încât  $F$  este o subformulă a lui  $H$  și  $G$  este tare echivalentă cu  $F$ . Fie  $H'$  formula obținută din  $H$  prin înlocuirea (unei apariții fixate a) lui  $F$  cu  $G$ . atunci  $H \equiv H'$ .

**Demonstrație.** Se procedează prin inducție structurală, într-un mod identic cu cazul  $\mathbf{LP}$ . Situația în care sunt utilizați cuantificatorii trebuie tratat suplimentar. Reamintim astfel că afirmația de demonstrat este  $(\forall H \in \mathbf{LP1})P(H)$ , unde

$$P(H): (\forall F, G, H' \in \mathbf{LP1})((F \in \text{subf}(H)) \text{ și}$$

$(H'$  se obține din  $H$  înlocuind o apariție fixată a lui  $F$  cu  $G$ ) și



$$(F \equiv G) \Rightarrow H \equiv H'.$$

În **Pasul inductiv** singurul caz netratat anterior, în **Teorema 2.5** (pentru  $H = (\exists x)H_1$  se procedează similar) este:

(iv) **H** = (" **x**)**H**<sub>1</sub>. Fie orice  $F \in \text{subf}(H) = \{H\} \cup \text{subf}(H_1)$ . Dacă  $F = H$ , nu mai avem ce demonstra, la fel ca în cazul **Baza**. Să presupunem că  $F \in \text{subf}(H_1)$  și fie  $H_1'$  formula care se obține din  $H_1$  înlocuind o apariție a lui  $F$  cu un  $G$  (nu uităm că  $F \equiv G$ ). Știind, conform ipotezei inductive, că  $H_1 \equiv H_1'$ , faptul că  $H' = (\forall x)H_1'$  este tare echivalentă cu  $H$  rezultă acum direct din definiții. ■

Echivalențele deja cunoscute (**Teorema 2.4**) pot fi completate cu altele, care se referă la cuantori.

**Teorema 3.3.** Pentru fiecare  $F, G \in \mathbf{LP1}$  și fiecare  $x, y \in X$ , sunt adevărate următoarele echivalențe:

1. 
$$\begin{aligned} \top (\forall x)F &\equiv (\exists x)(\top F) \\ \top (\exists x)F &\equiv (\forall x)(\top F) \end{aligned}$$
2. Dacă  $x$  nu apare liber în  $G$ , atunci:
 
$$\begin{aligned} (\forall x)(F \wedge G) &\equiv (\forall x)(F) \wedge G \\ (\forall x)(F \vee G) &\equiv (\forall x)(F) \vee G \\ (\exists x)(F \wedge G) &\equiv (\exists x)(F) \wedge G \\ (\exists x)(F \vee G) &\equiv (\exists x)(F) \vee G \end{aligned}$$
3. 
$$\begin{aligned} (\forall x)(F) \wedge (\forall x)(G) &\equiv (\forall x)(F \wedge G) \\ (\exists x)(F) \vee (\exists x)(G) &\equiv (\exists x)(F \vee G) \end{aligned}$$
4. 
$$(\forall x)(\forall y)F \equiv (\forall y)(\forall x)F$$

$$(\exists x)(\exists y)F \equiv (\exists y)(\exists x)F$$

5. Dacă  $x$  nu apare liber în  $F$ , atunci:

$$(\forall x)F \equiv F$$

$$(\exists x)F \equiv F.$$

**Demonstrație.** Începem cu prima echivalență din 5. Fie astfel orice structură (corectă)  $S = \langle U_S, I_S \rangle$  astfel încât  $S \models (\forall x)F$ . Aceasta înseamnă că  $S((\forall x)F) = 1$ , adică pentru fiecare  $u \in U_S$  avem  $S_{[x/u]}(F) = 1$ . Dar  $S_{[x/u]}$  coincide cu  $S$ , exceptând faptul că  $S_{[x/u]}(x) = u$  și aceasta pentru fiecare apariție (liberă) a lui  $x$  în  $F$ . Prin urmare, pentru că  $x$  nu are apariții libere în  $F$ ,  $S_{[x/u]}(F) = S(F)$  (**q. e. d.**). A doua echivalență din 5. rezultă similar, iar 4. reprezintă o consecință imediată din definiții. Mai demonstrăm doar prima echivalență din 2., restul lăsându-le pe seama cititorului. Fie din nou orice structură  $S = \langle U_S, I_S \rangle$  corectă pentru formulele care intervin. Avem succesiv:

$$S((\forall x)(F) \wedge G) = 1 \text{ dacă și numai dacă}$$

pentru fiecare  $u \in U_S$ ,  $S_{[x/u]}(F)$  și  $S(G) = 1$  dacă și numai dacă (folosim 5., pentru  $G$ )

pentru fiecare  $u \in U_S$ ,  $S_{[x/u]}(F) = 1$  și  $S_{[x/u]}(G) = 1$  dacă și numai dacă

pentru fiecare  $u \in U_S$ ,  $S_{[x/u]}(F \wedge G) = 1$  dacă și numai dacă

$$S((\forall x)(F \wedge G)) = 1. \blacksquare$$

**Observație.** Ca o consecință a teoremei anterioare rezultă că sunt adevărate și echivalențele  $(\forall x)(\forall x)F \equiv (\forall x)F$  și  $(\exists x)(\exists x)F \equiv (\exists x)F$  (chiar și variantele  $(\exists x)(\forall x)F \equiv (\forall x)F$ , respectiv  $(\forall x)(\exists x)F \equiv (\exists x)F$ ),

care se pot generaliza pentru oricâte apariții ale cuantorilor (F ar putea conține la rândul ei cuantificatori). Se poate arăta însă că **nu sunt adevărate** echivalențele (mai exact, există  $x \in X$ , există  $F, G \in \mathbf{LP1}$  astfel încât, mai jos, formulele din membrul stâng nu sunt echivalente cu cele corespunzătoare din membrul drept):

$$(\forall x)F \vee (\forall x)G \equiv (\forall x)(F \vee G)$$

$$(\exists x)F \wedge (\exists x)G \equiv (\exists x)(F \wedge G)$$

Să considerăm de exemplu prima relație. Chiar la nivel intuitiv cei doi membri nu exprimă același lucru deoarece faptul că F sau G este satisfăcută pentru o (aceeași) valoare a lui x nu înseamnă că fie F, fie G, sunt satisfăcute pentru (alte) valori ale lui x, eventual diferite. Astfel, putem lua drept contraexemplu:

- $F = P(x)$ .
- $G = Q(x)$ .
- $S = \langle U_S, I_S \rangle$ ,  $U_S = \mathbf{N}$ ,  $P^S(n) = 1$  dacă și numai dacă n este număr par,  $Q^S(n) = 1$  dacă și numai dacă n este număr impar (interpretarea lui x în S nu contează, toate aparițiile sale fiind în acest caz legate).

Nici echivalența  $(\forall x)(\exists y)F \equiv (\exists y)(\forall x)F$  **nu este adevărată** pentru fiecare formulă F. ■

Putem trage și concluzia că scrierea succesivă a unui aceluiași cuantor este inutilă, la fel fiind și o nouă cuantificare a unei apariții deja cuantificate (interpretarea unei variabile care nu are apariții libere într-o formulă neavând efect în stabilirea valorii de adevăr a acelei formule,

indiferent de structura considerată). Deoarece **echivalențele anterioare pot fi interpretate și ca modalități de transformare a formulelor care păstrează validitatea** („mutând cuantificatorii în față”), putem considera că am făcut un prim pas spre obținerea *formelor normale* pentru formulele din **LP1**. Deși nu este imediat util, am preferat să începem cu rezultatul următor datorită importanței sale în justificarea unor afirmații și caracterului didactic al demonstrației. *Intuitiv, teorema afirmă că pentru a găsi semantica unei formule  $F$  (cu ajutorul unei structuri  $S$ ), formulă în care am substituit aparițiile libere ale unei variabile  $x$  cu un term de bază  $t$ , putem afla mai întâi semantica lui  $t$  în structura considerată și apoi să găsim semantica lui  $F$  în noua structură  $S_{[x/S(t)]}$ .*

**Teorema 3.4 (lema de translație).** Fie  $F \in \mathbf{LP1}$ ,  $x \in X$ ,  $t \in T$  un term de bază și  $S$  orice structură corectă pentru formulele care apar. Atunci:

$$S((F)[x/t]) = S_{[x/S(t)]}(F)$$

**Demonstrație.** Arătăm prin inducție structurală adevărul metaformulei  $(\forall F)P(F)$ , unde:

$P(F)$ : (pentru orice  $x \in X$ )(pentru orice  $t \in T$ , term de bază)

(pentru orice structură  $S$ )( $S((F)[x/t]) = S_{[x/S(t)]}(F)$ ).

**Baza.**  $F \in At$ .

(i)  $F = P \in P_0$ . Neexistând variabile, afirmația este imediată.

(ii) Fie  $n \in \mathbf{N}^*$ ,  $t_1, t_2, \dots, t_n \in T$ ,  $P \in P_n$  astfel încât  $F = P(t_1, t_2, \dots, t_n)$ .

Dacă  $x$  nu apare (liber) în  $F$ , afirmația este din nou evidentă. În caz contrar, este suficient să demonstrăm rezultatul pentru cazul în care există o (singură) apariție (liberă) a lui  $x$  în  $F$ . Putem presupune, fără a restrânge generalitatea, că  $x$  apare la poziția  $i$  (de la stânga la dreapta) în  $t_1$ . În acest caz, este clar că  $t_1$  are forma  $f(\mathbf{K}, x_i, \mathbf{K})$ , sau poate chiar

coincide cu  $x$ . Atunci:

$$\begin{aligned} S((F)[x/t]) &= S((P(t_1, t_2, \dots, t_n))[x/t]) = \\ &= S((P((t_1)[x/t], t_2, \dots, t_n))) = \\ &= S((P(f(\dots, x, \dots)))[x/t], t_2, \dots, t_n) = \\ &= S((P(f(\dots, t, \dots))), t_2, \dots, t_n) = \\ &= S(P^S(f^S(\dots, t^S, \dots)), S(t_2), \dots, S(t_n)) = \\ &= S_{[x/S(t)]}(F), \text{ direct din definiții, în contextul teoremei.} \end{aligned}$$

**Pas inductiv.**

$$(i) \mathbf{F} = (\neg \mathbf{G}). \text{ Atunci } S((F)[x/t]) = S((\neg \mathbf{G})[x/t]) = S(\neg((G)[x/t])) = S((G)[x/t]).$$

Aplicând acum ipoteza inductivă pentru  $G$ , obținem în continuare:

$$S((G)[x/t]) = S_{[x/S(t)]}(G) = S_{[x/S(t)]}(\neg G) = S_{[x/S(t)]}(F).$$

Să mai observăm că egalitățile anterioare sunt triviale dacă  $x$  nu are nici o apariție liberă în  $F$ .

(ii)  $\mathbf{F} = (\mathbf{G} \circ \mathbf{H})$ ,  $\circ \in \{\wedge, \vee\}$ . Se demonstrează analog cu cazul precedent, folosindu-se ipoteza inductivă pentru  $G$  și  $H$ , regulile semantice corespunzătoare pentru  $\vee$  și  $\wedge$ , precum și faptul că  $t$  nu conține variabile (este term de bază).

(iii)  $\mathbf{F} = (\forall y)\mathbf{G}$  (cazul  $\mathbf{F} = (\exists y)\mathbf{G}$  poate fi demonstrat similar). Dacă  $y = x$  și  $x$  nu apare liber în  $G$ , suntem într-un caz similar cu cel tratat în **Baza**. Presupunem că  $y \neq x$  și  $x$  apare liber în  $G$  (deci și în  $F$ ) măcar odată. Ca urmare, rezultă  $S(\mathbf{F})[x/t] = S((\forall y)\mathbf{G})[x/t] = S((\forall y)((\mathbf{G})[x/t]))$ . Atunci:

$S((\forall y)((\mathbf{G})[x/t])) = 1$  dacă și numai dacă pentru fiecare  $u \in U_S$  avem  $S_{[y/u]}((\mathbf{G})[x/t]) = 1$  dacă și numai dacă (aplicăm ipoteza inductivă pentru  $G$  și  $S_{[y/u]}$ )

$S_{[y/u][x/S(t)]}(\mathbf{G}) = 1$  dacă și numai dacă ( $t^S, u \in U_S$ , și  $y \neq x$ )

$S_{[x/S(t)][y/u]}(\mathbf{G}) = 1$  dacă și numai dacă

$S_{[x/S(t)]}((y)\mathbf{G}) = 1$  dacă și numai dacă

$S_{[x/S(t)]}(\mathbf{F}) = 1$ . ■

Cu următorul rezultat se „începe construcția” formelor normale.

**Teorema 3.5 (lema de redenumire a variabilelor legate).** Fie

$\mathbf{F} = (\circ x)\mathbf{G}$ ,  $\circ \in \{\forall, \exists\}$ , o formulă oarecare din **LP1**. Fie  $y$  o variabilă nouă (în sensul că ea nu apare în  $G$ ). Atunci :

$$\mathbf{F} \equiv (\circ y)(\mathbf{G}[x/y]).$$

**Demonstrație.** Imediată, prin inducție structurală. ■

În cele de mai sus era suficient să presupunem că  $y$ , variabila nouă, nu apare liber în  $G$ .

**Definiția 3.10 (forma normală rectificată).** O formulă  $F \in \mathbf{LP1}$  se numește *rectificată* (sau, se află în formă normală rectificată, pe scurt **FNR**) dacă nu conține variabile care apar atât libere cât și legate și nu are cuantificatori care să lege aceeași variabilă, dar pe poziții diferite în formulă (indiferent dacă este vorba de cuantificatori existențiali sau universali). ■

**Teorema 3.6.** Pentru orice formulă din  $F \in \mathbf{LP1}$ , există o formulă rectificată  $F' \in \mathbf{LP1}$ , astfel încât  $F' \equiv F$ .

**Demonstrație.** Se aplică de un număr finit de ori **Lema de redenumire** formulei inițiale  $F$  (implicit, se aplică și **Teorema 3.3** și **Teorema de substituție**). ■

**Definiția 3.11 (forma normală prenex).** O formulă  $F \in \mathbf{LP1}$  este în *formă normală prenex* (**FNP**, pe scurt) dacă  $F = (\circ_1 y_1) \dots (\circ_n y_n)G$ , unde  $n \in \mathbf{N}$ ,  $\circ_i \in \{\exists, \forall\}$  ( $i \in [n]$ ), iar  $y_1, \dots, y_n$  sunt toate variabilele distincte care apar (liber) în  $G$ . În plus,  $G$  nu mai conține cuantificatori. ■

În cele de mai sus, cazul  $n = 0$  se referă la absența cuantificatorilor ( $[0] = \emptyset$ ).

**Teorema 3.7.** Pentru fiecare formulă  $F \in \mathbf{LP1}$ , există o formulă  $F' \in \mathbf{LP1}$ , care este în **FNP** și este tare echivalentă cu  $F$ .

**Demonstrație.** Fără a restrânge generalitatea, putem presupune că toate formulele care intervin sunt rectificate. Procedăm prin inducție structurală, (meta)afirmația de demonstrat fiind evidentă.

**Baza.**  $F \in \mathcal{A}t$ . Aceasta este deja în **FNP**, având numărul de cuantificatori egal cu zero.

**Pas inductiv.**

(i)  $F = (\bar{u} F_1)$ . Presupunem că  $F_1$  este în **FNP** și arătăm că  $F$  este în **FNP**. Conform ipotezei inductive pentru  $F_1$ , există  $G_1$  aflată în **FNP** rectificată astfel încât  $G_1 \equiv F_1$ . Aceasta are aspectul  $G_1 = (\circ_1 y_1) \dots (\circ_n y_n) G$ , unde  $\circ_1, \circ_2, \dots, \circ_n \in \{\exists, \forall\}$  și  $G$  nu mai conține cuantificatori. Atunci:

$F = (\bar{u} F_1) \equiv \bar{u}(\circ_1 y_1) \dots (\circ_n y_n) G \equiv (d'_1 y_1) \dots (d'_n y_n) (\bar{u} G)$ , unde  $d'_i = \exists$  dacă  $\circ_i = \forall$  și reciproc, pentru fiecare  $i \in [n]$ . Ultima formulă este în **FNP**.

(ii)  $F = (F_1 \circ F_2)$ ,  $\circ \in \{\bar{u}, \bar{u}'\}$ . Conform ipotezei inductive, rezultă că există formulele

$$G_1 = (\circ_1 x_1) \dots (\circ_m x_m) G'_1, \circ_i \in \{\forall, \exists\}, i \in [m] \text{ și}$$

$$G_2 = (d'_1 y_1) \dots (d'_k y_k) G'_2, d'_i \in \{\forall, \exists\}, i \in [k], G_1 \text{ și } G_2 \text{ fiind în}$$

**FNPR** (formă normală prenex rectificată) și echivalente respectiv cu  $F_1$  și  $F_2$ . Atunci  $F \equiv G_1 \circ G_2 = (\circ_1 y_1) \dots (\circ_m y_m) G'_1 \circ$

$$(d'_1 z_1) \dots (d'_k z_k) G'_2 \equiv (d'_1 z_1) \dots (d'_k z_k) (\circ_1 y_1) \dots (\circ_m y_m) (G'_1 \circ G'_2).$$

În cele de mai sus, am aplicat de un număr finit de ori **Lema de redenumire**, **Teorema 3.3** și **Teorema de substituție**.



(iii)  $F = (\circ x)G$ ,  $\circ \in \{ ", \$ \}$ . Conform ipotezei inductive, există o formulă tare echivalentă cu  $G$ , aflată în **FNPR**,  $G = (\circ_1 y_1) \dots (\circ_n y_n)G'$ ,  $\circ_i \in \{ \forall, \exists \}$ ,  $i \in [n]$ . Dacă  $x \in \{ y_1, y_2, \dots, y_n \}$ , atunci aplicăm mai întâi **Lema de redenumire**. Să presupunem acum că  $x \notin \{ y_1, y_2, \dots, y_n \}$ . Atunci  $F \equiv (\circ x)(\circ_1 y_1) \dots (\circ_n y_n)G'$ , formulă care se află în forma necesară. ■

**Observație.** Am arătat că pentru fiecare formulă din **LP1**, există o altă formulă din **LP1**, care este tare echivalentă cu ea și care este în **FNP** rectificată (pe scurt, **FNPR**). Conform **Teoremei 3.3**, putem presupune și că nu există în această formulă cuantificatori care să lege variabile care nu apar în ea și nici cuantificatori (relativ la o aceeași variabilă) cu apariții multiple. Cu alte cuvinte, din punctul de vedere al satisfiabilității, ne putem limita la studiul formulelor din **LP1** de forma  $F = (\circ_1 y_1) \dots (\circ_k y_k)F'$ , unde  $\text{free}(F') = \{ y_1, y_2, \dots, y_k \}$ , iar  $F'$  este chiar matricea lui  $F$ , nemaiconținând alți cuantori ( $\circ_1, \circ_2, \dots, \circ_k \in \{ \forall, \exists \}$ ). Prin urmare, folosind și **Exercițiul 3.3** (o formulă este satisfiabilă dacă și numai dacă închiderea sa existențială este satisfiabilă), putem spune că pentru testarea satisfiabilității unei formule din **LP1**, putem să ne limităm la clasa de formule având aspectul sintactic  $F = (\circ_1 y_1)(\circ_2 y_2) \dots (\circ_k y_k)F^*$ , unde  $F^*$  este matricea lui  $F$  iar  $\text{leg}(F) = \text{var}(F) = \text{free}(F^*) = \{ y_1, y_2, \dots, y_k \}$ . Prin urmare, această formulă este și închisă, neconținând apariții libere de variabile. ■

Vom arăta în continuare că dacă există cuantificatori existențiali în forma anterioară pentru o formulă, aceștia pot fi eliminați, fără a se afecta satisfiabilitatea. Testul de satisfiabilitate pentru formulele din **LP1** poate fi redus astfel la un test de satisfiabilitate pentru formule de forma  $F = (\forall y_1) \dots (\forall y_n) F^*$ , unde  $\{y_1, y_2, \dots, y_k\} = \text{free}(F^*)$ ,  $F^*$  neconținând cuantificatori.

**Definiția 3.12 (forma normală Skolem).** O formulă  $F \in \mathbf{LP1}$  este în *formă normală Skolem* (**FNS**, pe scurt), dacă are aspectul  $F = (\forall x_1) \dots (\forall x_k) G$  unde  $G$  nu mai conține cuantificatori (este matricea lui  $F$ ), iar  $x_1, x_2, \dots, x_k$  sunt variabile distincte și reprezintă exact variabilele care apar în  $G$  ( $\text{free}(G) = \{x_1, x_2, \dots, x_k\}$ ).  $F$  este în **formă normală Skolem clauzală** (**FNSC**, pe scurt), dacă este în **FNS** și matricea sa este în **FNC** (forma normală conjunctivă) într-un sens similar cu **LP** (literalii reprezentând acum formule atomice din **LP1** sau negații ale lor). ■

**Teorema 3.8.** Pentru fiecare formulă  $F$  din **LP1**, există o altă formulă  $F' \in \mathbf{LP1}$ , care este în **FNSC** și care este slab echivalentă cu ea.

**Demonstrație.** Vom prezenta un algoritm prin care formula  $F'$  va fi construită efectiv din formula  $F$ .

### Algoritm Skolem

**Intrare:**  $F \in \mathbf{LP1}$ . Fără a restrânge generalitatea, putem presupune că  $F$  este în **FNPR**, închisă.

**Ieșire:**  $F' \in \mathbf{LP1}$ , aflată în **FNS** (și închisă), slab echivalentă cu  $F$ .

**Metodă:**

**Pasul 1.**  $F' := F$ .

**Pasul 2. Cât\_timp** (există cuantificatori existențiali în  $F'$ )

**execută**

**2.1.** Alege un asemenea cuantificator și elimină-l.

**2.2.** Transformă formula  $F'$ .

**Sf\_Cât\_timp.**

**Comentarii.** Orice formulă intermediară prelucrată de algoritm are forma  $F' = (\forall y_1) \dots (\forall y_n)(\exists z)G$ , unde  $G$  poate să conțină și alți cuantificatori (am pus în evidență primul cuantificator existențial, alegerea fiind acum deterministă dacă ne gândim că parcurgem formula simbol cu simbol, de la stânga la dreapta). Atunci, în urma **Pașilor 2.1** și **2.2**,  $F'$  va căpăta forma  $F' = (\forall y_1) \dots (\forall y_n)((G)[z/f(y_1, \dots, y_n)])$  unde  $f$  este un simbol funcțional nou (în sensul că el nu mai apare în formulele considerate),  $f \in F_n$ . Să notăm  $H @ (\forall y_1) \dots (\forall y_k)(\exists z)G$ , formula de tip  $F'$  existentă înainte de execuția unui pas al algoritmului precedent și cu  $H' = (\forall y_1) \dots (\forall y_k)(G)[z/f(y_1, y_2, \dots, y_k)]$  formula rezultată după execuție. ■

Arătăm acum că  $H$  este slab echivalentă cu  $H'$ .

**Presupunem întâi că  $H'$  este satisfiabilă.** Există atunci o structură corectă  $S = \langle U_S, I_S \rangle$ , astfel încât  $S \models H'$ . Ca urmare, pentru fiecare  $u_1, u_2, \dots, u_k \in U_S$  avem

$$S_{[y_1/u_1][y_2/u_2]\dots[y_k/u_k]}((G)[z/f(y_1, y_2, \dots, y_k)]) = 1.$$

Pentru că variabilele  $y_1, y_2, \dots, y_k$  sunt distincte și apar liber în  $(G)[z/f(y_1, y_2, \dots, y_k)]$  și pentru că structura  $S'$  (adică

$S_{[y_1/u_1][y_2/u_2]\dots[y_k/u_k]}$ ) coincide cu  $S$ , exceptând valorile lui  $y_1, y_2, \dots, y_k$  care sunt „puse” pe  $u_1, u_2, \dots, u_k$ , semantica **LP1** ne permite să înlocuim în formula  $G' = (G)[z/f(y_1, y_2, \dots, y_k)]$  pe  $y_1, y_2, \dots, y_k$  respectiv cu niște constante noi (nu apar nicăieri în formulele care intervin), distincte,  $a_1, a_2, \dots, a_k$  și să extindem structurile  $S$  și  $S'$  prin  $a_i^S = a_i^{S'} = u_i$ , pentru fiecare  $i \in [k]$ . În acest moment, interpretarea termenului  $t = f(y_1, y_2, \dots, y_k)$  în structurile considerate va coincide cu cea a lui  $f(a_1, a_2, \dots, a_k)$ . Putem astfel chiar înlocui pe  $t$  cu  $f(a_1, a_2, \dots, a_k)$ , astfel încât el nu va conține variabile. Avem deci  $S'(f(y_1, y_2, \dots, y_k)) = S'(f(a_1, a_2, \dots, a_k)) = f^{S'}(\langle S'(a_1), \dots, S'(a_k) \rangle) = f^S(\langle u_1, \dots, u_k \rangle)$ . Atunci  $S'((G)[z/t]) = 1$  și  $t$  nu conține variabile, astfel încât putem aplica **Lema de translație**, găsim  $S'_{[z/S'(t)]}(G) = 1$ , adică  $S'_{[z/S(t)]}(G) = 1$ . Sintetizând, putem spune că pentru fiecare  $u_1, \dots, u_k \in U_S = U_{S'}$ , există  $v \in U_{S'}$  ( $v = t^S = f^S(\langle u_1, \dots, u_k \rangle)$ ) astfel încât  $S'_{[z/v]}(G) = 1$ . Prin urmare,  $S$  este model (și) pentru  $H$ , adică  $H$  este satisfiabilă.

Invers, să **presupunem că H este satisfiabilă**. Fie orice  $S = \langle U_S, I_S \rangle$ , structură corectă pentru H (atenție,  $S$  nu este definită pentru simbolul  $f$ ), astfel încât  $S \models H$ . Conform definiției, pentru fiecare  $u_1, \dots, u_k \in U_S$  există măcar un  $v \in U_S$  astfel încât  $S_{[y_1/u_1][y_2/u_2]\dots[y_k/u_k][z/v]}(G) = 1$ . Extindem pe  $S$  la  $S'$ , corectă și pentru  $H'$ , punând (în rest,  $S'$  coincide cu  $S$ ):  $f^{S'}(\langle u_1, \dots, u_k \rangle) = v$ . Prin urmare, pentru fiecare  $u_1, \dots, u_k \in U_S = U_{S'}$ , avem  $S'_{[y_1/u_1][y_2/u_2]\dots[y_k/u_k][z/f(u_1, \dots, u_k)]}(G) = 1$ . Luăm acum din nou termenul  $t$ ,  $t = f(y_1, \dots, y_k)$  și folosind aceeași argumentație ca mai sus, putem aplica „invers” **Lema de translație**, obținând  $S'_{[y_1/u_1][y_2/u_2]\dots[y_k/u_k]}((G)[z/f(y_1, \dots, y_k)]) = 1$ , ceea ce înseamnă că  $S'$  satisface  $H'$ , adică  $H'$  este satisfiabilă.

Prin urmare, orice execuție a corpului buclei din **Algoritmul Skolem** elimină un cuantor existențial, transformările făcute neschimbând caracterul formulei (de a fi sau nu satisfiabilă). Deci *fiecare* formulă intermediară de tipul  $F'$  este în **FNS** și este slab echivalentă cu  $F$ . În sfârșit, pentru a aduce  $(F')^*$  la forma clauzală, putem aplica orice algoritm cunoscut pentru cazul **LP** (renotând literalii pozitivi din  $F'$  prin nume de variabile propoziționale distincte din  $A$ ). ■

Putem sintetiza rezultatele obținute până în prezent în:

**Teorema 3.9.** Pentru fiecare formulă  $F \in \mathbf{LP1}$ , există o formulă  $F' \in \mathbf{LP1}$  astfel încât  $F' \equiv_s F$ ,  $F'$  fiind în **FNSC** închisă ( $F' = (\forall y_1) \dots (\forall y_n) F^*$ ,  $\{y_1, \dots, y_n\} = \text{free}(F^*)$ ,  $F^*$ , matricea lui  $F$ , fiind în formă normală conjunctivă). ■

Exemplul următor ne va sugera și un algoritm pentru aflarea (unei) **FNSC** (de acum încolo vom presupune că orice formulă aflată în **FNSC** este și închisă).

**Exemplu.** Fie

$$F = (\forall x)(\exists y)(P(x, g(y), z)) \vee (\neg(\forall x)(Q(x)) \wedge (\forall z)(\exists x) \neg R(f(x, z), z)).$$

Să se găsească o formulă  $F'$ , aflată în **FNSC**, slab echivalentă cu  $F$ .

**Forma rectificată.** Folosim **Lema de redenumire**, în principal (folosirea **Teoremei de substituție** este implicită, iar echivalențele exprimate prin **Teorema 3.3** trebuie utilizate imediat ce este posibil de a fi aplicate; *ordinea de aplicare a acestora nu este esențială*). Găsim (prin redenumire  $z$ , redenumire  $x$  în două poziții diferite și aplicare **Teorema 3.3**, punctul 1.):

$$\begin{aligned} F &\equiv (\forall x)(\exists y)((P(x, g(y), z))) \vee ((\exists x)(\neg Q(x)) \wedge (\forall z)(\exists x) \neg R(f(x, z), z)) \equiv \\ &\equiv (\forall x)(\exists y)(P(x, g(y), z)) \vee ((\exists u)(\neg Q(u)) \wedge (\forall t)(\exists v) \neg R(f(v, t), t)). \end{aligned}$$

**Forma normală prenex rectificată.**  $\wedge$  este prioritar față de  $\vee$  (puteam deci să ne fi dispensat de niște paranteze) și  $u$  nu apare liber în  $(\forall t)(\exists v) \neg R(f(v, t), t)$ . De aceea,  $(\exists u)(\neg Q(u)) \wedge (\forall t)(\exists v) \neg R(f(v, t), t) \equiv (\exists u)(\neg Q(u) \wedge (\forall t)(\exists v) \neg R(f(v, t), t))$ . În continuare,  $t$  și  $v$  nu apar liber în  $\neg Q(u)$  și atunci  $(\exists u)(\neg Q(u)) \wedge (\forall t)(\exists v) \neg R(f(v, t), t) \equiv (\exists u)(\forall t)(\exists v)$

$(\bigwedge Q(u) \wedge \bigwedge R(f(v, t), t))$ . Revenind,  $x$  și  $y$  nu apar (liber) în  $(\exists u)(\forall t)(\exists v)(\bigwedge Q(u) \wedge \bigwedge R(f(v, t), t))$  și atunci  $F \equiv (\forall x)(\exists y)(P(x, g(y), z) \vee (\exists u)(\forall t)(\exists v)(\bigwedge Q(u) \wedge \bigwedge R(f(v, t), t)))$ . Acum  $u, t, v$  nu apar liber în  $P(x, g(y), z)$  și atunci:

$$F \equiv (\forall x)(\exists y)(\exists u)(\forall t)(\exists v)(P(x, g(y), z) \vee \bigwedge Q(u) \wedge \bigwedge R(f(v, t), t)).$$

**Forma normală Skolem.** Trebuie să eliminăm întâi cuantificatorii existențiali, nu înainte de a porni cu o **formulă închisă**. Singura variabilă cu apariții libere în ultima formulă este  $z$  și obținem imediat:

$$F \equiv_s (\exists z)(\forall x)(\exists y)(\exists u)(\forall t)(\exists v)(P(x, g(y), z) \vee \bigwedge Q(u) \wedge \bigwedge R(f(v, t), t)).$$

Aplicăm acum formulei din dreapta **Algoritmul Skolem**, adică efectuăm un număr de 4 execuții a corpului buclei, în scopul de a elimina cei 4 cuantificatori existențiali.

**I.**  $F \equiv_s (\forall x)(\exists y)(\exists u)(\forall t)(\exists v)(P(x, g(y), b) \vee \bigwedge Q(u) \wedge \bigwedge R(f(v, t), t))$ , unde  $b \in F_0$ .

**II.**  $F \equiv_s (\forall x)(\exists u)(\forall t)(\exists v)(P(x, g(h(x)), b) \vee \bigwedge Q(u) \wedge \bigwedge R(f(v, t), t))$ , unde  $h \in F_1$ .

**III.**  $F \equiv_s (\forall x)(\forall t)(\exists v)(P(x, g(h(x)), b) \vee \bigwedge Q(i(x)) \wedge \bigwedge R(f(v, t), t))$ , unde  $i \in F_1$ .

**IV.**  $F \equiv_s (\forall x)(\forall t)(P(x, g(h(x)), b) \vee \bigwedge Q(i(x)) \wedge \bigwedge R(f(j(x, t), t), t))$ , unde  $j \in F_2$ .

**Forma normală Skolem clauzală.** Pornim cu:

$$F^* = P(x, g(h(x)), b) \vee \bigwedge Q(i(x)) \wedge \bigwedge R(f(j(x, t), t), t).$$

Notăm  $P(x, g(h(x)), b)$  cu  $A_1$ ,  $Q(i(x))$  cu  $A_2$  și  $R(f(j(x, t), t), t)$  cu  $A_3$ .

Atunci  $F^*$  devine  $F^* = A_1 \vee \bigwedge A_2 \wedge \bigwedge A_3 = A_1 \vee (\bigwedge A_2 \wedge \bigwedge A_3) \equiv$

$(A_1 \vee \bigwedge A_2) \wedge (A_1 \vee \bigwedge A_3)$ . Prin urmare obținem în final că formula noastră  $F$  este slab echivalentă (relațiile  $\equiv$  și  $\equiv_s$  sunt tranzitive; de asemenea, dacă  $G \equiv H$  atunci  $G \equiv_s H$ ) cu  $F'$ , unde:

$$F' = (\forall x)(\forall t)((P(x, g(h(x)), b) \vee \bigwedge Q(i(x))) \wedge (P(x, g(h(x)), b) \vee \bigwedge R(f(j(x, t), t), t))). \blacksquare$$

Existența formelor normale, precum și a altor rezultate importante, ne permite un studiu suficient de detaliat al problemei satisfiabilității pentru calculul cu predicate de ordinul I, cu sau fără egalitate.

#### §4. Decidabilitate în LP1 (LP1=)

Vom începe direct cu un rezultat cumva așteptat.

**Teorema 3.10.** Fie  $F$  o formulă din calculul cu predicate de ordinul I fără egalitate, închisă și aflată în **FNS**. Atunci  $F$  este satisfiabilă dacă și numai dacă  $F$  admite un model Herbrand.

**Demonstrație.**

**Presupunem că  $F$  admite model Herbrand.** Dacă  $F$  admite model Herbrand, ea este satisfiabilă prin definiție.

**Presupunem că  $F$  este satisfiabilă.** Atunci există o structură  $S = \langle U_S, I_S \rangle$  corectă pentru formula dată astfel încât  $S \models F$ . Arătăm că în acest caz  $F$  admite și un model Herbrand, notat  $H_S = \langle U_H, I_H \rangle$  (pe scurt, modelul va fi notat  $H$ ). Ținând cont de definiția unei structuri Herbrand (orice term de bază este interpretat prin el însuși), rămâne să



interpretăm simbolurile predicative care intervin în formulă. Astfel, pentru fiecare  $n \in \mathbf{N}$ , pentru fiecare  $P \in P_n$  care apare în  $F$  și pentru fiecare  $t_1, t_2, \dots, t_n \in U_H = D(F)$ , vom pune:

$$(*) P^H(\langle t_1, t_2, \dots, t_n \rangle) = 1 \text{ dacă și numai dacă } P^S(\langle t_1^S, t_2^S, \dots, t_n^S \rangle) = 1.$$

Dacă  $F$  nu conține constante, alegem o constantă arbitrară  $a \in F_0$ , căreia îi dăm de asemenea o interpretare arbitrară  $a^S \in U_S$ . Pentru a arăta că  $H \models F$ , demonstrăm o **afirmație mai tare**:

Fie  $G$  orice formulă din **LP1** aflată în **FNP**, fără cuantori existențiali, închisă și conținând simboluri funcționale și predicative care apar doar în  $F$ . Atunci, pentru fiecare structură  $S = \langle U_S, I_S \rangle$  corectă care este model pentru  $G$ , structura  $H$  construită ca mai sus este de asemenea model pentru  $G$ .

**Demonstrația afirmației.** Procedăm prin inducție asupra numărului  $n$  de cuantificatori universali care apar în  $G$ .

**Baza.**  $n = 0$ . Atunci  $G$  nu conține cuantificatori și este închisă. Acest lucru înseamnă că  $G$  nu are variabile și vom arăta că  $S(G) = H(G)$ . Procedăm acum prin inducție structurală asupra submulțimii lui **LP1** alcătuită din formulele de tipul  $G$  de mai sus.

**Baza 1.**  $G$  este formulă atomică fără variabile, deci  $G = P(t_1, t_2, \dots, t_n)$ ,  $n \in \mathbf{N}$ ,  $P \in P_n$  și apare în  $F$ , iar  $t_1, t_2, \dots, t_n \in U_H$ . Egalitatea  $S(G) = H(G)$  urmează atunci direct din (\*).

**Pas inductiv 1.** Fie  $G_1, G_2$  formule fără variabile (și închise și fără cuantificatori), având simbolurile funcționale și predicative din  $F$ , pentru care  $S(G_1) = H(G_1)$  și  $S(G_2) = H(G_2)$ . Trebuie să

arătăm că avem  $S(G) = H(G)$  pentru fiecare  $G$  de una din formele:  $G = (G_1 \wedge G_2)$ ,  $G = (G_1 \vee G_2)$  și  $G = (\neg G_1)$  (formula fiind închisă, nu mai trebuie considerate cazurile  $(\forall x)(G_1)$  sau  $(\exists x)(G_1)$ ). Și acest lucru rezultă imediat din (\*) și definiția semanticii unei formule din **LP1**.

**Pas inductiv.** Să presupunem că afirmația noastră este adevărată pentru formule  $G$  de tipul considerat și conținând  $n - 1$  cuantificatori. Fie acum  $G$  o formulă având cel mult  $n$  cuantificatori, închisă, aflată în **FNP**, neavând cuantori existențiali și fiind construită peste simboluri funcționale și predicative din  $F$ . Fie  $S$  o structură corectă pentru  $G$ , astfel încât  $S \models G$  și  $H$  structura Herbrand dată prin (\*). Arătăm că  $H \models G$ . Conform ipotezelor,  $G = (\forall x)G'$ , unde  $G'$  conține cel mult  $n - 1$  universalii și  $x$  apare liber în  $G'$  (în caz contrar, ajungem până la urmă din nou în situația din **Baza**). Astfel, deși  $G'$  are restul proprietăților cerute pentru a putea aplica ipoteza inductivă, ea nu este formulă închisă. Dar, din  $S(G) = 1$  rezultă  $S((\forall x)G') = 1$ , de unde găsim că pentru fiecare  $u \in U_S$  avem  $S_{[x/u]}(G') = 1$ . Această relație este adevărată și pentru acei  $u$  din  $U_S$  care au proprietatea că sunt imagini prin structura  $S$  ai oricăror termi  $t$  fără variabile construiți peste simboluri din  $F$ ,  $u = t^S \in D(G) \subseteq D(F)$ . Aplicând **Lema de translație** obținem:  $1 = S_{[x/u]}(G') = S_{[x/t]}(G') = S((G')[x/t])$ . În acest moment dispunem de formula  $G'([x/t])$  care este închisă ( $t$  este term de bază), este în **FNP**, conține cel mult  $n - 1$  cuantificatori universalii și este

construită peste simboluri din  $F$ . Putem în sfârșit aplica ipoteza inductivă, deoarece relația anterioară ne spune și că  $S \models (G')[x/t]$ .

Atunci  $H \models (G')[x/t]$  și putem din nou aplica (în ultima relație) **Lema de translație** pentru a găsi  $(H_S)_{[x/t]}(G') = 1$  pentru fiecare  $t$  din  $D(G)$ , adică  $H((\forall x)G') = 1$ , ceea ce înseamnă de fapt că  $H(G) = 1$ , adică  $H$  este model pentru  $G$ . Cu aceasta, afirmația este demonstrată (**q. e. d.**). Pentru a încheia demonstrația teoremei, aplicăm direct rezultatul precedent pentru  $G = F$ . ■

**Observație.** Tot ceea ce am afirmat sau demonstrat până la teorema anterioară este aplicabil și în cazul calculului cu predicate de ordinul I cu egalitate. **Teorema 3.10 nu este însă adevărată și pentru  $LP1_{=}$** , după cum se poate intui chiar urmărind demonstrația, pentru succesul căreia relația (\*) este esențială. Astfel, în cazul în care  $P$  este chiar egalitatea, ea trebuie interpretată standard în orice structură. Prin urmare, considerând  $b$  și  $c$  două constante distincte, putem găsi ușor structuri  $S$  în care  $b^S = c^S$ , dar  $b \neq c$  în **orice** structură Herbrand. Desigur că acest lucru nu înseamnă că n-am putea găsi o altă demonstrație a rezultatului anterior, aplicabil pentru  $LP1_{=}$ . Din păcate, faptul nu este posibil, conform rezultatelor care urmează (nu toate demonstrate). ■

**Teorema 3.11 (Löwenheim – Skolem).** Fiecare formulă satisfiabilă din **LP1** admite model cel mult numărabil.

**Demonstrație.** Se aplică direct **Teorema 3.10.** ■

**Definiția 3.13 (extensia Herbrand).** Pentru fiecare formulă  $F$  închisă, aflată în **FNS**,  $F = (\forall y_1) \dots (\forall y_n)F^*$ ,  $\{y_1, \dots, y_n\} = \text{free}(F^*)$ ,  $F^*$  fiind matricea lui  $F$ , extensia sa Herbrand este mulțimea

$$E(F) = \{(F^*)[y_1/t_1] \bullet [y_2/t_2] \bullet \dots \bullet [y_n/t_n] \mid t_1, t_2, \dots, t_n \in D(F)\}.$$

Dacă  $F$  este în **FNSC** ( $F$  are forma de mai sus, în plus  $F^*$  fiind în **FNC**,  $F^* = C_1 \wedge C_2 \wedge \dots \wedge C_k$ ,  $C_1, C_2, \dots, C_k$  reprezentând clauze, adică literalii din **LP1**), mulțimea

$$E'(F) = \bigcup_{i=1}^k E(C_i)$$

se numește **extensia Herbrand generalizată.** ■

Extensia Herbrand generalizată a unei formule este obținută practic prin considerarea tuturor instanțelor clauzelor care compun matricea sa (formula fiind deja în **FNSC** și considerată în reprezentarea cu mulțimi), instanțe obținute prin aplicarea tuturor substituțiilor posibile cu termi de bază din  $D(F)$ .

**Teorema 3.12 (Church).** Problema validității pentru logica cu predicate de ordinul I (fără egalitate) este nedecidabilă, dar semidecidabilă.

**Demonstrație.** Conform, de exemplu, [MAS1]. ■

**Observație.** Ținând cont de legătura dintre  $F$  și  $\neg F$ , se deduce imediat că problema **SAT1** este nedecidabilă, ca de altfel și problema nesatisfiabilității (validității). Pe de altă parte, *în cazul LP1 aceste probleme sunt totuși semidecidabile.* ■

**Teorema 3.13 (Gödel-Herbrand-Skolem).** O formulă  $F \in \text{LP1}$  este satisfiabilă dacă și numai dacă  $E(F)$  este satisfiabilă.

**Demonstrație.** Fără a restrânge generalitatea, putem presupune că  $F$  este în **FNS** închisă. Conform **Teoremei 3.10**,  $F$  este satisfiabilă dacă și numai dacă admite model Herbrand  $H$ . Ca de obicei, putem considera că  $F = (\forall y_1) \dots (\forall y_n) F^*$  și atunci, din  $H(F) = 1$  rezultă că pentru fiecare  $t_1, t_2, \dots, t_n \in D(F)$  avem  $H_{[y_1/t_1][y_2/t_2]\dots[y_n/t_n]}(F^*) = 1$ . Deoarece termii în cauză sunt termi de bază, putem aplica **Lema de translație** de  $n$  ori, rezultând că  $H((F^*)[y_1/t_1] \bullet [y_2/t_2] \bullet \dots \bullet [y_n/t_n]) = 1$ , pentru fiecare  $t_1, t_2, \dots, t_n \in D(F)$ . Prin urmare, am arătat că  $F$  este satisfiabilă dacă și numai dacă  $H \vdash G$  pentru fiecare  $G \in E(F)$  adică dacă și numai dacă extensia Herbrand a lui  $F$  este satisfiabilă. ■

În **Teorema 3.13** putem lua  $E'(F)$  în loc de  $E(F)$  și, mai mult, așa cum de altfel am mai punctat, **elementele lui  $E'(F)$  pot fi privite drept clauze în sensul logicii propoziționale**, deoarece literalii din **LP1** care le compun nu conțin variabile (astfel, interpretarea lor, 0 sau 1, în orice structură  $S$ , depinde practic doar de interpretarea singurului

simbol predicativ care apare, ceea ce se reduce la interpretarea unei variabile propoziționale în sensul **LP**). Atunci are loc:

**Teorema 3.14 (teorema lui Herbrand; teorema de compactitate pentru LP1).** O formulă  $F \in \mathbf{LP1}$  este nesatisfiabilă dacă și numai dacă există o submulțime finită a lui  $E'(F)$  care să fie nesatisfiabilă.

**Demonstrație.** Direct din **Teorema 3.13** și **Teorema de compactitate pentru LP**. ■

În acest moment putem spune că procedura următoare (intitulată chiar în întregime „**Procedura lui Gilmore**”) poate fi *folosită pentru testarea nesatisfiabilității oricărei formule din LP1*. **Pasul 1** este un algoritm în sine (a se vedea și exemplul care urmează imediat **Teoremei 3.9**), formula găsită la sfârșitul execuției sale fiind slab echivalentă cu formula inițială și având aspectul  $F = (\forall *)F^*$ , unde  $F^* = C_1 \wedge C_2 \wedge \dots \wedge C_k$ . Extensia Herbrand generalizată  $E'(F)$  „rezultată” în urma aplicării **Pasului 2** trebuie interpretată ca fiind o listă de clauze din **LP**. Datorită faptului că  $E'(F)$  nu este recursivă ci doar recursiv enumerabilă, **Pasul 2** reprezintă doar un semialgoritm. După cum se observă, nici n-ar fi nevoie de obținerea acestei liste dintr-o dată. Este nevoie doar de a putea selecta câte un nou element din ea „atunci când este necesar”, conform **Pasului 3.3.2**, care ar putea fi reformulat prin *Obține un nou element din  $E'(F)$* . Practic, pornind de la ordinea pe  $D(F)$  deja sugerată (bazată pe lungimea termenilor), se poate defini o ordine totală și pe  $E'(F)$  (acest lucru nu ar implica decât o

simplă extensie a unei relații de ordine definită pe o mulțime „suport” la un produs cartezian al acelei mulțimi cu ea însăși, de oricâte ori).

**Pasul 3**, și numai el, este de fapt semialgoritmul cunoscut în literatura de specialitate ca *Procedura lui Gilmore* (*Procedura rezoluției de bază*). Aceasta nu se termină în cazul în care  $F$  este satisfiabilă și conține măcar un simbol funcțional de aritate cel puțin 1.

**Semialgoritmul lui Gilmore**  
**(Procedura rezoluției de bază)**

**Intrare:** Orice formulă  $F \in \mathbf{LP1}$ .

**Ieșire:** „DA”, doar dacă  $F$  este nesatisfiabilă.

**Metodă:**

**Pasul 1.** Se transformă  $F$  într-o formulă aflată în **FNSC** (închisă), succesiv, prin rectificare (redenumire), găsirea **FNP** (**FNPR**), obținerea închiderii existențiale, obținerea **FNS** și apoi **FNSC**, formula rezultat notându-se, pentru simplitate, tot cu  $F$ .

**Pasul 2.** Se „obține”  $E'(F) = \{G_1, G_2, \dots, G_n, \dots\}$ .

**Pasul 3.**

**3.1.**  $M := \emptyset$ .

**3.2.**  $i := 0$ .

**3.3. Repetă**

**3.3.1.**  $i := i + 1$ .

**3.3.2.** Alege  $G_i \in E'(F)$ .

**3.3.3.**  $M := M \cup \{G_i\}$ .

**3.3.4.**  $M' := \text{Res}^*(M)$ .

**Până\_când** ( $\quad \in M'$ ).

**Pasul 4.** Tipărește „DA”.

Trebuie însă să arătăm că (semi)algoritmul precedent „face ceea ce dorim”. Să precizăm de la bun început că vom lua în considerare doar formule  $F \in \mathbf{LP1}$  pentru care  $E'(F)$  este infinită. În caz contrar, rezultatele obținute până în prezent ne „spun” că  $F$  poate fi privită ca o formulă din **LP**, nemaifiind necesară o tratare a acesteia în noul context.

**Teorema 3.15.** Procedura rezoluției de bază pentru **LP1** este corectă.



**Demonstrație.** Trebuie de fapt să arătăm că pentru fiecare  $F \in \mathbf{LP1}$  având  $E'(F)$  infinită, semialgoritmul anterior se oprește dacă și numai dacă  $F$  este nesatisfiabilă. Acest lucru rezultă imediat din faptul că  $E'(F)$  este recursiv enumerabilă, din **Teorema lui Herbrand** și din **Teorema rezoluției** pentru **LP**. ■

Putem acum justifica de ce *logica propozițională poate fi considerată ca o aproximare a logicii cu predicate de ordinul I*. Semnificația este aceea că  $E(F)$  (sau  $E'(F)$ ) aproximează pe  $F$ , deoarece **testul de nesatisfiabilitate pentru  $F$  poate fi făcut testând de nesatisfiabilitate submulțimile finite ale lui  $E(F)$  ( $E'(F)$ )**. Pentru detalii privind noțiunile de *număr cardinal și ordinal, funcție (mulțime) recursivă (recursiv enumerabilă)*, alte amănunte privind *calculabilitatea, complexitatea, tratabilitatea*, pot fi consultate: [ȚIP], [CAZ2], [COR], [AHO], [BÖR], [RIC], etc.

**Exemplu.** Fie formula

$$F = (\forall x)(\forall y)((\neg P(x) \vee \neg P(f(c)) \vee Q(y)) \wedge P(y) \wedge (\neg P(g(b, x)) \vee \neg Q(b))).$$

Să se decidă dacă formula este nesatisfiabilă (satisfiabilă, validă), folosind **Procedura rezoluției de bază** (sau, altfel spus, folosind rezoluția din calculul propozițional). Să observăm că  $F$  este deja în **FNSC**. Avem și  $\text{free}(F^*) = \{x, y\}$ , iar  $F^* = \{C_1, C_2, C_3\}$ , unde:

$$C_1 = \neg P(x) \vee \neg P(f(c)) \vee Q(y)$$

$$C_2 = P(y)$$

$$C_3 = \neg P(g(b, x)) \vee \neg Q(b),$$

care la rândul lor se pot scrie ca mulțimi de literali din **LP1**. Putem trece astfel direct la **Pasul 2** al **Semialgoritmului lui Gilmore**, adică la aflarea mulțimii (listei)  $E'(F)$ . O metodă generală amintită deja se bazează întâi pe enumerarea lui  $D(F)$ , termii de bază fiind ordonați crescător, în ordinea lungimii lor (să precizăm și că  $b$  precede pe  $c$ ,  $f$  precede pe  $g$ , etc., și abia apoi folosim indicația referitoare la lungime). Găsim, succesiv, ceva de genul:

$$D(F) = \{b, c, f(b), f(f(b)), \dots, f^{(n)}(b), \dots, f(c), f(f(c)), \dots, f^{(m)}(c), \dots, \\ , g(b, b), g(b, c), g(c, b), g(c, c), g(b, f(b)), \dots, \\ , g(b, g(b, b)), \dots \}$$

$$E(C_1) = \{(C_1)[x/t_1] \bullet [y/t_2] \mid t_1, t_2 \in D(F)\} = \{\neg P(b) \vee \neg P(f(c)) \vee Q(b), \\ \neg P(b) \vee \neg P(f(c)) \vee Q(c), \neg P(c) \vee \neg P(f(c)) \vee Q(b), \\ \neg P(c) \vee \neg P(f(c)) \vee Q(c), \neg P(b) \vee \neg P(f(c)) \vee Q(f(b)), \dots\} = \\ \{\{\neg P(b), \neg P(f(c)), Q(b)\}, \{\neg P(b), \neg P(f(c)), Q(b)\}, \\ \{\neg P(b), \neg P(f(c)), Q(b)\}, \dots\}$$

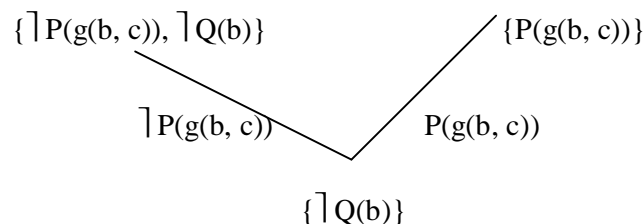
$$E(C_2) = \{\{P(b)\}, \{P(c)\}, \{P(f(b))\}, \{P(f(f(b)))\}, \{P(f^{(n)}(b))\}, \dots\}$$

$$E(C_3) = \{\neg P(g(b, b)) \vee \neg Q(b), \neg P(g(b, c)) \vee \neg Q(b), \\ \neg P(g(b, f(b))) \vee \neg Q(b), \neg P(g(b, f(f(b)))) \vee \neg Q(b), \\ \neg P(g(b, f^{(n)}(b))) \vee \neg Q(b), \dots\} = \{\{\neg P(g(b, b)), \neg Q(b)\}, \\ \{\neg P(g(b, c)), \neg Q(b)\}, \{\neg P(g(b, f(b))), \neg Q(b)\}, \\ \{\neg P(g(b, f(f(b)))) \vee \neg Q(b)\}, \{\neg P(g(b, f^{(n)}(b))), \neg Q(b)\}, \dots\}$$

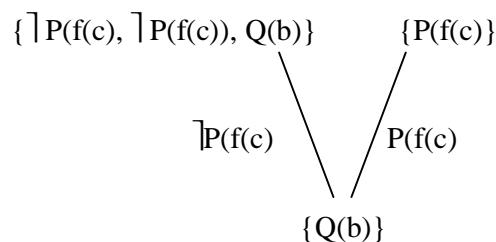
$$E'(F) = E(C_1) \cup E(C_2) \cup E(C_3).$$

Aplicarea **Pasului 3** *ad literam* este foarte anevoioasă, deoarece ar implica (ar trebui să amintim și de faptul că ar fi mai simplu să denotăm întâi formulele atomice de bază distincte de mai sus prin variabile

propoziționale distincte) un calcul succesiv nu numai de rezolvenți ci chiar de Res-uri, care, cel puțin teoretic, sunt (alte) mulțimi infinite recursiv enumerabile. **Este însă suficient (pentru a trage concluzia că F este nesatisfiabilă) să găsim o demonstrație prin rezoluție a clauzei vide pornind cu o submulțime (finită) de clauze ale lui E'(F).** Mai întâi găsim:



Cele două clauze care intervin provin din  $C_3$  (prin substituția  $[x/c]$ ) și respectiv  $C_2$  (prin substituția  $[y/g(b, c)]$ ). Conform definiției, termii respectivi sunt elemente din  $D(F)$ . Acum continuăm cu:



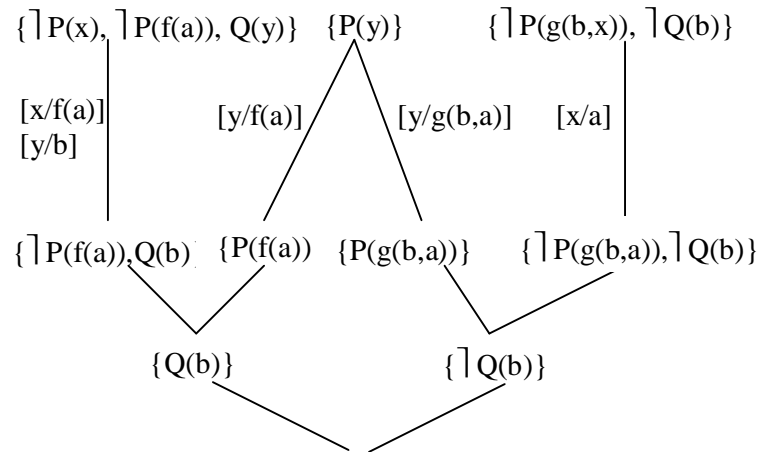
Deși la ultimul pas este incorectă scrierea de două ori a elementului  $\neg P(f(c))$ , am făcut-o pentru a se identifica mai ușor clauza de proveniență, care este  $C_1$  (substituția fiind  $[x/f(c)] \bullet [y/b]$ ). Pentru a doua clauză, ea este  $C_2$ , cu substituția  $[y/f(c)]$ . În final, din  $\{ \neg Q(b) \}$  și  $\{ Q(b) \}$  obținem  $\perp$ . ■

Bazându-ne pe exemplul anterior, putem enunța explicit rezultatul:

**Teorema 3.16 (a rezoluției de bază).** Fie  $F \in \mathbf{LP1}$  și  $E'(F')$  extensia Herbrand generalizată a unei formule  $F'$ , slab echivalente cu  $F$  și aflată în **FNSC** (închisă). Atunci  $F$  este nesatisfiabilă dacă și numai dacă există o demonstrație prin rezoluție (în sensul logicii propoziționale) a lui  $\perp$ , pornind cu (o parte finită din) elementele lui  $E'(F')$ .

**Demonstrație.** Imediată, din **Teorema lui Herbrand**, **Teorema 3.15** și ținând cont de legătura care există în **LP** între  $\text{Res}^*(F)$  și demonstrațiile prin rezoluție. ■

Ca urmare, folosind **Teorema 3.16**, **Pasul 3** al **Procedurii rezoluției de bază** poate fi ușor modificat. Pentru simplificarea înțelegerii, îi prezentăm doar execuția pentru cazul ultimului exemplu considerat:



Conchidem că aspectul **LP1** este următorul (subclasele indicate fiind nevide și disjuncte):

Formule valide	Formule satisfiabile, nevalide, având doar model infinit	Formule nesatisfiabile
	Formule satisfiabile, nevalide, având model finit	

Prin urmare, oricâte eforturi am depune, *într-un limbaj suficient de bogat tot vor exista formule satisfiabile, nevalide și având doar model infinit (numărabil), care nu va putea fi găsit algoritmic, într-un timp finit*. Totuși, posibilitatea *utilizării unei rezoluții proprii lui LP1*, prin care să se evite măcar „scufundarea” în **LP** dacă nu și aducerea formulelor la diferite forme normale, pare a fi o perspectivă benefică.

## §5. Rezoluție în LP1

**Rezoluția specifică** (numită și **pură**) pentru **LP1**, deși diferită de rezoluția de bază, păstrează ideea principală a rezoluției din **LP** și anume că **la fiecare pas de rezoluție se aleg două clauze și se obține o altă clauză (rezolvent), eliminând anumiți literalii prin „reducere” cu negațiile lor**. Eliminările devin mai complicate, iar „esența” lor este (sub)procedura de **unificare**. A unifica două sau mai multe formule atomice din **LP1** înseamnă a găsi o substituție pentru variabilele care intervin în acele formule (substituția nefiind elementară sau de bază) astfel încât în urma aplicării substituției formulele atomice respective să coincidă (textual, ca șiruri de caractere). Obținerea unui rezolvent nu va însemna numai simplă reducere a unui  $A$  cu un  $\neg A$ , ci și o unificare prealabilă a unor literalii, prin care se „desemnează cine este  $A$ ”. Nu vom intra în prea multe detalii, pentru amănunte suplimentare putându-se consulta [MAS1], [CAZ1], etc.

Din motive legate de spațiul tipografic, nu vom trata în această carte nici problematica legată de aplicarea rezoluției pure pentru cazul formulelor Horn și nici situația rafinărilor rezoluției pure (**Capitolul 5** va conține totuși o scurtă introducere).

**Exemplu.** Vom relua formula din ultimul exemplu, despre care am arătat deja că este nesatisfiabilă utilizând rezoluția de bază. Vom arăta că ea este nesatisfiabilă folosind rezoluția pură din **LP1**. Să precizăm și faptul că vom păstra toate notațiile pe care le-am introdus pentru **LP**

(cum ar fi Res, Res\*, etc.) deși semnificațiile exacte sunt de multe ori schimbate. După cum deja știm, avem:

$$F = (\forall x)(\forall y)((\neg P(x) \vee \neg P(f(c)) \vee Q(y)) \wedge P(y) \wedge (\neg P(g(b, x)) \vee \neg Q(b))),$$

$$C_1 = \neg P(x) \vee \neg P(f(c)) \vee Q(y),$$

$$C_2 = P(y),$$

$$C_3 = \neg P(g(b, x)) \vee \neg Q(b),$$

$$F^* = \{C_1, C_2, C_3\} =$$

$$\{\{\neg P(x), \neg P(f(c)), Q(y)\}, \{P(y)\}, \{\neg P(g(b, x)), \neg Q(b)\}\}.$$

Considerăm pe rând următoarele cupluri de clauze:

- **C<sub>1</sub> și C<sub>2</sub>.** Din motive tehnice, nu trebuie să existe variabile comune în clauzele considerate în momentul în care încercăm să aplicăm un pas al rezoluției pure. De aceea facem mai întâi *substituția de redenumire* [y/z] în C<sub>2</sub>, găsim C'<sub>2</sub> = {P(z)}. Apoi, prin [x/z], *unificăm* mulțimea { $\neg P(x)$ ,  $\neg P(z)$ } (acest din urmă literal fiind complementarul celui conținut de C'<sub>2</sub>), găsim Res(C<sub>1</sub>, C'<sub>2</sub>) = { $\neg P(f(c))$ , Q(y)} = C<sub>4</sub>.
- **C<sub>4</sub> și C<sub>2</sub>.** Facem aceeași redenumire în C<sub>2</sub> și lucrăm tot cu C'<sub>2</sub>. Aplicând [z/f(c)], vom unifica mulțimea { $\neg P(f(c))$ ,  $\neg P(z)$ }, găsim Res(C<sub>4</sub>, C'<sub>2</sub>) = {Q(y)} = C<sub>5</sub>.
- **C<sub>3</sub> și C<sub>2</sub>.** Nu mai avem nevoie de redenumiri, putând unifica { $\neg P(g(b, x))$ , P(y)}, prin [y/g(b, x)]. Găsim Res(C<sub>3</sub>, C<sub>2</sub>) = { $\neg Q(b)$ } = C<sub>6</sub>.
- **C<sub>5</sub> și C<sub>6</sub>.** Nu avem nevoie de redenumiri, unificăm {Q(y),  $\neg Q(b)$ } prin [y/b] și obținem în final Res(C<sub>5</sub>, C<sub>6</sub>) = .

Astfel, am găsit o respingere utilizând rezoluția pură, adică demonstrația:  $C_1, C_2, C_4, C_5, C_3, C_6, \dots$  ■

Deși multe dintre rezultatele corespunzătoare pentru **LP** nu sunt adevărate în cazul **LP1** (dacă înlocuim rezoluția **LP**, sau rezoluția de bază pentru **LP1** cu rezoluția pură), **Teorema rezoluției (pure)** are un enunț similar cu **Teorema rezoluției pentru LP** și în demonstrația acesteia se folosește în mod esențial **Teorema rezoluției de bază**. În acest mod, *găsirea unei respingeri de tipul anterior este suficientă pentru a trage concluzia că formula în cauză este nesatisfiabilă*.

Nu putem încheia acest capitol fără a sublinia încă o dată faptul că **situația în LP1<sub>=</sub> este și mai neplăcută**, cel puțin din punct de vedere teoretic. Se știe astfel că, dacă limbajul inițial este suficient de bogat, problema **SAT1** pentru **LP1<sub>=</sub>** este chiar nedecidabilă „total”, adică nu este nici măcar semidecidabilă (conform **Teoremei lui Church**), ca în cazul **LP1**. Astfel, este suficient ca el să conțină ([MAS1]), înafara simbolului care reprezintă egalitatea, un simbol funcțional constant (de aritate 0), două simboluri funcționale de aritate unu și un simbol predicativ de aritate doi ( $\wedge$ ,  $\neg$  și  $\exists$  sunt de asemenea necesari, ca și existența în  $X$  a cel puțin două variabile distincte). Pentru ca **SAT1** pentru **LP1<sub>=</sub>** să fie (măcar) semidecidabilă, este evident că ar trebui să putem arăta (măcar) că: **Orice formulă satisfiabilă din LP1<sub>=</sub> admite model cel mult numărabil**. Acest lucru nu poate fi, din păcate, adevărat, după cum se poate intui și din enunțul teoremelor următoare (demonstrațiile pot fi găsite de asemenea în [MAS1]):



**Teorema 3.17.** Fie  $F \in \mathbf{LP1}$ , satisfiabilă, care admite un model finit  $S = \langle U_S, I_S \rangle$  și să presupunem că  $|U_S| = n \in \mathbf{N}^*$ . Atunci, pentru fiecare  $m \geq n$ , există un model  $S_m$  pentru  $F$ , cu cardinalul universului egal cu  $m$ . Mai mult,  $F$  are și un model numărabil, care poate fi obținut și ca o „aproximare” a modelelor finite anterioare. ■

**Teorema 3.18.** Există o formulă închisă, satisfiabilă  $F \in \mathbf{LP1}$ , astfel încât pentru fiecare model  $S = \langle U_S, I_S \rangle$  al lui  $F$  avem  $|U_S| \geq 3$ . ■

**Teorema 3.19.** Există o formulă satisfiabilă  $F \in \mathbf{LP1}_=$ , astfel încât pentru fiecare model  $S = \langle U_S, I_S \rangle$  al lui  $F$  avem  $|U_S| \leq 2$ . ■

Inconvenientul practic al diferenței dintre  $\mathbf{LP1}$  și  $\mathbf{LP1}_=$  poate fi reparat parțial (după cum am mai precizat, exemplificând chiar acest lucru în **Capitolul 5**) prin „ascunderea” simbolului „=” într-un simbol predicativ de aritate superioară.

## §6. Recapitulare și Index

Deși teoria este foarte dezamăgitoare în privința rezultatelor pozitive (chiar dacă ne referim doar la calculul propozițional, unde decidabilitatea **SAT** este „atenuată” de **netratabilitate**), necesitatea exprimării simple a situațiilor reale a condus la acceptarea introducerii unor limbaje mai complexe (calculul cu predicate de ordinul I fără egalitate, calculul cu predicate de ordinul I cu egalitate, logicile cu predicate de ordinul II și mai mare, etc.). Desigur că din punct de

vedere formal rezultatele vor fi și mai descurajante. **Există însă câteva argumente solide care fac ca utilizarea logicii să constituie într-adevăr o modalitate interesantă de a interpreta programarea** (vom analiza acest lucru și în **Capitolul 5**). Pe parcursul **Capitolului 4**, am arătat că din punct de vedere formal atât **sintaxa** cât și **semantica logicii cu predicate de ordinul I** constituie o **extensie** simplă și naturală a **LP**, că logica propozițională este chiar o **aproximare** a **LP1** (în cel mai pur sens matematic), că noțiunile deja folosite și înțelese pentru **LP** pot fi extinse la **LP1** sau **LP1<sub>=</sub>** (rezultatele generale fiind și ele similare). Este vorba tot despre *exprimarea realității prin formule atomice și literali*, despre *obținerea de formule compuse din formule elementare* (variabilele, simbolurile funcționale și predicative și **cuantificatorii** fac diferența, dar la nivel intuitiv ele reprezintă chiar o necesitate de limbaj), despre *forme normale* și despre *posibilități efective de a testa satisfiabilitatea unei formule* (de natură sintactică sau semantică). Problema **SAT1** pentru **LP1<sub>=</sub>** este **nedecidabilă**, iar pentru **LP1** este **semidecidabilă**. Utilizarea rezoluției de bază ne arată o legătură mai profundă între **LP1** și **LP**. Posibilitatea folosirii unui tip specific de rezoluție - **rezoluția pură** - pentru **LP1**, a unor clase particulare de formule (cum ar fi **clasa formulelor Horn**), **adaptarea strategiilor și restricțiilor rezoluției** pentru cazul prezenței variabilelor, „scufundarea” realității în alte tipuri de logici, rămân alternative viabile în **programarea logică**.

Pentru **Index**, vom aminti:

*calculul cu predicate de ordinul I fără egalitate, 122*  
*constantă funcțională, 123*  
*simbol funcțional de aritate  $i$ , 123*  
*variabilă predicativă, 123*  
*simbol predicativ de aritate  $i$ , 123*  
*variabilă (funcțională), 123*  
*cuantificator (cuantor) existențial (universal), 123*  
*termi (funcționali), 125*  
*formule atomice, 125*  
*formule (subformule, arbori), 125*  
*apariții libere și legate ale variabilelor, 129*  
 *$free(F)$ ,  $var(F)$ ,  $leg(F)$ ,  $restvar(F)$ , 129*  
*formule închise, 130*  
*închiderea universală (existențială) a unei formule, 130*  
*matricea unei formule, 130*  
*substituții, substituții elementare, substituții permise, 130*  
*substituții normalizate, substituția vidă, substituții echivalente, 131*  
*termi (formule, substituții) de bază, 132*  
*domeniul sintactic al (apariției) unui cuantor, 132*  
*structuri și interpretări, 139*  
*extensie imediată, 140*  
*calculul cu predicate de ordinul I cu egalitate, 142*  
*univers și structuri Herbrand, 143*  
*mulțime recursiv enumerabilă, 146*  
*logica cu predicate de ordinul II, 150*  
*logici nestandard și de ordin superior, 150*

*formulă rectificată, 158*

*formă normală prenex, 159*

*formă normală prenex rectificată, 160*

*formă normală Skolem, 161*

*formă normală Skolem clauzală, 161*

*extensie Herbrand, extensie Herbrand generalizată, 171*

*rezoluție de bază (procedura lui Gilmore), 174*

*unificare, 181*

*rezoluție pură, 181*

## §7. Exerciții

1. Definiți constructiv  $\text{free}(F)$ ,  $F \in \mathbf{LP1}$ .
2. Determinați  $\text{subf}(F)$  pentru  

$$F = (\forall x)((P(x, g(a)) \wedge Q(z)) \vee R(u, f(v))).$$
3. Rezolvați **Exercițiul 3.1**.
4. Rezolvați **Exercițiul 3.2**.
5. Să se aplice substituția:  $s = [y/h(z)] \bullet [z/h(x)] \bullet [x/g(f(y))]$  formulei  

$$F = (\forall x)(P(x, f(x)) \wedge Q(g(a, z))).$$
6. Rezolvați **Exercițiul 3.3**.
7. Fie formula  $F = (\forall x)(\exists y)P(x, y, f(z))$ . Să se decidă dacă formula este satisfiabilă, validă, sau contradicție.
8. Găsiți o formulă  $F \in \mathbf{LP1}$  care să conțină un simbol predicativ  $P \in P_2$  și care să exprime faptul că  $P$  este o relație antisimetrică.

9. Găsiți o formulă  $F \in \mathbf{LP1}_=$  care să conțină un simbol predicativ  $P \in P_2$  și care să exprime faptul că  $P$  este o relație de echivalență.

10. Fie formula

$F = (\exists x)(\exists y)(\exists z)(P(x, y) \wedge P(z, y) \wedge P(x, z) \wedge \neg P(z, x))$  și structurile  $S = \langle U_S, I_S \rangle$  date prin:

- $U_S = \mathbf{N}, P^S = \{ \langle m, n \rangle \mid m, n \in \mathbf{N}, m < n \}$ .
- $U_S = \mathbf{N}, P^S = \{ \langle m, m+1 \rangle \mid m \in \mathbf{N} \}$ .
- $U_S = 2^{\mathbf{N}}, P^S = \{ \langle A, B \rangle \mid A, B \subseteq \mathbf{N}, A \subseteq B \}$ .

Să se decidă care dintre aceste structuri sunt modele pentru  $F$ .

11. Arătați că pentru fiecare formulă din  $\mathbf{LP1}$ :

- $F$  este validă dacă și numai dacă închiderea sa universală este validă.
- $F$  este satisfiabilă dacă și numai dacă închiderea sa existențială este satisfiabilă.

12. Demonstrați în detaliu **Teorema 3.1**.

13. Rezolvați **Exercițiul 3.4**.

14. Rezolvați **Exercițiul 3.5**.

15. Completați demonstrația **Teoremei 3.2**.

16. Demonstrați în detaliu **Teorema 3.5**.

17. Găsiți o formulă  $F'$  din  $\mathbf{LP1}$ , aflată în  $\mathbf{FNSC}$  și slab echivalentă cu formula:

$$F = (\forall x)(\exists y)((P(x, g(y), z) \vee \neg (\forall x)Q(x)) \wedge (\forall z)(\exists x) \neg R(f(x, z), z)).$$

18. Arătați că în  $\mathbf{LP1}$  există formule satisfiabile dar care nu admit nici un model finit.

19. Arătați că:

$$(\exists x)P(x) \rightarrow P(y) \equiv (\forall x)(P(x) \rightarrow P(y)).$$