BAZE DE CUNOŞTINŢE

Sinteza 2-3:

Semantici ale programelor Horn.

Nicolae Ţăndăreanu

Facultatea de Matematică-Informatică,

Universitatea din Craiova,

str.A.I.Cuza 13, 1100-Craiova, Romania

e-mail: ntand@oltenia.ro

1 Obiective

Obiectivele acestui capitol sunt următoarele:

- înţelegerea conceptului de semantică a unui program logic
- înţelegerea conceptului de semantică a celui mai mic model pentru programe Horn
- înţelegerea conceptului de semantică de punct fix pentru programele Horn
- însuşirea metodei de calcul a semanticii de punct fix şi implicit a semanticii celui mai mic model, cu ajutorul puterilor unui operator

2 Semantici ale programelor logice

Conceptul de **semantică** a unui program logic are la bază conceptele de model, model minimal şi cel mai mic model. Un model Herbrand al unui program ne permite să definim o anumită semantică prin intermediul căreia orice formulă peste baza programului poate fi evaluată din punct de vedere a valorii de adevăr. Pentru precizarea acestei idei să considerăm că P este un program logic peste baza B. Considerăm un model Herbrand I pentru P. Modelului I putem să-i asociem o relaţie notată cu $\models_I$, definită astfel: pentru orice formulă ground φ peste baza B, $P \models_I \varphi$

dacă şi numai dacă $truth\_val_{\Sigma_I}(\varphi, F) = true$. Astfel, să considerăm ca exemplu programul P de mai jos:

$$\begin{cases} p(a) \leftarrow \\ q(a) \leftarrow \\ p(f(x)) \leftarrow q(x) \end{cases}$$

Universul Herbrand al acestui program este

$$UH_P = \{a, f(a), \ldots, f^n(a), \ldots\}$$

iar baza Herbrand este $BH_P = \{p(f^n(a)), q(f^n(a))\}_{n \geq 0}$. Să considerăm următoarele două interpretări Herbrand:

$$I = \{p(a), q(a), p(f(a))\}$$

$$J = \{p(a), q(a), p(f(a)), p(f(f(a)))\}$$

Este uşor de verificat că ambele interpretări sunt modele Herbrand pentru programul P. Mai mult, I este cel mai mic model Herbrand pentru P. Într-adevăr, fie K cel mai mic model Herbrand al lui P. Aşadar $P \subseteq vd_{\Sigma_K}$, deci

$$truth\_val_{\Sigma_K}(p(a), F) = true$$

$$truth\_val_{\Sigma_K}(q(a), F) = true$$

$$truth\_val_{\Sigma_K}(\forall x(\neg q(x) \vee p(f(x))), F) = true$$

Din primele două relaţii rezultă $p(a) \in K$ şi $q(a) \in K$. Din a treia relaţie avem

$$truth\_val_{\Sigma_K}(\neg q(x) \vee p(f(x)), F\{x|c\}) = true$$

pentru orice $c \in UH_P$. În particular vom avea adevărată această relaţie pentru $c = a$. Dar $q(a) \in K$, deci

$$truth\_val_{\Sigma_K}(\neg q(x), F\{x|a\}) =$$

$$\neg truth\_val_{\Sigma_K}(q(x), F\{x|a\}) = \neg true = false$$

Aşadar trebuie să avem $truth\_val_{\Sigma_K}(p(f(x)), F\{x|a\}) = true$. Această proprietate are loc numai dacă $p(f(a)) \in K$. Rezultă că $I \subseteq K$. Dar K este cel mai mic model Herbrand al lui P, deci $K \subseteq I$ şi în consecinţă $I = K$.

Observăm că avem următoarele proprietăţi referitoare la cele două modele I şi J şi formula $\varphi = \neg p(f(f(a)))$:

- $I \subseteq J$
- $P \models_I \varphi$
- $P \not\models_J \varphi$

Într-adevăr, $truth\_val_{\Sigma_I}(\varphi, F) = \neg truth\_val_{\Sigma_I}(p(f(f(a)), F) = \neg false = true$ şi $truth\_val_{\Sigma_J}\ (\varphi, F) = \neg truth\_val_{\Sigma_J}(p(f(f(a)), F) = \neg true = false$. Astfel putem concluziona că formula φ este deductibilă din P în raport cu semantica definită de I, dar nu este deductibilă în raport cu semantica definită de J.

Să observăm că în raport cu o semantică precizată putem calcula valoarea de adevăr a oricărei formule peste baza B a programului P. De exemplu, să considerăm $\mathcal{S}_V = \{x, y\}$ şi formula $\varphi(x) = \exists y(p(f(f(x))) \wedge q(y))$. Nu există nici o asignare $F : \mathcal{S}_V \longrightarrow UH_P$ pentru care $\varphi(x)$ să aibă valoarea *true* în raport cu semantica definită de I. Într-adevăr, $truth\_val_{\Sigma_I}(\varphi(x), F) = true$ dacă şi numai dacă există $c \in UH_P$ astfel încât $truth\_val_{\Sigma_I}(p(f(f(x))) \wedge q(y), F\{y|c\}) = true$. Este evident că va trebui să luăm $c = a$, deoarece aceasta este singura alegere a lui c pentru care $truth\_val_{\Sigma_I}(q(y), F\{y|c\}) = true$. Mai departe, constatăm că $p(f(f(d))) \notin I$ pentru orice $d \in UH_P$. Nu acelaşi lucru putem spune în semantica definită de J. Într-adevăr, asignarea $F\{x|a, y|a\}$ are proprietatea că $truth\_val_{\Sigma_J}(\varphi(x), F) = true$.

Putem defini funcţia răspuns a unui program logic P în raport cu semantica definită de $I \subseteq BH_P$. Dacă $w \in FORM(B, \mathcal{S}_V)$ atunci definim

$$Ans_I(P, w) = \begin{cases} yes \text{ dacă } P \models_I w, w \text{ formula ground} \\ \\ no \text{ dacă } P \not\models_I w, w \text{ formulă ground sau } Assign_I(w) = \emptyset \\ \\ Assign_I(w) \text{ altfel} \end{cases}$$

unde $Assign_I(w) = \{F \mid truth\_val_{\Sigma_I}(w, F) = true\}$.
Pentru formula considerată mai înainte avem:

$$Ans_I(P, \varphi(x)) = no$$

$$Ans_J(P, \varphi(x)) = \{F\{x|a, y|a\}\}.$$

După modul de abordare şi modul de calcul a semanticii unui program logic, există mai multe categorii de semantici:

1) semantica celui mai mic model

2) semantica de punct fix

3) semantica de model stabil

4) semantica bine fondată

Din punct de vedere intuitiv a defini o semantică pentru un program logic înseamnă a specifica un anumit model Herbrand I a programului respectiv. Pentru fiecare tip de semantică enumerat mai sus trebuie precizată în mod corespunzător această mulţime I. De îndată ce se cunosc elementele lui I, se poate calcula valoarea de adevăr a oricărei formule peste baza programului prin utilizarea funcţiei $truth\_val_{\Sigma_I}$.

În acest capitol vom studia semantica celui mai mic model şi semantica de punct fix pentru programele Horn.

3 Semantica celui mai mic model pentru programe Horn

În clasificarea programelor logice prezentată în capitolul anterior, programele Horn sunt cele mai simple din punct de vedere sintactic. O piesă de cunoştinţe dată se poate uşor transpune într-un asemenea program şi orice persoană care citeşte un program Horn poate intui înţelesul regulilor prezentate. Problema care apare este aceea că uneori modelarea unei piese de cunoştinţe se face satisfăcător printr-un program Horn, alteori modelarea este nesatisfăcătoare. La analiza acestui fenomen se ţine seama de pierderea de semnificaţie a lumii prezentate într-un model sau altul. Uneori, chiar atunci când cunoaştem tipul programului logic în care dorim să transpunem piesa de cunoştinţe, trebuie să definim clauzele programului în aşa fel încât deducţia obţinută să fie cât mai aproape de lumea modelată. Pentru a evidenţia acest aspect să considerăm următoarea piesă de cunoştinţe:

Ana este mama lui Petre şi a lui George. O persoană X este fratele lui Y dacă X şi Y au aceeaşi mamă.

Evident, un exemplu de program logic care să modeleze această piesă de cunoştinţe este următo- rul:

$$\begin{cases} mama(ana, petre) \leftarrow \\ mama(ana, george) \leftarrow \\ frate(X, Y) \leftarrow mama(Z, X), mama(Z, Y) \end{cases}$$

Dar acest program nu modelează suficient de bine piesa considerată. Într-adevăr, pe baza acestui program putem deduce că *Petre este fratele lui George* şi *George este fratele lui Petre*, dar mai deducem că *Petre este fratele lui Petre* şi *George este fratele lui George*. Un asemenea raţionament ar permite să afirmăm că orice persoană este fratele acelei persoane. Pentru ca să excludem o astfel de deducţie, în locul ultimei clauze se consideră clauza

$$frate(X, Y) \leftarrow mama(Z, X), mama(Z, Y), X \neq Y$$

Pentru a evidenţia un alt aspect al modelării prin programe logice, să considerăm de asemenea următoarea piesă de cunoştinţe:

> *O societate de transport rutier STR organizează deplasări directe de la Bucureşti la Craiova, de la Bucureşti la Iaşi, de la Craiova la Timişoara şi de la Iaşi la Cluj. Există o deplasare de la localitatea X la localitatea Y dacă $X \neq Y$ şi există o deplasare directă de la X la Y sau există cel puţin o escală Z în drumul de la X la Y*

Problema care se pune este de a modela piesa considerată astfel încât un pasager să poată afla dacă se poate deplasa de la localitatea X la localitatea Y cu societatea STR.

Piesa de cunoştinţe considerată se poate modela în următorul program logic:

$$\begin{cases} direct(bucuresti, craiova) \leftarrow \\ direct(bucuresti, iasi) \leftarrow \\ direct(craiova, timisoara) \leftarrow \\ direct(iasi, cluj) \leftarrow \\ deplasare(X, Y) \leftarrow X \neq Y, direct(X, Y). \\ deplasare(X, Y) \leftarrow X \neq Y, direct(X, Z), deplasare(Z, Y). \end{cases}$$

Notăm cu P acest program Horn. De obicei predicatele standard $\neq, <, >$ $, \ldots$ se numesc **predicate built-in**, iar mulţimea acestora se notează cu BI. Ele sunt definite în cadrul softwareului cu care se lucrează şi interesează numai predicatele definite de utilizator. De aceea, baza programului P este $B = (\mathcal{S}_C, \emptyset, \mathcal{S}_P)$ unde

$$\mathcal{S}_C = \{bucuresti, craiova, iasi, timisoara, cluj\}$$

$$\mathcal{S}_P = \{direct^{(2)}, deplasare^{(2)}\}$$

În definitiv, a defini semantica unui program logic înseamnă să definim şi/sau să calculăm tot ceea ce este adevărat în virtutea faptelor cunoscute şi a utilizării regulilor enunţate în program. Astfel, intuitiv observăm că următoarele afirmaţii sunt adevărate relativ la programul P:

- există transporturi direct de la Bucureşti la Craiova, de la Bucureşti la Iaşi, de la Craiova la Timişoara şi de la Iaşi la Cluj

- există o deplasare de la Bucureşti la Timişoara şi există o deplasare de la Bucureşti la Cluj

- nu putem deduce că există o deplasare de la Craiova la Cluj şi prin urmare, putem considera că *nu există o deplasare de la Craiova la Cluj*; în asemenea raţionament spunem că utilizăm negaţia prin eşec.

Tratarea formalizată a semanticii programelor Horn se face prin definirea **semanticii celui mai mic model**, introdusă de Van Emden şi Kowalski. Ea se notează cu $LMS(P)$, iniţialele provin de la *least model semantics*, P este programul Horn considerat şi ea se defineşte astfel:

Definiţia 3.1 *Fie P un program Horn cu baza B. Mulţimea*

$$LMS(P) = \{p(c_1, \ldots, c_n) \in BH_P \mid P \models p(c_1, \ldots, c_n)\}$$

se numeşte **semantica celui mai mic model** *pentru P.*

Din definiţia dată se constată că $LMS(P) \subseteq BH_P$. Prin urmare $LMS(P)$ este o interpretare Herbrand. Să considerăm structura $\Sigma_{LMS(P)}$ ataşată acestei interpretări. Pentru orice atom ground $p(c_1, \ldots, c_n)$ peste baza B avem $truth\_val_{\Sigma_{LMS(P)}}(p(c_1, \ldots, c_n), F) = true$ dacă şi numai dacă

$p(c_1, \ldots, c_n) \in LMS(P)$. De aceea, valoarea de adevăr a oricărui element din $LMS(P)$ este $true$ în raport cu semantica celui mai mic model şi este $false$ pentru orice atom ground din afara lui $LMS(P)$.

Teorema care urmează justifică denumirea semanticii de mai sus.

Teorema 3.1 *(Van Emden, Kowalski)*
Fie P un program Horn. Notăm cu $MH(P)$ mulţimea tuturor modelelor Herbrand pentru P. Atunci:

$$LMS(P) = \bigcap_{W \in MH(P)} W$$

$$LMS(P) \in MH(P)$$

Proof. Să notăm $I = \bigcap_{W \in MH(P)} W$ şi să verificăm că $I \in MH(P)$. Echivalent, aceasta revine la a verifica că structura $\Sigma_I = (UH_P, k_I)$ ataşată lui I este un model pentru P, adică pentru orice $\varphi \in P$ avem $\varphi \in vd_{\Sigma_I}$. Deoarece P este un program Horn, este suficient să considerăm pentru φ numai două cazuri:

1) $\varphi = p(t_1, \ldots, p_n) \in BH_P \cap P$

Ţinând seama de definiţia lui I avem următoarele relaţii echivalente:

- $truth\_val_{\Sigma_I}(\varphi, F) = k_I(p)(t_1, \ldots, t_n) = true$ pentru orice F
- $p(t_1, \ldots, t_n) \in I$
- $p(t_1, \ldots, t_n) \in W$ pentru orice $W \in MH(P)$

- $k_W(p)(t_1, \ldots, p_n) = true$ pentru orice $W \in MH(P)$
- $truth\_val_{\Sigma_W}(\varphi, F) = true$ pentru orice $W \in MH(P)$,

iar ultima evaluare este adevărată deoarece pentru orice $W \in MH(P)$ avem $P \subseteq vd_W$, deci în particular $\varphi \in vd_{\Sigma_W}$.

2) φ este clauza $A(x_1, x_2) \leftarrow B(x_1, x_2)$, adică $\varphi = \forall x_1 \forall x_2 [A(x_1, x_2) \leftarrow B(x_1, x_2)]$. Va trebui să verificăm că

$$truth\_val_{\Sigma_I}(A(x_1, x_2) \vee \neg B(x_1, x_2), F\{x_1|c_1, x_2|c_2\}) = true$$

pentru orice $c_1, c_2 \in UH_P$.

Echivalent, trebuie să verificăm că pentru orice $c_1, c_2 \in UH_P$ avem

$$k_I(A)(c_1, c_2) \vee \neg k_I(B)(c_1, c_2) = true \tag{1}$$

Ţinând seama de definiţia lui I avem:

$$k_I(A)(c_1, c_2) = \bigwedge_{W \in MH(P)} k_W(A)(c_1, c_2) \tag{2}$$

$$k_I(B)(c_1, c_2) = \bigwedge_{W \in MH(P)} k_W(B)(c_1, c_2) \tag{3}$$

Pe de altă parte, dacă W este arbitrar în $MH(P)$ atunci

$$truth\_val_{\Sigma_W}(A(x_1, x_2) \vee \neg B(x_1, x_2), G\{x_1|c_1, x_2|c_2\}) = true$$

pentru orice $c_1, c_2 \in UH_P$, ceea ce înseamnă că pentru orice $W \in MH(P)$ şi orice $c_1, c_2 \in UH_P$ avem

$$k_W(A)(c_1, c_2) \vee \neg k_W(B)(c_1, c_2) = true \tag{4}$$

Fie c_1 şi c_2 două elemente arbitrare în UH_P. Avem două cazuri posibile:

1) pentru orice $W \in MH(P)$ avem $k_W(A)(c_1, c_2) = true$; în acest caz $k_I(A)(c_1, c_2) = true$

2) există $W \in MH(P)$ astfel încât $k_W(A)(c_1, c_2) = false$; în acest caz din (4) avem $k_W(B)(c_1, c_2) = false$, deci $k_I(B)(c_1, c_2) = false$ şi prin urmare (1) este adevărată.

Cu aceasta am verificat că $I = \bigcap_{W \in MH(P)} W \in MH(P)$ şi prin urmare I este cel mai mic model Herbrand pentru P.

Rămâne să demonstrăm că $LMS(P) = I$. Fie $p(c_1, \ldots, c_n) \in BH_P$. Avem următoarele relaţii echivalente:

- $p(c_1, \ldots, c_n) \in LMS(P)$
- $P \models p(c_1, \ldots, c_n)$
- $P \cup \{\neg p(c_1, \ldots, c_n)\}$ este inconsistentă
- nu există nici un model Herbrand pentru $P \cup \{\neg p(c_1, \ldots, c_n)\}$
- $k_W(p)(c_1, \ldots, c_n) = true$ pentru orice $W \in MH(P)$
- $k_I(p)(c_1, \ldots, c_n) = true$
- $p(c_1, \ldots, c_n) \in I$

aşadar $LMS(P) = I$.

Corolarul 3.1 *$LMS(P)$ este cel mai mic model Herbrand pentru programul Horn P.*

Remarcă 3.1 *$LMS(P)$ nu depinde de ordinea în care sunt aşezate clauzele în programul P*

Remarcă 3.2 *Teorema lui Van Emden şi Kowalski nu prezintă un algoritm pentru calculul lui $LMS(P)$*

4 Semantica de punct fix a programelor Horn

4.1 Elemente de punct fix

O **mulţime parţial ordonată** este o pereche $(P, \leq)$ unde P este o mulţime nevidă, iar $\leq$ este o relaţie binară pe mulţimea P astfel încât $\leq$ este reflexivă, antisimetrică şi tranzitivă.

Dacă $Q \subseteq P$ atunci un **majorant** pentru Q este un element $z \in P$ astfel încât $x \leq z$ pentru orice $x \in Q$. Cel mai mic majorant pentru Q (dacă există) se notează cu **sup Q** şi se numeşte **supremumul** mulţimii Q. Dual apare conceptul de **minorant** pentru Q şi cel mai mare minorant al lui Q, numit **infimumul** lui Q şi este notat **inf Q** (dacă există).

Definiţia 4.1 *O* **latice** L *este o mulţime parţial ordonată cu proprietatea că pentru orice* $x, y \in L$ *există* $sup\{x, y\}$ *şi există* $inf\{x, y\}$.

Deoarece într-o mulţime parţial ordonată supremumul şi infimumul unei mulţimi sunt unic determinate, dacă există, putem defini pentru o latice L două legi de compoziţie, numite *reuniune* şi *intersecţie*:

$$\cup : L \times L \longrightarrow L, \quad x \cup y = sup\{x, y\}$$

$$\cap : L \times L \longrightarrow L, \quad x \cup y = inf\{x, y\}$$

Într-o latice $(L, \leq)$ putem defini **duala relaţiei** $\leq$, pe care o notăm $\geq$:

$x \geq y$ dacă şi numai dacă $y \leq x$

Relaţia duală este tot o relaţie parţială de ordine. Putem uşor observa că un element t_0 este un majorant pentru mulţimea Q în relaţia duală dacă şi numai dacă t_0 este un minorant pentru Q în relaţia iniţială. La fel, conceptul de minorant în relaţia duală este conceptul de majorant în relaţia iniţială. Pentru acest motiv cele două concepte se numesc *concepte duale*: dualul conceptului de minorant este conceptul de majorant şi dualul conceptului de majorant este conceptul de minorant. Tot aşa conceptele de infimum şi supremum sunt concepte duale. Se observă că în teoria laticilor este valabil *principiul dualităţii*: de îndată ce este adevărată o anumită proprietate, este adevărată şi duala ei, care se obţine din proprietatea iniţială înlocuind fiecare noţiune cu duala ei. Astfel, duala proprietăţii **supremumul unei mulţimi, dacă există, este unic determinat** este proprietatea *infimumul unei mulţimi, dacă există, este unic determinat.*

Propoziţia 4.1 *Într-o latice L operaţiile $\cup$ şi $\cap$ satisfac proprietăţile:*

1) comutativitatea:

$$x \cup y = y \cup x; \quad x \cap y = y \cap x$$

pentru orice $x, y \in L$

2) asociativitatea:

$$x \cup (y \cup z) = (x \cup y) \cup z; \quad x \cap (y \cap z) = (x \cap y) \cap z$$

pentru orice $x, y, z \in L$

3) legile de absorbţie:

$$x \cup (x \cap y) = x; \quad x \cap (x \cup y) = x$$

pentru orice $x, y \in L$

Proof.

1) Pentru orice $x, y \in L$ avem $x \cup y = sup\{x, y\} = sup\{y, x\} = y \cup x$; dual se verifică comutativitatea intersecţiei.

2) Pentru a demonstra asociativitatea reuniunii se observă că pentru orice $x, y, z \in L$ avem

$$x \cup (y \cup z) = sup\{x, y, z\}$$

Într-adevăr, $x \leq x \cup (y \cup z)$ deoarece $x \cup (y \cup z) = sup\{x, y \cup z\}$, $y \leq y \cup z \leq x \cup (y \cup z)$, $z \leq y \cup z \leq x \cup (y \cup z)$, deci $x \cup (y \cup z)$ este un majorant pentru $\{x, y, z\}$. Fie t_0 un alt majorant pentru mulţimea $\{x, y, z\}$:

$$x \leq t_0, \quad y \leq t_0, \quad z \leq t_0$$

Deoarece $y \cup z = sup\{y, z\}$ şi t_0 este un majorant pentru mulţimea $\{y, z\}$ rezultă că $y \cup z \leq t_0$. Aşadar t_0 este un majorant pentru mulţimea $\{x, y \cup z\}$, deci $x \cup (y \cup z) \leq t_0$. În concluzie, $x \cup (y \cup z)$ este cel mai mic majorant pentru mulţimea $\{x, y, z\}$, deci $x \cup (y \cup z) = sup\{x, y, z\}$. Acum asociativitatea reuniunii rezultă imediat deoarece $x \cup (y \cup z) = sup\{x, y, z\} = sup\{z, x, y\} = z \cup (x \cup y) = (x \cup y) \cup z$ în baza comutativităţii reuniunii.

Dual se demonstrează că $x \cap (y \cap z) = inf\{x, y, z\}$ şi prin urmare operaţia de intersecţie este asociativă.

3) Avem $x \leq x \cup (x \cap y)$ deoarece $x \cup (x \cap y) = sup\{x, x \cap y\}$. Dar $x \cap y \leq x$ şi deci x este un majorant pentru mulţimea $\{x, x \cap y\}$. Rezultă că $x \cup (x \cap y) = sup\{x, x \cap y\} \leq x$. Prin antisimetrie rezultă că $x = x \cup (x \cap y)$. Dual se demonstrează că $x = x \cap (x \cup y)$. ∎

Să observăm dualitatea relaţiilor din **Propoziţia 4.1**:

$$x \cup y = y \cup x; \quad x \cap y = y \cap x$$

$$x \cup (y \cup z) = (x \cup y) \cup z; \quad x \cap (y \cap z) = (x \cap y) \cap z$$

$$x \cup (x \cap y) = x; \quad x \cap (x \cup y) = x$$

În consecinţă, operaţiile de reuniune şi intersecţie într-o latice sunt *operaţii duale.*

Să considerăm tripletul $(L, \cup, \cap)$ în care $\cup : L \times L \longrightarrow L$, $\cap : L \times L \longrightarrow L$ satisfac proprietăţile:

$x \cup y = y \cup x; \quad x \cap y = y \cap x$ pentru orice $x, y \in L$

$x \cup (y \cup z) = (x \cup y) \cup z; \quad x \cap (y \cap z) = (x \cap y) \cap z$ pentru orice $x, y, z \in L$

$x \cup (x \cap y) = x; \quad x \cap (x \cup y) = x$ pentru orice $x, y \in L$

Observăm că în acest context avem următoarea proprietate:

$$x \cup y = y \Longleftrightarrow x \cap y = x$$

Într-adevăr, dacă $x \cup y = y$ atunci $x \cap y = x \cap (x \cup y) = x$. Reciproc, dacă $x \cap y = x$ atunci $x \cup y = (x \cap y) \cup y = y \cup (y \cap x) = y$.

Definim următoarea relaţie binară:

$$x \leq y \Longleftrightarrow x \cup y = y$$

Aplicând de două ori legile de absorbţie constatăm că $x = x \cup (x \cap (x \cup x)) = x \cup x$ pentru orice $x \in L$ şi prin urmare relaţia $\leq$ este reflexivă. Dacă $x \leq y$ şi $y \leq x$ atunci $x \cup y = y$, $y \cup x = x$ deci $x = y$ în baza comutativităţii operaţiei $\cup$. Tranzitivitatea relaţiei $\leq$ se obţine imediat: dacă $x \leq y$ şi $y \leq z$ atunci $x \cup y = y$, $y \cup z = z$ deci $x \cup z = x \cup (y \cup z) = (x \cup y) \cup z = y \cup z = z$.

Aşadar $(L, \leq)$ devine o mulţime parţial ordonată. Putem uşor verifica faptul că în această mulţime L există $sup\{x, y\}$ şi există $inf\{x, y\}$ pentru orice $x, y \in L$. Mai mult, $sup\{x, y\} = x \cup y$ şi $inf\{x, y\} = x \cap y$. Într-adevăr, $x \cup (x \cup y) = (x \cup x) \cup y = x \cup y$ deci $x \leq x \cup y$. Similar $y \leq x \cup y$. Dacă $x \leq t_0$ şi $y \leq t_0$ atunci $x \cup t_0 = t_0$, $y \cup t_0 = t_0$, deci $(x \cup y) \cup t_0 = x \cup (y \cup t_0) = x \cup t_0 = t_0$. Astfel $x \cup y$ este cel mai mic majorant al mulţimii $\{x, y\}$, adică $x \cup y = sup\{x, y\}$. Similar, sau prin dualitatea proprietăţilor celor două operaţii de reuniune şi intersecţie avem $x \cap y = inf\{x, y\}$. În consecinţă $(L, \cup, \cap)$ este o latice.

Definiţia 4.2 *O latice $(L, \leq)$ se numeşte* **completă** *dacă orice submulţime nevidă a ei are supremum şi infimum. Un operator $T : L \longrightarrow L$ se numeşte* **monoton** *dacă din $x \leq y$ rezultă $T(x) \leq T(y)$. Un element $x \in L$ se numeşte* **punct fix** *al operatorului T dacă $T(x) = x$. Un element x_0 este* **un cel mai mic punct fix** *al lui T dacă $T(x_0) = x_0$ şi pentru orice $a \in L$ pentru care $T(a) = a$ avem $x_0 \leq a$.*

Propoziţia 4.2 *Cel mai mic punct fix al unui operator, dacă există, este* *unic.*

Proof. Dacă x_0, y_0 sunt cele mai mici puncte fixe pentru T atunci $T(x_0) = x_0$, $T(y_0) = y_0$, $x_0 \leq y_0$ pentru că x_0 este un cel mai mic punct fix; $y_0 \leq x_0$ pentru că y_0 este un cel mai mic punct fix pentru T. Prin antisimetria relaţiei $\leq$ rezultă $x_0 = y_0$. ∎

Cel mai mic punct fix al operatorului T, dacă există, îl notăm cu $lfp(T)$.

Teorema 4.1 *(Knaster-Tarski)*
Fie $(L, \leq)$ o latice completă. Dacă $T : L \longrightarrow L$ este un operator monoton atunci T admite lfp şi

$$lfp(T) = inf\{x \in L \mid T(x) = x\} = inf\{x \in L \mid T(x) \leq x\}$$

Proof. Notăm

$$A = \{x \in L \mid T(x) = x\}, \quad B = \{x \in L \mid T(x) \leq x\}$$

Vom arăta la început că $A \neq \emptyset$ şi $B \neq \emptyset$. Într-adevăr, L fiind o latice completă există $supL$, pe care îl notăm cu 1. Evident $T(1) \leq 1$ deoarece 1 este cel mai mare element al lui L. Astfel, $1 \in B$. Notăm $g_2 = infB$, acest element există pentru că $B \neq \emptyset$. Din definiţia elementului g_2 rezultă că $g_2 \leq x$ pentru orice $x \in B$, deci $T(g_2) \leq T(x) \leq x$ pentru orice $x \in B$. Aşadar $T(g_2)$ este un minorant pentru B, deci $T(g_2) \leq g_2$ şi prin urmare $g_2 \in B$. Operatorul T este monoton deci $T(T(g_2)) \leq T(g_2)$, prin urmare

$T(g_2) \in B$. Deoarece $g_2 = infB$ şi $T(g_2) \in B$ rezultă că $g_2 \leq T(g_2)$. Aşadar $T(g_2) = g_2$, deci $g_2 \in A$. Rezultă că $A \neq \emptyset$ şi prin urmare există $g_1 = infA$. Pentru orice $x \in A$ avem $g_1 \leq x$ deci $T(g_1) \leq T(x) = x$. Dar g_1 este cel mai mare minorant pentru A, deci $T(g_1) \leq g_1$. Rezultă că $g_1 \in B$, deci $g_2 \leq g_1$. Deoarece $g_1 = infA$, rezultă că $g_1 \leq g_2$. Prin urmare $g_1 = g_2$. Deoarece $T(g_2) = g_2$ şi $g_2 = g_1$ rezultă $T(g_1) = g_1$ şi deci g_1 este un punct fix pentru T. Evident g_1 este cel mai mic punct fix pentru T. ∎

Definiţia 4.3 *Fie L o latice. O submulţime $X \subseteq L$ se numeşte* **mulţime dirijată** *dacă pentru orice parte finită $F \subseteq X$ avem $supF \in X$.*

Definiţia 4.4 *Fie L o latice completă. Un operator $T : L \longrightarrow L$ se numeşte* **operator continuu** *dacă $T(supX) = supT(X)$ pentru orice submulţime dirijată X a lui L.*

Propoziţia 4.3 *Orice operator continuu este monoton.*

Proof. Presupunem că $x \leq y$. Mulţimea $X = \{x, y\}$ este dirijată, deci $T(y) = T(supX) = supT(X) = sup\{T(x), T(y)\}$. Rezultă că $T(x) \leq T(y)$. ∎

Remarcă 4.1 *Să observăm că dacă $T : L \longrightarrow L$ este un operator monoton atunci $sup\{T(X)\} \leq T(supX)$. Într-adevăr, pentru orice $x \in X$ avem $x \leq supX$ deci $T(x) \leq T(supX)$. Aşadar $T(supX)$ este un majorant pentru $T(X)$, deci $supT(X) \leq T(supX)$.*

4.2 Puterile unui operator monoton şi proprietăţi ale acestora

Notăm cu $\emptyset$ mulţimea vidă. *Ordinalele finite* se notează cu $0, 1, 2, 3, \ldots$ şi se definesc astfel:

$$0 = \emptyset$$

$$1 = \{0\};\ 2 = \{0, 1\};\ 3 = \{0, 1, 2\}, \ldots$$

Se notează $\omega = \{0, 1, 2, \ldots\}$ şi acesta este primul *ordinal infinit*. Pentru orice ordinal α putem defini $\alpha + 1 = \alpha \cup \{\alpha\}$. Observăm că $0 + 1 = 1$, $1+1 = 2$, $2+1 = 3$ etc. Pentru două ordinale α şi β definim $\alpha < \beta$ dacă şi numai dacă $\alpha \in \beta$. Se poate demonstra că pentru orice ordinal α, numărul $\alpha + 1$ este cel mai mic ordinal care depăşeşte pe α în relaţia definită mai sus. Numărul $\alpha+1$ se numeşte *succesorul* lui α. Un ordinal α se numeşte *ordinal limită* dacă nu există un ordinal β astfel încât $\alpha = \beta+1$. Numărul 0 este primul ordinal limită. Următorul ordinal limită este ω. Numerele $\omega + 1$, $\omega + 2$, ... sunt ordinale succesoare. Următorul ordinal limită este $\omega + \omega$ şi acesta se notează cu $\omega.2$.

Considerăm o latice completă L. Notăm $0 = inf L$. În definiţia care urmează definim puterile ordinale ale unui operator.

Definiţia 4.5 *Pentru un operator monoton $T : L \longrightarrow L$ definim:*

$$T \uparrow 0 = 0$$

$$T \uparrow \alpha = \begin{cases} T(T \uparrow (\alpha - 1)) & \text{dacă } \alpha \text{ este ordinal succesor} \\ \\ sup\{T \uparrow \beta \mid \beta < \alpha\} & \text{dacă } \alpha \text{ este ordinal limită} \end{cases}$$

Observăm că puterile unui operator $T : L \longrightarrow L$ sunt elemente din L. În continuare vom demonstra prin inducţie transfinită câteva proprietăţi ale puterilor unui operator, necesare definirii semanticii de punct fix a unui program Horn.

Propoziţia 4.4 *Pentru orice ordinal α avem $T \uparrow \alpha \leq lfp(T)$*

Proof.

- Pentru $\alpha = 0$ proprietatea este adevărată
- Presupunem că proprietatea este adevărată pentru orice ordinal $\beta < \alpha$ şi să arătăm că ea este adevărată pentru α.
  - Dacă α este un ordinal succesor atunci $T \uparrow \alpha = T(T \uparrow (\alpha - 1)) \leq T(lfp(T)) = lfp(T)$
  - Dacă α este ordinal limită atunci $T \uparrow \alpha = sup\{T \uparrow \beta \mid \beta < \alpha\} \leq lfp(T)$

■

Propoziţia 4.5 *Pentru orice ordinal α avem $T \uparrow \alpha \leq T \uparrow (\alpha + 1)$*

Proof.

- Pentru $\alpha = 0$ proprietatea este adevărată
- Presupunem că proprietatea este adevărată pentru orice ordinal $\beta < \alpha$ şi să arătăm că ea este adevărată pentru α.
  - Dacă α este un ordinal succesor atunci $T \uparrow \alpha = T(T \uparrow (\alpha - 1)) \leq T(T \uparrow \alpha)) = T \uparrow (\alpha + 1)$
  - Dacă α este un ordinal limită atunci $T \uparrow \alpha = sup\{T \uparrow \beta \mid \beta < \alpha\} \leq sup\{T \uparrow (\beta + 1) \mid \beta < \alpha\} \leq T(sup\{T \uparrow \beta \mid \beta < \alpha\}) = T(T \uparrow \alpha) = T \uparrow (\alpha + 1)$

■

Propoziţia 4.6 *Dacă $\alpha < \beta$ atunci $T \uparrow \alpha \leq T \uparrow \beta$*

Proof. Demonstrăm proprietatea prin inducţie după β. Pentru $\beta = 1$ proprietatea este evident adevărată.

- Presupunem că β este un ordinal succesor. Fie $\alpha < \beta$, deci $\alpha \leq \beta - 1$. Prin ipoteza inductivă proprietatea este adevărată pentru $\beta - 1$, deci $T \uparrow \alpha \leq T \uparrow (\beta - 1)$. Prin propoziţia 4.5 avem $T \uparrow (\beta - 1) \leq T \uparrow \beta$ şi prin urmare proprietatea este verificată în acest caz.
- Fie β un ordinal limită. Fie $\alpha < \beta$ şi γ un ordinal astfel încât $\alpha < \gamma < \beta$. Avem: $T \uparrow \alpha \leq T \uparrow \gamma$ prin ipoteza inductivă şi $T \uparrow \gamma \leq sup\{T \uparrow \delta) \mid \delta < \beta\} = T \uparrow \beta$

■

Propoziţia 4.7 *Dacă $\alpha < \beta$ şi $T \uparrow \alpha = T \uparrow \beta$ atunci $T \uparrow \alpha = lfp(T)$.*

Proof. În ipotezele propoziţiei de demonstrat avem $T \uparrow \alpha \leq T \uparrow (\alpha + 1) \leq T \uparrow \beta$, deci $T \uparrow \alpha = T \uparrow (\alpha + 1) = T \uparrow \beta$. Aşadar $T \uparrow \alpha = T \uparrow (\alpha + 1) = T(T \uparrow \alpha)$ şi astfel $T \uparrow \alpha$ esste un punct fix pentru T. Deoarece $lfp(T)$ este cel mai mic punct fix rezultă că $lfp(T) \leq T \uparrow \alpha$. Aplicând propoziţia 4.4 rezultă că $T \uparrow \alpha = lfp(T)$. ■

Propoziţia 4.8 *Există un ordinal β astfel încât pentru orice $\gamma \geq \beta$ avem $T \uparrow \gamma = lfp(T)$.*

Proof. Fie α cel mai mic ordinal cu $Card(\alpha) > Card(L)$. Presupunem că pentru orice $\delta < \alpha$ avem $T \uparrow \delta \neq lfp(T)$. Definim funcţia $h : \alpha \longrightarrow L$ prin $h(\delta) = T \uparrow \delta$. Prin propoziţia 4.7 rezultă că funcţia h este injectivă, ceea ce este imposibil prin alegerea lui α. Aşadar există $\beta < \alpha$ astfel încât $T \uparrow \beta = lfp(T)$ Din monotonia operatorului T şi propoziţia 4.4 avem $lfp(T) = T \uparrow \beta \leq T \uparrow \alpha \leq lfp(T)$, deci $T \uparrow \alpha = lfp(T)$. Fie $\gamma \geq \beta$.

- dacă $\gamma \leq \alpha$ atunci $T \uparrow \beta \leq T \uparrow \gamma \leq T \uparrow \alpha$ deci $T \uparrow \beta = T \uparrow \gamma = T \uparrow \alpha = lfp(T)$
- dacă $\gamma > \alpha$ atunci $T \uparrow \gamma \geq T \uparrow \alpha = lfp(T)$; din propoziţia 4.4 avem $T \uparrow \gamma \leq lfp(T)$, deci $T \uparrow \gamma = lfp(T)$.

■

Propoziţia 4.9 *Fie L o latice completă. Dacă $T : L \longrightarrow L$ este un operator continuu atunci $lfp(T) = T \uparrow \omega$.*

Proof. Să demonstrăm că $T \uparrow \omega$ este un punct fix pentru T. Mulţimea $\{T \uparrow n \mid n \in \omega\}$ este dirijată deoarece T este monoton. Avem:
$T(T \uparrow \omega) = T(sup\{T \uparrow n \mid n \in \omega\} = sup\{T \uparrow (n+1) \mid n \in \omega\} = sup\{T \uparrow n \mid n < \omega\} = T \uparrow \omega$.
Deoarece $T \uparrow \omega$ este punct fix pentru T, rezultă că $lfp(T) \leq T \uparrow \omega$. Prin propoziţia 4.4 avem $T \uparrow \omega \leq lfp(T)$, deci $T \uparrow \omega = lfp(T)$. ■

4.3 Definiţia semanticii de punct fix pentru programe Horn

Considerăm un program Horn P şi baza sa Herbrand BH_P. Mulţimea $(2^{BH_P}, \subseteq)$ a tuturor părţilor lui BH_P este o latice completă în raport cu relaţia de incluziune din teoria mulţimilor.

Definiţia 4.6 *Definim operatorul $T_P : 2^{BH_P} \longrightarrow 2^{BH_P}$ astfel: $T_P(I)$ este mulţimea tuturor atomilor ground $p(c_1, \ldots, c_n)$ pentru care există $p(t_1, \ldots, t_n) \leftarrow B_1, \ldots, B_k$ în P şi există o substituţie ground θ astfel încât $B_1\theta \in I$, ..., $B_k\theta \in I$ şi $p(t_1, \ldots, t_n)\theta = p(c_1, \ldots, c_n)$.*

Să observăm că operatorul T_P definit mai sus este monoton, adică $T_P(I) \subseteq T_P(J)$ dacă $I \subseteq J$. De asemenea, observăm următoarea particularitate a laticii $L = 2^{BH_P}$: fie $X \subseteq L$ o mulţime dirijată şi atomii ground

$A_1, \ldots, A_n$; avem $\{A_1, \ldots, A_n\} \subseteq supX$ dacă şi numai dacă există $I \in X$ astfel încât $\{A_1, \ldots, A_n\} \subseteq I$. Într-adevăr, dacă $A_1, \ldots, A_n \in supX = \bigcup_{I \in X} I$ atunci există $i_1, \ldots, i_n$ şi există $I_{i_1}, \ldots, I_{i_n} \in X$ astfel încât $A_{i_1} \in I_{i_1}$, ..., $A_{i_n} \in I_{i_n}$. Aşadar $\{A_1, \ldots, A_n\} \subseteq sup\{I_{i_1}, \ldots, I_{i_n}\}$, iar ultima mulţime este din X deoarece X este dirijată.

Cu ajutorul operatorului T_P putem caracteriza acele interpretări Herbrand care sunt modele Herbrand pentru P. Această caracterizare este prezentată în propoziţia următoare:

Propoziţia 4.10 *Fie P un program Horn. O interpretare Herbrand I a lui P este un model Herbrand pentru P dacă şi numai dacă $T_P(I) \subseteq I$.*

Proof. Presupunem că I este un model Herbrand pentru P, deci structura $\Sigma_I = (UH_P, k_I)$ este un model pentru P. Să verificăm incluziunea $T_P(I) \subseteq I$. Considerăm un element $p(c_1, \ldots, c_n) \in T_P(I)$. Aceasta înseamnă că există o formulă $\forall x_1 \ldots \forall x_s[p(t_1, \ldots, t_n) \leftarrow B_1, \ldots, B_k]$ a programului P şi există o substituţie ground $\theta = \{x_1|d_1, \ldots, x_s|d_s\}$, unde $d_1, \ldots, d_s$ sunt elemente din UH_P, astfel încât $\{B_1\theta, \ldots, B_k\theta\} \subseteq I$ şi $p(t_1, \ldots, t_n)\theta = p(c_1, \ldots, c_n)$. Deoarece Σ_I este un model pentru programul P, avem

$$truth\_val_{\Sigma_I}(\forall x_1 \ldots \forall x_s[p(t_1, \ldots, t_n) \leftarrow B_1, \ldots, B_k], F) = true$$

pentru orice asignare F. Echivalent, aceasta înseamnă că

$$truth\_val_{\Sigma_I}([p(t_1, \ldots, t_n) \leftarrow B_1, \ldots, B_k], F\{x_1|c_1, \ldots, x_s|c_s\}) = true$$

pentru orice $c_1, \dots, c_s \in UH_P$. În particular vom avea

$$truth\_val_{\Sigma_I}([p(t_1, \dots, t_n) \leftarrow B_1, \dots, B_k], F\{x_1|d_1, \dots, x_s|d_s\}) = true$$

ceea ce ne conduce la următoarea evaluare

$$truth\_val_{\Sigma_I}([p(t_1, \dots, t_n) \vee \dots \vee \neg B_1 \vee \dots \vee \neg B_k], F\{x_1|d_1, \dots, x_s|d_s\}) =$$
$$= true$$

Din condiţia $\{B_1\theta, \dots, B_k\theta\} \subseteq I$ deducem $truth\_val_{\Sigma_I}(B_i, F\theta) = true$ pentru fiecare $i \in \{1, \dots, k\}$, deci $truth\_val_{\Sigma_I}(p(t_1, \dots, t_n), F\theta) = true$ sau echivalent, $p(c_1, \dots, c_n) \in I$.

Reciproc, să considerăm că $T_P(I) \subseteq I$ şi să verificăm că $\Sigma_I = (UH_P, k_I)$ este un model pentru programul P. Fie

$$\forall x_1 \dots \forall x_s [p(t_1, \dots, t_n) \leftarrow B_1, \dots, B_k]$$

o formulă oarecare a programului P. Trebuie să verificăm că

$$truth\_val_{\Sigma_I}(p(t_1, \dots, t_n) \vee \neg B_1 \vee \dots \vee \neg B_k, F\{x_1|d_1, \dots, x_s|d_s\}) = true \quad (5)$$

pentru orice $d_1, \dots, d_s \in UH_P$.

Considerăm elementele $d_1, \dots, d_s \in UH_P$. Fie $\theta = \{x_1|d_1, \dots, x_s|d_s\}$. Notăm $p(c_1, \dots, c_n) = p(t_1, \dots, t_n)\theta$. Avem de analizat două cazuri:

1) Există $i \in \{1, \dots, k\}$ astfel încât $B_i\theta \notin I$.

În acest caz

$$truth\_val_{\Sigma_I}(B_i, F\{x_1|d_1, \dots, x_s|d_s\}) = false$$

deci

$$truth\_val_{\Sigma_I}(\neg B_i, F\{x_1|d_1, \ldots, x_s|d_s\}) = true$$

şi prin urmare (5) este adevărată.

2) Pentru orice $i \in \{1, \ldots, k\}$ avem $B_i\theta \in I$.

În acest caz

$$p(c_1, \ldots, c_n) \in T_P(I)$$

deci

$$p(c_1, \ldots, c_n) \in I$$

deoarece $T_P(I) \subseteq I$. Rezultă că

$$truth\_val_{\Sigma_I}(p(t_1, \ldots, t_n), F\{x_1|d_1, \ldots, x_s|d_s\}) = true$$

şi prin urmare (5) este adevărată.

■

Propoziţia 4.11 *Operatorul T_P ataşat unui program Horn este un operator continuu.*

Proof. Trebuie să verificăm că $supT_P(X) = T_P(supX)$ pentru orice mulţime dirijată X. Fie $X \subseteq L$, X dirijată. Atomul $p(c_1, \ldots, c_n)$ este în $T_P(supX)$ dacă şi numai dacă există o clauză $p(t_1, \ldots, t_n) \leftarrow B_1, \ldots, B_k$ în programul P şi există o substituţie ground θ astfel încât

$$\{B_1\theta, \ldots, B_k\theta\} \subseteq supX$$

şi $p(t_1, \ldots, t_n)\theta = p(c_1, \ldots, c_n)$. Dar X este o mulţime dirijată şi prin urmare condiţia $\{B_1\theta, \ldots, B_k\theta\} \subseteq supX$ este echivalentă cu faptul că există $I \in X$ astfel încât $\{B_1\theta, \ldots, B_k\theta\} \subseteq I$. În consecinţă, $p(c_1, \ldots, c_n) \in T_P(supX)$ dacă şi numai dacă există $I \in X$ astfel încât $p(c_1, \ldots, c_n) \in T_P(I)$. Evident, ultima condiţie este echivalentă cu faptul că

$$p(c_1, \ldots, c_n) \in supT_P(X)$$

■

Aplicând propoziţia 4.9 rezultă că există cel mai mic punct fix al operatorului T_P şi se justifică definiţia semanticii de punct fix dată în definiţia de mai jos:

Definiţia 4.7 *Fie P un program Horn şi T_P operatorul ataşat lui P ca în definiţia 4.6.* **Semantica de punct fix** *a programului P este notată cu $LFS(P)$ şi este definită prin*

$$LFS(P) = lfp(T_P) = T_P \uparrow \omega$$

În consecinţă, în raport cu această semantică orice atom din $LFS(P)$ are valoarea logică **true** *şi orice atom din $BH_P \setminus LFS(P)$ are valoarea logică* **false**.

Conform acestei definiţii putem spune că semantica de punct fix a unui program Horn este dată de cel mai mic punct fix al operatorului ataşat programului.

Acum putem demonstra că semantica de punct fix a unui program Horn coincide cu semantica celui mai mic model.

Teorema 4.2 *(Van Emden, Kowalski)*
Dacă P este un program Horn atunci $LMS(P) = LFS(P)$.

Proof. Fie T_P operatorul ataşat programului P. Conform teoremei lui Knaster- Tarski avem:

$$lfp(T_P) = inf\{I \mid I \in 2^{BH_P}, T_P(I) \subseteq I\}$$

Aplicând propoziţia 4.10 vom avea

$$lfp(T_P) = inf\{I \in 2^{BH_P} \mid I \in MH(P)\} = \bigcap_{W \in MH(P)} W$$

deci $LFS(P) = LMS(P)$. ■

5 TEME

TEMA 1

Consideraţi programul Horn P de mai jos:

$$\begin{cases} int(0) \leftarrow \\ int(add(x,1)) \leftarrow int(x) \end{cases}$$

unde $\mathcal{S}_C = \{0,1\}$, $\mathcal{S}_F = \{add^{(2)}\}$, $\mathcal{S}_P = \{int^{(1)}\}$.

a) Calculaţi $LMS(P)$

b) Arătaţi că $LMS(P)$ este mulţime infinită

TEMA 2

Fie $X = \{p_1, \ldots, p_n\}$, unde $p_1, \ldots, p_n$ sunt numere prime. Fiecărei submulţimi $A \subseteq X$ i se asociază un număr natural n_A astfel: dacă $A = \{p_{i_1}, \ldots, p_{i_s}\} \neq \emptyset$ atunci $n_A = p_{i_1} \ldots p_{i_s}$; dacă $A = \emptyset$ atunci $n_A = 1$. Pe mulţimea $M = \{n_A \mid A \in 2^X\}$ definim două legi de compoziţie:

$$n_A \cup n_B = c.m.m.d.c.(n_A, n_B)$$

$$n_A \cap n_B = c.m.m.m.c.(n_A, n_B)$$

Arătaţi că $(M, \cup, \cap)$ este o latice.

TEMA 3

1) Calculaţi semantica de punct fix pentru programul:

$$\begin{cases} p(a) \leftarrow \\ p(b) \leftarrow \\ q(f(x)) \leftarrow p(x) \end{cases}$$

2) Calculaţi semantica de punct fix pentru programul:

$$\begin{cases} p(a, b) \leftarrow \\ p(c, b) \leftarrow \\ p(x, z) \leftarrow p(x, y), p(y, z) \\ p(x, y) \leftarrow p(y, x) \end{cases}$$

TEMA 4

1) Calculaţi semantica de punct fix pentru programul:

$$\begin{cases} p(a) \leftarrow p(x), q(x) \\ p(f(x)) \leftarrow p(x) \\ p(b) \leftarrow \\ q(b) \leftarrow \\ q(f(x)) \leftarrow q(x) \end{cases}$$

2) Calculaţi semantica de punct fix pentru programul:

$$\begin{cases} p(c,b) \leftarrow \\ p(a,c) \leftarrow \\ p(x,z) \leftarrow p(x,y), p(y,z) \\ p(x,a) \leftarrow p(a,x), p(x,b) \end{cases}$$

TEMA 5

1) Calculaţi semantica de punct fix pentru programul:

$$\begin{cases} p(a,b) \leftarrow \\ p(b,c) \leftarrow \\ p(x,z) \leftarrow p(x,a), p(a,z) \\ p(x,y) \leftarrow p(y,x) \end{cases}$$

2) Calculaţi semantica de punct fix pentru programul:

$$\begin{cases} p(a,b) \leftarrow \\ p(x,z) \leftarrow p(x,b), p(b,z) \\ p(x,y) \leftarrow p(y,x) \end{cases}$$

3) Calculaţi semantica de punct fix pentru programul:

$$\begin{cases} p(c,c) \leftarrow \\ p(a,c) \leftarrow \\ p(x,b) \leftarrow p(x,y), p(y,y) \end{cases}$$