BAZE DE CUNOŞTINŢE

Sinteza 2-4:

Programe logice stratifiabile.

Semantica de model perfect.

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 conceptelor de program stratifiabil şi local stratifiabil
- însuşirea conceptului de *model perfect* şi semantica de model perfect

2 Programe normale stratifiabile

Vom considera în acest paragraf programe logice normale. În asemenea programe subscopurile unei clauze pot fi literali negativi, adică o clauză este de forma

$$A \leftarrow B_1, \ldots, B_k, \neg C_1, \ldots, \neg C_m$$

Considerarea negaţiei în corpul unei clauze aduce modificări importante cu privire la semantica acestor programe. De exemplu, se ştie că intersecţia tuturor modelelor Herbrand ale unui program Horn este de asemenea un model al programului respectiv. Programele normale nu se mai bucură de această proprietate, după cum se poate constata în următorul exemplu. Considerăm programul P de mai jos:

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

Putem uşor calcula modelele Herbrand minimale ale programului P dacă observăm că prima regulă este echivalentă cu formula $\forall x[p(x) \vee q(x)]$. Deoarece $UH_P = \{a, b\}$, se constată că P admite următoarele patru modele Herbrand minimale:

$$M_1 = \{r(a), r(b), p(a), p(b)\},\ M_2 = \{r(a), r(b), q(a), q(b)\}$$

$$M_3 = \{r(a), r(b), p(a), q(b)\}, M_4 = \{r(a), r(b), p(b), q(a)\}$$

Intersecţia acestor patru modele este mulţimea $\{r(a), r(b)\}$ şi în consecinţă intersecţia tuturor modelelor Herbrand ale programului P este aceeaşi mulţime, care nu este model pentru P. Astfel, programul P admite modele Herbrand minimale, dar nu admite cel mai mic model Herbrand.

Unul din conceptele importante ale programelor normale este *stratifiabilitatea* acestora, concept prezentat în definiţia care urmează. Reamintim că am notat cu BI mulţimea simbolurilor de predicate built-in.

Definiţia 2.1 *Fie P un program normal. P se numeşte* **stratifiabil** *dacă există o partiţie $\{P_0, \ldots, P_k\}$ a mulţimii $\mathcal{S}_P \setminus BI$ astfel încât pentru fiecare clauză $r \in P$:*

- *dacă $p \in \mathcal{S}_P \setminus BI$ apare în concluzia lui r şi $q \in \mathcal{S}_P \setminus BI$ apare negat în corpul lui r atunci*

$$p \in P_i,\ q \in P_j \Longrightarrow i > j$$

- *dacă* $p \in \mathcal{S}_P \setminus BI$ *apare în concluzia lui* r *şi* $q \in \mathcal{S}_P \setminus BI$ *apare pozitiv în corpul lui* r *atunci*

$$p \in P_i,\ q \in P_j \Longrightarrow i \geq j$$

Elementele $P_0, \ldots, P_k$ se numesc **straturile** programului P. Intuitiv vom considera că aceste straturi sunt aşezate unele peste altele astfel încât P_0 este primul strat, P_1 este al doilea strat şi el este aşezat deasupra lui P_0 şi în general stratul P_{i+1} este aşezat imediat deasupra stratului P_i. Adoptând această ierarhie de straturi vedem că definiţia de mai sus poate fi citită şi astfel:

- dacă $\neg q$ apare în corpul unei clauze şi p este simbolul de predicat care defineşte atomul din concluzia clauzei atunci q trebuie să apară într-un strat strict inferior stratului lui p

- dacă q apare în corpul unei clauze şi p este simbolul de predicat care defineşte atomul din concluzia clauzei atunci q poate să apară fie într-un strat strict inferior stratului lui p, fie în acelaşi strat cu p

Partiţia $\{P_0, \ldots, P_k\}$ se mai numeşte **stratifiere** a programului P. Evident, dacă P este stratifiabil atunci el poate admite mai multe stratifieri după cum se poate constata în exemplul care urmează.

Exemplul 2.1 *Considerăm programul P:*

$$\begin{cases} p(x,y) \leftarrow q(x), \neg r(y), u(x,y), v(y,y) \\ r(x) \leftarrow u(x,x), \neg v(x,x) \\ u(a,a) \leftarrow \\ u(b,a) \leftarrow \\ v(a,a) \leftarrow \\ r(a) \leftarrow \\ q(b) \leftarrow \end{cases}$$

Considerăm straturile $P_0 = \{u, v, q\}$, $P_1 = \{r\}$, $P_2 = \{p\}$. *Mulţimea* $\{P_0, P_1, P_2\}$ *este o stratificare a programului* P. *Evident* $P_0 = \{u, v\}$, $P_1 = \{r\}$, $P_2 = \{p, q\}$ *ne oferă o altă stratificare pentru acelaşi program.*

Există programe normale care nu admit stratificări. De exemplu, programul

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

nu admite stratificări. Explicaţia constă în faptul că fiecare din simbolurile p, q trebuie să se găsească într-un strat superior celuilalt simbol, după cum se poate constata din primele două clauze ale programului.

În cele ce urmează vom prezenta o condiţie necesară şi suficientă pentru ca un program normal să fie stratificabil.

Definiţia 2.2 *Fie P un program normal şi $\mathcal{S}_P$ mulţimea simbolurilor de predicate care apar în P.* **Graful de dependenţă** *ataşat programului P este graful direcţionat etichetat $G_P = (V, E)$ unde mulţimea nodurilor este $V = \mathcal{S}_P \setminus BI$, iar arcele sunt definite astfel: $(q, p) \in E$ dacă şi numai dacă q apare simplu sau negat în corpul unei clauze, iar p apare în concluzia clauzei respective. Dacă în corpul clauzei apare q pozitiv atunci arcul este etichetat cu $+$; dacă în corp apare $\neg q$ atunci arcul (q, p) este etichetat cu $-$.*

Propoziţia 2.1 *Un program normal P este stratifiabil dacă şi numai dacă graful de dependen- ţă G_P nu conţine cicluri care să aibă arce negative.*

Demonstraţie.

Presupunem că G_P conţine un ciclu de la p la p care să conţină cel puţin un arc negativ. Notăm cu $(a_1, \ldots, a_n)$ acest ciclu, deci $a_1 = a_n = p$. Conform presupunerii există $i \in \{2, \ldots, n-1\}$ astfel încât arcul (a_{i-1}, a_i) să fie un arc negativ. Prin absurd presupunem că P este un program stratifiabil. Notăm cu P_{j_1}, ..., $P_{j_{n-1}}$ straturile la care aparţin respectiv elementele $a_1, \ldots, a_{n-1}$. Din definiţia stratifiabilităţii rezultă că $j_1 \leq \ldots \leq j_{i-1} < j_i \leq \ldots \leq j_1$, ceea ce este fals. Aceasta ne arată că presupunerea făcută nu este adevărată şi prin urmare P nu este stratifiabil. Să presupunem acum că G_P nu conţine cicluri cu arce negative. Avem de analizat două cazuri:

1) Cazul I: G_P nu conţine cicluri.

În acest caz se poate obţine o stratifiere dacă se aplică următorul algoritm:

- $i := 0$;
- notez cu P_i mulţimea nodurilor în care nu intră nici un arc; ştergem din graful G_P toate nodurile din P_i împreună cu arcele care ies din ele
- $i := i + 1$; repetăm pasul anterior

2) Cazul II: G_P conţine cicluri.

Orice arc din G_P este un arc pozitiv. Strângem toate nodurile unui ciclu maximal s într-o mulţime A_s şi înlocuim ciclul s cu un nod ataşat lui A_s. Obţinem în acest fel un graf fără cicluri, căruia îi aplicăm algoritmul de la cazul I. ■

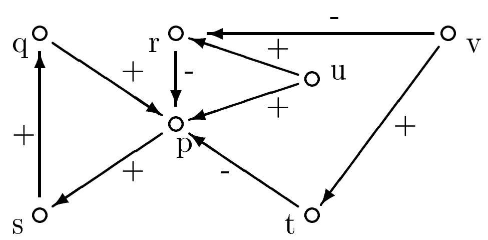

Figura 1: Graf de dependenţă

Ca exemplu de aplicare a acestui algoritm considerăm următorul program:

$$\begin{cases} p(x,y) \leftarrow q(x), \neg r(y), u(x,y) \\ r(x) \leftarrow u(x,x), \neg v(x,x) \\ q(x) \leftarrow s(x) \\ s(x) \leftarrow p(x,a) \\ p(x,a) \leftarrow \neg t(x) \\ u(a,a) \leftarrow \\ u(b,a) \leftarrow \\ v(a,a) \leftarrow \\ r(a) \leftarrow \\ q(b) \leftarrow \\ t(b) \leftarrow v(a,a) \end{cases}$$

Graful de dependenţă ataşat acestui program este desenat în Figura 1. Aplicând algoritmul prezentat mai sus obţinem următoarele straturi:

$P_0 = \{u, v\}$

$P_1 = \{r, t\}$

$P_2 = \{p, q, s\}$

3 Modele perfecte

În acest paragraf vom considera programe logice care pot să conţină clauze Horn, clauze Horn generale şi *clauze disjunctive restricţionate*, adică clauze de forma:

$$A_1 \vee \ldots \vee A_n \leftarrow B_1, \ldots, B_m, \neg C_1, \ldots, \neg C_k$$

unde $n > 0$, $m \geq 0$ şi $k \geq 0$. Programele logice care conţin cel puţin o asemenea clauză se numesc *programe disjunctive restricţionate*. Denumirea provine de la faptul că:

– în concluzia clauzei este prezentă disjuncţia şi de aici provine denumirea de *clauză disjunctivă*

– în concluzia clauzei apar numai literali pozitivi şi de aceea clauza se numeţe *restricţionată*

Dacă dăm voie ca în concluzia clauzei să apară şi literali negativi atunci clauza se numeşte *clauză disjunctivă*, iar un program care conţine o asemenea clauză se numeşte *program disjunctiv*. Un program Horn sau

Horn general poate fi privit ca un caz particular de program disjunctiv restricţionat.

Pe baza Herbrand a unui program disjunctiv restricţionat se definesc două relaţii binare, care au implicaţii majore în studiul semanticii acestuia.

Definiţia 3.1 *Fie P un program disjunctiv restricţionat şi BH_P baza sa Herbrand. Notăm cu $<_P$ şi $\leq_P$ cele mai mici relaţii binare pe BH_P, care satisfac următoarele proprietăţi:*

- $A <_P B$ *dacă există o instanţă ground $r\theta$ a unei clauze $r \in P$ care conţine pe $\neg B$ în corpul ei şi pe A în concluzia clauzei instanţiate*
- $A \leq_P B$ *dacă există o instanţă ground $r\theta$ a unei clauze $r \in P$ care conţine pe B în corp şi pe A în concluzie sau A şi B se găsesc în concluzia clauzei $r\theta$*
- *dacă $A \leq_P B$ şi $B \leq_P C$ atunci $A \leq_P C$*
- *dacă $A \leq_P B$ şi $B <_P C$ sau $A <_P B$ şi $B \leq_P C$ atunci $A <_P C$*
- *dacă $A <_P B$ atunci $A \leq_P B$*

Din punct de vedere intuitiv putem gândi aceste relaţii ca *relaţii de prioritate* între elementele bazei Herbrand. Astfel, prin scrierea $A <_P B$ înţelegem că prioritatea atomului B este mai mare strict decât prioritatea atomului A. Cu alte cuvinte, în calculul valorii de adevăr a atomilor A şi B, atomul B are prioritate mai mare decât atomul A.

Observăm că dacă aplicăm definiţia de mai sus unui program Horn P atunci relaţia $<_P$ este vidă deoarece un asemenea program nu conţine literali negativi. În consecinţă, numai relaţia $\leq_P$ are sens pentru un program Horn. Aceeaşi afirmaţie este adevărată pentru un program disjunctiv restricţionat care nu conţine literali negativi.

Pentru simplificarea scrierii vom utiliza următoarea convenţie: în loc de simbolul $<_P$ sau $\leq_P$ vom utiliza $<$, respectiv $\leq$ atunci când nu există nici un pericol de confuzie. Evident cele două relaţii depind de programul P şi două programe diferite P şi Q cu aceeaşi bază pot genera relaţii $<_P$ şi $<_Q$ diferite; aceeaşi observaţie este adevărată pentru $\leq_P$ şi $\leq_Q$.

Remarcă 3.1 *Este uşor de observat că relaţia $<$ este tranzitivă. Într-adevăr, dacă $A < B$ şi $B < C$ atunci $A \leq B$ şi $B < C$, deci $A < C$.*

Exemplul 3.1 *Considerăm următorul program disjunctiv restricţionat P:*

$$\begin{cases} p(x) \vee q(x) \leftarrow r(x), \neg s(x) \\ s(x) \leftarrow \neg r(x) \\ r(a) \leftarrow \\ r(b) \leftarrow \end{cases}$$

Baza Herbrand este mulţimea

$$BH_P = \{p(a), p(b), q(a), q(b), r(a), r(b), s(a), s(b)\}$$

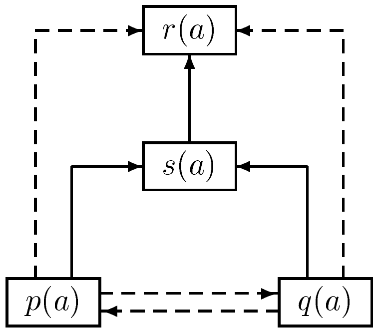

Figura 2: Relaţii de bază $<_P$ şi $\leq_P$

Direct din program se obţine diagrama din Figura 2, unde cu săgeată întreruptă am notat relaţia $\leq$*, iar cu săgeată simplă am desemnat relaţia* $<$.

Rezultă următoarele relaţii de bază:

$$p(a) \leq q(a),\ q(a) \leq p(a),\ p(a) \leq r(a)$$

$$q(a) \leq r(a),\ p(a) < s(a),\ q(a) < s(a),\ s(a) < r(a)$$

Aplicând proprietăţile de legătură dintre $<$ *şi* $\leq$ *obţinem:*

$$p(a) \leq p(a),\ q(a) \leq q(a),\ p(a) \leq s(a),\ q(a) \leq s(a)$$

$$s(a) \leq r(a),\ p(a) < r(a),\ q(a) < r(a)$$

Evident obţinem relaţiile corespunzătoare pentru $p(b)$ *şi* $q(b)$*:*

$$p(b) \leq q(b),\ q(b) \leq p(b),\ p(b) \leq r(b)$$

$q(b) \leq r(b),\ p(b) < s(b),\ q(b) < s(b),\ s(b) < r(b)$

$p(b) \leq p(b),\ q(b) \leq q(b),\ p(b) \leq s(b),\ q(b) \leq s(b)$

$s(b) \leq r(b),\ p(b) < r(b),\ q(b) < r(b)$

Definiţia 3.2 *Fie P un program disjunctiv restricţionat, M_1 şi M_2 două modele Herbrand distincte ale lui P. Spunem că modelul M_1* **este preferabil** *lui M_2 şi scriem $M_1 << M_2$ dacă pentru fiecare atom $A \in M_1 \setminus M_2$ există un atom $B \in M_2 \setminus M_1$ astfel încât $A < B$.*

Dacă scriem prima clauză a programului din **Exemplul 3.1** sub forma

$$\forall x[p(x) \vee q(x) \vee s(x) \vee \neg r(x)]$$

atunci este uşor de constatat că $M_1^0 = \{r(a), r(b), p(a), q(b)\}$ şi $M_2^0 = \{r(a), r(b), s(a), s(b)\}$ sunt modele Herbrand pentru programul P şi $M_1^0 << M_2^0$, dar nu avem $M_2^0 << M_1^0$. Într-adevăr, avem:

$M_1^0 \setminus M_2^0 = \{p(a), q(b)\}$

$M_2^0 \setminus M_1^0 = \{s(a), s(b)\}$

$p(a) < s(a)$, $q(b) < s(b)$

Observăm de asemenea că deşi $s(a) < r(a)$ şi $s(b) < r(b)$, totuşi relaţia $M_2^0 << M_1^0$ nu are loc deoarece nu avem nici una din relaţiile $s(a) < p(a)$, $s(a) < q(b)$, $s(b) < p(a)$, $s(b) < q(b)$. Datorită simetriei raţionamentului în a şi b, observăm că $M_3^0 = \{r(a), r(b), q(a), p(b)\}$ este de asemenea un model pentru programul P şi $M_3^0 << M_2^0$. Prin urmare *există mai multe modele preferabile unui model dat.*

Definiţia 3.3 *Fie P un program disjunctiv restricţionat. Un model Herbrand M al lui P se numeşte* **model perfect** *dacă nu există nici un model Herbrand M' al lui P astfel încât $M' << M$.*

Revenind la **Exemplul 3.1** constatăm că M_1^0 este un model perfect deoarece orice model Herbrand al lui P trebuie să conţină pe $r(a)$ şi $r(b)$ şi nu există nici un atom $x \in BH_P$ astfel încât $x < p(a)$ şi $x < q(b)$. La fel M_3^0 este un model perfect. Acest exemplu ne arată că *există programe disjunctive restricţionate care admit mai multe modele perfecte.*

4 Proprietăţi ale modelelor perfecte

Vom prezenta acum câteva proprietăţi ale modelelor perfecte.

Propoziţia 4.1 *Fie P un program disjunctiv restricţionat, $M_1, M_2 \in MH(P)$. Dacă $M_1 \subset M_2$ atunci $M_1 << M_2$. În particular orice model perfect este minimal.*

Demonstraţie. Vom observa la început că dacă M_1 şi M_2 sunt modele Herbrand ale unui program disjunctiv restricţionat P şi $M_1 \subset M_2$ atunci $M_1 << M_2$. Într-adevăr, dacă $M_1 \subset M_2$ atunci $M_1 \setminus M_2 = \emptyset$ şi deci condiţia ca $M_1 << M_2$ este satisfăcută.

Fie acum M un model perfect pentru P. Dacă prin absurd presupun că M nu este minimal atunci există un model Herbrand M' al lui P astfel încât $M' \subset M$. În conformitate cu proprietatea demonstrată mai sus vom avea $M' << M$, ceea ce înseamnă că M nu este model perfect. ■

Definiţia 4.1 *Fie P un program disjunctiv restricţionat. Notăm cu $PERF(P)$ mulţimea tuturor modelelor Herbrand pentru P care sunt modele perfecte.*

Propoziţia 4.2 *Fie P un program disjunctiv pozitiv, $M_1, M_2 \in MH(P)$. Avem $M_1 \subset M_2$ dacă şi numai dacă $M_1 << M_2$. În particular $M \in PERF(P)$ dacă şi numai dacă M este minimal.*

Demonstraţie. Implicaţia de la stânga la dreapta se obţine din **Propoziţia 4.1**. Să presupunem acum că $M_1 << M_2$. Avem două cazuri: ori $M_1 \subset M_2$, ori $M_1 \setminus M_2 \neq \emptyset$. Situaţia $M_1 \subset M_2$ este cea prezentată în enunţul propoziţiei. Să considerăm a doua situaţie. Dacă $A \in M_1 \setminus M_2$ atunci din condiţia $M_1 << M_2$ rezultă că există $B \in M_2 \setminus M_1$ astfel încât $A < B$. Acest lucru este imposibil deoarece programul P fiind un program disjunctiv pozitiv, relaţia $<$ este vidă. ■

Am arătat în paragraful anterior că există programe disjunctive restricţionate care admit mai multe modele perfecte. Vom arăta acum că *există programe disjunctive restricţionate care nu admit nici un model* *perfect*. Pentru aceasta vom considera programul P de mai jos:

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

Avem $BH_P = \{p(a), q(a), r(a)\}$. Evident $M_1 = \{p(a), r(a)\}$ şi $M_2 = \{q(a), r(a)\}$ sunt modele Herbrand minimale pentru P. Avem $p(a) < q(a)$ şi $q(a) < p(a)$ deci $M_1 << M_2$ şi $M_2 << M_1$. Aşadar nici unul din modelele M_1, M_2 nu este model perfect. Deoarece orice model perfect este minimal şi nu există alte modele minimale pentru programul P, rezultă că P nu admite modele perfecte.

5 Programe disjunctive restricţionate stratifiabile. Semantica de model perfect.

În acest paragraf vom defini stratifiabilitatea programelor disjunctive restricţionate şi vom caracteriza proprietatea de stratifiabilitate prin intermediul unor relaţii binare.

Definiţia 5.1 *Fie P un program disjunctiv restricţionat şi $B = (\mathcal{S}_C, \mathcal{S}_F, \mathcal{S}_P)$ baza sa. Relaţia $\prec_P \subseteq \mathcal{S}_P \times \mathcal{S}_P$ este definită astfel: $p \prec_P q$ dacă există $p(a), q(b) \in BH_P$ astfel încât $p(a) <_P q(b)$*

Facem observaţia că ori de câte ori nu există pericol de confuzie vomm nota $\prec$ în loc de $\prec_P$.

Definiţia 5.2 *O relaţie binară* $\rho \subseteq X \times X$ *se numeşte* **noetheriană** *dacă nu există o secvenţă infinită* $a_1, a_2, \ldots$ *din* X *astfel încât* $a_1 \rho a_2 \rho \ldots$

Astfel, pentru programul:

$$\begin{cases} p(x) \vee q(x) \leftarrow \neg r(x) \\ r(a) \leftarrow \neg p(a) \end{cases}$$

avem $p(a) < r(a)$ şi $r(a) < p(a)$ deci $p(a) < p(a)$. În consecinţă există secvenţa $p(a) < p(a) < p(a) < \ldots$, deci relaţia $<$ definită de acest program pe baza Herbrand nu este o relaţie noetheriană.

Propoziţia 5.1 *Relaţia* $\prec$ *este noetheriană dacă şi numai dacă este ireflexivă.*

Demonstraţie. Să presupunem că relaţia $\prec$ este ireflexivă şi prin absurd presupunem că ea nu este noetheriană. Există o secvenţă infinită $p_1 \prec p_2 precp_3 < \ldots$ de simboluri de predicate. Deoarece $\mathcal{S}_P$ este o mulţime finită, rezultă că există i şi j cu $i \neq j$ astfel încât $a_i = p$ şi $a_j = p$. Aşadar $p \prec p$ şi prin urmare relaţia $\prec$ nu ar fi ireflexivă.
Reciproc, dacă relaţia $\prec$ nu ar fi ireflexivă ar rezulta că există $p \in \mathcal{S}_P$ cu $p \prec p$. În acest caz am avea $p \prec p \prec p \prec \ldots$ şi prin urmare relaţia $\prec$ nu ar fi noetheriană. ∎

Propoziţia 5.2 *Dacă relaţia $\prec$ este noetheriană atunci relaţia $<$ este noetheriană.*

Demonstraţie. Presupunem că relaţia $\prec$ este noeteriană. Să presupunem prin absurd că relaţia $<$ nu este noetheriană. Există o secvenţă $a_1 < a_2 < a_3 < \ldots$ de elemente ale bazei Herbrand. Deoarece mulţimea simbolurilor de predicate este finită, există un simbol de predicat p şi există i, j cu $i \neq j$ astfel încât $a_i = p(a)$, $a_j = p(b)$ pentru două elemente a şi b ale universului Herbrand. Aşadar avem $p \prec p$, deci există secvenţa $p \prec p \prec p \prec \ldots$ şi prin urmare relaţia $\prec$ nu este noetheriană. ■

Vom introduce acum conceptul de stratifiabilitate şi local stratifiabilitate pentru programe disjunctive restricţionate.

Definiţia 5.3 *Fie P un program disjunctiv restricţionat. Programul P se numeşte* **stratifiabil** *dacă există o partiţie $\{P_0, \ldots P_k\}$ a mulţimii $\mathcal{S}_P \backslash BI$ astfel încât pentru orice clauză din P*

$$p_1(X_1) \vee \ldots \vee p_s(X_s) \leftarrow q_1(Y_1), \ldots, q_n(Y_n), \neg r_1(Z_1), \ldots, \neg r_m(Z_m)$$

unde X_i, Y_j, Z_k reprezintă notaţii vectoriale, există $h \in \{0, \ldots, k\}$ astfel încât să fie îndeplinite următoarele condiţii:

1) $\{p_1, \ldots, p_s\} \subseteq P_h$

2) $\{q_1, \ldots, q_n\} \subseteq \bigcup_{j \leq h} P_j$

3) $\{r_1, \ldots, r_m\} \subseteq \bigcup_{j < h} P_j$

Se observă imediat că acest concept este o extensie a conceptului de stratifiabilitate a unui program normal (cazul $s = 1$).

Definiţia 5.4 *Fie P un program disjunctiv restricţionat. Programul P se numeşte* **local stratifiabil** *dacă există o partiţie $\{H_0, H_1, \ldots\}$ a bazei Herbrand BH_P astfel încât pentru orice instanţă ground*

$$A_1 \vee \ldots \vee A_p \leftarrow B_1, \ldots, B_m, \neg C_1, \ldots, \neg C_n$$

a unei clauze din P există k astfel încât următoarele condiţii să fie îndeplinite:

$\{A_1, \ldots, A_p\} \subseteq H_k$

$\{B_1, \ldots, B_m\} \subseteq \bigcup_{j \le k} H_j$

$\{C_1, \ldots, C_n\} \subseteq \bigcup_{j < k} H_k$

Propoziţia 5.3 *Orice program disjunctiv restricţionat stratifiabil este local stratifiabil.*

Demonstraţie. Fie $\{P_0, \ldots, P_k\}$ o stratifiere a programului P. Pentru fiecare $j \in \{0, \ldots, k\}$ definim mulţimile:

$$H_j = \{p(c_1, \ldots, c_n) \in BH_P \mid p \in P_j\}$$

Atunci, $\{H_0, \ldots, H_k\}$ formează o partiţie a lui BH_P în raport cu care P este local stratifiabil. ∎

În cele ce urmează vom nota cu PDR_s familia programelor disjunctive restricţionate stratifiabile şi cu PDR_{ls} familia programelor disjunctive restricţionate local stratifiabile.

Vom caracteriza acum programele acestor două familii cu ajutorul relaţiilor binare introduse în paragrafele anterioare. Deoarece ne interesează în mod special familia PDR_{ls}, următorul rezultat referitor la familia PDR_s este prezentat fără demonstraţie.

Teorema 5.1 *$P \in PDR_s$ dacă şi numai dacă relaţia $\prec_P$ este noetheriană.*

Referitor la familia PDR_{ls} avem următoarea caracterizare:

Teorema 5.2 *$P \in PDR_{ls}$ dacă şi numai dacă relaţia $<_P$ este noetheriană.*

Demonstraţie. Presupunem că $P \in PDR_{ls}$ şi fie $\{H_0, H_1, H_2, \ldots\}$ o stratificare locală a lui P. Prin absurd să presupunem că relaţia $<$ nu este noetheriană. Există o secvenţă $a_0 < a_1 < a_2 < \ldots$ de elemente din BH_P. Fie $i < j$. Deoarece $a_i < a_j$ rezultă că există o instanţă ground de forma

$$\ldots \vee a_i \vee \ldots \leftarrow \ldots, \neg a_j, \ldots$$

a unei clauze din P. Din definiţia local stratifiabilităţii rezultă că există k_i astfel încât $a_i \in H_{k_i}$ şi $a_j \in \bigcup_{t<k_i} H_t$. Aşadar există $k_j < k_i$ astfel încât $a_j \in H_{k_j}$. Dacă aplicăm această observaţie pentru secvenţa $a_0 < a_1 < \ldots$

de la care am plecat rezultă că există următorul şir descrescător de numere naturale $k_0 > k_1 > k_2 > \ldots$, ceea ce este imposibil.

Reciproc, să presupunem că relaţia $<$ este noetheriană. Următorul algoritm calculează o stratificare locală:

$i := 0$; $X := BH_P$

REPEAT

$H_i := \{A \in X \mid \nexists B : B \in X, A < B\}$

$X := X \setminus H_i$

$i := i + 1$

UNTIL $X = \emptyset$

Observăm că la fiecare pas avem $H_i \neq \emptyset$ deoarece relaţia $<$ este noetheriană. ∎

Teoremele 5.1 şi 5.2 au aplicaţii interesante. În primul rând, prin **Propoziţia 5.2** rezultă din nou incluziunea $PDR_s \subseteq PDR_{ls}$ demonstrată în **Propoziţia 5.3**. În al doilea rând, se poate uşor dovedi că pentru programul P de mai jos:

$$\begin{cases} nr\_par(s(x)) \leftarrow \neg nr\_par(x) \\ nr\_par(0) \leftarrow \end{cases}$$

relaţia $\prec_P$ nu este noetheriană, iar relaţia $<$ este noeteriană. Aceasta ne conduce la concluzia că $P \in PDR_{ls} \setminus PDR_s$. Aşadar avem incluziunea strictă $PDR_s \subset PDR_{ls}$.

Un rezultat deosebit de interesant este prezentat în teorema care urmează, pe care o prezentăm fără demonstraţie.

Teorema 5.3 *Orice program local stratifiabil are cel puţin un model perfect. Orice program general Horn local stratifiabil admite un singur model perfect.*

Această teoremă ne arată importanţa conceptului de program local stratifiabil: pentru orice program disjunctiv restricţionat local stratifiabil P avem $PERF(P) \neq \emptyset$ şi în consecinţă definiţia care urmează este justificată.

Definiţia 5.5 *Fie* $P \in PDR_{ls}$. **Semantica de model perfect** *a lui* P *este*

$$PMS(P) = \bigcap_{M \in PERF(P)} M$$

În raport cu această semantică un atom $A \in BH_P$ *este* **true** *dacă şi numai dacă* $A \in PMS(P)$.

Să observăm că $PMS(P)$ nu este neapărat un model Herbrand pentru P. Într-adevăr, să considerăm următorul program P:

$$\left\{ \; p(a) \vee p(b) \leftarrow \right.$$

Prin **Propoziţia 4.2** mulţimile $M_1 = \{p(a)\}$ şi $M_2 = \{p(b)\}$ sunt modele perfecte pentru programul P. Aplicând definiţia semanticii de model perfect avem $PMS(P) = \emptyset$, care nu este model al lui P.

6 Tratarea operatorială a semanticii de model perfect pentru programe normale stratifiabile

În acest paragraf vom arăta cum putem calcula semantica de model perfect pentru programe normale stratifiabile cu ajutorul puterilor de operatori. Pentru a realiza acest scop vom extinde definiţia operatorului ataşat unui program Horn pentru cazul programelor Horn generale.

Definiţia 6.1 *Considerăm un program Horn general P şi baza sa Herbrand BH_P. 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*

$$p(t_1, \ldots, t_n) \leftarrow B_1, \ldots, B_k, \neg C_1, \ldots, \neg C_l$$

în P şi există o substituţie ground θ astfel încât $B_1\theta \in I$, ..., $B_k\theta \in I$, $C_1\theta \notin I$, ..., $C_l\theta \notin I$ şi $p(t_1, \ldots, t_n)\theta = p(c_1, \ldots, c_n)$.

Să observăm de la început faptul că operatorul T_P nu se mai bucură de proprietatea de monotonie. Într-adevăr, să considerăm programul P de mai jos:

$$\begin{cases} p(a) \leftarrow \neg q(b) \\ q(a) \leftarrow \neg r(a) \end{cases}$$

Se observă că dacă luăm $I_1 = \emptyset$, $I_2 = \{q(b)\}$ atunci $T_P(I_1) = \{p(a), q(a)\}$ şi $T_P(I_2) = \{q(a)\}$, deci operatorul T_P nu este monoton.

Definiţia 6.2 *Iteraţiile operatorului T_P sunt notate cu $T_P \uparrow n$ pentru $n \in N \cup \{\omega\}$; ele sunt operatori $T_P \uparrow n : 2^{BH_P} \longrightarrow 2^{BH_P}$ şi se definesc astfel:*

$$T_P \uparrow 0(I) = I$$

$$T_P \uparrow (n+1)(I) = T_P(T_P \uparrow n(I)) \cup T_P \uparrow n(I)$$

$$T_P \uparrow \omega(I) = \bigcup_{n \geq 0} T_P \uparrow n(I)$$

Fie Π un program normal stratifiabil şi $\{P_0, \ldots, P_k\}$ o stratifiere a acestuia. Programul Π se poate descompune în programele $\{\Pi_0, \ldots, \Pi_k\}$, unde pentru fiecare $i \in \{0, \ldots, k\}$ programul Π_i este format din toate clauzele lui Π cu proprietatea că atomii concluziilor au simbolurile de predicate în P_i.

Teorema 6.1 *Fie Π un program general Horn stratifiabil şi $\{\Pi_1, \ldots, \Pi_n\}$*

o stratifiere a acestuia. Notăm:

$$\begin{cases} M_1 = T_{\Pi_1} \uparrow \omega(\emptyset) \\ M_2 = T_{\Pi_2} \uparrow \omega(M_1) \\ \dots\dots\dots\dots\dots\dots\dots \\ M_n = T_{\Pi_n} \uparrow \omega(M_{n-1}) \end{cases}$$

Atunci M_n este unicul model perfect al lui P. În consecinţă avem $PMS(\Pi) = M_n$

Să observăm că dacă aplicăm teorema de mai sus pentru un program Horn Π, atunci $PMS(\Pi) = LMS(\Pi)$ deoarece stratifierea programului are un singur strat.

7 TEME

TEMA 1

1) Consideraţi programul normal de mai jos:

$$\begin{cases} p \leftarrow r, \neg s \\ q \leftarrow r, \neg s \\ r \leftarrow q \\ p \leftarrow \neg r \\ s \leftarrow t \end{cases}$$

Construiţi graful de dependenţă a acestui program. Arătaţi că programul este stratifiabil şi aplicaţi algoritmul din **Propoziţia 2.1** pentru a obţine o stratifiere a acestuia.

2) Scrieţi un program în SWI Prolog pentru a testa dacă un program normal este stratifiabil.

TEMA 2

Consideraţi programul normal P de mai jos:

$$\begin{cases} int(0) \leftarrow \\ int(succ(x)) \leftarrow int(x) \\ odd(succ(x)) \leftarrow int(x), \neg odd(x) \end{cases}$$

1) Verificaţi că P nu este stratifiabil

2) Aflaţi modelele Herbrand minimale ale lui P. Există cel mai mic model Herbrand?

TEMA 3

1) Consideraţi următorul program disjunctiv restricţionat P:

$$\begin{cases} p(x) \vee q(x) \leftarrow \neg u(x), \neg v(x) \\ u(x) \leftarrow r(x) \\ v(x) \leftarrow r(x) \\ r(a) \leftarrow \\ r(b) \leftarrow \end{cases}$$

a) Calculaţi relaţiile $<_P, \leq_P, \prec_P$

b) Arătaţi că relaţia $\prec_P$ este noetheriană

2) Consideraţi următorul program disjunctiv restricţionat P:

$$\begin{cases} p(x) \vee q(x) \leftarrow \neg u(x), \neg v(x) \\ u(x) \leftarrow r(x) \\ v(x) \leftarrow r(x) \\ r(a) \leftarrow \\ r(b) \leftarrow \end{cases}$$

a) Arătaţi că programul P este stratifiabil şi găsiţi o stratifiere a lui P

b) Calculaţi $PMS(P)$

TEMA 4

Calculaţi $PMS(P)$ pentru programul normal P de mai jos:

$$\begin{cases} p(x) \leftarrow \neg q(x), \neg u(x), \neg v(x) \\ u(x) \leftarrow r(x) \\ v(x) \leftarrow r(x) \\ r(a) \leftarrow \\ r(b) \leftarrow \end{cases}$$

TEMA 5

Consideraţi următoarea piesă de cunoştinţe:

Un conducător auto este prudent dacă este asigurat şi nu a cauzat accidente. Dacă un conducător este prudent atunci el plăteşte o primă mică de asigurare. Dacă o persoană plăteşte primă mică de asigurare atunci ea este un conducător prudent. Dacă o persoană este asigurată şi *nu este un conducător prudent atunci ea plăteşte o primă de asigurare mare. Ana şi Petre au asigurări auto. Petre a cauzat accident*

1) Reprezentaţi această piesă sub forma unui program logic normal.

2) Arătaţi că programul obţinut este stratifiabil şi calculaţi o stratificare a acestuia.

3) Calculaţi toate modelele Herbrand minimale ale programului.

4) Reprezentaţi aceeaşi piesă de cunoştinţe sub forma unui program disjunctiv restricţionat.

5) Calculaţi pentru programul disjunctiv restricţionat obţinut relaţiile $<$, $\leq$ şi $\prec$.

6) Studiaţi stratifiabilitatea locală a programului

7) Calculaţi modelele perfecte

8) Calculaţi valoarea de adevăr pentru diferite elemente din baza Herbrand în raport cu semantica de model perfect.

TEMA 6

1) Consideraţi programul Horn general P de mai jos:

$$\begin{cases} p(a) \leftarrow \\ p(b) \leftarrow \\ q(c) \leftarrow \\ u(b) \leftarrow \\ r(x) \leftarrow p(x), \neg q(x) \\ r(x) \leftarrow u(x) \end{cases}$$

Calculaţi $PMS(P)$.

2) Consideraţi programul Horn general P de mai jos:

$$\begin{cases} lung(x,y,1) \leftarrow arc(x,y) \\ lung(x,y,s(k)) \leftarrow lung(x,z,k), arc(z,y), \neg lung(x,y,k) \\ arc(a,b) \leftarrow \\ arc(b,c) \leftarrow \\ arc(a,e) \leftarrow \\ arc(e,f) \leftarrow \\ arc(f,d) \leftarrow \end{cases}$$

1) Arătaţi că $lung \prec_P lung$

2) Studiaţi stratifiabilitatea lui P

3) Arătaţi că următoarele mulţimi formează o stratifiere locală a lui P:

$$H_0 = \{arc(a,b) \mid arc(a,b) \in BH_P\}$$

$$H_k = \{lung(x,y,k) \mid lung(x,y,k) \in BH_P\},\ k \geq 1$$