УДК 681. 3:622. 276

ДОСЛІДЖЕННЯ СЕМАНТИКИ МОДИФІКАЦІЙНИХ ПРЕДИКАТНИХ ЗАПИТІВ ДЛЯ ІНФОРМАЦІЙНИХ СИСТЕМ НА ОСНОВІ БАЗ ДАНИХ І ЗНАНЬ НАФТОГАЗОВОЇ ПРЕДМЕТНОЇ ОБЛАСТІ

В.І. Шекета
ІФНТУНГ, кафедра прикладної математики, sheketa@mail.ru

Статья посвящена исследованию вопроса о соотношении формально-логического аппарата модификационных предикатных запросов с классическими теориями логического программирования. В результате введены определения модификационных предикатных запросов для информационных систем на основе баз данных и знаний нефтегазовой предметной области на основе подхода, построенного на семантике стабильных моделей.

Given paper is dealing with an exploration of parity between the formal - logic apparatus of modification predicate queries and classical theories of logic programming. As a result definitions of modification predicate queries for information systems on the basis of databases and knowledge bases of an oil-and-gas subject domain are given , from the point of view of an approach constructed on stable models semantics .

В даний час нафтогазова галузь має автоматизовані системи різних рівнів і типів, а також різні програмні комплекси для прогнозу нафтогазоносності, пошуку та розвідки родовищ нафти і газу, але масштаби та ефективність застосування ЕОМ для кожного рівня різні [1]. Це пов'язано з тим, що недостатньо використовуються розвинені бази даних і відсутня типова методика набуття і накопичення знань про нафтогазоносні поклади та прогнозування необхідності проведення пошуково-розвідувальних робіт з використанням нових інформаційних технологій.

Впровадження нових інформаційних технологій в нафтогазовій справі дає можливість суттєво змінити економіко-екологічний підхід до пошуково-розвідувальних робіт на нафту і газ . Оскільки основними джерелами інформації при таких пошуково-розвідувальних роботах є буріння свердловин, які вимагають значних капітальних витрат, важливим є накопичення баз знань, що можуть бути використані при прогнозуванні нафтогазонасичених покладів з використанням експертних систем(ЕС). Експертні системи, як системи штучного інтелекту, використовують знання висококваліфікованих спеціалістів-експертів для розв'язуванння задач в порівняно вузьких проблемних областях. Проблемна спрямованість і можливість вирішувати широкий клас задач зробила експертні системи незамінними в багатьох галузях промисловості і науки . Сьогодні на перший план виходять проблеми, пов'язані з практичною реалізацією ЕС в великих і складних проблемних областях, зокрема нафтогазовій предметній області [1] .

Основною перешкодою на шляху створення і поширення експертних систем в нафтогазовій промисловості України є проблема набуття і представлення знань . Значна кількість інформації, її слабка структурованість, неуміння експерта пояснити послідовність своїх міркувань щодо прийняття певного рішення призводять до того, що процес передачі знань від експерта до системи здійснюється з великими труднощами, методом «спроб, помилок і виправлень» . Існуючі методики стосовно роботи з експертом є далеко неповними . Слабким місцем є їх нездатність контролювати процес прийняття рішень експертом і підтримка в експерта постійного інтересу до процесу передачі знань . В результаті процес передачі знань перетворюється в абсолютно нецікаве і монотонне заняття для експерта, а процес формування бази знань розтягується в часі. З другого боку, при представленні вже набутих знань виникають труднощі щодо вибору правильного способу представлення і відповідних інструментальних засобів. В нафтогазовій предметній області часто виникає потреба в комбінованому застосуванні вже існуючих способів представлення знань, а також у впровадженні нових специфічних прийомів. І, нарешті, ще більш складною проблемою є процес логічного висновку на основі інформації нафтогазової предметної області. Вузьким місцем цієї проблеми є необхідність здійснювати процес логічного висновку в умовах неповної і неточної інформації, що має місце при прогнозуванні нафтогазоносних покладів [1].

Таким чином, існує реальна потреба в розробці нових, більш ефективних методів набуття знань, що враховують всю сукупність проблем, які виникають в процесі передачі знань і стимулюють творчу активність експерта, а також використання нових підходів з представлення набутих знань і здійснення логічного висновку на основі неповної і нечіткої інформації нафтогазової предметної області.

Одним із таких методів є введена в роботі [2] ідея використання модифікаційних предикатних запитів як ефективного механізму підтримки діалогу з користувачем в інформаційних системах на основі баз даних і знань нафтогазової предметної області. Зокрема, розглядаються запити типу

Q_M={K_{B+}(насиченість_породи>1)<<,K_{B+}(порода_колектор)<< K_{B+}(насиченість_породи>1),K_{B-}(пористість_породи>1) }

Загалом ідея побудови семантики [3,4] модифікаційних предикатних запитів для їх стабільних моделей [5] не вкладається в рамки парадигм мов логічного програмування [6]. Класична парадигма мов логічного програмування передбачає наявність для кожної логічної програми тільки однієї моделі в той час як ідея семантики стабільних моделей передбачає побудову для кожної логічної програми скінченої множини моделей, причому така множина може бути і порожньою. Вирішення даної проблеми розглядається в двох підходах [7,8,9]. Перший підхід полягає в звуженні класу програм, для яких будуються моделі, і послабленні семантичної структури відображення [10-13], що має своїм наслідком накладання синтаксичних обмежень на процедури логічного висновку. В результаті для кожної програми, що пройшла процедуру накладання обмежень, виконується побудова єдиної так званої ідеальної моделі. Суть другого підходу [3] полягає в побудові так званої обґрунтованої моделі. При такому підході множина всіх предикатів логічної програми розбивається на такі групи: обґрунтовані атоми (тобто такі, які мають відповідність в моделі), необґрунтовані атоми (такі, що не мають відповідності) і решта атомів, що не потрапляють в жодну групу. Результатом успішного застосування таких підходів є побудова ефективних механізмів обробки даних в інформаційних системах.

Таким чином, недослідженим залишається питання про співвідношення введеного формально-логічного апарату модифікаційних предикатних запитів із класичними теоріями логічного програмування, зокрема, що стосується питання побудови семантики для стабільних моделей модифікаційних предикатних запитів як інструменту набуття нових знань нафтогазової предметної області.

Тому **ціллю даної статті** є обґрунтування модифікаційних предикатних запитів з точки зору основних ідей , що лежать в основі логічного програмування [10].

Розглянемо деяку множину O. Позначимо множину всіх підмножин O через $S(O)$. Розглядатимемо оператор λ на множині $S(O)$ як функцію λ: $S(O)\rightarrow S(O)$. Нехай λ - оператор, заданий на множині $S(O)$. Множину $O_f \leq O$ будемо називати фіксованим значенням для оператора λ, якщо $\lambda(O_f)=O_f$. Найменше фіксоване значення (якщо воно існує) для λ позначатимемо через $min_f(\lambda)$. Якщо існує найбільше фіксоване значення для λ, то позначатимемо його через $max_f(\lambda)$. Оператор λ є монотонним, якщо для будь-яких двох підмножин O_1, O_2

$$\forall O_1,O_2,\ O_1,O_2 \leq O,\ O_1 \leq O_2 => \lambda(O_1) \leq \lambda(O_2).$$

Теорема 1. (Теорема Хастера [10]).

Якщо λ - монотонний оператор, то λ має найменше і найбільше фіксоване значення. Оператор λ - є антимонотонним, якщо

$$\forall O_1,O_2 \leq O,\ O_1 \leq O_2 => \lambda(O_2) \leq \lambda(O_1).$$

Теорема 2 [10] Якщо λ - є антимонотонним оператором, тоді:

1) λ^2 є монотонним;
2) $\lambda(min_f(\lambda^2))= max_f(\lambda^2)$ і $\lambda(max_f(\lambda^2)= min_f(\lambda^2))$;
3) для будь-якого фіксованого значення O_f оператора λ, $min_f(\lambda^2) \leq O_f \leq max_f(\lambda^2)$.

Якщо O є скінченим, тоді для будь-якого антимонотонного оператора λ ми можемо обчислити найменше і найбільше фіксоване значення оператора λ^2 , виконуючи оператор λ скінчене число разів. Власне нам треба обчислити $\lambda^0(\varnothing)=\varnothing$, $\lambda^n(\varnothing)=\lambda(\lambda^{n-1}(\varnothing))$ для $n=1,2,\ldots$ доки не знайдемо n, таке, що $\lambda^n(\varnothing)=\lambda^{n+2}(\varnothing)$. Тоді можна стверджувати, що одне із $\lambda^n(\varnothing)$, $\lambda^{n+1}(\varnothing)$ є найменшим фіксованим значенням для λ^2 , а друге є найбільшим фіксованим значенням, залежно від того, чи n є парним чи непарним.

Під алфавітом Δ_A будемо розуміти скінчену, або зліченну множину констант, змінних, предикатів і функцій. Означення терма на множині Δ_A дано рекурсивно через змінну, константу або вираз

$F_\Delta(T_{\Delta_1},\ldots,T_{\Delta_n})$, де F_Δ - функція на множині Δ_A, T_{Δ_i} - і-й терм для Δ_A. Вирази виду $\Pi(T_{\Delta_1},\ldots,T_{\Delta_n})$, де П – предикат на Δ_A, а T_{Δ_i} - і-й терм називатимемо атомами для Δ_A. Терм вважатимемо базовим, якщо він не містить змінних. Множину базових термів для Δ_A будемо називати базисом Гербранда [10] для Δ_A. Аналогічно атом будемо вважати базовим, якщо він не містить змінних. Множину базових атомів для Δ_A будемо називати А-базисом Гербранда [10].

Означення 1. Твердженням Горна [7,8] називається вираз одного із типів

$$S_A \leftarrow R_{A_1},\ldots,R_{A_l}, \quad (l \geq 0)$$

або

$$\leftarrow R_{A_1},\ldots,R_{A_l}, \quad (l \geq 1)$$

де $R_A, S_{A_1},\ldots,S_{A_l}$ є атомами для Δ_A.

Вирази першого типу називають твердженнями програми, а вирази другого типу – цілями програми. Твердження $S_A \leftarrow (l=0)$ вважатимемо фактом. Теорія Горна – це множина тверджень Горна. Горн-програма – множина програмних тверджень. Множину атомів E_M будемо називати моделлю твердження $S_A \leftarrow R_{A_1},\ldots,R_{A_l}$, якщо $S_A \in E_M$ для всіх $R_{A_1},\ldots,R_{A_l} \in E_M$. Множину атомів E_M називатимемо моделлю цілі $\leftarrow R_{A_1},\ldots,R_{A_l}$, якщо $\{R_{A_1},\ldots,R_{A_l}\} \leq E_M$. Множина атомів є моделлю теорії Горна T_H, якщо E_M задовольняє кожне твердження Горна (тобто є його моделлю) в T_H.

Теорія Горна не обов'язково повинна мати модель [8]. Але кожна Горн-програма завжди має модель, що ми й покажемо в наступному означенні:

Означення 2. Нехай Δ_H – Горн-програма. Тоді відповідний оператор λ_{Δ_H} на множині атомів можна означити таким чином: $S_A \in \lambda_{\Delta_H}(O_1)$ тоді і тільки тоді, коли твердження $(S_A \leftarrow R_{A_1},...,R_{A_l}) \in \Delta_H$ є таким, що $R_{A_1},...,R_{A_l} \in O_1$.

Оператор λ_{Δ_H} є монотонним. Згідно з теоремою Хастера він має найменше фіксоване значення. Фактично $\min_f(\lambda_{\Delta_H})$ є найменшою моделлю для Δ_H. Позначимо найменшу модель для Горн-програми Δ_H через $min(\Delta_H)$. Наступна лема показує, що найменша модель для Горн-програми не змінюється, якщо ми додамо до програми правила, які задовольняються найменшою моделлю.

Лема 1. Нехай E_M найменша модель для Горн-програми Δ_H і модель для Горн-програми Δ'_H. Тоді E_M є найменшою моделлю для $\Delta_H \cup \Delta'_H$.

Теорію Горна [7,8,10] можна розбити на дві підмножини T_{H_z} - підмножина цілей і T_{H_c} - підмножина програмних тверджень.

Лема 2. Нехай T_H – теорія Горна, і $T_H = T_{H_z} \cup T_{H_c}$ декомпозиція на підмножини цілей і програмних тверджень. Тоді:

1) E_M є моделлю для T_H тоді і тільки тоді, коли E_M є моделлю для T_{H_c} і задовольняє всі цілі з T_{H_z};

2) T_H має модель тоді і тільки тоді, коли $\min(T_{H_c})$ задовольняє всі цілі із T_{H_z}.

Нехай O – скінчена або зліченна множина, елементи якої називатимемо атомами.

Означення 3. Твердженням логічної програми вважатимемо вираз

$S_A \leftarrow R_{A_1},...,R_{A_l},(V_{A_1}),...,(V_{A_m})$, де $S_A, R_{A_1},...,R_{A_l}, V_{A_1},...,V_{A_m}$ - атоми.

Для множини атомів $X \leq O$ введемо позначення факту виключення елемента із множини через $\{a \in A\}$.

Означення 4. Логічна програма є множиною тверджень логічної програми.

Множина $E_M \leq O$ є моделлю (або будемо говорити, що E_M задовольняє) твердження логічної програми , якщо $\{R_{A_1},...,R_{A_l}\} \leq E_M$ і $\{V_{A_1},...,V_{A_m}\} \cap E_M = \varnothing \Rightarrow S_A \in E_M$. Множина E_M є моделлю (або задовольняє) логічну програму Δ_L, якщо E_M є моделлю для кожного твердження із Δ_L.

Ідея семантики стабільних моделей була введена в роботі [3].

Означення 5. Нехай Δ_L – логічна програма. Залишком Δ_L відносно E_M, $\Delta_L^{E_M}$ є множина, отримана із Δ_L в результаті виконання наступних дій:

1) видалення всіх тверджень, які містять "$\overline{r}$", такі що r є істинним в E_M;

2) видалення всіх заперечувальних передпосилок "$\overline{r}$" (для всіх $r \in O$) з усіх тверджень, що залишились в результаті дії пункту 1.

$\Delta_L^{E_M}$ є Горн-програмою. Вона має модель $min(\Delta_L^{E_M})$.

Означення 6. E_M є стабільною моделлю для Δ_L, якщо $E_M = min(\Delta_L^{E_M})$

Теорема 3. [13] Якщо E_M є стабільною моделлю для Δ_L, тоді E_M є також звичайною моделлю для Δ_L.

Наведені нами теореми будуть використані для розвитку формально-логічного апарату модифікаційних предикатних запитів.

Означення 7. Для довільної програми Δ означимо оператор λ_{1_Δ} згідно з рівнянням

$$\lambda_{1_\Delta}(O_1) = \min(\Delta^{O_1}).$$

Оператор λ_{1_Δ} є антимонотонним. Згідно з означенням 1 оператор $\lambda_{1_\Delta}^2$ є монотонним і має найменше і найбільше фіксоване значення.

Означення 8. Атоми, що належать до найменшого фіксованого значення для $\lambda_{1_\Delta}^2$ є обґрунтованими відносно Δ. Атоми, що не належать до найбільшого фіксованого значення $\lambda_{1_\Delta}^2$, є необґрунтованим відносно Δ. Решту атомів співвідносять як такі, що мають значення "невизначено".

Теорема 4. [4,9] Для довільної стабільної моделі логічної програми Δ_L має місце:

1) Стабільна модель E_M^S включає всі атоми, що є обґрунтованими відносно Δ_L.

2) Не включає атоми, що є необґрунтованими відносно Δ_L.

Теорема 5. [4,9] Якщо кожний атом є або обґрунтованим або необґрунтованим відносно логічної програми Δ_L, множина обгрунтованих атомів є єдиною стабільною моделлю E_M^S для Δ_L.

Нехай O – деяка скінчена множина, елементи якої будемо називати атомами. Розглядатимемо деяку множину $\{K_{B_1},...,K_{B_l}\}$ баз знань, таких, що $K_{B_i} \subseteq O$, де $1 \leq i \leq l$. Вирази виду $K_B(o)$, K_{B}-(o), де $o \in O$ деякий атом, будемо називати модифікаційними літералами. Модифікаційні літери $K_B()$, K_B-$()$ розглядатимемо як взаємообернені і зарезервуємо для їх позначення в загальному випадку послідовність грецьких букв, починаючи з τ: τ, υ, ϖ. Обернений модифікаційний літерал для τ позначатимемо через τ^{-1}.

Для довільної множини атомів $X \subseteq O$, означимо

$$X^{pp} = \{K_{B+}(o) : o \in X\} \cup \{K_{B-}(o), o \notin X\}.$$

Ми можемо говорити про X^{pp} як про повне представлення для X, якщо для кожного атома $o \in O$, X^{pp} показує, чи O належить X чи не належить.

Тепер для довільної множини модифікаційних літералів M_L означимо

$$M_L^{in} = \{o \in O : K_{B+}(o) \in M_L\},$$
$$M_L^{out} = \{o \in O : K_{B-}(o) \in M_L\}.$$

Означення 9. Множина модифікаційних літералів M_L є когерентною, якщо вона не містить пари взаємообернених літералів, тобто, що $M_L^{in} \cap M_L^{out} = \varnothing$.

Нехай K_B – база знань. Означимо множину модифікаційних літералів M_L так:

$$K_B \circ M_L = (K_B \setminus M_L^{out}) \cup M_L^{in}.$$

Виходячи із означення 9, можемо стверджувати, що якщо M_L є когерентним, то

$$(K_B \setminus M_L^{out}) \cup M_L^{in} = (K_B \cup M_L^{in}) \setminus M_L^{out}.$$

Означення 10. Модифікаційне правило є виразом одного із двох видів

$$K_{B+}(o) << K_{B+}(o_1),\dots,K_{B+}(o_l), K_{B-}(p_1),\dots,K_{B-}(p_m). \qquad (1)$$

$$K_{B-}(o) << K_{B+}(o_1),\dots,K_{B+}(o_l), K_{B-}(p_1),\dots,K_{B-}(p_m), \qquad (2)$$

де o, o_i, $p_i \in O$.

Модифікаційні предикатні правила мають декларативну інтерпретацію в термінах обмежень, що накладаються на базу знань. Зокрема, правило накладає на базу знань таку умову:

$$\forall o \in K_B$$
$$(\exists o_i, 1 \le i \le l, o_i \notin K_B) \cup (\exists p_j, 1 \le j \le m, p_j \in K_B).$$

Модифікаційні предикатні правила можна також інтерпретувати в термінах виконання обчислень, що втілюють накладені обмеження. Введемо наступну інтерпретацію: нехай атоми $o_i, 1 \le i \le l$ належать поточній базі знань K_B, і жодний із атомів $p_j, 1 \le j \le m$ не належить K_B. Тоді, щоби втілити обмеження атом O повинен бути доданий до бази знань (або вилучений із неї), чи певний атом o_i видалений із неї, чи деякий атом p_i доданий.

Звернемося тепер до інтерпретації, прийнятої в теорії Prolog-програмування [11,12]. В Prolog в кожному правилі виділяють заголовок правила head() і тіло правила body(). Тоді, якщо модифікаційне предикатне правило P_{Q_m} є типу (1), то

$$head(P_{Q_m}) \Leftrightarrow K_{B+}(o)$$

а у випадку, коли P_{Q_m} належать до типу (2)

$$head(P_{Q_m}) \Leftrightarrow K_{B-}(o) \text{ і}$$

$$body(P_{Q_m}) = \{K_{B+}(o_1),\dots,K_{B+}(o_1), K_{B-}(p_1),\dots,K_{B-}(p_m)\}.$$

Означення 11. Модифікаційний предикатний запит Q_m будемо розглядати, як набір модифікаційних предикатних правил $\{P_{Q_m}\}$, де $1 \le i \le l$.

Означення 12. Множина атомів $E_M \le O$ є моделлю (або задовольняє) модифікаційний літерал $K_{B+}(o)$ (або відповідно $K_{B-}(o)$), якщо $o \in E_M$ (відповідно $o \notin E_M$). Множина атомів E_M є моделлю (або задовольняє) модифікаційне правило P_{Q_m}, якщо E_M є моделлю $head(P_{Q_m})$. Множина атомів E_M є моделлю (або задовольняє) модифікаційний предикатний запит Q_m, якщо E_M є моделлю кожного модифікаційного правила в Q_m.

Проілюструємо введений формально-логічний апарат на наступному прикладі.

Приклад 1. Нехай алфавіт O_A складається з двох констант $\{true, false\}$, однієї змінної $\{Y\}$ і двох унарних предикатів $\{\textit{колектор}(), \textit{продуктивний\_колектор}()\}$. Нехай модифікаційний предикатний запит Q_M задано у вигляді

$$K_{B+}(\textit{продуктивний\_колектор}(Y)) << K_{B+}(\textit{колектор}(Y))$$

$$K_{B-}(\textit{колектор}(Y)) << K_{B+}(\textit{продуктивний\_колектор}(Y)).$$

Нехай

$$K_{B_1} = \{(\textit{колектор}(Y), (\textit{продуктивний\_колектор}(Y)\}.$$

Якщо $\{true, false\}$ приймемо в якості універсуму Гербранда, то тоді базисом Гербранда буде

$$\{\textit{колектор}(true), \textit{колектор}(false),$$
$$\textit{продуктивний\_колектор}(true), \textit{продуктивний\_колектор}(true)\}$$

Екземпляром Гербранда для Q_M буде:

$$\{K_{B-}(\textit{продуктивний\_колектор}(true)) << K_{B+}(\textit{колектор}(true))$$
$$K_{B-}(\textit{продуктивний\_колектор}(false)) << K_{B+}(\textit{колектор}(false))$$
$$K_{B-}(\textit{колектор}(true)) << K_{B+}(\textit{продуктивний\_колектор}(true))$$
$$K_{B-}(\textit{колектор}(false)) << K_{B+}(\textit{продуктивний\_колектор}(false))\}.$$

А екземпляром Гербранда для K_{B_1} буде

$$\{\textit{колектор}(true), \textit{колектор}(false),$$
$$\textit{продуктивний\_колектор}(true), \textit{продуктивний\_колектор}(true)\}$$

Таким чином, Q_M - модифікацією для K_{B_1} будуть

$$K_{B_2}^1 = \{\textit{колектор}(true), \textit{продуктивний\_колектор}(false)\}$$

$$K_{B_2}^2 = \{\textit{продуктивний\_колектор}(true), \textit{колектор}(false)\}.$$

Введені означення є обґрунтованими, оскільки при їх побудові ми не виходили за рамки процедури обчислення обґрунтованих семантик прийнятих в теорії абстрактного логічного програмування і стабільних семантик для логічних програм Горна.

Таким чином, введений формально-логічний апарат закладає основу для виконання порівняльного аналізу основних ідей логічного програмування і теорії модифікаційних предикатних запитів і їх застосування до процедур одержання знань в інформаційних системах на основі баз даних і знань нафтогазової предметної області.

Висновки: В даній статті введено означення модифікаційних предикатних запитів для інформаційних систем на основі баз даних і знань нафтогазової предметної області на основі підходу побудованого на семантиці стабільних моделей. Подальші розвідки даного напряму будуть напрямлені на дослідження властивостей модифікаційних запитів і побудови відображення множини логічних програм на множину модифікаційних предикатних запитів в рамках конкретних процедур набуття знань нафтогазової предметної області.

Література

1. Шекета В.І. Інформаційна система для прогнозування нафтогазоносних покладів: *Дисертація на здобуття наукового ступеня кандидата технічних наук.* - Херсон,1999.-130с.
2. Шекета В.І. Модифікаційні предикатні запити як інструмент підтримки діалогу з користувачем в інформаційних системах на основі баз даних і знань // *Вісник Тернопільського державного технічного університету. Серія: Математичне моделювання.* - 2003. -Том 8-№4 -2003- С.113-119.
3. K. Berman, J. Schlipf, and J.Franco. Computing the well-founded semantics faster. In *Logic Programming and Nonmonotonic Reasoning (Lexington, KY, 1995)*, volume 928 of *Lecture Notes in Computer Science*, pages 113–125, Berlin, 1995. Springer.

I. Niemel¨ a and P. Simons. Efficient implementation of the well-founded and stable model semantics. In M.J. Maher, editor, *Proceedings of the 1996 Joint International Conference and Syposium on Logic Programming (JICSLP-96) (Bonn, Germany, September 2-6, 1996)*, pages 289–303. MIT Press, 1996.

4. . Syrjanen. Implementation of local grounding for logic programs with stable model semantics.Technical Report B18, Digital Systems Laboratory, Helsinki University of Technology, October 1998.
5. J.J. Alferes, J.A. Leite, L.M. Pereira, H. Przymusinska, and T.C. Przymusinski. Dynamic logic programming. In *Principles of Knowledge Representation and Reasoning. Proceedings of the 6th International Conference, KR'98, Trento, Italy, June 2-5, 1998*, pages 98–111.Morgan Kaufmann, 1998.
6. J.J. Alferes and L.M. Pereira. Update-programs can update programs. In *Non-Monotonic Extensions of Logic Programming (Bad Honnef, 1996)*, volume 1216 of *Lecture Notes in Computer Science*, pages 110–131, Berlin, 1997. Springer.
7. J.J. Alferes, L.M. Pereira, H. Przymusinska, and T.C. Przymusinski. LUPS – a language for updating logic programs. In *Logic Programming and Nonmonotonic Reasoning, 5th Interna-tional Conference, LPNMR'99*, volume 1730 of *Lecture Notes in Computer Science*, pages 162–176. Springer-Verlag, 1999.
8. T. Eiter and G. Gottlob. On the computational cost of disjunctive logic programming: propositional case. *Annals of Mathematics and Artificial Intelligence*, 15(3-4):289–323, 1995.
9. V. Lifschitz. Foundations of logic programming. In *Principles of Knowledge Representation*, pages 69–127. CSLI Publications, 1996.
10. V. Lifschitz, L. R. Tang, and H. Turner. Nested expressions in logic programs. *Annals of Mathematics and Artificial Intelligence*, 25(3-4):369–389, 1999.
11. T. C. Przymusinski and H. Turner. Update by means of inference rules. *Journal of Logic Programming*, 30(2):125–143, 1997.
12. C. Sakama and K. Inoue. Embedding circumscriptive theories in general disjunctive programs. In *Logic programming and nonmonotonic reasoning (Lexington, KY, 1995)*, volume 928 of *Lecture Notes in Computer Science*, pages 344–357, Berlin, 1995. Springer.