ІНТЕРПРЕТАЦІЯ ПРОЦЕСУ ОБРОБКИ ПРЕДИКАТНИХ ЗАПИТІВ МЕТОДОМ РОЗДІЛЕННЯ ЗМІННИХ

Василь Шекета, Тетяна Ваврик, Лідія Гобир

Івано-Франківський національний технічний університет нафти і газу
76019, м. Івано-Франківськ, вул. Карпатська 15, тел.:+38(03422) 4-21-27

Аналіз розділення змінних має справу із визначенням надмножини для множини пар змінних, які в даній точці виконання предикатного запиту, можуть бути прив'язані до двох термів, які розділяють деякі змінні. Аналіз розділення змінних є частковим випадком аналізу розділення множин [1,2,3,4,5,6], де не тільки досліджуються пари змінних, але і деякі абстрактні множини змінних. Теорія розділення змінних є корисною для побудови процедур дебаггінгу і автоматичної синхронізації предикатних запитів [1,5]. Як показано в роботі [7], інформація про попарне розділення змінних є необхідною для абстрактного аналізу запитів і виконання їх трансформації та модифікації. Аналіз зв'язності змінних [2,8,4,5,6] досліджує питання визначення підмножини змінних які гарантовано будуть зв'язані з деякими змінними в фіксованій точці виконання запиту. Аналіз зв'язності є корисним з точки зору оптимізації уніфікаційних процедур, перевпорядкування приоритетів цілей і уникнення багаторазової перевірки відповідності типів. Як показано в роботах [1,5] виконання аналізу розділення і зв'язності змінних в кон'юктивній зв'язці значно покращує точність аналізу обидвох властивостей. Серед виконаного нами аналізу робіт тільки в роботах [9] і [6] виконано побудову абстрактних доменів для аналізу розділення, які є прийнятними, і можуть бути використані для виконання абстрактної компіляції.

В роботі [10] база знань інформаційної системи розглядається як набір інформаційних сутностей атомарних предикатів з деякого скінченого інформаційного простору $\Re$. Всі зміни, що відбуваються в базі знань, розглядаються , як наслідок модифікаційних предикатних запитів Q_M, що генеруються інтелектуальною інформаційною системою відповідно до вказівок користувача. Основою самих запитів є набір модифікаційних предикатних правил. Розглядаються два типи правил:

$$Q_M \longleftrightarrow (K_B)^{<<} \Big\|_{K_{B\pm}(o\ )} <<,$$

де $o, o_i, p_i \in \Re$. Основна ідея такого запису правил полягає в тому, що $K_{B+}(o\ )$ означає, що атомарний предикат o повинен бути включений в базу знань K_B, а K_{B-} означає, що o – повинен бути виключений з бази знань, а $(K_B)^{<<}$ - означає модифікацію бази знань на рівні логічної зв'язаності предикатних правил, як наслідок виконання операцій додавання і вилучення правил в базовому домені виконання модифікації; $<<$ -дескриптор модифікації.

Недослідженим залишається питання введення механізму виконання розділення змінних модифікаційних предикатних запитів, як необхідної передумови виконання їх абстрактної інтерпретації чи компіляції.

Таким чином, **метою даної статті** є аналіз розділення змінних модифікаційних предикатних запитів, як результат лінійного уточнення базового домену модифікаційних предикатних запитів.

В загальному випадку аналіз розділення має справу із визначенням надмножини для множини програмних змінних, яка визначає розділення змінних в кожний конкретний момент виконання предикатного запиту.

Нехай задано множину змінних W і множину функцій M_F, причому кожній функцій присвоєна певна розмірність . Для $l \in N$ означимо множину термів T:

1. $\mathrm{T}^{\varnothing}\left(M_F, W\right) W$.
2. $\mathrm{T}^{l+1}\left(M_F, W\right) = \mathrm{T}^{l}\left(M_F, W\right) V \left\{ f\left(t_1, \ldots t_n\right) \middle| f^n \in M_F \left\{t_1, \ldots t_n\right\} \subseteq T^{l}\left(M_F, W\right) \right\}$.
3. $\mathrm{T}\left(M_F, W\right) = \bigcap_{c \geq 0} \mathrm{T}^{c}\left(M_F, W\right)$.

Ми виходимо з припущення, що M_F містить принаймні один символ арності $\varnothing$. Позначимо через $V\left(t\right)$ множину змінних, які зустрічаються в термі t. Якщо $V\left(t\right) = \varnothing$ будемо говорити, що терм t є базовим. Для заданої множини змінних W і конкретної змінної x будемо вважати, що $W \bigcup x$

означає $W \bigcup \{x\}$, а $W \setminus x$ означає $V \setminus \{x\}$.

Під підстановкою σ будемо розуміти відображення множини змінних на множину термів. Додатково означимо наступні множини:

1. $M_{dom}(\sigma) = \{x | \sigma(x) \neq x\}$.
2. $M_{rang}(\sigma) = \bigcup x \in M_{dom}(\sigma) V(\delta(x))$.

На множину $M_{dom}(\sigma)$ накладається вимога скінченності, що дозволить нам представляти підстановку σ як $\sigma\{W_1 \to t_1, \ldots W_n \to t_n\}$ за умови , що $\sigma(W_i) = t_i$ для всіх $i = 1, \ldots n$ і $\sigma(W) = W$ для кожного $w \in W \setminus \{w_i | i = 1, \ldots n\}$.

В загальному випадку, чим більша величина W, тим більш складним є обчислення абстрактних операцій $*^{\Gamma_W}$ і Δ^{Γ_W}. Це означає, що дуже важливо є утримання значення W настільки малим, наскільки це є можливим. Відмітимо, що для заданого $g \in \Gamma_W$, де Γ_W множина, що описується обмеженнями Гербранда, ми маємо, що $g \in \Gamma_{W'}$ для кожного $W' \geq W$.

Домен для аналізу розділення [1] оголошується як $r(r(W))$ для довільного $W \in r_f(W_1)$, де W_1 – множина програмних змінних, r_f –абстрактний оператор . Застосування аналізу розділення власне полягає в обробці інформації про пари змінних. В свою чергу, інформація про множини змінних завжди є достатньою для виконання попарного розділення змінних **[7].** Тому в даному дослідженні, ми будемо оперувати інформацією про попарне розділення, тобто виконувати обчислення надмножини множини пар змінних, розділення яких і виконується в даний конкретний момент виконання предикатного запиту. Це по своїй суті є аналогічним до обчислення підножини множини пар змінних, розділення яких гарантовано не виконується в заданий момент виконання предикатного запиту.

Означення 1. Непопарне розділення (NPR) [7] є однією із локальних властивостей змінних предикатних запитів. Означимо дану властивість наступним чином:

$npr = \left\langle \{npr_w\} w \in r_f(w_1), \left\{\{w_1, w_2\} \middle| \{w_1, w_2\} \subseteq W\right\} \right\rangle$, для кожного $\Delta_v h \in \Gamma_w$ і $\{w_1, w_2\} \subseteq W$. $npr(\Delta_v h, \{w_1, w_2\})$ є істинним тоді і тільки тоді, якщо $V_r(h(w_1)) \bigcap V_r(h(w_2)) = \varnothing$, де V -множина виділених змінних . Крім того, абстрактна ширина становить $\rho(npr) = 2$.

Означення 2. Для заданих $W \in r_f(W_1)$, означимо домен непопарного розділення змінних модифікаційних предикатних запитів $NPR_w = \Omega(Cr(\{npr\}))_w$, де $\Omega(\ )$ і $Cr(\ )$- відповідно представлення зовнішніх та внутрішніх зв'язків змінних.

Домен NPR_w є ізоморфним до домену для аналізу попарного розділення, оголошення якого виконано в роботі [7], як результат абстрагування ширшого домену для аналізу розділення множин [1].

Побудова оптимальної апроксимації операції конкретної кон'юнкції є досить складним завданням із точки зору точності. Причиною цього є те, що в загальному випадку домен для аналізу розділення не є достатньо точним для виконання розділення тих екзистенційних обмежень Гербранда, чия поведінка варіює по відношенню до операції конкретної кон'юнкції.

Проблема неточності була вирішена в роботах [7,1] за рахунок введення процедури гібридної кон'юнкції, яка виконує обчислення кон'юнкції абстрактного і конкретного елементів, що дозволяє відрізняти наприклад елементи c і C. Перевагою такого підходу є забезпечення більш точного результату для операції абстрактної кон'юнкції в порівняні із використання підходу на основі двох абстрактних елементів, де один елемент є абстракцією другого.

Для вирішення даної проблеми в нашому випадку, ми виконаємо лінійне уточнення домену NPR_w по відношенню до операції конкретної кон'юнкції.

Розглянемо уточнений домен $NPR_w^1 = NPR_w >>*^{r(\Gamma_w)} NPR_w$.

Твердження 1. Для заданого $W \in r_f(W_1)$ ми матимемо $NPR_w^1 \not\subset NPR_w >>*^{r(\Gamma_w)} NPR_w$.

Проте, незв'язаність змінної не може бути виражена в NPR_w^1. Це є джерелом неточності для NPR_w^1 і значить і для NPR_w також. Тому використаємо ідею запропоновану в роботі [5] про обчислення інформації про незв'язаність програмних змінних для того, щоби уточнити інформацію про розділення змінних модифікаційних предикатних запитів.

Аналіз незв'язаності оперує з проблемою визначення підмножини програмних змінних, які гарантовано будуть зв'язані із виділеними змінними в конкретний заданий момент виконання предикатного запиту. Найпростіший домен для аналізу незв'язаності співпадає з однойменною властивістю.

Означення 3. Для заданих $W \in r_f(W_1)$ означимо домен для аналізу незв'язаності змінних модифікаційних предикатних запитів, як $NZ_w = \Omega\left(Cr\left(\{nz\}\right)\right)_w$.

Виконаємо лінійне уточнення домену NZ_w. Розглянемо уточнений домен $NZ_w^1 = NZ_w >>^{*r(\Gamma_w)} NZ_w$.

Твердження 2. Для заданих $W \in r_f(W_1)$, матимемо $NZ_w \not\subset NZ_w >>^{*r(\Gamma_w)} NZ_w$.

Означення 4. Для заданих $W \in \Gamma_f(W_1)$ означимо:

$$NPR(NZ)_w = \Omega\left(Cr\left(\{npr, nz\}\right)\right)_w$$

$$NPR\left(NZ_w^1\right) = NPR\left(NZ_w\right) >>^{*r(\Gamma w)} NPR(NZ)_w.$$

Тепер введемо спосіб використання стрілок в новому домені .

Твердження 3. Для заданого $W \in \Gamma_f(W_1)$, матимемо, що

$$NPR(NZ)_w \subseteq NPR(NZ)_w >>^{*r(\Gamma w)} NPR(NZ)_w,$$

і як наслідок, що

$$NPR(NZ)_w \subseteq NPR\left(NZ_w^1\right) = NPR(NZ)_w >>^{*r(\Gamma w)} NPR(NZ)_w.$$

Твердження 4. Для заданого $W \in \Gamma_f(W_1)$, $NPR\left(NZ_w^1\right)$ містить інформацію про функтори і лінійність і більш точну інформацію про базовість чим $NPR(NZ)_w$.

Введемо представлення для домену $NPR\left(NZ_w^1\right)$ як $PR\left(\{npr, nz\}\right)_w$. Використовуватимемо для скорочення позначення PR_w для $PR\left(\{npr, nz\}\right)_w$. Введемо також дві виділених змінних O_1 і O_2.

Нехай задано $W \in r_f(W_1)$ і $\{L, L_1, L_2\} \subseteq PR_w$. Якщо $L_1 *^{PR_w} L_2 >> (L_1 \cup L_2) >> (L_2, L_1)$, де відображення $*^{r(\Gamma_w)}$ є комутативним, то тоді $\left\{(D_1, D_2) = \left\{a_1 \ldots a_k >> pr \,\middle|\, pr_1 \ldots pr_k >> pr \in D_2 \cup S\right\}\right.$ і $a_1 >> pr_i \in D_1 \cup S$. А значить також , що

$$S = \left\{(w_1, w_1) >> (w_2, w_2) \,\middle|\, \{w_1, w_2\} \subseteq W\right\} \cup \left\{(w, w) >> (w, w) \,\middle|\, w \in W\right\} \cup$$

$$\left\{w(w, w) >> (w_1, w_2) \,\middle|\, \{w, w_1, w_2\} \subseteq W\right\}\}.$$

Відмітимо, що деякі стрілки в S є тавтологічними, оскільки коли змінна не розділяє саму себе, тоді вона повинна бути базовою, і тому вона не може розділятися із жодною іншою змінною. Крім того, базова змінна не може втратити свою базовість в запиті, і базова змінна не може бути одночасно незв'язаною. Тому згідно синтаксису операцій розділення, обмеження, перейменування та уніфікації введених в [13] матимемо:

$$RZ_x^{PR_w}(L) = L \cup \left\{a\left[x / O_1\right] >> pr\left[x / O_1\right] \,\middle|\, a >> pr \in L\right\},$$

$$OB_x^{PR_w}(L) = \left\{a \setminus X >> pr \setminus X \,\middle|\, a >> pr \in L,\ pr \setminus X \neq 0 \ i\ (w, w) \notin a\right\},$$

де $X=\{x\}\cup\{(x,w)\mid w\in W\}$, $PN_{x\to k}^{PR_w}(L)=L[k/x]$.

$$\Delta_x^{PR_w}(L)=OB_x^{PR_{w\cup k}}\left(RZ_x^{w\cup k\setminus x}\left(PN_{x\to k}^{w}(L)\right)\right),$$

$$U^{PR_w}(L_1,L_2)=\{a_1a_2>>pr\mid a_1>>pr\in L_1:a_2>>pr\in L_2\}.$$

Висновки:

В даній статті виконано застосування формального апарату лінійного уточненення доменів до випадку абстрактного аналізу змінних модифікаційних предикатних запитів. В результаті одержано оголошення домену для аналізу розділення і зв'язаності змінних модифікаційних предикатних запитів. Одержаний домен є більш точним, чим стандартний домен [1,5] і буде використаний для виконання абстрактної інтерпретації модифікаційних предикатних запитів.

Темою подальших досліджень буде виконання абстрактної інтерпретації модифікаційних предикатних запитів в одержаному домені.

Література:

[1] Jacobs D., Langen A. Static Analysis of Logic Programs for Independent AND Parallelism. Journal of Logic Programming, 13(2 & 3).– 1992.— P.291–314.

[2] Bruynooghe M., Codish M. Freeness, Sharing, Linearity and Correctness .– All at Once. In proceedings of the 3rd International Workshop on Static Analysis, volume 724 of Lectures Notes in Computer Science, Springer-Verlag.– 1993. — P. 153–164.

[3] King A. A Synergistic Analysis for Sharing and Groundness which Traces Linearity. In D. Sannella, editor, Proc. of the 5th European Symp. on Programming, volume 788 of Lecture Notes in Computer Science, Edinburgh, UK, Springer-Verlag, Berlin.– 1994. —P. 363–378.

[4] King A., Soper P. Depth-k Sharing and Freeness. In P. Van Hentenryck, editor, Proceedings of the 11th International Conference on Logic Programming, Santa Margherita Ligure, Italy.– 1994. —P. 553–568.

[5] Muthukumar K., Hermenegildo M. Combined Determination of Sharing and Freeness of Program Variables through Abstract Interpretation. In K. Furukawa, editor, Proceedings of the 8th International Conference on Logic Programming, Paris. The MIT Press. – 1991. — P. 49–63.

[6] Bruynooghe M., Demoen B., Boulanger D., Denecker M., Mulkers A. A Freeness and Sharing Analysis of Logic Programs Based on a Pre-Interpretation. In R. Cousot and D. A. Schmidt, editors, 3rd Int. Symp. on Static Analysis, volume 1145 of Lecture Notes in Computer Science, Aachen, Germany. Springer Verlag.– 1996— P. 128–142.

[7] Bagnara R., Hill P. M., Zaanella E. Set-Sharing is Redundant for Pair-Sharing. In P. Van Hentenryck, editor, Proc. of the 4th Int. Symp. on Static Analysis, volume 1302 of Lecture Notes in Computer Science, Paris, France, Springer-Verlag, Berlin. –1997. — P. 53–67.

[8] Codish M., Dams D., File G., Bruynooghe M. Freeness Analysis for Logic Programs and Correctness. In D. S. Warren, editor, Proc. of the 10th Int. Conf. on Logic Programming, Budapest, Hungary. The MIT Press.– 1993 — P. 116–131.

[9] Codish M., Lagoon V., Bueno F. An Algebraic Approach to Sharing Analysis of Logic Programs. Journal of Logic Programming, 42(2).– 2000. —P.10-100.

[10] Шекета В.І. Модифікаційні предикатні запити. Науковий журнал «Проблеми програмування» Інституту Програмних Систем НАН України.-2004.–№2-3.–С.339-343// Спеціальний випуск за матеріалами 4-ї МНПК "УкрПрог`2004", 1-3 червня 2004.– Київ, Кібернетичний центр НАН України.

[11] File G., Giacobazzi R., Ranzato F. A Unifying View on Abstract Domain Design. ACM Computing Surveys, 28(2).– 1996. — P.333–336.

[12] Giacobazzi R., Scozzari F. A Logical Model for Relational Abstract Domains. ACM Transactions on Programming Languages and Systems, 20(5).– 1998. —P.1067–1109.

[13] Шекета В.І. Аналіз семантики шаблонів виклику модифікаційних предикатних запитів для інформаційних систем на основі баз даних і знань // Комп'ютерна інженерія та інформаційні технології.— Вісник національного університету "Львівська політехніка"—м.Львів.— 2003. — № 496. — С.217—228 .