Мінімізація графів з поміченими вершинами

Чепурко Віталій

Кафедра програмного забезпечення автоматизованих систем, Державний університет інформатики і штучного інтелекту, УКРАЇНА, м.Донецьк, пр. Богдана Хмельницького 84, E-mail: kven@inbox.ru

The notion minimization are used for graph reduction and are widely employed in many areas: Modal Logic, Set Theory, Formal verification. The main task is to find the equivalent of states graphs with marked vertices. Proposed a new algoritm minimize oriented, deterministic graphs with marked vertices. Defined by its time complexity.

Ключові слова – algoritm, minimization, equivalent, deterministic graphs.

Однією з центральних і актуальних як у теоретичному, так і в прикладному аспектах проблем, що виникають при дослідженнях взаємодії автоматів та операційного середовища, є проблема аналізу або розпізнавання властивостей цього середовища при різній апріорній інформації і при різних способах взаємодії автомата та операційного середовища.

Операційна середа розглядається як неорієнтований граф з поміченими вершинами[1]. Такі графи виникли спочатку як блок-схеми та схеми програм, а в даний час знаходять застосування в задачах навігації роботів.

Проблема еквівалентності програм відноситься до числа найбільш важливих проблем теорії програмування. Особливе значення проблеми еквівалентності обумовлено тим, що саме до цієї проблеми еквівалентних перетворень зводиться більшість задач оптимізації, верифікації та аналізу програм[2]. Тому розробка і впровадження ефективних алгоритмів перевірки еквівалентності програм надає помітний вплив на підвищення продуктивності та якості багатьох інструментальних засобів аналізу і перетворення програм. В останні роки поряд із завданнями підвищення ефективності і надійності програм не меншу актуальність придбало завдання своєчасного виявлення стороннього коду, яке включає виявлення недекларованих можливостей програмних продуктів, а також виявлення та усунення програм-вірусів.

У даній роботі розглядається проблема мінімізації орієнтованих графів з поміченими вершинами. Ця проблема дозволяє знаходити еквівалентні графи, за допомогою яких можна представити різні системи. Завдання мінімізації орієнтованих графів з поміченими вершинами є основою для створення різних методів еквівалентності схем програм. Вона є актуальною в теоретичному і прикладном аспектах. Її прикладна актуальність визначається тим, що проблеми самолокалізації та контролю для таких графів далекі від вирішення та їх дослідження знаходяться в зародковому стані. Теоретична важливість і актуальність визначаються тим, що аналіз графів проводиться методами, аналогічними методам теорії автоматів. Ці методи створені для графових систем, що не є кінцевими автоматами, але є в певному сенсі автоматоподобними системами.

Нехай G = (S, E, M, μ) кінцевий, зв'язний, орієнтований граф, без петель і кратних дуг [1], де S - кінцеве множина вершин, E - множина ребер, M - множина відміток, μ: S→M сюррективна функція размечування вершин така, що помітки всіх вершин, суміжних одній вершині, попарно різні. Кінцеву послідовність вершин p = g1 ... gk таку, що <gi,gi+1> є G, 1 <i <= k, назвемо шляхом довжини (k-1) у графі G. Слово μ (p) = μ (g1) ... μ (gk) назвемо відміткою шляху p. Позначку будь-якого шляху, що виходить з вершини g є S, будемо називати словом, породженим вершиною g. Мова Lg визначимо як множина всіх слів, породжених вершиною g є S [1]. Вершини g, h - назвемо еквівалентними, якщо Lg = Lh, і відрізняючимися в іншому випадку. Граф G називається приведеним, якщо всі його вершини попарно відрізняються.

Вводиться спеціальний клас детермінованих відмічених орграфів, у яких у множині наступників кожної вершини немає однаково відмічених вершин.

Множиною наступників Γ_g^- вершини g називається множина всіх вершин, які є кінцями дуг вихідних з g.

Помічений орграф $\mathcal{B}$ назвемо детермінованими орграфом або D-орграфом, якщо для будь-якої вершини g ∈ G та любих вершин s,t ∈ Γ_g^- из s ≠ t слідує, що μ(s) ≠ μ (t). В іншому випадку $\mathcal{B}$ назвемо ND-орграфом.

Введем операцію $\star$: $G \times M^+ \rightarrow 2^G$ співвідношенням: для кожної вершини g ∈ G та любого слова $w \in M^+$ через $g \star w$ позначимо множину всіх вершин $h \in G$ таких, що існує шлях, з'єднуючий вершини g *i h*. та його відмітка рівна *w*. Якщо такого шляху не існує, то $g \star w = \varnothing$. Якщо d(w) = 1 та $w = \mu(g)$, то $g \star w$ = g. Розповсюдимо операцію $\star$ на підмножин множини G природним чином: $H \star w = \bigcup_{h \in H} (h \star w)$.

Покажемо, що помічений орграф $\mathcal{B}$ є D-орграфом тоді і тільки тоді, коли для любої вершини g ∈ G та любого слова $\omega \in M^+$ виконується $|g \star \omega| \leq 1$, де $|g \star \omega| = 1$, якщо $\omega \in L_g$, и $|g \star \omega| = 0$ в противному випадку. Дійсно, для всіх слов довжини 1 та 2 це для всіх слів довжини 1 і 2 це твердження очевидне, а любе слово довжини більше 2 єдиним чином записується у вигляді композиції своїх дволітерних підслов. З іншої сторони, з того, що для кожної вершини g ∈ G та кожного слова $\omega \in M^+$ виконується $|g \star \omega| \leq 1$ слідкує, що у множині наступників $\Gamma_g^- = \bigcup_{d(w)=2} (g \star w)$

вершини g немає двох однаково відмічених вершин. Таким чином, D-орграф визначається або через локальні властивості його вершин, або через нелокальні властивості множин всіх шляхів, що виходять з вершин.

Введемо поняття повністю детермінованого орграфа або PD - орграфа. Граф $\mathcal{B}$ буде PD-графом, коли в околицях Γ_g^{-1} і Γ_g^{+1} вершини g нема двух однаково помічених вершин. Виходячи з цього визначення, якщо в множині наступників Γ_g^{-1} вершини g нема однаково помічених вершин, то будемо називати граф детермінованим назад. Якщо у множині наступників Γ_g^{+1} вершини g нема однаково помічених вершин, то будемо називати граф детермінованим вперед. Якщо граф $\mathcal{B}$ детермінован вперед і назад, то граф $\mathcal{B}$ є PD-графом.

Нехай $L_G=\{L_g\}_{g\in G}$. Графи G і H називаються еквівалентними, якщо $L_G=L_H$. В роботі розглядається задача мінімізації графів, тобто задача побудови по довільному графу G еквівалентного йому графа з найменшою кількістю вершин. Дотримуючись [1] ця задача зводиться до знаходження класів еквівалентних вершин графа G. В [1] показано, що мінімальний граф виходить з вихідного ототожненням його еквівалентних вершин. Відомо кілька алгоритмів мінімізації - це алгоритм Сапунова [1] написаний на основі алгоритму Мура для детермінованих графів. Його часова складність дорівнює O (m * n2), де n - це кількість вершин, а m - кількість оцінок графа. В загальному випадку: $O(2^n)$. В [4] описаний алгоритм мінімізації ациклічних орієнтованих графів та надані наступні оцінки складності:

- дерева – O(n);
- детерміновані ациклічні графи – O(m*n);
- недетерміновані ациклічні графи – O(n2);

У даній роботі запропоновано новий алгоритм мінімізації графів грунтується на ідеї Хопкрофта яка полягає в наступному:

розглянемо спеціальний тип розщеплення, названий розбиття. Задача розбиття множини виникає досить часто, і рішення, яке ми продемонструємо тут, повчально само по собі. Нехай дано множину S і розбиття π множини S на непересікаючиєся блоки {B1, B2, ..., Bp}. Крім того, дана функція f, що відображає S на S.

Наша задача полягає у тому, щоб знайти таке грубіше(з найменшою кількістю блоків) розбиття π' = {E1, E2,…, Eq} множини S таке, що

1) π' — підрозбиттся розбіттся π (т. е. кожна множина Ei є підмножиною деякого блоку Bj),

2) якщо a і b належать Ei то f(a) і f(b) належать Ej для деякого j.

Будемо називати π' грубійшим розбитям множини S, сумісним з π і f.

Очевидне рішення полягає в повторному потоншенні блоків вихідного розбиття наступним способом. Нехай Bi - який-небудь блок. Розглянемо f(a) для кожного a з Bi. Розіб'ємо Bi так, щоб два елементи a і b потрапляли в один блок тоді і тільки тоді, коли f (a) і f (b) обидва належать деякому блоку Bj. Процес повторюється до тих пір, поки вже не можна буде проводити подальші потоншення. Цей метод дає алгоритм складності O (n2), оскільки кожне потоншення займає час O (n), а всього може бути O (n) потоншень.

Недолік цього методу полягає в тому, що потоншення блоку може зажадати O (n) кроків, навіть якщо з нього видаляється тільки один елемент. Зараз ми опишемо алгоритм розбиття, який для поділу блоку на два підблока вимагає час, пропорційне розміру меншого подблока. Цей підхід призводить до алгоритму складності O (n log n).

Для кожного блоку B ⊆ S покладемо f -1 (B) = (b | f (b) ⊆ У). Замість того, щоб розбивати Bt за значеннями f (a) для a ^ Bit розіб'ємо щодо Bt ті блоки Bj, які містять хоча б один елемент з f -1 (Bi) і один елемент не з f -1 (Bi). Іншими словами, кожен такий блок B розбивається на множини (b | b ⊆ Bj і f (b) ⊆ Bi) та (b | b ⊆ B, і f (b) ⊆ Bi).

Як тільки проведено розбиття щодо блоку Bi більше вже не потрібно проводити розбиття щодо нього, поки він сам не буде розщеплено. Якщо спочатку f (b) ⊆ Bi для кожного елемента b ⊆ Bj і Bi розщеплено на B'i і B '', то можна розбити Bj щодо B'i або B'' і одержати при цьому один результат, оскільки (b | b ⊆ Bj, і f (b) ⊆ B'i) збігається з Bj-(b | b ⊆ Bj, і f (b) ⊆ B ''i).

Так як можна вибирати, по відношенню до якого з блоків B'i або B ''проводить розбиття, то ми розбиває щодо того, для якого це зробити простіше. Точніше ми вибираємо менше з множин f -1 (B'i) і f -1 (B ''i).

Алгоритм Хопрофта призначений для графів з поміченими дугами.

Запропонований в даній роботі алгоритм мінімізації графів призначений для графів з поміченими вершинами. Він полягає у наступному:

Алгоритм

ВХІД: множина вершин S з n елементів та функція f: S→S.

ВИХІД: розбиття p={B[1],…,B[q]} множини S, на класси еквівалентних станів.

ОБЄ'КТИ: граф заданий списком вершин, та інші об'єкти описані нижче:

v_i.class – клас приналежності;

v_i.mark – відмітка вершини;

B[j] – j-й клас, містить номери еквівалентних вершин;

B[j].prev – кількість попередніх вершин j-го класу;

BT[k] – тимчасовий класс містить єквівалентні вершини ;

$f^{-1}(v_i)$ – множина попередніх вершин поточній;

WAIT – множина номерів класів, які будуть оброблятьcя;

q – кількість класів;

ІНІЦІАЛІЗАЦІЯ:

Переглянемо всі вершини і кожній призначимо номер класу рівний номеру мітки v.class = v.mark. Номер вершини розмістимо в клас з індексом рівним класу вершини.

Переглянемо всі дуги графа і створимо списки попередніх вершин.

У множину WAIT розмістимо всі початкові класи.

МЕТОД:

1. Поки WAIT не пусто вибираємо та вилучаємо і клас, який знаходиться останній у списку.
2. Для всіх вершин класу B[i] знаходимо списки попередніх вершин та розбиваємо їх у відповідності з классом, якому належать.
3. Для кожного отриманого з пункту 2 класу BT[k] виконуємо:
   3.1 Якщо | BT[k]|<>| B[k]| то розбиваємо класс B[k]
   3.2 q=q+1
   3.3 B[q]= BT[k]
   3.4 B[k]=B[k]-B[q]
   3.5 Якщо k не міститься у WAIT то поміщаємо у WAIT номер того класса потужність якого по попереднім вершинам менше.

Доведено, що результатом роботи алгоритму є класи еквівалентних вершин, тобто він вирішує завдання мінімізації. Визначена часова складність алгоритму.

При підрахунку тимчасової складності алгоритму вийшло два варіанти оцінки. Варіанти відрізняються між собою мінімальним кроком виконання алгоритму взятим за одиницю часу.

У першому випадку за одиницю часу будемо вважати операції:

- операція додавання єлемента у множину;
- операція видалення елемента з множини;
- порівняння двух чисел;
- інші одиничні оперецїї.

У такому випадку асимптотична часова складність алгоритму дорівнює O(m*n), де n – кількість вершин графу, m – кількість відміток у графі.

Якщо у якості мінімальної одиниці часу брати найпростіші операції, як оперецація порівняння, присвоєння, то асимптотична часова складність дорівнює O(m*n*log(n)).

Складність алгоритму зростає за рахунок операцій з множинами, яким необхідно O(log(n)) шагів.

Список литературы

[1] Сапунов С.В. Анализ графов с помеченными вершинами: Дисс… канд. физ.-мат. наук: 27.09.07. – Донецк.: 2007. – 150 с.

[2] R. Gentilini, C. Piazza and A. Policriti From Bisimulation to Simulation. Coarest partition problems // Dip. Di Matematica e Informatica – Universita di Udine Via Le Scienze, 206 – 33100 Udine – Italy.

[3] А.Ахо, Дж. Хопкрофт, Дж.Ульман. Построение и анализ вычислительных алгоритмов. –М.: «МИР». 1979. – 235с.

[4] Чепурко В.А. Минимизация ориентированных ациклических графов с отмеченными вершинами // Материалы VI международной научно-практической конференции «Математическое и программное обеспечение интеллектуальных систем». Днепропетровск, 12-14 ноября 2008 г. Днепропетровск.: 2008. – С. 331 – 332.

Чисельне моделювання функції розподілу електронів за енергіями для рівняння Больцмана та інших параметрів плазми розряду

Василь Чигінь, Павло Горун

Кафедра прикладної математики та фундаментальних наук, Національний університет “Львівська політехніка”, УКРАЇНА, м.Львів, вул. Митрополита Андрея 5, E-mail: vchygin@polynet.lviv.ua, Pawlissimo@gmail.com

Abstract* – *using processed numerical methods, were founded and compared the electron energy distribution functions for Boltzmann equation, that was represented in stationary case, electronic processes rates, power loss coefficients and other parameters of negative corona in the mixture of Ar-Kr-Xe and Cl_2.

Ключові слова – чисельне моделювання, коронний розряд, рівняння Больцмана.

I. Вступ

Проводилося чисельне моделювання кінетичного рівняння Больцмана для функції розподілу електронів за енергіями (ФРЕЕ), на підставі якого отримані транспортні характеристики електронів плазми, а також питомі втрати потужності розряду при різних значеннях приведеної напруженості електричного поля. Моделювання параметрів плазми виконане для чотирьохкомпонентної суміші (Ar-Kr-Xe-Cl_2) газів, найбільш оптимальної для роботи багатохвильового джерела УФ-ВУФ випромінювання. Проведений якісний аналіз найбільш важливих електронних процесів в багатокомпонентній плазмі, які визначають механізми сумісного утворення монохлоридів аргону, ксенону і криптону в поперечному розряді.