УДК 510.662

К.т.н. доцент Россошинський Д.О., магістрант Чаюн В.Г.

Національний технічний університету України
«Київський політехнічний інститут»

ОПТИМІЗАЦІЯ МЕТОДІВ РЕЗОЛЮТИВНОГО ВИВОДУ

Abstract

Dmitriy Rososhinskiy, PhD; Viktor Chaun, undergraduate
Optimisation of the methods of finding resolutive deduction

This paper concerns the conceptual approach for designing internal memory structures of the resolution methods. The idea of optimization based on linear conception of the logical program in memory. This paper also reproduces approach of creating effective the finding procedure of clause. The Idea based on the allocation necessary and sufficient condition can be used in other search procedure with complex data structures.

Вступ

Для пошуку розв’язку все більшої кількості таких специфічних задач як планування, розуміння природної мови, побудови експертних систем та інших задач штучного інтелекту все частіше поширюється підхід із застосуванням логічного програмування, основаному на методі резолюції. Досить широкий клас задач потребує отримання рішень в реальному часі, тому задача ефективної реалізації методів резолюції є актуальною.

В роботі розглядаються структури та деякі підходи, що дозволяють більш ефективно реалізувати метод пошуку резолютивного виводу.

Постановка задачі

Розробити структури для лінійного представлення логічної програми в уніфікованому виді. Запропонувати основні підходи ефективної реалізації алгоритмів пошуку резолютивного виводу, зокрема:

- пошуку найбільш загального уніфікатора (НЗУ);
- пошуку резольвенти в множині диз’юнктів.

Структури лінійного представлення логічної програми

Основою всіх реалізацій алгоритмів пошуку резолютивного виводу є пошук найбільш загального уніфікатора. Процес уніфікації побудований на пошуку невідповідності між заданими формулами і усуненням їх за допомогою замін змінних. Заміна може відбуватися на [1]:

- інші логічні змінні;
- константи;
- функціональні символи, що не містять заміняючої змінної.

Пропонується ввести єдину таблицю унікальних імен, що дозволить в подальшому оперувати лише ідентифікаторами записів таблиць, а не значеннями рядків, пришвидшивши роботу із рядковими об’єктами.

Формули (клаузи), предикати, функціональні символи логічної програми мають різну довжину, це призводить до проблеми представлення програми в пам’яті комп’ютера та ускладнення алгоритмів обробки структур програми [2]. Саме тому, пропонується привести представлення логічної програми в деякий уніфікований, лінійний вид. Програма в пам’яті буде розміщуватися послідовно із формул, які в свою чергу складаються із предикатів, а предикати містять змінні, константи та функціональні символи — це і буде лінійним представленням програми. Кожна структура програми містить додаткові поля, що дозволяють пришвидшити швидкість формування логічного виводу (кількість мінімальних символів; посилання на додаткову інформацію про клаузу, тобто яким чином була отримана резольвента; кількість предикатів; арність). Всі змінні та константи реалізовані як посилання на записи у відповідні таблиці змінних і констант. Таблиця змінних має додаткові поля:

- тип посилання;
- ідентифікатор.

Якщо в процесі пошуку НЗУ, буде знайдено заміну змінної, то достатньо буде її вказати в одному місці за допомогою посилання. Виконана заміна буде автоматично поширена на всі змінні предикатів формули.

Для розробки структур єдиного уніфікованого способу представлення логічної програми виділимо наступні правила:

- Будь яка формула складається із предикатних символів, в іншому випадку маємо пустий диз'юнкт.
- Всі предикати помічені номерами. Предикат формули із мінімальним номером називається мінімальним предикатним символом.
- Якщо формула не пустий диз'юнкт, то вона має, як мінімум, один мінімальний предикатний символ.
- Кількість мінімальних символів не може перевищувати кількості предикатів в формулі.
- Кожен предикатний символ має арність більше або рівну 1.
- В якості аргументів предикату можуть виступати: змінні, константи, функціональні символи. Інших аргументів не може бути.
- Кожен функціональний символ має арність більше або рівну 1.

В середині формули виконується сортування предикатних символів по іменам та/або наскрізній нумерації.

Для оптимізації процесу пошуку формули, пропонується виділити необхідні та достатні умови рівності функціональних символів, предикатів, формул. Відомо, якщо виконуються достатні умови, то обов'язково виконуються і необхідні, але не навпаки [1]. Таким чином, якщо не виконується необхідні умови рівності формул, то одразу стає зрозуміло, що немає сенсу далі виконувати перевірку, процедура одразу поверне *false*.

Введемо необхідні та достатні умови рівності формул, предикатів, функціональних символів. Необхідними умовами рівності двох формул є рівність кількості предикатних символів. Необхідними умовами рівності предикатів є співпадіння імен, знаків та їх арності. Достатньою умовою рівності предикатів є рівність аргументів з точністю до імен, але із їх урахуванням взаємного положення в предикатах. Таким чином, навіть якщо імена змінних двох предикатів відрізняються (це може бути пов'язано із перейменуванням змінних в процесі виводу), але їх положення в предикатах співпадає, то формули вважаються рівними. Наприклад, наступні формули однакові незалежно від того, що вони мають різні змінні [3]:

$$P(x_{1-1}, y_{2-3}, f(x_{1-1}, y_{2-3}, g(x_{1-1}))) \text{ та } P(t_{7-2}, z_{5-8}, f(t_{7-2}, z_{5-8}, g(t_{7-2})))$$

Необхідні та достатні умови рівності функціональних символів аналогічні, що і для предикатів. Дві логічні формули рівні тоді і тільки тоді, коли всі предикати формул співпадають.

Визначимо наступний порядок перевірки необхідних і достатніх умов:

1. необхідні умови рівності формул;
2. необхідні умови рівності предикатів;
3. необхідні умови рівності функціональних символів;
4. достатні умови рівності предикатів (функціональних символів).

Оскільки в алгоритмі всі перевірки умов розміщені в порядку від найбільш загальних (необхідних умов) до найбільш детальних (достатніх умов), то даний підхід дозволить одразу відсікати варіанти, що не підходять вже на початкових етапах. Тільки у випадку, коли обидві формули співпадають, процедура перевірить всі умови рівності.

Висновок

Запропонований підхід реалізації лінійного представлення логічної програми та метод пошуку формули в множині диз'юнктів, дозволяє значно підвищити швидкість та ефективність процесу уніфікації формул та роботи резолютивних методів зокрема.

Підхід пошуку логічної формули в множині диз'юнктів можна аналогічно застосувати і в інших методах багатокритеріального пошуку виділивши свої власні необхідні та достатні ознаки з наступним їх відсортуванням від найбільш загальних (необхідних умов) до найбільш детальних (достатніх умов).

Література

1. *Таран Т.А.* Основы дискретной математики. — К.: Просвіта. 2003. — 288 с.
2. Бартко И. Алгоритмы искусственного интеллекта на языке Prolog. — М.: Издательский дом «Вильямс», 2004. — 640 с.
3. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. — М.: Наука. 1983. —360 с.