УДК 004.4(045)

ДЕДУКТИВНИЙ СИНТЕЗ АЛГОРИТМУ СОРТУВАННЯ МЕТОДОМ РЕЗОЛЮЦІЇ З ПОБУДОВОЮ ДЕРЕВ СПРОСТУВАННЯ ТА ДОВЕДЕННЯ

Радішевський М.Ф., Лозицький В.В.

Національний авіаційний університет

mykola.Radshevskiy@live.nau.net

Стаття присвячена опису дедуктивного принципу синтезу програм з використанням методу резолюції на прикладі отримання алгоритму сортування. У статті показано принцип, що дає можливість отримувати твердження-відповіді на основі використання методу побудови дерев спростування та доведення.

This article is dedicated to the deductive program synthesis with using of resolution method on an example of sorting algorithm creation. In the article there is a shown principle that allows getting assertions-answers on the basis of method refutation and leading trees construction using.

Вступ

У теорії синтезу програм (*program synthesis*) розрізняють чотири основні підходи: *дедуктивний, індуктивний, трансформаційний* та *утилітарний*.

Дедуктивний підхід реалізовує побудову програми використовуючи доказ твердження, що розв'язк задачі існує. За індуктивного підходу програма будується лише на прикладах, що безпосередньо задають відповідь для деякого початкових даних. Підхід, за яким програма отримується через перетворення початкового опису завдання за правилами, сукупність яких і являє знання про розв'язування задач, називається трансформаційним підходом.

За утилітарного підходу програма будується з практичних потреб на основі конкретних закономірностей і прийомів [1; 2].

Постановка завдання

За дедуктивного підходу задача синтезу предподається у вигляді теореми:

$$\Phi_0 \to \mathrm{T}, \tag{1}$$

де Φ_0 являє собою множину формул, а T — формула, що логічно випливає з неї.

Доведення теореми (1) полягає в тому, аби показати, що кожна інтерпретація, що задовольняє Φ_0, задовольняє і T, або, що те ж саме, об'єднання

$$\Phi_1 = \Phi_0 \cup \overline{\mathrm{T}} \tag{2}$$

не може бути виконаним. Зазвичай використовують саме другий підхід, тобто доводять, що множина (2) не може бути виконаною. Задачу доведення можна вирішити за допомогою методу резолюції, з можливим використанням певної стратегії [3; 4].

У задачу синтезу алгоритму сортування насамперед виділимо базові відношення та елементарні функції. Нехай задані такі відношення: $R(x, y)$, де x — довільний список, а y — упорядкований список, що містить ті ж числа, що й x, але розміщені у спадному порядку; $S(y)$ — список упорядкований; $I(x, y)$ — два списки x і y складаються з одних і тих же елементів.

Нехай мають місце такі елементарні функції:

$\mathrm{car}(x)$: $x = (x_1,\dots,x_n) \to x_1$ — відображає список x у перший елемент цього списку;

$\mathrm{cdr}(x): x = (x_1,\dots,x_n) \to (x_2,\dots,x_n)$ — відображає список x у той же список без першого елемента;

$\mathrm{cons}(x_1, y): \{x_1, y = (y_1,\dots,y_n)\} \to (x_1, y_1,\dots,y_n)$ — відображає елемент x_1 і список y в новий список, який отримується у разі розміщення x_1 перед списком y;

$\mathrm{merge}(x_1, y)$ (відображає елемент x_1 і впорядкований список y в упорядкований список, що містить елементи списку y й елемент x_1). Тобто дана елементарна функція находить відповідне місце у відсортованому списку y для такого розміщення там елементу x, щоб результуючий список не потребував сортування.

Відношення $R(x, y)$ за допомогою введених відношень $S(y)$ і $I(x, y)$ можна визначити так:

$$\begin{aligned}&\forall x \forall y \{\{R(x, y) \to [S(y) \wedge I(x, y)\ ]\} \wedge \\ &\wedge \{[S(y) \wedge I(x, y)\ ] \to R(x, y)\ \}\}.\end{aligned} \tag{3}$$

Відношення $S(y)$ і $I(x, y)$ через елементарні функції визначаються такими формулами:

$$\begin{aligned}&\forall x \forall y \forall u \{I(x, y) \to \\ &\to I[\mathrm{cons}(u, x),\ \mathrm{merge}(u, y)\ ]\};\end{aligned} \tag{4}$$

$$I(\mathrm{nil}, \mathrm{nil}); \tag{5}$$

$$\forall x \forall y \{S(y) \to S[\mathrm{merge}(x, y)\ ]\}; \tag{6}$$

$$S(\mathrm{nil}). \tag{7}$$

Сукупність наведених формул є формальним (тобто на мові числення предикатів) визначенням введених відношень та елементарних формул. Виконаємо перетворення формул (3)—(7) у множину речень.

Формула (3) в сукупності після виконання послідовності тотожних перетворень.

$$\overline{R(x,y)} \vee S(y), \overline{R(x,y)} \vee I(x,y),$$
$$\{\overline{S(y)} \vee \overline{I(x,y)} \vee R(x,y)\}. \quad (8)$$

Перетворення формул (4)—(7) у відповідні речення полягає у виконанні над ними всіма правила виключення знаків імплікації, якщо ті мають місце. Звідки матимемо:

$$\overline{I(x,y)} \vee I[\mathrm{cons}(u,x),\ \mathrm{merge}(u,y)]; \quad (9)$$
$$I(nil,nil); \quad (10)$$
$$\overline{S(y)}\,Ú\,S[\mathrm{merge}(x,y)]; \quad (11)$$
$$S(nil). \quad (12)$$

Результуюча множина речень M являє собою об'єднання ДКНФ з (11)—(12):

$$M = \{\overline{R(x,y)} \vee S(y);\ \overline{R(x,y)} \vee I(x,y);$$
$$\overline{S(y)} \vee \overline{I(x,y)} \vee R(x,y);$$
$$\overline{I(x,y)} \vee I[\mathrm{cons}(u,x), \mathrm{merge}(u,y)\ ];$$
$$I(nil,nil);\ S(nil);\ \overline{S(y)} \vee S[\mathrm{merge}(x,y)\ ]\}. \quad (13)$$

Завдання синтезу алгоритму полягає в тому, щоб виразити відображення $y = \mathrm{sort}(x)$ (sort — оператор упорядкування) через наведені елементарні функції.

Спробуємо довести припущення $\forall x \exists y R(x,y)$ методом математичної індукції: спочатку для списку довжиною нуль, потім, припустивши, що воно вірне, для списків довжини $n \geq 0$ і доведемо, що воно справедливе для списків довжиною $n+1$.

Отже, якщо списки довжини $n = 0$, то припущення можна сформувати у вигляді $\exists y\ R(nil,y)$. Виконавши заперечення матимемо:

$$\overline{R(nil,y)} \quad (14)$$

Включивши речення (припущення) (17) у множину речень (16) і використовуючи принцип резолюції, побудуємо спочатку дерево спростування:

(1) $\overline{R(x,y)} \vee S(y)$

(2) $\overline{R(x,y)} \vee I(x,y)$

(3) $\overline{S(y)} \vee \overline{I(x,y)} \vee R(x,y)$

(4) $\overline{I(x,y)} \vee I[\mathrm{cons}(u,x), \mathrm{merge}(u,y)\ ]$

(5) $I(nil,nil)$

(6) $\overline{S(y)} \vee S[\mathrm{merge}(x,y)]$

(7) $S(nil)$

(8) $\overline{R(nil,y)}$

(9) За правилами (8),(3):

$$(nil,x)\ \frac{\overline{R(nil,y)}\ , \overline{S(y)} \vee \overline{I(nil,y)} \vee R(nil,y)}{\overline{S(y)} \vee \overline{I(nil,y)}}$$

(10) За правилами (9),(7):

$$(nil,y)\ \frac{\overline{S(nil)} \vee \overline{I(nil,nil)}, S(nil)}{\overline{I(nil,nil)}}$$

(11) За правилами (10),(5):

$$\frac{\overline{I(nil,nil)} \vee I(nil,nil)}{nil}$$

Після цього побудуємо відповідне модифіковане дерево доведення, користуючись такими правилами: (1) $\overline{R(x,y)} \vee S(y)$

(2) $\overline{R(x,y)} \vee I(x,y)$

(3) $\overline{S(y)} \vee \overline{I(x,y)} \vee R(x,y)$

(4) $\overline{I(x,y)} \vee I[\mathrm{cons}(u,x), \mathrm{merge}(u,y)\ ]$

(5) $I(nil,nil)$

(6) $\overline{S(y)} \vee S[\mathrm{merge}(x,y)]$

(7) $S(nil)$

(8) $\overline{R(nil,y)} \vee R(nil,y)$

(9) За правилами (8),(3):

$$(nil,x)\frac{\overline{R(nil,y)} \vee R(nil,y), \overline{S(y)} \vee \overline{I(nil,y)} \vee (nil,y)}{\overline{S(y)} \vee \overline{I(nil,y)} \vee R(nil,y))}$$

(10) За правилами (9),(7):

$$(nil,y)\frac{\overline{S(nil)} \vee \overline{I(nil,nil)} \vee R(nil,nil), S(nil)}{\overline{I(nil,nil)} \vee R(nil,nil))}$$

(11) За правилами (10),(5):

$$\frac{\overline{I(nil,nil)} \vee (nil,nil), I(nil,nil)}{R(nil,nil)}$$

Таким чином, отримавши у відповідь вираз $R(nil,nil)$, робимо висновок, що якщо довжина списку x рівна нулю, то $y = nil$.

Тепер сформулюємо припущення індукції, що для кожного непустого списку x значення $cdr(x)$ можна відсортувати (припущення індукції). Тоді це припущення можна записати у вигляді:

$$R(cdr(x),\ sort(cdr(x))) \quad (15)$$

Для доведення $\forall x \exists y R(x,y)$ нам знадобиться наступне співвідношення між car, cdr ³ $cons$:

$$\forall x\ \{cons(car(x), cdr(x)) = x\}; \quad (16)$$

уведемо це співвідношення у вигляді речення:

$$\overline{R(\mathrm{cons}(car(x), cdr(x)\ ), y)} \vee R(x,y). \quad (17)$$

Запереченням речення $\forall x \exists y R(x,y)$ слугує:

$$\overline{R(a,y)}, \quad (18)$$

де a — функція Сколема, яка з'являється, коли речення містить змінні, що відносяться до квантора загальності.

Включимо аксіоми (18), (20), (21) у множину речень (16) і, використовуючи принцип резолюції, побудуємо спочатку дерево спростування:

(1) $\overline{R(x,y)} \vee S(y)$

(2) $\overline{R(x,y)} \vee I(x,y)$

(3) $\overline{S(y)} \vee \overline{I(x,y)} \vee R(x,y)$

(4) $\overline{I(x,y)} \vee I[\mathrm{cons}(u,x), \mathrm{merge}(u,y)\ ]$

(5) $I(nil,nil)$

(6) $\overline{S(y)} \vee S[\mathrm{merge}(x,y)]$

(7) $S(nil)$

(8) $R(cdr(x), sort(cdr(x)))$

(9) $\overline{R(\mathrm{cons}(car(x),cdr(x)\ ),y)} \vee R(x,y)$

(10) $\overline{R(a,y)}$

(11) За правилами (10),(9):

$$(a,x)\frac{R(a,y)\vee \overline{R(\mathrm{cons}(car(x),cdr(x)\ ),y)}\vee R(x,y)}{\overline{R(\mathrm{cons}(car(a),cdr(a)\ ),y)}}$$

(12) За правилами (11),(3):

$(\mathrm{cons}(car(a),cdr(a)\ ),x)$

$$\frac{\overline{R(\mathrm{cons}(car(a),cdr(a)),y)},\overline{S(y)}\vee}{\overline{S(y)}\vee\overline{I(\mathrm{cons}(car(a),cdr(a)),y)}}$$

$$\frac{\vee\overline{I(\mathrm{cons}(car(a),cdr(a)),y)}\vee R(\mathrm{cons}(car(a),cdr(a)),y)}{\overline{S(y)}\vee\overline{I(\mathrm{cons}(car(a),cdr(a)),y)}}$$

(13) За правилами (12),(4):

$(car(a),u;\ \ cdr(a),x;\ \mathrm{merge}(car(a),y),y)$

$\overline{I(\mathrm{cons}(car(a),cdr(a)\ ),\mathrm{merge}(car(a),y)\ )}\vee$

$\vee S(\mathrm{merge}(car(a),y)\ ),$

$$\frac{I(\mathrm{cons}(car(a),cdr(a)),\mathrm{merge}(car(a),y))\vee I(cdr(a),y)}{\overline{S(merge(car(a),y))}\vee\overline{I(cdr(a),y)}}$$

(14) За правилами (13),(2):

$(cdr(a),x)\times$

$$\times\frac{\overline{S(\mathrm{merge}(car(a),y))\vee\ I(cdr(a),y)}\ ,\overline{R(cdr(a),y)}\vee I(cdr(a),y)}{\overline{S(\mathrm{merge}(car(a),y))}\ \vee\ \overline{R(cdr(a),y)}}$$

(15) За правилами (14),(6):

$$(car(a),x)\frac{\overline{S(merge(car(a),y))}\vee\overline{R(cdr(a),y)}\ ,}{\overline{R(cdr(a),y)}\ \vee\overline{S(y)}}$$

$$\frac{\overline{S(y)}\vee S[merge(car(a),y)\ ]}{\overline{R(cdr(a),y)}\ \vee\overline{S(y)}}$$

(16) За правилами (15),(1)

$$\frac{\overline{R(cdr(a),y)}\vee\overline{S(y)},\overline{R(x,y)}\vee S(y)}{\overline{R(cdr(a),y)}\ \vee\overline{R(x,y)}}$$

(17) За правилами (16),(8):

$(x,a;\ sort(cdr(x)\ ),y;\ cdr(x),x)$

$$\frac{R(cdr(x),sort(cdr(x)))\vee\overline{R(cdr(x),sort(cdr(x)))},}{nil}$$

$$\frac{R(cdr(x),sort(cdr(x)))}{nil}$$

Тепер побудуємо відповідне модифіковане дерево доведення, користуючись правилами, описаними вище:

(1) $\overline{R(x,y)}\ \vee S(y)$ (2) $\overline{R(x,y)}\ \vee I(x,y)$

(3) $\overline{S(y)}\ \vee\overline{I(x,y)}\ \vee R(x,y)$

(4) $\overline{I(x,y)}\ \vee I[\mathrm{cons}(u,x),\mathrm{merge}(u,y)\ ]$

(5) $I(nil,nil)$

(6) $\overline{S(y)}\ \vee S[\mathrm{merge}(x,y)]$

(7) $S(nil)$

(8) $R(cdr(x), sort(cdr(x)))$

(9) $\overline{R(\mathrm{cons}(car(x),cdr(x)\ ),y)}\vee R(x,y)$

(10) $\overline{R(a,y)}\vee R(a,y)$

(11) За правилами (10),(9):

$$(a,x)\frac{\overline{R(a,y)}\vee(a,y),\overline{R(\mathrm{cons}(car(a),cdr(a)),y)}\vee(a,y))}{R(a,y)\vee\overline{R(\mathrm{cons}(car(a),cdr(a)),y)}}$$

(12) За правилами (11),(3):

$(\mathrm{cons}(car(a),cdr(a)),x)$

$$\frac{R(a,y)\ \ \vee\overline{R(cons(car(a),cdr(a)),y)}\ ,\overline{S(y)}\vee}{R(a,y)\vee\overline{S(y)}\ \ \vee\overline{I(cons(car(a),cdr(a)),y)}}$$

$$\frac{\vee\overline{I(cons(car(a),cdr(a)),y)}\ \ \vee}{R(a,y)\vee\overline{S(y)}\ \ \vee\overline{I(cons(car(a),cdr(a)),y)}}$$

$$\frac{\vee R(\mathrm{cons}(car(a),cdr(a)),y)}{R(a,y)\vee\overline{S(y)}\ \ \vee\overline{I(\mathrm{cons}(car(a),cdr(a)),y)}}$$

(13) За правилами (12),(4):

$(car(a),u;cdr(a),x;\mathrm{merge}(car(a),y),y)$

$R(a,\mathrm{merge}(car(a),y))\vee\overline{S(\mathrm{merge}(car(a),y))}\ \ \vee$

$\overline{I(\mathrm{cons}(car(a),cdr(a)),\mathrm{merge}(car(a),y))}\ ,$

$$\frac{\overline{I(cdr(a),y)}\vee I[\mathrm{cons}(car(a),cdr(a)),\mathrm{merge}(car(a),y)]}{R(a,\mathrm{merge}(car(a),y))\vee\overline{S(\mathrm{merge}(car(a),y))}\vee\overline{I(cdr(a),y)}}$$

(14) За правилами (13),(2):$(cdr(a),x)$

$$\frac{R(a,merge(car(a),y))\vee\overline{S(merge(car(a),y))}\vee\overline{I(cdr(a),y)},\overline{R(cdr(a),y)}\vee I(cdr(a),y)}{R(a,merge(car(a),y))\vee\overline{S(merge(car(a),y))}\vee\overline{R(cdr(a),y)}}$$

(15) За правилами (14), (6): $(car(a),x)$

$$\frac{R(a,\mathrm{merge}(car(a),y))\vee\overline{S(\mathrm{merge}(car(a),y))}\vee\overline{R(cdr(a),y)},\ \ \overline{S(y)}\vee S[\mathrm{merge}(car(a),y)]}{R(a,\mathrm{merge}(car(a),y))\vee\overline{R(cdr(a),y)}\vee\overline{S(y)}}$$

(16) За правилами (15),(1):

$$\frac{R(a,\mathrm{merge}(car(a),y))\vee\overline{R(cdr(a),y)}\vee\overline{S(y)},\overline{R(x,y)}\vee S(y)}{R(a,\mathrm{merge}(car(a),y))\vee\overline{R(cdr(a),y)}\ \ \vee\overline{R(x,y)}}$$

(17) За правилами (16),(8):

$(x,a;\ \ sort(cdr(x)),y;\ \ cdr(x),x)$

$$\frac{R(x,\mathrm{merge}(car(x),sort(cdr(x))))\vee\overline{R(cdr(x),sort(cdr(x)))}\vee \vee\overline{R(cdr(x),sort(cdr(x)))}\ ,R(cdr(x),sort(cdr(x)))}{R(x,\mathrm{merge}(car(x),sort(cdr(x))))}$$

Висновок. Отримавши у відповідь вираз $R(x,\mathrm{merge}(car(x),\quad sort(cdr(x))))$, робимо висновок, що якщо довжина списку x нерівна нулю, то $y=\mathrm{merge}(car(x),\ \ sort(cdr(x)))$.

Об'єднавши результати, отримані при $n=0$ і $n\neq 0$, знаходимо:

$$\mathrm{sort}(x) = \begin{cases} \mathrm{nil}, \text{ якщо } x = nil; \\ \mathrm{merge}(car(x), \mathrm{sort}(cdr(x))) - \text{в іншому випадку} \end{cases}$$

Дане співвідношення визначає рекурсивну програму упорядкування списку чисел довільної довжини.

ЛІТЕРАТУРА

1. *Тыугу Э.Х.* Концептуальное программирование. — М.: Наука, 1984. — 256 с.

2. *Башмаков А.И., Башмаков И.А.* Интеллектуальные информационные технологии. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. — 304 с.

3. *Назаретов В.М., Ким Д.П.* Робототехника и гибкие автоматизированные производства. В 9 кн. Кн. 6. Техническая имитация интеллекта: учеб. пособ. для втузов. — М.: Высш. шк., 1986. — 144 с.

Чень Ч. Ли Р. Математическая логика и автоматическое доказательство теорем. М.: Наука. Глав. ред. физ.-мат. лит-ры, 1983. — 360 с.

Стаття надійшла до редакції 16.02.09