ФУНКЦІОНАЛЬНЕ ПРОГРАМУВАННЯ

Лекція 12

АЛГОРИТМІЧНА ПОВНОТА ФУНКЦІОНАЛЬНОЇ ПРОГРАМИ. НЕРУХОМІ ТОЧКИ

2020

Повний текст лекції буде розміщений на сайті baklaniv.at.ua

АЛГОРИТМІЧНА ПОВНОТА ФУНКЦІОНАЛЬНОЇ ПРОГРАМИ

Властивість *алгоритмічної повноти* множини ФП полягає в тому, що кожна часткова функція на рядках, обчислювана в інтуїтивному сенсі, може бути описана деякою ФП.

Ми припускаємо, що для кожної часткової функції f на рядках, обчислюваною в інтуїтивному сенсі, існує машина Тьюрінга M, така, що f збігається з функцією, яку обчислює M.

Звернемо увагу на те, що для кожної машини Тьюринга M існує ФП така, що описує функцію, яку обчислює машина M.
Звідки це маємо?

Нехай M буде машина Т'юрінга, компонентами якої є
• множина станів Q = {q0, q1,. . . , Qn}, в якій виділені
початковий стан q0,
і заключний стан qn.
• алфавіт символів A = {a1,. . . , Am}, які можуть бути написані в клітинках стрічки, причому A містить символ пробілу
• відображення переходів
δ: Q x A → Q x A x {left, right}

Поставимуу відповідність кожному стану $q_i \in Q$ функціональну змінну φ_i, де
$type\ (\varphi_i) = (string, string) \rightarrow string$.

ФП Σ_M, которая описывает функцию, вычисляемую машиной M, состоит из следующих уравнений:

- уравнение, соответствующее той функции, которую вычисляет машина M:

$$\varphi(x) = \varphi_0(\varepsilon, x)$$

- если q_i – незаключительное состояние, и

$$\delta(q_i, a_1) = (q_j, a_k, \texttt{right})$$
$$\delta(q_i, a_2) = \ldots$$

то ФП Σ_M содержит уравнение

$$\varphi_i(x, y) = \quad (y = \varepsilon) \quad ? \; \varphi_i(x, \square)$$
$$: (\hat{y} = a_1) \quad ? \; \varphi_j(a_k \cdot x, y')$$
$$: (\hat{y} = a_2) \; ? \ldots$$

где

  - значения, принимаемые переменной x, соответствуют записям на ленте машины M слева от головки, читаемым справа налево, и
  - значения, принимаемые переменной y, соответствуют записям на ленте справа от головки (включая символ под головкой)

- если q_i – незаключительное состояние, и

$$\delta(q_i, a_1) = (q_j, a_k, \texttt{left})$$
$$\delta(q_i, a_2) = \ldots$$

то ФП Σ_M содержит уравнение

$$\begin{aligned}
\varphi_i(x, y) = & \\
= (y = \varepsilon) \quad & ? \ \varphi_i(x, \square) \\
: (x = \varepsilon) \quad & ? \ \varphi_i(\square, y) \\
: (\hat{y} = a_1) \quad & ? \ \varphi_j(x', \hat{x} \cdot a_k \cdot y') \\
: (\hat{y} = a_2) \ & ? \ldots
\end{aligned}$$

- уравнение, соответствующее заключительному состоянию q_n:

$$\varphi_n(x, y) = (y = \varepsilon)\ ?\ \varepsilon$$
$$: (last(y) = \square)\ ?\ \varphi_n(x, rem(y)) : y$$

где

  - функция $last$ возвращает последний символ своего аргумента, она описывается ФП

$$\varphi(x) = (x' = \varepsilon)\ ?\ \hat{x} : \varphi(x')$$

  - функция rem возвращает строку, получаемую удалением последнего символа из своего аргумента, она описывается ФП

$$\varphi(x) = (x' = \varepsilon)\ ?\ \varepsilon : \hat{x} \cdot \varphi(x')$$

  - функция, соответствующая функциональной переменной φ_n, возвращает строку, получаемую из значения второго аргумента удалением пробелов в его конце. ■

Обчислення значень найменших нерухомих точок функціональних програм

Тепер ми розглядаємо задачу обчислення значень ННТ ФП на заданих значеннях аргументів.

З метою простоти викладу, ми розглядаємо лише ФП з однієї функціональної змінної φ. Для кожного розглянутого терма r виконана умова

$F\,Var\,(r) \subseteq \{\varphi\}$.

Завдання обчислення значень ННТ ФП полягає в побудові алгоритму, який

• по заданій ФП Σ виду

$$\Sigma: \varphi(\bar{x}) = e \qquad (32)$$

• і заданому списку $\bar{d} \in D\bar{x}$ повинен обчислити значення $\sigma(\bar{d})$.

Відзначимо, що для вирішення даного завдання не потрібно знаходження терма, якому відповідав би функція σ.

Один з можливих методів вирішення даного завдання полягає в тому, що за заданими

• ФП Σ виду (32), і

• списку $\bar{d} \in D\bar{x}$

будується послідовність термів

$$C^0_{\Sigma,\bar{d}} \quad C^1_{\Sigma,\bar{d}} \quad C^2_{\Sigma,\bar{d}} \quad \ldots \tag{33}$$

звана обчислювальною послідовністю. Кожен її член є, в деякому сенсі, аппроксимацией шуканого значення $\sigma(\bar{d})$.

Якщо обчислювальна послідовність (33) кінцева, то її останній елемент повинен бути константою, значення якої дорівнює σ (d̄).

Обчислювальна послідовність (33) будується наступним чином:

1. Терм $C^0_{\Sigma,\bar{d}}$ имеет вид $\varphi(\bar{d})$.

2. Если терм $C^i_{\Sigma,\bar{d}}$ содержит функциональную переменную φ, то терм $C^{i+1}_{\Sigma,\bar{d}}$ получается из $C^i_{\Sigma,\bar{d}}$

   (a) заменой в нём некоторых подтермов вида

   $$\varphi(e_1, \ldots, e_n) \tag{34}$$

   на термы вида

   $$e(e_1/x_1, \ldots, e_n/x_n) \tag{35}$$

   где $(x_1, \ldots, x_n)$ – это список $\bar{x}$ в (32), и

   (b) упрощением получившегося терма.

3. Если терм $C^{i}_{\Sigma,\bar{d}}$ не содержит φ, то он является последним членом последовательности (33).

Ниже мы будем называть

- те подтермы вида (34) терма $C^i_{\Sigma,\bar{d}}$, которые выбираются для замены согласно пункту (2a), **раскрываемыми** подтермами, и
- замену (34) на (35) – **раскрытием** подтерма (34).

Если среди раскрываемых подтермов терма $C^i_{\Sigma,\bar{d}}$ одни подтермы содержатся в других, то раскрытие таких подтермов осуществляется по принципу “от меньших к большим”, т.е. каждый раскрываемый подтерм раскрывается только после того, как будут раскрыты все содержащиеся в нём раскрываемые подтермы.

ОБЧИСЛЮВАЛЬНІ ПРАВИЛА

Правило вибору розкриваних підтермів терма $C^i_{\Sigma,\bar{d}}$ ми будемо називати *обчислювальним правилом*.

Нижче розглядатимуться такі обчислювальні правила.

1. **PO (Parallel Outermost)** розкриваються всі самі зовнішні подтерми виду (34) (тобто не містяться ні в якому подтерме виду (34)).
2. **LO (Left Outermost)** розкривається найлівіший з найбільш зовнішніх подтермов виду (34).
3. **PI (Parallel Innermost)** розкриваються всі самі внутрішні подтерми виду (34) (тобто не містять подтермов виду (34)).
4. **LI (Left Innermost)** розкривається найлівіший з найбільш внутрішніх подтермов виду (34).

Ми будемо вважати, що символ C в (33) позначає обчислювальний правило, використовується при побудові цієї послідовності. Якщо це правило має спеціальне позначення, то ми можемо використовувати це позначення замість символу C. Тобто, наприклад, якщо при побудові термів обчислювальної послідовності використовується правило PO, то терми, що входять в дану послідовність, можуть позначатися записом:

$$\mathbf{PO}^{i}_{\Sigma,\bar{d}}$$

Спрощення терма

Спрощення терма, про який йдеться в пункті (2b) опису побудови обчислювальної послідовності (33), являє собою послідовність спрощуючих перетворень.

Для визначення поняття спрощуючих перетворень введемо такі допоміжні поняття.

1. Термы r и s называются **эквивалентными**, если

$$\forall f \in D_{type(\varphi)} \quad r[f/\varphi](\bar{x}) = s[f/\varphi](\bar{x}) \tag{36}$$

где $\bar{x}$ состоит из переменных, входящих в r и s.

Если r и s эквивалентны, то мы будем обозначать этот факт записью

$$r \equiv s.$$

2. Если терм v имеет вид

$$g(v_1, \ldots, v_n) \quad (g \in Fun) \tag{37}$$

то запись $\hat{v}$ обозначает терм

$$g(h_1, \ldots, h_n) \tag{38}$$

где $\forall\, i = 1, \ldots, n$

$$h_i \stackrel{\text{def}}{=} \begin{cases} v_i, & \text{если } v_i - \text{константа} \\ x_i & (\text{переменная того же типа,} \\ & \text{что и } v_i), \text{ иначе} \end{cases} \tag{39}$$

причём все переменные, входящие в (38), различны.

Ми будемо говорити, що терм s отриманий з терма r спрощуючим перетворенням, якщо s є результатом заміни в r подтерма v виду (37) на терм v 0, що дорівнює

• константі d, якщо v ≡ d, або

• терму vi, якщо hi змінна, і v ≡ hi.

Неважко бачити, що при даній заміні

• v 0 ≡ v, тому s ≡ r, і

• довжина v 0 менше довжини v, тому довжина s менше довжини r.

Терм r називається неспрощеним, якщо не існує терма, який виходить з r спрощує перетворенням.

Терм s називається спрощенням терма r, якщо
• s неспрощемий, і
• або s = r, або існує послідовність термів r1,. . ., rn, така, що r1 = r, rn = s, і ∀ i = 1,. . . , N - 1 ri + 1 отриманий з ri спрощуючим перетворенням.

Більш докладніше з питаннями перетворення термів та теорії функціональних програм можна познайомитися в роботі:

А. М. Миронов, Основные понятия теории функциональных программ, Интеллектуальные системы. Теория и приложения, 2016, том 20, выпуск 1, 79–180

На наступній лекції ми розглянемо сучасні напрямки розвитку функціональних мов програмування.