Секция 6

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

УДК 519.682 DOI 10.17223/2226308X/11/39

СИНТАКСИЧЕСКИЙ АНАЛИЗ ПРОГРАММ МЕТОДОМ ИНТЕГРАЛЬНЫХ ПРЕДСТАВЛЕНИЙ[1]

О. И. Егорушкин, И. В. Колбасина, К. В. Сафонов

Предложен новый метод синтаксического анализа мономов контекстно-свободного языка как модели языков программирования, основанный на интегральном представлении синтаксического полинома программы. При этом показано, что интеграл фиксированной кратности по циклу позволяет найти синтаксический полином монома (программы) с неограниченным числом символов, что даёт новый подход к проблеме синтаксического анализа. Предполагается, что интеграл по циклу может быть вычислен с помощью теории вычетов.

Ключевые слова: *формальный степенной ряд, коммутативный образ, синтаксический анализ, интегральное представление.*

Одной из важных проблем, связанных с разработкой систем и языков программирования, является проблема синтаксического анализа программ [1]. Как известно, большинство языков программирования является кс-языками, которые можно представить в виде формального степенного ряда (ФСР), поэтому каждая программа, написанная на языке программирования, может рассматриваться как моном соответствующего ФСР. В связи с этим рассмотрим проблему синтаксического анализа мономов кс-языка.

Для того чтобы сформулировать её, рассмотрим подробнее систему полиномиальных уравнений Хомского — Щутценберже, которая определяет кс-язык. Как известно [1, 2], грамматика кс-языка является множеством правил подстановки

$$z_j \to\ q_{j1}(z,x),\ \ldots,\ z_j \to\ q_{jp_j}(z,x),\ j=1,\ldots,n,$$

где $q_{jk}(z,x)$ является мономом от некоммутативных символьных переменных с числовым коэффициентом, равным единице. Правила подстановки можно применять к начальному символу z_1, а затем к другим мономам в любом порядке неограниченное число раз, что позволяют выводить новые «правильные» мономы, образующие кс-язык.

В [3] предложен метод мономиальных меток, который позволяет провести беступиковый синтаксический анализ монома v от терминальных символов $x_1,\ldots,x_m$. Метод состоит в следующем. Сначала каждое правило подстановки $z_j \to q_{jk}(z,x)$ заменяется правилом $z_j \to t_{jk}q_{jk}(z,x)$, имеющим мономиальную метку t_{jk}, которая является символом из расширенного алфавита, и для новых правил вывода рассматривается

[1]Исследование выполнено при финансовой поддержке РФФИ и Правительства Красноярского края в рамках научного проекта № 17-47-240318.

соответствующая система уравнений Хомского — Щутценберже:

$$z_j = Q_j^*(z,x,t) \overset{\mathrm{def}}{=} t_{j1}q_{j1}(z,x) + \ \ldots \ + t_{jp_j}q_{jp_j}(z,x), \ j=1,\ldots,n. \tag{1}$$

Решение этой системы можно получить методом последовательных приближений в виде ФСР, в том числе в виде ФСР представлена первая компонента решения

$$z_1 = z_1^*(x,t) = \sum_{i=0}^{\infty} \langle z_1^*, w_i\rangle \ w_i, \tag{2}$$

где w_i — мономы от символов $x_1,\ldots,x_m,t_{11},t_{12},\ldots,t_{np_n}$.

Синтаксический анализ монома v кс-языка $z_1(x)$ можно провести следующим образом. Считывая мономы степени $\deg_x(v)$ относительно символов $x_1,\ldots,x_m$ и пропуская символы t_{jk}, можно установить, есть ли среди них моном v, а значит, можно ли вывести его с помощью системы продукций. При этом каждая мономиальная метка t_{jk}, содержащаяся в таком мономе, показывает, что при его выводе использовалось правило $z_j \to t_{jk}q_{jk}(z,x)$. В самом деле, из системы уравнений (1) и метода последовательных приближений нетрудно видеть, что, применяя это правило вывода к моному, мы умножаем его слева на символ t_{jk}. Следовательно, мономиальные метки монома решают проблему его синтаксического анализа, показывая, какие правила вывода кс-языка и сколько раз использовались при выводе этого монома, с точностью до порядка их применения.

Информацию о мономиальных метках монома можно получить в виде $(n+m)$-кратного интеграла по циклу, где числа n и m не зависят от степени монома и равны числу нетерминальных и терминальных символов грамматики кс-языка соответственно.

Рассмотрим коммутативный образ [4] ФСР (2)

$$ci(z_1^*(x,t)) = \sum_{\alpha} s_\alpha(t)x^\alpha, \tag{3}$$

сгруппированный по степеням x^α в кратный ряд Гартогса [5, 6].

Назовём *синтаксическим полиномом* монома v относительно кс-языка $z_1(x) = z_1^*(x,e)$ коэффициент $s_\alpha(t)$ ряда Гартогса (3), такой, что $x^\alpha = ci(v)$.

Мономиальные метки, содержащиеся в некоммутативных мономах кс-языка, не исчезают при переходе от ФСР (2) к его коммутативному образу (3) и сохраняются в виде мономов синтаксических полиномов, поскольку все коэффициенты ФСР (3) являются целыми положительными числами [7, 8]. Следовательно, если синтаксический полином монома относительно кс-языка равен нулю, то моном не принадлежит этому языку.

Следующая теорема даёт принципиальную возможность получить синтаксические полиномы $s_\alpha(t)$ в виде кратного интеграла по циклу, который может быть вычислен с помощью многомерных вычетов.

Теорема 1. При всех t, достаточно близких к нулю, и всех мультииндексах α синтаксический полином $s_\alpha(t)$ задаётся равенствами

$$s_\alpha(t) = \frac{1}{(2\pi i)^{n+m}} \int\limits_{\gamma_z\times\Gamma_x} \frac{z_1 \det\bigl(\delta_{ij} - (ci(Q_i^*(z,x,t)))'_{z_j}\bigr) \ dz\wedge dx}{(z - ci(Q^*(z,x,t))) \ x^{\alpha+I}}; \tag{4}$$

$$s_\alpha(t) = \frac{1}{(2\pi i)^n \alpha!} \int\limits_{\gamma_z} \frac{\partial^{\|\alpha\|}}{\partial x^\alpha}\Bigl(\frac{z_1 \det\bigl(\delta_{ij} - (ci(Q_i^*(z,x,t)))'_{z_j}\bigr)}{(z - ci(Q^*(z,x,t)))}\Bigr)\Big|_{x=0} \ dz, \tag{5}$$

где $\gamma_z = \{|z_1| = \ldots = |z_n| = \varepsilon\}$ и $\Gamma_x = \{|x_1| = \ldots = |x_m| = \delta\}$ — циклы интегрирования; $0 < \delta \ll \varepsilon \ll 1$; $dz = dz_1 \wedge \ldots \wedge dz_n$; $dx = dx_1 \wedge \ldots \wedge dx_m$; $x^{\alpha+I} = x_1^{\alpha_1+1} \cdot \ldots \cdot x_m^{\alpha_m+1}$; $\alpha! = \alpha_1! \cdot \ldots \cdot \alpha_m!$; $(z - ci(Q^*(z,x,t))) = (z_1 - ci(Q_1^*(z,x,t))) \cdot \ldots \cdot (z_n - ci(Q_n^*(z,x,t)))$; $\dfrac{\partial^{\|\alpha\|}}{\partial x^\alpha} = \dfrac{\partial^{\alpha_1+\ldots+\alpha_m}}{\partial x_1^{\alpha_1} \cdot \ldots \cdot \partial x_m^{\alpha_m}}$; δ_{ij} — дельта Кронекера.

Формулы (4) и (5) позволяют эффективно осуществлять синтаксический анализ монома, находя его синтаксический полином. Так, кратный интеграл (4) можно вычислить как повторный, используя формулу Коши и разложение в степенной ряд подынтегральной функции.

ЛИТЕРАТУРА

1. *Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л.* Алгебра. Языки. Программирование. Киев: Наукова думка, 1973.
2. *Salomaa A. and Soitolla M.* Automata-Theoretic Aspects of Formal Power Series. N.Y.: Springer Verlag, 1978.
3. *Сафонов К. В., Егорушкин О. И.* О синтаксическом анализе и проблеме В. М. Глушкова распознавания контекстно-свободных языков Хомского // Вестник Томского государственного университета. 2006. Приложение № 17. С. 63–67.
4. *Семёнов А. Л.* Алгоритмические проблемы для степенных рядов и контекстно-свободных грамматик // Доклады АН СССР. 1973. № 212. С. 50–52.
5. *Safonov K. V.* On power series of algebraic and rational functions in C^n // J. Math. Analysis Appl. 2000. V. 243. P. 261–277.
6. *Сафонов К. В.* Об условиях алгебраичности и рациональности суммы степенного ряда // Матем. заметки. 1987. Т. 41. Вып. 3. С. 325–332.
7. *Егорушкин О. И., Колбасина И. В., Сафонов К. В.* О совместности систем символьных полиномиальных уравнений и их приложении // Прикладная дискретная математика. Приложение. 2016. № 9. С. 119–121.
8. *Egorushkin O. I., Kolbasina I. V., and Safonov K. V.* On solvability of systems of symbolic polynomial equations // Журн. СФУ. Сер. Матем. и физ. 2016. Т. 9. Вып. 2. С. 166–172.

УДК 004.4 DOI 10.17223/2226308X/11/40

РАБОТА СО СТЕКОМ В ЛЯПАСЕ[1]

М. С. Недяк, В. О. Сафонов

Описана работа со стеком в модульном трансляторе с ЛЯПАСа, которая включает в себя механизмы вызова функции, работу с локальными переменными и параметрами функции. Рассматриваются механизмы обработки композиции функций.

Ключевые слова: *ЛЯПАС, язык программирования, транслятор, соглашение о вызове.*

Введение

Рассмотрим модуль для работы со стеком модульного транслятора [1, 2] с ЛЯПАСа [3]. Далее этот модуль и его язык называются стекоязом.

В силу ограниченного количества переменных в языке, в текущей версии транслятора соглашение о вызовах выглядит следующим образом: перед вызовом функции

[1]Работа поддержана грантом РФФИ, проект № 17-01-00354.