1970 год ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ Выпуск 42

УДК 681.142.2.

О МЕТАЯЗЫКЕ СИНТАКСИЧЕСКИ УПРАВЛЯЕМОГО ТРАНСЛЯТОРА

И.В. Вельбицкий

Эффективность работы синтаксически управляемого транслятора в значительной степени определяется метаязыком, используемым для описания грамматик [1] . В данной работе обсуждается метаязык, ориентированный на применение в синтаксически управляемых трансляторах.

1° Пусть S – словарь синтермов [2] , элементы которого в дальнейшем, где это не оговорено, будут обозначаться малыми начальными буквами латинского алфавита; $\mathcal{A}$ – некоторый вспомогательный словарь, содержащий пустой символ, $\mathcal{A} = \{\emptyset, \nu_i\}_{i\in\Gamma}$, $\Gamma = \{1,\dots, \kappa\}$; $\mathcal{R}$ – множество имен некоторых конечных множеств (возможно, пустых) подмножеств правил вида:

$$\alpha \xrightarrow{W_m} z \qquad (1)$$

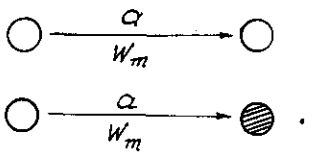
или

$$\alpha \xrightarrow{\overline{W_m}} z \ , \qquad (2)$$

где $\alpha\in S$; $z\in\mathcal{R}$; $\mathcal{R} = \{\emptyset, z_i\}_{i\in\Gamma_1}$, $\Gamma_1 = \{1,\dots, \ell\}$, $\emptyset$ – имя пустого множества правил; $\xrightarrow{W_m}$ или $\xrightarrow{\overline{W_m}}$ – многоместное отношение, означающее, что синтерм слева от него некоторым образом конкатенирует с синтермом левой части любого правила из множества, имя

которого указано справа от W_m – отношения; $W_m = W_m(\alpha_1, \alpha_2, \dots, \alpha_n)$, α_i – слово в словаре $\mathcal{A}\cup\mathcal{R}$, $i = 1,\dots, n$; конкатенация синтермов определяется типом m ($m = 0, 1, 2$) W_m-отношения. Число n аргументов W_m – отношения соответствует числу стеков, используемых при интерпретации правил грамматики. При этом предполагается, что на вершине пустых стеков помещены символы $\emptyset$; наличие символа $\emptyset$ в качестве аргумента W_m – отношения означает, что соответствующий этому аргументу стек не используется в интерпретации любого правила.

Для приложений удобно правила (1),(2) изображать соответственно в виде графов:

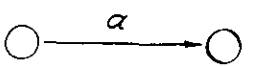


В этих графах дуга соответствует синтерму правила и несет информацию о W_m – отношении; граничные вершины дуги – именам множества правил; начало дуги – имени множества правил, которому принадлежит данное правило; конец дуги – имени множества правил преемников [3].

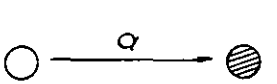
Правило вида (1) интерпретируется следующим образом:

А. При $m = 0$ (отношение нулевого типа) синтерм α может конкатенировать с синтермами, встречающимися в левых частях правил из множества с именем z ; тем самым данное правило определяет свои возможные правила – преемники из множества $\mathcal{R}$.

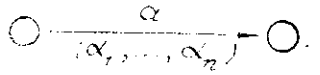
W_0 – отношение изображается графом:



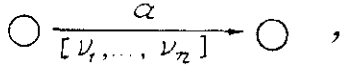
Если в качестве z стоит символ $\emptyset$, то α соответствует последнему символу предложения языка. В этом случае конец дуги графа выделяется (например, штриховкой):



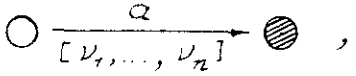
B. При $m = 1$ (отношение первого типа)указанная конкатенация и преемственность правил имеет место при одновременной записи непустых α_i в соответствующие i - е стеки. Очевидно, $W_1 \equiv W_0$, если все α_i из W_1 - пустые символы. W_1-отношение изображается графом:



C. При $m = 2$ (отношение второго типа) проверяется условие совпадения всех непустых символов $\nu_i \in \alpha$ с символами в вершинах соответствующих стеков. При выполнении условия происходит конкатенация, преемственность правил и выборка этих стеков. При отсутствии такого совпадения ни конкатенация, ни преемственность, ни выборка из соответствующих стеков не имеют места. W_2 - отношения изображаются графами:

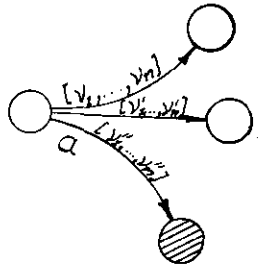


,



,

если α соответствует конечному символу некоторого предложения языка. Если дуги, выходящие из одной вершины, отличаются лишь аргументами при W_2 - отношении, то они изображаются в виде метелки:



Все сказанное о правилах вида (1) справедливо и для правил вида (2) с тем отличием, что синтерм α может быть конечным (может соответствовать конечному символу) некоторого предложения языка даже при t, отличном от $\emptyset$. При этом на графе конец соответствующей дуги выделяется (штриховкой).

Таким образом, интерпретация правил (1),(2) включает определения как допустимого множества конкатенаций с данным синтермом правила, так и множества правил преемников, а также указание на то, что данный синтерм может быть конечным.

1.1. Несколько слов о соответствии термов синтермам. Синтерм является элементом метаязыка, терм - языка. Соответствие между ними зададим функцией $F = F(t, \tau)$, где $t \in T$ - терм, T - терминальный словарь языка, τ - некоторый параметр соответствующего множества из $\mathcal{R}$. Каждому имени этого множества поставлен в соответствие некоторый индекс τ из множества индексов $\mathcal{M}$, который в дальнейшем будем записывать в виде верхнего индекса соответствующего имени из $\mathcal{R}$. На графе индекс τ записывается в вершине, инцидентной началу дуги.

Содержательно: каждому терму в зависимости от его места в предложении языка могут соответствовать различные синтермы. И, наоборот, в каждой локальной конструкции языка одному синтерму могут соответствовать несколько термов. Например, в АЛГОЛе - 60 символу "+" можно поставить в соответствие три различных синтерма, которые условно обозначим: "$+_1$", "$+_2$", "$+_3$".

Синтерму "$+_1$" соответствует конструкция языка, которую неформально можно назвать как:"перед арифметическим выражением". Этому синтерму соответствует два терма: "+", "-".

Синтерм "$+_2$" соответствует конструкции языка, которую можно определить как: " внутри арифметического выражения". Этому синтерму соответствует шесть термов" "+","-","x","/","÷","↑".

Синтерм "$+_3$" соответствует конструкции: " внутри комментария". Этому синтерму соответствует 115 термов: 52 буквы латинского алфавита (заглавные и прописные), 10 цифр, 29 ограничителей, 24 буквенных выражения.

Очевидно, что конструкции языка могут быть локализованы

лишь с помощью правил его грамматики. Естественно каждому множеству правил, соответствующему определенной конструкции языка, поставить в соответствие некоторый параметр τ, благодаря которому термы, имеющие одинаковое продолжение, можно обозначать одним синтермом и представлять одним правилом в соответствующем множестве правил $z^{\tau} \in \mathcal{R}$.

Таким образом, использование словаря синтермов, определяемого с помощью некоторой функции $F = F(t, \tau), t \in T, \tau \in \mathcal{M}$, позволяет существенно уменьшить число правил в каждом множестве z^{τ}, следовательно, во всем множестве $\mathcal{R}$. Это на практике позволяет компактно записывать грамматики исходных языков.

I.2. Выше было определено, что z^{τ} – множество подмножеств правил вида (I), (2). Будем в дальнейшем подмножество правил, совпадающее с некоторым множеством правил, имеющем имя z_i^{τ}, обозначать именем z_i. Следовательно, записи множеств:

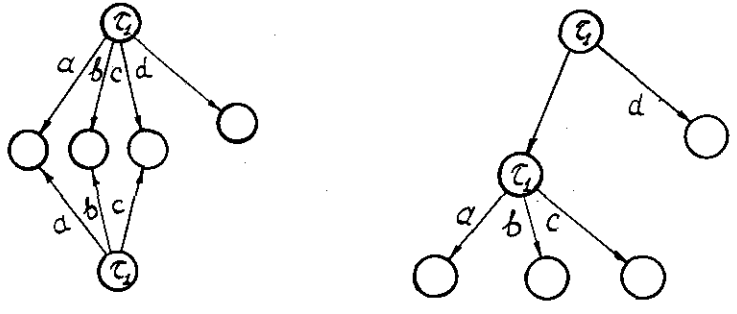
$$z_i^{\tau}: \{A_1, \ldots, A_p, A_{p+1}, \ldots, A_{p+q}\},$$

$$z_j^{\tau}: \{A_1, \ldots, A_p\},$$

$$z_i^{\tau}: \{z_j, A_{p+1}, \ldots, A_{p+q}\},$$

$$z_j^{\tau}: \{A_1, \ldots, A_p\}$$

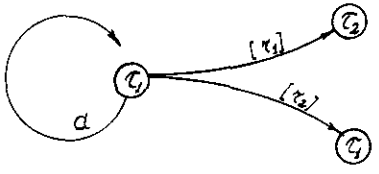
эквивалентны между собой, где $A_1, \ldots, A_{p+q}$ – правила вида (I), (2). На графе указанная ситуация изображается дугой без обозначений, направленной к вершине z_j^{τ}. Например, изображенные ниже графы эквивалентны:



I.3. Множество правил

$$z_i^{\tau *}: \{A_1, \ldots, A_q\},$$

элементами которого являются правила множества $\{A_1, \ldots, A_q, z\}$, где z – расположенное на вершине стековой памяти имя некоторого множества правил, назовем р а з о м к н у т ы м. На графе разомкнутым множествам соответствуют дуги, в обозначении которых отсутствуют синтермы, например:



Подводя итог, можно сказать, что грамматикой языка, неориентированной для синтаксического анализа (не содержащей определений семантики), понимается семерка:

$$G = (T, S, \mathcal{M}, \mathcal{A}, F, \mathcal{R}, z_0), \qquad (3)$$

где

T – терминальный словарь языка;

S – словарь синтермов;

$\mathcal{M}, \mathcal{A}$ – вспомогательные словари, $\mathcal{M} \cap \mathcal{A} = \emptyset$;

$F = F(t, \tau)$ – функция соответствия термов ($t \in T$) синтермам в зависимости от параметра $\tau \in \mathcal{M}$ грамматики языка;

$\mathcal{R} = \{z_i^{\tau}\}_{i \in I, \tau \in \mathcal{M}}$ – множество имен некоторых конечных множеств (возможно, пустых) подмножеств правил вида (I), (2);

z_0 – аксиома грамматики, некоторое выделенное начальное множество правил.

ПРИМЕР. Простое арифметическое выражение языка АЛГОЛ-60, в определении [4] которого: <первичное выражение> ::= <целое> <простая переменная> <(простое арифметическое выражение)> – задаётся следующей $\mathcal{R}$ – грамматикой

$$G = (T, S, \mathcal{M}, \mathcal{A}, F, R, z_0),$$

$T = \{a, \dots, z, A, \dots, Z, 0, 1, \dots, 9, +, -, \times, /, \div, \wedge,), (\}$,

$S = \{\delta, \delta_ц, ц, +, \times,), (\}$,

$m = \{1, 2, 3\}$, $\alpha = \phi$,

F – функция, которую можно задать таблицей вида:

τ \ T	a	b	…	y	z	A	B	…	Y	Z	0	1	…	9	+	-	×	/	÷	^	)	(
I	δ	δ	…	δ	δ	δ	δ	…	δ	δ	ц	ц	…	ц	+	+	x	x	x	x	)	(
2	δ	δ	…	δ	δ	δ	δ	…	δ	δ	ц	ц	…	ц	x	x	x	x	x	x	)	(
3	$\delta_ц$	$\delta_ц$	…	$\delta_ц$	$\delta_ц$	$\delta_ц$	$\delta_ц$	…	$\delta_ц$	$\delta_ц$	$\delta_ц$	$\delta_ц$	…	$\delta_ц$	x	x	x	x	x	x	)	(

$R = \{z_0', z_1', z_2^{3*}, z_3^{2*}, z_4^{2}, z_5', z_6', z_7', z_8^{3*}, z_9^{2*}, z_{10}^{2}, z_{11}'\}$,

$z_0' : \{+ \xrightarrow{W_0} z_1, z_1\}$,

$z_1' : \{\delta \xrightarrow{W_1(z_4)} z_2,\ ц \xrightarrow{W_1(z_4)} z_3, (\xrightarrow{W_1(z_5)} z_6)\}$,

$z_2^{3*} : \}\delta_ц \xrightarrow{W_0} . z_2\{$,

$z_3^{2*} : \}ц \xrightarrow{W_0} . z_3\{$,

$z_4^{2} : \{x \xrightarrow{W_0} z_1\}$,

$z_5' : \{) \xrightarrow{W_0} . z_4\}$,

$z_6' : \{+ \xrightarrow{W_0} z_7, z_7\}$,

$z_7' : \{\delta \xrightarrow{W_1(z_{10})} z_8,\ ц \xrightarrow{W_1(z_{10})} z_9,\ (\xrightarrow{W_1(z_{11})} z_6\}$,

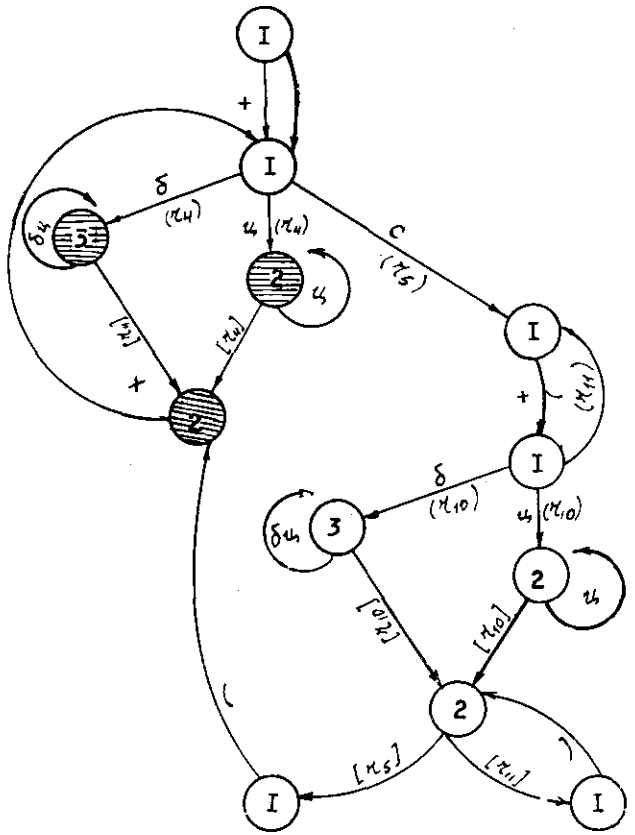
$z_8^{3*} : \}\delta_ц \xrightarrow{W_0} z_8\{$,

$z_9^{2*} : \}ц \xrightarrow{W_0} z_9\{$,

$z_{10}^{2*} : \}x \xrightarrow{W_0} z_7\{$,

$z_{11}' : \{) \xrightarrow{W_0} z_{10}\}$.

Графически эта же грамматика изображена на рис.I.



τ \ T	a	b	…	y	z	A	B	…	Y	Z	0	1	…	9	+	-	×	/	÷	^	)	(
I	δ	δ	…	δ	δ	δ	δ	…	δ	δ	ц	ц	…	ц	+	+	x	x	x	x	)	(
2	δ	δ	…	δ	δ	δ	δ	…	δ	δ	ц	ц	…	ц	x	x	x	x	x	x	)	(
3	$\delta_ц$	$\delta_ц$	…	$\delta_ц$	$\delta_ц$	$\delta_ц$	$\delta_ц$	…	$\delta_ц$	$\delta_ц$	$\delta_ц$	$\delta_ц$	…	$\delta_ц$	x	x	x	x	x	x	)	(

Рис. I.

Граф фрагмента простого арифметического выражения языка АЛГОЛ-60.

2^0. Рассмотрим некоторые свойства $\mathcal{R}$ – грамматики. Пусть для простоты $S = T$. Цепочка символов $\mathcal{Y}=(a_1,a_2,\dots,a_n)$ является W – выводом $\mathcal{Y}$ (обозначение $a_1 \Rightarrow \mathcal{Y}$), если существует последовательность правил преемников, порождающих эту цепочку, начиная от a_1. Если при этом a_k порождается правилом вида (I) с символом ϕ в правой части или правилом вида (2), то W – вывод $\mathcal{Y}$ называется законченным и обозначается $a_1 \models\!\Rightarrow \mathcal{Y}$.

Язык, порождаемый $\mathcal{R}$ – грамматикой (3):

$$\mathcal{L}(G)=\{\mathcal{Y}\mid \mathcal{Y}=a_1,\dots,a_n;\ a_i\in T,\ 1\le i\le n,\ a_1\in\tau_0,\ a_1 \overset{W}{\Rightarrow} \mathcal{Y}\}.$$

Терминальные символы a_j и a_{j+k} $(k>0)$ в цепочке $\mathcal{Y}=(a_1\dots a_j\dots a_{j+k})$, $a_1\in\tau_0$, порождаемые правилами с $W_1=(\nu_1,\dots,\nu_n)$ и $W_2(\nu_1',\dots,\nu_n')$, соответственно, называются базовым начальным и базовым конечным для i – го стека, если $\nu_i=\nu_i'$, $1\le i\le n$.

Правило $\mathcal{R}$ – грамматики называется фиктивным, если не существует законченного W – вывода $\mathcal{Y}$, в котором данное правило применялось по крайней мере один раз. Грамматика, не содержащая фиктивных правил, называется приведенной. $\mathcal{R}$ – грамматика является детерминированной, если терминальные символы каждого множества правил совпадают лишь в правилах, содержащих W_2 с различными наборами аргументов.

Выделим интересные для практических приложений классы детерминированных $\mathcal{R}$ – грамматик:

I. монарные, мощность всех множеств правил которых равна I;

2. односторонние линейные, для всех правил которых $m = 0$;

3. левоконтекстные, для всех правил которых $m \neq 2$;

4. с ранним определением синтерма, у которых все подмножества любого из множеств τ_i правил имеют одинаковый признак $\tilde{c}$;

5. с поздним определением синтерма, у которых хотя бы два подмножества любого из множеств τ_i правил имеют различные признаки $\tilde{c}$;

6. последовательные, у которых элементы множества $\mathcal{R}$ могут быть упорядочены так, что при любом порождении преемников может быть либо правило из этого же множества, либо из любого следующего в упорядоченной последовательности множеств правил.

Монарные грамматики допускают построение универсального алгоритма исправления одиночных ошибок, однако они порождают весьма узкий класс языков.

Односторонние линейные $\mathcal{R}$ – грамматики позволяют проводить синтаксический анализ без стековой памяти.

Левоконтекстные $\mathcal{R}$ – грамматики допускают построение эффективного алгоритма локализации синтаксических ошибок и позволяют компактно описать довольно широкий класс языков.

$\mathcal{R}$ – грамматики с ранним определением синтерма позволяют строить более быстрые схемы синтаксического анализа по сравнению с $\mathcal{R}$ – грамматиками с поздним определением синтерма.

Последовательные $\mathcal{R}$ – грамматики допускают весьма простые технические средства для реализации синтаксического анализа.

3. Пусть $\mathcal{U}$ – множество информатив, некоторое множество стандартных подпрограмм и таблиц, однозначно описывающих некоторый язык $\mathcal{L}_i$, $\mathcal{A}$ – множество правил $\mathcal{R}$ – грамматики, задающей язык $\mathcal{L}_{i-1}$; $\mathcal{D}$ – множество директив, задающих отображение множества $\mathcal{A}$ в $\mathcal{U}$. Тогда под семантическим описанием правил $\mathcal{R}$ – грамматики будем понимать некоторое множество $C\subseteq(\mathcal{D}\times\mathcal{U})$, элементы которого поставлены во взаимно однозначное соответствие элементам множества $\mathcal{A}$. Под полным описанием $\mathcal{R}$ – грамматики языка понимается восьмерка

$$G=(T,S,\mathcal{M},\mathcal{A},F,\mathcal{R},\tau_0,C),$$

каждое правило которой имеет вид:

$$a \xrightarrow{W_m} \tau\,|\,(d,u) \qquad (4)$$

или

$$a \xrightarrow{W_m} \tau\,|\,(d,u), \qquad (5)$$

где $d\in\mathcal{D}$, $u\in\mathcal{U}$ из C.

Приведенная $\mathcal{R}$ – грамматика весьма эффективна при распознавании в синтаксически управляемых устройствах. Действительно, последовательность символов является предложением языка, если синтерм, соответствующий ее первому символу, является символом, встречающимся в левой части одного из правил – преемников примененного правила, а последний символ встречается в одном из

правил вида (5) или в правиле вида (4) с символом ϕ в правой части. На каждом этапе распознавания текущему символу исходной последовательности, или текущему символу, и некоторой последовательности предшествующих символов по правилам, предписанным в α, ставится в соответствие последовательность символов из ω, где α и ω-директивное и информативное описания семантики правила $\mathcal{R}$ - грамматики, применяемого к текущему символу исходной последовательности.

Транслятор строится таким образом, что каждому его блоку ставится в соответствие как некоторый параметр $\mathcal{R}$ - грамматика входного для этого блока языка. Блоки работают последовательно. Каждый символ исходного предложения последовательно "проходит" через блоки транслятора до блока, в котором либо заканчивается его обработка, либо происходит накопление для последующих этапов трансляции. Затем выбирается следующий символ исходного предложения (управление передается первому блоку) и т. д. Все блоки на всех этапах трансляции работают по одной и той же программе синтаксического анализа, которая является основной программой всего транслятора. По окончании работы i - го блока, и включении (i + I)-го или первого в i - м блоке формируется управляющее слово о состоянии программы синтаксического анализа в момент включения. Включение (i + I) - го блока- это настройка общей программы синтаксического анализа с помощью соответствующего этому блоку управляющего слова. Каждый блок, кроме синтаксического анализа, может доопределить текущий символ семантически. Благодаря этому сокращаются $\mathcal{R}$ - грамматики последующих блоков, упрощаются соответствующие им информативные и директивные стандартные подпрограммы.

При многопроходной схеме трансляция ведется последовательно, несколькими итеративно работающими блоками. Важно отметить, что и в этом случае все блоки при всех проходах работают по одной и той же программе синтаксического анализа.

Описанный метаязык позволяет весьма компактно представлять грамматики существующих алгоритмических языков. Так, например, грамматика языка АЛГОЛ - 60 задается приблизительно 250 правилами (машинными словами). Программа синтаксического анализа, реализующая распознавание правил $\mathcal{R}$ - грамматики в процессе работы синтаксически управляемого транслятора, содержит около I00 команд и работает со скоростью нескольких десятков команд на каждый символ исходного предложения.

Л И Т Е Р А Т У Р А

I. И.В.ВЕЛЬБИЦКИЙ, Ю.Л.ЮЩЕНКО. Метаязык, ориентированный для синтаксического анализа и контроля. Кибернетика, № 2, I970.

2. И.В.ВЕЛЬБИЦКИЙ. Первичные средства подготовки синтаксически правильных программ. Труды II Всесоюзной конференции по ВС., М., I969.

3. К.БЕРЖ. Теория графов и её применение. Изд-во Иностранной литературы, М., I962.

4. Алгоритмический язык АЛГОЛ-60 (пересмотренное сообщение). М., Изд-во "Мир", I965.

Поступила в редакцию
20 июня I970 г.