

ПОСТРОЕНИЕ TR-ГРАММАТИК ПО ГРАММАТИКАМ В БЕКУСОВО-НАУРОВСКОЙ ФОРМЕ

В.И. Константинов

Вопросы синтаксического анализа входных языков занимают одно из центральных мест в современной теории и практике автоматизации программирования. Процесс синтаксического анализа существенно зависит от метаязыка, используемого для задания грамматик входных языков.

В [1,2] разработан метаязык R -грамматик, допускающий простые и быстро работающие алгоритмы синтаксического анализа. Однако процесс перехода от общепринятой бекусово-науровской формы (НФ) записи грамматик языков к заданиям их R -грамматикам оказался трудоемким.

Действительно, алгоритм синтеза R -грамматик [3] по НФ включает следующие этапы:

- переход от НФ к модифицированной НФ (МНФ);
- выделение канонической формы МНФ (КФ);
- введение A -термов;
- определение базовых A -термов КФ (МНФ) грамматики;
- построение для каждого начального термина множества θ -цепочек и т.д.

Ввиду этого была предложена [4] новая форма задания грамматик языков - запись их TR-грамматиками, которые являются расширением R -грамматик на нетерминальный алфавит. Переход от НФ к TR-грамматикам проще, чем к R -грамматикам.

1. Под грамматикой в НФ будем понимать четверку

$$G = (A, V_N, S_0, P),$$

где $A = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ - терминальный алфавит,

$V_N = \{S_0, S_1, \dots, S_m\}$ - нетерминальный алфавит, $A \cap V_N = \emptyset$;

S_0 - аксиома грамматики языка (начальный символ);

P - множество правил вида $S_i ::= \beta$, где β - цепочка над алфавитом $A \cup V_N$, то есть $\beta \in (A \cup V_N)^*$, или множество цепочек над $A \cup V_N$, разделенных знаком $|$ (исключительно "или"), то есть $\beta = \beta_1 | \beta_2 | \dots | \beta_p$; $\beta_i \in (A \cup V_N)^*$, $1 \leq i \leq p$.

Пусть $x, y \in (A \cup V_N)^*$. Скажем, что из x следует y ($x \rightarrow y$), если существуют $z_1, z_2, \beta \in (A \cup V_N)^*$ и $S \in V_N$ такие, что $x = z_1 S z_2$, $y = z_1 \beta z_2$, $S ::= \beta$ - правило из P .

Скажем, что из x выводимо y ($x \Rightarrow y$), если существует цепочка $w_0, \dots, w_n$ над алфавитом $A \cup V_N$ такая, что $w_0 \Rightarrow w_1, \dots, w_{n-1} \Rightarrow w_n$, где $w_0 = x$, $w_n = y$.

Множество $L(G) = \{x \in A^* / S_0 \Rightarrow x\}$ назовем языком, порожденным грамматикой G .

Ограничим класс грамматик в НФ, для которых строятся ТР-грамматики, грамматиками, не содержащими правил вида $S ::= \varepsilon$, ε - пустая цепочка. Ограничение это несущественно, т.к. в [5] показана эквивалентность контекстно-свободных (КС-языков) и языков, порождаемых грамматиками в НФ (в [5] они названы языками типа АЛГОЛ); там же показано, что если G - КС-грамматика и $\varepsilon \in L(G)$, то по грамматике G может быть эффективно построена КС-грамматика, порождающая язык $L(G)$, но не содержащая правил вида $S ::= \varepsilon$. Грамматики в НФ, не содержащие правил вида $S ::= \varepsilon$, назовем приведенными. Исключение правил вида $S ::= \varepsilon$ связано с необходимостью устранения недетерминированности, возникающей в таких грамматиках при анализе входной строки.

Действительно, пусть G содержит правила:

$$S_1 ::= \varphi S_2 \psi,$$

$$S_2 ::= \varepsilon | \alpha_1 | \dots | \alpha_n,$$

причем одно из α_i таково, что $\alpha_i \Rightarrow \varphi \beta \psi$, $S_2 \in V_N$, $\varphi, \alpha_1, \dots, \alpha_n, \beta \in (A \cup V_N)^*$, $\varphi \in A^*$, $\varphi, \alpha_i, \psi \neq \varepsilon$. Тогда при анализе цепочки $\varphi \psi$ в момент прихода $A \Rightarrow \alpha$ - первого элемента цепочки φ , нельзя сказать, следует ли распознавать его в S_2 , или остаться в S_1 .

П. В качестве исходной формы задания грамматик входных языков предлагается, как и в [6], модифицированная НФ (МНФ). Задача построения по произвольной НФ соответствующей МНФ алгоритмически разрешима, алгоритм перехода подробно описан в [5]. Отметим, что по сравнению с НФ:

а) в МНФ отсутствует левосторонняя рекурсия (то есть не существует $S \in V_N$ и непустой цепочки $\varphi \in (A \cup V_N)^*$ такой, что $S \Rightarrow S\varphi$; она исключена с помощью итерационных скобок);

б) никакие два правила не имеют совпадающих левых частей.

ПРИМЕР применения итерационных скобок. $S ::= S\varphi | \psi$ с помощью итерационных скобок можно записать в виде $S ::= \psi \{ \varphi \}$, где $\{ \varphi \}$ означает, что φ может быть повторено нуль или более раз.

Ш. Рассмотрим грамматику в МНФ, не содержащую правил с вложенными итерационными скобками (то есть правил вида $\{ \{ \varphi \} \{ \psi \} \}$). Пусть она задается множеством правил вида:

$$S_0 ::= \alpha_0, S_1 ::= \alpha_1, \dots, S_m ::= \alpha_m,$$

где $S_i \in V_N$, $0 \leq i \leq m$, α_i есть

а) либо цепочка, принадлежащая $(A \cup V_N)^* \setminus \varepsilon$;

б) либо цепочка, принадлежащая $(A \cup V_N)^* \setminus \varepsilon$, некоторые подцепочки которой заключены в $\{ \}$;

в) либо цепочки вида а)-б), разделенные знаком $|$.

С каждым нетерминальным символом S связана атомная ТР-грамматика [4]: $G(S) = (A, V_N, P, R^{G(S)}, z_0^{G(S)}, R_{G(S)}^*)$.

Интерпретация правил ТР-грамматик подробно описана в [4]. Поскольку каждое правило с нетерминальной левой частью интерпретируется с помощью стека, переход от грамматик в НФ к грамматикам в МНФ сокращает стековые операции анализатора и гарантирует от закликивания (между любыми двумя обращениями к одной и той же грамматике будет считан по крайней мере один символ входной строки).

Исходной грамматике G , заданной системой (I), соответствует - вует система атомных ТР-грамматик $G(S_0), \dots, G(S_m)$. Построить ТР-грамматику G - значит построить образующие её грамматики $G(S_i)$, $0 \leq i \leq m$.

Возьмем любое правило из (I). Пусть это правило имеет вид $S ::= \beta_1 | \beta_2 | \dots | \beta_n$, β_i - цепочки из $(A \cup V_N)^* \setminus \varepsilon$, некоторые подцепочки которых могут быть заключены в итерационные скобки.

Алгоритм построения атомной ТР-грамматики $G(S)$ есть следующая последовательность действий.

I. Определение имени формируемого множества правил. Для грамматики $G(S)$ на первом шаге это имя есть $z_0^{G(S)}$ (аксиома грамматики $G(S)$).

2. Выбор первых элементов цепочек $\beta_1, \dots, \beta_n$; пусть это будут $\alpha_1, \dots, \alpha_k$. Так как для цепочек β_i вида $\{\theta_1\} \dots \{\theta_p\}$ ψ первыми элементами являются начальные элементы цепочек $\theta_1, \dots, \theta_p$, ψ число k может превосходить n .

Сформируем множество M , в которое попадут все различные элементы из $\alpha_1, \dots, \alpha_k$. Множество M назовем множеством левых частей формируемого множества правил (на первом шаге это множество левых частей правил из $\tau_0^{(S)}$, что обозначим $M(\tau_0^{(S)})$).

Число элементов в M может меняться от 1 (когда все $\alpha_1, \dots, \alpha_k$ совпадают) до k (когда они все различны). Элемент $m \in M$ назовем элементом кратности ℓ , если он ℓ раз, $1 \leq \ell \leq k$, встречается в последовательности $\alpha_1, \dots, \alpha_k$.

3. Определение типов правил из $\tau_0^{(S)}$, для чего анализируются подцепочки, следующие за m . Возможны следующие случаи:

- m либо $m|$, то есть m — единственный элемент цепочки;
- $m\psi, \psi \in (A \cup V_N)^* \setminus \varepsilon$;
- $m\{\psi_1\} \dots \{\psi_n\}$ либо $m\{\psi_1\} \dots \{\psi_n\}|, n \geq 1, \psi_i \in (A \cup V_N)^* \setminus \varepsilon$;
- $m\{\psi_1\} \dots \{\psi_n\}\psi, \psi_i, \psi \in (A \cup V_N)^* \setminus \varepsilon, n \geq 1$;
- $m|\{\psi_1\} \dots \{\psi_n\}\psi, n \geq 1, \psi, \psi_i \in (A \cup V_N)^* \setminus \varepsilon$.

Кроме того, если m не есть первый элемент цепочки (то есть $m \in M(\tau_0^{(S)})$), следует добавить еще

- $m|\{\psi_1\} \dots \{\psi_n\}$ либо $m|\{\psi_1\} \dots \{\psi_n\}|, \psi_i \in (A \cup V_N)^* \setminus \varepsilon, n \geq 1$.

Если m — кратности 1, то в случаях а), б), г), д) правило с левой частью m есть правило первого типа, в остальных случаях — правило второго типа. Если формируемое правило второго типа, множество правил-преемников его разомкнуто.

Если кратность $m > 1$, то имеется несколько подцепочек (число их совпадает с кратностью m), следующих за m . В этом случае тип правила определяется следующим образом:

— если все подцепочки, начинающиеся с m , есть α -подцепочки, формируемое правило — первого типа, при этом имя множества правил-преемников данного правила τ_φ ;

— если все подцепочки, начинающиеся с m , есть б-, либо г-, либо д-подцепочки, соответствующее правило есть правило первого типа. Во всех остальных случаях формируемое правило есть правило второго типа, а множество правил-преемников его разомкнуто.

4. Определение правил-преемников формируемого множества правил. Если m имеет кратность 1, множество правил-преемников его состоит из:

- правила с первым элементом подцепочек ψ в случае б);
- правил с первыми различными элементами подцепочек $\psi_1, \dots, \psi_n$ в случае в);
- правил с первыми различными элементами подцепочек $\psi_1, \dots, \psi_n, \psi$ в случае г).

В случае а) множество правил-преемников пусто, имя его — τ_φ . Отдельно рассмотрим ситуацию, когда за m следует $\}$, что возможно в случаях д) и е).

Пусть m — последний элемент цепочки ψ_0 , тогда множество правил-преемников формируемого правила состоит из:

- правил с первыми различными элементами подцепочек $\psi_0, \psi_1, \dots, \psi_n, \psi$ в случае д);
- правил с различными элементами подцепочек $\psi_0, \psi_1, \dots, \psi_n$ в случае е).

Заметим, что цепочка ψ_0 один раз уже пройдена, т.к. m — последний элемент этой цепочки, но в общем случае необходимо пройти по ней еще один раз. Действительно, пусть имеется, например, цепочка $\psi\{\psi_1\}\{\psi_2\}\{\psi_3\}\psi$ — и пусть m — последний элемент ψ_2 .

Положим $\psi_1 = S\psi'_1, \psi_2 = S\psi'_2$, тогда первое прохождение по цепочке ψ_2 при определении преемников правила с левой частью ψ (в — последний элемент цепочки ψ) совмещено с прохождением цепочки ψ_1 , поэтому, хотя цепочка ψ_2 уже пройдена, нельзя воспользоваться полученной ранее цепочкой правил, так как, например, правило с левой частью S будет иметь преемников и из ψ'_1 , и из ψ'_2 . Второй проход по цепочке ψ_2 из последовательности $\psi\{\psi_0\} \dots \{\psi_n\}$ совмещается с прохождением по цепочкам $\psi_{i+1}, \dots, \psi_n$. При этом множество левых частей правил-преемников последнего элемента цепочки ψ_0 — то же, что и последнего элемента цепочки ψ , а последнего элемента цепочки $\psi_i, i \geq 1$, — то же, что при первом проходе ψ_i .

Если известно, что в последовательности $\psi\{\psi_1\} \dots \{\psi_n\}\psi$ подцепочки $\psi_1, \dots, \psi_n, \psi$ начинаются с разных элементов, процедура нахождения множеств правил-преемников существенно упрощается. Пусть τ_m — имя множества правил-преемников последнего элемента цепочки ψ . Тогда множество правил-преемников последнего элемента цепочки ψ_1 есть $(\tau_m, 1)$, множество правил-преемников последнего элемента цепочки ψ_2 есть $(\tau_m, 2)$ и т.д. Индекс j показывает, начиная с какого по счету правила множества правил τ_m стоят правила-преемники данного правила.

Если $m \in M$ имеет кратность > 1 (скажем, ℓ), считаем, что имеется ℓ элементов кратности 1, для которых определяем множества правил-преемников так, как это указано в п.4, после чего берем пересечение этих множеств правил и объявляем его множеством правил-преемников с левой частью m .

Выполнив п.4, мы сформировали новые множества M . Присвоив очередные имена из числа $z_1^{G(S)}, \dots, z_n^{G(S)}$ соответствующим множествам правил, переходим на п. 3. Так как правая часть правила $S ::= \beta_1 | \dots | \beta_n$ состоит из конечного числа элементов, за конечное число шагов TR -грамматика $G(S)$ будет построена. Затем строим следующую атомную грамматику и т.д., до полного построения $G = \bigcup_{i=0}^{\infty} G(S_i)$.

IV. В заключение рассмотрим простой пример, иллюстрирующий приведенный алгоритм.

Пусть дана грамматика $G = \{A, V_N, S_0, P\}$, где $A = \{a, b, c\}$, $V_N = \{X, Y\}$, $S_0 = X$, $P = \{X ::= Xa | Y, Y ::= b | c\}$. Соответствующая МНФ имеет вид: $X ::= Y\{a\}$, $Y ::= b | c$.

Построим TR -грамматику $G(X)$.

1. Определим имя формируемого множества правил - это $z_0^{G(X)}$.

2. Построим $M(z_0^{G(X)})$. Множество это состоит из элемента $Y \in V_N$ кратности 1, то есть $z_0^{G(X)}$ содержит единственное правило, левая часть которого Y .

3. Определим тип формируемого правила. Цепочка $Y\{a\}$ - типа b , поэтому формируемое правило - второго типа: $z_0^{G(X)} = \{Y \rightarrow \cdot\}$, множество правил-преемников его разомкнуто. Присвоим ему имя $z_1^{G(X)}$, $z_0^{G(X)} = \{Y \rightarrow \cdot, z_1^{G(X)}\}$.

4. Сформируем $z_1^{G(X)}$. Как отмечалось, множество это разомкнуто, поэтому выделим его в соответствии с $[I]$, в $\{ \}$. Оно состоит из правила с левой частью a , $z_1^{G(X)} = \{a\}$. Поскольку a - последний элемент цепочки, стоящей в $\{ \}$ (случай е), данное правило - второго типа: $z_1^{G(X)} = \{a \rightarrow \cdot\}$. Так как a - последний элемент в единственной из цепочек типа $\{y_1\} \dots \{y_n\}$, второй проход по цепочке не нужен, а множество правил-преемников правила с левой частью a - то же, что и Y , то есть $z_1^{G(X)}$. Получаем, таким образом, что TR -грамматика $G(X)$ есть:

$$G(X): \quad \begin{aligned} z_0^{G(X)} &= \{Y \rightarrow \cdot, z_1^{G(X)}\} \\ z_1^{G(X)} &= \{a \rightarrow \cdot, z_1^{G(X)}\} \end{aligned}$$

Построим теперь $G(Y)$.

1. Имя формируемого множества правил - $z_0^{G(Y)}$.

2. $M(z_0^{G(Y)}) = \{b, c\}$, это элементы кратности 1, $z_0^{G(Y)}$ состоит из двух правил с левыми частями b и c .

3. Определим типы этих правил. Так как подцепочки, начинающиеся с b, c есть α -подцепочки, данные правила - первого типа, а имена их правил-преемников - $z_\emptyset$. Получаем

$$G(Y): \quad z_0^{G(Y)} = \{b \rightarrow z_\emptyset, c \rightarrow z_\emptyset\}.$$

Таким образом, соответствующая G система атомных грамматик имеет вид:

$$\begin{aligned} G(X): \quad & z_0^{G(X)} = \{Y \rightarrow \cdot, z_1^{G(X)}\} \\ & z_1^{G(X)} = \{a \rightarrow \cdot, z_1^{G(X)}\} \\ G(Y): \quad & z_0^{G(Y)} = \{b \rightarrow z_\emptyset, c \rightarrow z_\emptyset\}. \end{aligned}$$

Л и т е р а т у р а

1. ВЕЛЬИЦКИЙ И.В. О метаязыке синтаксически управляемого транслятора. - "Вычислительные системы", Новосибирск, 1970, вып.42, с. 22-33.

2. ВЕЛЬИЦКИЙ И.В., ИЩЕНКО Е.Л. Метаязык, ориентированный для синтаксического анализа и контроля. - "Кибернетика", Киев, 1970, №2, с. 50-53.

3. ВЕЛЬИЦКИЙ И.В. К вопросу перехода от бэкусово-науровской формы записи грамматик к грамматикам R -типа. - Автоматизация программирования, Киев, 1969, № 3, с. 43-59.

4. КОНСТАНТИНОВ В.И., НУРИЕВ Р.М. Метаязык трансляции контекстно-свободных языков. Настоящий сборник, с.74-83.

5. ГИНЗБУРГ С. Математический анализ контекстно-свободных языков. М., "Мир", 1970.

6. ВЕЛЬИЦКИЙ И.В., МЕДВЕДЕВА В.Н., ИЩЕНКО В.И. О формальном описании синтаксиса языка АЛГОЛ-60 и транслятора ТЛ-2. - Автоматизация программирования, Киев, 1969, № 3, с. 85-92.

Поступила в ред.-изд.отд.

6 февраля 1973 года