АЛГОРИТМ РАСПОЗНАВАНИЯ БЕСКОНЕЧНЫХ ПРАВИЛ ДЛЯ КОНТЕКСТНО-СВОБОДНОЙ ГРАММАТИКИ

С. Б. КАЛАЧЕВА

Санкт-Петербургский институт информатики и автоматизации РАН

СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург, 199178

<svet@iias.spb.su>

УДК 681.3

Калачева С. Б. **Алгоритм распознавания бесконечных правил для контекстно-свободной грамматики** // Труды СПИИРАН. Вып. 3, т. 2. — СПб.: Наука, 2006.

Аннотация. *Приводится и обосновывается алгоритм распознавания бесконечных правил для контекстно-свободной грамматики.* — Библ. 2 назв.

UDC 681.3

Kalacheva S. B. **Formal Algorithm of Recognition of the Infinite Rules for the Context Free Grammars** // SPIIRAS Proceedings. Issue 3, vol. 2. — SPb.: Nauka, 2006.

Abstract. *Formal algorithm of recognition of the infinite rules for the context free grammars is presented and substantiated.* — Bibl. 2 items.

1. Введение

Контекстно-свободную грамматику будем называть КС-грамматикой. Общеизвестно определение грамматики как четверки множеств $G=(N,T,P,S)$ [1], где N — множество нетерминальных символов, или нетерминалов, T — множество терминальных символов, или терминалов, P — множество правил, S — начальный символ грамматики. Дополним определение множеством семантик S^*, получим грамматику $G^*=(N,T,S^*,P,S)$ [2].

Бесконечным правилом называется правило, которое не порождает конечной цепочки из терминальных символов и семантик в качестве значения стоящего в левой части правила нетерминала.

2. Алгоритм нахождения бесконечных правил

Ниже предлагается простой алгоритм выявления бесконечных правил. Покажем, что алгебраическая структура $(G^*,*,<\textit{пробел}>,|)$ изоморфна структуре $(B^*,\backslash,\wedge,\vee)$, где $B^*=(B,\{1\},\{1\},P_b,S_b)$, в которой B — множество булевых переменных, сопоставляемых нетерминалам $n\in N$ и принимающих значение $\{0,1\}$, $S_b\in B$ — начальный символ, P_b — множество правил, которые строятся по множеству правил P грамматики G^*. Введем функцию, отображающую

$G^*\to B^*$ (поэлементно),

операцию итерации:

$$n_1 * n_2 \to b_1 \backslash b_2, n_1, n_2 \in N\cup T\cup S^*, b_1, b_2 \in B\cup\{1\},$$

Труды СПИИРАН. 2006. Вып. 3. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

операцию конкатенации:

$n_1 < \textit{пробел} > n_2 \to b_1 \wedge b_2, n_1, n_2 \in N \cup T \cup S^*, b_1, b_2 \in B \cup \{1\}$,

операцию дизъюнкции:

$n_1 \mid n_2 \to b_1 \vee b_2, n_1, n_2 \in N \cup T \cup S^*, b_1, b_2 \in B \cup \{1\}$.

В грамматике G^* заменяются нетерминалы $n \in N$, а переменные $b \in B$, терминальные символы $t \in T$ и семантики $s \in S^*$ заменяются на 1, затем P — множество правил КСР-грамматики — заменяется на множество P_b, причем бинарные операции $\{*, < \textit{пробел} >, |\}$ заменяются на $\{\backslash, \wedge, \vee\}$ соответственно.

Пример.

Исходная грамматика:

```
R1:t|$s<пробел>t*R2;
R2:R1|t<пробел>(p*R2|R1);
```

где $R1, R2 \in N, t \in T, \$s \in S^*$.

Замена:

$b1 : 1 \vee 1 \wedge 1$,

$b2 : b1 \vee 1 \wedge (1 \vee b1)$.

Алгоритм.

1) Построить правила для изоморфных булевых множеств.

2)Всем булевским переменным присвоить значение ноль: $\forall b_i \in B : b_j = 0$, $0 \le j < n$, n — число правил.

3) Подставить значения b_i в правила, построенные по исходным правилам и посчитать по их правым частям новые значения.

4) Проанализировать новые полученные значения. Если хотя бы одно значение изменилось, перейти к пункту 3) (вычисляются только те значения b_i, которые имеют значение 0).

5) Процесс заканчивается, когда все значения b_i остаются прежними.

3. Пример применения алгоритма

Рассмотрим следующий фрагмент грамматики.

```
01 Program: [ | '-' ] Expression ;
02 Expression: Expression '+' Expression |
03      Expression '-' Expression  |
04      Expression '*' Expression |
05      Expression '/' Expression |
06      '(' [ | '-' ] Expression ')';
07 Decimal_number: Unsigned_integer |
                        Unsigned_integer Decimal_fraction ;
08 Decimal_fraction: '.' Unsigned_integer;
09 Unsigned_integer: Digit ' ';
10 Digit: '0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9';
```

Очевидно, что правило, начинающееся в строке 2 бесконечно. Рассмотрим работу алгоритма.

Труды СПИИРАН. 2006. Вып. 3. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

1)Правила для изоморфных булевых множеств:

$b1 : [1 \vee 1] \wedge b2;$

$b2 : b2 \wedge 1 \wedge b2 \vee b2 \wedge 1 \wedge b2 \vee b2 \wedge 1 \wedge b2 \vee b2 \wedge 1 \wedge b2 \vee 1 \wedge [1 \vee 1] \wedge b2;$

$b3 : b4 \vee b4 \wedge b5;$

$b4 : 1 \wedge b5;$

$b5 : b6 \wedge 1;$

$b6 : 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1;$

1) Присвоим всем булевым переменным 0:

$b1 = b2 = b3 = b4 = b5 = b6 = 0.$

Получаем:

$b1 = 0$

$b2 = 0$

$b3 = 0$

$b4 = 0$

$b5 = 0$

$b6 = 1.$

Второй проход алгоритма дает:

$b1 = 0$

$b2 = 0$

$b3 = 0$

$b4 = 0$

$b5 = 1.$

На третьем проходе алгоритма получаем:

$b1 = 0$

$b2 = 0$

$b3 = 0$

$b4 = 1.$

Четвертый проход дает:

$b1 = 0$

$b2 = 0$

$b3 = 1.$

Наконец, пятый проход алгоритма не изменяет значения булевых переменных:

$b1 = 0$

$b2 = 0.$

Значит, правило на строках 02–06 бесконечно, а с ним бесконечно и правило на строке 01.

Исправленная грамматика:

```
01  Program: [ | '-' ] Expression ;
02  Expression: Decimal_number |
03        Expression '+' Expression |
04        Expression '-' Expression  |
05        Expression '*' Expression |
06        Expression '/' Expression |
```


Труды СПИИРАН. 2006. Вып. 3. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

```
07        '(' [ | '-' ] Expression ')';
08  Decimal_number: Unsigned_integer |
                            Unsigned_integer Decimal_fraction
;
09  Decimal_fraction: '.' Unsigned_integer;
10  Unsigned_integer: Digit+ ' ' ;
11  Digit: '0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9';
```

Правило на строках 02–07 теперь заменяется на:

$$b2 : b3 \vee b2 \wedge 1 \wedge b2 \vee b2 \wedge 1 \wedge b2 \vee b2 \wedge 1 \wedge b2 \vee b2 \wedge 1 \wedge b2 \vee 1 \wedge [1 \vee 1] \wedge b2;$$

Пятый проход алгоритма дает:

$$b1 = 0$$

$$b2 = 1.$$

И шестой проход дает:

$$b1 = 1.$$

Бесконечных правил нет.

Разумеется, такие выкладки громоздки и алгоритм предназначен для компьютерной обработки.

4. Обосновывающая теорема

Теорема.

Правило КС-грамматики не является бесконечным тогда и только тогда, если после завершения работы приведенного выше алгоритма ему будет сопоставлено значение $b_i = 1$.

Доказательство.

Необходимость.

То, что правило не является бесконечным, означает, что нетерминал в его левой части может быть выражен через терминалы, семантики и другие нетерминалы, выражаемые, в конечном счете, через терминалы и семантики, с помощью операций объединения, произведения и обобщенной итерации. Так как терминалам и семантикам сопоставляется значение 1, а все булевы операции изоморфного множества и операция выделения первого операнда сохраняют истинность, то для данного правила $b_i = 1$.

Достаточность.

Пусть для i-го правила $b_i = 1$. Покажем, что тогда правило не является бесконечным. Доказательство методом математической индукции по числу операций.

Для удобства оформления перенумеруем правила так, чтобы значения $b_j = 1$ шли подряд: $1 \le j \le l$. Рассмотрим случай, когда правило содержит не более 1 операции. Простым перебором вариантов устанавливается, что $b_i = 1$ получается в случаях следующих замен: $1,1 \vee \nu_i, \nu_j \vee 1,1 \wedge 1,1 \setminus \nu_j$, где $\nu_i \in B \cup \{1\}$. Такие ситуации могут возникнуть лишь в следующих случаях:

$$o_s, o_s; n_i, n_i; o_s, o_s, o_r, o_s * n_i, 1 \le s \le l, 1 \le r \le l, n_i \in N \cup T \cup S^*, o_s, o_r \in N' \cup T \cup S^*$$

$N' = \{n \in N \mid n$ — не бесконечно$\}$.

Труды СПИИРАН. 2006. Вып. 3. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Каждое из этих выражений не является бесконечным. Начальное утверждение метода математической индукции доказано.

Пусть для правил, содержащих $k < m$ операций, утверждение теоремы верно. Покажем его справедливость при $k \le m$ операциях.

$n_1 : \psi_1(o_{11}, o_{12},..., o_{1m+1})$,

...

$n_l : \psi_l(o_{l1}, o_{l2},...o_{lm+1})$.

Выделим последнюю операцию в выражении:

$n_1 : \varphi_1(o_{11}, o_{12},..., o_{lm+1}) \oplus \rho_1(o_{11}, o_{12},..., o_{1m+1})$,

...

$n_l : \varphi_l(o_{l1}, o_{l2},...o_{lm+1}) \oplus \rho_l(o_{l1}, o_{l2},...o_{lm+1})$.

1. Предположим, что последняя операция в i-ом выражении $\oplus = *$. Тогда данная операция должна быть заменена операцией выделения первого операнда. По определению этой операции, для того, чтобы $b_{\psi_i} = 1$, должно быть $b_{\varphi_i} = 1$. Но тогда по допущению выражение $\varphi_i(o_{i1}, o_{i2},...o_{im+1})$ не является бесконечным, а тогда по определению обобщенной итерации правило $n_i : \psi_i(o_{i1}, o_{i2},...o_{im+1})$ не является бесконечным.

2. Предположим, что последняя операция в i-ом выражении $\oplus =<$ *пробел* $>$. Тогда данная операция должна быть заменена конъюнкцией. По определению конъюнкции, для того, чтобы $b_{\psi_i} = 1$, должно быть $b_{\varphi_i} = 1$ и $b_{\rho_i} = 1$. Но тогда по допущению выражение $\varphi_i(o_{i1}, o_{i2},...o_{im+1})$ и выражение $\rho_i(o_{i1}, o_{i2},...o_{im+1})$ не является бесконечными, а тогда по определению операции произведения правило $n_i : \psi_i(o_{i1}, o_{i2},...o_{im+1})$ не является бесконечным.

3. Предположим, что последняя операция в i-ом выражении $\oplus = |$. Тогда данная операция должна быть заменена дизъюнкцией. По определению дизъюнкции, для того, чтобы $b_{\psi_i} = 1$, должно быть $b_{\varphi_i} = 1$ или $b_{\rho_i} = 1$. Но тогда по допущению выражение $\varphi_i(o_{i1}, o_{i2},...o_{im+1})$ не является бесконечным или выражение $\rho_i(o_{i1}, o_{i2},...o_{im+1})$ не является бесконечным, а тогда по определению операции объединения правило $n_i : \psi_i(o_{i1}, o_{i2},...o_{im+1})$ не является бесконечным.

Теорема доказана.

Следствие 1.

Для любой КС-грамматики бесконечные правила отсутствуют тогда и только тогда, когда приведенный выше алгоритм после своего завершения выдает все значения $b_j = 1$, $i = 1,2,..., n$, где n — число правил.

Следствие 2.

Правило КС-грамматики является бесконечным тогда и только тогда, если после окончания работы приведенного выше алгоритма ему будет сопоставлено значение $b_i = 0$.

Эти утверждения непосредственно следуют из формулировки теоремы.

Труды СПИИРАН. 2006. Вып. 3. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

5. Заключение

На базе данного алгоритма реализуется распознавание бесконечных правил в программном верификаторе грамматики. Это необходимо для компьютерной проверки грамматик перед разбором. К сожалению, алгоритм не исправляет бесконечные правила на конечные, здесь требуется участие человека.

Литература

1. *Hopcroft J. E., Ullman J. D.* Formal Languages and Their Relation to Automata. Boston: Addison-Wesley Publishing Company, 1969. 329 p.
2. *Мартыненко Б. К., Федорченко Л. Н.* Эквивалентные преобразования КСР-грамматик в регулярной форме в практике построения языковых процессоров. Часть первая. Определение и распознавание КСР-языков посредством синтаксических граф-схем. Л.: ЛНИВЦ, 1983. 36 с.