УДК 519.766.2

МИНИМИЗАЦИЯ КОНТЕКСТНО-СВОБОДНЫХ ГРАММАТИК[1]

Ю. Д. Рязанов, С. В. Назина

Белгородский государственный технологический университет им. В. Г. Шухова, г. Белгород, Россия

Решается задача преобразования исходной контекстно-свободной грамматики (КС-грамматики) без лишних символов в эквивалентную ей грамматику меньшей сложности. Предлагается способ минимизации КС-грамматики, основанный на введённом отношении на множестве нетерминалов, обладающим свойством эквивалентности. Это отношение разбивает множество нетерминалов на классы эквивалентности, и новая КС-грамматика строится на нетерминалах, являющихся представителями классов эквивалентности. В результате получается КС-грамматика с меньшим количеством нетерминалов и правил.

Ключевые слова: *формальный язык, формальная грамматика, отношение эквивалентности, минимизация.*

DOI 10.17223/20710410/45/10

MINIMIZATION OF CONTEXT-FREE GRAMMARS

Yu. D. Ryazanov, S. V. Nazina

Belgorod State Technological University named after V. G. Shukhov, Belgorod, Russia

E-mail: razanov.yd@bstu.ru

This paper solves the problem of transforming the initial context-free grammar (CF-grammar) without excess characters into equivalent CF-grammar with less complexity. To solve this problem, the following relation on the set of a CF-grammar non-terminals is introduced: $E = \{(X, Y) : (X = Y) \vee (X \to \alpha \Leftrightarrow Y \to \beta \wedge |\alpha| = |\beta| \wedge \forall i\,(\alpha(i) = = \beta(i) \vee (\alpha(i), \beta(i)) \in E))\}$ where X, Y are non-terminals, α, β are chains of terminal and non-terminals, possibly blank, $\alpha(i)$ is the i-th character in chain α, $\beta(i)$ is the i-th character in chain β. It is proved that the relation E has the equivalence property and splits the set of non-terminals into equivalence classes. An algorithm is proposed for splitting a set of non-terminals into equivalence classes based on the method of sequential decomposition of the set of non-terminals into subsets so that non-equivalent non-terminals fall into different subsets. New CF-grammar is built on a set of non-terminals N, which elements are representatives of equivalence classes. From the set of rules of the initial CF-grammar, the rules with the left parts belonging to the set N are chosen. If there is a non-terminal in the left side of any selected rule that does not belong to the set N, then it is replaced by its equivalent non-terminal from the set N. After such transformations in the CF-grammar, sets of identical rules may appear. From each set of identical rules, we leave only one rule. The result is a CF-grammar containing less rules and non-terminals than the initial CF-grammar. The paper provides an example of the implementation of the described transformations.

Keywords: *formal language, formal grammar, equivalence relation, minimization.*

[1]Работа поддержана грантом РФФИ № 16-07-00487.

Введение

Для построения трансляторов, интерпретаторов и других программ обработки формальных языков широко используются методы, основанные на применении контекстно-свободных грамматик [1, 2]. Формальный язык можно задать КС-грамматиками, которые могут существенно различаться по сложности. Оценкой сложности КС-грамматики может служить суммарное количество символов (терминалов и нетерминалов) в правых частях правил грамматики. Размер программ обработки языков зависит от сложности используемой КС-грамматики, поэтому для сокращения размера программ целесообразно применять КС-грамматики меньшей сложности. Одним из способов уменьшения сложности КС-грамматики является устранение лишних символов [1, 3]. В данной работе решается задача преобразования исходной КС-грамматики без лишних символов в эквивалентную ей КС-грамматику меньшей сложности за счёт минимизации мощности множества нетерминалов. Для решения этой задачи предлагается способ, основанный на введённом отношении на множестве нетерминалов, обладающем свойством эквивалентности. Это отношение разбивает множество нетерминалов на классы эквивалентности, и новая КС-грамматика строится на нетерминалах, являющихся представителями этих классов. В результате получается КС-грамматика меньшей сложности.

Подход к минимизации дискретных объектов на основе некоторого отношения эквивалентности используется в минимизации состояний конечных автоматов [4 – 6], состояний и магазинных символов распознавателей с магазинной памятью [7, 8], минимизации челночных сплайновых процессоров [9], минимизации КСР-грамматик [10], минимизации автоматной модели визуального описания синтаксического разбора [11] и других объектов. В данном случае объектом минимизации является КС-грамматика и отношение эквивалентности вводится на множестве нетерминалов грамматики.

1. Отношение на множестве нетерминалов контекстно-свободной грамматики

Контекстно-свободной грамматикой [1] называется четвёрка $G = \langle N, \Sigma, P, S\rangle$, где

N — алфавит нетерминальных символов (нетерминалов);

Σ — непересекающийся с N алфавит терминальных символов (терминалов);

P — конечное множество правил вывода вида $A \to \alpha$, где $A \in N$, α — цепочка символов из $N \cup \Sigma$;

S — начальный нетерминал, $S \in N$.

Приведём пример грамматики G_1:

$S \to AaC \,|D\,|\, SbF$
$A \to DaS \,|\, E \,|\, AbB$
$B \to EaC \,|\, D \,|\, BbA$
$C \to AaS \,|\, E \,|\, SbC$
$D \to bSEa \,|\, bBSa \,|\, bCDa \,|\, c$
$E \to bFDa \,|\, bASa \,|\, c$
$F \to BaF \,|\, D \,|\, FbF$

В этой грамматике $\{S, A, B, C, D, E, F\}$ — множество нетерминалов; S — начальный нетерминал; $\{a, b, c\}$ — множество терминалов и двадцать два правила. Сложность грамматики равна 57.

Определим отношение на множестве нетерминалов:

$$E = \{(X, Y) : (X = Y) \vee (X \to \alpha \Leftrightarrow Y \to \beta \wedge |\alpha| = |\beta| \wedge \forall i\, (\alpha(i) = \beta(i) \vee (\alpha(i), \beta(i)) \in E))\},$$

где X, Y — нетерминалы; α, β — цепочки терминалов и нетерминалов, возможно пустые; $\alpha(i)$ — i-й символ в цепочке α; $\beta(i)$ — i-й символ в цепочке β.

Правила $X \to \alpha$ и $Y \to \beta$, в которых $|\alpha| = |\beta|$ и $\forall i\,(\alpha(i) = \beta(i) \vee (\alpha(i), \beta(i)) \in E)$, назовём эквивалентными.

Цепочки одинаковой длины α^i и α^j назовём эквивалентными, если $\forall(k)\,(\alpha^i(k) = \alpha^j(k) \vee (\alpha^i(k), \alpha^j(k)) \in E)$, где $\alpha^i(k)$ — k-й символ в цепочке α^i; $\alpha^j(k)$ — k-й символ в цепочке α^j. Множество всех терминальных цепочек, выводимых из нетерминала X, обозначим $L(X)$.

Теорема 1. Если $(X_i, X_j) \in E$, то $L(X_i) = L(X_j)$.

Доказательство. Начнём один вывод из нетерминала X_i, а второй — из X_j. В исходном состоянии цепочка α^i состоит только из нетерминала X_i, а цепочка α^j — из X_j. Эти цепочки эквивалентны, так как $(X_i, X_j) \in E$. Пусть на очередном шаге вывода к k-му символу цепочки α^i, который является нетерминалом, применяется правило $\alpha^i(k) \to \beta$, в результате чего получается цепочка γ^i. Тогда на этом же шаге применим к k-му символу цепочки α^j правило $\alpha^j(k) \to \delta$, которое эквивалентно правилу $\alpha^i(k) \to \beta$, в результате чего получим цепочку γ^j, эквивалентную цепочке γ^i. Продолжая таким образом выводы, будем получать эквивалентные цепочки вплоть до терминальных. Терминальные эквивалентные цепочки равны. Таким образом, любую терминальную цепочку, выводимую из нетерминала X_i, можно вывести из нетерминала X_j, и наоборот, следовательно, $L(X_i) = L(X_j)$. ■

Из $L(X_i) = L(X_j)$ для $(X_i, X_j) \in E$ следует очевидное эквивалентное преобразование: все вхождения нетерминала X_j в правые части правил заменить на X_i, а правила с левой частью X_j исключить. В результате такого преобразования количество нетерминалов и правил грамматики уменьшится.

Теорема 2. Отношение E обладает свойством эквивалентности.

Доказательство. Отношение обладает свойством эквивалентности, если оно рефлексивно, симметрично и транзитивно. Рефлексивность и симметричность отношения следует из его определения. Отношение транзитивно, если $((X, Y) \in E \wedge \wedge (Y, Z) \in E) \Rightarrow (X, Z) \in E$ для всех X, Y, Z.

Пусть $(X, Y) \in E$ и $(Y, Z) \in E$. Если в грамматике есть правило $X \to \alpha$ и $(X, Y) \in \in E$, то правило $Y \to \beta$, такое, что $|\alpha| = |\beta|$ и $\alpha(i) = \beta(i)$, если i-й символ — терминал в цепочках α и β, и $(\alpha(i), \beta(i)) \in E$, если i-й символ — нетерминал, тоже есть в этой грамматике. Аналогично если правило $Y \to \beta$ есть в грамматике и $(Y, Z) \in E$, то в грамматике есть правило $Z \to \gamma$, такое, что $|\beta| = |\gamma|$ и для любого символа $\beta(i) = \gamma(i)$ или $(\beta(i), \gamma(i)) \in E$. Так как для любого нетерминала $\alpha(i)$ выполняется $(\alpha(i), \beta(i) \in E)$, а для любого терминала $\alpha(i) = \beta(i)$, то правило $X \to \alpha$ можно заменить правилом $X \to \beta$. Поскольку $|\beta| = |\gamma|$, а $\beta(i) = \gamma(i)$ или $(\beta(i), \gamma(i)) \in E$, условие принадлежности пары (X, Z) отношению E является истинным. ■

2. Разбиение множества нетерминалов на классы эквивалентности

Отношение E определяет разбиение множества нетерминалов на классы эквивалентности. Применим метод последовательного разбиения множества нетерминалов на подмножества так, чтобы неэквивалентные нетерминалы попали в разные подмножества.

Сначала разобьём множество нетерминалов на подмножества $Q_1, Q_2, \ldots, Q_n$ так, чтобы нетерминалы, для которых условие $X \to \alpha \Leftrightarrow Y \to \beta \wedge |\alpha| = |\beta| \wedge \forall i\,(\alpha(i) \in \in \Sigma \Rightarrow \alpha(i) = \beta(i))$ ложно, оказались в разных подмножествах. Для этого построим

таблицу, столбцы которой соответствуют нетерминалам, а строки — цепочкам, составленным из правых частей правил грамматики заменой нетерминалов символом «$*$», не принадлежащим множеству терминалов или нетерминалов грамматики. Цепочку, полученную из правой части правила $X \to \alpha$ заменой нетерминалов символом «$*$», обозначим α^*. Таблицу заполняем следующим образом: если в грамматике существует правило $X \to \alpha$, то в клетку, соответствующую нетерминалу X и цепочке α^*, записываем «1».

Если столбцы, соответствующие нетерминалам X и Y, различаются, то эти нетерминалы нужно включить в разные подмножества. Для грамматики G_1 эта таблица представлена в табл. 1.

Таблица 1

α^*	S	A	B	C	D	E	F
$*a*$	1	1	1	1			1
$*$	1	1	1	1			1
$*b*$	1	1	1	1			1
$b**a$					1	1	
c					1	1	

Анализируя табл. 1, получаем подмножества $Q_1 = \{S, A, B, C, F\}$ и $Q_2 = \{D, E\}$, которые образуют разбиение $R = \{Q_1, Q_2\}$ множества нетерминалов.

Нетерминалы, принадлежащие разным подмножествам, явно неэквивалентны. Так как мы не учитывали условие $(\alpha(i), \beta(i)) \in E$ для нетерминальных символов цепочек, нетерминалы, принадлежащие одному подмножеству, тоже могут быть неэквивалентными.

Для дальнейшего разбиения построим таблицу, структура которой аналогична таблице, построенной на предыдущем шаге, за исключением строк, которые не содержат символ «$*$». В клетку таблицы будем записывать множество упорядоченных последовательностей чисел $(i_1, i_2, \ldots, i_n)$, где i_k — номер подмножества в последнем разбиении, $k = 1, \ldots, n$, n — количество нетерминалов в правой части каждого правила, соответствующего строке таблицы. Если в грамматике есть правило $X \to \alpha$ и α содержит n нетерминалов $X_1, X_2, \ldots X_n$, то в клетку, соответствующую нетерминалу X и цепочке α^*, добавляем последовательность $(i_1, i_2, \ldots, i_n)$, где i_k — номер подмножества в последнем разбиении, которому принадлежит нетерминал X_k.

Если нетерминалы X и Y принадлежат одному подмножеству и соответствующие им столбцы различаются, то пара нетерминалов X и Y не принадлежит отношению E и их нужно включить в разные подмножества нового разбиения R'.

Анализируя таблицу, формируем новое разбиение R'. Подмножество Q'_k разбиения R' формируется из элементов некоторого подмножества Q_k разбиения R: Q'_k является максимальным по мощности подмножеством множества Q_k, таким, что для каждой пары нетерминалов X и Y, принадлежащих подмножеству Q'_k, столбцы таблицы, соответствующие X и Y, равны.

Если $R' \neq R$, то полагаем $R := R'$, заново строим таблицу по описанным правилам и формируем новое разбиение. Если $R' = R$, то R' представляет собой множество классов эквивалентности.

Построим новую таблицу для G_1 (табл. 2).

В этой таблице сначала расположены столбцы, соответствующие подмножеству Q_1, а затем — столбцы, соответствующие подмножеству Q_2. Рассмотрим правило $S \to AaC$

Таблица 2

α^*	Q_1					Q_2	
	S	A	B	C	F	D	E
$*a*$	(1, 1)	(2, 1)	(2, 1)	(1, 1)	(1, 1)		
$*$	(2)	(2)	(2)	(2)	(2)		
$*b*$	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)		
$b**a$						(1, 2) (1, 1)	(1,2) (1,1)

грамматики G_1. Правая часть соответствует цепочке $*a*$. Оба нетерминала правой части принадлежат подмножеству Q_1, поэтому в клетку, находящуюся в строке $*a*$ и столбце S, записываем последовательность (1,1). Клетка в строке $b**a$ и столбце S остаётся пустой, так как в грамматике нет правила с нетерминалом S в левой части, правая часть которого соответствует цепочке $b**a$. В клетку в строке $b**a$ и столбце D записываем последовательность (1,2), так как в грамматике есть правило $D \to bSEa$ и $S \in Q_1$, $E \in Q_2$. В эту же клетку записываем пару (1,1), так как в грамматике есть правило $D \to bBSa$ и $B \in Q_1$, $S \in Q_1$. Аналогично заполняем все клетки таблицы.

Анализируя табл. 2, видим, что Q_1 разбивается на два подмножества, а Q_2 не разбивается. В результате получаем новое разбиение $R' = \{Q'_1, Q'_2, Q'_3\}$, где $Q'_1 = \{S, C, F\}$, $Q'_2 = \{A, B\}$, $Q'_3 = \{D, E\}$. Поскольку $R' \neq R$, полагаем $R := R'$ и строим новую таблицу (табл. 3).

Таблица 3

α^*	Q_1			Q_2		Q_3	
	S	C	F	A	B	D	E
$*a*$	(2, 1)	(2, 1)	(2, 1)	(3, 1)	(3, 1)		
$*$	(3)	(3)	(3)	(3)	(3)		
$*b*$	(1, 1)	(1, 1)	(1, 1)	(2, 2)	(2, 2)		
$b**a$						(1, 3) (2, 1)	(1,3) (2,1)

Анализируя табл. 3, приходим к выводу, что подмножества Q_1, Q_2 и Q_3 не разбиваются и поэтому представляют собой классы эквивалентности.

3. Построение упрощенной грамматики

Новую грамматику G_2 формируем на множестве нетерминалов, являющихся представителями классов эквивалентности. Таким множеством может быть, например, $\{S, A, D\}$. Из множества правил грамматики G_1 берём правила, левые части которых принадлежат множеству $\{S, A, D\}$:

$S \to AaC \,|D\,|\, SbF$
$A \to DaS \,|\, E \,|\, AbB$
$D \to bSEa \,|\, bBSa \,|\, bCDa \,|\, c$

Последовательно просматриваем все правила и если в правой части есть нетерминал, который не принадлежит множеству $\{S, A, D\}$, то заменяем его на эквивалентный ему нетерминал из множества $\{S, A, D\}$:

$S \to AaS \,|D\,|\, SbS$
$A \to DaS \,|\, E \,|\, AbA$
$D \to bSDa \,|\, bASa \,|\, bSDa \,|\, c$

Как видно, в грамматике появились одинаковые правила. Оставляя из каждого множества одинаковых правил по одному, получаем грамматику G_2:

$S \to AaS \,|\, D \,|\, SbS$
$A \to DaS \,|\, E \,|\, AbA$
$D \to bSDa \,|\, bASa \,|\, c$

Полученная грамматика имеет сложность 23, что значительно меньше сложности исходной грамматики.

Заключение

В процессе выполнения описанных преобразований количество правил грамматики никогда не увеличивается. Уменьшение сложности грамматики достигается за счёт исключения из исходной грамматики правил, определяющих нетерминалы, не принадлежащие новому множеству нетерминалов, и замены возникающих в процессе преобразований множеств одинаковых правил одним. Следовательно, уменьшить сложность грамматики можно за счёт минимизации множества нетерминалов.

ЛИТЕРАТУРА

1. *Aho A. V. and Ullman J. D.* The Theory of Parsing, Translation and Compiling. NJ, USA: Prentice-Hall Inc., 1972. V. 1. 560 p.
2. *Aho A. V., Lam M. S., Sethi R., and Ullman J. D.* Compilers: Principles, Techniques, and Tools. Addison-Wesley, 2007, 1009 p.
3. *Конюхова О. В., Кравцова Э. А.* Программная реализация алгоритмов упрощения контекстно-свободных грамматик на языках программирования Haskell и Prolog // Информационные системы и технологии. 2017. № 4. С. 77–86.
4. *Hopcroft J. E.* An $n \log n$ algorithm for minimizing states in a finite automaton // Theory of Machines and Computations. N.Y.: Academic Press, 1971. P. 189–196.
5. *Hopcroft J. E., Motwani R., and Ullman J. D.* Introduction to Automata Theory, Languages, and Computation. Pearson, 2013, 496 p.
6. *Мартыненко Б. К.* Ещё один метод минимизации конечных автоматов // Компьютерные инструменты в образовании. 2017. № 1. С. 5–14.
7. *Polyakov V. M. and Ryazanov Yu. D.* Reducing the number of states in pushdown recognizers by means of equivalence relation // Intern. J. Pharmacy & Technology. 2016. V. 8. No. 4. P. 22578–22587.
8. *Рязанов Ю. Д.* Сокращение количества магазинных символов в распознавателях с магазинной памятью и одним состоянием // Вестник БГТУ им. В. Г. Шухова. 2017. № 6. С. 152–157.
9. *Мартыненко Б. К.* Синтаксически управляемая обработка данных. СПб.: Изд-во С.-Петербург. ун-та, 2004. 316 с.
10. *Федорченко Л. Н.* Минимизация трансляционной КСР-грамматики и состояний синтаксического анализатора КСР-языка // Вестник Бурятского государственного университета. Математика, информатика. 2013. № 2. С. 39–49.
11. *Стасенко А. П.* Автоматная модель визуального описания синтаксического разбора // Вычислительные технологии. 2008. Т. 13. № 5. С. 70–87.

REFERENCES

1. *Aho A. V. and Ullman J. D.* The Theory of Parsing, Translation and Compiling. New Jersey, Prentice-Hall, 1972, vol. 1, 560 p.

2. *Aho A. V., Lam M. S., Sethi R.,and Ullman J. D.* Compilers: Principles, Techniques, and Tools. Addison-Wesley, 2007, 1009 p.
3. *Konyuxova O. V. and Kravcova E. A.* Programmnaya realizaciya algoritmov uproshcheniya kontekstno-svobodnyh grammatik na yazykah programmirovaniya Haskell i Prolog [The implementation of the simplifying context-free grammars algorithms in Haskell and in Prolog]. Information Systems and Technologies, 2017, no. 4, pp. 77–86. (in Russian)
4. *Hopcroft J. E.* An $n \log n$ algorithm for minimizing states in a finite automaton Theory of Machines and Computations. N.Y., Academic Press, 1971, pp. 189–196.
5. *Hopcroft J. E., Motwani R., and Ullman J. D.* Introduction to Automata Theory, Languages, and Computation. Pearson, 2013. 496 p.
6. *Martynenko B. K.* Eshche odin metod minimizatsii konechnykh avtomatov [One more method for minimization of finite automata]. Computer Tools in Education, 2017, no. 1, pp. 5–14. (in Russian)
7. *Polyakov V. M. and Ryazanov Yu. D.* Reducing the number of states in pushdown recognizers by means of equivalence relation. Intern. J. Pharmacy & Technology, 2016, vol. 8, no. 4, pp. 22578–22587.
8. *Ryazanov Yu. D.* Sokrashchenie kolichestva magazinnykh simvolov v raspoznavatelyakh s magazinnoy pamyatyu i odnim sostoyaniem [Reducing the Number of Pushdown Symbols in One-state Pushdown Recognizers]. Bulletin of BSTU named after V. G. Shukhov, 2017, no. 6, pp. 152–157. (in Russian)
9. *Martynenko B. K.* Sintaksicheski upravlyaemaya obrabotka dannykh [Syntax-directed data processing]. Saint-Petersburg, St Petersburg University, 2004. 316 p. (in Russian)
10. *Fedorchenko L. N* Minimizatsiya translyatsionnoy KSR-grammatiki i sostoyaniy sintaksicheskogo analizatora KSR-yazyka [Minimization of Translational CFR-grammar and Conditions of Generated Parser CFR-language]. Bulletin of the Buryat State University. Mathematics, Informatics, 2013, no. 2, pp. 39–49. (in Russian)
11. *Stasenko A. P.* Avtomatnaya model vizualnogo opisaniya sintaksicheskogo razbora [Automaton model of visual description of syntax parsing]. Computational Technologies, 2008, vol. 13, no. 5. pp. 70–87. (in Russian)