

Построение контекстно-свободной грамматики по мультиграфу автомата с магазинной памятью

07, июль 2014

DOI: 10.7463/0714.0717777

Белоусов А. И.¹, Ткачев С. Б.^{1,2,a}

УДК 519.76

¹Россия, МГТУ им. Н.Э. Баумана

²Россия, ИСА РАН

^astkachev@bmstu.ru

Статья посвящена разработке алгоритма построения контекстно-свободной грамматики по мультиграфу автомата с магазинной памятью (МП-автомата). Алгоритм основан на перечислении определенных путей в мультиграфе. Дается описание алгоритма в псевдокоде и вычисляется верхняя оценка сложности, составляющей величину порядка куба числа дуг мультиграфа

Ключевые слова: автомат с магазинной памятью; МП-автомат; ориентированный мультиграф; контекстно-свободная грамматика; КС-грамматика; сбалансированный путь; поиск в глубину

Введение

В последние годы контекстно-свободные грамматики [1, 2] используются не только для решения традиционных задач, связанных с построением и анализом различных формальных языковых конструкций, но и для разработки средств поиска информации в сети Internet [3], а также для конструирования алгоритмов управления робототехническими системами [4, 5]. Например, при синтезе двухуровневой системы управления робототехнической системой КС-грамматика оказалась удобным инструментом для решения задачи логического управления на верхнем уровне.

В [5] предложен подход к решению задач управления нелинейными системами с использованием конечных автоматов. Отметим, что метод нахождения языка, допускаемого конечным автоматом, а также метод построения соответствующей регулярной грамматики хорошо известны [1, 2].

Использование для синтеза алгоритмов управления автоматов с магазинной памятью (МП-автоматов) позволяет существенно расширить круг решаемых задач. В случае, когда объектом анализа является МП-автомат, метод построения по нему КС-грамматики также известен [1, 2], однако получающаяся грамматика является весьма громоздкой с большим числом бесполезных нетерминалов.

Проблема оптимизации КС-грамматики, построенной по системе команд МП-автомата, имеет самостоятельное теоретическое значение. В данной работе, являющейся продолжением работы [6], предлагается алгоритм оптимизации КС-грамматики, построенной по системе команд МП-автомата. Как показано в [6], одним из удобных способов представления МП-автомата является размеченный мультиграф. В основе предлагаемого алгоритма оптимизации лежит алгоритм поиска путей, обладающих специальными свойствами, в мультиграфе МП-автомата.

Материал статьи организован следующим образом. В разд. 1 описан подход к решению задачи оптимизации КС-грамматики, в разд. 2 обоснован алгоритм нахождения путей в графе МП-автомата, обладающих специальными свойствами (сбалансированных путей). В разд. 3 получена оценка сложности предложенного алгоритма.

1. Постановка задачи оптимизации КС-грамматики

Следуя [1], рассмотрим некоторый МП-автомат

$$M = (Q, V, \Gamma, q_0, F, Z_0, \delta),$$

где Q — множество состояний МП-автомата; V — терминальный алфавит; Γ — нетерминальный алфавит; $q_0 \in Q$ — начальное состояние; $F \subseteq Q$ — множество заключительных состояний; Z_0 — начальный магазинный символ; δ — система команд МП-автомата.

Напомним, что при построении КС-грамматики по системе команд МП-автомата по команде

$$q a Z \rightarrow r \gamma, \quad (1)$$

где $q \in Q$ — состояние МП-автомата; $a \in V$ — символ терминального алфавита; $Z \in \Gamma$ — магазинный символ; $r \in Q$ — состояние МП-автомата, $\gamma \in \Gamma^*$ — цепочка, которая замещает в магазине символ Z , при непустой цепочке $\gamma = \gamma(1) \gamma(2) \dots \gamma(k)$, $k > 0$, строится система КС-правил вида

$$[qZs_k] \rightarrow a[r\gamma(1)s_1][s_1\gamma(2)s_2] \dots [s_{k-1}\gamma(k)s_k] \quad (2)$$

для всех возможных кортежей состояний МП-автомата $(s_1, s_2, \dots, s_k)$ размерности k .

В каждом КС-правиле вида (2) в качестве нетерминалов КС-грамматики рассматриваются тройки вида $[qXr]$, где $q, r \in Q$, а $X \in \Gamma$ — магазинный символ. Таким образом, по одной команде (1) строится система из $|Q|^k$ правил (2), причем многие из вводимых таким образом нетерминалов-троек вида $[qXr]$ оказываются бесполезными. Поэтому значительный интерес представляет задача разработки алгоритма, позволяющего исключать из рассмотрения бесполезные нетерминалы и соответствующие им КС-правила на этапе построения КС-грамматики.

В пособии [7] приведен алгоритм, позволяющий получить весьма простую КС-грамматику по заданному МП-автомату. Однако для использования этого алгоритма предварительно требуется преобразовать систему команд МП-автомата к некоторому специальному виду. Та-

ким образом, проблема оптимизации КС-грамматики заменяется проблемой преобразования к требуемому виду системы команд МП-автомата, но соответствующий алгоритм в [7] не рассматривается.

2. Основной принцип оптимизации

Перейдем к решению задачи оптимизации КС-грамматики с правилами вида (2). Будем предполагать, что для анализируемого МП-автомата по методу, предложенному в [6], построен соответствующий ему мультиграф, в котором дуге (q, r) мультиграфа, соответствующей команде вида (1), поставлена в соответствие метка дуги (Z, a, γ) .

Отметим, что далее всюду будем рассматривать в МП-автомате допуск с пустым магазином [1], т.е. будем считать, что, допуская цепочку во входном алфавите, автомат приходит в одно из заключительных состояний с пустым магазином.

В этом случае в силу результатов работы [6] входная цепочка принадлежит языку МП-автомата тогда и только тогда, когда она в мультиграфе МП-автомата является входной меткой такого пути, начинающегося со стартовой дуги, что его последняя дуга заходит в заключительное состояние с пустой магазинной меткой. Отметим, что для указанного пути определена магазинная метка [6]. Любой такой путь будем называть сбалансированным. Кроме того, ради краткости, путь, начинающийся дугой, исходящей из вершины q мультиграфа и заканчивающийся дугой, заходящей в вершину r , будем обозначать как путь, ведущий из вершины (или состояния) q в вершину (состояние) r .

Тогда состояния $s_1, s_2, \dots, s_k$ в написанном выше правиле вывода (2) выбираются так, что они лежат на некотором сбалансированном пути в мультиграфе МП-автомата, ведущем из начального состояния в заключительное, и первые компоненты меток исходящих из них дуг равны $\gamma(2), \dots, \gamma(k)$ соответственно, а дуга, исходящая из состояния q в состояние r (непосредственно достижимого из q), имеет в качестве первой компоненты метки символ $\gamma(1)$.

При этом в определенных случаях логика построения системы команд МП-автомата позволяет для оптимизации КС-грамматики применить такой прием: при рассмотрении очередной команды МП-автомата и построении по ней правила КС-грамматики не вводить таких новых нетерминалов-троек вида $[pXq]$, которые не фигурируют в КС-правилах частично построенной к данному шагу оптимизированной грамматики.

Рассмотрим некоторые примеры.

Пример 1. На рис. 1 изображен мультиграф МП-автомата, допускающего язык $L_1 = \{a^n b^n \mid n \geq 1\}$, со следующей системой команд:

$$\begin{aligned} q_0 a Z &\rightarrow q_0 a Z, \\ q_0 a a &\rightarrow q_0 a a, \\ q_0 b a &\rightarrow q_1 \lambda, \\ q_1 b a &\rightarrow q_1 \lambda, \\ q_1 \lambda Z &\rightarrow q_2 \lambda. \end{aligned}$$

На рис. 1 метка каждой дуги разделена на три компоненты, первая из которых помещена у начала дуги, вторая помещена в середине дуги, а третья — у конца дуги.

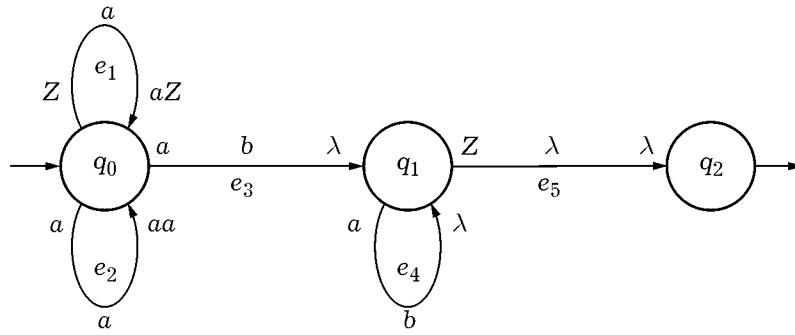


Рис. 1

Следуя алгоритму, приведенному в [1, с. 647], построим КС-грамматику по МП-автомату. Она будет включать следующие правила:

$$\begin{aligned}
 S &\rightarrow [q_0 Z q_2], \\
 [q_0 Z s_2] &\rightarrow a[q_0 a s_1][s_1 Z s_2], \\
 [q_0 a s_2] &\rightarrow a[q_0 a s_1][s_1 a s_2], \\
 [q_0 a q_1] &\rightarrow b, \\
 [q_1 a q_1] &\rightarrow b, \\
 [q_1 Z q_2] &\rightarrow \lambda,
 \end{aligned}$$

где S — аксиома, а второе и третье правила — групповые, соответствующие каждой упорядоченной паре состояний (s_1, s_2) и включающие каждое по девять правил.

Условимся далее начальной магазинной меткой дуги называть первую компоненту ее метки, тогда как третью компоненту будем называть заключительной магазинной меткой, или даже просто магазинной меткой.

Заметим, что из состояния q_0 выходят две дуги с начальной магазинной меткой a : петля $e_2 = (q_0, q_0)$ и дуга $e_3 = (q_0, q_1)$. Если в качестве значения параметра s_1 выберем q_0 , то тогда следует искать дугу (q_0, s_2) с начальной магазинной меткой Z . Но это может быть только петля e_1 , которая, как нетрудно видеть, не сцепляется с петлей e_2 . Следовательно, нужно положить $s_1 = q_1$, выбрав тем самым дугу e_3 . Тогда остается признать, что состояние s_2 необходимо выбрать как q_2 . Соответствующий сбалансированный путь есть, например, $e_1 e_3 e_5$ с допуском входной цепочки ab .

Итак, из второй строки записанной выше системы правил остается только одно правило (из девяти возможных):

$$[q_0 Z q_2] \rightarrow a[q_0 a q_1][q_1 Z q_2].$$

Переходим к третьей строке системы правил грамматики. Согласно сформулированному выше принципу оптимизации, возможны такие варианты:

$$[q_0 a q_1] \rightarrow a[q_0 a q_1][q_1 a q_1]$$

(дуги e_3 и e_4),

$$[q_0 a q_1] \rightarrow a[q_0 a q_0][q_1 a q_1]$$

(дуги e_2 и e_3).

Во втором случае возникает нетерминал $[q_0 a q_0]$, который может быть «раскрыт» только таким правилом:

$$[q_0 a q_0] \rightarrow a[q_0 a q_0][q_0 a q_0].$$

При построении этого правила дважды повторена самосцепляющаяся петля e_2 . Но, как нетрудно видеть, это правило приводит к появлению бесполезного нетерминала. Это значит, что следует оставить только первый вариант.

Приведем вывод цепочки $a^n b^n$ в оптимизированной грамматике:

$$\begin{aligned} S \vdash [q_0 Z q_2] \vdash a[q_0 a q_1][q_1 Z q_2] \vdash \dots \vdash \\ \vdash a a^{n-1} [q_0 a q_1][q_1 a q_1]^{n-1} [q_1 Z q_2] \vdash a^n b [q_1 a q_1]^{n-1} [q_1 Z q_2] \vdash^n a^n b^n. \end{aligned}$$

Поскольку множество правил оптимизированной грамматики является подмножеством правил исходной грамматики, которую можно построить по МП-автомату стандартным алгоритмом, то никаких слов, не принадлежащих языку, допускаемому рассматриваемым МП-автоматом, новая грамматика не породит.

Поскольку грамматика построена на основе анализа всех путей в мультиграфе, представляющем МП-автомат, то потери слов также не произойдет.

Следовательно, построенная оптимизированная грамматика эквивалентна грамматике, построенной стандартным алгоритмом.

Пример 2. Рассмотрим МП-автомат со следующей системой команд:

$$\begin{aligned} q_0 a Z &\rightarrow q_1 a Z, \\ q_1 a a &\rightarrow q_1 a a, \\ q_1 b a &\rightarrow q_2 \lambda, \\ q_2 b a &\rightarrow q_2 \lambda, \\ q_2 \lambda Z &\rightarrow q_0 \lambda, \\ q_0 \lambda Z &\rightarrow q_0 \lambda. \end{aligned}$$

(см. [1], пример 8.17; начальное и заключительное состояние МП-автомата — q_0). Этот МП-автомат, как и автомат, приведенный в примере 1, допускает язык $L_2 = \{a^n b^n \mid n \geq 0\}$. Мультиграф изображен на рис. 2.

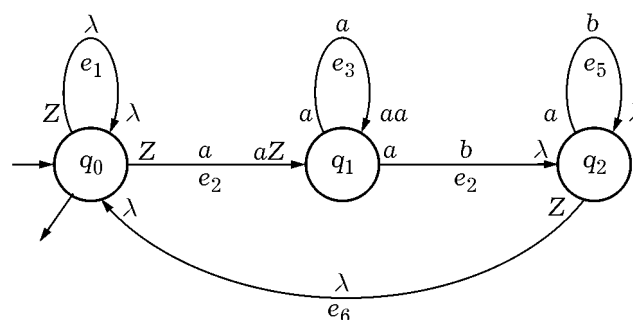


Рис. 2

Построим КС-грамматику по стандартному алгоритму:

$$\begin{aligned} S &\rightarrow [q_0 Z q_0], \\ [q_0 Z s_2] &\rightarrow a[q_1 a s_1][s_1 Z s_2], \\ [q_1 a s_2] &\rightarrow a[q_1 a s_1][s_1 a s_2], \\ [q_1 a q_2] &\rightarrow b, \\ [q_2 a q_2] &\rightarrow b, \\ [q_2 Z q_0] &\rightarrow \lambda, \\ [q_0 Z q_0] &\rightarrow \lambda. \end{aligned}$$

где второе и третье правила распространяются на все упорядоченные пары (s_1, s_2) из множества состояний $\{q_0, q_1, q_2\}$.

Нетрудно видеть, что для второй строки получаем единственный вариант: $s_1 = q_2$, $s_2 = q_0$ (дуги e_4 и e_6 , лежащие на сбалансированном пути).

Для третьей строки, используя принцип недобавления новых нетерминалов, будем иметь: $s_1 = s_2 = q_2$. Можно видеть, что попытка ввести здесь нетерминал $[q_1 a q_1]$ также приведет к появлению бесполезного нетерминала. Таким образом, остается одно правило

$$[q_1 a q_2] \rightarrow a[q_1 a q_2][q_2 a q_2].$$

Эквивалентность оптимизированной грамматики исходной можно обосновать так же, как в предыдущем примере.

Пример 3. Рассмотрим МП-автомат, распознающий палиндромы в алфавите $\{a, b\}$, заданный следующей системой команд:

$$\begin{aligned} 1 \quad & q_0 \lambda Z \rightarrow q_2 \lambda, \\ 2 \quad & q_0 a Z \rightarrow q_0 a Z \mid q_1 Z, \\ 3 \quad & q_0 b Z \rightarrow q_0 b Z \mid q_1 Z, \\ 4 \quad & q_0 a a \rightarrow q_0 a a \mid q_1 a \mid q_1 \lambda, \\ 5 \quad & q_0 a b \rightarrow q_0 a b \mid q_1 b, \\ 6 \quad & q_0 b a \rightarrow q_0 b a \mid q_1 a, \\ 7 \quad & q_0 b b \rightarrow q_0 b b \mid q_1 b \mid q_1 \lambda, \\ 8 \quad & q_1 \lambda Z \rightarrow q_2 \lambda. \end{aligned}$$

Здесь q_0 — начальное состояние, q_2 — заключительное состояние. Команда (1) допускает пустое слово, первая команда второй строки читает первую букву a в магазин, вторая команда этой строки предназначена для допуска однобуквенного палиндрома a . Команды третьей строки делают то же самое для буквы b . Команды строки (4) читают букву a в магазин с верхним символом a , «угадывают» середину нечетной и четной цепочки соответственно. Аналогично для буквы b предназначены команды строки (7). Строки (5) и (6) читают в магазин одну из букв, когда в верхушке магазина находится другая буква, и «угадывают» середину палиндрома нечетной длины. Команда (8) — команда окончательного допуска непустой цепочки.

Мультиграф МП-автомата представлен на рис. 3. Здесь некоторые дуги объединены. Номера дуг и соответствующие метки указаны к скобкам. Совпадающие компоненты меток повторены только один раз.

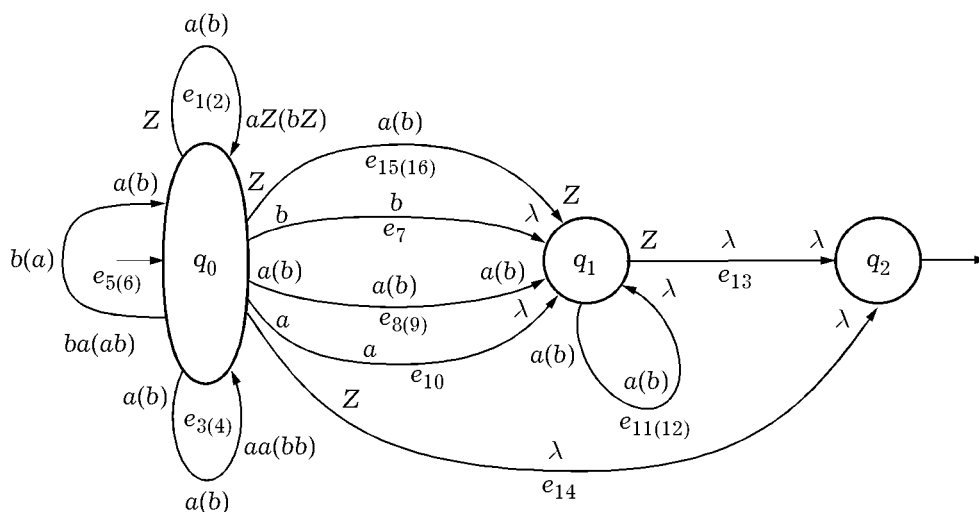


Рис. 3

Строим КС-грамматику:

- 1 $S \rightarrow [q_0 Z q_2],$
- 2 $[q_0 Z s_2] \rightarrow a[q_0 a s_1][s_1 Z s_2],$
- 3 $[q_0 Z s_2] \rightarrow b[q_0 b s_1][s_1 Z s_2],$
- 4 $[q_0 a s_2] \rightarrow a[q_0 a s_1][s_1 a s_2],$
- 5 $[q_0 b s_2] \rightarrow b[q_0 b s_1][s_1 b s_2],$
- 6 $[q_0 a s_2] \rightarrow b[q_0 b s_1][s_1 a s_2],$
- 7 $[q_0 b s_2] \rightarrow a[q_0 a s_1][s_1 b s_2],$
- 8 $[q_0 a s_1] \rightarrow a[q_1 a s_1],$
- 9 $[q_0 a s_1] \rightarrow b[q_1 a s_1],$
- 10 $[q_0 b s_1] \rightarrow b[q_1 b s_1],$
- 11 $[q_0 b s_1] \rightarrow a[q_1 b s_1],$
- 12 $[q_0 Z s_1] \rightarrow a[q_1 Z s_1],$
- 13 $[q_0 Z s_1] \rightarrow b[q_1 Z s_1],$
- 14 $[q_0 Z q_2] \rightarrow \lambda,$
- 15 $[q_1 Z q_2] \rightarrow \lambda,$
- 16 $[q_0 a q_1] \rightarrow a,$
- 17 $[q_0 b q_1] \rightarrow b,$
- 18 $[q_1 a q_1] \rightarrow a,$
- 19 $[q_0 b q_1] \rightarrow a,$

Здесь правила 2–13 групповые. Первые шесть задают по девять правил, а последующие шесть — по три правила. Взять петлю e_3 в качестве дуги, исходящей из вершины q_0 и имеющей начальную магазинную метку a , нельзя, так как тогда (т.е. при $s_1 = q_0$) придется использовать петлю e_1 , но путь $e_3 e_1$ не имеет магазинной метки и потому не может лежать ни на каком сбалансированном пути. Тогда для строки (2) следует оставить правило

$$[q_0 Z q_2] \rightarrow a[q_0 a q_1][q_1 Z q_2].$$

Рассуждая далее подобным образом, получим следующую оптимизированную грамматику:

- 1 $S \rightarrow [q_0 Z q_2]$
- 2 $[q_0 Z q_2] \rightarrow a[q_0 a q_1][q_1 Z q_2],$
- 3 $[q_0 Z q_2] \rightarrow b[q_0 b q_1][q_1 Z q_2],$
- 4 $[q_0 a q_1] \rightarrow a[q_0 a q_1][q_1 a q_1],$
- 5 $[q_0 b q_1] \rightarrow b[q_0 b q_1][q_1 b q_1],$
- 6 $[q_0 a q_1] \rightarrow b[q_0 b q_1][q_1 a q_1],$
- 7 $[q_0 b q_1] \rightarrow a[q_0 a q_1][q_1 b q_1],$
- 8 $[q_0 a q_1] \rightarrow a[q_1 a q_1],$
- 9 $[q_0 a q_1] \rightarrow b[q_1 a q_1],$
- 10 $[q_0 b q_1] \rightarrow b[q_1 b q_1],$
- 11 $[q_0 b q_1] \rightarrow a[q_1 b q_1],$
- 12 $[q_0 Z q_2] \rightarrow a[q_1 Z q_2],$
- 13 $[q_0 Z q_2] \rightarrow b[q_1 Z q_2],$
- 14 $[q_0 Z q_2] \rightarrow \lambda,$
- 15 $[q_1 Z q_2] \rightarrow \lambda,$
- 16 $[q_0 a q_1] \rightarrow a,$
- 17 $[q_0 b q_1] \rightarrow b,$
- 18 $[q_1 a q_1] \rightarrow a,$
- 19 $[q_1 b q_1] \rightarrow a.$

Порождение произвольного палиндрома (отличного от пустого и однобуквенного слова) происходит следующим образом. Применяем правила (2) или (3) — в зависимости от первой буквы. Затем, произвольно чередуя (4)–(7), порождаем произвольный терминальный префикс. Например:

$$\begin{aligned}
 S \vdash [q_0 Z q_2] \vdash a[q_0 a q_1][q_1 Z q_2] \vdash aa[q_0 a q_1][q_1 a q_1][q_1 Z q_2] \vdash \\
 \vdash aab[q_0 b q_1][q_1 a q_1][q_1 a q_1][q_1 Z q_2] \vdash aaba[q_0 a q_1][q_1 b q_1][q_1 a q_1][q_1 a q_1][q_1 Z q_2] \vdash \\
 \vdash aabab[q_0 b q_1][q_1 a q_1][q_1 b q_1][q_1 a q_1][q_1 a q_1][q_1 Z q_2].
 \end{aligned}$$

Справа от него образуется (в квадратных скобках) его зеркальное отражение. Если порождается палиндром четной длины, то самый левый нетерминал следует заменить согласно правилам (16) или (17) (при чтении порождаемой цепочки МП-автоматом это соответствует «угадыванию» середины слова).

В данном примере следует применить правило (17). Тогда в итоге, после «сбрасывания» всех скобок получим слово *aababbabaa*.

Если же порождается слово нечетной длины, то самый левый нетерминал следует заменить согласно одному из правил (8)–(11) (это соответствует «угадыванию» средней буквы слова). В примере могут быть применены правила (10) или (11). Тогда порождаются в итоге слова *aababbbabaa* и *aababababaa* соответственно.

Таким образом, можно показать, что все палиндромы в алфавите $\{a, b\}$ порождаются оптимизированной грамматикой и что тем самым она эквивалентна исходной, поскольку не содержит никаких правил, не принадлежащих множеству правил исходной грамматики.

3. Алгоритм поиска сбалансированных путей

Метод поиска сбалансированных путей аналогичен поиску в глубину в обычном ориентированном графе (см. [1]). Сама процедура поиска является рекурсивной. Дуги помещаются в стек в порядке их прохождения. Как только обнаруживается уже пройденная дуга, фиксируется контур и управление передается специальной процедуре анализа контуров. Исключение составляют петли с пустой магазинной меткой.

В процессе поиска с них не снимается признак новизны, и они проходятся, пока выполняется условие сцепляемости с уже пройденным путем. Каждая такая петля отвечает процессу чтения из магазина МП-автомата. Самосцепляющиеся контуры (петли в том числе) проходятся один раз.

Переходим к описанию алгоритма. Напомним, что понятие сцепления магазинных меток дуг и путей введено в работе [6]. Пример нахождения сцепления приведен ниже (см. пример 4).

Начало

```

1  Все дуги пометить как новые;
2  Для всех стартовых дуг выполнять:
    начало
3      Поместить в СТЕК;
4      Если магазинная метка дуги не пуста, то снять отметку "новая";
5      Путь := стартовая дуга;
6      Текущая метка := магазинная метка дуги;
7      Текущая длина := длина(Текущая метка);
8      Метка пути := Текущая метка;
9      Вызвать процедуру МЕТКА (Путь, Текущая метка);

```

конец

Конец

Процедура *МЕТКА* (Путь, Метка);

```

1  Для всех Дуг, сцепляющихся с Путем выполнять:
2      Если Дуга новая, то
        начало
3          Поместить Дугу в СТЕК;
4          Если Магазинная метка Дуги не пуста, то снять отметку "новая";
5          Путь := Путь + Дуга;
6          Метка := Метка "сцепить" Магазинная метка дуги;
7          Текущая длина := Текущая длина + длина(Магазинная метка) дуги – 1;
8          Вызвать процедуру МЕТКА (Путь, Метка);

```

конец

```

9      иначе вызвать процедуру КОНТУР (Путь, Дуга);
10     Удалить Дугу из СТЕКА;

```

Конец *МЕТКА*

Процедура КОНТУР (Путь, Дуга);

- 1 Считать из *СТЕКА* (без изменения содержимого) все дуги вплоть до данной Дуги; (это будут дуги, образующие найденный контур)
- 2 Вычислить магазинную метку контура;
- 3 Проверить условие самосцепляемости контура (входная магазинная метка его первой дуги совпадает с первым символом его магазинной метки);
- 4 **Если** условие выполняется, **то** выйти из процедуры;
 иначе
 начало
5 Если магазинная метка контура равна пустой цепочке, то
 начало
6 Запомнить длину Пути;
7 Пройти контур еще раз;
8 **Если** новая длина окажется меньше запомненной в п. 5, **то** проходить контур, пока выполняется условие сцепляемости с пройденным путем;
 конец
 конец
 Конец КОНТУР

Пример 4. Обратимся к мультиграфу на рис. 3. Выбираем стартовую дугу e_1 (другие стартовые дуги — $e_2, e_{14}, e_{15}, e_{16}$). С ней сцепляются дуги с входной магазинной меткой a : e_3, e_5, e_8, e_{10} . Продолжаем с дуги e_3 : вычисляем магазинную метку пути e_1e_3 (длину метки специально здесь указывать не будем):

$$aZ \otimes_a (aa) = aa(a^{-1}aZ) = aaZ.$$

С полученной меткой сцепляется любая дуга с входной магазинной меткой a , в частности снова дуга e_3 . Но поскольку один раз эта дуга, представляющая из себя самосцепляющийся контур, была пройдена, то выбираем дугу e_5 . Имеем метку пути $e_1e_3e_5$

$$aaZ \otimes_a (ba) = baaZ.$$

Этот путь можно сцепить с любой дугой с входной магазинной меткой b . По порядку номеров дуг выбираем дугу e_4 , получим путь $e_1e_3e_5e_4$ с меткой

$$baaZ \otimes_b (bb) = bbaaZ.$$

Далее, игнорируя сцепление петли e_4 с собой, переходим к дуге e_6 и вычисляем метку пути $e_1e_3e_5e_4e_6$:

$$bbaaZ \otimes_b (ab) = abbaaZ.$$

Полученный путь может быть сцеплен с любой дугой с входной магазинной меткой a , начиная с дуги e_3 . Тем самым зафиксирован контур $e_3e_5e_4e_6$. Проверяем условие самосцепляемости: метка контура равна

$$(((aa \otimes_a ba) \otimes_b (bb)) \otimes_b (ab)) = ((baa \otimes_b (bb)) \otimes_b (ab)) = bbaa \otimes_b (ab) = abbaa,$$

и условие самосцепляемости выполняется (буква a является входной магазинной меткой дуги e_3 и, одновременно, первой буквой метки контура).

Следовательно, данный контур больше не проходится, и в качестве продолжения выбираем следует выбрать дугу e_5 . Но, как нетрудно показать, возникающий при этом контур $e_5e_4e_6$ также является самосцепляющимся, и тогда следует уже выбрать продолжение e_8 (если просматривать дуги в порядке их номеров; еще подходит e_{10}). Получим:

$$abbaaZ \otimes_a a = abbaaZ$$

(заметим, что этот переход соответствует выполнению команды, читающей очередной символ ленты, но не изменяющей содержимое магазина).

Путь $e_1e_3e_5e_4e_6e_8$ (завершившийся в состоянии q_1) может быть продолжен любой дугой, исходящей из q_1 и имеющей входную магазинную метку a . В данном случае это единственная дуга e_{11} :

$$abbaaZ \otimes_a \lambda = bbaaZ.$$

Дальнейшие продолжения очевидны:

$$e_1e_3e_5e_4e_6e_8e_{11}e_{12}e_{12}e_{11}e_{11}e_{13},$$

т.е. получен сбалансированный путь с входной меткой $aabbaaabbbaa$. Полученный палиндром имеет длину 11 (т.е. нечетное количество символов), так как при прохождении пути был совершен переход, соответствующий выполнению команды чтения ленты без изменения содержимого магазина, что содержательно означает «угадывание» средней буквы палиндрома нечетной длины.

В разобранный пример прослежена только одна ветвь поиска. В силу рекурсивной природы алгоритма, он, дойдя до конца очередной ветви (т.е. получив сбалансированный путь), возвращается в ближайшую точку ветвления и рассматривает отложенные продолжения до их исчерпания и т.д.

В приведенном примере ближайшей (с конца) точкой ветвления будет путь $e_1e_3e_5e_4e_6$, после чего в качестве отложенного продолжения должна рассматриваться дуга e_{10} . Выбрав ее, получим метку пути $e_1e_3e_5e_4e_6e_{10}$, равную

$$abbaaZ \otimes_a \lambda = bbaaZ.$$

Дальнейшее единственное продолжение

$$e_1e_3e_5e_4e_6e_{10}e_{12}e_{12}e_{11}e_{11}e_{13}$$

с допуском входной цепочки $aabbaaabbbaa$ (палиндром четной длины).

Заметим в заключение, что предложенный алгоритм сам по себе может служить порождающей моделью языка, допускаемого МП-автоматом. Для этого достаточно дополнить его числовым параметром, фиксирующим максимальное число повторений самосцепляющихся контуров.

Теорема 1. Построенная по результатам алгоритма поиска сбалансированных путей грамматика эквивалентна исходной.

Доказательство. Множество правил оптимизированной грамматики есть, по построению, подмножество множества правил исходной грамматики. Это значит, что язык, порождаемый оптимизированной грамматикой, содержится (как подмножество) в языке, порождаемом исходной грамматикой.

С другой стороны [1], выводимость $[q_0 Z_0 q_f] \vdash^* x$, где x — входная цепочка, а q_f — заключительное состояние МП-автомата, имеет место тогда и только тогда, когда на множестве конфигураций МП-автомата имеет место выводимость

$$(q_0, x, Z_0) \vdash^* (q_f, \lambda, \lambda),$$

что, в свою очередь, в силу результатов работы [6], означает равенство цепочки x входной метке некоторого сбалансированного пути в мультиграфе МП-автомата, ведущего из начального состояния в заключительное. Поскольку при построении оптимизированной грамматики учитываются все такие пути, то она породит все цепочки, допускаемые МП-автоматом, и, следовательно, будет эквивалентна исходной. Теорема доказана.

4. Оценка сложности алгоритма

Теорема 2. Описанный выше алгоритм правильно находит все пути с определенной магазинной меткой (сбалансированные пути в том числе) в мультиграфе МП-автомата за время, не превышающее $O(m^3)$, где m — число дуг в мультиграфе.

Доказательство. Полнота перечисления путей сразу следует из стратегии поиска в глубину. Кроме того, ввиду ограничения на число проходов самосцепляющихся контуров, алгоритм заканчивает работу за конечное число шагов. Дадим верхнюю оценку этого числа.

Нижняя оценка, очевидно, составит число m всех дуг мультиграфа. Сверх этого основной вклад во время работы алгоритма вносят проходы самосцепляющихся контуров и контуров с пустой магазинной меткой. Проходы последних соответствуют выполнению команд чтения из магазина.

Фиксируем некоторый контур G с пустой магазинной меткой. Число N_G проходов такого контура будет не больше, чем длина магазинной цепочки, «накопленной» в магазине к моменту начала прохода контура G . В свою очередь, эта длина пропорциональна числу дуг пройденного к указанному моменту пути, включая однократное прохождение каждого встретившегося ранее самосцепляющегося контура.

Коэффициент пропорциональности не может быть больше максимальной длины L магазинной цепочки в системе команд МП-автомата. Число дуг в «линейных» участках пройденного пути (т.е. участках вне самосцепляющихся контуров) не может быть больше числа m всех дуг мультиграфа.

Это же число является грубой верхней оценкой числа дуг в самосцепляющемся контуре и числа таких контуров. В итоге для фиксированного контура G получим верхнюю оценку времени его прохождения в виде $Lm + Lm^2 = Lm(1 + m)$. Но число контуров с пустой магазинной меткой также не может быть больше m . В итоге имеем грубую верхнюю оценку сложности рассматриваемого алгоритма в виде $Lm^2(1 + m) \sim O(m^3)$. Теорема доказана.

Отметим, что полученная оценка сложности является, конечно, весьма грубой. Поиск более тонких оценок требует отдельного исследования.

Заключение

Рассмотренный алгоритм поиска сбалансированных путей в мультиграфе МП-автомата позволяет, с одной стороны, упростить конструкцию КС-грамматики, эквивалентной заданному МП-автомату, с другой — сам по себе может служить порождающей моделью для языка МП-автомата. Сложность алгоритма, оцениваемая сверху кубом числа дуг мультиграфа, позволяет считать предложенный алгоритм достаточно эффективным.

Работа выполнена при частичной финансовой поддержке Программы поддержки ведущих научных школ (грант НШ-53.2014.1).

Список литературы

1. Белоусов А.И., Ткачев С.Б. Дискретная математика. М.: Изд-во МГТУ им. Н.Э. Баумана, 2006. 744 с.
2. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. В 2 т. Т. 1: пер. с англ.: М.: Мир, 1978. 612 с.
3. Крищенко В.А. Использование LR таблиц для разбора естественного языка // Исследовано в России: электрон. журнал. 2000. Т. 3. С. 948–945. Режим доступа: <http://zhurnal.apelarn.ru/articles/2000/067.pdf> (дата обращения 25.06.2014).
4. Dantam N., Stilman M. The Motion Grammar: Analysis of a Linguistic Method for Robot Control // IEEE Transactions on Robotics. 2013. Vol. 29, no. 3. P. 704–718. DOI: [10.1109/TRO.2013.2239553](https://doi.org/10.1109/TRO.2013.2239553)
5. Frazzoli E., Dahleh M.A., Feron E. Maneuver-based motion planning for nonlinear systems with symmetries // IEEE Transactions on Robotics. 2005. Vol. 21, no. 6. P. 1077–1091. DOI: [10.1109/TRO.2005.852260](https://doi.org/10.1109/TRO.2005.852260)
6. Белоусов А.И., Ткачев С.Б. Мультиграфовое представление автоматов с магазинной памятью // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2012. № 9. DOI: [10.7463/0912.0460973](https://doi.org/10.7463/0912.0460973)
7. Пентус А.Е., Пентус М.Р. Математическая теория формальных языков. М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний. 2006. 247 с.

Construction of Context Free Grammar Based on Push-Down Automation Multigraph

07, July 2014

DOI: 10.7463/0714.0717777

Belousov A. I.¹, Tkachev S. B.^{1,2,a}

¹Bauman Moscow State Technical University, Moscow, Russian Federation

²ISA RAS, Moscow, Russian Federation

^as.tkachev@bmstu.ru

Keywords: [push-down automata](#), [PD-automata](#), [oriented multigraph](#), [context-free grammar](#), [CF-Grammar](#), [balanced way](#), [depth-first search](#)

The paper is devoted to the development of an algorithm for constructing a context-free grammar for multigraph pushdown automaton (PDA).

Development of an algorithm is based on the idea of representation the PDA as a directed multigraph. This approach optimizes context -free grammar (CFG), which can be obtained by the known algorithm using the program of PDA.

The algorithm for constructing optimized CFG is based on the enumeration of certain paths in the multigraph. A method is similar to the standard depth-first search in a directed graph. Proposed method lists so-called balanced ways, the passage of which corresponds to reading the input word by PDA until it passes a sequence of states from the initial state to the final state with the devastation store, thus allowing the input word. This frees CFG equivalent to this PDA from useless nonterminals and ensure causing all strings allowed PDA. Thus, a CFG being constructed by the algorithm will generate exactly the language that allowed by the PDA. In addition, the algorithm presented in the article can be considered as the generative model of the language permitted by PDA.

The main result of this paper is design of the algorithm for constructing optimized CFG starting with the PDA multigraph. A description of the algorithm is given using the pseudocode. It is shown that calculated upper bound for the algorithm complexity is of the cubic order of the number of the multigraph edges.

PDA multigraphs are discussed in several papers. But mostly, multigraphs are used solely as an illustration. In this paper a multigraph is used not only as a visual representation of PDA, but

as a working model, which is defined in terms of the language of PDA. The equivalence of such a model to the known models is proved too.

Thus, the paper proposes an efficient algorithm for enumerating paths in a directed multigraph, on the basis of which it is possible to build an optimized CFG equivalent to the original PDA.

Among the tasks which will continue the started study there is a more accurate upper bound finding on the algorithm complexity.

References

1. Belousov A.I., Tkachev S.B. *Diskretnaya matematika* [Discrete mathematics]. Moscow, Bauman MSTU Publ., 2006. 744 p. (in Russian).
2. Aho A.V., Ullman J.D. *The Theory of Parsing, Translation and Compiling. Vol. 1. Parsing*. Printice Hall, 1972. (Russ. ed.: Aho A.V., Ullman J.D. *Teoria sintaksicheskogo analiza, perevoda i kompiliatsii. V 2 t. T. 1. Sintaksicheskii analiz*. Moscow, Mir Publ., 1978. 612 p.)
3. Krishchenko V.A. [Using of LR tables for analysis of a natural language]. *Issledovano v Rossii – Investigated in Russia*, 2000, vol. 3, pp. 948–945. Available at: <http://zhurnal.apelarn.ru/articles/2000/067.pdf>, accessed 05.07.2014. (in Russian).
4. Dantam N., Stilman M. The Motion Grammar: Analysis of a Linguistic Method for Robot Control. *IEEE Transactions on Robotics*, 2013, vol. 29, no. 3, pp. 704–718. DOI: [10.1109/TRO.2013.2239553](https://doi.org/10.1109/TRO.2013.2239553)
5. Frazzoli E., Dahleh M.A., Feron E. Maneuver-based motion planning for nonlinear systems with symmetries. *IEEE Transactions on Robotics*, 2005, vol. 21, no. 6, pp. 1077–1091. DOI: [10.1109/TRO.2005.852260](https://doi.org/10.1109/TRO.2005.852260)
6. Belousov A.I., Tkachev S.B. [Presentation of push-down automation as oriented multigraphs]. *Nauka i obrazovanie MGTU im. N.E. Bauman – Science and Education of the Bauman MSTU*, 2012, no. 9. DOI: [10.7463/0912.0460973](https://doi.org/10.7463/0912.0460973) (in Russian).
7. Pentus A.E., Pentus M.R. *Matematicheskaya teoriya formal'nykh yazykov* [The mathematical theory of formal languages]. Moscow, Internet-University of Information Technologies Publ.; BINOM. Laboratoria znaniy Publ., 2006. 247 p. (in Russian).