

СОКРАЩЕНИЕ КОЛИЧЕСТВА МАГАЗИННЫХ СИМВОЛОВ В РАСПОЗНАВАТЕЛЯХ С МАГАЗИННОЙ ПАМЯТЬЮ И ОДНИМ СОСТОЯНИЕМ

Ryazanov.iurij@yandex.ru

В статье рассматривается задача эквивалентного преобразования распознавателя с магазинной памятью и одним состоянием в более компактный распознаватель. Уменьшение размера распознавателя достигается за счет сокращения количества магазинных символов. Для сокращения количества магазинных символов вводится отношение на множестве этих символов, обладающее свойством эквивалентности, такое, что “стягивание” класса эквивалентности в один символ дает распознаватель, эквивалентный исходному. Предложен алгоритм разбиения множества магазинных символов на классы эквивалентности и алгоритм построения распознавателя, эквивалентного исходному, но с меньшим (не большим) количеством магазинных символов. Предложенный алгоритм может быть использован при разработке программ обработки формальных языков.

Ключевые слова: контекстно-свободный язык, распознаватель с магазинной памятью, магазинный символ, переход, эквивалентные преобразования.

Введение. В теории формальных языков важное место занимают контекстно-свободные языки, проблема распознавания для которых решается с помощью распознавателей с магазинной памятью (МП-распознаватели) [1–6]. В работе рассматривается один из классов МП-распознавателей с одним состоянием. Два различных МП-распознавателя называются эквивалентными, если равны распознаваемые ими языки. Эквивалентные МП-распознаватели могут иметь разный размер. На практике предпочтение отдается распознавателям меньшего размера. Уменьшить размер МП-распознавателя с одним состоянием можно за счет сокращения магазинных символов. В работе решается задача преобразования заданного МП-распознавателя с одним состоянием в эквивалентный ему распознаватель с меньшим (не большим) количеством магазинных символов. Для решения этой задачи вводится отношение на множестве магазинных символов МП-распознавателя, обладающее свойством эквивалентности, показывается, что преобразование МП-распознавателя, заключающееся в “стягивании” пары символов, принадлежащих введенному отношению, в один символ, не изменяет распознаваемого языка, т. е. является эквивалентным преобразованием. От-

сюда следует, что “стягивание” класса эквивалентности в один символ также является эквивалентным преобразованием. Предлагается алгоритм разбиения множества магазинных символов МП-распознавателя на классы эквивалентности и алгоритм построения МП-распознавателя, эквивалентного исходному, с не большим количеством магазинных символов, чем исходный МП-распознаватель.

Определение распознавателя с магазинной памятью и одним состоянием. В работе будем рассматривать класс МП-распознавателей [7 – 9], которые можно построить по диаграммам Вирта [7] или синтаксическим диаграммам с многоходовыми компонентами [10]. МП-распознаватель из этого класса формально можно представить следующим образом:

$$M = (X, \Gamma, I, P, W, \delta, \lambda, g_0, \gamma_0) \quad ,$$

где X – конечное множество входных символов, включающее концевой маркер \dagger , которым заканчивается входная цепочка, $X = \{x_1, x_2, \dots, x_n, \dagger\}$; Γ – конечное множество магазинных символов, включающее маркер дна магазина ∇ , $\Gamma = \{g_0, g_1, g_2, \dots, g_m, \nabla\}$; I – конечное множество операций над головкой, $I = (\text{сдвиг}, \text{держать})$; P – множество операций над магазином,

$$P = \{\text{заменить}(g_i) \mid g_i \in \Gamma\} \cup \{\text{заменить}(g_i g_j) \mid g_i \in \Gamma \text{ и } g_j \in \Gamma\} \cup \{\text{вытолкнуть}\} \cup \{\text{не изменять}\}.$$

W – конечное множество значений выхода, $W = \{\text{допустить}, \text{отвергнуть}\}$; g_0 – начальный символ магазина, $g_0 \in \Gamma$; γ_0 – начальное содержимое магазина, $\gamma_0 = \nabla g_0$ (магазин содержит маркер дна и начальный символ магазина);

$\delta : X \times \Gamma \rightarrow P \times I$ – частичная функция переходов, которая символу входной цепочки (нахо-

дящемуся под головкой) и верхнему символу магазина ставит в соответствие операцию над головкой и магазином, причем множество видов значений на двойке (x, g_s) равно $\{(\text{заменить}(g_i), \text{сдвиг}), (\text{заменить}(g_i g_j), \text{держать}), (\text{вытолкнуть}, \text{держать})\}$. Если значение функции на двойке (x, g_s) равно $(\text{заменить}(g_i), \text{сдвиг})$, то такой пере-

ход будем обозначать $(g_s, x, (g_i, \rightarrow))$, если значение функции на этой двойке равно (*заменить*(g_i, g_j), *держат*), то переход обозначим $(g_s, x, (g_i, g_j))$, если же значение на этой двойке равно (*втолкнуть*, *держат*), то переход обозначим (g_s, x, \uparrow) .

$\lambda : X \times \Gamma \rightarrow W$ – частичная функция выходов, которая символу входной цепочки (находящемуся под головкой) и верхнему символу магазина ставит в соответствие значение выхода – *допустить* или *отвергнуть*. Значение функции на двойке (\uparrow, ∇) равно *допустить*, а на всех остальных, на которых функция определена – *отвергнуть*.

Области определения функций δ и λ не пересекаются, а их объединение равно области отправления.

Двойка $(\alpha, \nabla \gamma)$, где α – часть входной цепочки, начиная с символа под головкой и заканчивая концевым маркером, γ – содержимое магазина, называется конфигурацией МП-распознавателя. Исходной конфигурацией является $(\alpha_0, \nabla g_0)$, где α_0 – вся входная цепочка (головка находится над первым символом).

Пусть конфигурацией МП-распознавателя является двойка $(x\alpha, \nabla \gamma g_s)$, где x – символ под головкой, g_s – верхний символ магазина. Если на

двойке (x, g_s) определена функция переходов δ , то ее значение определяет операции над головкой и магазином. При выполнении этих операций конфигурация изменяется. Если на двойке (x, g_s) определена функция выходов λ , то процесс распознавания заканчивается с результатом, равным значению функции λ . Такую конфигурацию назовем заключительной. Итак, работа МП-распознавателя заключается в изменении конфигураций. Последней является заключительная конфигурация, в которой определяется результат распознавания.

МП-распознаватель с одним состоянием можно задать таблицей T , в которой строки соответствуют входным символам, а столбцы – магазинным символам. Клетку таблицы T в строке x и столбце g_s будем обозначать $T[x, g_s]$. Клетки таблицы T заполняются по следующим правилам. Если на паре (x, g_s) определен переход $(g_s, x, (g_i, \rightarrow))$, то в клетку $T[x, g_s]$ запишем (g_i, \rightarrow) , если определен переход $(g_s, x, (g_i, g_j))$, то в клетку $T[x, g_s]$ запишем (g_i, g_j) , и, если определен переход (g_s, x, \uparrow) , то в клетку $T[x, g_s]$ запишем \uparrow . В клетку $T[\uparrow, \nabla]$ запишем «доп», она соответствует выходу *допустить*. Пустые клетки соответствуют выходу *отвергнуть*. Пример МП-распознавателя представлен в табл.1.

Таблица 1

Таблица МП-распознавателя

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	∇
a	4 →	6 →	9 →	10,2	11,3	12,2		13,3	16 →	14 →		15 →			↑		
b	5 →	7 →	8 →	10,2	11,3	12,2	15 →	13,3			14 →		16 →			↑	
\uparrow														↑			доп

Для обеспечения большей возможности сокращения количества магазинных символов, преобразуем МП-распознаватель следующим образом: если в какой либо клетке некоторого столбца распознавателя записан символ \uparrow , то все пустые клетки этого столбца заполнить символом \uparrow . Такое преобразование не изменит распо-

знаваемого языка, оно лишь отложит обнаружение ошибки на некоторое число шагов, причем без продвижения по входной цепочке. Выполнив это преобразование для МП-распознавателя (табл. 1), получим МП-распознаватель, представленный в табл. 2.

Таблица 2

Таблица МП-распознавателя

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	∇
a	4 →	6 →	9 →	10,2	11,3	12,2		13,3	16 →	14 →		15 →		↑	↑	↑	
b	5 →	7 →	8 →	10,2	11,3	12,2	15 →	13,3			14 →		16 →	↑	↑	↑	
\uparrow														↑	↑	↑	доп

Отношение R на множестве магазинных символов, обладающее свойством эквивалентности. Пара символов g_m и g_m' принадлежит отношению R если:

1) переход $(g_m, x, (g_i, \rightarrow))$ существует тогда и только тогда, когда существует переход $(g_m', x, (g_i', \rightarrow))$ и $(g_i, g_i') \in R$;

2) переход $(g_m, x, (g_i, g_j))$ существует тогда и только тогда, когда существует переход $(g_m', x, (g_i', g_j'))$ и $(g_i, g_i') \in R$ и $(g_j, g_j') \in R$;

3) переход (g_m, x, \uparrow) существует тогда и только тогда, когда существует переход (g_m', x, \uparrow) .

Это отношение рефлексивно, симметрично и транзитивно.

Пару $(g_m, g_m') \in R$ будем называть парой эквивалентных символов или говорить, что символы g_m и g_m' эквивалентны.

Из третьего условия эквивалентности состояний следует, что если для символов g_m и g_m' существуют только переходы с выполнением операции *вытолкнуть*, то символы g_m и g_m' эквивалентны. В остальных же случаях для эквивалентности символов g_m и g_m' необходима эквивалентность других пар символов.

Эквивалентное преобразование распознавателя с магазинной памятью, сокращающее количество магазинных символов. Рассмотрим МП-распознаватель M_1 , в котором есть пара эквивалентных магазинных символов (g_m, g_m') , представленный в виде таблицы.

Построим таблицу МП-распознавателя M_2 следующим образом:

1) возьмем таблицу МП-распознавателя M_1 и удалим в ней столбец, соответствующий магазинному символу g_m ;

2) все вхождения символа g_m в таблице распознавателя заменим на g_m' .

Такое преобразование МП-распознавателя M_1 условно можно назвать “стягиванием” символов g_m и g_m' в один символ g_m' .

Пусть МП-распознаватели M_1 и M_2 обрабатывают цепочку α и на некотором шаге вверху магазина распознавателя M_1 оказался символ g_m , а вверху магазина распознавателя M_2 — символ g_m' . К этому моменту МП-распознаватели M_1 и M_2 обработали одинаковое количество символов цепочки α и содержимое магазинов этих распознавателей одинаковое. Из определения эквивалентности символов g_m и g_m' следует:

1) если M_1 выполнит переход $(g_m, x, (g_i, \rightarrow))$, то M_2 выполнит переход $(g_m', x, (g_i', \rightarrow))$, после чего верхние символы в магазинах распознавателей M_1 и M_2 будут эквивалентными и распознаватели будут обрабатывать символ входной цепочки, следующий за символом x ;

2) если M_1 выполнит переход $(g_m, x, (g_i, g_j))$, то M_2 выполнит переход $(g_m', x, (g_i', g_j'))$, после чего в магазинах эквивалентными будут верхние символы, и символы, находящиеся непосредственно под верхними символами и распознаватели будут обрабатывать тот же символ x входной цепочки;

3) если M_1 выполнит переход (g_m, x, \uparrow) , то M_2 выполнит переход (g_m', x, \uparrow) , после чего вверху магазинов будут эквивалентные (или одинаковые) символы и распознаватели будут обрабатывать тот же символ x входной цепочки;

4) если M_1 отвергнет цепочку α , то M_2 также отвергнет эту цепочку.

Таким образом, на каждом шаге работы МП-распознавателей M_1 и M_2 :

1) обработанная часть цепочки α распознавателем M_1 равна обработанной части цепочки α распознавателем M_2 ;

2) количество символов в магазине распознавателя M_1 равно количеству символов в магазине распознавателя M_2 , причем i -й символ в магазине распознавателя M_1 эквивалентен (или равен) i -му символу в магазине распознавателя M_2 .

Следовательно, если цепочка α допускается (отвергается) одним из распознавателей, то она допускается (отвергается) другим распознавателем, т. е. распознаватели M_1 и M_2 распознают один и тот же язык.

Если в МП-распознавателе существует класс эквивалентных магазинных символов, в котором количество символов больше двух, то последовательное “стягивание” пар эквивалентных символов приведет к “стягиванию” класса эквивалентных символов в один символ. “Стягивание” каждого класса эквивалентных символов в один символ позволяет сократить количество магазинных символов МП-распознавателя.

Разбиение множества магазинных символов на классы эквивалентности по отношению R . Отношение R обладает свойством эквивалентности и определяет разбиение множества магазинных символов МП-распознавателя на классы эквивалентности. Для разбиения множества магазинных символов на классы эквивалентности применим следующий метод.

С учетом условий принадлежности пары символов отношению R будем последовательно разбивать множество магазинных символов на подмножества так, чтобы неэквивалентные символы попали в различные подмножества.

Рассмотрим более детально условия принадлежности пары символов (g_m, g_m') отношению R .

Первое условие.

Первое условие состоит из двух частей:

1 часть: переход $(g_m, x, (g_i, \rightarrow))$ существует тогда и только тогда, когда существует переход $(g_m', x, (g_i', \rightarrow))$;

2 часть: $(g_i, g_i') \in R$.

Это условие истинно, если истинны обе части.

Второе условие.

Второе условие состоит из двух частей:

1 часть: переход $(g_m, x, (g_i, g_j))$ существует тогда и только тогда, когда существует переход $(g_m', x, (g_i', g_j'))$;

2 часть: $(g_i, g_i') \in R$ и $(g_j, g_j') \in R$.

Это условие истинно, если истинны обе части.

Третье условие.

Переход (g_m, x, \uparrow) существует тогда и только тогда, когда существует переход (g_m', x, \uparrow) .

Сначала разобьем множество магазинных символов на подмножества так, чтобы для любых двух символов подмножества истинными были третье условие и только первые части первых двух условий. Для МП-распознавателя (см. табл. 2) это будут подмножества: $G_1 = \{1,2,3\}$, $G_2 = \{4,5,6,8\}$, $G_3 = \{7,11,13\}$, $G_4 = \{9,10,12\}$, $G_5 = \{14,15,16\}$. Эти подмножества образуют разбиение $H = \{G_1, G_2, G_3, G_4, G_5\}$.

Символы, принадлежащие различным подмножествам, явно неэквивалентны, но два символа из одного подмножества могут быть неэквивалентными, так как при разбиении мы не учитывали вторые части первого и второго условия принадлежности пары символов отношению R .

Для учета вторых частей условий построим таблицу S , в которой столбцы соответствуют магазинным символам МП-распознавателя (не включая маркер дна), а строки – входным символам (включая концевой маркер).

Клетку таблицы S в строке x и столбце g_m будем обозначать $S[x, g_m]$.

Переходу $(g_m, x, (g_n, \rightarrow))$ соответствует клетка $S[x, g_m]$. Если $g_n \in G_i$, то в клетку $S[x, g_m]$ запишем i . Если символы g_m и g_m' принадлежат одному подмножеству и существуют переходы $(g_m, x, (g_n, \rightarrow))$ и $(g_m', x, (g_n', \rightarrow))$ и $S[x, g_m] \neq S[x, g_m']$, то это говорит о ложности второй части первого условия и, следовательно, $(g_m, g_m') \notin R$ и символы g_m и g_m' нужно включить в разные подмножества нового разбиения H' .

Переходу $(g_m, x, (g_s, g_n))$ соответствует клетка $S[x, g_m]$. Если $g_s \in G_i$ и $g_n \in G_j$, то в клетку $S[x, g_m]$ запишем (i, j) . Если символы g_m и g_m' принадлежат одному подмножеству и существуют переходы $(g_m, x, (g_s, g_n))$ и $(g_m', x, (g_s', g_n'))$ и $S[x, g_m] \neq S[x, g_m']$, то это говорит о ложности второй части второго условия и, следовательно, $(g_m, g_m') \notin R$ и символы g_m и g_m' нужно включить в разные подмножества нового разбиения H' .

Анализируя таблицу S , формируем новое разбиение H' . Подмножество G_i' разбиения H' формируется из элементов некоторого подмножества G_j разбиения H . G_i' является максимальным по мощности подмножеством множества G_j , таким, что для каждой пары символов g_m и g_m' , принадлежащих подмножеству G_i' , для всех строк l таблицы S верно, что $S[l, g_m] = S[l, g_m']$.

Если $H' \neq H$, то принимаем, что $H = H'$, заново строим таблицу S по описанным выше правилам и формируем новое разбиение.

Если $H' = H$, то H' представляет собой множество классов эквивалентности.

Первая таблица S для МП-распознавателя (см. табл. 2) представлена в табл. 3.

Таблица 3

Таблица для формирования разбиения

	G_1			G_2				G_3			G_4			G_5		
	1	2	3	4	5	6	8	7	11	13	9	10	12	14	15	16
a	2	2	4	4	3	4	3				5	5	5			
b	2	3	2	4	3	4	3	5	5	5						
\dagger				1	1	1	1									

По этой таблице видно, что подмножество $\{1,2,3\}$ разбивается на одноэлементные подмножества $\{1\}$, $\{2\}$ и $\{3\}$, подмножество $\{4,5,6,8\}$ – на $\{4,6\}$ и $\{5,8\}$, а остальные под-

множества не разбиваются и в результате получаем разбиение $\{\{1\}, \{2\}, \{3\}, \{4,6\}, \{5,8\}, \{9,10,12\}, \{14,15,16\}\}$. По этому разбиению строим новую таблицу S (табл. 4).

Таблица 4

Таблица для формирования разбиения

	G_1	G_2	G_3	G_4		G_5		G_6			G_7			G_8		
	1	2	3	4	6	5	8	7	11	13	9	10	12	14	15	16
a	4	4	7	7	7	6	6				8	8	8			
b	5	6	5	7	7	6	6	8	8	8						
\dagger				2	2	3	3									

В этой таблице нет различных столбцов, соответствующих символам, принадлежащим какому либо одному подмножеству. Поэтому полученное разбиение является разбиением на классы эквивалентности.

Построение распознавателя с магазинной памятью и меньшим количеством магазинных символов. Используя отношение R на множестве состояний МП-распознавателя M_1 , можно построить МП-распознаватель M_2 , эквивалентный распознавателю M_1 , который будет иметь не больше магазинных символов, чем распознаватель M_1 . Сокращение количества магазинных символов достигается за счет “стягивания” всех символов, принадлежащих одному классу эквивалентности, в один символ.

МП-распознаватель M_2 строим следующим образом.

Возьмем по одному символу из каждого класса эквивалентности, добавим маркер дна и получим множество магазинных символов МП-распознавателя M_2 .

Возьмем таблицу МП-распознавателя M_1 и оставим в ней только те столбцы, которые соответствуют магазинным символам МП-распознавателя M_2 . Если в клетке полученной таблицы встретится магазинный символ МП-распознавателя M_1 , который не принадлежит множеству магазинных символов МП-распознавателя M_2 , то его нужно заменить на эквивалентный ему символ, принадлежащий множеству магазинных символов МП-распознавателя M_2 .

На множестве магазинных символов МП-распознавателя (см. табл. 2) построено разбиение на классы эквивалентности $\{\{1\}, \{2\}, \{3\}, \{4,6\}, \{5,8\}, \{9,10,12\}, \{14,15,16\}\}$. Из каждого класса выберем символ с меньшим номером и построим таблицу (табл. 5) нового МП-распознавателя по описанным выше правилам. В результате получим МП-распознаватель с меньшим количеством магазинных символов.

Таблица 5

Таблица МП-распознавателя

	1	2	3	4	5	7	9	14	▽
a	4 →	4 →	9 →	9,2	7,3		14 →	↑	
b	5 →	7 →	5 →	9,2	7,3	14 →		↑	
†								↑	доп

Выводы. В статье решена задача преобразования исходного МП-распознавателя в эквивалентный ему распознаватель с меньшим (не большим) количеством магазинных символов. Но задача получения МП-распознавателя с наименьшим количеством магазинных символов, распознающим заданный язык, остается открытой. Для дальнейшего сокращения количества магазинных символов может быть использовано какое либо другое отношение эквивалентности на множестве магазинных символов распознавателя, разбивающее множество символов на меньшее количество классов эквивалентности, такое, что “стягивание” класса эквивалентности в один символ приводит к получению распознавателя, эквивалентного исходному.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Schutzenberger M.P. On context-free languages and pushdown automata. *Information and Control*. 1963. 6:3. P. 246 – 264.
2. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. М.: Мир, 1978. Т. 1, 612 с.

3. Льюис Ф., Розенкранц Д., Стирнз Р. Теоретические основы проектирования компиляторов. М.: Мир, 1979. 656 с.
4. Мартыненко Б.К. Синтаксически управляемая обработка данных. Изд. 2-е, дополн. СПб: Изд-во С.-Петербургского университета, 2004 г. 317 с.
5. Ахо А, Лам М., Сети Р., Ульман Дж. Компиляторы. Принципы, технологии и инструментарий. М: Издательский дом “Вильямс”, 2008. 1185 с.
6. Hopcroft J.E., Motwani R., J.D. Ullman J.D. *Introduction to Automata Theory, Languages, and Computation* (3rd ed.). Pearson., 2013. p. 496.
7. Рязанов Ю. Д. Синтез распознавателей с магазинной памятью по детерминированным синтаксическим диаграммам // Вестник ВГУ. Системный анализ и информационные технологии. 2014. №1. С. 138 – 145.
8. Рязанов Ю.Д., Савёлова И.Н. Преобразование распознавателя с магазинной памятью и одним состоянием в распознаватель с конечным множеством состояний // Моделирование, оптимизация и информационные технологии. 2015. № 4 (11). С. 13.

9. Рязанов Ю. Д. Преобразование распознавателя с магазинной памятью и конечным множеством состояний в распознаватель с одним состоянием // Вестник БГТУ им. В.Г. Шухова. 2016. № 1. С. 194 – 199.

10. Polyakov V.M., Ryazanov Y.D. Virt Charts to Multiport Component Syntactic Charts Transformation // Global Journal of Pure and Applied Mathematics. 2015. Т. 11. № 5. С. 3939 – 3952.

Ryazanov Yu.D.

REDUCING THE NUMBER OF PUSHDOWN SYMBOLS IN ONE-STATE PUSHDOWN RECOGNIZERS

The article deals with the task of equivalent transforming of a one-state pushdown recognizer into a more compact recognizer. The size of a recognizer is reduced due to reducing the number of pushdown symbols. To reduce the number of pushdown symbols a relation for a set of symbols is introduced, which possesses the equivalence property, so that “contraction” of an equivalence class into a symbol produces a recognizer, equivalent to the initial one. An algorithm of partitioning a set of pushdown symbols to equivalence classes and an algorithm of developing a recognizer, equivalent to the initial one, but with smaller (not larger) number of pushdown symbols, have been suggested. The suggested algorithm can be used at developing formal language processing programs.

Key words: context-free language, pushdown recognizer, pushdown symbol, transition, equivalent transforming.

Рязанов Юрий Дмитриевич, аспирант кафедры программного обеспечения вычислительной техники и автоматизированных систем.

Белгородский государственный технологический университет им. В.Г. Шухова.

Адрес: Россия, 308012, Белгород, ул. Костюкова, д. 46.

E-mail: Ryazanov.iurij@yandex.ru