УДК 519.685.3

Ю.Д. Рязанов, И.Н. Савёлова

ПРЕОБРАЗОВАНИЕ РАСПОЗНАВАТЕЛЯ С МАГАЗИННОЙ ПАМЯТЬЮ И ОДНИМ СОСТОЯНИЕМ В РАСПОЗНАВАТЕЛЬ С КОНЕЧНЫМ МНОЖЕСТВОМ СОСТОЯНИЙ

Белгородский государственный технологический университет им. В.Г. Шухова

В работе определяется класс распознавателей с магазинной памятью и одним состоянием, которые могут быть преобразованы в эквивалентные распознаватели с магазинной памятью и конечным множеством состояний и предлагается алгоритм преобразования. Распознаватель с конечным множеством состояний выполняет меньше операций над магазином, чем эквивалентный ему распознаватель с одним состоянием.

Ключевые слова: контекстно-свободный язык, распознаватель с магазинной памятью, состояние, эквивалентные преобразования.

Одной из задач обработки формальных языков является задача распознавания, которая заключается в определении принадлежности заданной цепочки заданному языку. Для решения задачи распознавания контекстно-свободных языков используются распознаватели с магазинной памятью (МП-распознаватели). МП-распознаватель определяется пятью объектами [1]:

1) конечным множеством входных символов, в которое входит и концевой маркер (┤);

2) конечным множеством магазинных символов, включающим маркер дна (∇);

3) конечным множеством состояний, включающим начальное состояние;

4) управляющим устройством, которое каждой комбинации входного символа, магазинного символа и состояния ставит в соответствие переход или выход. Переход заключается в выполнении операций над магазином, состоянием и входом. Над магазином выполняются операции *втолкнуть*(m), где m – магазинный символ, *вытолкнуть* и расширенная операция*заменить*, которая заключается в выталкивании верхнего символа магазина и последующем выполнении нескольких вталкиваний. Над состоянием выполняется единственная операция – переход в состояние. Над входом возможны операция *сдвиг*, которая заключается в переходе к следующему символу в обрабатываемой цепочке, и *держать*, которая указывает, что на следующем шаге работы МП-распознавателя будет обрабатываться тот же символ входной цепочки.МП-распознаватель имеет два выхода — *допустить* и *отвергнуть*;

5) начальным содержимым магазина, которое представляет собой маркер дна, за которым следует (возможно, пустая) цепочка других магазинных символов.

В классической теории компиляторов для распознавания формальных языков обычно используются МП-распознаватели с одним состоянием [1, 2]. МП-распознаватель с одним состоянием на каждом шаге обработки входной цепочки выполняет операции над магазином. МП-распознаватель с конечным множеством состояний может выполнить шаг обработки входной цепочки без анализа и изменения магазина, поэтому он может обработать входную цепочку быстрее, чем МП-распознаватель с одним состоянием.

В статье рассматривается только класс МП-распознавателей с одним состоянием ($МП^1$), синтез которых описан в работе [3], и предлагается алгоритм преобразования МП-распознавателей, принадлежащих этому классу, в эквивалентные МП-распознаватели с конечным множеством состояний ($МП^n$).

Для рассматриваемого класса $МП^1$-распознавателей характерно следующее:

1) начальное содержимое магазина представляет собой маркер дна, за которым следует один магазинный символ;

2) на каждом шаге обработки входной цепочки над магазином выполняется одна операция из множества {*вытолкнуть*, *заменить*(*x*), *заменить*(*x*,*y*)}, т. е. магазинный символ может быть вытолкнут либо заменен на один или два символа;

3) операция *сдвиг* над входной цепочкой выполняется только тогда, когда верхний символ магазина заменяется одним символом;

4) цепочка допускается, если после обработки всех ее символов магазин становится пустым.

Пример распознавателя рассматриваемого класса, в котором {*a*,*b*,*c*,*d*,*e*} – множество входных символов, {1,2,3,4,5,6,7,8,9,10,11} – множество магазинных символов, маркер дна и символ 1 – начальное содержимое магазина, представлен в табл. 1. Здесь строки соответствуют магазинным символам, столбцы – входным символам, а в клетках записаны операции. Операция *держать* не указана, а пустым клеткам соответствует выход *отвергнуть*.

Будем считать, что магазинные символы $МП^1$-распознавателя представляют собой состояния, тогда работу распознавателя можно представить следующим образом.

Выполнение операций *зам*(*x*), *сдвиг* при верхнем магазинном символе *a* и входном символе *z* соответствует переходу из состояния *a* в состояние *x* при входном символе *z* и продвижению по входной цепочке.

Выполнение операций *зам*(*x,y*), *держать* при верхнем магазинном символе *a* и входном символе *z* соответствует переходу из состояния *a* в состояние *y* с запоминанием состояния *x* в магазине при входном символе *z* без продвижения по входной цепочке.

Таблица 1

МП-распознаватель с одним состоянием

	a	*b*	*c*	*d*	*e*	┤
1	*зам* (3) *сдвиг*	*зам* (2 5)	*зам* (2 5)	*зам* (2 5)	*зам* (2 5)	
2			*зам* (4) *сдвиг*			
3				*зам* (4 9)	*зам* (4 9)	
4				*зам* (2 9)	*зам* (2 9)	*вытолкнуть*
5		*зам* (6) *сдвиг*	*вытолкнуть*	*зам* (7 9)	*зам* (7 9)	
6				*зам* (8 9)	*зам* (8 9)	
7				*зам* (8) *сдвиг*		
8	*зам* (5) *сдвиг*		*вытолкнуть*			
9				*зам* (11) *сдвиг*	*зам* (10) *сдвиг*	
10				*зам* (11 9)	*зам* (11 9)	
11	*вытолкнуть*		*вытолкнуть*	*вытолкнуть*	*вытолкнуть*	*вытолкнуть*
∇						*допустить*

Выполнение операций *вытолкнуть*, *держать* при верхнем магазинном символе *a* и входном символе *z* соответствует переходу из состояния*a* в состояние, которое находится в магазине под верхним символом при входном символе *z* без продвижения по входной цепочке.

Состояние *a* является допускающим, если символ *a* может находится сразу над маркером дна магазина и если он верхний и цепочка закончилась, распознаватель выполняет действие *вытолкнуть*.

Начальным состоянием является символ, который находится вверху магазина вначале обработки входной цепочки.

МП^n-распознаватель с конечным множеством состояний можно задать таблицей, состоящей из четырех столбцов. В первом столбце указывается состояние, во втором – множество входных символов, в третьем – магазинный символ или пусто, в четвертом – действия, которые

должен выполнить распознаватель в ситуации, определяемой первыми тремя столбцами.

Алгоритм преобразования МП^1-распознавателя с одним состоянием в МП^n-распознаватель с конечным множеством состояний следующий.

Последовательно обработать все клетки таблицы МП^1-распознавателя.

Если в строке *a* и столбце *z* таблицы МП^1-распознавателя записаны действия *заменить*(*x*) и *сдвиг,* то в таблицу МП^n-распознавателя добавить строку, содержащую в первом столбце состояние *a*, во втором – входной символ *z*, в третьем – пусто, в четвертом – действия *состояние*(*b*) и *сдвиг.*

Если в строке *a* и столбце *z* таблицы МП^1-распознавателя записано действие *заменить*(*x*,*y*), то в таблицу МП^n-распознавателя добавить строку, содержащую в первом столбце состояние *a*, во втором – входной символ *z*, в третьем – пусто, в четвертом – действия *состояние*(*y*) и *втолкнуть*(*x*).

Если в строке *a* и столбце *z* таблицы МП^1-распознавателя записано действие *вытолкнуть*, то сначала нужно найти множество символов *M,* которые могут быть в магазине под символом *a*. Это будут состояния, в которые перейдет МП^n-распознаватель из состояния *a* по входному символу *z*. Для этого определим множество символов *M*′, под которыми в магазине могут быть такие же символы, как и под *a* по следующему алгоритму.

1. $M' := \{a\}$.

2. Если в таблице *МП*^{1} есть строка *b* с действием *заменить*(*x*) или *заменить*(*x*,*y*), где $x \in M'$, то символ *b* добавить в множество *M*′.

3. Повторять п.2 пока множество *M*′ растет.

Затем находим множество *M* по правилу: если в таблице *МП*^{1} есть действие *заменить*(*x*,*y*), где $y \in M'$, то символ *x* добавить в изначально пустое множество *M*. Множество *M* найдено и теперь для каждого элемента $x \in M$ в таблицу МП^n добавляем строку, содержащую в первом столбце состояние *a*, во втором – входной символ *z*, в третьем – символ *x*, в четвертом – *состояние*(*x*) и *вытолкнуть*.

Допускающему состоянию соответствует символ *b*, который может находится сразу над маркером дна магазина и если он верхний и цепочка закончилась (входной символ ⊣), распознаватель выполняет действие *вытолкнуть*. Найти множество *M* символов, которые могут находиться в магазине сразу над маркером дна, можно по следующему алгоритму.

1. $M := \{s\}$, где *s* – символ, который находится над маркером дна магазина вначале обработки входной цепочки.

2. Если в строке *b* ($b \in M$) таблицы *МП*^{1}-распознавателя есть действие *заменить*(*x*) или *заменить*(*x*,*y*), то символ *x* добавить в множество *M*.

3. Повторять п.2 пока множество *M* растет.

Если в строке *b* ($b \in M$) таблицы *МП*[1]-распознавателя и столбце ⊣ записано *вытолкнуть*, то состояние *b* считать допускающим и в таблицу МП[n]-распознавателя добавить строку, содержащую в первом столбце состояние *b*, во втором – концевой маркер ⊣, в третьем – маркер дна ∇, в четвертом – *допустить*.

Для сокращения количества строк в таблице МП[n]-распознавателя каждые две строки, которые различаются только во втором столбце, объединяются в одну строку.

Начальное содержимое магазина – магазин пуст.

В результате применения алгоритма к МП[1]-распознавателю (табл. 1) получим МП[n]-распознаватель, представленный в табл. 2.

Таблица 2

МП-распознаватель с конечным множеством состояний

Текущее состояние	Входные символы	Верх магазина	Действия
1	*a*		*сост*(3), *сдвиг*
1	*b, c, d, e*		*сост*(5), *втолкнуть*(2)
2	*c*		*сост*(4), *сдвиг*
3	*d, e*		*сост*(9), *втолкнуть*(4)
4	*d, e*		*сост*(9), *втолкнуть*(2)
4	⊣	∇	*допустить*
5	*b*		*сдвиг, сост*(6)
5	*c*	2	*сост*(2), *вытолкнуть*
5	*d, e*		*сост*(9), *втолкнуть*(7)
6	*d, e*		*сост*(9), *втолкнуть*(8)
7	*d*		*сдвиг, сост*(8)
8	*c*	2	*сост*(2), *вытолкнуть*
8	*a*		*сост*(5), *сдвиг*
9	*e*		*сост*(10) , *сдвиг*
9	*d*		*сост*(11), *сдвиг*
10	*d, e*		*сост*(9), *втолкнуть*(11)
11	*a, c, d, e,* ⊣	2	*сост*(2), *вытолкнуть*
11	*a, c, d, e,* ⊣	4	*сост*(4), *вытолкнуть*
11	*a, c, d, e,* ⊣	7	*сост*(7), *вытолкнуть*
11	*a, c, d, e,* ⊣	8	*сост*(8), *вытолкнуть*
11	*a, c, d, e,* ⊣	11	*сост*(11), *вытолкнуть*

Здесь состояние 1 – начальное, состояние 4 – допускающее, начальное содержимое магазина – магазин пуст.

Таким образом определен класс МП[1]-распознавателей с одним состоянием, которые могут быть преобразованы в эквивалентные МП[n]-распознаватели с конечным множеством состояний и предложен алгоритм преобразования. Распознаватель с конечным множеством состояний

выполняет меньше операций над магазином, чем эквивалентный ему распознаватель с одним состоянием.

ЛИТЕРАТУРА

1. Льюис Ф., Розенкранц Д., Стирнз Р. Теоретические основы проектирования компиляторов. – М. : Мир, 1979. – 656 с.
2. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. – М.: Мир, 1978. – т. 1, 612 с., – т. 2, 487 с.
3. Рязанов Ю. Д. Синтез распознавателей с магазинной памятью по детерминированным синтаксическим диаграммам // Вестник ВГУ. Системный анализ и информационные технологии. 2014. №1. с. 138 – 145.

Y.D. Ryazanov, I.N. Savelova

TRANSFORMING OF THE PUSHDOWN RECOGNIZER WITH ONE STATE INTO RECOGNIZER WITH FINITE SET OF STATES

Belgorod State Technological University after V.G. Shukhov

In this paper, we define the class of pushdown recognizers with one state which can be transformed to equivalent pushdown recognizers with finite set of states and the algorithm of transformation. Recognizer with a finite set of states performing fewer operations over the pushdown memory than equivalent recognizer with one state.

Keywords: context-free language, pushdown recognizer, state, equivalent transforming.