**Лабораторная работа №3**

**Недетерминированные магазинные автоматы.**

**Теоретическая часть**

Модель магазинного автомата (рис.1) состоит из

* входной ленты,
* устройства управления и
* вспомогательной ленты, называемой магазином или стеком.

**Рис.1.** Модель магазинного автомата

Входная лента разделяется на клетки (позиции), в каждой из которых может быть записан символ входного алфавита. При этом предполагается, что в неиспользуемых клетках входной ленты расположены пустые символы .

Вспомогательная лента также разделена на клетки, в которых могут располагаться символы магазинного алфавита. Начало вспомогательной ленты называется дноммагазина. Связь устройства управления с лентами осуществляется двумя головками, которые могут перемещаться вдоль лент.

Головка входной ленты может перемещаться только в одну сторону - вправо или оставаться на месте. Она может выполнять только чтение. Головка вспомогательной ленты способна выполнять как чтение, так и запись, но эти операции связаны с перемещением головки определенным образом:

* при записи головка предварительно сдвигается на одну позицию вверх, а затем символ заносится на ленту,
* при чтении символ, находящийся под головкой считывается с ленты, а затем головка сдвигается на одну позицию вниз, т.о. головка всегда установлена против последнего записанного символа. Позицию, находящуюся в рассматриваемый момент времени под головкой, называют вершиной магазина.

***Определение*.** Магазинный автомат **М** определяется следующей совокупностью семи объектов: **M={S, P, Z, , sо, hо, F},** где

**S** - алфавит состояний,

**P** - входной алфавит,

**Z** - алфавитмагазинных символов, записываемых на вспомогательную ленту,

**** - функция переходов, ** : {S** x **P** **** {} x **H**} **** **S** x **H\***, если М-автомат - детерминированный, и ** : {S** x **P **{} x **H**} **** 2**S x H\***, если М-автомат - недетерминированный.

**sо** - начальное состояние, **sо  S**.

**hо**- маркер дна, он всегда записывается на дно магазина , **hо  H**.

**F** - множество конечных состояний. **F** является подмножеством **S**.

Функция **** отображает тройки (pi , sj , hk) в пары (sr , ) , где  **** **H\*** и hk - символ в вершине магазина, для детерминированного автомата или в множество таких пар для недетерминированного автомата.

Эта функция описывает изменение состояния магазинного автомата, происходящее при чтении символа с входной ленты и перемещении входной головки.

В дальнейшем при построении магазинных автоматов потребуются две разновидности функций переходов:

1. функция переходов с пустым символом в качестве входного символа:

0(s, , h) = (s', ), которая, независимо от того, какой символ находится под читающей головкой входной ленты, предписывает прочитать символ h из вершины магазина, изменить состояние автомата на s' и записать цепочку  в магазин.

2) функция переходов с определенным входным символом:

 (s, a, h) = (s', ), которая предписывает прочитать входной символ а, изменить состояние автомата на s’ и заменить верхний символ магазина h цепочкой .

Работа магазинного автомата

Чтобы описать, как работает автомат, введем понятие конфигурации.

***Определение*.** Конфигурацией автомата **М** называют тройку (s, , )**S x P\* x H\***, где

s - текущее состояние управляющего устройства,

 - неиспользованная часть входной цепочки P\*, самый левый символ этой цепочки находится под головкой. Если a =  , то считается, что вся входная цепочка прочитана.

 - цепочка, записанная в магазине, H\*, самый правый символ цепочки считается вершиной магазина. Если =, то магазин пуст.

Работа автомата может быть представлена как смена конфигураций. Один такт работы автомата заключается в определении новой конфигурации по заданной. Это записывается так:

(s, a, h ) ├ (s', , )

Такая смена конфигураций возможна, если функция **(s, a, h )** (или **(s, , h)**) определена и имеет значение **(s', ).** При этом предполагается, что автомат

* читает символ a, находящийся под головкой. Или не читает ничего, в случае входного символа .
* определяет новое состояние s'
* читает символ h, находящийся в вершине магазина и
* записывает цепочку  в магазин вместо символа h. Если  = , то верхний символ оказывается удаленным из магазина.

Таким образом, могут быть три случая при работе автомата:

* **(s, a, h)** определена и выполняется такт работы,
* **(s, a, h)** не определена, но определена функция **(s, , h)** и выполняется пустой такт (без чтения входной информации).
* функции **(s, a, h)** и **(s, , h)** не определены, в этом случае дальнейшая работа автомата невозможна.

*Начальной конфигурацией* называется конфигурация (s0, , h0), где s0 - начальное состояние,  - исходная цепочка, h0 - маркер дна магазина.

*Заключительная конфигурация* – (s, , ), где s принадлежит множеству заключительных состояний F.

Для обозначения последовательности сменяющих друг друга конфигураций условимся использовать знак **├**\*. Таким образом последовательность

( s1, 1, 1 ) **├** ( s2, 2, 2) **├** ... **├** ( sn, n,n )

записывается в сокращенном виде как:

(s1, 1, 1 ) **├**\* ( sn, n, n ).

Язык, допускаемый магазинным автоматом

***Определение****.* Цепочка  называется **допустимой** для автомата М, если существует последовательность конфигураций, в которой первая конфигурация является начальной c цепочкой , а последняя – заключительной. (s**о**, , h**о**) **├**\* (s1,  , ) , где s1  F .

***Определение*.** Множество цепочек, допускаемых автоматом М, называется языком, **допускаемым** или определяемым автоматом М, и обозначается **L(М)**.

**L(М)= { | ( sо, , hо ) ├\* ( s', , ) и s'  F }**

Чтобы лучше представить себе работу магазинного автомата, рассмотрим два примера. Пусть задан магазинный автомат М1 в следующем виде:

М1: P = {a , b}; S = {s0 , s1 , s2}; Z = {h0 , a}; F = {s0};

 (s0 , a , h0) = (s1 , h0a),

 (s1 , a , a) = (s1 , aa),

 (s1 , b , a) = (s2 , ),

 (s2 , b , a) = (s2 , ),

 (s2 ,  , h0) = (s0 , ).

Этот автомат является **детерминированным**, поскольку каждому набору аргументов соответствует единственное значение функции. Работу автомата при распознавании входной цепочки aabb можно представить в виде последовательности конфигураций:

(s0,aabb,h0) **├** (s1,abb,h0a) **├** (s1,bb,h0aa) **├** (s2,b,h0a) **├** (s2,,h0) **├** (s0,,) .

Нетрудно проверить, что при задании входной цепочки aabbb автомат не сможет закончить работу. Следовательно, эта цепочка не принадлежит языку, допускаемому автоматом M1.

Магазинный автомат М2, заданный следующим описанием:

М2: P = {a , b}; S = {s0, s1 , s2}; Z = {h0, a , b}; F = {s2};

(1)  (s0 , a , h0) = (s0, h0a),

(2)  (s0 , b , h0) = (s0, h0b),

(3)  (s0 , a , a) = {(s0,aa) , (s1 , )},

(4)  (s0 , b , a) = (s0,ab),

(5)  (s0 , a , b) = (s0 , ba),

(6)  (s0 , b , b) = {(s0 , bb) , (s1 , )},

(7)  (s1 , a , a) = (s1, ),

(8)  (s1 , b , b) = (s1, ),

(9)  (s2 ,  , h0) = (s2 , ),

является **недетерминированным** автоматом, поскольку одному и тому же набору аргументов, например (sо , a, a), соответствуют два значения функции. Работу автомата рассмотрим для входной цепочки abba. Если использовать последовательность команд (1),(4),(6.1),(5), то получим последовательность конфигураций:

(s0,abba,h0)

**├** (s0,bba,h0a), (1)

**├** (s0,ba,h0ab), (4)

**├** (s0,a,h0abb), (6.1)

**├** (s0,,h0abba). (5)

которая показывает, что дальнейшая работа невозможна, т.к. входная цепочка прочитана и переход (s0,,h0abba) не определен. Если же использовать последовательность команд (1),(4),(6.2),(3),(9), то получим заключительную конфигурацию:

(s0,abba,h0)

**├** (s0,bba,h0a), (1)

**├** (s0,ba,h0ab), (4)

**├** (s1,a,h0a), (6.2)

**├** (s1,,h0), (3)

**├** (s2,,) . (9).

Т.о. можно сделать вывод о том, что цепочка abba допускается автоматом М2.

Построение магазинного автомата

Пусть задана грамматика G = {VN, VT, I, P}. Определим компоненты автомата М следующим образом:

**S = {s0}, P = VT, Z = VN  VT  {h0} , F = {s0},**

в качестве начального состояния автомата примем s0 и построим функцию переходов так:

1. Для всех **A  VN** таких, что встречаются в левой части правил **A ** , построим команды вида:

**0(s0, , A) = (s0, R ),**

где **R** – реверс цепочки **** .

1. Для всех **a** **VT** построим команды вида

** ( s0, a, a) = ( s0, )**

1. Для перехода в конечное состояние построим команду

** ( s0, , h0) = ( s0, )**

1. Начальную конфигурацию автомата определим в виде:

**( s0, , h0 I),**

где ** – **исходная цепочка, записанная на входной ленте.

Автомат, построенный по приведенным выше правилам, работает следующим образом. Если в вершине магазина находится терминал, и символ, читаемый с входной ленты, совпадает с ним, то по команде типа (2) терминал удаляется из магазина, а входная головка сдвигается. Если же в вершине магазина находится нетерминал, то выполняется команда типа (1), которая вместо терминала записывает в магазин цепочку, представляющую собой правую часть правила грамматики. Следовательно, автомат, последовательно заменяя нетерминалы, появляющиеся в вершине магазина, строит в магазине левый вывод входной цепочки, удаляя полученные терминальные символы, совпадающие с символами входной цепочки. Это означает, что каждая цепочка, которая может быть получена с помощью левого вывода в грамматике G, допускается построенным автоматом М.

# **Пример построения автомата**

Процедуру построения автомата рассмотрим на примере грамматики с правилами:

**E  E + T | T**

**T  T \* F | F**

**F  ( E ) | a**

Для искомого автомата имеем:

P = {a, +, \*, ), ( }, Z = {E, T, F, a, +, \*) , h0, ( }, S = {s0 }, F = {s0}

Для всех правил грамматики строим команды типа (1):

1. 0 (s0 ,  , E) = {(s0 , T+E) ; (s0 , T)},
2. 0 (s0 ,  , T) = {(s0 , F\*T) ; (s0 , F)},
3. 0 (s0 ,  , F) = {(s0 , (E) ) ; (s0 , a)},

Для всех терминальных символов строим команды типа (2):

1.  ( s0, a, a ) = ( s0,  ),
2.  ( s0 , + , + ) = (s0 ,  ),
3.  ( s0 , \* , \* ) = (s0 ,  ),
4.  ( s0 , ( , ( ) = (s0 ,  ),
5.  ( s0 , ) , ) ) = (s0 ,  ),

Для перехода в конечное состояние построим команду:

(9)  (s0 ,  , h0) = ( s0 ,  ).

Построенный автомат является недетерминированным.

Начальную конфигурацию с цепочкой a + a\*a запишем так: (s0 , a+a\*a , h0E).

Последовательность тактов работы построенного автомата, показывающая, что заданная цепочка допустима, имеет вид:

(s0 , a+a\*a , h0E) ├ (s0 , a+a\*a , h0T+E) ├ (s0 , a+a\*a , h0T+T) ├ (s0 , a+a\*a , h0T+F) ├

(s0 , a+a\*a , h0T+a) ├ (s0 , +a\*a , h0T+) ├ (s0 , a\*a , h0T) ├ (s0 , a\*a , h0F\*T) ├

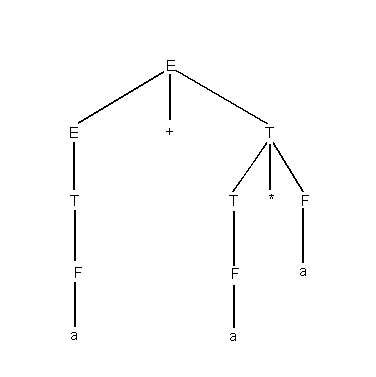
(s0 , a\*a , h0F\*F) ├ (s0 , a\*a , h0F\*a) ├ (s0 , \*a , h0F\* ) ├ (s0 , a , h0F) ├

(s0 , a , h0a) ├ (s0 ,  , h0) ├ (s0 ,  , ).

Отметим, что последовательность правил, используемая построенным автоматом, соответствует левому выводу входной цепочки:

E **** E+T **** T+T **** F+T **** a+T **** a+T\*F **** a+F\*F **** a+a\*F ** **a+a\*a.

Если по такому выводу строить дерево, то построение будет происходить сверху вниз, т.е. от корня дерева к листьям.

Такой способ построения дерева по заданной цепочке называется **нисходящим**.

Магазинные автоматы называют часто распознавателями, поскольку они определяют, является ли цепочка, подаваемая на вход автомата, допустимой или нет, и следовательно, отвечают на вопрос, принадлежит ли эта цепочка языку, пораждаемому грамматикой, использованной для построения автомата.

Учитывая характер построения вывода в магазине, автоматы рассмотренного типа называют **нисходящими распознавателями**.

**Задание на лабораторную работу**

Написать программу, реализующую работу недетерминированного магазинного автомата.

Входные данные:

1. Текстовый файл с описанием грамматики, для которой строится магазинный автомат.

Каждая строка в файле может задавать несколько правил грамматики с одинаковой левой частью. В этом случае правила отделяются друг от друга символом ‘|’.

Для отделения левой части правила от правой используется символ ‘>’. В одной строке может быть несколько символов ‘>’. В этом случае первый из них трактуется, как символ, отделяющий правую и левую части продукции, все последующие – как терминальный символ.

Все нетерминалы задаются с помощью прописных букв латинского алфавита.

Все символы, не описанные выше, являются терминальными символами.

Правил в грамматике (а значит и во входном файле) – неограниченное количество.

**Рекомендуется** считать пробельные символы в файле незначащими, а для терминала «пробел» использовать какой-нибудь другой символ (например, ~,либо заключать пробел в апострофы).

Пример входного файла

E > E + T | T

T > T \* F | F

F > ( E ) | a | ~

1. Строка символов, которую нужно проанализировать с помощью построенного автомата и дать заключение о допустимости (или недопустимости) автоматом данной цепочки символов.

Выходные данные

1. На оценку «удовлетворительно»: заключение о допустимости (или недопустимости) автоматом цепочки символов.
2. На оценку «отлично»: значение множеств P, Z и т.д.; список команд (1) – (4) (см раздел «Построение магазинного автомата»); цепочка конфигураций магазинного автомата, полученная в процессе его работы; заключение о допустимости (или недопустимости) автоматом цепочки символов.

**Литература**

При подготовка данной лабораторной работы были использованы материалы сайта <http://mf.grsu.by/>.

Copyright © 2005 – 2010 Voldem@r