

# Лекция 15

10 декабря

## 7.6. Распознавание методом “перенос-свёртка”

- Автомат из предыдущего подраздела исследует несколько верхних символов, чтобы решить, какую из операций свёртки нужно выполнить.
- Далее рассматривается подход, который использует расширенный магазинный алфавит для кодирования дополнительной информации о магазине.
- Закодированной информации достаточно, чтобы по верхнему символу магазина и текущему входному символу выяснить, находится ли основа в верхней части магазина, и, если да, то определить основывающее правило, не просматривая других символов магазина.
- Однако, для каждого правила соответствующая ему операция Свёртка просматривает символ магазина, расположенный под основой.

## Кодирование магазинных символов

- Возьмём ту же грамматику, что и в начале этого раздела:
- 
- 1.  $\langle S \rangle \rightarrow ( \langle A \rangle \langle S \rangle )$
- 2.  $\langle S \rangle \rightarrow ( b )$
- 3.  $\langle A \rangle \rightarrow ( \langle S \rangle a \langle A \rangle )$
- 4.  $\langle A \rangle \rightarrow ( a )$
- 
- Будем различать одинаковые символы грамматики, входящие в правые части разных правил.
- При этом новый символ будет одновременно кодировать последовательность символов от начала правой части до самого этого символа.
- Схема кодирования приведена в следующей таблице:

# Схема кодирования

Представляемый символ	Магазинный символ	Кодируемая цепочка
a	$a_1$	( $<S>$ a
	$a_2$	( a
b	$b_1$	( b
(	$(_1$	(
)	$)_1$	( $<A>$ $<S>$ )
	$)_2$	( b )
	$)_3$	( $<S>$ a $<A>$ )
	$)_4$	( a )
$<S>$	$<S>_1$	( $<A>$ $<S>$
	$<S>_2$	( $<S>$
	$<S>_3$	$\nabla <S>$
$<A>$	$<A>_1$	( $<A>$
	$<A>_2$	( $<S>$ a $<A>$
Нет	$\nabla$	$\nabla$

- Каждый магазинный символ понимается одновременно как грамматический символ и как код цепочки.
- Каждый символ магазина представляет грамматический символ (кроме маркера дна), получаемый из него отбрасыванием индекса.
- Наряду с этим каждый магазинный символ рассматривается как код некоторой цепочки.
- Например,  $\langle S \rangle_1$  рассматривается как код цепочки ( $\langle A \rangle \langle S \rangle$ ).
- Самым правым символом каждой кодируемой цепочки является грамматический символ, представляемый магазинным символом.
- Исключение составляет маркер дна.
- Он не представляет никакого грамматического символа, но рассматривается как код цепочки, состоящей из него самого.
- Новый МП-автомат будет построен так, что магазинный символ вталкивается в магазин только тогда, когда кодируемая этим символом цепочка совместима с цепочкой, которую буде представлять магазин после вталкивания.



- Магазинный символ  $\langle A \rangle_1$  оказался бы несовместимым, потому что после вталкивания магазин будет содержать
- 
- $\nabla(1(1\langle S \rangle_2 a_1 \langle A \rangle_1$
- 
- и цепочка  $\langle A \rangle$ , закодированная верхним символом магазина, не является суффиксом представляемой магазином цепочки
- $((\langle S \rangle a \langle A \rangle$
- 
- и не является конкатенацией  $\nabla$  и представляемой цепочки.

- Магазинный алфавит построен так, что для каждого магазинного символа, за исключением  $\langle S \rangle_3$  и  $\nabla$ , кодируемая цепочка является префиксом правой части некоторого правила.
- И, наоборот, каждый непустой префикс правой части кодируется магазинным символом.
- Например, правая часть правила 1 – это цепочка  $(\langle A \rangle \langle S \rangle)$ , которая имеет четыре префикса:
- $($ ,  $(\langle A \rangle$ ,  $(\langle A \rangle \langle S \rangle$ ,  $(\langle A \rangle \langle S \rangle)$
- 
- Эти четыре префикса кодируются четырьмя магазинными символами:
- 
- $(_1$ ,  $\langle A \rangle_1$ ,  $\langle S \rangle_1$ ,  $)_1$
- 
- Аналогично и для других правил.

- Управление МП-автоматом осуществляется при помощи двух таблиц.
- Во-первых, обычная управляющая таблица (будет рассмотрена позже) используется для выбора между переносом и свёрткой.
- Если принятное решение включает вталкивание, то оно определяет только, какой грамматический символ должен быть представлен новым символом магазина.
- Для определения того, какой из магазинных символов, представляющих данный грамматический символ, нужно втолкнуть, используется таблица вталкиваний.
- Для нашего примера она выглядит следующим образом:

# Таблица вталкиваний

Магазинные символы	<b>a</b>	<b>b</b>	<b>(</b>	<b>)</b>	<b>&lt;S&gt;</b>	<b>&lt;A&gt;</b>
$a_1$			$(_1$			$<A>_2$
$a_2$			$(_1$	$)_4$		
$b_1$			$(_1$	$)_2$		
$(_1$	$a_2$	$b_1$	$(_1$		$<S>_2$	$<A>_1$
$<S>_1$			$(_1$	$)_1$		
$<S>_2$	$a_1$		$(_1$			
$<S>_3$			$(_1$			
$<A>_1$			$(_1$		$<S>_1$	
$<A>_2$			$(_1$	$)_3$		
$\nabla$			$(_1$		$<S>_3$	

- Выбор делается исходя из символа, находящегося на верху магазина перед вталкиванием, и грамматического символа, представляемого новым символом магазина.
- Таблица содержит строку для каждого магазинного символа, поверх которого можно втолкнуть новый магазинный символ, и столбец для каждого символа грамматики.
- Каждый непустой элемент таблицы указывает символ, который нужно втолкнуть для данной комбинации символа магазина и символа грамматики.
- Пустые элементы таблицы относятся к комбинациям, при которых автомат отвергает входную цепочку.
- Таблица не содержит строк для магазинных символов  $J_1, J_2, J_3$  и  $J_4$ , потому что в данной схеме управления никогда не делается попытка втолкнуть символ, если верхушка магазина представляет правую скобку.

- Если конкатенация грамматического символа и префикса, закодированного магазинным символом, даёт новый префикс, то вталкивается тот магазинный символ, который кодирует этот новый префикс.
- В том случае, когда грамматический символ – левая скобка, предполагается, что она начинает новую правую часть, и автомат вталкивает в магазин символ  $(_1$ , который кодирует односимвольную цепочку  $($ .
- Если грамматический символ нельзя использовать для продолжения префикса, закодированного верхним символом магазина, или в качестве начала нового префикса, то автомат отвергает вход на том основании, что оставшиеся символы не могут завершить правую часть правила и, следовательно, не могут завершить допустимую цепочку.
- Исключение составляет случай магазинного символа  $\nabla$  и грамматического символа  $\langle S \rangle$ , потому что из единственного символа  $\langle S \rangle$  можно породить допустимую цепочку (и даже все допустимые цепочки).
- В этом случае таблица вталкиваний даёт символ  $\langle S \rangle_3$ , который кодирует цепочку  $\nabla \langle S \rangle$ .

- Рассмотрим теперь управляющую таблицу МП-автомата.
- По этой таблице для каждой комбинации входного символа и верхнего символа магазина выбирается переходная процедура (здесь опущено слово ‘Держать’ для процедур, которые не сдвигают вход).

# Управляющая таблица МП-автомата

Магазинные символы	(	а	б	)	⊥
$a_1$	Перенос	Перенос	Перенос	Перенос	Отвергн
$a_2$	Перенос	Перенос	Перенос	Перенос	Отвергн
$b_1$	Перенос	Перенос	Перенос	Перенос	Отвергн
$(_1$	Перенос	Перенос	Перенос	Перенос	Отвергн
$)_1$	Свёртка(1)	Свёртка(1)	Свёртка(1)	Свёртка(1)	Свёртка(1)
$)_2$	Свёртка(2)	Свёртка(2)	Свёртка(2)	Свёртка(2)	Свёртка(2)
$)_3$	Свёртка(3)	Свёртка(3)	Свёртка(3)	Свёртка(3)	Свёртка(3)
$)_4$	Свёртка(4)	Свёртка(4)	Свёртка(4)	Свёртка(4)	Свёртка(4)
$<S>_1$	Перенос	Перенос	Перенос	Перенос	Отвергн
$<S>_2$	Перенос	Перенос	Перенос	Перенос	Отвергн
$<S>_3$	Перенос	Перенос	Перенос	Перенос	Допустить
$<A>_1$	Перенос	Перенос	Перенос	Перенос	Отвергн
$<A>_2$	Перенос	Перенос	Перенос	Перенос	Отвергн
▽	Перенос	Перенос	Перенос	Перенос	Отвергн
Начальное содержимое магазина ▽					

## Элементы управляющей таблицы

- Перенос: ТАВТОЛК (текущий входной символ), Сдвиг
- Свёртка(1): Вытолк, Вытолк, Вытолк, Вытолк, ТАВТОЛК(<S>)
- Свёртка(2): Вытолк, Вытолк, Вытолк, ТАВТОЛК(<S>)
- Свёртка(3): Вытолк, Вытолк, Вытолк, Вытолк, Вытолк, ТАВТОЛК(<A>)
- Свёртка(4): Вытолк, Вытолк, Вытолк, ТАВТОЛК(<A>)
- 
- Операция ТАВТОЛК использует таблицу вталкиваний для определения того, какой символ следует втолкнуть в магазин.
- Рассмотрим обработку цепочки
- 
- (( ( b ) a ( a ) )( b ))

# Обработка цепочки

Содержимое магазина	Текущ входн. цепочка	Операция
$\nabla$	$(( ( b ) a ( a ) ) ( b ) ) \perp$	Перенос
$\nabla (_1$	$( ( b ) a ( a ) ) ( b ) ) \perp$	Перенос
$\nabla ( _1 ( _1$	$( b ) a ( a ) ) ( b ) ) \perp$	Перенос
$\nabla ( _1 ( _1 ( _1$	$b ) a ( a ) ) ( b ) ) \perp$	Перенос
$\nabla ( _1 ( _1 ( _1 b_1$	$) a ( a ) ) ( b ) ) \perp$	Перенос
$\nabla ( _1 ( _1 ( _1 b_1 )_2$	$a ( a ) ) ( b ) ) \perp$	Свёртка(2)
$\nabla ( _1 ( _1 <S>_2$	$a ( a ) ) ( b ) ) \perp$	Перенос
$\nabla ( _1 ( _1 <S>_2 a_1$	$( a ) ) ( b ) ) \perp$	Перенос
$\nabla ( _1 ( _1 <S>_2 a_1 ( _1$	$a ) ) ( b ) ) \perp$	Перенос
$\nabla ( _1 ( _1 <S>_2 a_1 ( _1 a_2$	$) ) ( b ) ) \perp$	Перенос

## Обработка цепочки (продолжение)

Содержимое магазина	Текущ входн. цепочка	Операция
$\nabla (1 (1 <S>_2 a_1 (1 a_2)_4$	) ( b ) ) \perp	Свёртка(4)
$\nabla (1 (1 <S>_2 a_1 <A>_2$	) ( b ) ) \perp	Перенос
$\nabla (1 (1 <S>_2 a_1 <A>_2)_3$	( b ) ) \perp	Свёртка(3)
$\nabla (1 <A>_1$	( b ) ) \perp	Перенос
$\nabla (1 <A>_1 (1$	b ) ) \perp	Перенос
$\nabla (1 <A>_1 (1 b_1$	) ) \perp	Перенос
$\nabla (1 <A>_1 (1 b_1)_2$	) \perp	Свёртка(2)
$\nabla (1 <A>_1 <S>_1$	) \perp	Перенос
$\nabla (1 <A>_1 <S>_1)_1$	\perp	Свёртка(1)
$\nabla <S>_3$	\perp	Допустить

## 7.7. Построение автомата типа “перенос-свёртка”

- В предыдущем параграфе мы обсудили основную идею метода ‘перенос-свёртка’.
- Теперь рассмотрим, как построить МП-автомат, реализующий этот метод.
- Рассмотрим задачу восходящего грамматического разбора для следующей грамматики:
- 
- 1.  $\langle S \rangle \rightarrow a \langle A \rangle b$
- 2.  $\langle S \rangle \rightarrow c$
- 3.  $\langle A \rangle \rightarrow b \langle S \rangle$
- 4.  $\langle A \rangle \rightarrow \langle B \rangle b$
- 5.  $\langle B \rangle \rightarrow a \langle A \rangle$
- 6.  $\langle B \rangle \rightarrow c$
- 
- Эта грамматика не является простой ССП-грамматикой, т.к. имеется конфликт переноса-опознания. Проверим это.

## Множества *Перв* и *След*

- Построим сначала множества *Перв* и *След*:
- 
- $\text{Перв}(<S>) = \{\langle S \rangle, a, c\}$
- $\text{Перв}(<A>) = \{\langle A \rangle, \langle B \rangle, a, b, c\}$
- $\text{Перв}(<B>) = \{\langle B \rangle, a, c\}$
- 
- $\text{След}(<S>) = \{b, \perp\}$
- $\text{След}(<A>) = \{b\}$
- $\text{След}(<B>) = \{b\}$
- $\text{След}(a) = \{a, b, c\}$
- $\text{След}(b) = \{b, \perp\}$
- $\text{След}(c) = \{b, \perp\}$
- 
- Далее построим отношения Свёртывается-По и Под.

## Отношение Свёртывается-По

Магазинные символы	a	b	c	⊥
<S>		1		1
<A>		1		
<B>				
a				
b		1		1
c		1		1

## Отношение Под

Магазинные символы	$\langle S \rangle$	$\langle A \rangle$	$\langle B \rangle$	a	b	c
$\langle S \rangle$						
$\langle A \rangle$					1	
$\langle B \rangle$					1	
a		1	1	1	1	1
b	1			1		1
c						
$\nabla$	1			1		1

- Из этих отношений видно, что имеет место конфликт переноса-опознания для верхнего магазинного символа **<A>** и текущего входного символа **b**, т.к. для этой комбинации символов одновременно выполняются отношения Свёртывается-По и Под.
- Применим к этой грамматике подход, основанный на методе “перенос-свёртка”.