

Лекция 13

26 ноября

Управление автоматом типа “перенос-опознание” (продолжение)

- Грамматика примера:

-

1. $\langle S \rangle \rightarrow b \langle A \rangle \langle S \rangle \langle B \rangle$

2. $\langle S \rangle \rightarrow b \langle A \rangle$

3. $\langle A \rangle \rightarrow d \langle S \rangle c a$

4. $\langle A \rangle \rightarrow e$

5. $\langle B \rangle \rightarrow c \langle A \rangle a$

6. $\langle B \rangle \rightarrow c$

Пример (продолжение)

- Множество *Перв* для символов данной грамматики имеет следующий вид:
-
- $\text{Перв}(\langle S \rangle) = \{ b, \langle S \rangle \}$
- $\text{Перв}(\langle A \rangle) = \{ d, e, \langle A \rangle \}$
- $\text{Перв}(\langle B \rangle) = \{ c, \langle B \rangle \}$
- $\text{Перв}(a) = \{ a \}$
- $\text{Перв}(b) = \{ b \}$
- $\text{Перв}(c) = \{ c \}$
- $\text{Перв}(d) = \{ d \}$
- $\text{Перв}(e) = \{ e \}$

Пример (продолжение)

- Множества *След*:
-
- $\text{След}(\langle S \rangle) = \{ c, \perp \}$
- $\text{След}(\langle A \rangle) = \{ a, b, c, \perp \}$
- $\text{След}(\langle B \rangle) = \{ c, \perp \}$
- $\text{След}(a) = \{ a, b, c, \perp \}$
- $\text{След}(b) = \{ d, e \}$
- $\text{След}(c) = \{ a, c, d, e, \perp \}$
- $\text{След}(d) = \{ b \}$
- $\text{След}(e) = \{ a, b, c, \perp \}$

Отношение Свёртывается-По

- Если A – символ грамматики, а t – входной символ, то
-
- A Свёртывается-По t
-
- тогда и только тогда, когда выполняется одно из следующих условий:
- 1. Существует нетерминал L и цепочка x , такие что
- $$L \rightarrow xA$$
- есть правило грамматики, и t принадлежит множеству $\text{След}(L)$.
-
- 2. A – начальный символ грамматики, t – концевой маркер.

Отношение Под

- Если **A** – магазинный символ, а **t** – входной символ, то
-
- $A \text{ Под } t$
-
- тогда и только тогда, когда выполняется одно из следующих условий:
- 1. Имеется символ грамматики **B**, такой что в правую часть некоторого правила входит цепочка **AB**, и **t** принадлежит множеству *Перв(B)*.
- 2. **A** – маркер дна магазина, а **t** принадлежит множеству *Перв(S)*, где **S** – начальный символ грамматики.

Построение управляющей таблицы автомата

- Мы говорим, что грамматика содержит конфликт переноса-опознания тогда и только тогда, когда существуют символы грамматики **A** и **t** такие, что одновременно
 - **A** Свёртывается-По **t** и **A** Под **t**
 - Для грамматики без конфликтов переноса-опознания управляющая таблица автомата строится следующим образом:
 1. Элементы для магазинного символа **A** и входного символа **t** таких, что **A** Свёртывается-По **t**, должны содержать процедуры опознания.
 2. Элементы для магазинного символа **A** и входного символа **t** таких, что **A** Под **t**, должны содержать операцию *Перенос*.
 3. Остальные элементы должны содержать операцию *Отвергнуть*.

Отношение Свёртывается-По

	a	b	c	d	e	\perp
<S>						1
<A>			1			1
			1			1
a	1	1	1			1
b						
c			1			1
d						
e	1	1	1			1

Отношение Под

	a	b	c	d	e
<S>			1		
<A>	1	1			
					
a					
b				1	1
c	1			1	1
d		1			
e					
∇		1			

Управляющая таблица МП-автомата для примера

Магазинные символы	a	b	c	d	e	\perp
<S>	Отвергн	Отвергн	Перенос	Отвергн	Отвергн	Опозн
<A>	Перенос	Перенос	Опозн	Отвергн	Отвергн	Опозн
	Отвергн	Отвергн	Опозн	Отвергн	Отвергн	Опозн
a	Опозн	Опозн	Опозн	Отвергн	Отвергн	Опозн
b	Отвергн	Отвергн	Отвергн	Перенос	Перенос	Отвергн
c	Перенос	Отвергн	Опозн	Перенос	Перенос	Опозн
d	Отвергн	Перенос	Отвергн	Отвергн	Отвергн	Отвергн
e	Опозн	Опозн	Опозн	Отвергн	Отвергн	Опозн
∇	Отвергн	Перенос	Отвергн	Отвергн	Отвергн	Отвергн
Начальное содержимое магазина ∇						

7.3.Бессуффиксные ПО-грамматики

- Рассмотрим теперь вторую часть задачи построения автомата – выбор подходящих процедур опознания.
- Введём сначала некоторые определения.
- Цепочка **x** называется суффиксом цепочки **y**, если **y** оканчивается цепочкой **x**. Заметим, что всякая цепочка является своим собственным суффиксом, и цепочка **e** является суффиксом любой цепочки.
- Грамматика называется бессуффиксной, если правая часть её любого правила не является суффиксом правой части какого-либо другого правила или суффиксом цепочки $\nabla \langle S \rangle$.
- Бессуффиксная грамматика, не имеющая конфликтов переноса-опознания, называется бессуффиксной ПО-грамматикой.

Процедура опознания

- Вернёмся к грамматике подраздела 7.2.
- Правые части правил этой грамматики не являются суффиксами каких-либо других правых частей или цепочки $\nabla \langle S \rangle$.
- Следовательно, эта грамматика является бессуффиксной ПО-грамматикой.
- Особенностью таких грамматик является то, что если на верху магазина имеется основа, то она только одна.
- В этом случае процедура опознания сравнивает верхнюю часть магазина с правыми частями правил, и при их совпадении выполняет соответствующую свёртку.

Процедура опознания

- Опознать: Если на верху магазина $b \langle A \rangle \langle S \rangle \langle B \rangle$, то *Свёртка(1)*,
- иначе
- если на верху магазина $b \langle A \rangle$, то *Свёртка(2)*,
- иначе
- если на верху магазина $d \langle S \rangle c a$, то *Свёртка(3)*,
- иначе
- если на верху магазина e , то *Свёртка(4)*,
- иначе
- если на верху магазина $c \langle A \rangle a$, то *Свёртка(5)*,
- иначе
- если на верху магазина c , то *Свёртка(6)*,
- иначе
- если на верху магазина $\nabla \langle S \rangle$ и тек. вх симв. \perp , то
- *Допустить*,
- иначе
- *Отвергнуть*.

Модифицированная управляющая таблица МП-автомата

Магазинные символы	a	b	c	d	e	⊥
<S>	Отвергн	Отвергн	Перенос	Отвергн	Отвергн	Опозн1
<A>	Перенос	Перенос	Опозн2	Отвергн	Отвергн	Опозн2
	Отвергн	Отвергн	Опозн3	Отвергн	Отвергн	Опозн3
a	Опозн4	Опозн4	Опозн4	Отвергн	Отвергн	Опозн4
b	Отвергн	Отвергн	Отвергн	Перенос	Перенос	Отвергн
c	Перенос	Отвергн	Опозн5	Перенос	Перенос	Опозн5
d	Отвергн	Перенос	Отвергн	Отвергн	Отвергн	Отвергн
e	Опозн6	Опозн6	Опозн6	Отвергн	Отвергн	Опозн6
∇	Отвергн	Перенос	Отвергн	Отвергн	Отвергн	Отвергн
Начальное содержимое магазина ∇						

Процедуры опознания

- Опознать1: Если на верху магазина $\nabla \langle S \rangle$, то *Допустить*, иначе
- *Отвергнуть*.
- Опознать2: Если на верху магазина $b \langle A \rangle$, то *Свёртка(2)*, иначе
- *Отвергнуть*.
- Опознать3: Если на верху магазина $b \langle A \rangle \langle S \rangle \langle B \rangle$, то *Свёртка(1)*,
иначе *Отвергнуть*.
- Опознать4: Если на верху магазина $d \langle S \rangle c a$, то *Свёртка(3)*, иначе,
если на верху магазина $c \langle A \rangle a$, то *Свёртка(5)*,
- иначе *Отвергнуть*.
- Опознать5: *Свёртка(6)*.
- Опознать6: *Свёртка(4)*.

7.4. Грамматики слабого предшествования

- Рассмотрим следующий пример. Пусть дана грамматика:
- 1. $\langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle$
- 2. $\langle E \rangle \rightarrow \langle T \rangle$
- 3. $\langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle$
- 4. $\langle T \rangle \rightarrow \langle P \rangle$
- 5. $\langle P \rangle \rightarrow (\langle E \rangle)$
- 6. $\langle P \rangle \rightarrow I$
- Начальный символ $\langle E \rangle$.
-

Грамматики слабого предшествования (продолжение)

- Допустим, что на верху магазина находятся символы $\langle E \rangle + \langle T \rangle$.
- Предположим, что мы при этом выполняем *Свёртку(2)*, тогда на верху магазина будет цепочка $\langle E \rangle + \langle E \rangle$.
- Выясним, может ли символ $\langle E \rangle$ следовать за символом '+' в какой-либо допустимой цепочке.
- Мы имеем только одну правую часть с символом '+', а именно $\langle E \rangle + \langle T \rangle$, таким образом, за символом '+' может следовать только символ, принадлежащий множеству $\text{Перв}(\langle T \rangle)$. Множество $\text{Перв}(\langle T \rangle)$ имеет следующий вид:
 -
 - $$\text{Перв}(\langle T \rangle) = \{ \langle T \rangle, \langle P \rangle, I, '(' \}$$
 -
- Символ $\langle E \rangle$ не принадлежит этому множеству и, следовательно, не может следовать за символом + в какой-либо допустимой цепочке.
- Таким образом, правило 2 в данном случае не является основывающим, и необходимо выполнять *Свёртку(1)*.

Расширение отношения Под

- Для более строгого описания проиллюстрированного подхода расширим отношение Под следующим образом:
- Если **A** и **X** – символы некоторой грамматики, то
-
- **A Под X**
-
- тогда и только тогда, когда выполняется одно из следующих условий:
- 1. Существует грамматический символ **B**, такой что цепочка **AB** содержится в правой части некоторого правила, и **X** принадлежит множеству *Перв(B)*.
- 2. **A** – это маркер дна, а **X** принадлежит множеству *Перв(S)*, где *S* – начальный символ грамматики.
- Это определение отличается от определения в 7.2 лишь тем, что теперь допускается появление нетерминальных символов справа от Под.

Принцип переноса

- *Принцип переноса* из 7.2 теперь можно обобщить до следующего *принципа вталкивания*:
- Если для данной грамматики имеется распознаватель типа *перенос-опознание*, и заданы два магазинных символа **A** и **X**, то допустимая цепочка, при обработке которой символ **X** вталкивается непосредственно над **A**, существует тогда и только тогда, когда выполняется отношение
- $A \text{ Под } X$.
- Дадим теперь определение грамматики слабого предшествования.

Грамматика слабого предшествования

- Контекстно-свободная грамматика называется *грамматикой слабого предшествования*, если выполняются следующие четыре условия:
 - 1. В грамматике нет конфликтов переноса-опознания.
 - 2. Правые части любых двух правил не совпадают.
 - 3. Для любых двух правил вида
 - $\langle A \rangle \rightarrow x P y$
 - $\langle B \rangle \rightarrow y$где x и y – цепочки, а P – символ, отношение $P \text{ Под } \langle B \rangle$ не имеет места.
 - 4. Неверно, что $\langle S \rangle \Rightarrow^+ \langle S \rangle$, где $\langle S \rangle$ – начальный символ грамматики.

Множества *Перв* и *След*

- По данной грамматике слабого предшествования можно построить автомат типа “перенос-опознание” примерно так же, как и по бессуффиксной ПО-грамматике.
- Отличие состоит в том, что процедуры опознания при выборе операции свёртки должны анализировать символ, находящийся под верхним магазинным символом, и проверять отношение Под.
- Вернёмся к нашему примеру. Множества *Перв* и *След* имеют следующий вид:
 - $Перв(<E>) = \{<E>, <T>, <P>, '(', | \}$
 - $Перв(<T>) = \{<T>, <P>, '(', | \}$
 - $Перв(<P>) = \{<P>, '(', | \}$
 - $След(<E>) = \{ +, ')', \# \}$
 - $След(<T>) = \{ +, *, ')', \# \}$
 - $След(<P>) = \{ +, *, ')', \# \}$
- Отношение Под для данной грамматики приведено в следующей таблице.

Отношение Под для данной грамматики

Y									
X	Маг. СИМВ.	+	*	<i>I</i>	()	$\langle E \rangle$	$\langle T \rangle$	$\langle P \rangle$
	+			1	1			1	1
	*			1	1				1
	<i>I</i>								
	(1	1		1	1	1
)								
	$\langle E \rangle$	1				1			
	$\langle T \rangle$		1						
	$\langle P \rangle$								
	∇			1	1		1	1	1
X Под Y									

Грамматика слабого предшествования

- Предполагается, что пустые элементы заполнены нулями.
- Как видно из таблицы, отношение $+ \underline{Под} <E>$ не выполняется, поэтому конфликт суффиксов между правилами 1 и 2 разрешается в пользу более длинного правила (правила 1).
- Также не имеет места отношение $* \underline{Под} <T>$, и, значит, конфликт суффиксов между правилами 3 и 4 разрешается в пользу правила 3.
- Так как других конфликтов суффиксов, а также конфликтов переноса-опознания в грамматике нет, то она является грамматикой слабого предшествования.
- Построим теперь автомат типа “перенос-опознание” для данной грамматики.