

Глава 4. Восходящий синтаксический анализ

4.6. *LR*-таблицы разбора

4.6.2. *SLR*(1)-грамматики

Более мощный подкласс *LR*(1)-грамматик – так называемые *SLR*(1)-грамматики – простые (Simple) *LR*(1)-грамматики. Во многом *SLR*(1)-метод построения *SLR*(1)-таблицы разбора совпадает с *LR*(0)-методом. Как и в *LR*(0)-методе, пункты не содержат предпросмотра, т. е. используются *LR*(0)-пункты. Отличие заключается в том, что *SLR*(1)-таблица разбора может содержать конфликты определенного вида, которые могут быть разрешены путем просмотра очередного входного символа.

$SLR(1)$ -метод требует вычисления функции

$$Follow(X) = \{a \mid S \xRightarrow{*} \alpha X a \beta\}, a \in V_T, \alpha \in V_T^*, \beta \in (V_T \cup V_N)^*,$$

для каждого нетерминала X грамматики. Эта функция определяет множество терминалов, которые могут следовать непосредственно за нетерминалом X в какой-либо сентенциальной форме, выводимой из начального нетерминала S . Формальный процесс вычисления функции $Follow$ подробно изложен при рассмотрении $LL(1)$ -грамматик. Значения данной функции используются только для реализации свертки. Если имеется пункт $[A \rightarrow \alpha \bullet]$, свертка должна выполняться для всех терминалов $a \in Follow(A)$, т. е. в $SLR(1)$ -таблицу разбора элементы свертки вносятся только в столбцы, соответствующие возможным символам-следователям.

Рассмотрим процесс построения $SLR(1)$ -таблицы разбора для грамматики с productions

1) $S \rightarrow AB$

2) $A \rightarrow aA$

3) $A \rightarrow a$

4) $B \rightarrow bB$

5) $B \rightarrow \varepsilon$

Функция $Follow$ для нетерминалов грамматики имеет следующие значения: $Follow(S) = \{\perp\}$, $Follow(A) = \{b, \perp\}$, $Follow(B) = \{\perp\}$.

Таблица 4.8

SLR(1)-автомат

Состояние	Пункты	Символ перехода	Состояние-преемник	Свертка
I_0	$S' \rightarrow \bullet S \perp$	S	$stop$	
	$S \rightarrow \bullet AB$	A	I_1	
	$A \rightarrow \bullet aA$	a	I_5	
	$A \rightarrow \bullet a$			
I_1	$S \rightarrow A \bullet B$	B	I_2	
	$B \rightarrow \bullet bB$	b	I_3	
	$B \rightarrow \bullet$	\perp		$R5$
I_2	$S \rightarrow AB \bullet$	\perp		$R1$
I_3	$B \rightarrow b \bullet B$	B	I_4	
	$B \rightarrow \bullet bB$	b	I_3	
	$B \rightarrow \bullet$	\perp		$R5$
I_4	$B \rightarrow bB \bullet$	\perp		$R4$
I_5	$A \rightarrow a \bullet A$	A	I_6	
	$A \rightarrow a \bullet$	b, \perp		$R3$
	$A \rightarrow \bullet aA$	a	I_5	
	$A \rightarrow \bullet a$			
I_6	$A \rightarrow aA \bullet$	b, \perp		$R2$

SLR(1)-автомат представлен в табл. 4.8. Значения функции *Follow* включены в столбец «символ перехода» для строк, соответствующих свертке. Это удобно для визуальной проверки состояний на наличие конфликтов.

1) $S \rightarrow AB$

2) $A \rightarrow aA$

3) $A \rightarrow a$

4) $B \rightarrow bB$

5) $B \rightarrow \varepsilon$

Функция *Follow* для нетерминалов грамматики:

$Follow(S) = \{\perp\},$

$Follow(A) = \{b, \perp\},$

$Follow(B) = \{\perp\}.$

Следует обратить внимание на то, что для продукции $S' \rightarrow S\perp$ построены не все пункты, а только $[S' \rightarrow \bullet S\perp]$ в состоянии I_0 и $[S' \rightarrow S\bullet\perp]$ в состоянии *stop* (в таблице соответствующая строка не показана). Это связано с тем, что в грамматике не существует более ни одной продукции, в правой части которой содержался бы нетерминал S . Поэтому пункт $[S' \rightarrow S\bullet\perp]$ однозначно устанавливает успешное завершение разбора, и нет смысла строить дополнительное состояние для пункта $[S' \rightarrow S\perp\bullet]$.

Из $SLR(1)$ -автомата видно, что данная грамматика не является $LR(0)$ -грамматикой, поскольку состояния I_1 , I_3 и I_5 имеют конфликты типа «перенос/свертка». Такие состояния называются *неадекватными*.

В состояниях I_1 и I_3 необходимо выполнить два перехода по символам B и b и одну свертку для продукции $B \rightarrow \varepsilon$. Чтобы разрешить этот конфликт, очередной входной символ сравнивается с возможным символом-следователем для нетерминала B , т. е. для нетерминала из левой части продукции, для которой выполняется свертка. Поскольку $Follow(B) = \{\perp\}$, то, если входным символом является символ \perp , должна выполняться свертка. Таким образом, в состояниях I_1 и I_3 по символам B и b выполняется перенос, а по символу \perp – свертка. Эти символы не совпадают, конфликта нет, неадекватность снимается.

В состоянии I_5 перенос выполняется по символам A и a , а свертка – по символам b и \perp , поскольку $Follow(A) = \{b, \perp\}$, т. е. если очередным входным символом является b или \perp , выполняется свертка для продукции $A \rightarrow a$. Неадекватность снимается.

В состояниях I_2 , I_4 и I_6 выполняются только свертки (конфликтов нет), поэтому при построении таблицы разбора можно было бы включить соответствующие элементы свертки во все столбцы этих состояний, как это делалось в $LR(0)$ -методе. Однако $SLR(1)$ -метод, рассматривая только возможные символы-следователи, исключает из таблицы несколько элементов свертки. В состоянии I_2 выполняется свертка для продукции $S \rightarrow AB$, а $Follow(S) = \{\perp\}$, т. е. за S может следовать только символ \perp и никакой другой. Поэтому для состояния I_2 элемент свертки $R1$ включается только в столбец \perp . Аналогично в состоянии I_4 элемент $R4$ (свертка для $B \rightarrow bB$) включается в столбец \perp , поскольку $Follow(B) = \{\perp\}$, а в состоянии I_6 элемент $R2$ (свертка для $A \rightarrow aA$) записывается в столбцы b и \perp , так как $Follow(A) = \{b, \perp\}$. Если этого не сделать, то некоторые синтаксические ошибки будут обнаружены на более поздних шагах анализа (но не позднее в смысле считанного текста). Соответствующая $SLR(1)$ -таблица разбора представлена в табл. 4.9.

Таблица 4.9

SLR(1)-таблица разбора

Номер состояния	<i>S</i>	<i>A</i>	<i>B</i>	<i>a</i>	<i>b</i>	\perp
0	stop	<i>S1</i>		<i>S5</i>		
1			<i>S2</i>		<i>S3</i>	<i>R5</i>
2						<i>R1</i>
3			<i>S4</i>		<i>S3</i>	<i>R5</i>
4						<i>R4</i>
5		<i>S6</i>		<i>S5</i>	<i>R3</i>	<i>R3</i>
6					<i>R2</i>	<i>R2</i>

Таким образом, если все конфликты можно разрешить рассмотренным выше способом, т. е. достаточно проанализировать значения функции *Follow* при реализации свертки, грамматика относится к подклассу *SLR(1)*-грамматик. В противном случае необходимо использовать более сложные и мощные методы построения таблиц разбора.

SLR(1)-метод часто используется на практике, так как большинство стандартных синтаксических конструкций языков программирования легко выражаются с помощью *SLR(1)*-грамматик. Однако возможны конструкции, с которыми *SLR(1)*-метод не может справиться.

Для наглядности на одной странице приведены и $SLR(1)$ -автомат, и $SLR(1)$ -таблица разбора

$SLR(1)$ -таблица разбора

Номер состояния	S	A	B	a	b	\perp
0	stop	$S1$		$S5$		
1			$S2$		$S3$	$R5$
2						$R1$
3			$S4$		$S3$	$R5$
4						$R4$
5		$S6$		$S5$	$R3$	$R3$
6					$R2$	$R2$

Состояние	Пункты	Символ перехода	Состояние-преемник	Свертка
I_0	$S' \rightarrow \bullet S \perp$	S	$stop$	
	$S \rightarrow \bullet AB$	A	I_1	
	$A \rightarrow \bullet aA$	a	I_5	
	$A \rightarrow \bullet a$			
I_1	$S \rightarrow A \bullet B$	B	I_2	
	$B \rightarrow \bullet bB$	b	I_3	
	$B \rightarrow \bullet$	\perp		$R5$
I_2	$S \rightarrow AB \bullet$	\perp		$R1$
I_3	$B \rightarrow b \bullet B$	B	I_4	
	$B \rightarrow \bullet bB$	b	I_3	
	$B \rightarrow \bullet$	\perp		$R5$
I_4	$B \rightarrow bB \bullet$	\perp		$R4$
I_5	$A \rightarrow a \bullet A$	A	I_6	
	$A \rightarrow a \bullet$	b, \perp		$R3$
	$A \rightarrow \bullet aA$	a	I_5	
	$A \rightarrow \bullet a$			
I_6	$A \rightarrow aA \bullet$	b, \perp		$R2$