

Глава 4. Восходящий синтаксический анализ

4.6. *LR*-таблицы разбора

4.6.4. *LR*(1)-грамматики

Наиболее общей и мощной технологией построения таблиц разбора является *LR*(1)-метод. Он, по сути, тот же, что и рассмотренный выше *LALR*(1)-метод. Отличие заключается в том, что в *LR*(1)-методе состояния с идентичными ядрами, но с несовпадающими предпросмотрами пунктов считаются различными, т. е. их нельзя объединять, как это делалось в *LALR*(1)-методе. Это позволяет разрешать ряд конфликтов, с которыми не справляется *LALR*(1)-метод.

Рассмотрим грамматику с продукциями

- | | |
|------------------------|------------------------|
| 1) $S \rightarrow aAd$ | 4) $S \rightarrow bAe$ |
| 2) $S \rightarrow bBd$ | 5) $A \rightarrow c$ |
| 3) $S \rightarrow aBe$ | 6) $B \rightarrow c$ |

LR(1)-автомат представлен в табл. 4.12, а соответствующая *LR*(1)-таблица разбора – в табл. 4.13.

Таблица 4.12 $LR(1)$ -автомат

Состояние	Пункты	Символ перехода	Состояние- преемник	Свертка
I_0	$S' \rightarrow \bullet S\perp, \{\varepsilon\}$	S	$stop$	
	$S \rightarrow \bullet aAd, \{\perp\}$	a	I_1	
	$S \rightarrow \bullet aBe, \{\perp\}$			
	$S \rightarrow \bullet bBd, \{\perp\}$	b	I_2	
	$S \rightarrow \bullet bAe, \{\perp\}$			
I_1	$S \rightarrow a\bullet Ad, \{\perp\}$	A	I_3	
	$S \rightarrow a\bullet Be, \{\perp\}$	B	I_4	
	$A \rightarrow \bullet c, \{d\}$	c	I_5	
	$B \rightarrow \bullet c, \{e\}$			
I_2	$S \rightarrow b\bullet Bd, \{\perp\}$	B	I_6	
	$S \rightarrow b\bullet Ae, \{\perp\}$	A	I_7	
	$A \rightarrow \bullet c, \{e\}$	c	I_8	
	$B \rightarrow \bullet c, \{d\}$			
I_3	$S \rightarrow aA\bullet d, \{\perp\}$	d	I_9	
I_4	$S \rightarrow aB\bullet e, \{\perp\}$	e	I_{10}	
I_5	$A \rightarrow c\bullet, \{d\}$	d		$R5$
	$B \rightarrow c\bullet, \{e\}$	e		$R6$
I_6	$S \rightarrow bB\bullet d, \{\perp\}$	d	I_{11}	
I_7	$S \rightarrow bA\bullet e, \{\perp\}$	e	I_{12}	
I_8	$A \rightarrow c\bullet, \{e\}$	e		$R5$
	$B \rightarrow c\bullet, \{d\}$	d		$R6$
I_9	$S \rightarrow aAd\bullet, \{\perp\}$	\perp		$R1$
I_{10}	$S \rightarrow aBe\bullet, \{\perp\}$	\perp		$R3$
I_{11}	$S \rightarrow bBd\bullet, \{\perp\}$	\perp		$R2$
I_{12}	$S \rightarrow bAe\bullet, \{\perp\}$	\perp		$R4$

$$1) S \rightarrow aAd$$

$$2) S \rightarrow bBd$$

$$3) S \rightarrow aBe$$

$$4) S \rightarrow bAe$$

$$5) A \rightarrow c$$

$$6) B \rightarrow c$$

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

$$Follow(A) = \{d, e\},$$

$$Follow(B) = \{d, e\}.$$

Таблица 4.13

LR(1)-таблица разбора

Номер состояния	<i>S</i>	<i>A</i>	<i>B</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	\perp
0	stop			<i>S1</i>	<i>S2</i>				
1		<i>S3</i>	<i>S4</i>			<i>S5</i>			
2		<i>S7</i>	<i>S6</i>			<i>S8</i>			
3							<i>S9</i>		
4								<i>S10</i>	
5							<i>R5</i>	<i>R6</i>	
6							<i>S11</i>		
7								<i>S12</i>	
8							<i>R6</i>	<i>R5</i>	
9									<i>R1</i>
10									<i>R3</i>
11									<i>R2</i>
12									<i>R4</i>

Грамматика является $LR(1)$ -грамматикой, поскольку все конфликты успешно разрешаются. Но она не относится к подклассу $LALR(1)$, так как в соответствии с $LALR(1)$ -методом состояния I_5 и I_8 , имеющие одно и то же ядро $\{[A \rightarrow c\bullet], [B \rightarrow c\bullet]\}$, должны объединяться в одно состояние с пунктами $[A \rightarrow c\bullet, \{d, e\}]$ и $[B \rightarrow c\bullet, \{d, e\}]$, что вызывает конфликт «сверка/сверка».

Рассмотренная выше грамматика специально усложнена для иллюстрации $LR(1)$ -метода. Генерируемый ею язык включает в себя четыре строки и может быть представлен эквивалентной $LR(0)$ -грамматикой с продукциями $S \rightarrow acd \mid bcd \mid ace \mid bce$.

Если для некоторой грамматики применить все рассмотренные методы построения таблиц разбора, то $LR(0)$ -, $SLR(1)$ - и $LALR(1)$ -таблицы разбора будут иметь одинаковое число состояний, а $LR(1)$ -таблица разбора – значительно больше. Таким образом, методы $LR(0)$, $SLR(1)$ и $LALR(1)$ проще и экономичнее общего $LR(1)$ -метода. Поскольку большинство конструкций языков программирования легко представляются с помощью $SLR(1)$ - или $LALR(1)$ -грамматик, лучше сначала опробовать $SLR(1)$ -метод. При успешной попытке грамматика будет $SLR(1)$ -грамматикой. В противном случае пробуются $LALR(1)$ -метод, и если это разрешает все конфликты, то данная грамматика обладает признаком $LALR(1)$. Если конфликты остаются, то используется наиболее общий $LR(1)$ -метод, и если он не приводит к успеху, то либо необходимо преобразовать грамматику, либо язык не относится к классу $LR(1)$ -языков и, следовательно, для него не может существовать $LR(1)$ -грамматики.

Очевидно, что классификация $LR(1)$ -грамматик включающая, т. е. все грамматики с признаками $LR(0)$, $SLR(1)$ и $LALR(1)$ являются $LR(1)$ -грамматиками, все грамматики с признаками $LR(0)$ и $SLR(1)$ являются $LALR(1)$ -грамматиками и т. д.

Рассмотренные выше таблицы разбора обеспечивают быструю выборку и широкие диагностические возможности. Главный их недостаток – для хранения требуется большой объем памяти. Можно использовать известные методы хранения неплотных матриц, но обычно это достигается за счет увеличения времени разбора и более позднего обнаружения синтаксических ошибок.