

Глава 4. Восходящий синтаксический анализ

4.6. LR -таблицы разбора

4.6.1. $LR(0)$ -грамматики

Рассмотрим самый легкий в реализации, но наименее мощный $LR(0)$ -метод построения LR -таблицы разбора, основанный на применении $LR(0)$ -грамматик. Для таких грамматик основа распознается без исследования предварительно просматриваемого символа, а пункт не содержит предпросмотра, т. е. $LR(0)$ -пункт $[A \rightarrow \alpha \bullet \beta]$ представляет собой только продукцию с маркерной точкой.

Процесс построения $LR(0)$ -автомата начинается с исходного пункта $[S' \rightarrow \bullet S \perp]$, где S' – начальный символ пополненной грамматики, которая включается в исходное состояние автомата. Затем выполняется *операция замыкания*, которая заключается в следующем. Если пункт вида $[A \rightarrow \alpha \bullet B \beta]$, $B \in V_N$, входит в некоторое состояние, то все пункты вида $[B \rightarrow \bullet \gamma]$, $\gamma \in (V_T \cup V_N)^*$, включаются в это же состояние (если их еще там нет). Этот процесс продолжается рекурсивно до тех пор, пока в состояние не будут включены все возможные пункты. Таким образом, если в пункте маркерная точка стоит перед нетерминалом B , то в текущее состояние включаются все пункты, в которых маркерная точка стоит перед первым символом правой части продукции с нетерминалом B в левой части. Если в каком-то из добавленных пунктов маркерная точка стоит перед нетерминалом, процесс повторяется. Такое выполнение операции замыкания объясняется следующим образом. Наличие $[A \rightarrow \alpha \bullet B \beta]$ в некотором состоянии указывает, что в данный момент в процессе синтаксического анализа предполагается, что во входном потоке может встретиться подстрока, выводимая из $B\beta$. Но если имеется продукция $B \rightarrow \gamma$, то, естественно, в этот момент может встретиться и строка, выводимая из γ . Поэтому $[B \rightarrow \bullet \gamma]$ должна быть включена в это же состояние (пункты $[A \rightarrow \alpha \bullet B \beta]$ и $[B \rightarrow \bullet \gamma]$ являются неразличимыми).

Из заданного состояния I , содержащего хотя бы один пункт, не соответствующий окончанию продукции, строится переход в другое состояние I' по символу грамматики, перед которым стоит маркерная точка. При этом точка перемещается на одну позицию вправо, т. е. будет стоять после этого символа. Состояние I' называется *преемником* состояния I . Новый пункт, который получается при образовании преемника, называется *базовым* для состояния I' . Таким образом, если пункт $[A \rightarrow \alpha \bullet X \beta]$, $X \in V_T \cup V_N$ принадлежит состоянию I , то базовым пунктом для состояния I' будет пункт $[A \rightarrow \alpha X \bullet \beta]$ (переход по символу X). Затем для базового пункта выполняется операция замыкания, чтобы включить в новое состояние все остальные неразличимые пункты. Следует отметить, что если состояние I содержит несколько пунктов, в которых точка стоит перед одним и тем же символом X , то в состоянии I' будет множество базовых пунктов, в которых точка будет стоять после X . Это связано с тем, что $LR(0)$ -автомат должен быть детерминированным, а в таких автоматах не допускается, чтобы из одного состояния существовал переход по одному и тому же символу в разные состояния.

Процесс последовательного выполнения операций замыкания и образования преемника продолжается до тех пор, пока все возможные пункты грамматики не окажутся включенными в какие-либо состояния. Следует иметь в виду, что автомат не должен иметь несколько состояний с одинаковыми множествами пунктов.

Рассмотрим процесс построения $LR(0)$ -автомата для грамматики со следующими productions (перед production указан ее порядковый номер):

1) $S \rightarrow SbA$

2) $S \rightarrow A$

3) $A \rightarrow a$

Поскольку мы рассматриваем пополненные грамматики, то имеется еще production $S' \rightarrow S\perp$. $LR(0)$ -автомат для данной грамматики представлен на рис. 4.6. Для удобства записи в пунктах квадратные скобки опущены.

- 1) $S \rightarrow SbA$
- 2) $S \rightarrow A$
- 3) $A \rightarrow a$

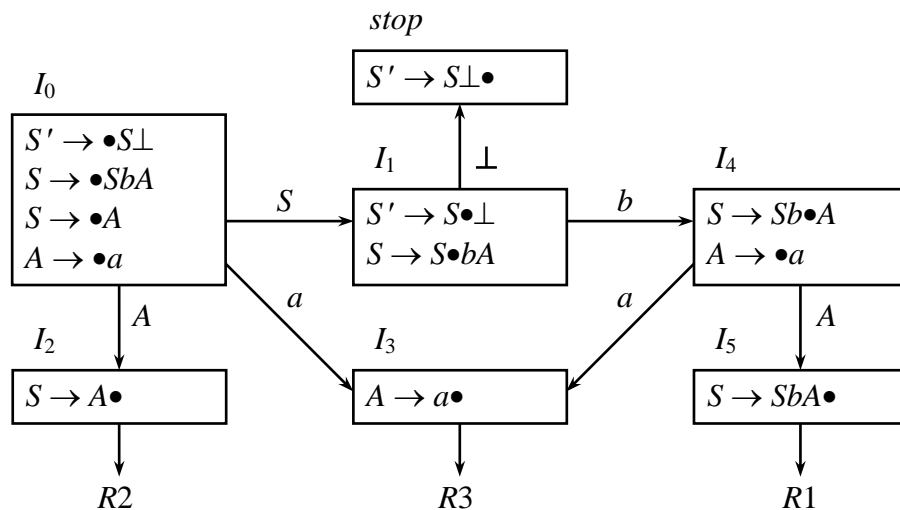


Рис. 4.6. $LR(0)$ -автомат

Образует исходное состояние I_0 . Базовым пунктом является начальный пункт $[S' \rightarrow \bullet S \perp]$. Выполним операцию замыкания. Поскольку маркерная точка стоит перед нетерминалом S , в состоянии I_0 необходимо включить пункты $[S \rightarrow \bullet SbA]$ и $[S \rightarrow \bullet A]$ для продукций с нетерминалом S в левой части. Среди добавленных пунктов появился нетерминал A , перед которым стоит точка, следовательно, необходимо включить в I_0 пункт $[A \rightarrow \bullet a]$. Операция замыкания выполнена, и определены все пункты для состояния I_0 .

В состоянии I_0 перед тремя разными символами стоит маркерная точка, т. е. будет три состояния-преемника: I_1 по символу S (базовые пункты $[S' \rightarrow S \bullet \perp]$ и $[S \rightarrow S \bullet bA]$), I_2 по символу A (базовый пункт $[S \rightarrow A \bullet]$) и I_3 по символу a (базовый пункт $[A \rightarrow a \bullet]$). Следует отметить, что состояния I_1 , I_2 и I_3 исчерпываются базовыми пунктами, так как в них нет пунктов с маркерными точками перед нетерминалами. Пункты в состояниях I_2 и I_3 соответствуют окончаниям продукций, поэтому они не имеют преемников. Это означает, что в этих состояниях должна выполняться свертка для соответствующих продукций. На рис. 4.6 действия свертки обозначены так же, как и элементы свертки в таблице разбора. Например, в состоянии I_2 выполняется свертка для продукции $S \rightarrow A$ с номером два (обозначено $R2$).

У состояния I_1 имеются два состояния-преемника: *stop* по символу \perp (базовый пункт $[S' \rightarrow S\perp\bullet]$) и I_4 по символу b (базовый пункт $[S \rightarrow Sb\bullet A]$). Состояние *stop* означает, что процесс разбора завершен. Операция замыкания базового пункта в состоянии I_4 добавляет в него пункт $[A \rightarrow \bullet a]$. Преемниками состояния I_4 являются состояния I_3 по символу a и I_5 по символу A (базовый пункт $[S \rightarrow SbA\bullet]$). В состоянии I_5 выполняется свертка $R1$ для продукции $S \rightarrow SbA$.

На этом процесс построения $LR(0)$ -автомата завершен, поскольку все возможные пункты заданной грамматики включены в соответствующие состояния. Следует отметить, что один пункт может входить в более чем одно состояние. Например, пункт $[S \rightarrow \bullet a]$ содержится в состояниях I_0 и I_4 .

В полученном автомате никаких конфликтов нет – это особенность $LR(0)$ -грамматик. Конфликт «свертка/свертка» возникает, если в одном состоянии возможно несколько сверток. Конфликт «перенос/свертка» возникает, если в состоянии есть свертка и возможен переход в другое состояние.

$LR(0)$ -автомат удобно представлять в табличной форме. Пример такого представления полученного $LR(0)$ -автомата показан в табл. 4.6. Очевидно, что для состояния *stop* нет смысла выделять специальную строку в таблице, поскольку это состояние не предполагает никаких действий, а только сигнализирует об успешном завершении разбора.

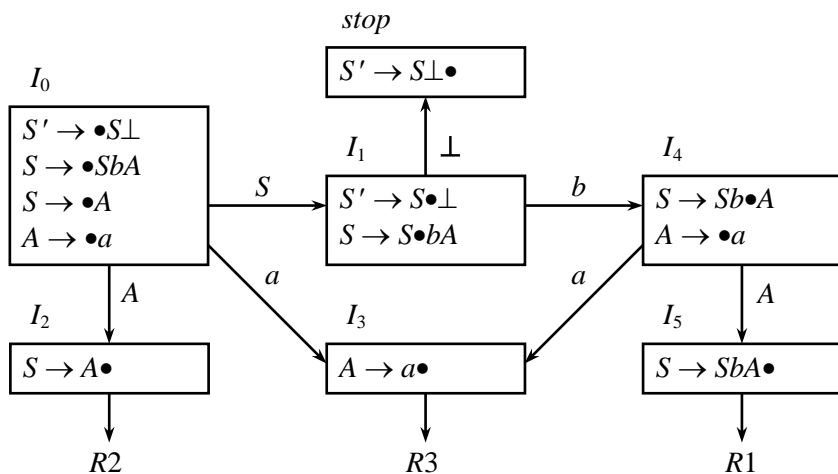


Таблица 4.6

Табличное представление $LR(0)$ -автомата

Состояние	Пункты	Символ перехода	Состояние-приемник	Свертка
I_0	$S' \rightarrow \bullet S \perp$	S	I_1	
	$S \rightarrow \bullet SbA$			
	$S \rightarrow \bullet A$		I_2	
	$A \rightarrow \bullet a$		I_3	
I_1	$S' \rightarrow S \bullet \perp$	\perp	$stop$	
	$S \rightarrow S \bullet bA$	b	I_4	
I_2	$S \rightarrow A \bullet$			$R2$
I_3	$A \rightarrow a \bullet$			$R3$
I_4	$S \rightarrow Sb \bullet A$	A	I_5	
	$A \rightarrow \bullet a$		I_3	
I_5	$S \rightarrow SbA \bullet$			$R1$

По $LR(0)$ -автомату легко строится таблица разбора. Сначала заносятся элементы переноса. Переход из одного состояния в другое соответствует действию переноса. Поэтому элементы переноса вносятся в таблицу разбора на основании информации о состояниях и переходах автомата. Если из состояния I_i существует переход в состояние I_j по символу X , то на пересечении строки i и столбца X записывается элемент переноса S_j . Если осуществляется переход в состояние $stop$, то в таблицу разбора вносится элемент останова $stop$. Следует отметить, что внесение элементов переноса не зависит от рассматриваемой $LR(1)$ -грамматики или ее подкласса. Затем в таблицу разбора вносятся элементы свертки. Поскольку для $LR(0)$ -грамматик не учитывается предварительно просматриваемый символ, а в $LR(0)$ -автомате нет конфликтов, элементы свертки могут помещаться в каждый столбец в любом состоянии, соответствующем окончанию продукции (табл. 4.7).

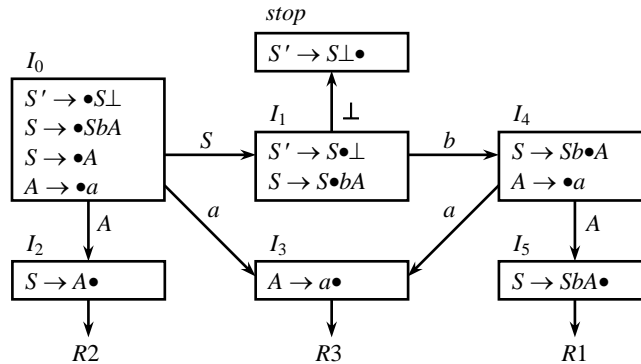


Таблица 4.7

LR(0)-таблица разбора

Номер состояния	S'	S	A	a	b	\perp
0		$S1$	$S2$	$S3$		
1					$S4$	<i>stop</i>
2	$R2$	$R2$	$R2$	$R2$	$R2$	$R2$
3	$R3$	$R3$	$R3$	$R3$	$R3$	$R3$
4			$S5$	$S3$		
5	$R1$	$R1$	$R1$	$R1$	$R1$	$R1$

Построенная рассмотренным выше методом таблица разбора называется $LR(0)$ -таблицей разбора, а соответствующая грамматика – $LR(0)$ -грамматикой. Таким образом, чтобы определить, обладает ли грамматика признаком $LR(0)$, необходимо попытаться построить $LR(0)$ -таблицу разбора. Если это можно осуществить без возникновения каких-либо конфликтов (элементы свертки можно включить в каждый столбец в любом состоянии, соответствующем окончанию продукции), то грамматика является $LR(0)$ -грамматикой. Если же конфликты возникают, грамматика не относится к классу $LR(0)$ и следует применить более универсальные методы.