

## Глава 3. Нисходящий синтаксический анализ

### 3.6. Табличные методы нисходящего разбора

Табличные методы синтаксического анализа аналогичны рекурсивному спуску. Здесь исключаются многочисленные вызовы процедур благодаря представлению грамматики в табличном виде (*таблицы разбора, управляющие таблицы*) и использованию независящего от анализируемого языка модуля компилятора, проводящего синтаксический разбор по таблице.

Основным достоинством табличных методов разбора является то, что модуль синтаксического анализатора можно применять многократно в компиляторах для различных языков, изменив только содержимое таблицы разбора. Процесс формирования таблиц разбора для  $LL(1)$ -грамматик обычно носит детерминированный характер. Поэтому этот процесс можно легко автоматизировать, написав программу для получения соответствующей таблицы разбора. В результате сроки проектирования компиляторов существенно сокращаются.

Таблицы разбора организуются таким образом, что модуль синтаксического анализа компилятора всегда указывает на то место в синтаксисе, которое соответствует текущему входному символу. Модулю требуется стек, содержащий последовательность символов грамматики или адреса возврата при обработке новой порождающей продукции, соответствующей какому-либо нетерминалу. Представление синтаксиса в таблице должно быть таким, чтобы обеспечить эффективность синтаксического анализатора в отношении скорости работы.

Возможны различные виды таблиц разбора, которые являются, по сути, различными формами представления магазинного автомата, принимающего данный контекстно-свободный язык.

### 3.6.1. Таблица переходов автомата с магазинной памятью

Часто в качестве управляющей таблицы используют таблицу переходов детерминированного МП-автомата (возможно, слегка модифицированную). В общем случае недетерминированный МП-автомат  $M = (K, T, \Gamma, \delta, k_0, Z_0, F)$ , распознающий язык, заданный КС-грамматикой  $G = (V_T, V_N, P, S)$ , определяется как  $M = (\{k\}, V_T, V_T \cup V_N, \delta, k, S, \{k\})$ , где функция переходов  $\delta$  определяется следующим образом:

$$\delta(k, \varepsilon, A) = \{(k, \alpha) \mid A \rightarrow \alpha \in P\} \text{ для всех } A \in V_N;$$

$$\delta(k, a, a) = \{(k, \varepsilon)\} \text{ для всех } a \in V_T.$$

Наличие множеств направляющих символов  $DS(A \rightarrow \alpha)$  для продукций  $LL(1)$ -грамматики позволяет определить детерминированный МП-автомат, распознающий язык, заданный соответствующей  $LL(1)$ -грамматикой. Для этого модифицируется определение функции переходов  $\delta$ :

$$\text{для всех продукций } A \rightarrow \alpha \in P \quad \delta(k, a, A) = (k, \alpha) \text{ для всех } a \in DS(A \rightarrow \alpha);$$

$$\delta(k, a, a) = (k, \varepsilon) \text{ для всех } a \in V_T.$$

Тогда таблицу переходов МП-автомата, используемую в качестве управляющей таблицы, можно задать отображением множества  $(V_T \cup V_N) \times V_T$  (маркер конца ввода  $\perp \in V_T$  выполняет также функцию *маркера дна* стека; если он находится в вершине стека, то стек пуст). Таким образом, строкам таблицы соответствуют элементы стекового алфавита (объединение множеств терминалов и нетерминалов), столбцы соответствуют входному алфавиту (множеству терминалов).

Элементами управляющей таблицы  $M$  являются:

1. Правая часть продукции  $A \rightarrow \alpha$ , соответствует значению функции переходов  $\delta(k, a, A) = (k, \alpha)$ . В ряде модификаций в качестве элемента используют пару  $(\alpha, i)$ , где  $i$  – номер продукции, или просто номер  $i$  продукции.

2. Элемент «выброс», соответствует значению функции переходов  $\delta(k, a, a) = (k, \varepsilon)$ ,  $a \neq \perp$ .

3. Элемент «допуск», соответствует значению функции переходов  $\delta(k, \perp, \perp) = (k, \varepsilon)$ .

4. Элемент «ошибка» – незаполненный элемент таблицы, соответствует синтаксической ошибке.

В начальной конфигурации входной буфер содержит анализируемую строку  $w\perp$ ,  $w \in (V_T - \{\perp\})^*$ , текущим входным символом является первый символ этой строки, стек содержит  $S\perp$ , где  $S$  – начальный символ грамматики. При нисходящем анализе содержимое стека удобно представлять строкой, в которой самый левый символ строки считается верхним символом стека.

По элементу  $M[X, a]$  управляющей таблицы, где  $X$  – символ в вершине стека,  $a$  – текущий входной символ, определяются действия, предпринимаемые синтаксическим анализатором в процессе разбора входной строки:

1. Если  $M[X, a] = \alpha$  (в этом случае символ  $X$  в вершине стека может быть только нетерминалом), символ  $X$  в вершине стека замещается строкой  $\alpha$ .

2. Если  $M[X, a] = \text{«выброс»}$  и  $a \neq \perp$  (в этом случае символ  $X$  в вершине стека может быть только терминалом  $a \neq \perp$ , т. е. символ в вершине стека совпадает с текущим входным символом), символ  $a$  исключается из стека и реализуется переход к анализу следующего входного символа. Это означает, что входной символ  $a$  принимается синтаксическим анализатором.

3. Если  $M[X, a] = \text{«допуск»}$  (случай возможен только для  $X = \perp$  и  $a = \perp$ ), входная строка принимается, разбор завершается.

4. Если  $M[X, a] = \text{«ошибка»}$ , разбор прекращается и формируется соответствующее сообщение об ошибке.

Рассмотрим  $LL(1)$ -грамматику со следующими продукциями, предполагая наличие продукции  $S' \rightarrow S\perp$  (для каждой продукции справа указаны множества направляющих символов):

$$S \rightarrow TC \quad \{a, c, \perp\}$$

$$T \rightarrow aTb \quad \{a\}$$

$$T \rightarrow \varepsilon \quad \{b, c, \perp\}$$

$$C \rightarrow cC \quad \{c\}$$

$$C \rightarrow \varepsilon \quad \{\perp\}$$

Управляющая таблица для этой грамматики представлена на рис. 3.2.

	$a$	$b$	$c$	$\perp$
$S$	$TC$		$TC$	$TC$
$T$	$aTb$	$\varepsilon$	$\varepsilon$	$\varepsilon$
$C$			$cC$	$\varepsilon$
$a$	<i>выброс</i>			
$b$		<i>выброс</i>		
$c$			<i>выброс</i>	
$\perp$				<i>допуск</i>

Рис. 3.2. Управляющая таблица

Процесс разбора строки  $aabbcs\perp$ , которая выводится в соответствии с левосторонней схемой

$S' \Rightarrow S\perp \Rightarrow TC\perp \Rightarrow aTbC\perp \Rightarrow aaTbbC\perp \Rightarrow aabbC\perp \Rightarrow aabbcsC\perp \Rightarrow aabbcs\perp$ ,  
приведен в табл. 3.1.

Таблица 3.1. Процесс разбора строки  $aabbc\perp$

Входная строка	Содержимое стека	Комментарии
$aabbc\perp$	$S\perp$	$M[S, a] = TC$ , замещение $S$ строкой $TC$
$aabbc\perp$	$TC\perp$	$M[T, a] = aTb$ , замещение $T$ строкой $aTb$
$aabbc\perp$	$aTbC\perp$	$M[a, a] = \text{«выброс»}$ , $a$ принимается
$abbc\perp$	$TbC\perp$	$M[T, a] = aTb$ , замещение $T$ строкой $aTb$
$abbc\perp$	$aTbbC\perp$	$M[a, a] = \text{«выброс»}$ , $a$ принимается
$bbc\perp$	$TbbC\perp$	$M[T, b] = \varepsilon$ , замещение $T$ пустой строкой $\varepsilon$
$bbc\perp$	$bbC\perp$	$M[b, b] = \text{«выброс»}$ , $b$ принимается
$bc\perp$	$bC\perp$	$M[b, b] = \text{«выброс»}$ , $b$ принимается
$c\perp$	$C\perp$	$M[C, c] = cC$ , замещение $T$ строкой $cC$
$c\perp$	$cC\perp$	$M[c, c] = \text{«выброс»}$ , $c$ принимается
$\perp$	$C\perp$	$M[C, \perp] = \varepsilon$ , замещение $C$ пустой строкой $\varepsilon$
$\perp$	$\perp$	$M[\perp, \perp] = \text{«допуск»}$ , разбор успешно завершен



Следует отметить, что в любой строке управляющей таблицы, соответствующей терминалу, возможен только один элемент «выброс» («допуск» для  $\perp$ ), а все остальные элементы являются элементами «ошибка». Действия, связанные с этими элементами, реализуются только в том случае, если в вершине стека находится терминал (включая и маркер  $\perp$ ). Такую ситуацию легко можно распознать и включить соответствующие действия в алгоритм синтаксического анализа. Поэтому на практике для уменьшения размеров в управляющую таблицу обычно включают только строки, соответствующие нетерминалам. Синтаксический анализ с применением модифицированной управляющей таблицы представлен алгоритмом 3.1,

где  $SP$  – рабочий стек синтаксического анализатора (в начальном состоянии он должен содержать  $S\perp$ , где  $S$  – начальный символ грамматики,  $\perp$  – маркер конца ввода);

процедура  $read(a)$  считывает из входной строки очередной символ и присваивает его значение переменной  $a$  (соответствует обращению к сканеру за очередным токеном);

процедура  $syntax\_error$  каким-либо образом обрабатывает синтаксическую ошибку и прекращает разбор.

---

**Алгоритм 3.1.** Предиктивный синтаксический анализ с применением модифицированной управляющей таблицы

**Вход:** Управляющая таблица, анализируемая строка

**Выход:** Сообщения о результатах синтаксического анализа

---

$SP \leftarrow \perp$ ;  $SP \leftarrow S$  // начальное состояние стека  $SP$   
 $read(a)$  // в  $a$  первый символ входной строки  
**while**  $SP \neq \emptyset$  **do** // пока стек  $SP$  не пуст  
     $X \leftarrow SP$  // в  $X$  символ, исключенный из вершины стека  $SP$   
    **if**  $X = a$  **then**  $\left\{ \begin{array}{l} \text{// входной символ } a \text{ принимается} \\ read(a) \text{ // в } a \text{ очередной входной символ} \end{array} \right.$   
    **else if**  $X \in V_T$  **then**  $\left\{ \begin{array}{l} \text{// входной символ не совпадает} \\ \text{// с терминалом в вершине стека} \\ syntax\_error \end{array} \right.$   
    **else if**  $M[X, a] = error$  **then**  $syntax\_error$   
    **else if**  $M[X, a] = \alpha$  **then**  $\left\{ \begin{array}{l} \text{// поместить в стек } SP \text{ строку } \alpha \\ SP \leftarrow \alpha \end{array} \right.$

---