

Глава 4. Восходящий синтаксический анализ

4.1. Построение дерева разбора

При восходящем синтаксическом анализе дерево разбора для входной строки строится снизу вверх, начиная с листьев и завершая корнем дерева. Этот процесс можно рассматривать как свертку входной строки к начальному символу грамматики. Входная строка анализируется слева направо в поисках подстроки, которая может быть свернута, т. е. ищется правая часть продукции грамматики, совпадающая с подстрокой. Если такая продукция найдена, то соответствующая подстрока заменяется нетерминалом левой части продукции. В результате такой замены может быть получена сентенциальная форма грамматики. Процесс повторяется до тех пор, пока входная строка не преобразуется в начальный символ грамматики (корень дерева разбора).

Действие замены подстроки нетерминалом левой части продукции называется *сверткой*, а свертываемая подстрока – *основой правосторонней сентенциальной формы* (или просто *основой*). Таким образом, основа – это подстрока, совпадающая с правой частью продукции, свертка которой в левую часть продукции представляет собой один шаг правосторонней схемы вывода в обратном направлении. Формально основой правосторонней сентенциальной формы $\alpha\beta w$ ($\alpha, \beta \in (V_T \cup V_N)^*$, $w \in V_T^*$) является подстрока β , если существует продукция $A \rightarrow \beta$ такая, что β может быть заменена нетерминалом A для получения предыдущей сентенциальной формы в правосторонней схеме вывода

$$S \xRightarrow{*} \alpha A w \Rightarrow \alpha \beta w.$$

Следует обратить внимание на то, что подстрока w может состоять только из терминалов. Из определения следует, что не любая подстрока β , соответствующая правой части некоторой продукции $A \rightarrow \beta$, является основой, поскольку свертка для этой продукции может привести к строке, которая не может быть в последующем свернута к начальному символу грамматики.

Пример построения дерева разбора снизу вверх для грамматики с продукциями

$$S \rightarrow AB$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow bB \mid b$$

при разборе строки $aabb$ представлен на рис. 4.1. Это дерево соответствует правосторонней схеме вывода

$$S \Rightarrow AB \Rightarrow AbB \Rightarrow Abb \Rightarrow aAbb \Rightarrow aabb .$$

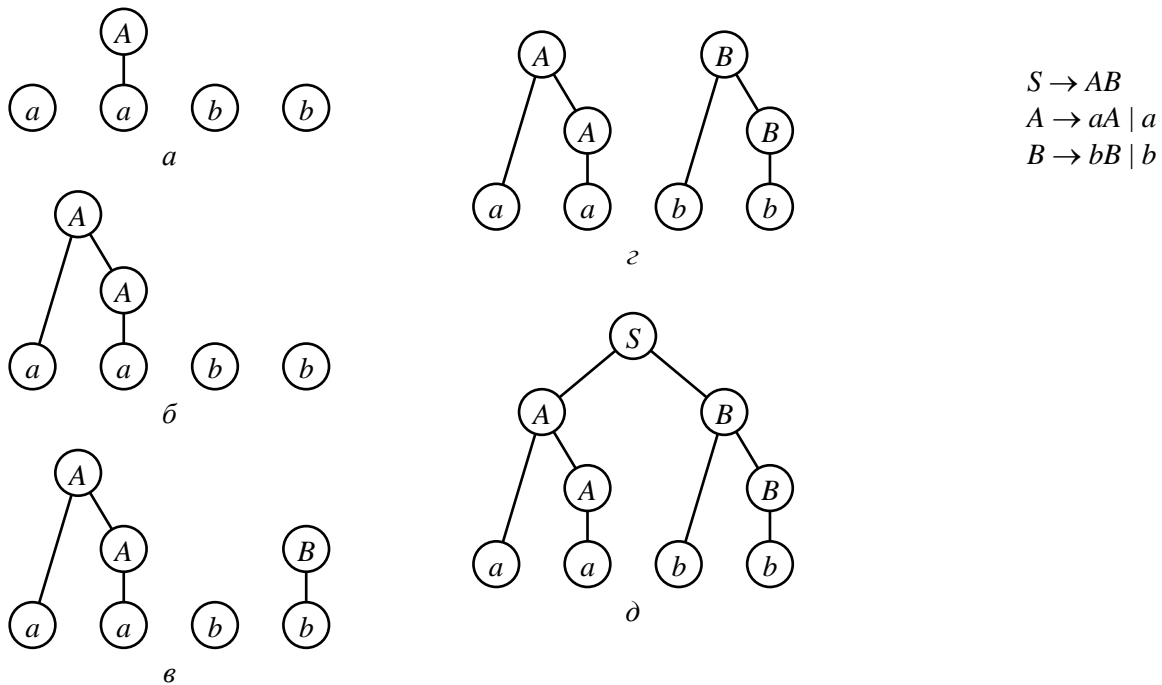


Рис. 4.1. Процесс построения дерева разбора снизу вверх

правосторонняя схема вывода: $S \Rightarrow AB \Rightarrow AbB \Rightarrow Abb \Rightarrow aAbb \Rightarrow aabb$.

В приведенном примере строка $aabb$ представляет собой правостороннюю сентенциальную форму, основой которой является второй символ a (но не первый!) в соответствии с продукцией $A \rightarrow a$. В результате свертки получается правосторонняя сентенциальная форма $aAbb$ (рис. 4.1, *a*), основой которой является подстрока aA в соответствии с продукцией $A \rightarrow aA$. В результате свертки получается строка Abb (рис. 4.1, *б*). Такая последовательность операций определения основы и свертки продолжается до полного построения дерева разбора: основа сентенциальной формы Abb – второй символ b (но не первый!) в соответствии с продукцией $B \rightarrow b$, результат свертки – строка AbB (рис. 4.1, *в*); основа сентенциальной формы AbB – подстрока bB в соответствии с продукцией $B \rightarrow bB$, результат свертки – строка AB (рис. 4.1, *г*); сентенциальная форма AB сама является основой в соответствии с продукцией $S \rightarrow AB$, результат свертки – начальный символ грамматики S , который является корнем построенного дерева разбора (рис. 4.1, *д*).

Следует заметить, что если бы в строке $aabb$ заменили первый символ a на нетерминал A , то получили бы строку $Aabb$, которую невозможно свернуть в S , т. е. первый символ a не является основой, хотя и имеется продукция $A \rightarrow a$. По аналогичной причине и первый символ b не является основой для строки Abb .

Таким образом, при восходящем синтаксическом анализе основной проблемой является поиск и своевременная свертка основы сентенциальной формы на каждом шаге построения дерева разбора.

В синтаксических анализаторах процесс восходящего анализа реализуется с помощью стека, который соответствует части магазинного автомата. При этом анализатор выполняет две основные операции:

- 1) *перенос*, во время которого считывается символ входной строки и помещается в вершину стека (поэтому основа будет находиться в верхней части стека);
- 2) *свертка* заключается в замене множества элементов в верхней части стека (основы) на один элемент – нетерминал из левой части соответствующей продукции.

В связи с этим построенные на таких принципах синтаксические анализаторы часто называют анализаторами типа «перенос-свертка».

Процесс анализа строки $aabb$ для приведенной выше грамматики показан на рис. 4.2. Содержимое стека представляется строкой, в которой самый правый символ находится в вершине стека, символ \perp показывает дно стека.

Разбор считается успешно завершенным, когда в стеке остается только начальный символ грамматики и полностью считана входная строка.

Вводимая строка	Содержимое стека	Выполняемые действия
$aabb$	\perp	Перенос a в стек
abb	$\perp a$	Перенос a в стек
bb	$\perp aa$	Свертка для продукции $A \rightarrow a$
bb	$\perp aA$	Свертка для продукции $A \rightarrow aA$
bb	$\perp A$	Перенос b в стек
b	$\perp Ab$	Перенос b в стек
ε	$\perp Abb$	Свертка для продукции $B \rightarrow b$
ε	$\perp AbB$	Свертка для продукции $B \rightarrow bB$
ε	$\perp AB$	Свертка для продукции $S \rightarrow AB$
ε	$\perp S$	Разбор успешно завершён

Рис. 4.2. Процесс разбора строки $aabb$

При такой организации восходящего анализа проблема поиска основы и ее своевременной свертки сводится к проблеме разрешения конфликтов типа «перенос/свертка» (анализатор в конкретной ситуации должен знать, какое из этих действий необходимо выполнить) и «свертка/свертка» (если возможны разные свертки, необходимо знать, какую из них выполнить и выполнять ли их вообще). Очевидно, что необходимым условием для выполнения свертки является то, чтобы правая часть какой-либо продукции появилась в верхней части стека. Но это условие не является достаточным. Такие ситуации встречались и в рассмотренном выше примере. Поэтому для разрешения конфликтов анализатор должен иметь дополнительную информацию.