

## Глава 3. Нисходящий синтаксический анализ

### 3.3. $LL(k)$ -грамматики

Нисходящие методы синтаксического анализа основаны на просмотре входной строки и построении дерева разбора, начиная с начального нетерминала грамматики. Дерево строится до тех пор, пока не получится анализируемая строка терминалов. Если такую строку удастся получить, то анализируемая строка принадлежит языку, если нет – не принадлежит. Этот процесс равносильен процессу построения соответствующей схемы вывода анализируемой строки.

Во все рассматриваемые далее грамматики введем специальный символ  $\perp$ , принадлежащий множеству терминалов и обозначающий конец вводимой строки (*маркер конца ввода*). Такие грамматики можно назвать *пополненными*. Даже если маркер явно не указан в грамматике, будем подразумевать, что всегда имеется продукция вида  $S' \rightarrow S\perp$ , где  $S'$  и  $S$  – начальные нетерминалы соответственно пополненной и исходной грамматик.

Рассмотрим так называемый  $LL$ -разбор. Для успешного  $LL$ -разбора строк некоторого КС-языка на порождающую его грамматику должны быть наложены строгие ограничения. Практически это означает, что методом  $LL$ -разбора можно воспользоваться лишь на подмножестве класса детерминированных языков.

Рассмотрим грамматику с продукциями

$$S \rightarrow A\perp$$

$$A \rightarrow aFaB \mid bFbB$$

$$F \rightarrow b \mid ba$$

$$B \rightarrow b \mid bB$$

Построим дерево нисходящего синтаксического разбора строки  $abaabb\perp$  (рис. 3.1). Корень дерева помечен символом  $S$  и, следовательно, единственной продукцией грамматики, которая может быть использована в данном случае, является продукция  $S \rightarrow A\perp$ . Перейдем теперь к вершине  $A$ . Можно воспользоваться продукциями  $A \rightarrow aFaB$  и  $A \rightarrow bFbB$ . Поскольку рассматриваемая строка  $abaabb\perp$  начинается с символа  $a$ , воспользуемся первой из этих продукций, т. е. выбирается та продукция, правая часть которой начинается с очередного введенного входного символа. Итак, один из листьев дерева уже помечен символом  $a$ .

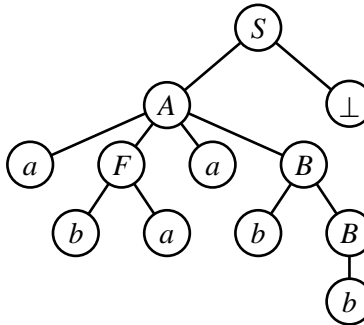


Рис. 3.1. Дерево разбора строки  $abaabb\perp$

Далее необходимо показать, что строка  $FaB\perp$  может породить строку  $baabb\perp$ . Рассмотрим для этого продукции  $F \rightarrow b|ba$ . На этот раз знать следующий введенный символ и даже два следующих введенных символа входной строки недостаточно для выбора одной из этих продукций. Однако если станут известны три очередных символа входной строки и если это символы  $bab$ , то следует воспользоваться первой продукцией, а если это символы  $baa$ , то второй продукцией. В данном случае очередные символы входной строки – это символы  $baa$ , поэтому следует выбрать продукцию  $F \rightarrow ba$ . Теперь необходимо убедиться, что из терминала  $B$  можно породить строку  $bb$ . Для этого надо рассмотреть продукции  $B \rightarrow b|bB$ . Опять знания следующего символа  $b$  недостаточно, но если известны два очередных символа и это символы  $b\perp$ , то следует использовать продукцию  $B \rightarrow b$ , а если это символы  $bb$ , – продукцию  $B \rightarrow bB$ . В данном случае очередные символы – это символы  $bb$ , поэтому используем продукцию  $B \rightarrow bB$ . Затем, согласно двум очередным символам входной строки, применим продукцию  $B \rightarrow b$ . В результате будет получена левосторонняя схема вывода

$$S \Rightarrow A\perp \Rightarrow aFaB\perp \Rightarrow abaaB\perp \Rightarrow abaabB\perp \Rightarrow abaabb\perp.$$

Эта грамматика является примером  $LL(3)$ -грамматики, т. е. для однозначного определения дерева вывода достаточно знать не более трех очередных символов входной строки на каждом этапе построения дерева.

В общем случае  $LL(k)$ -грамматикой называют такую грамматику  $G = (V_T, V_N, P, S)$ , что для любой ее сентенциальной формы  $wAy$ ,  $w \in V_T^*$ ,  $A \in V_N$ ,  $y \in (V_T \cup V_N)^*$ , полученной в результате некоторого левостороннего вывода, для однозначного выбора продукции, имеющей в левой части нетерминал  $A$ , достаточно знать  $k$  очередных символов входной строки. Аббревиатура  $LL$  означает "левосторонний ввод – левосторонний вывод".

Чем больше  $k$ , тем больший класс языков может быть представлен  $LL(k)$ -грамматикой, но и разбор выполняется значительно сложнее. Поэтому на практике наибольшее применение имеют  $LL(1)$ -грамматики, для которых детерминированный распознаватель работает, анализируя по одному входному символу, расположенному в текущей позиции.