

Лекция №8**Иерархия целей разбора**

Целью разбора на каждом шаге алгоритма является нетерминальный символ, опознание отдельных нетерминалов потребует в свою очередь того куста дерева формальной грамматики узлами которого они являются. Существуют цели текущие, глобальные и достигнутые.

В ходе выполнения алгоритма происходит перемещение по графу формальной грамматики, каждый вновь встреченный нетерминал объявляется текущей целью, после того как он стаёт текущей целью выполнятся переход на правило, которое его описывает. Перемещение по графу формальной грамматики, пока не будет достигнут финальный символ. После этого он сопоставляется с текущим символом. Цель считается опознанной, если в ходе разбора удаётся достигнуть конца правила. Также нормальным завершением работы алгоритма служить ситуация, когда достигнут конец строки.

Недостатки:

1. Время анализа программы в общем случае случайно поскольку порядок следования правил грамматики может не соответствовать порядку следования реальной программы
2. Для работы необходимо отсутствие лево-рекурсивных правил (конвертировать в грамматику формы Грейбах)

Алгоритм простого нисходящего разбора представлен в файле теория.алгоритмов.

Грамматика $G[S]$ является $LL(k)$ грамматикой, если для её сентанциальной формы $S \Rightarrow^* vAw$ ($A \in V_N$, $v \in V^+$, $w \in V^*$) выбор очередного правила с нетерминалом A в левой части однозначно определяется как текущими символами анализируемой строки.

Алгоритм трестирующий на $LL(k)$ грамматику начинают с $k = 1$:

1. Каждому правилу формальной грамматики необходимо поставить в соответствие множество цепочек длины k , которые выводятся при его использовании. Для индекса $k=1$ первоначально в множество попадают все нетерминалы, которые являются головами правых частей правил.
2. Если голова правой части правила нетерминальный символ, то ей ставится в соответствие множество цепочек, полученных для этого нетерминального символа.
3. Если множество цепочек для правил с одинаковыми левыми частями не пересекаются то данная грамматика является $LL(k)$ грамматикой.

Метод рекурсивного спуска состоит в том, что каждому символу языка ставится в соответствие синтаксическая функция, которая проверяет корректность написания данного элемента. Метод прост в реализации (последовательность вызовов синтаксических функций совпадает с последовательностью их появления в правых частях соответствующих правил). Такие анализаторы работают до первой обнаруженной ошибки.