

Лекція 8. Синтаксичний аналізатор. Парсінг

Назначение синтаксического разбора

Синтаксический разбор (распознавание) является первым этапом синтаксического анализа. Именно при его выполнении осуществляется подтверждение того, что входная цепочка символов является программой, а отдельные подцепочки составляют синтаксически правильные программные объекты. Вслед за распознаванием отдельных подцепочек осуществляется анализ их семантической корректности на основе накопленной информации. Затем проводится добавление новых объектов в объектную модель программы или в промежуточное представление.

Разбор предназначен для доказательства того, что анализируемая входная цепочка, записанная на входной ленте, принадлежит или не принадлежит множеству цепочек порождаемых грамматикой данного языка. Выполнение синтаксического разбора осуществляется распознавателями, являющимися автоматами. Поэтому данный процесс также называется распознаванием входной цепочки. Цель доказательства в том, чтобы ответить на вопрос: принадлежит ли анализируемая цепочка множеству правильных цепочек заданного языка. Ответ "да" дается, если такая принадлежность установлена. В противном случае дается ответ "нет". Получение ответа "нет" связано с понятием отказа. Единственный отказ на любом уровне ведет к общему отказу.

Чтобы получить ответ "да" относительно всей цепочки, надо его получить для каждого правила, обеспечивающего разбор отдельной подцепочки. Так как множество правил образуют иерархическую структуру, возможно с рекурсиями, то процесс получения общего положительного ответа можно интерпретировать как сбор по определенному принципу ответов для листьев, лежащих в основе дерева разбора, что дает положительный ответ для узла, содержащего эти листья. Далее анализируются обработанные узлы, и уже в них полученные ответы складываются в общий ответ нового узла. И так далее до самой вершины. Данный принцип обработки сильно напоминает бюрократическую систему, используемую в организационном управлении любого предприятия. Так поднимается вверх информация, подтверждающая выполнение указания начальника организации. До этого, теми же путями, вниз спускалось и разделялось исходное указание.

Классификация методов синтаксического разбора

Если попытаться формализовать задачу на уровне элементарного метаязыка, то она будет ставиться следующим образом. Дан язык $L(G)$ с грамматикой G , в которой S - начальный нетерминал. Построить дерево разбора входной цепочки

$$a = a_1 a_2 a_3 \dots a_n.$$

Естественно, что существует огромное количество путей решения данной задачи, и целью разработчика распознавателя является выделение приемлемых вариантов его реализации. Для того, чтобы понять, что, и каким образом, влияет на принципы функционирования

распознавателя, а следовательно, и на организацию разбора, рассмотрим некоторые возможные варианты. Общая классификация рассматриваемых вариантов построения распознавателя представлена на рис. 6.1.

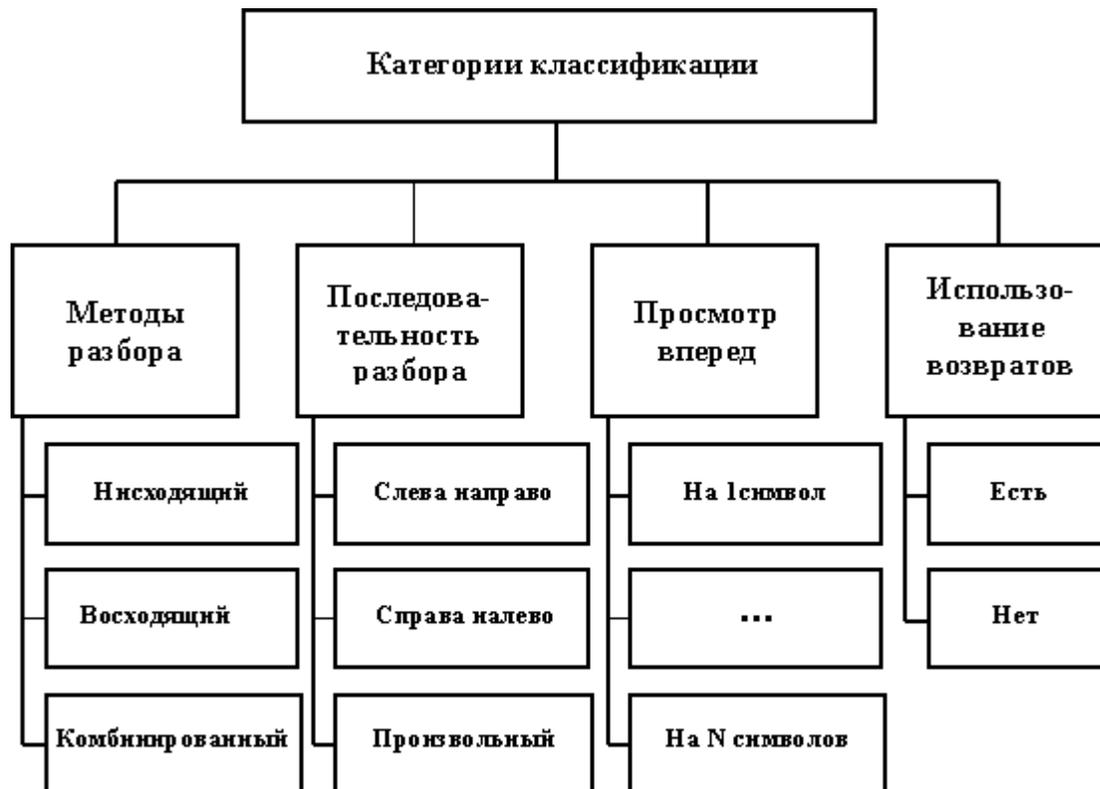


Рис. 6.1. Классификация методов организации синтаксического разбора.

На самом верхнем уровне выделяются:

- методы разбора;
- последовательность разбора;
- использование просмотра вперед;
- использование возвратов.

Методы разбора

Выделяются два основных метода синтаксического разбора:

- нисходящий разбор;
- восходящий разбор.

Кроме этого можно использовать комбинированный разбор, сочетающий особенности двух предыдущих.

Нисходящие и восходящие подходы широко используются в различных областях человеческой деятельности, особенно в тех из них, которые связаны с анализом и синтезом

искусственных систем. В частности, можно отметить методы разработки программного обеспечения сверху вниз (нисходящий) и снизу вверх (восходящий).

Нисходящий разбор заключается в построении дерева разбора, начиная от корневой вершины. Разбор заключается в заполнении промежутка между начальным нетерминалом и символами входной цепочки правилами, выводимыми из начального нетерминала. Подстановка основывается на том факторе, что корневая вершина является узлом, состоящим из листьев, являющихся цепочкой терминалов и нетерминалов одного из альтернативных правил, порождаемых начальным нетерминалом. Подставляемое правило в общем случае выбирается произвольно. Вместо новых нетерминальных вершин осуществляется подстановка выводимых из них правил. Процесс протекает до тех пор, пока не будут установлены все связи дерева, соединяющие корневую вершину и символы входной цепочки, или пока не будут перебраны все возможные комбинации правил. В последнем случае входная цепочка отвергается. Построение дерева разбора подтверждает принадлежность входной цепочки данному языку. При этом, в общем случае, для одной и той же входной цепочки может быть построено несколько деревьев разбора. Это говорит о том, что грамматика данного языка является недетерминированной.

Эти рассуждения иллюстрируются следующим примером. Пусть будет дана грамматика G :

$$G_6 = (\{S\}, \{a, +, *\}, P, S)$$

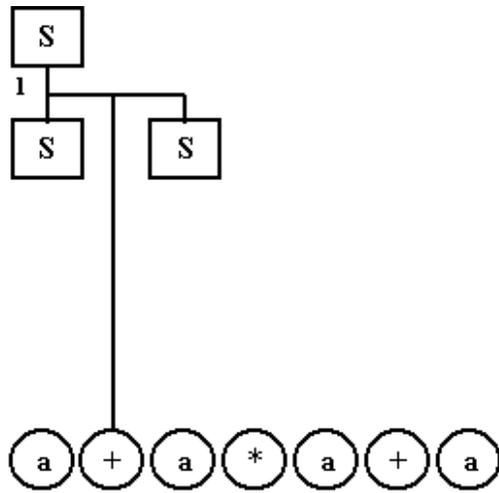
Где P определяется как:

1. $S \rightarrow a$
2. $S \rightarrow S + S$
3. $S \rightarrow S * S$

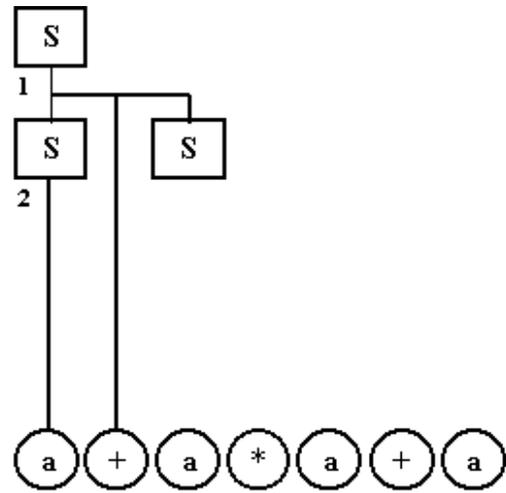
Цепочки, порождаемые данной грамматикой можно интерпретировать как выражения, состоящие из операндов "a", а также операций "+" и "*". Недетерминированность грамматики позволяет порождать одну и ту же терминальную цепочки с использованием различных выводов. Например, выражение "a+a*a+a" можно получить следующими способами:

1. $S \Rightarrow S+S \Rightarrow a+S \Rightarrow a+S*S \Rightarrow a+ a*S \Rightarrow a+a*S+S \Rightarrow a+a*a+S \Rightarrow a+a*a+a$
2. $S \Rightarrow S+S \Rightarrow S+a \Rightarrow S*S+a \Rightarrow S*a+a \Rightarrow S+S*a+a \Rightarrow S+a*a+a \Rightarrow a+a*a+a$ (6.1)
3. $S \Rightarrow S*S \Rightarrow S+S*S \Rightarrow S+S*S+S \Rightarrow a+ S*S+S \Rightarrow a+a*S+S \Rightarrow a+a*S+a \Rightarrow a+a*a+a$

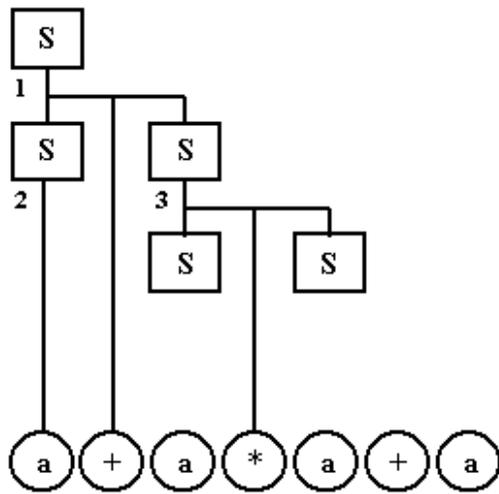
И так далее. В этом пример число вариантов одной и той же произвольной цепочки вывода настолько велико, что не имеет и смысла говорить о практическом применении данной грамматики. Но в данном случае она позволяет показать, каким образом могут порождаться различные деревья при нисходящем разборе. Пошаговое построение различных деревьев показано на рис. [6.2](#), [6.3](#), [6.4](#). Можно отметить, что процесс построения дерева совпадает с последовательностью шагов вывода входной цепочки.



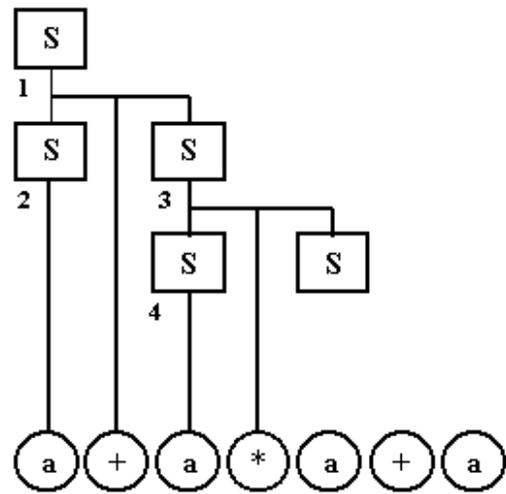
а) Шаг 1



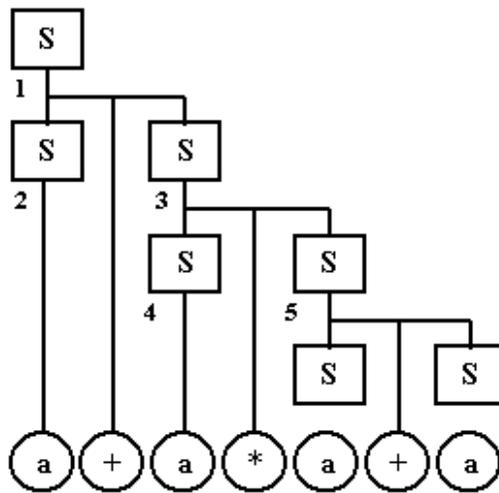
б) Шаг 2



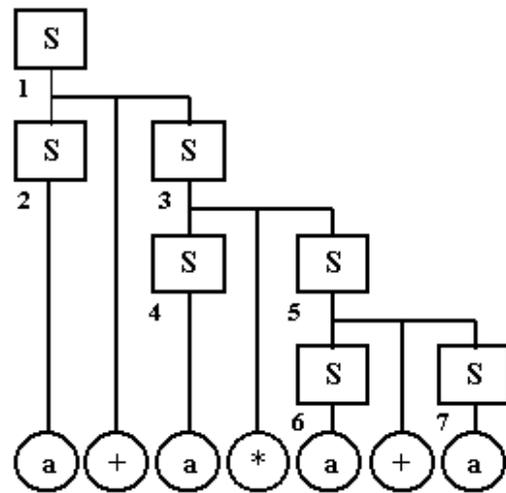
в) Шаг 3



г) Шаг 4



д) Шаг 5



е) Шаги 6-7

Рис. 6.2. Нисходящий разбор слева направо.

