Глава 4. Восходящий синтаксический анализ

4.5. *LR*(*k*)-грамматики

Наиболее общим случаем восходящего синтаксического анализа является *LR*-разбор, который применим к широкому классу *LR*(*k*)-грамматик.

LR(*k*)-*грамматикой* называется грамматика, при использовании которой всякая основа сентенциальной формы однозначно определяется (т. е. все конфликты типа «перенос/свертка» и «свертка/свертка» можно разрешать) на основании уже прочитанного текста (левый контекст) и фиксированного числа предварительно просматриваемых символов (максимум *k*). Аббревиатура *LR* означает «левосторонний ввод – правосторонний вывод», т. е. входная строка анализируется слева направо и используется правосторонняя схема вывода.

При *LR*-разборе всегда будем рассматривать пополненные грамматики, т. е. в КС-грамматике всегда предполагается наличие продукции вида $S' \to S\perp$, где S' и S – начальные нетерминалы пополненной и исходной грамматик соответственно, терминал $\perp$ – маркер конца ввода. При этом нетерминал S' не должен встречаться в правых частях других продукций.

Практический интерес представляют случаи $k = 0$ и $k = 1$. Рассмотрение даже двух предварительно просматриваемых символов делает *LR*-метод разбора довольно громоздким. Кроме того, из теории формальных языков и грамматик известно, что любой *LR*(k)-язык является также *LR*(1)-языком, т. е. может генерироваться *LR*(1)-грамматикой. Это означает, что для любой *LR*(k)-грамматики можно построить эквивалентную *LR*(1)-грамматику. Если *LR*(k)-язык удовлетворяет *условию собственности префиксов*, то он может генерироваться *LR*(0)-грамматикой. Это условие означает, что если x – строка языка, то никакой ее собственный префикс не принадлежит этому же языку. Поэтому, если используется маркер конца ввода $\perp$, язык удовлетворяет условию собственности префиксов и, следовательно, может генерироваться *LR*(0)-грамматикой.

Таким образом, в отличие от *LL*(k)-грамматик, где увеличение значения k позволяет представлять больший класс языков, в *LR*(k)-грамматиках этого не происходит. На практике обычно используются *LR*(1)-грамматики (или ее более простые подклассы), поскольку многие типичные свойства грамматик языков программирования относятся к *LR*(1)-, а не к *LR*(0)-свойствам, т. е. использование *LR*(1)- вместо *LR*(0)-грамматик позволяет избежать затруднений, связанных с преобразованием грамматик.

Основными достоинствами *LR*-разбора являются:

1) метод применим ко всем языкам, которые можно разбирать детерминированно, т. е. имеет универсальный характер и охватывает широкий класс языков и грамматик;

2) относительно редко возникает необходимость в преобразованиях грамматик;

3) имеются хорошие диагностические характеристики и возможность исправления ошибок.

Для разрешения конфликтов типа «перенос/свертка» и «свертка/свертка» *LR*-анализатор должен отслеживать, где именно он находится в процессе синтаксического анализа. Чтобы ссылаться на конкретную позицию в продукции грамматики, вводится понятие «пункт» (в литературе встречаются также термины «ситуация», «конфигурация»).

В общем случае *LR*(*k*)-*пункт* представляет собой пару $[A \to \alpha \bullet \beta, w]$, где $A \in V_N$, $\alpha, \beta \in (V_T \cup V_N)^*$, $w \in V_T^*$. Первый элемент пары – это продукция грамматики, в правой части которой перед одним из символов стоит маркерная точка, а второй элемент w – терминальная строка, которая может следовать за продукцией (т. е. за нетерминалом A). Строка w (ее длина не должна превышать k символов) называется *предпросмотром* или *контекстом* пункта. Очевидно, что в *LR*(1)-пункте $[A \to \alpha \bullet \beta, a]$ предпросмотр представляет собой единственный терминальный символ $a \in V_T$, а в *LR*(0) пункте $[A \to \alpha \bullet \beta]$ предпросмотр отсутствует.

Поскольку на практике рассматривают случаи $k = 0$ и $k = 1$, остановимся подробнее на $LR(1)$-пункте $[A \to \alpha\bullet\beta, a]$. Первый элемент $LR(1)$-пункта указывает, какая часть продукции уже исследована в данной точке синтаксическим анализатором. Например, продукция $A \to BCD$ дает четыре варианта расположения маркерной точки:

$A \to \bullet BCD$

$A \to B\bullet CD$

$A \to BC\bullet D$

$A \to BCD\bullet$

Первый вариант определяет, что во входном потоке ожидается строка, порождаемая BCD. Второй вариант указывает, что порожденная B строка уже исследована, а во входном потоке ожидается строка, порождаемая CD. Следует заметить, что продукция вида $A \to \varepsilon$ дает только один варианта расположения маркерной точки: $A \to \bullet$.

Что касается предпросмотра пункта, то он не играет никакой роли в *LR*(1)-пункте вида $[A \to \alpha\bullet\beta, a]$, где $\beta \neq \varepsilon$, а важен для *LR*(1)-пункта вида $[A \to \alpha\bullet, a]$ (этот пункт соответствует окончанию продукции $A \to \alpha$), когда выполняется свертка для продукции $A \to \alpha$ только тогда, когда очередным входным символом является терминал *a*. Другими словами, в *LR*(1)-пункте предпросмотр – это символ-следователь, который при разборе строки языка может оказаться предварительно просматриваемым символом в тех случаях, когда выполняется свертка в соответствии с порождающей продукцией.

Основой построения *LR*-анализатора является детерминированный конечный автомат, который используется для принятия решений в процессе синтаксического анализа. Такой автомат будем называть *LR-автоматом*. Чтобы найти все состояния *LR*-автомата, все возможные для заданной грамматики пункты по определенным правилам группируются в множества, соответствующие состояниям автомата. Пункты, которые неразличимы для анализатора, объединяются в одно состояние. Правила объединения пунктов зависят от используемого подкласса *LR*(1)-грамматик и будут изложены при рассмотрении соответствующих подклассов.