Глава 3. Нисходящий синтаксический анализ

3.4. *LL*(1)-грамматики

3.4.4. Основные приемы преобразования КС-грамматик в LL(1)-форму

Не существует полностью универсального автоматического процесса преобразования грамматик в $LL(1)$-форму. Это связано с тем, что данная проблема алгоритмически неразрешима. Отсутствие общего решения проблемы не означает невозможности ее решения для частных случаев.

Во многих случаях КС-грамматики, которые не обладают $LL(1)$-свойствами, можно преобразовать в $LL(1)$-грамматики с помощью левой факторизации.

Рассмотрим это преобразование на примере следующей грамматики:

$S \to aSb \mid aSc \mid \varepsilon$

Определим множества направляющих символов (учитывая, что грамматика пополнена маркером конца строки $\perp$, т. е. в предположении, что в грамматике имеется продукция $S' \to S\perp$):

$DS(S \to aSb) = \{a\}$,

$DS(S \to aSc) = \{a\}$,

$DS(S \to \varepsilon) = Follow(S) = \{b, c, \perp\}$.

Направляющий символ a является общим для двух первых продукций, т. е. это не $LL(1)$-грамматика. Применим для них правило левой факторизации (общим префиксом является aS). В результате получится $LL(1)$-грамматика

$S \to aSX \mid \varepsilon$

$X \to b \mid c$

Таким образом, факторизация как бы выносит за скобки направляющие символы.

Многие языки программирования содержат такие часто используемые конструкции, как списки. Поэтому рассмотрим описание различных типов списков с помощью $LL(1)$-грамматик.

Грамматика $L \to a\,\mathbf{;}\,L \mid a$ для порождения списка символов a, разделенных символом-разделителем (для определенности в качестве символа-разделителя используем точку с запятой), не обладает $LL(1)$-свойствами. Применив левую факторизацию, получим $LL(1)$-грамматику:

$L \to aR$

$R \to \mathbf{;}\,aR \mid \varepsilon$

Аналогично, применив левую факторизацию к продукциям $L \to aL \mid a$, порождающим список символов a без символов-разделителей, получим $LL(1)$-грамматику

$L \to aR$

$R \to aR \mid \varepsilon$

Рассмотренные грамматики не допускают пустой список. Если это необходимо, то достаточно добавить в грамматики продукцию $L \to \varepsilon$. Для списка из нуля или более символов a возможно применение грамматики $L \to aL \mid \varepsilon$, но иногда она может оказаться непригодной для трансляции.

Следует отметить, что грамматика, содержащая левую рекурсию, не является $LL(1)$-грамматикой. Рассмотрим продукции

$A \to A\alpha$ (леворекурсивная продукция по A),

$A \to a$

$DS(A \to A\alpha) = \{a\}$ и $DS(A \to a) = \{a\}$, т. е. множества направляющих символов пересекаются.

По тем же причинам грамматика, содержащая леворекурсивный цикл, не может быть $LL(1)$-грамматикой.

Как уже отмечалось, левую рекурсию всегда можно исключить из грамматики, преобразовав ее в правую, которая не вызывает никаких проблем для нисходящего разбора.

После устранения левой рекурсии для получения $LL(1)$-грамматики всегда требуется последующая левая факторизация. Можно сформулировать правило, объединяющее эти преобразования.

Пусть множество продукций содержит леворекурсивные продукции $A \to A\alpha_1 | A\alpha_2 | \ldots | A\alpha_m$ и все остальные A-продукции $A \to \beta_1 | \beta_2 | \ldots | \beta_n$, не являющиеся леворекурсивными. Результатом устранения левой рекурсии будут продукции

$A \to \beta_1 | \beta_2 | \ldots | \beta_n | \beta_1 A' | \beta_2 A' | \ldots | \beta_n A'$

$A' \to \alpha_1 | \alpha_2 | \ldots | \alpha_m | \alpha_1 A' | \alpha_2 A' | \ldots | \alpha_m A'$

Как видим, имеются продукции с общими префиксами, поэтому выполним левую факторизацию:

$A \to \beta_1 X | \beta_2 X | \ldots | \beta_n X$

$X \to \varepsilon | A'$

$A' \to \alpha_1 X | \alpha_2 X | \ldots | \alpha_m X$

Выполнив замену вхождений нетерминала A', получим окончательный результат:

$A \to \beta_1 X | \beta_2 X | \ldots | \beta_n X$

$X \to \varepsilon | \alpha_1 X | \alpha_2 X | \ldots | \alpha_m X$

Таким образом, модифицированное правило устранения левой рекурсии можно сформулировать следующим образом.

Пусть множество продукций содержит леворекурсивные продукции $A \to A\alpha_1 | A\alpha_2 | \dots | A\alpha_m$ и остальные A-продукции $A \to \beta_1 | \beta_2 | \dots | \beta_n$, не являющиеся леворекурсивными. Тогда новая эквивалентная грамматика будет иметь продукции вида:

$A \to \beta_1 X | \beta_2 X | \dots | \beta_n X$

$X \to \varepsilon | \alpha_1 X | \alpha_2 X | \dots | \alpha_m X$

Для иллюстрации преобразуем в $LL(1)$-форму следующую грамматику (предполагая наличие продукции $E' \to E\perp$):

$E \to E + T | T$

$T \to T \times F | F$

$F \to (E) | i$

Она не является $LL(1)$-грамматикой, поскольку содержит леворекурсивные продукции. Применим рассмотренное выше модифицированное правило устранения левой рекурсии:

$E \to TX$

$X \to +TX | \varepsilon$

$T \to FY$

$Y \to \times FY | \varepsilon$

$F \to (E) | i$

Это $LL(1)$-грамматика. Покажем это, определив множества направляющих символов для продукций с одинаковыми левыми частями:

$DS(X \to + TX) = \{+\}$,

$\mathrm{DS}(X \to \varepsilon) = \{),\ \perp\}$,

$DS(Y \to \times\ FY) = \{\times\}$,

$\mathrm{DS}(Y \to \varepsilon) = \{+,\ ),\ \perp\}$,

$DS(F \to (E)) = \{(\}$,

$\mathrm{DS}(F \to i) = \{i\}$.

Множества направляющих символов всех пар альтернативных продукций не пересекаются, что позволяет при разборе строки детерминированно выбирать нужную продукцию.

Процесс вывода строки $i\times(i+i)\perp$ соответствует следующей левосторонней схеме вывода:

$$
\begin{aligned}
&E' \Rightarrow E\perp \Rightarrow TX\perp \Rightarrow FYX\perp \Rightarrow iYX\perp \Rightarrow i\times FYX\perp \Rightarrow i\times(E)YX\perp \\
&\Rightarrow i\times(TX)YX\perp \Rightarrow i\times(FYX)YX\perp \Rightarrow i\times(iYX)YX\perp \Rightarrow i\times(iX)YX\perp \\
&\Rightarrow i\times(i+TX)YX\perp \Rightarrow i\times(i+FYX)YX\perp \Rightarrow i\times(i+iYX)YX\perp \\
&\Rightarrow i\times(i+iX)YX\perp \Rightarrow i\times(i+i)YX\perp \Rightarrow i\times(i+i)X\perp \Rightarrow i\times(i+i)\perp.
\end{aligned}
$$

К сожалению, процесс факторизации нельзя распространить на общий случай. Следующий пример показывает, что может произойти. Рассмотрим грамматику

$S \to Ac \mid Bd$

$A \to eAf \mid a$

$B \to eBg \mid b$

Первые две S-продукции в своих множествах направляющих символов содержат общий символ e. Для проведения факторизации предварительно выполним замену вхождений нетерминалов A и B, чтобы направляющие символы явно присутствовали в этих продукциях:

$S \to eAfc \mid ac \mid eBgd \mid bd$

Выполняя факторизацию, эти продукции можно заменить следующими:

$S \to ac \mid bd \mid eS_1$

$S_1 \to Afc \mid Bgd$

Продукции для S_1 аналогичны первоначальным продукциям для S и имеют пересекающиеся множества направляющих символов. Можно повторить преобразование этих продукций, как это было сделано с продукциями для S

$S_1 \to eAffc \mid afc \mid eBggd \mid bgd$

В результате факторизации получим

$S_1 \to afc \mid bgd \mid eS_2$

$S_2 \to Affc \mid Bggd$

Продукции для S_2 аналогичны продукциям для S_1 и S, но длиннее их, и теперь очевидно, что этот процесс бесконечен.

Это означает, что все попытки преобразовать грамматику в $LL(1)$-форму не всегда приводят к результату. Это прямое следствие того, что $LL(1)$-языки – только подклассом более широкого класса языков, допускающих детерминированный разбор.