Критерий применимости метода рекурсивного спуска

- Метод рекурсивного спуска применим к контекстно-свободной грамматике G, <u>если и только если</u> для любых двух её правил $X \to \alpha \mid \beta$ выполняются условия:

1. $first(\alpha) \cap first(\beta) = \varnothing$
2. Справедливо не более, чем одно из двух соотношений: $\alpha \Rightarrow \varepsilon$, $\beta \Rightarrow \varepsilon$
3. Если $\beta \Rightarrow \varepsilon$, то $first(X) \cap follow(X) = \varnothing$

Рекурсивный спуск

- Примени́м ли метод рекурсивного спуска к грамматике с перечисленными правилами? Ответ обосновать.
- $S \to A \mid B$ $\qquad A \to bA \mid \varepsilon$ $\qquad B \to cB \mid b \mid \varepsilon$
- Для символа S не выполняются первые два условия критерия применимости метода:
- S: $first\ (A) = \{ b \}$
- S: $first\ (B) = \{ b, c \}$ $\quad => \quad$ $\{ b \}$
- $S \to A \to \varepsilon$ $\qquad S \to B \to \varepsilon$
- Метод рекурсивного спуска к грамматике не примени́м

Рекурсивный спуск

- Примени́м ли метод рекурсивного спуска к грамматике с перечисленными правилами? Ответ обосновать.
- $S \rightarrow aASb \mid cfAd$ $\qquad A \rightarrow bA \mid c \mid \varepsilon$
- Имеется нетерминальный символ, из которого может быть выведена пустая цепочка, но первые два условия критерия применимости метода выполняются:
- *A: first (bA) = { b }* *first (c) = { c }* => $\varnothing$
- Цепочка ε выводится из *A* единственным образом
- *A: first (A) = { b, c }* *follow (A) = { a, c, d}* => *{ c }*
- Метод рекурсивного спуска к грамматике не примени́м

Рекурсивный спуск

- Примени́м ли метод рекурсивного спуска к грамматике с перечисленными правилами? Ответ обосновать.
- $S \to aA\{xx\}$ $\quad\quad A \to bA \mid cBx \mid \varepsilon$ $\quad\quad B \to bSc$
- Имеются нетерминальный символ, из которого может быть выведена пустая цепочка, и правило с итерацией, нужно проверять условия критерия применимости:
- A: $first\ (bA) = \{\ b\ \}$ $\quad first\ (cBx) = \{\ c\ \}$ $\quad => \varnothing$
- Цепочка ε выводится из A единственным образом
- A: $first\ (A) = \{\ b, c\ \}$ $\quad follow\ (A) = \{\ c, x\ \}$ $\quad => \{\ c\ \}$
- Метод рекурсивного спуска к грамматике не примени́м

Рекурсивный спуск

- Примени́м ли метод рекурсивного спуска к грамматике с перечисленными правилами? Ответ обосновать.
- $S \to aAbc \mid A \qquad A \to bB \mid cBc \qquad B \to bcB \mid a \mid \varepsilon$
- Правила грамматики не имеют канонического вида: для символа *S* одна из альтернатив начинается с нетерминального символа
- *S: first (aAbc) = { a }* *first (A) = { b, c }* *=>* $\varnothing$
- *A: first (bB) = { b }* *first (cBc) = { c }* *=>* $\varnothing$
- *B: first (bcB) = { b }* *first (a) = { a }* *follow (B) = { b, c }*
- Метод рекурсивного спуска к грамматике не примени́м

Пример преобразований грамматики

- Грамматика:

$$
\begin{aligned}
S &\rightarrow fASd \mid \varepsilon \\
A &\rightarrow Aa \mid Ab \mid dB \mid f \\
B &\rightarrow bcB \mid \varepsilon
\end{aligned}
$$

Устранение левой рекурсии

- Если в грамматике для цепочек есть нетерминальные символы, правила вывода которых леворекурсивны :
  $A \to A\alpha_1 \mid \ldots \mid A\alpha_n \mid \beta_1 \mid \ldots \mid \beta_m \quad \alpha_i \in (T \cup N)^+ \;\; i = 1,2,\ldots,n$
  $\beta_j \in (T \cup N)^* \;\; j = 1,2,\ldots,m$
  применять метод рекурсивного спуска нельзя
- Непосредственную левую рекурсию можно заменить правой (цепочки $\beta_j \{\alpha_i\}$):
  $A \to \beta_1 A' \mid \ldots \mid \beta_m A'$
  $A' \to \alpha_1 A' \mid \ldots \mid \alpha_n A' \mid \varepsilon$
- Если для символа есть одни лишь леворекурсивные правила (альтернативы β_j отсутствуют), то символ A' не вводится, а правила для символа A становятся такими:
  $A \to \alpha_1 A \mid \ldots \mid \alpha_n A \mid \varepsilon$

Пример преобразований грамматики

- Грамматика:

$$S \to fASd \mid \varepsilon$$
$$A \to Aa \mid Ab \mid dB \mid f$$
$$B \to bcB \mid \varepsilon$$

- Устранение левой рекурсии:

$$S \to fASd \mid \varepsilon$$
$$A \to dBA' \mid fA'$$
$$A' \to aA' \mid bA' \mid \varepsilon$$
$$B \to bcB \mid \varepsilon$$

из исходных правил:

A: $\alpha_1 = a$ $\alpha_2 = b$

A: $\beta_1 = dB$ $\beta_2 = f$

- Вычисление множеств *first* и *follow*:

$first\ (S) = \{f\}$ $follow\ (S) = \{d\}$ $\oplus$

$first\ (B) = \{b\}$ $follow\ (B) = \{a, b, f, d\}$ $\otimes$

ε-правила

- Грамматику с правилом, в котором для некоторого символа A имеется ε-альтернатива, можно преобразовать, введя символ A' $(A' \equiv A\beta)$:

$$A \to \alpha_1 A \mid \ldots \mid \alpha_n A \mid \beta_1 \mid \ldots \mid \beta_m \mid \varepsilon \qquad A, B \in N$$

$$B \to \alpha A \beta \qquad \alpha, \beta, \alpha_i, \beta_j \in (T \cup N)^*$$

$$first\ (A) \cap follow\ (A) \neq \varnothing$$

$$A \to \alpha_1 A \mid \ldots \mid \alpha_n A \mid \beta_1 \mid \ldots \mid \beta_m \mid \varepsilon$$

$$B \to \alpha A'$$

$$A' \to \alpha_1 A' \mid \ldots \mid \alpha_n A' \mid \beta_1 \beta \mid \ldots \mid \beta_m \beta \mid \beta$$

- Из B выводятся цепочки вида $\alpha\,\{\alpha_i\}\,\beta_j\,\beta$, либо $\alpha\,\{\alpha_i\}\,\beta$

Пример преобразований грамматики

- Удаление ε-альтернативы для *B* (*B' = BA'*):

$S \to fASd \mid \varepsilon$	*из исходных правил:*
$A \to dB' \mid fA'$	*A*: $\alpha = d$ $\beta = A'$
$A' \to \mathrm{a}A' \mid bA' \mid \varepsilon$	*B*: $\alpha_1 = bc$
$B' \to bcB' \mid A'$	*(B заменено на BA')*
$B \to bcB \mid \varepsilon$	*недостижимое правило*

Подстановка нетерминалов

- В грамматике есть нетерминальный символ, у которого несколько правил вывода, и среди них есть правила, начинающиеся нетерминальными символами:

$$A \to B_1\alpha_1 \mid \ldots \mid B_n\alpha_n \mid \mathrm{a}_1\beta_1 \mid \ldots \mid \mathrm{a}_m\beta_m$$

$$B_1 \to \gamma_{11} \mid \ldots \mid \gamma_{1k}$$

$$\ldots$$

$$B_n \to \gamma_{n1} \mid \ldots \mid \gamma_{np}$$

где $B_i \in N \quad \mathrm{a}_j \in T \quad \alpha_i, \beta_j \in (T \cup N)^* \quad \gamma_{ij} \in (T \cup N)^+$

- Можно заменить символы B_i их правилами:

$$A \to \gamma_{11}\alpha_1 \mid \ldots \mid \gamma_{1k}\alpha_1 \mid \ldots \mid \gamma_{n1}\alpha_n \mid \ldots \mid \gamma_{np}\alpha_n \mid \mathrm{a}_1\beta_1 \mid \ldots \mid \mathrm{a}_m\beta_m$$

Пример преобразований грамматики

- Удаление ε-альтернативы для B ($B' = BA'$):

$S \rightarrow fASd \mid \varepsilon$

$A \rightarrow dB' \mid fA'$

$A' \rightarrow aA' \mid bA' \mid \varepsilon$

$B' \rightarrow bcB' \mid A'$

из исходных правил:

A: $\alpha = d$ $\beta = A'$

B: $\alpha_1 = bc$

(B заменено на BA')

- Подстановка символа A' в правило для B' :

$S \rightarrow fASd \mid \varepsilon$

$A \rightarrow dB' \mid fA'$

$A' \rightarrow aA' \mid bA' \mid \varepsilon$

$B' \rightarrow bcB' \mid aA' \mid bA' \mid \varepsilon$

из исходных правил:

A': $\gamma_{11} = aA'$ $\gamma_{12} = bA'$ $\gamma_{13} = \varepsilon$

B': $\alpha_1 = \varepsilon$ $a_1 = b$ $\beta_1 = cB'$

Левая факторизация

- В грамматике есть правила, начинающиеся одинаковыми символами:

$$A \to \mathrm{a}\alpha_1 \mid \mathrm{a}\alpha_2 \mid \ldots \mid \mathrm{a}\alpha_n \mid \beta_1 \mid \ldots \mid \beta_m$$

где $\mathrm{a} \in (T \cup N)^+$; α_i, $\beta_j \in (T \cup N)^*$, β_j не начинается с ‘a’

- Можно объединить правила с общими началами в одно правило, введя новый символ A':

$$A \to \mathrm{a}A' \mid \beta_1 \mid \ldots \mid \beta_m$$
$$A' \to \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_n$$

Пример преобразований грамматики

- Левая факторизация правил для символа *B'*:

$S \rightarrow fASd \mid \varepsilon$

$A \rightarrow dB' \mid fA'$

$A' \rightarrow aA' \mid bA' \mid \varepsilon$

$B' \rightarrow aA' \mid bB'' \mid \varepsilon$

$B'' \rightarrow cB' \mid A'$

из исходных правил:

B': $\alpha_1 = cB'$ $\alpha_2 = A'$

B': $\beta_1 = aA'$ $\beta_2 = \varepsilon$

Пример преобразований грамматики

- Левая факторизация правил для символа *B′*:

$S \to fASd \mid \varepsilon$
$A \to dB' \mid fA'$
$A' \to aA' \mid bA' \mid \varepsilon$
$B' \to aA' \mid bB'' \mid \varepsilon$
$B'' \to cB' \mid A'$

из исходных правил:

B': $\alpha_1 = cB'$ $\quad \alpha_2 = A'$

B': $\beta_1 = aA'$ $\quad \beta_2 = \varepsilon$

- Вычисление множеств *first* и *follow*:

$first\,(A') = \{a, b\}$ $\quad follow\,(A') = \{f, d\}$ $\quad \oplus$

$first\,(B') = \{a, b\}$ $\quad follow\,(B') = \{f, d\}$ $\quad \oplus$

Пример преобразований грамматики

- Подстановка нетерминала *A’* в правило для *B’’*:

$$
\begin{aligned}
S &\to fASd \mid \varepsilon \\
A &\to dB' \mid fA' \\
A' &\to aA' \mid bA' \mid \varepsilon \\
B' &\to aA' \mid bB'' \mid \varepsilon \\
B'' &\to aA' \mid bA' \mid cB' \mid \varepsilon
\end{aligned}
$$

- Вычисление множеств *first* и *follow*:

$$first\,(B'') = \{a, b, c\} \qquad follow\,(B'') = \{f, d\} \qquad \oplus$$

- Грамматика имеет канонический вид

Последовательность применения преобразований грамматик

- Проблема “*Существует ли для произвольной контекстно-свободной грамматики эквивалентная грамматика, для которой метод рекурсивного спуска применим?*” **алгоритмически неразрешима**
- ***Шаг 1***. На основе множества нетерминальных символов N формируется множество обрабатываемых нетерминальных символов $V_{обр} = N$ и исключается левая рекурсия из всех правил для всех $A \in V_{обр}$ Формируются новые значения:

$$V_{обр} = V_{обр} \cup V_{лр} \qquad N = N \cup V_{лр}$$

Последовательность применения преобразований грамматик

- ***Шаг 2***. В случае необходимости подставляются нетерминальные символы для $A \in V_{обр}$, удаляются (возможно появляющиеся) недостижимые символы и правила:

$$V_{обр} = V_{обр} \setminus V_{не\ дост} \qquad N = N \setminus V_{не\ дост}$$

- ***Шаг 3***. Проводится левая факторизация для $A \in V_{обр}$, создаётся новое множество V_{new}:

$$V_{new} = V_{лф} \qquad N = N \cup V_{лф}$$

Последовательность применения преобразований грамматик

- ***Шаг 4***. Проводится обработка ε-правил для $A \in V_{обр}$, для которых это необходимо из-за правила $A \to \varepsilon$ и $first\ (A) \cap follow\ (A) \neq \varnothing$, после чего удаляются недостижимые символы и правила:

$$V_{new} = V_{new} \cup V_{\varepsilon} \qquad N = N \cup V_{\varepsilon} \qquad N = N \setminus V_{не\ дост}$$

- ***Шаг 5***. Если $V_{new} \neq \varnothing$, на его основе формируется множество обрабатываемых нетерминальных символов $V_{обр}$ и выполняется переход к ***шагу 2*** (или к ***шагу 1*** в случае глубокой неявной рекурсии):

$$V_{обр} = V_{new} \qquad \textit{go to } \textbf{Шаг 2}$$

Преобразование итераций

- Грамматика со списком элементов, ограниченных символом, совпадающим с внутренним разделителем элементов списка, как пример итерации в правилах:

$S \to LB$	$S \to LB$
$L \to a\ \{,\ a\}$	$L \to a\ \vert\ a\ ,\ L$
$B \to ,\ b$	$B \to ,\ b$

- Вводится дополнительный нетерминальный символ *L’*:

$S \to LB$

$L' \to ,\ L\ |\ \varepsilon$ $first\ (L') = \{,\}$

$L \to aL'$ $follow\ (L') = follow\ (L) = first\ (B) = \{,\}$

$B \to ,\ b$ $first\ (L') \cap follow\ (L') \neq \varnothing$

Преобразование итераций

- Подправляется правило для L' ($L' \to ,aL' \mid \varepsilon$):

$S \to LB$	*из исходных правил:*
$L'' \to ,aL'' \mid \varepsilon$	L (B): $\alpha = a \quad \beta = \varepsilon$
$L \to aL''$	L' (A): $\alpha_1 = ,a \quad \beta_i = \varepsilon$
$B \to , b$	
$\cancel{L' \to ,aL' \mid \varepsilon}$	*недостижимое правило*

- Очередное поколение правил в точности повторяет предыдущее, поэтому преобразования по показанным методикам не могут привести к получению грамматики, которые методом рекурсивного спуска смогут обрабатывать подобные списки