Лекции по теории формальных языков

Лекция 6.
Преобразования контекстно-свободных грамматик

Александр Сергеевич Герасимов
http://gas-teach.narod.ru

Кафедра математических и информационных технологий
Санкт-Петербургского академического университета
Российской академии наук.
Весенний семестр 2010/11 учебного года

18 марта 2011 г.

План

1. Неукорачивающие контекстно-свободные грамматики
2. Приведённые контекстно-свободные грамматики
3. Контекстно-свободные грамматики без циклов
4. Нелеворекурсивные контекстно-свободные грамматики (начало)

План

1. Неукорачивающие контекстно-свободные грамматики
2. Приведённые контекстно-свободные грамматики
3. Контекстно-свободные грамматики без циклов
4. Нелеворекурсивные контекстно-свободные грамматики (начало)

Определение неукорачивающей КС-грамматики

Далее под грамматикой по умолчанию понимаем КС-грамматику.

КС-грамматика с аксиомой S называется *неукорачивающей* (грамматикой *без ε-правил* или *ε-свободной* грамматикой), если либо

- в ней нет ε-правил, либо
- в ней есть единственное ε-правило $S \to \varepsilon$ и S не входит в правую чать никакого правила.

Алгоритм построения эквивалентной неукорачивающей КС-грамматики

Вход. КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход. Неукорачивающая КС-грамматика G', эквивалентная G.

Посредством $\alpha \preceq \beta$ будем записывать тот факт, что цепочка α является подпоследовательностью цепочки β, и все символы из β, не входящие в α, являются аннулирующими нетерминалами грамматики G.

1. **вычислить** $Ann(G)$;
2. $P' := \{A \to \alpha \mid \exists \beta \in (\Sigma \cup \Gamma)^* \ ((A \to \beta) \in P \wedge \varepsilon \neq \alpha \wedge \alpha \preceq \beta)\}$;
3. **если** $(S \notin Ann(G))$
4. $\qquad G' = (\Sigma, \Gamma, P', S)$;
5. **иначе** $P' := P' \cup \{S' \to S|\varepsilon\}$, где S' — новый нетерминал;
6. $\qquad G' := (\Sigma, \Gamma \cup \{S'\}, P', S')$;

В результате применения этого алгоритма число правил вывода грамматики может вырасти экспоненциально.

Пример построения эквивалентной неукорачивающей КС-грамматики

- Грамматика G_1:
  - $S \longrightarrow \mathtt{a}BC|AE$,
  - $A \longrightarrow \mathtt{b}CD|\varepsilon$,
  - $B \longrightarrow ACA$,
  - $C \longrightarrow \varepsilon$,
  - $E \longrightarrow CA$,
  - $D \longrightarrow \mathtt{b}ES|\mathtt{c}$.
- $Ann(G_1) = \{A, C, B, E, S\}$.
- Эквивалентная неукорачивающая грамматика G_1':
  - $S' \longrightarrow S|\varepsilon$,
  - $S \longrightarrow \mathtt{a}BC|\mathtt{a}B|\mathtt{a}C|\mathtt{a}|AE|A|E$,
  - $A \longrightarrow \mathtt{b}CD|\mathtt{b}D$,
  - $B \longrightarrow ACA|AC|CA|AA|A|C$,
  - $E \longrightarrow CA|C|A$,
  - $D \longrightarrow \mathtt{b}ES|\mathtt{b}E|\mathtt{b}S|\mathtt{b}|\mathtt{c}$.

Теорема о неукорачивающей КС-грамматике

Теорема

Для любой КС-грамматики G алгоритм на слайде 5 строит эквивалентную ей неукорачивающую КС-грамматику G'.

Доказательство.

- По построению G' является неукорачивающей КС-грамматикой.
- Ясно, что $\varepsilon \in L(G')$ тогда и только тогда, когда $\varepsilon \in L(G)$.
- Для доказательства эквивалентности G и G' достаточно показать, что для любой цепочки $w \in \Sigma^+$ имеет место следующее утверждение: $S \Rightarrow^*_{G'} w$ тогда и только тогда, когда $S \Rightarrow^*_G w$.
- При этом можно считать, что грамматика $G' = (\Sigma, \Gamma, P', S)$ определена в строке 4 рассматриваемого алгоритма.

Теорема о неукорачивающей КС-грамматике: продолжение доказательства

Докажем, что для любой $w \in \Sigma^+$, если $S \Rightarrow^*_{G'} w$, то $S \Rightarrow^*_G w$.

- По построению G' из $A \Rightarrow_{G'} \alpha$ следует, что существует такая цепочка β, что $A \Rightarrow_G \beta$ и $\alpha \preceq \beta$.
- $\alpha \preceq \beta$ влечёт $\beta \Rightarrow^*_G \alpha$. Тогда имеем $A \Rightarrow_G \beta \Rightarrow^*_G \alpha$.
- Таким образом, $A \Rightarrow_{G'} \alpha$ влечёт $A \Rightarrow^*_G \alpha$. Теперь индукцией по длине вывода легко показать, что если $\gamma_1 \Rightarrow^*_{G'} \gamma_2$, то $\gamma_1 \Rightarrow^*_G \gamma_2$.
- Отсюда получаем, что $S \Rightarrow^*_{G'} w$ достаточно для $S \Rightarrow^*_G w$.

Теорема о неукорачивающей КС-грамматике: продолжение доказательства

Докажем, что для любой $w \in \Sigma^+$, если $S \Rightarrow_G^* w$, то $S \Rightarrow_{G'}^* w$.

- Пусть $w \neq \varepsilon$ и $S \Rightarrow_G^* w$. Возьмём произвольное дерево вывода T цепочки w в грамматике G.
- Обозначим через T' дерево, полученное из T удалением каждого поддерева, все листья которого помечены ε (причём у которого имеется хотя бы один лист, помеченный ε).
- Достаточно показать, что T' — дерево вывода цепочки w в грамматике G'.
- Корень дерева T' помечен S, поскольку $w \neq \varepsilon$ и корень дерева T помечен S.
- w есть конкатенация всех перечисленных слева направо пометок листьев дерева T'.

Теорема о неукорачивающей КС-грамматике: окончание доказательства

- Рассмотрим какой угодно внутренний узел N дерева T'. Сыновья

$$X_1, X_2, \ldots, X_k$$

узла N в T' образуют подпоследовательность последовательности

$$X_{0,1}, \ldots, X_{0,m_0}, X_1, X_{1,1}, \ldots, X_{1,m_1}, X_2, \ldots, X_k, X_{k,1}, \ldots, X_{k,m_k}$$

его сыновей в T, причём каждый $X_{i,j} \in Ann(G)$. (Здесь мы отождествляем узлы и их метки.)

- Тогда

$$X_1 X_2 \ldots X_k \preceq X_{0,1} \ldots X_{0,m_0} X_1 X_{1,1} \ldots X_{1,m_1} X_2 \ldots X_k X_{k,1} \ldots X_{k,m_k}.$$

Значит, $N \to X_1 X_2 \ldots X_k \in P'$.

- Итак, T' — дерево вывода цепочки w в грамматике G'.

План

1. Неукорачивающие контекстно-свободные грамматики
2. Приведённые контекстно-свободные грамматики
3. Контекстно-свободные грамматики без циклов
4. Нелеворекурсивные контекстно-свободные грамматики (начало)

Производящие нетерминалы КС-грамматики

- С точки зрения порождения грамматикой языка, бесполезны нетерминалы,
  - из которых не выводима ни одна терминальная цепочка, а также
  - которые не входят ни в одну цепочку, выводимую из аксиомы.

  Научимся устранять бесполезные нетерминалы из грамматик.
- Нетерминал КС-грамматики назовём *производящим* (или *продуктивным*), если из него выводима некоторая цепочка терминалов.
- Пример. Грамматика G_2:
  $S \to \mathtt{a}SA\mathtt{c} | B\mathtt{a}C,\ A \to \mathtt{ab},\ B \to A\mathtt{d},\ C \to BD\mathtt{e}|\mathtt{c},\ D \to EC\mathtt{a},\ E \to AE.$
  - A, C — производящие, поскольку $A \Rightarrow \mathtt{ab}$, $C \Rightarrow \mathtt{c}$;
  - B — производящий, поскольку $B \Rightarrow A\mathtt{d} \Rightarrow^* u$;
  - S — производящий, поскольку $S \Rightarrow B\mathtt{a}C \Rightarrow^* v$;
  - D, E — непроизводящие.

Алгоритм нахождения производящих нетерминалов КС-грамматики

Вход. КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход. Множество Γ^p всех производящих нетерминалов грамматики G.

1. $i := 1$; $\Gamma_0 := \emptyset$; $\Gamma_1 := \{A \in \Gamma \mid \exists \alpha \in \Sigma^* \ ((A \to \alpha) \in P)\}$
2. **пока** $(\Gamma_{i-1} \neq \Gamma_i)$ **повторять**
3. $\quad\quad \Gamma_{i+1} := \{A \in \Gamma \mid \exists \alpha \in (\Sigma \cup \Gamma_i)^* \ ((A \to \alpha) \in P)\}$;
4. $\quad\quad i := i + 1$;
5. $\Gamma^p := \Gamma_i$

Пример работы этого алгоритма для грамматики G_2 с правилами $S \to \mathtt{a}SA\mathtt{c} | B\mathtt{a}C$, $A \to \mathtt{ab}$, $B \to A\mathtt{d}$, $C \to BD\mathtt{e} | \mathtt{c}$, $D \to EC\mathtt{a}$, $E \to AE$.

- $\Gamma_1 = \{A, C\}$;
- $\Gamma_2 = \{A, C, B\}$;
- $\Gamma_3 = \{A, C, B, S\} = \Gamma_4 = \Gamma^p$.

Корректность алгоритма нахождения производящих нетерминалов КС-грамматики

- Индукцией по i легко показать, что $\Gamma_{i-1} \subseteq \Gamma_i$ для любого $i \geqslant 1$.
- Алгоритм завершается, выполнив тело цикла не более $|\Gamma|$ раз, поскольку иначе выполнялось бы

$$\emptyset = \Gamma_0 \subsetneq \Gamma_1 \subsetneq \Gamma_2 \subsetneq \ldots \subsetneq \Gamma_{|\Gamma|} \subsetneq \Gamma_{|\Gamma|+1} \subseteq \Gamma$$

  и в $\Gamma_{|\Gamma|+1}$ было бы более $|\Gamma|$ нетерминалов.
- Далее, индукцией по i покажем, что $A \in \Gamma_i$ влечёт $A \Rightarrow^* w$ для некоторой цепочки $w \in \Sigma^*$. Поэтому все нетерминалы, возвращаемые алгоритмом, являются производящими.
- Заметим, что если $\Gamma_{i-1} = \Gamma_i$, то $\Gamma_{i-1} = \Gamma_j$ для любого $j \geqslant i$.
- Наконец, индукцией по длине вывода докажем, что $A \Rightarrow^* w$ влечёт $A \in \Gamma_i$ для некоторого i. Поэтому все производящие нетерминалы возвращаются алгоритмом.

Достижимые нетерминалы КС-грамматики

- Нетерминал КС-грамматики назовём *достижимым*, если он является аксиомой или входит в некоторую цепочку, выводимую из аксиомы.
- Пример. Грамматика G_3:
  $S \to \mathtt{b}A\mathtt{c}|\mathtt{c}B$, $A \to \mathtt{ab}$, $B \to E\mathtt{a}$, $C \to BD\mathtt{e}$, $D \to BC\mathtt{a}$, $E \to F\mathtt{b}$, $F \to \mathtt{a}$.
  - S — достижимый;
  - A, B — достижимые, так как S достижим и имеются правила $S \to \mathtt{b}A\mathtt{c}|\mathtt{c}B$;
  - E — достижимый, так как B достижим и имеется правило $B \to E\mathtt{a}$;
  - F — достижимый, так как E достижим и имеется правило $E \to F\mathtt{b}$;
  - C, D — недостижимые.

Алгоритм нахождения достижимых нетерминалов КС-грамматики

Вход. КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход. Множество Γ^r всех достижимых нетерминалов грамматики G.

1. $i := 1$; $\Gamma_0 := \emptyset$; $\Gamma_1 := \{S\}$;
2. **пока** $(\Gamma_{i-1} \neq \Gamma_i)$ **повторять**
3. $\quad\Gamma_{i+1} := \Gamma_i \cup \{B \in \Gamma \mid \exists A \in \Gamma_i,\ \alpha, \beta \in (\Sigma \cup \Gamma)^* \ ((A \to \alpha B \beta) \in P)\}$;
4. $\quad i := i + 1$;
5. $\Gamma^r := \Gamma_i$

Пример работы этого алгоритма для грамматики G_3 с правилами $S \to \mathtt{b}A\mathtt{c}|\mathtt{c}B$, $A \to \mathtt{ab}$, $B \to E\mathtt{a}$, $C \to BD\mathtt{e}$, $D \to BC\mathtt{a}$, $E \to F\mathtt{b}$, $F \to \mathtt{a}$.

- $\Gamma_1 = \{S\}$;
- $\Gamma_2 = \{S, A, B\}$;
- $\Gamma_3 = \{S, A, B, E\}$;
- $\Gamma_4 = \{S, A, B, E, F\} = \Gamma_5 = \Gamma^r$.

Определение приведённой КС-грамматики. Алгоритм построения эквивалентной приведённой КС-грамматики

КС-грамматику назовём *приведённой*, если каждый нетерминал этой грамматики является производящим и достижимым.

Вход. КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход. Приведённая КС-грамматика G', эквивалентная G.

1. **вычислить** Γ^p;
2. **если** $(S \notin \Gamma^p)$
3. $\quad G' := (\emptyset, \{S\}, \emptyset, S)$;
4. **иначе** $\widehat{P} := \{(A \to \alpha) \in P \mid A, \alpha \in (\Gamma^p \cup \Sigma)^*\}$;
5. $\quad \widehat{G} := (\Sigma, \Gamma^p, \widehat{P}, S)$;
6. $\quad$ **вычислить** $(\Gamma^p)^r$ в $\widehat{G}$;
7. $\quad P' := \{(A \to \alpha) \in \widehat{P} \mid A, \alpha \in ((\Gamma^p)^r \cup \Sigma)^*\}$;
8. $\quad G' := (\Sigma, (\Gamma^p)^r, P', S)$

Пример построения эквивалентной приведённой КС-грамматики

- Грамматика $G_4 = (\Sigma, \Gamma, P, S)$, где $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, $\Gamma = \{S, A, B\}$, множество правил P таково:
  - $S \to \mathtt{a} | A$,
  - $A \to AB$,
  - $B \to \mathtt{b}$.
- Эквивалентная приведённая грамматика G_4' получается так:
  - $\Gamma^p = \{S, B\}$;
  - $\widehat{G}_4$: $S \to \mathtt{a}$, $B \to \mathtt{b}$;
  - $(\Gamma^p)^r = \{S\}$;
  - G_4': $S \to \mathtt{a}$.
- Если же сначала найти достижимые нетерминалы, а затем производящие, то результат будет неверным:
  - $\Gamma^r = \{S, A, B\}$;
  - $\widehat{G}_4 = G$;
  - $(\Gamma^r)^p = \{S, B\}$;
  - G_4': $S \to \mathtt{a}$, $B \to \mathtt{b}$.

Теорема о приведённой КС-грамматике

Теорема

Для любой КС-грамматики G алгоритм на слайде 17 строит эквивалентную ей приведённую КС-грамматику G'.

Д о к а з а т е л ь с т в о.

- Эквивалентность грамматик G и G' очевидна.
- Чтобы установить приведённость грамматики G', достаточно показать, что каждый нетерминал $A \in (\Gamma^p)^r$ является производящим в G'.
- Каждый нетерминал $A \in (\Gamma^p)^r$ является
  - производящим в грамматиках G и $\widehat{G}$ и
  - достижимым в грамматиках $\widehat{G}$ и G'.
- Следовательно, существует вывод $S \Rightarrow^*_{\widehat{G}} \alpha A \beta \Rightarrow^*_{\widehat{G}} xwy$, где
  - $\alpha \Rightarrow^*_{\widehat{G}} x$, $\beta \Rightarrow^*_{\widehat{G}} y$,
  - $A \Rightarrow^*_{\widehat{G}} w$, причём в этот вывод не входят недостижимые нетерминалы грамматики $\widehat{G}$.
- Тогда $A \Rightarrow^*_{G'} w$ и нетерминал A является производящим в G'.

План

Циклы в КС-грамматиках: определения и причины появления

- Вывод $A \Rightarrow^{+} A$ называется *циклическим* выводом (или *циклом*).
- КС-грамматика называется *циклической* (или грамматикой *с циклами*), если в ней существует циклический вывод. В противном случае КС-грамматика называется *ациклической* (или грамматикой *без циклов*).
- Правило вывода вида $A \to B$ называется *цепным*.
- Циклы в КС-грамматике могут возникать лишь если в ней есть
  - цепные правила (например, в грамматике имеются правила $A \to B$, $B \to C$, $C \to A$ и цикл $A \Rightarrow B \Rightarrow C \Rightarrow A$) или
  - ε-правила (например, в грамматике есть правила $A \to BC$, $B \to CA$, $C \to \varepsilon$ и цикл $A \Rightarrow BC \Rightarrow CAC \Rightarrow AC \Rightarrow A$).
- Но мы можем преобразовать рассматриваемую КС-грамматику к эквивалентной неукорачивающей КС-грамматике.

Лемма об устранении цикла из КС-грамматики

Лемма

Пусть КС-грамматика G содержит цепные правила

$$A_1 \to A_2, \ \ldots, \ A_{n-1} \to A_n, \ A_n \to A_1. \qquad (\star)$$

Пусть также КС-грамматика G' получается из G заменой нетерминалов $A_2, \ldots, A_n$ всюду на нетерминал A_1 с последующим удалением правила $A_1 \to A_1$. Тогда G' эквивалентна G.

Доказательство.

- Пусть $S = \alpha_0 \Rightarrow_G \alpha_1 \Rightarrow_G \ldots \Rightarrow_G \alpha_m = w$, где S — аксиома грамматики G.
- Выделим все номера j переходов $\alpha_{j-1} \Rightarrow_G \alpha_j$, на которых не применяется правило из $(\star)$: $i_1 < \ldots < i_k = m$.
- Тогда $S' = \alpha'_{i_1} \Rightarrow_{G'} \ldots \Rightarrow_{G'} \alpha'_{i_k} = w$, где S' — аксиома грамматики G', α'_{i_l} получается из α_{i_l} заменой $A_2, \ldots, A_n$ на A_1.

Лемма: окончание доказательства

- Обратно, пусть $S' = \alpha_0 \Rightarrow_{G'} \alpha_1 \Rightarrow_{G'} \ldots \Rightarrow_{G'} \alpha_k = w$, где S' — аксиома грамматики G'.
- Если при переходе $\alpha_i \Rightarrow_{G'} \alpha_{i+1}$ применяется правило $B \to \beta$ грамматики G' и $B \neq A_1$, то
$$\alpha_i = \alpha_i' B \alpha_i'' \Rightarrow_G \alpha_i' \gamma \alpha_i'' \Rightarrow_G^* \alpha_i' \beta \alpha_i'' = \alpha_{i+1},$$
где $B \to \gamma$ — правило грамматики G, из которого получилось правило $B \to \beta$ грамматики G' в результате замены $A_2, \ldots, A_n$ на A_1.
- Если при переходе $\alpha_i \Rightarrow_{G'} \alpha_{i+1}$ применяется правило $A_1 \to \beta$ грамматики G', то
$$\alpha_i = \alpha_i' A_1 \alpha_i'' \Rightarrow_G^* \alpha_i' A_j \alpha_i'' \Rightarrow_G \alpha_i' \beta_j \alpha_i'' \Rightarrow_G^* \alpha_i' \beta \alpha_i'' = \alpha_{i+1},$$
где $A_j \to \beta_j$ — правило грамматики G, из которого получилось правило $A_1 \to \beta$ грамматики G'.
- Таким образом, существует вывод цепочки w из аксимомы в грамматике G.

Предложение о КС-грамматике без циклов

Предложение

Для любой КС-грамматики можно построить эквивалентную неукорачивающую КС-грамматику без циклов.

Д о к а з а т е л ь с т в о.

- Для исходой КС-грамматики строим эквивалентную неукорачивающую КС-грамматику.
- Затем применяем преобразование, описанное в предыдущей лемме, до устранения всех циклов.
- При этом преобразовании свойство КС-грамматики быть неукорачивающей сохраняется, и уменьшается число цепных правил, поэтому данный процесс конечен.

□

План

Левая рекурсия: определения

- Нетерминал B называется *леворекурсивным*, если $B \Rightarrow^+ B\gamma$, и *непосредственно леворекурсивным*, если $B \to B\gamma$ — правило вывода (такое правило вывода называется *леворекурсивным*).
- КС-грамматика называется *леворекурсивной*, если леворекурсивен хотя бы один её нетерминал. В противном случае КС-грамматика называется *нелеворекурсивной* (или грамматикой *без левой рекурсии*).
- *A-правилом* КС-грамматики называется правило вида $A \to \alpha$.

Лемма об устранении непосредственной левой рекурсии

Лемма

Пусть A — непосредственно леворекурсивный нетерминал грамматики G,

$$A \to A\alpha_1 | \ldots | A\alpha_k | \beta_1 | \ldots | \beta_m, \qquad (1)$$

— все A-правила грамматики G, причём ни одна из цепочек β_i не начинается с A. Тогда грамматика G', которая получается из G добавлением нового нетерминала A' и заменой правил (1) на правила

$$A \to \beta_1 A' | \ldots | \beta_m A', \quad A' \to \alpha_1 A' | \ldots | \alpha_k A' | \varepsilon, \qquad (2)$$

эквивалентна грамматике G.

Множество правил, полученное заменой правил (1) на правила (2) в множестве правил P, обозначим через $eilr(P, A)$.

Лемма об устранении непосредственной левой рекурсии: доказательство

Д о к а з а т е л ь с т в о.

- Применением правил (1) грамматики G из нетерминала A можно вывести в точности все цепочки вида $\beta_i \alpha_{j_1} \dots \alpha_{j_l}$, где $i \in \{1, \dots, m\}$, $l \geqslant 0$, $j_1, \dots, j_l \in \{1, \dots, k\}$.
- Применением правил (2) грамматики G' из нетерминала A можно вывести в точности такие же цепочки.
- Остальные правила грамматик G и G' и их аксиомы совпадают.

□

Пример устранения леворекурсивных правил из КС-грамматики

- Грамматика GA_2 (порождающая язык арифметических выражений LA) с множеством правил P:
  - $\underbrace{E}_{A} \to \underbrace{T}_{\beta_1} \mid \underbrace{E}_{A} \underbrace{+T}_{\alpha_1}$,
  - $T \to F \mid T * F$,
  - $F \to (E) \mid \mathrm{x}$.
- $P' = eilr(P, E)$:
  - $\underbrace{E}_{A} \to \underbrace{T}_{\beta_1} \underbrace{E'}_{A'}$, $\underbrace{E'}_{A'} \to \underbrace{+T}_{\alpha_1} \underbrace{E'}_{A'} |\varepsilon$,
  - $T \to F \mid T * F$,
  - $F \to (E) \mid \mathrm{x}$.
- Грамматика $GA_3 = eilr(P', T)$:
  - $E \to TE'$, $E' \to +TE'|\varepsilon$,
  - $T \to FT'$, $T' \to *FT'|\varepsilon$,
  - $F \to (E) \mid \mathrm{x}$.

Устранение правила из КС-грамматики

Лемма

Пусть КС-грамматика G содержит правило $A \to B\alpha$; $B \to \beta_1 | \ldots | \beta_k$ — все B-правила этой грамматики. Тогда грамматика G', которая получается из G удалением правила $A \to B\alpha$ и добавлением правил $A \to \beta_1\alpha | \ldots | \beta_k\alpha$, эквивалентна G.

Множество правил, полученное описанным в этой лемме преобразованием из множества правил P грамматики G, обозначим через $er(P,\ A \to B\alpha)$.

Алгоритм устранения левой рекурсии

Вход. Неукорачивающая КС-грамматика $G = (\Sigma, \Gamma, P, A_1)$ без правил вида $A \to A$, где $\Gamma = \{A_1, \ldots, A_n\}$.
Выход. Нелеворекурсивная КС-грамматика G', эквивалентная G.

(Для каждого правила вида $A_i \to A_j\alpha$ грамматики G' будет $i < j$.)

$P' := P$; $\Gamma' := \Gamma$; $i := 1$;
пока $i \leqslant n$ **повторять**
    $j := 1$;
    пока $(j < i)$ **повторять**
        для каждого $(A_i \to A_j\alpha) \in P'$ **выполнить**
            $P' := er(P',\ A_i \to A_j\alpha)$;
        $j := j + 1$;
    если $(\exists\beta\ ((A_i \to A_i\beta) \in P'))$
        $\Gamma' := \Gamma' \cup \{A_i'\}$;
        $P' := eilr(P', A_i)$;
    $i := i + 1$;
$G' := (\Sigma, \Gamma', P', A_1)$

Пример устранения левой рекурсии: начало

- Исходная грамматика:
  - $A_1 \to A_2A_3|\mathtt{a}$,
  - $A_2 \to A_3A_1|A_1\mathtt{b}$,
  - $A_3 \to A_1A_2|A_3A_3|\mathtt{a}$.
- $i = 1$: без изменений.
- $i = 2$, $j = 1$:
  - $A_1 \to A_2A_3|\mathtt{a}$,
  - $A_2 \to A_3A_1|\cancel{A_1\mathtt{b}}\underline{|A_2A_3\mathtt{b}|\mathtt{ab}}$,
  - $A_3 \to A_1A_2|A_3A_3|\mathtt{a}$.
- $i = 2$:
  - $A_1 \to A_2A_3|\mathtt{a}$,
  - $A_2 \to \cancel{A_3A_1|A_2A_3\mathtt{b}|\mathtt{ab}}\ \underline{A_3A_1A_2'|\mathtt{ab}A_2'}$, $\underline{A_2' \to A_3\mathtt{b}A_2'|\varepsilon}$,
  - $A_3 \to A_1A_2|A_3A_3|\mathtt{a}$.

Пример устранения левой рекурсии: окончание

- $i = 3$, $j = 1$:
  - $A_1 \to A_2A_3|\mathtt{a}$,
  - $A_2 \to A_3A_1A_2'|\mathtt{ab}A_2'$, $A_2' \to A_3\mathtt{b}A_2'|\varepsilon$,
  - $A_3 \to \cancel{A_1A_2}|A_3A_3|\mathtt{a}\underline{|A_2A_3A_2|\mathtt{a}A_2}$.
- $i = 3$, $j = 2$:
  - $A_1 \to A_2A_3|\mathtt{a}$,
  - $A_2 \to A_3A_1A_2'|\mathtt{ab}A_2'$, $A_2' \to A_3\mathtt{b}A_2'|\varepsilon$,
  - $A_3 \to A_3A_3|\mathtt{a}|\cancel{A_2A_3A_2}|\mathtt{a}A_2\underline{|A_3A_1A_2'A_3A_2|\mathtt{ab}A_2'A_3A_2}$.
- $i = 3$:
  - $A_1 \to A_2A_3|\mathtt{a}$,
  - $A_2 \to A_3A_1A_2'|\mathtt{ab}A_2'$, $A_2' \to A_3\mathtt{b}A_2'|\varepsilon$,
  - $A_3 \to \cancel{A_3A_3|\mathtt{a}|\mathtt{a}A_2|A_3A_1A_2'A_3A_2|\mathtt{ab}A_2'A_3A_2}\ \underline{\mathtt{a}A_3'|\mathtt{a}A_2A_3'|\mathtt{ab}A_2'A_3A_2A_3'}$,
    $\underline{A_3' \to A_3A_3'|A_1A_2'A_3A_2A_3'|\varepsilon}$.

Литература

Основная литература

- Замятин А. П., Шур А. М. Языки, грамматики, распознаватели: Учебное пособие. Екатеринбург : Изд-во Урал. ун-та, 2007 (электронный вариант книги — на http://elar.usu.ru, поиск).

Дополнительная литература

- Ахо А., Лам М., Сети Р., Ульман Дж. Компиляторы: принципы, технологии и инструментарий. М.: ООО "И.Д. Вильямс", 2008.
- Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. М.: Мир, 1978.
- Мартыненко Б. К. Языки и трансляции: Учеб. пособие. СПб.: Издательство С.-Петербургского университета, 2004 (электронный вариант книги — на http://www.math.spbu.ru/user/mbk).