А. А. Вылиток

Алгоритмы преобразования контекстно-свободных грамматик с помощью графов

1. Устранение бесполезных символов

Рассмотрим пример контекстно-свободной грамматики с алфавитом терминальных символов $\{a, b\}$ и начальным символом S:

$S \to aS \mid A \mid a$

$B \to b$

Нетрудно заметить, что для данной грамматики вспомогательные (нетерминальные) символы A и B не могут встречаться в сентенциальных формах выводов терминальных цепочек из начального символа S. Другими словами, они не принимают участия в порождении цепочек языка, то есть являются в этом смысле бесполезными.

Любую контекстно-свободную грамматику можно привести к форме, не содержащей бесполезных символов.

Пусть $G = (V_T, V_N, P, S)$ — контекстно-свободная грамматика (КС-грамматика).

Символ $x \in (V_T \cup V_N)$ называется *недостижимым* в грамматике G, если он не появляется ни в одной сентенциальной форме этой грамматики.

Символ $A \in V_N$ называется *бесплодным* в грамматике G, если множество выводимых из этого символа терминальных цепочек пусто.

КС-грамматика называется *приведенной*, если в ней нет недостижимых и бесплодных символов.

1.1. Алгоритм приведения контекстно-свободной грамматики к форме, не содержащей бесполезных символов

Алгоритм приведения состоит из двух шагов. Каждый шаг в свою очередь реализуется отдельным алгоритмом. Эти алгоритмы используют граф грамматики и будут рассмотрены ниже.

Алгоритм приведения КС-грамматики

1. Найти и удалить все бесплодные символы и правила, их содержащие.
2. Найти и удалить все недостижимые символы и правила, их содержащие.

После первого шага данного алгоритма для грамматики $G = (V_T, V_N, P, S)$ получаем эквивалентную грамматику $G_1 = (V_T, V_{N_1}, P_1, S)$, такую, что для любого $A \in V_{N_1}$ справедливо неравенство[1)]: $\{A \Rightarrow^* \omega \mid \omega \in V_T^*\} \neq \varnothing$.

---

1) Исключение может быть сделано для начального символа S грамматики G, если он окажется бесплодным (т.е. грамматика G порождает пустой язык). Тогда на первом шаге следует включить S в алфавит V_{N_1}, так как согласно определению грамматики, начальный символ является ее обязательной частью и принадлежит алфавиту нетерминальных символов. Однако правила, содержащие бесплодный символ S, в множество P_1 не включаются.

На втором шаге из G_1 получим эквивалентную грамматику $G_2 = (V_{T_2}, V_{N_2}, P_2, S)$, обладающую свойствами приведенной грамматики[2]:

(1) для любого символа $x \in (V_{T_2} \cup V_{N_2})$ существуют $\alpha_1, \alpha_2 \in (V_{T_2} \cup V_{N_2})^*$ такие, что $S \Rightarrow^* \alpha_1 x \alpha_2$;

(2) либо для любого $A \in V_{N_2}$ справедливо неравенство $\{A \Rightarrow^* \omega \mid \omega \in V_{T_2}{}^*\} \neq \varnothing$ либо $V_{N_2} = \{S\}$ и $\{S \Rightarrow^* \omega \mid \omega \in V_{T_2}{}^*\} = \varnothing$.

1.2. Граф контекстно-свободной грамматики

Для нахождения бесплодных и недостижимых символов полезен граф КС-грамматики:

- каждому символу из $V_T \cup V_N$ соответствует единственная вершина, помеченная этим символом; если в P есть правило с пустой правой частью ε, то граф имеет вершину, помеченную ε;
- вершина, помеченная символом X, соединяется с вершиной Y дугой (стрелкой), если в грамматике есть правило $X \to \alpha Y \beta$, где $\alpha, \beta \in (V_T \cup V_N)^*$;
- X соединяется с вершиной ε, если в грамматике есть правило $X \to \varepsilon$.

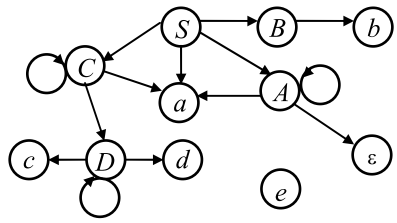
Пример. Грамматика G_0:

$G_0 = (\{a, b, c, d, e\}, \{S, A, B, C, D\}, P_0, S)$

P_0:

$S \to aAB \mid C$
$D \to cDc \mid d$
$C \to aCD$
$A \to aA \mid a \mid \varepsilon$
$B \to b$

Граф грамматики G_0:



1.3. Алгоритм удаления бесплодных символов

Для заданной грамматики G построить граф и выполнить следующие шаги:

1. Отметить (выделить) вершины графа, помеченные терминальными символами, а также вершину ε, если такая имеется.
2. Если в P есть правило $A \to \alpha$, где α состоит из символов уже отмеченных вершин, а вершина A еще не отмечена, то отметить эту вершину. Повторять шаг 2 пока возможно.
3. Из грамматики удалить символы неотмеченных вершин, а также правила, содержащие хотя бы один символ неотмеченной вершины.

Пример. Рассмотрим грамматику G_0 из п. 1.2.

$G_0 = (\{a, b, c, d, e\}, \{S, A, B, C, D\}, P_0, S)$

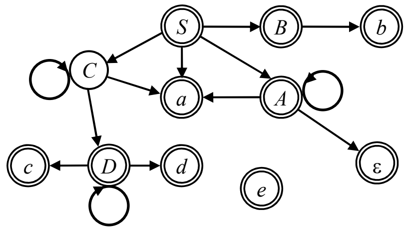
P_0:

$S \to aAB \mid C$
$D \to cDc \mid d$
$C \to aCD$
$A \to aA \mid a \mid \varepsilon$
$B \to b$

Граф G_0 после шагов 1, 2 алгоритма:
(отмеченные вершины выделены двойным кружком)

---

2) Может оказаться, что все терминальные символы недостижимы. Тогда хотя бы один терминальный символ должен быть оставлен в алфавите терминалов, так как алфавит не может быть пустым.



Неотмеченной на графе оказалась вершина C. Вычеркиваем в грамматике G_0 символ C и правила, его содержащие: $(\{a, b, c, d, e\}, \{S, A, B, \cancel{C}, D\}, P_0, S)$

P_0: $S \to aAB \cancel{\mid C}$
$D \to cDc \mid d$
$\cancel{C \to aCD}$
$A \to aA \mid a \mid \varepsilon$
$B \to b$

Получаем эквивалентную грамматику G_1, не содержащую бесплодных символов:

$G_1 = (\{a, b, c, d\}, \{S, A, B, D\}, P_1, S)$

P_1: $S \to aAB$
$D \to cDc \mid d$
$A \to aA \mid a \mid \varepsilon$
$B \to b$

1.4. Алгоритм удаления недостижимых символов

Построить граф грамматики. Используя граф, выполнить следующие шаги:

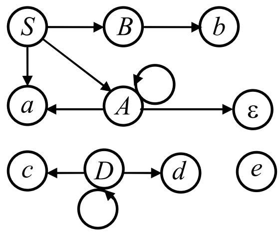
1. Отметить вершины графа, в которые есть путь из вершины S.
2. Удалить из грамматики символы неотмеченных вершин и правила, содержащие хотя бы один такой символ.

Пример. Рассмотрим грамматику G_1, полученную в п. 1.3.

$G_1 = (\{a, b, c, d, e\}, \{S, A, B, D\}, P_1, S)$

P_1: $S \to aAB$
$D \to cDc \mid d$
$A \to aA \mid a \mid \varepsilon$
$B \to b$

Граф грамматики G_1:



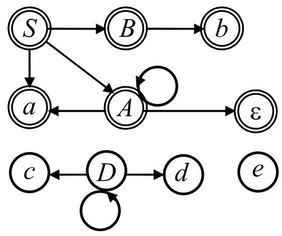
Находим недостижимые символы и вычеркиваем их из алфавита нетерминалов и правил грамматики G_1.

$G_1 = (\{a, b \cancel{, c, d, e}\}, \{S, A, B \cancel{, D}\}, P_1, S)$

P_1: $S \to aAB$
$\cancel{D \to cDc \mid d}$
$A \to aA \mid a \mid \varepsilon$

Граф грамматики G_1:
(отмеченные на шаге 1 вершины выделены двойным кружком)

$B \to b$



Неотмеченные в графе символы являются недостижимыми и подлежат удалению.

После удаления недостижимых символов из G_1, получим эквивалентную грамматику

$G_2 = (\{a, b\}, \{S, A, B\}, P_2, S)$

P_2: $S \to aAB$
$A \to aA \mid a \mid \varepsilon$
$B \to b$

Грамматика G_2 — приведенная, она получена из G_0 последовательным применением шагов 1 и 2 алгоритма приведения грамматики (п. 1.1)

$$L(G_0) = L(G_1) = L(G_2) = \{a^n b \mid n \geq 1\}.$$

1.5. О последовательности шагов в алгоритме приведения

Порядок шагов 1 и 2 в алгоритме приведения КС-грамматики важен. Их нельзя поменять местами.

Задача. Показать, что если к грамматике G_0 из п. 1.2 сначала применить алгоритм удаления недостижимых символов, а затем к результату применить алгоритм удаления бесплодных символов, то полученная таким способом грамматика не будет приведенной.

2. Устранение правил с пустой правой частью

Некоторые применяемые на практике алгоритмы разбора по КС-грамматикам требуют, чтобы в грамматиках не было правил с пустой правой частью.

Любую КС-грамматику, не порождающую пустую цепочку, можно преобразовать в эквивалентную, не имеющую правил с пустыми правыми частями. Если исходная грамматика порождает пустую цепочку, то по ней можно построить эквивалентную грамматику, содержащую единственное правило с пустой правой частью: $S \to \varepsilon$, где S — начальный символ; правые части остальных правил непустые и не содержат S.

Ниже приводится алгоритм, позволяющий преобразовать грамматику в описанную форму. На первом шаге алгоритма строится множество нетерминалов грамматики, из которых выводима пустая цепочка. Построение этого множества можно провести с помощью графа грамматики аналогично построению множества бесплодных символов (п. 1.3), отметив вначале лишь вершину, помеченную символом ε.

Пусть $G = (V_T, V_N, P, S)$ — КС-грамматика.

Алгоритм устранения правил с пустой правой частью

1. Построить множество $X = \{A \in V_N \mid A \Rightarrow^* \varepsilon\}$.
2. Удалить из множества правил P все правила с пустой правой частью.

3. Если $S \in X$, то ввести новый начальный символ S', в множество правил P добавить правило $S' \to S \mid \varepsilon$.
4. Для любого $A \in X$ правило вида $B \to \alpha_1 A \alpha_2 A ... \alpha_n A \alpha_{n+1}$, где $\alpha_i \in \big((V_N - \{A\}) \cup V_T\big)^*$, заменить 2^n правилами, соответствующими всем возможным комбинациям вхождений A между α_i:
   $B \to \alpha_1 \alpha_2 ... \alpha_n \alpha_{n+1}$
   $B \to \alpha_1 \alpha_2 ... \alpha_n A \alpha_{n+1}$
   …
   $B \to \alpha_1 \alpha_2 A ... \alpha_n A \alpha_{n+1}$
   $B \to \alpha_1 A \alpha_2 A ... \alpha_n A \alpha_{n+1}$
   Замечание: если $\alpha_i = \varepsilon$ для всех $i = 1, …, n + 1$, то получившееся на данном шаге правило $B \to \varepsilon$ не включать в множество P.
5. Удалить бесполезные символы и правила, их содержащие.

Пример. Исходная грамматика:

$G = (\{a, b, c\}, \{S, A, B, C\}, P, S)$

P:
$S \to BC \mid Ab$
$B \to \varepsilon$
$C \to c$
$A \to Aa \mid \varepsilon$

Эквивалентная грамматика, не содержащая правил с пустой правой частью:

$G' = (\{a, b, c\}, \{S, A, C\}, P', S)$

P':
$S \to C \mid b \mid Ab$
$C \to c$
$A \to Aa \mid a$