

Преобразования КС-грамматик

Основные понятия

Грамматики G_i и G_j называются *эквивалентными*, если равны порожаемые ими языки. Получение грамматики G_j , эквивалентной исходной грамматике G_i , назовём *эквивалентным преобразованием грамматики G_i в G_j* . Преобразование грамматики G_i в G_j выполняется применением определённых правил преобразования. Множество правил образуют *систему преобразований*. Система преобразований называется *полной*, если для любых двух эквивалентных грамматик G_i и G_j существует последовательность правил из заданной системы преобразований, результатом применения которой к грамматике G_i является грамматика G_j . Для КС-грамматик не существует полной системы преобразований, следовательно, рассматриваемые ниже правила преобразований не образуют полную систему преобразований, но обычно позволяют получить грамматику, эквивалентную исходной, обладающую заданными свойствами.

Устранение лишних символов

Среди символов КС-грамматики можно выделить две группы лишних символов:

1) *бесплодные нетерминалы* – это нетерминалы, которые не порождают ни одной терминальной цепочки. Такие нетерминалы не могут участвовать в выводе терминальных цепочек.

2) *недостижимые символы* – это символы (терминалы и нетерминалы), которые не встречаются ни в одной цепочки (терминальной или промежуточной), выводимой из начального нетерминала.

Правила, в которые входят лишние символы (в левую или правую часть правила), можно исключить из множества правил грамматики.

Поиск бесплодных нетерминалов сводится к нахождению продуктивных нетерминалов и исключению их из множества всех нетерминалов грамматики.

Терминальный или нетерминальный символ называется *продуктивным (живым)*, если из него выводится терминальная цепочка.

Любой терминальный символ продуктивный.

Нетерминальный символ A будет продуктивным, если существует правило $A \rightarrow \alpha$, в котором все символы в правой части продуктивны.

Алгоритм нахождения всех продуктивных нетерминалов.

1. Принять множество продуктивных нетерминалов $P = \emptyset$.
2. Если существует правило $A \rightarrow \alpha$, в котором все символы в правой части продуктивны, то нетерминал A включить в множество P .
3. Повторять п.2, пока множество P растёт.

Поиск недостижимых символов сводится к нахождению всех достижимых символов и исключению их из множества всех нетерминалов грамматики.

Терминальный или нетерминальный символ называется *достижимым*, если он может появиться в какой-нибудь цепочке, выводимой из начального нетерминала.

Начальный нетерминал - достижимый.

Терминальный или нетерминальный символ будет достижимым, если он находится в правой части правила $A \rightarrow \alpha$ и A – достижимый нетерминал.

Алгоритм нахождения всех достижимых символов.

1. Принять множество достижимых символов $P = \{S\}$, где S – начальный нетерминал.
2. Если существует правило $A \rightarrow \alpha$ и нетерминал A принадлежит множеству P , то все символы правой части включить в множество P .
3. Повторять п.2, пока множество P растёт.

Алгоритм устранения всех лишних символов.

1. В исходной грамматике G найти все бесплодные нетерминалы и исключить правила, связанные с ними. В результате получим грамматику G' без бесплодных нетерминалов.

2. В грамматике G' найти все недостижимые символы и исключить правила, связанные с ними. В результате получим грамматику G'' без лишних символов.

После исключения бесплодных нетерминалов в грамматике могут появиться недостижимые символы, которые в исходной грамматике были достижимыми. Если же в грамматике нет бесплодных нетерминалов, то они не могут появиться в результате исключения недостижимых символов.

Пример устранения лишних символов.

Грамматика:

- | | |
|------------------------|------------------------|
| 1. $S \rightarrow ac$ | 4. $B \rightarrow aSA$ |
| 2. $S \rightarrow bA$ | 5. $C \rightarrow bC$ |
| 3. $A \rightarrow cBC$ | 6. $C \rightarrow d$ |

1. Поиск продуктивных нетерминалов.

В множество продуктивных нетерминалов P включаем нетерминал S (правило 1) и нетерминал C (правило 6). Получаем $P = \{S, C\}$ и увеличить множество P не можем.

2. Поиск бесплодных нетерминалов.

Из множества всех нетерминалов исключаем все продуктивные и получаем множество $\{A, B\}$ бесплодных нетерминалов.

3. Исключение бесплодных нетерминалов.

Исключаем правила 2, 3 и 4, т.к. они содержат бесплодные нетерминалы. Получаем грамматику:

- | |
|-----------------------|
| 1. $S \rightarrow ac$ |
| 5. $C \rightarrow bC$ |
| 6. $C \rightarrow d$ |

4. Поиск достижимых символов.

Достижимыми символами являются S , a и c .

5. Поиск недостижимых символов.

Из множества всех символов исключаем все достижимые и получаем множество $\{C, b, d\}$ недостижимых символов.

6. Исключение недостижимых символов.

Исключаем правила 5 и 6, т.к. они содержат недостижимые символы. Получаем грамматику:

- | |
|-----------------------|
| 1. $S \rightarrow ac$ |
|-----------------------|

Заметим, что в исходной грамматике все символы достижимы.

Исключение лишних правил

Правило $A \rightarrow \alpha$ назовём *лишним*, если в грамматике существует вывод цепочки α из нетерминала A без участия этого правила.

Лишние правила можно исключить из множества правил грамматики.

Пример.

Исключить лишние правила из грамматики:

1. $A \rightarrow aABa$
2. $A \rightarrow bC$
3. $A \rightarrow bba$
4. $B \rightarrow dBC$
5. $B \rightarrow \varepsilon$
6. $C \rightarrow BA$
7. $C \rightarrow a$

В этой грамматике правило 3. $A \rightarrow bba$ лишнее, т.к. существует вывод цепочки bba из нетерминала A без участия этого правила:

$$\begin{array}{ccccccc} A & \Rightarrow & bC & \Rightarrow & bBa & \Rightarrow & bA & \Rightarrow & bbC & \Rightarrow & bba \\ 1 & & 6 & & 5 & & 2 & & 7 & & \end{array}$$

Других лишних правил в грамматике нет. Исключая третье правило получаем грамматику без лишних правил:

1. $A \rightarrow aABa$
2. $A \rightarrow bC$
4. $B \rightarrow dBC$
5. $B \rightarrow \varepsilon$
6. $C \rightarrow BA$
7. $C \rightarrow a$

Исключение ε -правил

Правило вида $A \rightarrow \varepsilon$ называется *ε -правилом*. Грамматику, порождающую язык, не содержащий пустую цепочку, можно преобразовать в эквивалентную ей грамматику без ε -правил, а грамматику, порождающую язык, содержащий пустую цепочку, можно преобразовать в эквивалентную ей грамматику с единственным ε -правилом $S \rightarrow \varepsilon$, где S – начальный нетерминал. Рассмотрим два алгоритма исключения ε -правил. В первом алгоритме используется понятие *аннулирующий нетерминал* – нетерминал, который может породить пустую цепочку. Для нахождения всех аннулирующих нетерминалов грамматики можно использовать следующий алгоритм:

1. Из множества правил грамматики исключить все правила, содержащие хотя бы один терминал в правой части (ε -правила не исключаются), т.к. они не могут участвовать в выводе пустой цепочки.

2. В полученной грамматике найти множество всех продуктивных нетерминалов. Оно так же является множеством всех аннулирующих нетерминалов в исходной грамматике.

Алгоритм 1 исключения ε -правил.

1. Найти множество всех аннулирующих нетерминалов.

2. Заменить каждое из правил, правые части которых содержат хотя бы по одному аннулирующему нетерминалу, множеством новых правил. Если правая часть правила содержит k вхождений аннулирующих нетерминалов, то множество, заменяющее это правило, состоит из 2^k правил, соответствующим всем возможным способам удаления некоторых (или всех) из этих вхождений.

3. Исключить из множества правил грамматики
все ε -правила
и правила вида $A \rightarrow A$.

Если в результате выполнения п.2 получены множества одинаковых правил, то из каждого такого множества оставить только одно.

4. Если исходная грамматика порождает пустую цепочку, то добавить правило $S \rightarrow \varepsilon$, где S – начальный нетерминал.

Пример.

Грамматика:

- | | |
|------------------------|--------------------------------|
| 1. $S \rightarrow AaB$ | 6. $A \rightarrow b$ |
| 2. $S \rightarrow aB$ | 7. $B \rightarrow Ba$ |
| 3. $S \rightarrow cC$ | 8. $B \rightarrow \varepsilon$ |
| 4. $A \rightarrow AB$ | 9. $C \rightarrow AB$ |
| 5. $A \rightarrow B$ | 10. $C \rightarrow c$ |

1. Находим множество аннулирующих нетерминалов. Исключая правила, содержащие хотя бы один терминал в правой части, получим грамматику:

- 4. $A \rightarrow AB$
- 5. $A \rightarrow B$
- 8. $B \rightarrow \varepsilon$
- 9. $C \rightarrow AB$

В этой грамматике все нетерминалы продуктивные, следовательно $\{A, B, C\}$ – множество аннулирующих нетерминалов.

2. Исключаем из каждого правила исходной грамматики всеми возможными способами аннулирующие нетерминалы, полученные правила добавляем в множество правил грамматики.

- | | | |
|--------------------------|----------------------------------|----------------------------------|
| 1_1. $S \rightarrow AaB$ | 4_1. $A \rightarrow AB$ | 7_1. $B \rightarrow Ba$ |
| 1_2. $S \rightarrow Aa$ | 4_2. $A \rightarrow A$ | 7_2. $B \rightarrow a$ |
| 1_3. $S \rightarrow aB$ | 4_3. $A \rightarrow B$ | 8_1. $B \rightarrow \varepsilon$ |
| 1_4. $S \rightarrow a$ | 4_4. $A \rightarrow \varepsilon$ | 9_1. $C \rightarrow AB$ |
| 2_1. $S \rightarrow aB$ | 5_1. $A \rightarrow B$ | 9_2. $C \rightarrow A$ |
| 2_2. $S \rightarrow a$ | 5_2. $A \rightarrow \varepsilon$ | 9_3. $C \rightarrow B$ |
| 3_1. $S \rightarrow cC$ | 6_1. $A \rightarrow b$ | 9_4. $C \rightarrow \varepsilon$ |
| 3_2. $S \rightarrow c$ | | 10_1. $C \rightarrow c$ |

3. Исключаем из множества правил грамматики все ε -правила, правила вида $A \rightarrow A$ и из повторяющихся оставляем только одно.

- | | | |
|--------------------------|-------------------------|-------------------------|
| 1_1. $S \rightarrow AaB$ | 4_1. $A \rightarrow AB$ | 9_1. $C \rightarrow AB$ |
| 1_2. $S \rightarrow Aa$ | 4_3. $A \rightarrow B$ | 9_2. $C \rightarrow A$ |
| 1_3. $S \rightarrow aB$ | 6_1. $A \rightarrow b$ | 9_3. $C \rightarrow B$ |
| 1_4. $S \rightarrow a$ | 7_1. $B \rightarrow Ba$ | 10_1. $C \rightarrow c$ |
| 3_1. $S \rightarrow cC$ | 7_2. $B \rightarrow a$ | |
| 3_2. $S \rightarrow c$ | | |

Алгоритм 2 исключения ϵ -правил.

Пока в грамматике есть ϵ -правила, выполнять п.1, 2 и 3.

1. Выбрать ϵ -правило $A \rightarrow \epsilon$.

2. Заменить каждое из правил, правые части которых содержат хотя бы один нетерминал A , множеством новых правил. Если правая часть правила содержит k вхождений нетерминала A , то множество, заменяющее это правило, состоит из 2^k правил, соответствующим всем возможным способам удаления некоторых (или всех) из этих вхождений.

3. Исключить из множества правил грамматики правило $A \rightarrow \epsilon$ и правила вида $B \rightarrow B$. Если в результате выполнения п.2 получены множества одинаковых правил, то из каждого такого множества оставить только одно.

Пример.

Грамматика:

- | | |
|------------------------|--------------------------------|
| 1. $S \rightarrow AaB$ | 6. $A \rightarrow b$ |
| 2. $S \rightarrow aB$ | 7. $B \rightarrow Ba$ |
| 3. $S \rightarrow cC$ | 8. $B \rightarrow \varepsilon$ |
| 4. $A \rightarrow AB$ | 9. $C \rightarrow AB$ |
| 5. $A \rightarrow B$ | 10. $C \rightarrow c$ |

1. Выбираем ε -правило 8. $B \rightarrow \varepsilon$.

2. Исключаем из правой части каждого правила исходной грамматики всеми возможными способами вхождение нетерминала B , полученные правила добавляем в множество правил грамматики.

- | | | |
|--------------------------|----------------------------------|----------------------------------|
| 1_1. $S \rightarrow AaB$ | 4_1. $A \rightarrow AB$ | 7_1. $B \rightarrow Ba$ |
| 1_2. $S \rightarrow Aa$ | 4_2. $A \rightarrow A$ | 7_2. $B \rightarrow a$ |
| 2_1. $S \rightarrow aB$ | 5_1. $A \rightarrow B$ | 8_1. $B \rightarrow \varepsilon$ |
| 2_2. $S \rightarrow a$ | 5_2. $A \rightarrow \varepsilon$ | 9_1. $C \rightarrow AB$ |
| 3_1. $S \rightarrow cC$ | 6_1. $A \rightarrow b$ | 9_2. $C \rightarrow A$ |
| | | 10_1. $C \rightarrow c$ |

3. Исключаем из множества правил грамматики правило 8_1. $B \rightarrow \varepsilon$ и правило 4_2. $A \rightarrow A$. Получаем грамматику:

- | | | |
|--------------------------|----------------------------------|-------------------------|
| 1_1. $S \rightarrow AaB$ | 4_1. $A \rightarrow AB$ | 7_1. $B \rightarrow Ba$ |
| 1_2. $S \rightarrow Aa$ | 5_1. $A \rightarrow B$ | 7_2. $B \rightarrow a$ |
| 2_1. $S \rightarrow aB$ | 5_2. $A \rightarrow \varepsilon$ | 9_1. $C \rightarrow AB$ |
| 2_2. $S \rightarrow a$ | 6_1. $A \rightarrow b$ | 9_2. $C \rightarrow A$ |
| 3_1. $S \rightarrow cC$ | | 10_1. $C \rightarrow c$ |

4. Выбираем ε -правило 5_2. $A \rightarrow \varepsilon$.

5. Исключаем из правой части каждого правила исходной грамматики всеми возможными способами вхождение нетерминала A , полученные правила добавляем в множество правил грамматики.

- | | | |
|----------------------------|------------------------------------|------------------------------------|
| 1_1_1. $S \rightarrow AaB$ | 4_1_1. $A \rightarrow AB$ | 9_1_1. $C \rightarrow AB$ |
| 1_1_2. $S \rightarrow aB$ | 4_1_2. $A \rightarrow B$ | 9_1_2. $C \rightarrow B$ |
| 1_2_1. $S \rightarrow Aa$ | 5_1_1. $A \rightarrow B$ | 9_2_1. $C \rightarrow A$ |
| 1_2_2. $S \rightarrow a$ | 5_2_1. $A \rightarrow \varepsilon$ | 9_2_2. $C \rightarrow \varepsilon$ |
| 2_1_1. $S \rightarrow aB$ | 6_1_1. $A \rightarrow b$ | 10_1_1. $C \rightarrow c$ |
| 2_2_1. $S \rightarrow a$ | 7_1_1. $B \rightarrow Ba$ | |
| 3_1_1. $S \rightarrow cC$ | 7_2_1. $B \rightarrow a$ | |

6. Исключаем из множества правил грамматики правило $5_2_1.A \rightarrow \varepsilon$. Из каждого множества одинаковых правил $\{1_1_2, 2_1_1\}$, $\{1_2_2, 2_2_1\}$, $\{4_1_2, 5_1_1\}$ оставляем по одному.

$1_1_1. S \rightarrow AaB$	$4_1_1. A \rightarrow AB$	$9_1_1. C \rightarrow AB$
$1_1_2. S \rightarrow aB$	$4_1_2. A \rightarrow B$	$9_1_2. C \rightarrow B$
$1_2_1. S \rightarrow Aa$	$6_1_1. A \rightarrow b$	$9_2_1. C \rightarrow A$
$1_2_2. S \rightarrow a$	$7_1_1. B \rightarrow Ba$	$9_2_2. C \rightarrow \varepsilon$
$3_1_1. S \rightarrow cC$	$7_2_1. B \rightarrow a$	$10_1_1. C \rightarrow c$

7. Выбираем ε -правило $9_2_2.C \rightarrow \varepsilon$.

8. Исключаем из правой части каждого правила исходной грамматики всеми возможными способами вхождение нетерминала C , полученные правила добавляем в множество правил грамматики.

$1_1_1_1. S \rightarrow AaB$	$4_1_1_1. A \rightarrow AB$	$9_1_1_1. C \rightarrow AB$
$1_1_2_1. S \rightarrow aB$	$4_1_2_1. A \rightarrow B$	$9_1_2_1. C \rightarrow B$
$1_2_1_1. S \rightarrow Aa$	$6_1_1_1. A \rightarrow b$	$9_2_1_1. C \rightarrow A$
$1_2_2_1. S \rightarrow a$	$7_1_1_1. B \rightarrow Ba$	$9_2_2_1. C \rightarrow \varepsilon$
$3_1_1_1. S \rightarrow cC$	$7_2_1_1. B \rightarrow a$	$10_1_1_1. C \rightarrow c$
$3_1_1_2. S \rightarrow c$		

9. Исключаем правило $9_2_2_1. C \rightarrow \varepsilon$.

$1_1_1_1. S \rightarrow AaB$	$4_1_1_1. A \rightarrow AB$	$9_1_1_1. C \rightarrow AB$
$1_1_2_1. S \rightarrow aB$	$4_1_2_1. A \rightarrow B$	$9_1_2_1. C \rightarrow B$
$1_2_1_1. S \rightarrow Aa$	$6_1_1_1. A \rightarrow b$	$9_2_1_1. C \rightarrow A$
$1_2_2_1. S \rightarrow a$	$7_1_1_1. B \rightarrow Ba$	$10_1_1_1. C \rightarrow c$
$3_1_1_1. S \rightarrow cC$	$7_2_1_1. B \rightarrow a$	
$3_1_1_2. S \rightarrow c$		

В полученной грамматике нет ε -правил, правил вида $B \rightarrow B$ и одинаковых правил.

Замена

Если грамматика содержит n правил $A \rightarrow \alpha_i$, где $1 \leq i \leq n$

и других правил с левой частью A нет,

и в грамматике есть правило $B \rightarrow \beta A \chi$,

то его можно заменить на n правил вида $B \rightarrow \beta \alpha_i \chi$, где $1 \leq i \leq n$.

Такое преобразование грамматики называется *заменой*,

а нетерминал A в правиле $B \rightarrow \beta A \chi$ — *заменяемым*.

В результате выполнения замен некоторые нетерминалы могут стать недостижимыми, и правила, их содержащие, нужно удалить.

Правило $A \rightarrow \alpha$ называется *одиночным*, если оно единственное с правой частью A . Замена всех вхождений нетерминалов, являющихся левыми частями одиночных правил, называется *одиночной заменой*. Одиночное правило при этом исключается (нетерминал A становится недостижимым).

В правиле грамматики назовём самое левое вхождение символа в его правую часть *краем* правила (ϵ -правило края не имеет). Если заменяемый нетерминал является краем правила, то такая замена называется *заменой края*.

Примеры.

а) замена края.

Исходная грамматика:

1. $A \rightarrow a$
2. $A \rightarrow Bc$
3. $B \rightarrow aA$
4. $B \rightarrow bB$

Результат замены края:

1. $A \rightarrow a$
- 2_1. $A \rightarrow aAc$
- 2_2. $A \rightarrow bBc$
3. $B \rightarrow aA$
4. $B \rightarrow bB$

В результате выполнения замены все нетерминалы остались достижимыми.

Исходная грамматика:

1. $A \rightarrow cA$
2. $A \rightarrow Bc$
3. $B \rightarrow aA$
4. $B \rightarrow b$

Результат замены края:

1. $A \rightarrow cA$
- 2_1. $A \rightarrow aAc$
- 2_2. $A \rightarrow bc$

В результате выполнения замены нетерминал B стал недостижимым и правила, содержащие B , исключены из множества правил грамматики.

б) одиночная замена.

Исходная грамматика:

1. $A \rightarrow aBV$
2. $A \rightarrow vB$
3. $A \rightarrow c$
4. $B \rightarrow aAb$

Результат одиночной замены :

1. $A \rightarrow a aAb aAb$
2. $A \rightarrow baAb$
3. $A \rightarrow c$

Левая факторизация

Если $n \geq 2$ правил грамматики имеют одинаковые левые части, допустим нетерминал A , и правые части начинаются одним или несколькими одинаковыми символами (имеют общий префикс α), т.е.

$$A \rightarrow \alpha \beta_1$$

$$A \rightarrow \alpha \beta_2$$

$$A \rightarrow \alpha \beta_i$$

$$A \rightarrow \alpha \beta_n$$

где $1 \leq i \leq n$, то можно общий префикс α вынести в отдельное правило $A \rightarrow \alpha V$, где V – новый нетерминал, и добавить n правил вида $V \rightarrow \beta_i$, где $1 \leq i \leq n$:

$$A \rightarrow \alpha V$$

$$V \rightarrow \beta_1$$

$$V \rightarrow \beta_2$$

$$V \rightarrow \beta_i$$

$$V \rightarrow \beta_n$$

Такое преобразование называется *левой факторизацией*. Результат применения левой факторизации неоднозначный, зависит от выбора префикса, выносимого в отдельное правило (например, в качестве префикса можно взять один символ или общий префикс наибольшей длины) и количества правил, участвующих в факторизации.

Пример 1.

Исходная грамматика	Первый шаг	Второй шаг
$S \rightarrow abBa$ $S \rightarrow abBb$ $S \rightarrow abA$ $B \rightarrow bB$ $B \rightarrow b$ $A \rightarrow a$	$S \rightarrow abC$ $C \rightarrow Ba$ $C \rightarrow Bb$ $C \rightarrow A$ $B \rightarrow bD$ $D \rightarrow B$ $D \rightarrow \varepsilon$ $A \rightarrow a$	$S \rightarrow abC$ $C \rightarrow BE$ $C \rightarrow A$ $E \rightarrow a$ $E \rightarrow b$ $B \rightarrow bD$ $D \rightarrow B$ $D \rightarrow \varepsilon$ $A \rightarrow a$

Пример 2.

Исходная грамматика	Первый шаг	Второй шаг
$S \rightarrow abBa$ $S \rightarrow abBb$ $S \rightarrow abA$ $B \rightarrow bB$ $B \rightarrow b$ $A \rightarrow a$	$S \rightarrow abBE$ $S \rightarrow abA$ $E \rightarrow a$ $E \rightarrow b$ $B \rightarrow bD$ $D \rightarrow B$ $D \rightarrow \varepsilon$ $A \rightarrow a$	$S \rightarrow abC$ $C \rightarrow BE$ $C \rightarrow A$ $E \rightarrow a$ $E \rightarrow b$ $B \rightarrow bD$ $D \rightarrow B$ $D \rightarrow \varepsilon$ $A \rightarrow a$

Пример 3.

Исходная грамматика	Первый шаг	Второй шаг	Третий шаг
$S \rightarrow abBa$ $S \rightarrow abBb$ $S \rightarrow abA$ $B \rightarrow bB$ $B \rightarrow b$ $A \rightarrow a$	$S \rightarrow aC$ $C \rightarrow bBa$ $C \rightarrow bBb$ $C \rightarrow bA$ $B \rightarrow bD$ $D \rightarrow B$ $D \rightarrow \varepsilon$ $A \rightarrow a$	$S \rightarrow aC$ $C \rightarrow bE$ $E \rightarrow Ba$ $E \rightarrow Bb$ $E \rightarrow A$ $B \rightarrow bD$ $D \rightarrow B$ $D \rightarrow \varepsilon$ $A \rightarrow a$	$S \rightarrow aC$ $C \rightarrow bE$ $E \rightarrow BF$ $E \rightarrow A$ $F \rightarrow a$ $F \rightarrow b$ $B \rightarrow bD$ $D \rightarrow B$ $D \rightarrow \varepsilon$ $A \rightarrow a$

Устранение цепных правил

Циклом (циклическим выводом) называется вывод вида $A \Rightarrow^* A$, где A – нетерминал грамматики. Циклический вывод бесполезен.

Циклы возможны только в том случае, если в грамматике есть *цепные правила* вида $A \rightarrow B$, где A и B – нетерминалы грамматики.

В алгоритме устранения цепных правил будем использовать множество M^A – множество нетерминалов, достижимых из нетерминала A только применением цепных правил.

Алгоритм нахождения множества M^A .

1. Исключить из правил грамматики все нецепные правила.
2. Принять множество $M^A = \{A\}$, где A – нетерминал.
3. Если существует правило $B \rightarrow C$ и нетерминал B принадлежит множеству M^A , то нетерминал C включить в множество M^A .
4. Повторять п.3, пока множество M^A растёт.
5. Исключить нетерминал A из множества M^A .

Алгоритм устранения цепных правил.

1. Для каждого нетерминала A из множества нетерминалов грамматики найти множество M^A .
2. Исключить из множества правил грамматики все цепные правила.
3. Для правил $A \rightarrow \alpha$ из множества правил грамматики добавить правило $B \rightarrow \alpha$, если A принадлежит множеству M^B .

Пример.

Грамматика с цепными правилами:

- | | |
|-------------------------|------------------------|
| 1. $S \rightarrow S+T$ | 4. $T \rightarrow E$ |
| 2. $S \rightarrow T$ | 5. $E \rightarrow (S)$ |
| 3. $T \rightarrow T^*E$ | 6. $E \rightarrow a$ |

1. Для каждого нетерминала находим множество нетерминалов, достижимых применением только цепных правил. Очевидно

$$M^S = \{T, E\},$$

$$M^T = \{E\} \text{ и}$$

$$M^E = \emptyset.$$

2. Исключить из множества правил грамматики все цепные правила.

- | |
|-------------------------|
| 1. $S \rightarrow S+T$ |
| 3. $T \rightarrow T^*E$ |
| 5. $E \rightarrow (S)$ |
| 6. $E \rightarrow a$ |

3. Для правила 3. $T \rightarrow T^*E$ добавляем правило 3_1. $S \rightarrow T^*E$, поскольку T принадлежит $M^S = \{T, E\}$.

Для правила 5. $E \rightarrow (S)$ добавляем правило 5_1. $S \rightarrow (S)$ и 5_2. $T \rightarrow (S)$, поскольку E принадлежит $M^S = \{T, E\}$ и $M^T = \{E\}$.

Для правила 6. $E \rightarrow a$ добавляем правило 6_1. $S \rightarrow a$ и 6_2. $T \rightarrow a$, поскольку E принадлежит $M^S = \{T, E\}$ и $M^T = \{E\}$.

В результате получим:

- | | | |
|-------------------------|---------------------------|--------------------------|
| 1. $S \rightarrow S+T$ | | |
| 3. $T \rightarrow T^*E$ | 3_1. $S \rightarrow T^*E$ | |
| 5. $E \rightarrow (S)$ | 5_1. $S \rightarrow (S)$ | 5_2. $T \rightarrow (S)$ |
| 6. $E \rightarrow a$ | 6_1. $S \rightarrow a$ | 6_2. $T \rightarrow a$ |

Устранение левой рекурсии

Правило $A \rightarrow \chi$ называется *рекурсивным*, если существует вывод $\chi \Rightarrow^* \alpha A \beta$.

Если $\alpha = \epsilon$ и $\beta \neq \epsilon$, то правило $A \rightarrow \chi$ называется *леворекурсивным* ($\chi \Rightarrow^* A \beta$).

Правило называется *самолеворекурсивным* ($A \rightarrow A \beta$), если его край совпадает с левой частью. Самолеворекурсивное правило также является и леворекурсивным.

1. Исключение самолеворекурсивных правил.

Предположим, что в грамматике для нетерминала A только два правила:

1. $A \rightarrow A\alpha$ (самоеворекурсивное)

2. $A \rightarrow \beta$ (несаморекурсивное)

Применяя эти правила, из A можно вывести:

$$\begin{array}{l} A \Rightarrow \beta \\ 2 \end{array}$$

$$\begin{array}{l} A \Rightarrow A\alpha \Rightarrow \beta\alpha \\ 1 \quad 2 \end{array}$$

$$\begin{array}{l} A \Rightarrow A\alpha \Rightarrow A\alpha\alpha \Rightarrow \beta\alpha\alpha \\ 1 \quad 1 \quad 2 \end{array}$$

$$\begin{array}{l} A \Rightarrow A\alpha \Rightarrow \dots \Rightarrow A\alpha \dots \alpha \Rightarrow \beta\alpha \dots \alpha \\ 1 \quad 1 \quad 2 \end{array}$$

Получается, что $L(A) = \{\beta\alpha^k \mid k \geq 0\}$

Обозначим $L(B) = \{\alpha^k \mid k \geq 0\}$. Тогда:

$$B \rightarrow \alpha B$$

$$B \rightarrow \varepsilon$$

Тогда грамматику языка $L(A) = \{\beta\alpha^k \mid k \geq 0\}$ можем записать так:

$$A \rightarrow \beta B$$

$$B \rightarrow \alpha B$$

$$B \rightarrow \varepsilon$$

Предположим, что в грамматике для нетерминала A следующие правила:

$$1. A \rightarrow A\alpha$$

$$2. A \rightarrow \beta_1$$

$$3. A \rightarrow \beta_2$$

$$n+1. A \rightarrow \beta_n$$

Применяя 1-е правила, будем получать цепочки вида $A\alpha \dots \alpha$.

Чтобы избавиться от A в начале цепочки, нужно применить несамолеворекурсивное правило.

В результате получим, что $L(A) = \{\beta_j \alpha^k \mid 1 \leq j \leq n, k \geq 0\}$

Грамматика $L(A)$ будет следующей:

$$A \rightarrow \beta_1 B$$

$$A \rightarrow \beta_2 B$$

$$A \rightarrow \beta_n B$$

$$B \rightarrow \alpha B$$

$$B \rightarrow \varepsilon$$

Предположим, что в грамматике для нетерминала A следующие правила:

$$1. A \rightarrow A\alpha_1$$

$$2. A \rightarrow A\alpha_2$$

$$m. A \rightarrow A\alpha_m$$

$$m+1. A \rightarrow \beta_1$$

$$m+2. A \rightarrow \beta_2$$

$$m+n. A \rightarrow \beta_n$$

Применять самолеворекурсивные правила можем в любом порядке и будем получать цепочку, в которой за A будет чередование цепочек α_i .

Это чередование обозначим B . Правила будут такими:

$$B \rightarrow \alpha_1 B$$

$$B \rightarrow \alpha_2 B$$

$$B \rightarrow \alpha_m B$$

$$B \rightarrow \varepsilon$$

Применять самолеворекурсивные правила можем в любом порядке и будем получать цепочку, в которой за A будет чередование цепочек α_i .

Чтобы убрать первое A , нужно применить одно из несамоеворекурсивных правил. В итоге получим грамматику:

$$A \rightarrow \beta_1 B$$

$$A \rightarrow \beta_2 B$$

$$A \rightarrow \beta_n B$$

$$B \rightarrow \alpha_1 B$$

$$B \rightarrow \alpha_2 B$$

$$B \rightarrow \alpha_m B$$

$$B \rightarrow \varepsilon$$

Предположим, что нетерминал A имеет m самоеворекурсивных правил $A \rightarrow A\alpha_i$, где $1 \leq i \leq m$, и n правил $A \rightarrow \beta_j$, где $1 \leq j \leq n$, которые не являются саморекурсивными и других правил с левой частью A нет. Эти правила заменяются следующими:

$$A \rightarrow \beta_j B, \quad \text{где } 1 \leq j \leq n, B - \text{новый нетерминал,}$$

$$B \rightarrow \alpha_i B, \quad \text{где } 1 \leq i \leq m,$$

$$B \rightarrow \varepsilon$$

2. Исключение леворекурсивных правил.

Алгоритм исключения леворекурсивных правил.

1. Обозначить нетерминалы грамматики A_1, A_2, \dots, A_n , где n – количество нетерминалов.
2. Для каждого нетерминала грамматики A_i , где $1 \leq i \leq n$, выполнить п.3 и 4.
3. Для каждого правила вида $A_i \rightarrow A_j \alpha$, где $1 \leq j \leq i-1$, выполнить замену края (новые правила необходимо учитывать при выполнении п.3).
4. Исключить самолеворекурсивные правила для нетерминала A_i (новые нетерминалы далее не рассматривать).

Алгоритм применим, если грамматика не имеет циклов (цепных правил) и ϵ -правил. Цепные правила и ϵ -правила могут быть удалены предварительно. Получающаяся грамматика без левой рекурсии может иметь ϵ -правила.

Пример.

Устранить левую рекурсию в грамматике:

$$A_1 \rightarrow A_1 a A_3$$

$$A_1 \rightarrow A_2 b$$

$$A_2 \rightarrow A_1 c$$

$$A_2 \rightarrow A_3 a$$

$$A_3 \rightarrow A_1 b$$

$$A_3 \rightarrow c$$

Рассматриваем нетерминал A_1 .

Правил вида $A_1 \rightarrow A_0 \alpha$ в грамматике нет, т.к. нет нетерминала A_0 , поэтому замену края (п.3) не выполняем.

Исключаем самолеворекурсивное правило $A_1 \rightarrow A_1 a A_3$, получаем грамматику:

$$A_1 \rightarrow A_2 b B_1$$

$$A_2 \rightarrow A_1 c$$

$$A_2 \rightarrow A_3 a$$

$$A_3 \rightarrow A_1 b$$

$$A_3 \rightarrow c$$

$$B_1 \rightarrow a A_3 B_1$$

$$B_1 \rightarrow \varepsilon$$

Рассматриваем нетерминал A_2 .

Выполняем замену края в правиле $A_2 \rightarrow A_1 c$, получаем грамматику:

$$A_1 \rightarrow A_2 b B_1$$

$$A_2 \rightarrow A_2 b B_1 c$$

$$A_2 \rightarrow A_3 a$$

$$A_3 \rightarrow A_1 b$$

$$A_3 \rightarrow c$$

$$B_1 \rightarrow a A_3 B_1$$

$$B_1 \rightarrow \varepsilon$$

Исключаем самолеворекурсивное правило $A_2 \rightarrow A_2 b B_1 c$, получаем грамматику:

$$A_1 \rightarrow A_2 b B_1$$

$$A_2 \rightarrow A_3 a B_2$$

$$A_3 \rightarrow A_1 b$$

$$A_3 \rightarrow c$$

$$B_1 \rightarrow a A_3 B_1$$

$$B_1 \rightarrow \varepsilon$$

$$B_2 \rightarrow b B_1 c B_2$$

$$B_2 \rightarrow \varepsilon$$

Рассматриваем нетерминал A_3 .

Выполняем замену края в правиле $A_3 \rightarrow A_1 b$, получаем грамматику:

$$\begin{aligned} A_1 &\rightarrow A_2 b B_1 \\ A_2 &\rightarrow A_3 a B_2 \\ A_3 &\rightarrow A_2 b B_1 b \\ A_3 &\rightarrow c \\ B_1 &\rightarrow a A_3 B_1 \\ B_1 &\rightarrow \epsilon \\ B_2 &\rightarrow b B_1 c B_2 \\ B_2 &\rightarrow \epsilon \end{aligned}$$

Выполняем замену края в правиле $A_3 \rightarrow A_2 b B_1 b$, получаем грамматику:

$$\begin{aligned} A_1 &\rightarrow A_2 b B_1 \\ A_2 &\rightarrow A_3 a B_2 \\ A_3 &\rightarrow A_3 a B_2 b B_1 b \\ A_3 &\rightarrow c \\ B_1 &\rightarrow a A_3 B_1 \\ B_1 &\rightarrow \epsilon \\ B_2 &\rightarrow b B_1 c B_2 \\ B_2 &\rightarrow \epsilon \end{aligned}$$

Исключаем самолеворекурсивное правило $A_3 \rightarrow A_3 a B_2 b B_1 b$, получаем грамматику:

$$\begin{aligned} A_1 &\rightarrow A_2 b B_1 \\ A_2 &\rightarrow A_3 a B_2 \\ A_3 &\rightarrow c B_3 \\ B_1 &\rightarrow a A_3 B_1 \\ B_1 &\rightarrow \epsilon \\ B_2 &\rightarrow b B_1 c B_2 \\ B_2 &\rightarrow \epsilon \\ B_3 &\rightarrow a B_2 b B_1 b B_3 \\ B_3 &\rightarrow \epsilon \end{aligned}$$

Грамматика без леворекурсивных правил получена. Далее можем выполнить две одиночные замены и получим грамматику:

$$\begin{aligned} A_1 &\rightarrow c B_3 a B_2 b B_1 \\ A_3 &\rightarrow c B_3 \\ B_1 &\rightarrow a c B_3 B_1 \\ B_1 &\rightarrow \epsilon \\ B_2 &\rightarrow b B_1 c B_2 \\ B_2 &\rightarrow \epsilon \\ B_3 &\rightarrow a B_2 b B_1 b B_3 \\ B_3 &\rightarrow \epsilon \end{aligned}$$

Устранение правой рекурсии

Правило $A \rightarrow \chi$ называется *рекурсивным*, если существует вывод $\chi \Rightarrow^* \alpha A \beta$. Если $\alpha \neq \varepsilon$ и $\beta = \varepsilon$, то правило $A \rightarrow \chi$ называется *праворекурсивным*. Правило называется *самоправорекурсивным*, если его самый правый символ нетерминал и он совпадает с левой частью. Самоправорекурсивное правило также является и праворекурсивным.

1. Исключение самоправорекурсивных правил.

Предположим, что нетерминал A имеет m самоправорекурсивных правил $A \rightarrow \alpha_i A$, где $1 \leq i \leq m$, и n правил $A \rightarrow \beta_j$, где $1 \leq j \leq n$, которые не являются самоправорекурсивными и других правил с левой частью A нет. Эти правила заменяются следующими:

$$\begin{aligned} A &\rightarrow B\beta_j, & \text{где } 1 \leq j \leq n, B - \text{новый нетерминал,} \\ B &\rightarrow B\alpha_i, & \text{где } 1 \leq i \leq m, \\ B &\rightarrow \varepsilon \end{aligned}$$

2. Исключение праворекурсивных правил.

Алгоритм исключения праворекурсивных правил.

1. Обозначить нетерминалы грамматики A_1, A_2, \dots, A_n , где n – количество нетерминалов.

2. Для каждого нетерминала грамматики A_i , где $1 \leq i \leq n$, выполнить п.3 и 4.

3. Для каждого правила вида $A_i \rightarrow \alpha A_j$, где $1 \leq j \leq i-1$, выполнить замену нетерминала A_j (новые правила необходимо учитывать при выполнении п.3).

4. Исключить самоправорекурсивные правила для нетерминала A_i (новые нетерминалы далее не рассматривать).

Алгоритм применим, если грамматика не имеет циклов (цепных правил) и ε -правил. Цепные правила и ε -правила могут быть удалены предварительно. Получающаяся грамматика без правой рекурсии может иметь ε -правила.

Пример.

Устранить правую рекурсию в грамматике:

$$\begin{aligned} A_1 &\rightarrow aA_3A_1 \\ A_1 &\rightarrow bA_2 \\ A_2 &\rightarrow cA_1 \\ A_2 &\rightarrow aA_3 \\ A_3 &\rightarrow bA_1 \\ A_3 &\rightarrow c \end{aligned}$$

Рассматриваем нетерминал A_1 .

Правил вида $A_1 \rightarrow \alpha A_0$ в грамматике нет, т.к. нет нетерминала A_0 , поэтому замену нетерминала A_1 (п.3) не выполняем.

Исключаем самоправорекурсивное правило $A_1 \rightarrow aA_3A_1$, получаем грамматику:

$$A_1 \rightarrow B_1bA_2$$

$$A_2 \rightarrow cA_1$$

$$A_2 \rightarrow aA_3$$

$$A_3 \rightarrow bA_1$$

$$A_3 \rightarrow c$$

$$B_1 \rightarrow B_1aA_3$$

$$B_1 \rightarrow \epsilon$$

Рассматриваем нетерминал A_2 .

Выполняем замену нетерминала A_1 в правиле $A_2 \rightarrow cA_1$, получаем грамматику:

$$A_1 \rightarrow B_1bA_2$$

$$A_2 \rightarrow cB_1bA_2$$

$$A_2 \rightarrow aA_3$$

$$A_3 \rightarrow bA_1$$

$$A_3 \rightarrow c$$

$$B_1 \rightarrow B_1aA_3$$

$$B_1 \rightarrow \epsilon$$

Исключаем самоправорекурсивное правило $A_2 \rightarrow cB_1bA_2$, получаем грамматику:

$$A_1 \rightarrow B_1bA_2$$

$$A_2 \rightarrow B_2aA_3$$

$$A_3 \rightarrow bA_1$$

$$A_3 \rightarrow c$$

$$B_1 \rightarrow B_1aA_3$$

$$B_1 \rightarrow \epsilon$$

$$B_2 \rightarrow B_2cB_1b$$

$$B_2 \rightarrow \epsilon$$

Рассматриваем нетерминал A_3 .

Выполняем замену нетерминала A_1 в правиле $A_3 \rightarrow bA_1$, получаем грамматику:

$$A_1 \rightarrow B_1bA_2$$

$$A_2 \rightarrow B_2aA_3$$

$$A_3 \rightarrow bB_1bA_2$$

$$A_3 \rightarrow c$$

$$B_1 \rightarrow B_1aA_3$$

$$B_1 \rightarrow \epsilon$$

$$B_2 \rightarrow B_2cB_1b$$

$$B_2 \rightarrow \epsilon$$

Выполняем замену нетерминала A_2 в правиле $A_3 \rightarrow bB_1bA_2$, получаем грамматику:

$$A_1 \rightarrow B_1bA_2$$

$$A_2 \rightarrow B_2aA_3$$

$$A_3 \rightarrow bB_1bB_2aA_3$$

$$A_3 \rightarrow c$$

$$B_1 \rightarrow B_1aA_3$$

$$B_1 \rightarrow \epsilon$$

$$B_2 \rightarrow B_2cB_1b$$

$$B_2 \rightarrow \epsilon$$

Исключаем самоправорекурсивное правило $A_3 \rightarrow bB_1bB_2aA_3$, получаем грамматику:

$$A_1 \rightarrow B_1bA_2$$

$$A_2 \rightarrow B_2aA_3$$

$$A_3 \rightarrow B_3c$$

$$B_1 \rightarrow B_1aA_3$$

$$B_1 \rightarrow \epsilon$$

$$B_2 \rightarrow B_2cB_1b$$

$$B_2 \rightarrow \epsilon$$

$$B_3 \rightarrow B_3bB_1bB_2a$$

$$B_3 \rightarrow \epsilon$$

Грамматика без праворекурсивных правил получена. Далее можем выполнить две одиночные замены и получим грамматику:

$$A_1 \rightarrow B_1bB_2aB_3c$$

$$B_1 \rightarrow B_1aB_3c$$

$$B_1 \rightarrow \epsilon$$

$$B_2 \rightarrow B_2cB_1b$$

$$B_2 \rightarrow \epsilon$$

$$B_3 \rightarrow B_3bB_1bB_2a$$

$$B_3 \rightarrow \epsilon$$

Нормальная форма Хомского

Любой КС язык может быть задан КС-грамматикой в нормальной форме Хомского (НФХ), а любую КС-грамматику можно преобразовать в эквивалентную ей грамматику в НФХ.

Грамматика в НФХ содержит только правила трёх видов:

- 1) $A \rightarrow BC$, где A, B, C – нетерминалы;
- 2) $A \rightarrow t$, где t – терминал;
- 3) $S \rightarrow \varepsilon$, если пустая цепочка принадлежит языку, причём

начальный нетерминал S не встречается в правых частях правил.

Другими словами, это грамматика без ε -правил, а правые части правил содержат либо два нетерминала, либо один терминал.

Для преобразования произвольной грамматики в НФХ предварительно нужно устранить лишние символы, ε -правила и цепные правила, т.е. *привести* грамматику.

Алгоритм преобразования приведённой грамматики в НФХ.

1. Для каждого правила вида $A \rightarrow X\alpha$, где A – нетерминал, X – терминал или нетерминал, α – цепочка терминалов или нетерминалов, содержащая более одного символа, выполнить следующее преобразование:

если в грамматике есть одиночное правило $C \rightarrow \alpha$, то правило $A \rightarrow X\alpha$ заменить на $A \rightarrow XC$, иначе ввести правило $N \rightarrow \alpha$, где N – новый нетерминал, и правило $A \rightarrow X\alpha$ заменить на $A \rightarrow XN$.

При выполнении п.1 учитывать вводимые правила.

В результате выполнения п.1 получим грамматику, правые части правил которой содержат не более двух символов.

2. Для каждого правила вида $A \rightarrow tB$, где A, B – нетерминалы, t – терминал, выполнить следующее преобразование:

если в грамматике есть одиночное правило $C \rightarrow t$, то правило $A \rightarrow tB$ заменить на $A \rightarrow CB$, иначе ввести правило $N \rightarrow t$, где N – новый нетерминал, и правило $A \rightarrow tB$ заменить на $A \rightarrow NB$.

3. Для каждого правила вида $A \rightarrow Bt$, где A, B – нетерминалы, t – терминал, выполнить следующее преобразование:

если в грамматике есть одиночное правило $C \rightarrow t$, то правило $A \rightarrow Bt$ заменить на $A \rightarrow BC$, иначе ввести правило $N \rightarrow t$, где N – новый нетерминал, и правило $A \rightarrow Bt$ заменить на $A \rightarrow BN$.

4. Для каждого правила вида $A \rightarrow t_1 t_2$, где t_1 и t_2 – терминалы, заменить их на нетерминалы так, как описано в п.2 и 3.

Пример.

Преобразовать заданную грамматику в НФХ.

Исходная грамматика	Пункт 1 алгоритма		Пункты 2, 3 и 4 алгоритма
$A \rightarrow aBCd$	$A \rightarrow aN_1$	$A \rightarrow aN_1$	$A \rightarrow N_2N_1$
$A \rightarrow Cd$	$N_1 \rightarrow BCd$	$N_1 \rightarrow BA$	$N_2 \rightarrow a$
$B \rightarrow aB$	$A \rightarrow Cd$	$A \rightarrow Cd$	$N_1 \rightarrow BA$
$B \rightarrow b$	$B \rightarrow aB$	$B \rightarrow aB$	$A \rightarrow CN_3$
$C \rightarrow ABA$	$B \rightarrow b$	$B \rightarrow b$	$N_3 \rightarrow d$
$C \rightarrow ab$	$C \rightarrow ABA$	$C \rightarrow AN_1$	$B \rightarrow N_2B$
	$C \rightarrow ab$	$C \rightarrow ab$	$B \rightarrow b$
			$C \rightarrow AN_1$
			$C \rightarrow N_2N_4$
			$N_4 \rightarrow b$

Нормальная форма Грейбах

Любой КС язык может быть задан КС-грамматикой в нормальной форме Грейбах (НФГ), а любую КС-грамматику можно преобразовать в эквивалентную ей грамматику в НФГ.

Грамматика в НФГ содержит только правила двух видов:

1) $A \rightarrow t\alpha$, где A – нетерминал, t – терминал, α - цепочка нетерминалов, возможно пустая;

2) $S \rightarrow \epsilon$, если пустая цепочка принадлежит языку, причём начальный нетерминал S не встречается в правых частях правил.

Другими словами, это грамматика без ϵ -правил, а правые части правил содержат только один терминал, занимающий крайнюю левую позицию.

Для преобразования произвольной грамматики в НФГ предварительно нужно привести грамматику и устранить левую рекурсию.

Алгоритм преобразования приведённой грамматики без левой рекурсии в НФГ.

1. Упорядочить правила грамматики следующим образом:

а) последовательно выписать правила вида $A \rightarrow B\alpha$ в таком порядке, чтобы для любого правила выполнялось условие:

правила с нетерминалом B в левой части должны располагаться ниже правила $A \rightarrow B\alpha$, а правила с краем A – выше правила $A \rightarrow B\alpha$.

б) последовательно выписать правила вида $A \rightarrow t\alpha$;

2. Просматривая правила грамматики снизу вверх, выполнять замену края, пока это возможно.

3. Если в правой части некоторого правила встречается терминал t , занимающий не крайнюю левую позицию, и в грамматике есть одиночное правило $A \rightarrow t$, то терминал t заменить нетерминалом A , если же в грамматике нет одиночного правила $A \rightarrow t$, то ввести его и терминал t заменить на нетерминал A . Выполнять п.3, пока это возможно.

Пример.

Преобразовать заданную грамматику в НФГ.

Выполнение пунктов 1 и 2 алгоритма представлено в таблице.

Исходная грамматика	Правила упорядочены	Выполнение замены края			Результат п.2
$S \rightarrow AC$	$B \rightarrow Sb$	$B \rightarrow Sb$	$B \rightarrow Sb$	$B \rightarrow aBCb$	$S \rightarrow aBC$
$C \rightarrow Aa$	$S \rightarrow AC$	$S \rightarrow AC$	$S \rightarrow aBC$	$B \rightarrow dCb$	$S \rightarrow dC$
$B \rightarrow Sb$	$C \rightarrow Aa$	$C \rightarrow aBa$	$S \rightarrow dC$	$S \rightarrow aBC$	$C \rightarrow aBa$
$B \rightarrow bB$	$B \rightarrow bB$	$C \rightarrow da$	$C \rightarrow aBa$	$S \rightarrow dC$	$C \rightarrow da$
$A \rightarrow aB$	$A \rightarrow aB$	$B \rightarrow bB$	$C \rightarrow da$	$C \rightarrow aBa$	$B \rightarrow aBCb$
$A \rightarrow d$	$A \rightarrow d$	$A \rightarrow aB$	$B \rightarrow bB$	$C \rightarrow da$	$B \rightarrow dCb$
		$A \rightarrow d$	$A \rightarrow aB$	$B \rightarrow bB$	$B \rightarrow bB$
			$A \rightarrow d$	$A \rightarrow aB$	$A \rightarrow aB$
				$A \rightarrow d$	$A \rightarrow d$

При выполнении п.3 алгоритма вводим два одиночных правила $D \rightarrow b$ и $E \rightarrow a$ и в правых частях правил, полученных в результате п.2, заменяем терминалы b , расположенные не в крайней левой позиции, на нетерминал D , а терминалы a , расположенные не в крайней левой позиции, на нетерминал E . В итоге получаем грамматику в нормальной форме Грейбах:

$S \rightarrow aBC$	$B \rightarrow bB$
$S \rightarrow dC$	$A \rightarrow aB$
$C \rightarrow aBE$	$A \rightarrow d$
$C \rightarrow dE$	$D \rightarrow b$
$B \rightarrow aBCD$	$E \rightarrow a$
$B \rightarrow dCD$	

Операторная КС-грамматика

Любой КС язык может быть задан операторной КС-грамматикой , а любую КС-грамматику можно преобразовать в эквивалентную ей операторную. *Операторной грамматикой* называется КС-грамматика без ϵ -правил, в которой правые части всех правил не содержат смежных нетерминалов. *Правило назовём операторным*, если в его правой части нет смежных нетерминалов.

Для преобразования произвольной грамматики в операторную предварительно получим грамматику, правые части правил которой начинаются терминалом (см. ‘Нормальная форма Грейбах’).

Алгоритм преобразования грамматики, правые части правил которой начинаются терминалом , в операторную.

1. Для каждого правила вида $A \rightarrow \alpha\beta\chi$,

где A – нетерминал;

α - цепочка терминалов и нетерминалов, начинающаяся с терминала;

β - цепочка нетерминалов, содержащая более одного символа;

χ - цепочка терминалов и нетерминалов, возможно пустая;

выполнить следующее преобразование:

если в грамматике есть одиночное правило $B \rightarrow \beta$, то правило $A \rightarrow \alpha\beta\chi$

заменить на $A \rightarrow \alpha B\chi$, иначе ввести правило $N \rightarrow \beta$, где N – новый не-

терминал, и правило $A \rightarrow \alpha\beta\chi$ заменить на $A \rightarrow \alpha N\chi$.

Выполнять п.1 , пока в грамматике есть правила вида $A \rightarrow \alpha\beta\chi$.

В результате выполнения п.1 правила, определяющие нетерминалы исходной грамматики, будут операторными.

2. Пока в грамматике есть правила, содержащие в правой части пару смежных нетерминалов XY , выполнять замену второго нетерминала Y .

Пример.

Преобразовать заданную грамматику в операторную.

Исходная грамматика	Результат выполнения п.1	Операторная грамматика
$A \rightarrow aBAC$	$A \rightarrow aN_1$	$A \rightarrow aN_1$
$A \rightarrow \epsilon$	$A \rightarrow \epsilon$	$A \rightarrow \epsilon$
$B \rightarrow dBACb$	$B \rightarrow dN_1b$	$B \rightarrow dN_1b$
$B \rightarrow b$	$B \rightarrow b$	$B \rightarrow b$
$C \rightarrow aBaAB$	$C \rightarrow aBaN_2$	$C \rightarrow aBaN_2$
	$N_1 \rightarrow BAC$	$N_1 \rightarrow BaN_1aBaN_2$
	$N_2 \rightarrow AB$	$N_1 \rightarrow BeC$
		$N_2 \rightarrow AdN_1b$
		$N_2 \rightarrow Ab$