Преобразования КС-грамматик

Основные понятия

Грамматики G_i и G_J называются *эквивалентными*, если равны порождаемые ими языки ($L(G_i) = L(G_i)$). 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 граматике 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 $A \to \alpha$, в котором все символы в правой части продуктивны.

Алгоритм нахождения всех продуктивных нетерминалов.

1. Принять множество продуктивных нетерминалов $P=\varnothing$.

2. Если существует правило $A \to \alpha$, 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 $A \to \alpha$ и A — достижимый нетерминал.

Алгоритм нахождения всех достижимых символов.

1. Принять множество достижимых символов P={S}, где S – начальный нетерминал.

2. Если существует правило $A \to \alpha$ 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. $S \to ac$	4. $B \to aSA$
2. $S \to bA$	5. $C \to bC$
3. $A \to cBC$	6. $C \to d$

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. $S \to ac$
5. $C \to bC$
6. $C \to d$

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. $S \to ac$

Заметим, что в исходной грамматике все символы достижимы.

Исключение лишних правил

Правило $A \to \alpha$ 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. $A \to aABa$
2. $A \to bC$
3. $A \to bba$
4. $B \to dBC$
5. $B \to \varepsilon$
6. $C \to BA$
7. $C \to a$

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

$$A \underset{1}{\Rightarrow} bC \underset{6}{\Rightarrow} bBA \underset{5}{\Rightarrow} bA \underset{2}{\Rightarrow} bbC \underset{7}{\Rightarrow} bba$$

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

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

Исключение ε-правил

Правило вида $A \to \varepsilon$ называется *ε-правилом*. Грамматику, порождающую язык, несодержащий 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 $S \to \varepsilon$, где S – начальный нетерминал. Рассмотрим два алгоритма ислючения 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. $S \to AaB$	6. $A \to b$
2. $S \to aB$	7. $B \to Ba$
3. $S \to cC$	8. $B \to \varepsilon$
4. $A \to AB$	9. $C \to AB$
5. $A \to B$	10. $C \to c$

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

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

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

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

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

Алгоритм 2 исключения ε-правил.

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

1. Выбрать ε-правило $A \to \varepsilon$.

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 $A \to \varepsilon$ и правила вида $B \to B$. Если в результате выполнения п.2 получены множества одинаковых правил, то из каждого такого множества оставить только одно.

Пример.

Грамматика:

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

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

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

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

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

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

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

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

6. Исключаем из множества правил грамматики правило 5_2_1.$A \to \varepsilon$. 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. $S \to AaB$	4_1_1. $A \to AB$	9_1_1. $C \to AB$
1_1_2. $S \to aB$	4_1_2. $A \to B$	9_1_2. $C \to B$
1_2_1. $S \to Aa$	6_1_1. $A \to b$	9_2_1. $C \to A$
1_2_2. $S \to a$	7_1_1. $B \to Ba$	9_2_2. $C \to \varepsilon$
3_1_1. $S \to cC$	7_2_1. $B \to a$	10_1_1. $C \to c$

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

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

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

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

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

Замена

Если грамматика содержит n правил $A \to \alpha_i$, где $1 \le i \le n$

и других правил с левой частью A нет,

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

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

Такое преобразование грамматики называется *заменой*,

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

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

Правило $A \to \alpha$ называется *одиночным*, если оно единственное с левой 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→a
2. A→Bc
3. B→aA
4. B→bB

Результат замены края:
1. A→a
2_1. A→aAc
2_2. A→bBc
3. B→aA
4. B→bB

В результате выполнения замены все нетерминалы остались достижимыми.

Исходная грамматика:
1. A→cA
2. A→Bc
3. B→aA
4. B→b

Результат замены края:
1. A→cA
2_1. A→aAc
2_2. A→bc

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

б) одиночная замена.

Исходная грамматика:
1. A→aBB
2. A→вB
3. A→c
4. B→aAb

Результат одиночной замены :
1. A→a aAb aAb
2. A→baAb
3. A→c

Устранение несаморекурсивных нетерминалов

Нетерминал А называется *саморекурсивным*, если в грамматике существует правило для нетерминала А (с нетерминалом А в левой части), в котором в правой части есть нетерминал А. В противном случае нетерминал А будет *несаморекурсивным*.

Если в грамматике есть несаморекурсивный нетерминал А (за исключением начального нетерминала), то её можно преобразовать в грамматику без этого нетерминала А (остальные нетерминалы в грамматике сохранятся и новые не появятся).

Примером может служить одиночная замена.

б) одиночная замена.

Исходная грамматика:	Результат одиночной замены :
1. A→aBB	1. A→a aAb aAb
2. A→вB	2. A→baAb
3. A→c	3. A→c
4. B→aAb	

Нетерминал в левой части одиночного правила обязательно несаморекурсивный, иначе он будет бесплодным и правила, его содержащие, нужно удалить.

Если же для несаморекурсивного нетерминала А существует более одного правила, то исключить нетерминал А из грамматики можно следующим способом:

1. Исключить правила для нетерминала А.

2. Пока в грамматике есть вхождения нетерминала А, выбрать одно из вхождений нетерминала А и заменить его правыми частями правил для нетерминала А.

Пример.

Исходная грамматика:

1. A→aBB
2. A→вB
3. A→c
4. B→aAb
5. B→b

Нетерминал B несаморекурсивный.

Исключаем правила 4 и 5 (с нетерминалом B в левой части):

1. A→aBB
2. A→вB
3. A→c

Здесь три вхождения нетерминала B.

Выбираем первое вхождение B в первом правиле и заменяем его правыми частями 4-го и 5-го правила:

1_1. A→aaAbB
1_2. A→abB
2. A→вB
3. A→c

Выбираем вхождение B в правиле 1_1 и заменяем его правыми частями 4-го и 5-го правила:

1_1_1. A→aaAbaAb
1_1_2. A→aaAbb
1_2. A→abB
2. A→вB
3. A→c

Выбираем вхождение B в правиле 1_2 и заменяем его правыми частями 4-го и 5-го правила:

1_1_1. A→aaAbaAb
1_1_2. A→aaAbb
1_2_1. A→abAbb
1_2_2. A→abb
2. A→вB
3. A→c

Выбираем вхождение B в правиле 2 и заменяем его правыми частями 4-го и 5-го правила:

1_1_1. A→aaAbaAb
1_1_2. A→aaAbb
1_2_1. A→abAbb
1_2_2. A→abb
2_1. A→вbAb
2_2. A→вb
3. A→c

Грамматика без несаморекурсивных нетерминалов получена.

Если в грамматике несколько несаморекурсивных нетерминалов, то устранять их нужно последовательно.

Устранение одного несаморекурсивного нетерминала может привести к тому, что некоторый несаморекурсивный нетерминал станет рекурсивным (в рассмотренном выше примере несаморекурсивный нетерминал A стал саморекурсивным)

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.

$A \to \alpha\beta_1$
$A \to \alpha\beta_2$

$A \to \alpha\beta_i$

$A \to \alpha\beta_n$

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

$A \to \alpha B$
$B \to \beta_1$
$B \to \beta_2$

$B \to \beta_i$

$B \to \beta_n$

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→abBa S→abBb S→abA B→bB B→b A→a	S→abC C→Ba C→Bb C→A B→bD D→B D→ε A→a	S→abC C→BE C→A E→a E→b B→bD D→B D→ε A→a

Пример 2.

Исходная грамматика	Первый шаг	Второй шаг
S→abBa S→abBb S→abA B→bB B→b A→a	S→abBE S→abA E→a E→b B→bD D→B D→ε A→a	S→abC C→BE C→A E→a E→b B→bD D→B D→ε A→a

Пример 3.

Исходная грамматика	Первый шаг	Второй шаг	Третий шаг
S→abBa S→abBb S→abA B→bB B→b A→a	S→aC C→bBa C→bBb C→bA B→bD D→B D→ε A→a	S→aC C→bE E→Ba E→Bb E→A B→bD D→B D→ε A→a	S→aC C→bE E→BF E→A F→a F→b B→bD D→B D→ε A→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 $A \to B$, где A и B – нетерминалы грамматики.

Для устранения цепных правил можно применять замену края:

пока есть цепные правила, выбрать цепное правило и применить замену края.

Такой алгоритм может зациклиться.

Например:

Исходная грамматика	Шаг 1	Шаг 2
$A \to B$ $A \to a$ $B \to C$ $B \to b$ $C \to B$ $C \to c$	$A \to C$ $A \to b$ $A \to a$ $B \to C$ $B \to b$ $C \to B$ $C \to c$	**$A \to B$** $A \to c$ $A \to b$ $A \to a$ $B \to C$ $B \to b$ $C \to B$ $C \to c$

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 $B \to C$ 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 $A \to \alpha$ из множества правил грамматики добавить правило $B \to \alpha$, если A принадлежит множеству M^B .

Пример.

Грамматика с цепными правилами:

1. $S \to S+T$ 4. $T \to E$
2. $S \to T$ 5. $E \to (S)$
3. $T \to T*E$ 6. $E \to a$

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

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

$M^T=\{E\}$ и

$M^E=\varnothing$.

2. Исключить из множества правил грамматики все цепные правила.

1. $S \to S+T$
3. $T \to T*E$
5. $E \to (S)$
6. $E \to a$

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

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

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

В результате получим:

1. $S \to S+T$
3. $T \to T*E$ 3_1. $S \to T*E$
5. $E \to (S)$ 5_1. $S \to (S)$ 5_2. $T \to (S)$
6. $E \to a$ 6_1. $S \to a$ 6_2. $T \to a$

Ещё пример.

$A \to B$
$A \to a$
$B \to C$
$B \to b$
$C \to B$
$C \to c$

Вычисляем множества M^A, M^B и M^C:
$M^A=\{B,C\}$
$M^B=\{C\}$
$M^C=\{B\}$

Исключаем из грамматики цепные правила:
$A \to a$
$B \to b$
$C \to c$

Добавляем правила:
$A \to a$
$B \to b$
$C \to c$
$A \to b$
$C \to b$
$A \to c$
$B \to c$

Исключим правила, содержащие недостижимые символы:
$A \to a$
$A \to b$
$A \to c$

Устранение левой рекурсии

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

Если $\alpha=\varepsilon$ и $\beta \neq \varepsilon$, 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. $A \to A\alpha$ (самолеворекурсивное)
2. $A \to \beta$ (несаморекурсивное)

Применяя эти правила, из A можно вывести:

$A \underset{2}{\Rightarrow} \beta$

$A \underset{1}{\Rightarrow} A\alpha \underset{2}{\Rightarrow} \beta\alpha$

$A \underset{1}{\Rightarrow} A\alpha \underset{1}{\Rightarrow} A\alpha\alpha \underset{2}{\Rightarrow} \beta\alpha\alpha$

$A \underset{1}{\Rightarrow} A\alpha \underset{1}{\Rightarrow} \ldots \Rightarrow A\alpha\ldots\alpha \underset{2}{\Rightarrow} \beta\alpha\ldots\alpha$

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

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

$$B \to \alpha B$$

$$B \to \varepsilon$$

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

$$A \to \beta B$$

$$B \to \alpha B$$

$$B \to \varepsilon$$

Предположим, что в грамматике для нетерминала A следующие правила:

1. $A \to A\alpha$

2. $A \to \beta_1$

3. $A \to \beta_2$

n+1. $A \to \beta_n$

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

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

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

Грамматика L(A) будет следующей:

$$A \to \beta_1 B$$

$$A \to \beta_2 B$$

$$A \to \beta_n B$$

$$B \to \alpha B$$

$$B \to \varepsilon$$

Предположим, что в грамматике для нетерминала A следующие правила:

1. $A \to A\alpha_1$

2. $A \to A\alpha_2$

m. $A \to A\alpha_m$

m+1. $A \to \beta_1$

m+2. $A \to \beta_2$

m+n. $A \to \beta_n$

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:

$$B \to \alpha_1 B$$

$$B \to \alpha_2 B$$

$$B \to \alpha_m B$$

$$B \to \varepsilon$$

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 несамолеворекурсивных правил. В итоге получим грамматику:

$A \to \beta_1 B$

$A \to \beta_2 B$

$A \to \beta_n B$

$B \to \alpha_1 B$

$B \to \alpha_2 B$

$B \to \alpha_m B$

$B \to \varepsilon$

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

$A \to \beta_j B$, где $1 \le j \le n$, B – новый нетерминал,

$B \to \alpha_i B$, где $1 \le i \le m$,

$B \to \varepsilon$

2. Исключение леворекурсивных правил.

Алгоритм исключения леворекурсивных правил.

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

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

3. Для каждого правила вида $A_i \to A_j \alpha$, где $1 \le j \le i-1$, 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:

$A_1 \to A_1aA_3$
$A_1 \to A_2b$
$A_2 \to A_1c$
$A_2 \to A_3a$
$A_3 \to A_1b$
$A_3 \to c$

Рассматриваем нетерминал A_1.

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

Исключаем самолеворекурсивное правило $A_1 \to A_1aA_3$, получаем грамматику:

$A_1 \to A_2bB_1$
$A_2 \to A_1c$
$A_2 \to A_3a$
$A_3 \to A_1b$
$A_3 \to c$
$B_1 \to aA_3B_1$
$B_1 \to \varepsilon$

Рассматриваем нетерминал A_2.

Выполняем замену края в правиле $A_2 \to A_1c$, получаем грамматику:

$A_1 \to A_2bB_1$
$A_2 \to A_2bB_1c$
$A_2 \to A_3a$
$A_3 \to A_1b$
$A_3 \to c$
$B_1 \to aA_3B_1$
$B_1 \to \varepsilon$

Исключаем самолеворекурсивное правило $A_2 \to A_2bB_1c$, получаем грамматику:

$A_1 \to A_2bB_1$
$A_2 \to A_3aB_2$
$A_3 \to A_1b$
$A_3 \to c$
$B_1 \to aA_3B_1$
$B_1 \to \varepsilon$
$B_2 \to bB_1cB_2$
$B_2 \to \varepsilon$

Рассматриваем нетерминал A_3.

Выполняем замену края в правиле $A_3 \to A_1b$, получаем грамматику:

$A_1 \to A_2bB_1$
$A_2 \to A_3aB_2$
$A_3 \to A_2bB_1b$
$A_3 \to c$
$B_1 \to aA_3B_1$
$B_1 \to \varepsilon$
$B_2 \to bB_1cB_2$
$B_2 \to \varepsilon$

Выполняем замену края в правиле $A_3 \to A_2bB_1b$, получаем грамматику:

$A_1 \to A_2bB_1$
$A_2 \to A_3aB_2$
$A_3 \to A_3aB_2bB_1b$
$A_3 \to c$
$B_1 \to aA_3B_1$
$B_1 \to \varepsilon$
$B_2 \to bB_1cB_2$
$B_2 \to \varepsilon$

Исключаем самолеворекурсивное правило $A_3 \to A_3aB_2bB_1b$, получаем грамматику:

$A_1 \to A_2bB_1$
$A_2 \to A_3aB_2$
$A_3 \to cB_3$
$B_1 \to aA_3B_1$
$B_1 \to \varepsilon$
$B_2 \to bB_1cB_2$
$B_2 \to \varepsilon$
$B_3 \to aB_2bB_1bB_3$
$B_3 \to \varepsilon$

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

$A_1 \to cB_3aB_2bB_1$
$A_3 \to cB_3$
$B_1 \to acB_3B_1$
$B_1 \to \varepsilon$
$B_2 \to bB_1cB_2$
$B_2 \to \varepsilon$
$B_3 \to aB_2bB_1bB_3$
$B_3 \to \varepsilon$

Устранение правой рекурсии

Правило $A \to \chi$ называется *рекурсивным*, если существует вывод $\chi \Rightarrow^* \alpha A \beta$. Если $\alpha \neq \varepsilon$ и $\beta = \varepsilon$, то правило $A \to \chi$ 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 $A \to \alpha_i A$, где $1 \leq i \leq m$, и n правил $A \to \beta_j$, где $1 \leq j \leq n$, которые не являются самоправорекурсивными и других правил с левой частью A нет. Эти правила заменяются следующими:

$A \to B\beta_j$, где $1 \leq j \leq n$, B – новый нетерминал,
$B \to B\alpha_i$, где $1 \leq i \leq m$,
$B \to \varepsilon$

2. Исключение праворекурсивных правил.

Алгоритм исключения праворекурсивных правил.

1. Обозначить нетерминалы грамматики $A_1, A_2, \ldots, A_n$, 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 $A_i \to \alpha A_j$, 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:

$A_1 \to aA_3A_1$
$A_1 \to bA_2$
$A_2 \to cA_1$
$A_2 \to aA_3$
$A_3 \to bA_1$
$A_3 \to c$

Рассматриваем нетерминал A_1.

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

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

$A_1 \to B_1bA_2$
$A_2 \to cA_1$
$A_2 \to aA_3$
$A_3 \to bA_1$
$A_3 \to c$
$B_1 \to B_1aA_3$
$B_1 \to \varepsilon$

Рассматриваем нетерминал A_2.

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

$A_1 \to B_1bA_2$
$A_2 \to cB_1bA_2$
$A_2 \to aA_3$
$A_3 \to bA_1$
$A_3 \to c$
$B_1 \to B_1aA_3$
$B_1 \to \varepsilon$

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

$A_1 \to B_1bA_2$
$A_2 \to B_2aA_3$
$A_3 \to bA_1$
$A_3 \to c$
$B_1 \to B_1aA_3$
$B_1 \to \varepsilon$
$B_2 \to B_2cB_1b$
$B_2 \to \varepsilon$

Рассматриваем нетерминал A_3.

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

$A_1 \to B_1bA_2$
$A_2 \to B_2aA_3$
$A_3 \to bB_1bA_2$
$A_3 \to c$
$B_1 \to B_1aA_3$
$B_1 \to \varepsilon$
$B_2 \to B_2cB_1b$

$B_2 \to \varepsilon$

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

$A_1 \to B_1bA_2$
$A_2 \to B_2aA_3$
$A_3 \to bB_1bB_2aA_3$
$A_3 \to c$
$B_1 \to B_1aA_3$
$B_1 \to \varepsilon$
$B_2 \to B_2cB_1b$
$B_2 \to \varepsilon$

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

$A_1 \to B_1bA_2$
$A_2 \to B_2aA_3$
$A_3 \to B_3c$
$B_1 \to B_1aA_3$
$B_1 \to \varepsilon$
$B_2 \to B_2cB_1b$
$B_2 \to \varepsilon$
$B_3 \to B_3bB_1bB_2a$
$B_3 \to \varepsilon$

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

$A_1 \to B_1bB_2aB_3c$
$B_1 \to B_1aB_3c$
$B_1 \to \varepsilon$
$B_2 \to B_2cB_1b$
$B_2 \to \varepsilon$
$B_3 \to B_3bB_1bB_2a$
$B_3 \to \varepsilon$

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 $A \to X\alpha$, 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 $C \to \alpha$, то
> правило $A \to X\alpha$ заменить на $A \to XC$, иначе ввести
> правило $N \to \alpha$, где N – новый нетерминал, и правило
> $A \to X\alpha$ заменить на $A \to XN$.

При выполнении п.1 учитывать вводимые правила.

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

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

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

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

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

4. Для каждого правила вида $A \to t_1 t_2$, 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
$A \to aBCd$	$A \to aN_1$	$A \to aN_1$	$A \to N_2N_1$
$A \to Cd$	$N_1 \to BCd$	$N_1 \to BA$	$N_2 \to a$
$B \to aB$	$A \to Cd$	$A \to Cd$	$N_1 \to BA$
$B \to b$	$B \to aB$	$B \to aB$	$A \to CN_3$
$C \to ABA$	$B \to b$	$B \to b$	$N_3 \to d$
$C \to ab$	$C \to ABA$	$C \to AN_1$	$B \to N_2B$
	$C \to ab$	$C \to ab$	$B \to b$
			$C \to AN_1$
			$C \to N_2N_4$
			$N_4 \to b$

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) $S \rightarrow \varepsilon$, 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 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 $A \to B\alpha$ в таком порядке, чтобы для любого правила выполнялось условие:

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

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

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 $A \to t$, то терминал t заменить нетерминалом A, если же в грамматике нет одиночного правило $A \to t$, 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
$S \to AC$	$B \to Sb$	$B \to Sb$	$B \to Sb$	$B \to aBCb$	$S \to aBC$
$C \to Aa$	$S \to AC$	$S \to AC$	$S \to aBC$	$B \to dCb$	$S \to dC$
$B \to Sb$	$C \to Aa$	$C \to aBa$	$S \to dC$	$S \to aBC$	$C \to aBa$
$B \to bB$	$B \to bB$	$C \to da$	$C \to aBa$	$S \to dC$	$C \to da$
$A \to aB$	$A \to aB$	$B \to bB$	$C \to da$	$C \to aBa$	$B \to aBCb$
$A \to d$	$A \to d$	$A \to aB$	$B \to bB$	$C \to da$	$B \to dCb$
		$A \to d$	$A \to aB$	$B \to bB$	$B \to bB$
			$A \to d$	$A \to aB$	$A \to aB$
				$A \to d$	$A \to d$

При выполнении п.3 алгоритма вводим два одиночных правила $D \to b$ и $E \to a$ 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:

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

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 $A \to \alpha\beta\chi$,

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 $B \to \beta$, то правило $A \to \alpha\beta\chi$ заменить на $A \to \alpha B\chi$, иначе ввести правило $N \to \beta$, где N – новый нетерминал, и правило $A \to \alpha\beta\chi$ заменить на $A \to \alpha N\chi$.

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

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
$A \rightarrow aBAC$	$A \rightarrow aN_1$	$A \rightarrow aN_1$
$A \rightarrow e$	$A \rightarrow e$	$A \rightarrow e$
$B \rightarrow dBACb$	$B \rightarrow dN_1b$	$B \rightarrow dN_1b$
$B \rightarrow b$	$B \rightarrow b$	$B \rightarrow b$
$C \rightarrow aBaAB$	$C \rightarrow aBaN_2$	$C \rightarrow aBaN_2$
	$N_1 \rightarrow BAC$	$N_1 \rightarrow BaN_1aBaN_2$
	$N_2 \rightarrow AB$	$N_1 \rightarrow BeC$
		$N_2 \rightarrow AdN_1b$
		$N_2 \rightarrow Ab$