**ИТМО Кафедра Информатики и прикладной математики**

Домашнее задание №3

«Приведение грамматик»   
Вариант 7

**Выполнил: студент группы P3217**

**Плюхин Дмитрий**

**Преподаватель: Лаздин А.В.**

**2017 год**

1. **Задание**

Для каждого пункта задания (всего четыре) необходимо представить: исходную грамматику, пошаговый процесс построения множества производящих и достижимых символов грамматики, поэтапное формирование новых правил при удалении цепных и ε-правил, грамматику полученную в результате преобразования (приведения).

* Преобразовать КС грамматику в эквивалентную грамматику, не содержащую бесполезных(непроизводящих и недостижимых) символов.
* Удалить из КС грамматики ε-правила.
* Преобразовать КС грамматику в эквивалентную, не содержащей цепных правил.
* Найти приведенную форму КС-грамматики

1. **Удаление бесполезных символов**

Исходная грамматика:

S → aABC | aE

A → SCD | c

B → bFD | b

C → aE

D → aD

E → aCE

F → AB

Последовательно просматриваем правила первый раз для составления множества производящих символов:

A – первый символ, который можно отнести к производящим, потому что присутствует правило, в котором он порождает терминал.

B – второй символ, относящийся к производящим, по аналогичной причине, что и A.

F – третий символ, он порождает цепочку, состоящую только из производящих символов.

На текущий момент множество производящих символов выглядит как {A, B, F}

Последовательно просматриваем правила второй раз и делаем вывод, что больше нельзя посчитать ни один символ производящим, так как в правой части каждого оставшегося правила имеется хотя бы один непроизводящий символ. Это говорит о том, что все производящие символы были выделены на предыдущем шаге.

Грамматика, из которой удалены непроизводящие символы, выглядит как (символ F заменен на S)

S → AB

A → c

B → b

Последовательно просматриваем правила первый раз для составления множества достижимых символов:

Символ S считается достижимым изначально. В правой части первого правила содержатся оба оставшихся нетерминала (которые не являются символом S), поэтому ни одно правило не может быть исключено при выделении достижимых символов.

Конечная грамматика:

S → AB

A → c

B → b

1. **Удаление ε - правил**

Исходная грамматика

S → 1A | B0

A → 1A | C

B → B0 | C

C → 1C0 | ε

Посмотрим, какие преобразования можно произвести:

S => 1A => 11A => 111A => 1nA => 1n1C0 => 1n11C00 => 1n 1m C0m => 1n+m 0m

Поскольку n >= 1, а m >= 0, то можно записать это следующим образом:

S => 1n 0m, где n > m, m >= 0

Аналогично построим альтернативную комбинацию:

S => B0 => B00 => B000 => B0n => 1C00n => 11C000n => 1m C0m0n => 1m0n+m

S => 1m 0n, где n > m, m >= 0

Объединением всех возможных случаев будет запись

S => 1a0b, где a <> b, a >= 0, b >= 0

Её можно задать неукорачивающей грамматикой.

Конечная грамматика:

S → 1A0 | 1B0 | A | B

A → 0 | A

B → 1 | B

Эта грамматика порождает язык {1n 0m 0n, 1n 1m 0n, n >=0, m >= 1} = {1n 0n+m, 1n+m 0n, n >=0, m >= 1}={1n 0m, n>= 0, m >=0, n<>m}, эквивалентный требуемому

1. **Удаление цепных правил**

Исходная грамматика

S → A | B

A → 1A0 | 1a0

B → 1B00 | 1b00

Для удаления цепных правил проделаем нехитрый алгоритм:

Разделим все правила на два множества, в первое из которых попадут правила вида N → M:

P1 = {S → A, S → B}

P2 = {A → 1A0, A → 1a0, B → 1B00, B → 1b00}

Поочередно будем рассматривать каждое правило из первого множества, и пытаться найти правила из второго множества с таким же символом в левой части, что и у правила в первом множестве в правой части:

Правилу S → A соответствуют A → 1A0, A → 1a0, поэтому в грамматику добавятся следующие новые правила: S → 1A0, S → 1a0.

Правилу S → B соответствуют B → 1B00, B → 1b00, поэтому в грамматику добавятся следующие новые правила: S → 1B00, S → 1b00.

Конечная грамматика

S → 1B00 | 1b00 | 1A0 | 1a0

A → 1A0 | 1a0

B → 1B00 | 1b00

1. **Найти приведенную форму КС-грамматики**

Исходная грамматика

S → A | B

C → S | a | ε

A → C | D

D → S | b

B → D | E

E → S | c | ε

1. Удаление непроизводящих символов:

Последовательно просматриваем все правила в грамматике первый раз:

С – производящий символ, потому что порождает терминал a, A – производящий символ, потому что порождает производящий символ C, D – производящий символ, потому что порождает терминал b, B – производящий символ, потому что порождает производящий символ D, E – производящий символ, потому что порождает терминал c.

Последовательно просматриваем все правила в грамматике второй раз:

S – производящий символ, потому что порождает производящий символ A, следовательно, все символы грамматики являются производящими.

1. Удаление недостижимых символов:

Символ S достижим изначально, следовательно символы A и B тоже достижимы, следовательно, символы C, D, E тоже достижимы, следовательно, все символы грамматики достижимы.

1. Удаление ε - правил:

Просматриваем комбинации, которые могут получиться (приведены только те, которые представляют интерес):

S → A → C → a

S → A → C → ε

S → A → D → b

S → B → E → c

S → B → E → ε

Становится очевидно, что грамматика может быть заменена одним простым правилом

S → a | b | c | ε

1. Удаление цепных правил:

В полученной грамматике отсутствуют правила вида N → M, следовательно, этот шаг не имеет смысла.

Конечная грамматика

S → a | b | c | ε