**Санкт-Петербургский национальный исследовательский университет**

**информационных технологий, механики и оптики**

**Кафедра вычислительной техники**

Формальные языки и грамматики

Домашние задание №3

Приведение грамматик

Выполнил: Владимир Бакшенов

Группа P3218

Преподаватель: Лаздин А.В.

2017 г.

**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

**Удаление бесплодных 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

**Удаление ε – правил**

Исходная грамматика

S → ASB | ε

A → aAS | a

B → SbS | A | bb

Посмотрим, какие преобразования можно провести:

Учитывая, что начальный символ, образующий аннулирующее правило, входит в правые части других правил грамматики, заменим правило S → ε правилами вида S` → ε и S` → S.

На основе вышеприведенных преобразований зададим неукорачивающую грамматику.

Конечная грамматика:

S` → S | ε

S → AB

A → aAS | a

B → SbS | A | bb

**Удаление цепных правил**

Исходная грамматика:

S → LA | LB

L → P:= | Q:=

P → i

A → F

Q → i

B → F

F → Q(i)

Для удаления цепных правил исполним алгоритм:

Разделим все правила на два множества, в первое из которых попадут правила вида A → B:

P1 = {A → F, B → F}

P2 = {S → LA | LB L → P:= | Q:=, P → i, Q → I, F → Q(i)}

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

Правилу A → F соответствуют F → Q(i), поэтому в грамматику добавятся следующее новое правило: A → Q(i).

Правилу B → F соответствуют F → Q(i), поэтому в грамматику добавятся следующее новое правило: B → Q(i).

Наша грамматика имеет три нетерминала A, B и F, каждый из которых порождает одинаковые цепочки. Устраним их, заменив все три нетерминалом A.

Конечная грамматика:

S → LA

L → P:= | Q:=

P → i

Q → i

A → Q(i)

**Найдите приведенную форму КС-грамматики**

Исходная грамматика:

S → a | A

B → b

A → AB

C → Sa | ε

**1. Удаление непроизводящих символов**

Последовательно просматриваем все правила в грамматике первый раз:

S – производящий символ, потому что порождает терминал a.

B – производящий символ, потому что порождает терминал b.

C – производящий символ, потому что порождает производящую цепочку Sa.

Последовательно просматриваем все правила в грамматике 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 так:

S → a

B → b

C → Sa | ε

**2. Удаление недостижимых символов**

Символ S достижим изначально. Из него выводится терминал, следовательно, все остальные нетерминалы недостижимы.

Конечная грамматика:

S → a

**3. Удаление ε – правил**

Схема грамматики не содержит аннулирующих правил, следовательно удалеять ε – правил не требуется.

**4. Удаление цепных правил**

В полученной грамматике отсутствуют правила вида A → B, следовательно, удалять цепные правила не требуется.

Конечная грамматика:

S → a