Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики

Кафедра информатики и прикладной математики

**Домашняя работа №3**

**Дисциплина «Прикладная математика»**

**Вариант 21**

**Выполнил:**

Съестов Дмитрий Вячеславович

Группа P3317

**Преподаватель:**

Лаздин Артур Вячеславович

Санкт-Петербург

2018

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).

|  |  |
| --- | --- |
| 1.8 | S → b | C | cCB A → Ab | ε B → Bb | cB C → Ca | Bc |d |

Преобразовать КС грамматику в эквивалентную грамматику, не содержащую бесполезных (непроизводящих и недостижимых) символов.

1. Удаление непроизводящих символов:

VN’1 = {S, A, C}

S → b | C  
A → Ab | ε   
C → Ca | d

1. Удаление недостижимых символов:

VN’1 = {S}

VN’2 = {S, C}

S → b | C  
C → Ca | d

|  |  |
| --- | --- |
| 2.6 | S → dAa | ε A → aSB | dSc B → b | ε |

Удалить из КС грамматики ε-правила.

1. Nullable = {S, B}

2. Удаляем правила вида A → ε:

S → dAa

A → aSB | dSc

B → b

3. Т.к. S ∈ Nullable, возвращаем правило S → ε:

S → dAa | ε

A → aSB | dSc

B → b

4. Делаем замену для всех нетерминалов в множестве Nullable:

S → dAa | ε

A → aSB | aS | aB | a | dSc | dc

B → b

|  |  |
| --- | --- |
| 3.10 | S → @nL | @mM L → M | Lab | Lcm | ε |

Преобразовать КС грамматику в эквивалентную, не содержащую цепных правил.

Присутствует цепное правило L → M.

Поскольку в данной грамматике из символа М не выводится ничто, удалить это цепное правило нельзя.

|  |  |
| --- | --- |
| 4.3 | S → aB| bA | cC A → cBS| bA | C | b | ε B → bSA | cCb |S C → Cd | aCa |

Найдите приведенную форму КС-грамматики.

1. Удаление непроизводящих символов:

VN’1 = {A}

VN’2 = {A, S}

VN’3 = {A, S, B}

S → aB | bA  
A → cBS| bA | b | ε  
B → bSA | S

1. Удаление недостижимых символов:

VN’1 = {S}

VN’2 = {S, A, B}

(без изменений)

1. Удаление ε-правил:
2. Nullable = {A}
3. Удаляем правила вида A → ε:

S → aB | bA  
A → cBS| bA | b  
B → bSA | S

1. S ∉ Nullable, пропускаем этот шаг
2. Делаем замену:

S → aB | bA | b  
A → cBS| bA | b  
B → bSA | bS | S

4. Удаление цепных правил:

Цепное правило: B → S

Заменяем правую часть на все возможные правые части правил, где в левой части стоит S:

S → aB | bA | b  
A → cBS| bA | b  
B → bSA | bS | aB | bA | b

Получена приведённая форма грамматики.