МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

**Факультет программной инженерии и компьютерной техники**

**Домашнее задание №1**

по дисциплине

“Разработка компиляторов”

**Вариант № 9**

**Студент:**

Румский А.

Группа P3307

**Преподаватель:**

Лаздин Артур Вячеславович

Санкт-Петербург, 2025 год

## Пункт 1.

**Задача:** для каждой грамматики из списка [9, 18, 38] определить тип по классификации Хомского. Для грамматик типа 2|3 привести вывод не менее двух предложений языка, принадлежащий порождаемому языку.

### Грамматика 1.

**Анализ:**

Грамматика не соответствует ограничениям типа 3, так как, как минимум, присутствует рекурсия () В данной грамматике слева присутствуют только одиночные не-терминалы, справа же набор из терминалов и не-терминалов. Это соответствует типу 2.

**Вывод: Грамматика второго типа (контекстно-свободная)**

**Предложения языка:**

1.   
Возьмем:

Применим правило:

Получим:

2.

Возьмем:

Применим правило, выбрав:

Сохраняя результат, применим правило:

Подставим в , получим:

### Грамматика 2.

**Анализ:** Грамматика соответствует всем параметрам для грамматики 3 типа: один нетерминальный символ слева; не более одного нетерминального символа, с одной и той же стороны, от цепочки терминалов во всех правилах одной грамматики справа.

**Вывод: Грамматика третьего типа (регулярная)**

**Предложения языка:**

1.

Возьмём

Получим

Применим

Получим

Применим

Получим

Применим

Получим

2.

Возьмём

Получим

Применим

Получим

Применим

Получим

Применим

Получим

### Грамматика 3.

Множество строк четной длины, которые содержат ровно один a.

**Анализ:** есть не-терминал в середине (), следовательно максимум второй тип. Все остальные условия для второго типа выполнены.

**Вывод: Грамматика второго типа(контекстно-свободная)**

**Предложения языка:**

1.

Возьмём

Получим

Применим

Получим

2.

Возьмём

Получим

Применим

Получим

Применим

Получим

Применим

Получим

Код для GraphViz

digraph DKA {

rankdir=LR;

// Состояния

q0 [label="q0 (S)" shape=circle];

q1 [label="q1 (X)" shape=circle];

q2 [label="q2 (X)" shape=circle];

q3 [label="q3 (F)" shape=circle];

q4 [label="q4 (E)" shape=doublecircle];

// Переходы

q0 -> q1 [label="a"];

q1 -> q2 [label="b, c"];

q2 -> q3 [label="b, c"];

q2 -> q4 [label="ε"];

q3 -> q2 [label="b, c"];

}

Код для проверки (Python)