МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Факультет программной инженерии и компьютерной техники

Домашняя работа №1

Вариант 10

Выполнил:

Круглов Егор Ильич, P3324

Преподаватель:

Лаздин Артур Вячеславович

Санкт-Петербург

2025

Задача №10:

$$S \rightarrow aASB \mid \varepsilon$$

$$A \rightarrow ad \mid d$$

$$B \rightarrow bb$$

Тип грамматики: 2

Примеры предложений:

1) ε
2) aadbb
3) adbb

Множественно-теоретический вид:

L(G) = $\{(aad \mid ad)^k (bb)^n \mid k, n \geq 1\} \cup \{\varepsilon\}$.

Задача №19:

$$S \rightarrow A \mid SA \mid SB$$

$$A \rightarrow a$$

$$B \rightarrow b$$

Тип грамматики: 2

Примеры предложений:

1) a
2) aa
3) ab

Множественно-теоретический вид:

$L(G) = \{a(a|b)^k \mid k \geq 0\}$

Задача №39:

Множество строк с нечетным числом вхождений подстроки ab.

$$A = \{a, b, c\}$$

Грамматика:

$S_0 \rightarrow aS_1 \mid bS_0 \mid cS_0$

$S_1 \rightarrow aS_1 \mid bS_2 \mid cS_0$

$S_2 \rightarrow aS_3 \mid bS_2 \mid cS_2 \mid \varepsilon$

$S_3 \rightarrow aS_3 \mid bS_0 \mid cS_2 \mid \varepsilon$

Опишем 4 состояния S_0, S_1, S_2, S_3:

S_0 – четное вхождение подстрок "ab", нет «подвешенной» (крайний правый символ) - "a".

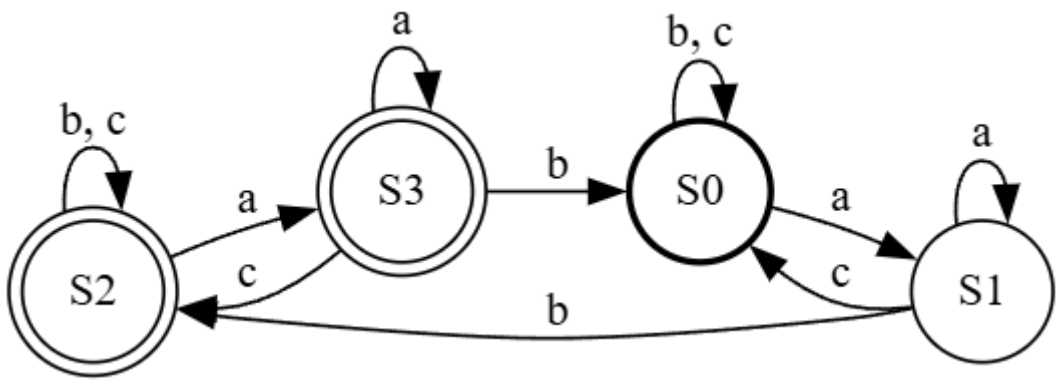
S_1 – четное вхождение подстрок "ab", есть «подвешенная» (крайний правый символ) - "a".

S_2 – нечетное вхождение подстрок "ab", нет «подвешенной» (крайний правый символ) - "a".

S_3 – нечетное вхождение подстрок "ab", есть «подвешенная» (крайний правый символ) - "a".

Переходы получаются простой логикой. Для S_2 и S_3 добавили переход → ε для выхода, так как нас удоволетворяют данные состояния (нечетное число вхождений подстроки "ab").

ДКА:



Код:

Ссылка на github: https://github.com/KruglovEgor/Compilers/blob/main/lab1/main.py

```
def simulate_dfa(input_str):
  """
  Моделирует работу ДКА для языка с нечетным числом вхождений подстроки 'ab'
  Алфавит: {a, b, c}

  Состояния:
   S0: чётное число 'ab', нет подвешенного 'a'
   S1: чётное число 'ab', есть подвешенное 'a'
   S2: нечетное число 'ab', нет подвешенного 'a'
   S3: нечетное число 'ab', есть подвешенное 'a'

```

```
    Допускающие состояния: S2 и S3
    """
    # Начальное состояние: S0
    state = 'S0'

    # Переходы
    transitions_dic = {
      'S0': {
        'a': 'S1',
        'b': 'S0',
        'c': 'S0'
      },
      'S1': {
        'a': 'S1',
        'b': 'S2',
        'c': 'S0'
      },
      'S2': {
        'a': 'S3',
        'b': 'S2',
        'c': 'S2'
      },
      'S3': {
        'a': 'S3',
        'b': 'S0',
        'c': 'S2'
      }
    }

    # Допускающие состояния
    exit_states = ('S2', 'S3')

    for ch in input_str:
      possible_transitions = transitions_dic[state]
      if ch in possible_transitions:
        state = possible_transitions[ch]
      else:
        # Неизвестный символ (не из нашего алфавита)
        return False

    return state in exit_states


# Тестирование автомата
test_strings = {
  "valid": [
    "ab",
```

```
    "cab",
    "aab",
    "abababab"
  ],
  "invalid": [
    "",
    "abab",
    "acabcab",
    "acac"
  ]
}

print("Тестирование корректных цепочек:")
for s in test_strings["valid"]:
  result = simulate_dfa(s)
  print(f'"{s}" -> {result}')

print("\nТестирование некорректных цепочек:")
for s in test_strings["invalid"]:
  result = simulate_dfa(s)
  print(f'"{s}" -> {result}')
```

Вывод в консоль: