

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

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

Факультет программной инженерии и компьютерной техники

Домашнее задание №2
по дисциплине
“Разработка компиляторов”
Вариант № 9

Студент:

Румский А.

Группа Р3307

Преподаватель:

Лаздин Артур Вячеславович

Санкт-Петербург, 2025 год

Оглавление

Вариант	3
Задание	3
НКА	4
ДКА	4
Минимизация ДКА	5
Код	7

Вариант

9	$(a b c)^*(a b)^*b$	23
---	---------------------	----

Для самопроверки использовался [этот сайт](#)

Задание

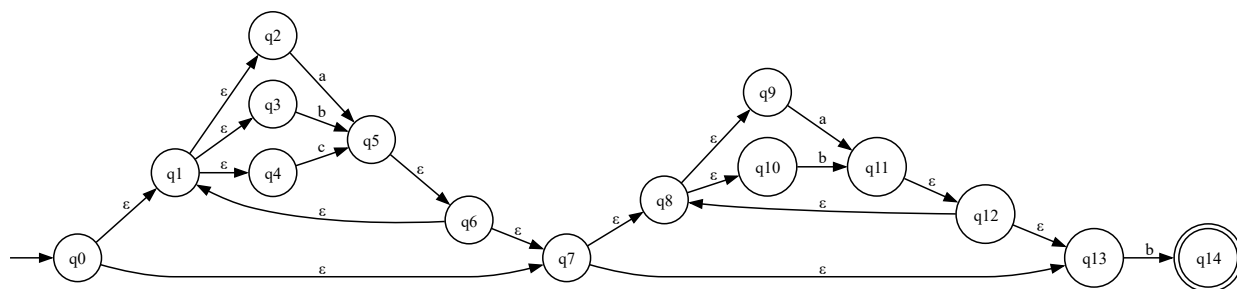
По заданному регулярному выражению (см. вариант)

- Построить недетерминированный КА;
- По полученному НДА построить ДКА;
- Минимизировать полученный ДКА;
- Для минимального ДКА написать программу-распознаватель предложений языка, порождаемого регулярным выражением.

Продемонстрировать работу распознавателя на различных примерах (не менее

трех правильных) предложений.

НКА



ДКА

Начальное состояние ДКА:

$\epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2, q_3, q_4, q_6, q_7, q_8, q_9, q_{10}, q_{12}, q_{13}\}$.

Назовем это состояние **S0**.

Построение таблицы переходов ДКА:

Для **S0** = {q0, q1, q2, q3, q4, q6, q7, q8, q9, q10, q12, q13} (не конечное):

$\delta(S_0, a): \epsilon\text{-closure}(\text{move}(S_0, a)) = \epsilon\text{-closure}(\{q_5, q_{11}\}) = \{q_1..q_{13}\}$. Назовем **S1**.

$\delta(S_0, b): \epsilon\text{-closure}(\text{move}(S_0, b)) = \epsilon\text{-closure}(\{q_5, q_{11}, q_{14}\}) = \{q_1..q_{14}\}$.

Назовем **S2**. **S2** содержит q14, значит это **конечное состояние**.

$\delta(S_0, c): \epsilon\text{-closure}(\text{move}(S_0, c)) = \epsilon\text{-closure}(\{q_5\}) = \{q_1..q_{10}, q_{12}, q_{13}\}$.

Назовем **S3**.

Для **S1** = {q1..q13} (не конечное):

$\delta(S_1, a): \epsilon\text{-closure}(\text{move}(S_1, a)) = \epsilon\text{-closure}(\{q_5, q_{11}\}) = S_1$.

$\delta(S_1, b): \epsilon\text{-closure}(\text{move}(S_1, b)) = \epsilon\text{-closure}(\{q_5, q_{11}, q_{14}\}) = S_2$.

$\delta(S_1, c): \epsilon\text{-closure}(\text{move}(S_1, c)) = \epsilon\text{-closure}(\{q_5\}) = S_3$.

Для **S2** = {q1..q14} (конечное):

$\delta(S_2, a): \epsilon\text{-closure}(\text{move}(S_2, a)) = \epsilon\text{-closure}(\{q_5, q_{11}\}) = S_1$.

$\delta(S_2, b): \epsilon\text{-closure}(\text{move}(S_2, b)) = \epsilon\text{-closure}(\{q_5, q_{11}, q_{14}\}) = S_2$.

$\delta(S_2, c): \epsilon\text{-closure}(\text{move}(S_2, c)) = \epsilon\text{-closure}(\{q_5\}) = S_3$.

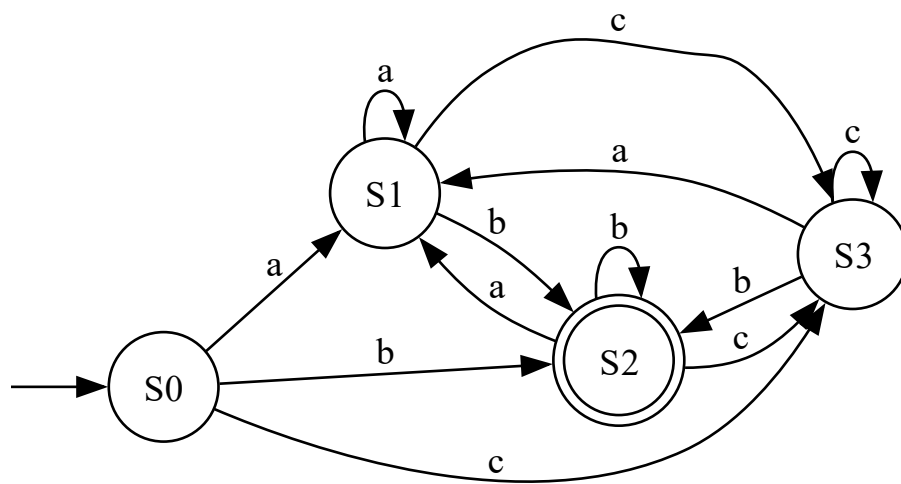
Для **S3** = {q1..q10, q12, q13} (не конечное):

$\delta(S_3, a): \epsilon\text{-closure}(\text{move}(S_3, a)) = \epsilon\text{-closure}(\{q_5, q_{11}\}) = S_1$.

$\delta(S3, b): \varepsilon\text{-closure}(\text{move}(S3, b)) = \varepsilon\text{-closure}(\{q5, q11, q14\}) = S2.$

$\delta(S3, c): \varepsilon\text{-closure}(\text{move}(S3, c)) = \varepsilon\text{-closure}(\{q5\}) = S3.$

Состояние	a	b	c
S0 (S)	S1	S2	S3
S1	S1	S2	S3
S2 (F)	S1	S2	S3
S3	S1	S2	S3



Минимизация ДКА

Начальное разбиение:

$G1 = \{S2\}$ (конечные)

$G2 = \{S0, S1, S3\}$

Проверка эквивалентности состояний:

Состояние из G2	Переход по a	Переход по b	Переход по c
S0	S1 (в G2)	S2 (в G1)	S3 (в G2)
S1	S1 (в G2)	S2 (в G1)	S3 (в G2)
S3	S1 (в G2)	S2 (в G1)	S3 (в G2)

Состояния неразличимы и их можно объединить в одно состояние.

Объединяем $\{S0, S1, S3\} \Rightarrow A.$

A становится начальным состоянием.

Состояние $\{S2\} \Rightarrow B.$

B становится конечным.

Переходы для A:

$\delta(A, a)$: из S_0 по $a \Rightarrow S_1$. S_1 теперь является частью A .
Значит, $\delta(A, a) = A$.

$\delta(A, b)$: из S_0 по $b \Rightarrow S_2$. S_2 теперь является B .
Значит, $\delta(A, b) = B$.

$\delta(A, c)$: из S_0 по $c \Rightarrow S_3$. S_3 теперь является частью A .
Значит, $\delta(A, c) = A$.

Переходы для B:

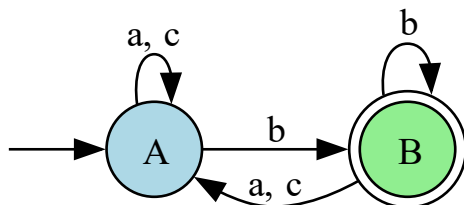
$\delta(B, a)$: из S_2 по $a \Rightarrow S_1$. S_1 теперь является частью A .
Значит, $\delta(B, a) = A$.

$\delta(B, b)$: из S_2 по $b \Rightarrow S_2$. S_2 теперь является B .
Значит, $\delta(B, b) = B$.

$\delta(B, c)$: из S_2 по $c \Rightarrow S_3$. S_3 теперь является частью A .
Значит, $\delta(B, c) = A$.

Итоговая таблица переходов минимального ДКА:

Состояние	a	b	c
A (S)	A	B	A
B (F)	A	B	A



Код

```
def create_recognizer_for_regex():
    STATE_A = "A (не заканчивается на b)"
    STATE_B = "B (заканчивается на b)"

    initial_state = STATE_A
    final_states = {STATE_B}
    alphabet = {'a', 'b', 'c'}

    transitions = {
        STATE_A: {
            'a': STATE_A,
            'b': STATE_B,
            'c': STATE_A
        },
        STATE_B: {
            'a': STATE_A,
            'b': STATE_B,
            'c': STATE_A
        }
    }

    def recognizer(input_string: str) -> bool:
        current_state = initial_state
        print(f"\nПроверка строки: '{input_string}'")
        print(f"Начальное состояние: {current_state}")

        for char in input_string:
            if char not in alphabet:
                print(f"Ошибка: символ '{char}' не принадлежит алфавиту
{{a, b, c}}.")
                return False

            previous_state = current_state
            current_state = transitions[current_state][char]
            print(f"Символ: '{char}', Переход: {previous_state} ->
{current_state}")

            is_accepted = current_state in final_states
            print(f"Конечное состояние: {current_state}")
            print(f"Результат: Строка {'принята' if is_accepted else
'отклонена'}.")
            return is_accepted

    return recognizer

if __name__ == "__main__":
    my_recognizer = create_recognizer_for_regex()

    print("---- Тестирование на правильных строках ----")
    my_recognizer("b")
    my_recognizer("ab")
    my_recognizer("cacabb")
    my_recognizer("bb")
```

```
print("\n--- Тестирование на неправильных строках ---")
my_recognizer("a")
my_recognizer("bca")
my_recognizer("")
my_recognizer("d")
```