

**Университет ИТМО**

**Кафедра ИПМ**

**Проектирование языков программирования и языков представления данных**

**Домашнее задание №2**

**«Формальные грамматики и языки»**

**Вариант 14**

Выполнил студент группы Р4117:

Пьянков Е.В.

Преподаватель: Лаздин А.В.

Санкт-Петербург

2018

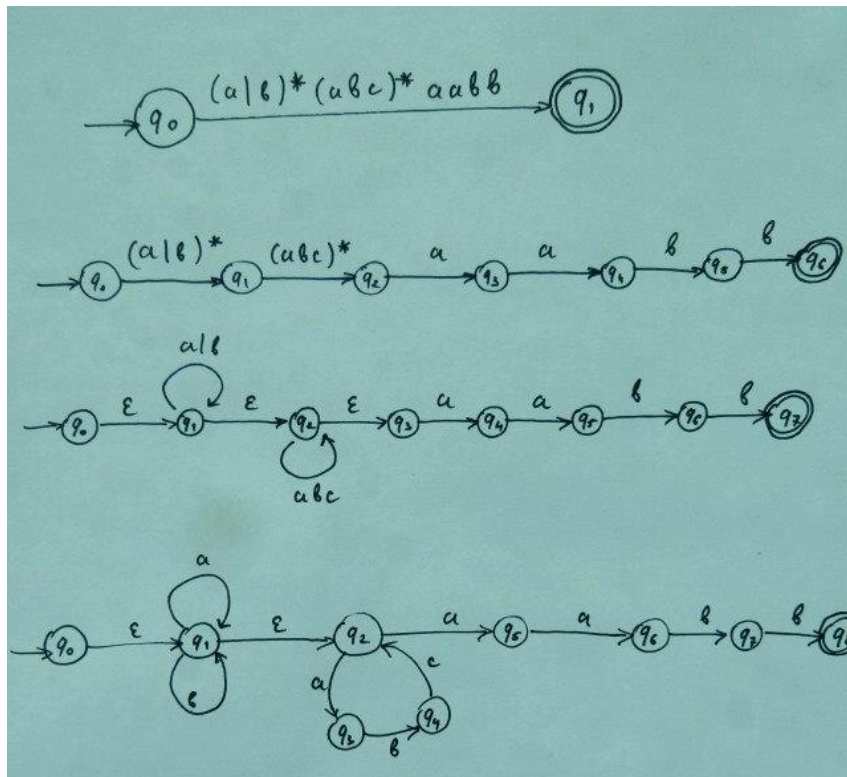
## Задание

По заданному регулярному выражению (см. вариант)

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

Продемонстрировать работу распознавателя на различных примерах (не менее трех правильных) предложений.

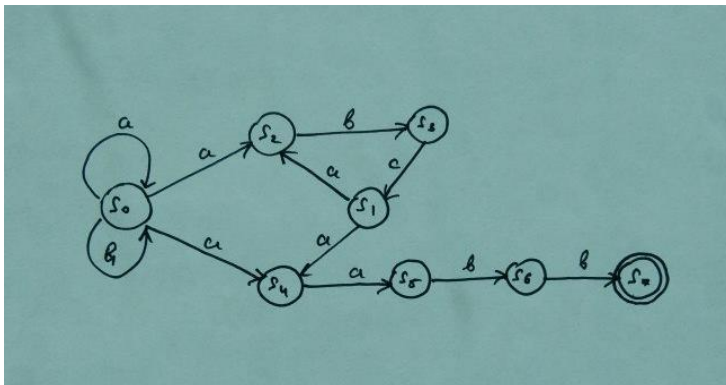
**$(a|b)^*(abc)^*aabb$**



Избавляемся от E-переходов:

	a	b	c
s0	q1, q3, q5	q1	-
s1	q1, q3, q5	q1	-
s2	q3, 5	-	-
s3	-	q4	-
s4	-	-	q2
s5	q6	-	-
s6	-	q7	-
s7	-	q8	-
s8	-	-	-

Строки 1 и 2 одинаковые - сокращаем количество состояний.



Строим ДКА. Так, например, если в НКА из состояния s0 по a можно перейти в состояния s0, s2 и s4, то к ДКА добавляем состояние p1 (s0, s2, s4).

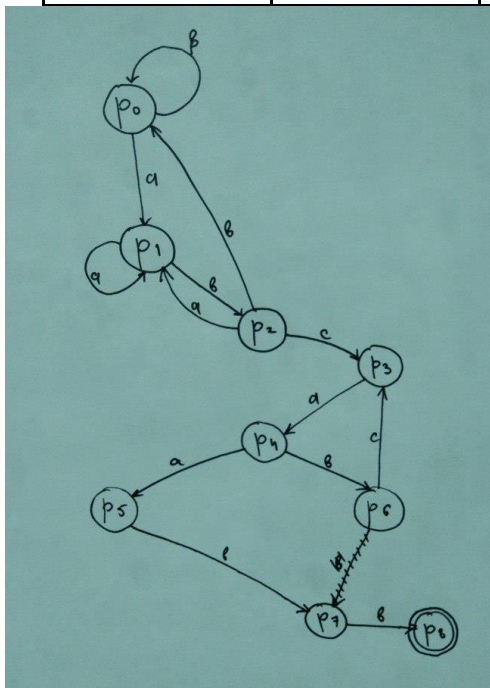
		a	b	c
--	--	---	---	---

p0	s0	p1	p0	-
p1	S0, s2, s4	p2	p3	-
p2	S0, s2, s4, s5	p2	p4	-
p3	S0, s3	p1	p0	p5
p4	S0, s3, s6	p1	p6	p5
p5	s1	p7	-	-
p6	S0, s7	p1	p0	-
p7	S2, s4	p8	p9	-
p8	s5	-	p10	-
p9	s3	-	-	p5
p10	s6	-	p11	-
p11	s7	-	-	-

Сокращаем одинаковые состояния:

		a	b	c
p0	s0	p1	p0	-
p1	S0, s2, s4	p1	p2	-
p2	S0, s3	p1	p0	P3
P3	s1	P4	-	-

P4	S2, s4	P5	P6	-
P5	s5	-	P7	-
P6	s3	-	-	P3
P7	s6	-	p8	-
p8	s7	-	-	-



Программа-распознаватель предложений языка:

```

public class DFSM {

    public boolean check(String input){
        int state = 0;
        for (int i = 0; i < input.length(); i++) {
            char letter = input.charAt(i);
            switch(state){
                case 0:
                    switch (letter){
                        case 'a':
                            state = 1;
                            break;
                        case 'b':

```

```

        state = 0;
        break;
    default:
        return false;
    }
    break;
case 1:
    switch (letter){
        case 'a':
            state = 1;
            break;
        case 'b':
            state = 2;
            break;
        default:
            return false;
    }
    break;
case 2:
    switch (letter){
        case 'a':
            state = 1;
            break;
        case 'b':
            state = 0;
            break;
        case 'c':
            state = 3;
            break;
        default:
            return false;
    }
    break;
case 3:
    switch (letter){
        case 'a':
            state = 4;
            break;
        default:
            return false;
    }
    break;
case 4:
    switch (letter){
        case 'a':
            state = 5;
            break;
        case 'b':
            state = 6;
            break;
        default:
            return false;
    }
    break;
case 5:
    switch (letter){
        case 'b':
            state = 7;
            break;
        default:
            return false;
    }
    break;
case 6:

```

```

        switch (letter){
            case 'c':
                state = 3;
                break;
            default:
                return false;
        }
        break;
    case 7:
        switch (letter){
            case 'b':
                state = 8;
                break;
            default:
                return false;
        }
        break;
    default:
        return false;
    }
}
return true;
}
}

```

Цепочка	Результаты работы программы
ababcaabb	true
caaabb	false
abcaabb	true
abcabcaabb	true
caabb	false