

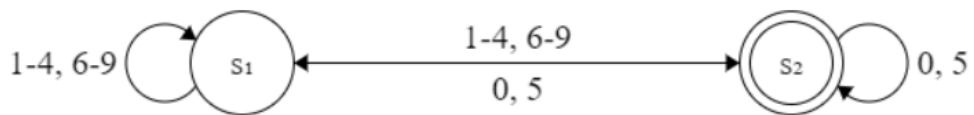
Формальные языки

Громов Федор

13 сентября 2021 г.

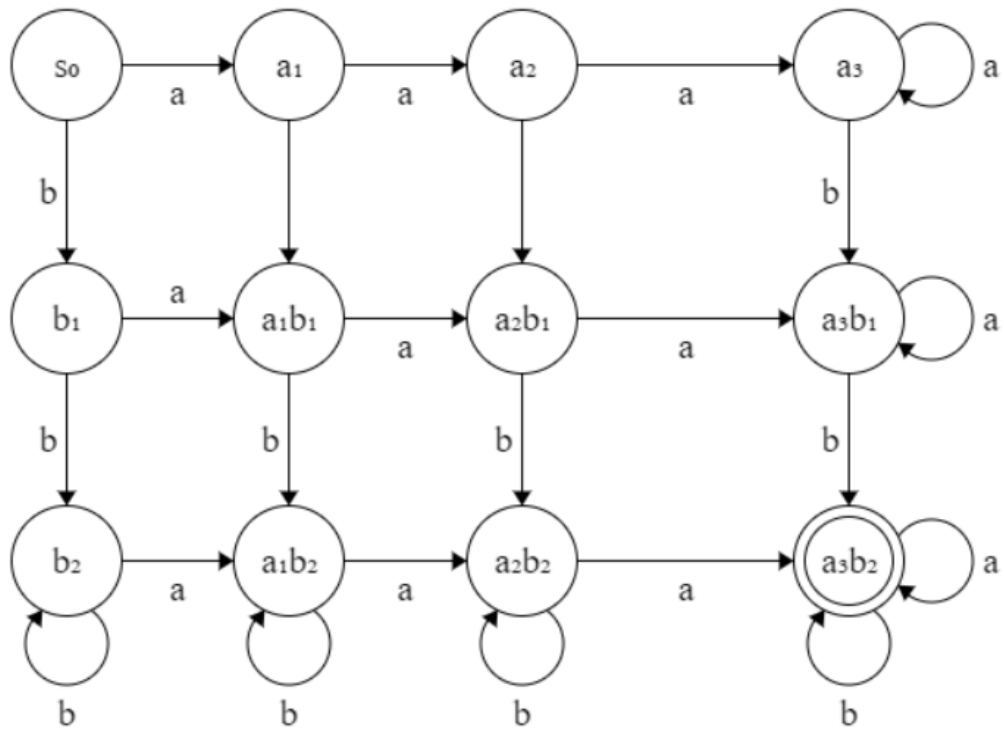
1.

s_1 - не делится, s_2 - делится



2.

s_0 - начальное



3.

Нашел документацию своего второго самого любимого языка программирования, а именно — [PascalABC](#). Открыл для себя много нового, например то, что идентификаторами (переменных, функций, и т.п.) могут служить последовательности латинских букв и цифр, начинающиеся с буквы, причем символ "_" — считается буквой.

Также приятно удивило наличие контекстно ключевых слов в данном прекрасном Тьюринг-полном языке, таких как например *read* и *write*, которые можно использовать в качестве идентификаторов в некоторых контекстах.

4.

Конечный автомат - упорядоченная пятерка множеств. Будем отделять множества фигурными скобками, а элементы множества в общем случае — переносом строки, в описании функции перехода — первое состояние, перенос строки, символ, по которому переходим, перенос строки, состояние, в которое переходим. Приведем описание автомата из первой задачи на нашем лаконичном и аккуратном языке:

```
1      { // Alphabet
2          0
3          1
4          2
5          3
6          4
7          5
8          6
9          7
10         8
11         9
12     }
13     { // Set of states
14         S_1
15         S_2
16     }
17     { // Base state
18         S_1
19     }
20     { // Set of terminal states
21         S_2
22     }
23     { // Move function
24         S_1    // first state
25         0      // symbol of alphabet
26         S_2    // second state
27         S_1    // and so on
28         1
29         S_1
30         S_1
31         2
32         S_1
33         S_1
34         3
35         S_1
36         S_1
37         4
38         S_1
39         S_1
40         5
```

41	S_2
42	S_1
43	6
44	S_1
45	S_1
46	7
47	S_1
48	S_1
49	8
50	S_1
51	S_1
52	9
53	S_1
54	S_2
55	0
56	S_2
57	S_2
58	1
59	S_1
60	S_2
61	2
62	S_1
63	S_2
64	3
65	S_1
66	S_2
67	4
68	S_1
69	S_2
70	5
71	S_2
72	S_2
73	6
74	S_1
75	S_2
76	7
77	S_1
78	S_2
79	8
80	S_1
81	S_2
82	9
83	S_1
84	}

Приведем также описание автомата из второй задачи, благо, на нашем языке это сделать проще простого:

```

1      {
2          a
3          b
4      }
5      {
6          S_0
7          a_1
8          a_2
9          a_3
10         b_1
11         b_2
12         a_1b_1
13         a_2b_1
14         a_3b_1
15         a_1b_2
16         a_2b_2
17         a_3b_2
18     }
19     {
20         S_0
21     }
22     {
23         a_3b_2
24     }
25     {
26         S_0
27         a
28         a_1
29         S_0
30         b
31         b_1
32         a_1
33         a
34         a_2
35         a_1
36         b
37         a_1b_1
38         a_2
39         a
40         a_3
41         a_2
42         b
43         a_2b_1
44         a_3
45         a
46         a_3

```

47	a_3
48	b
49	a_3b_1
50	b_1
51	a
52	a_1b_1
53	b_1
54	b
55	b_2
56	b_2
57	a
58	a_1b_2
59	b_2
60	b
61	b_2
62	a_1b_1
63	a
64	a_2b_1
65	a_1b_1
66	b
67	a_1b_2
68	a_2b_1
69	a
70	a_3b_1
71	a_2b_1
72	b
73	a_2b_2
74	a_3b_1
75	a
76	a_3b_1
77	a_3b_1
78	b
79	a_3b_2
80	a_1b_2
81	a
82	a_2b_2
83	a_1b_2
84	b
85	a_1b_2
86	a_2b_2
87	a
88	a_3b_2
89	a_2b_2
90	b
91	a_2b_2
92	a_3b_2
93	a
94	a_3b_2
95	a_3b_2

```
96     b
97     a_3b_2
98 }
```

А вот автомат, распознающий букву *a*:

```
1  {
2    a
3  }
4  {
5    s_0
6    s_1
7  }
8  {
9    s_0
10 }
11 {
12   s_1
13 }
14 {
15   s_0
16   a
17   s_1
18   s_1
19   a
20   s_1
21 }
```

5.

Подсветку я делать не научился(