

Homework №3

Вадим Шабашов

Задание 1

$D/3 \approx 3$
n1

I) С добавлением стокового состояния:

Diagram description: A directed graph with 6 nodes labeled A, B, C, D, E, and ⊥. Edges are labeled with 0 or 1. A to B: 0; A to C: 1; A to ⊥: 0; B to D: 0; C to D: 1; C to ⊥: 0; D to E: 1; D to ⊥: 1; E to ⊥: 1.

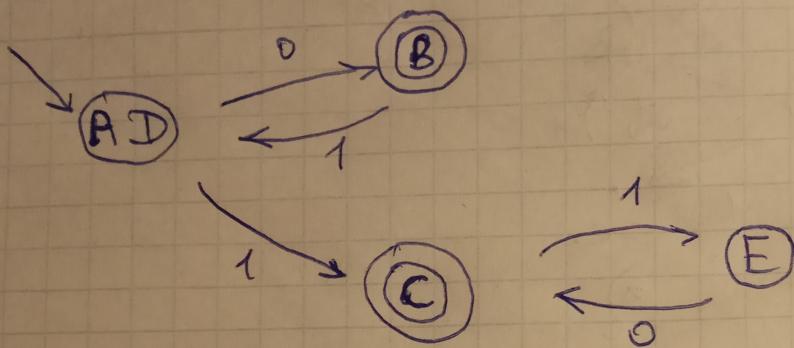
	0	1	A	B	C	D	E	⊥
A	-	-						
B	A, D	-						
C	E	A, D						
D	-	B						
E	-	C						
⊥	B, G	E, ⊥	V	V	V	V	V	V

{a) BD, BE, B⊥,
 BA, CA, CD,
 CE, C⊥

 {b) AE, DE, A⊥D⊥,
 BC E⊥

 {c) AD —
 — переход.

Схема:

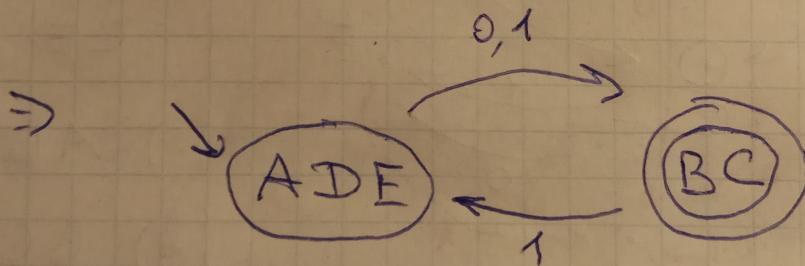


II) Добавление символов состояния:

	0	1	A	B	C	D	E
A	—	—					
B	A, D	—	✓				
C	E	A, D	✓				
D	—	B		✓	✓		
E	—	C		✓	✓		

a) BD, BE,
BA, CA, CD,
CE
¶

BC, AD, AE,
DE — неродн.



Задание 2

1. Я решил писать конечный автомат на питоне, т.к. знаю этот язык лучше всего. В нем довольно много библиотек для парсинга. В частности библиотека json, которую и решил использовать. Мне показалось, что так будет наибольшая читаемость у формата. Также формат известный, и каждый сможет разобраться, как все храниться очень быстро. Наконец библиотека имеет большую функциональность, поэтому без проблем может парсить сразу "из коробки".

2. Для представления конечного автомата в файле предлагаю использовать следующий вид:

```

1  {
2      "glossary": ["0", "1"],
3      "states": ["A", "B", "C", "D", "E"],
4      "initial_state": "A",
5      "terminal_states": ["B", "C"],
6      "edges": {
7          "A": [[{"C", "1"}, {"B", "0"}, {"B", "31"}]],
8          "B": [[{"D", "1"}, {"A", "32"}]],
9          "C": [{"E", "1"}],
10         "D": [{"B", "0"}, {"C", "1"}],
11         "E": [{"C", "0"}]
12     }
13 }
```

- В "glossary" перечисляем алфавит.
- В "states" перечисляем имена состояний.
- В "initial_state" указываем начальное состояние.
- В "terminal_states" перечисляем все терминальные состояния.
- В "edges" храним список смежности.

3. Особенности подхода:

- Имена вершин и их число может быть произвольным.
- Буквы алфавита могут быть произвольными.
- Человекочитаемость.
- Мне показалось, что хранить ребра в виде списка смежности будет удобнее всего, поскольку на лекции нам говорили о договоренности $|\Sigma| = const$, поэтому условно $|\text{states}| \gg |\text{glossary}|$. Значит если бы хранили в виде матрицы, то было бы очень много 0 (отсутствующих ребер). Кажется, что это будет не удобно, а пытаться как-то сжать (в виде разреженной матрицы хранить) — уже может быть не человекочитаемым.