Finite automata and formal languages Assignment 2

DIT323 (Finite automata and formal languages) at Gothenburg University

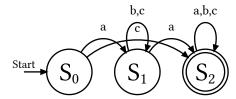
Sebastian Pålsson

Last updated: February 04, 2024

1

Answer: The language comprises of words formed by the letters a, b, and c, and it is mandatory for the words to have at least one c or two a's.

Motivation:



Looking at the automaton, we can see that any word must contain atlease one c or two a's.

2

2.1

Answer: See text file 2-1.txt

Motivation:

- **s0** is the start state and also the state when no 'b' has been encountered yet. On seeing an 'a', it remains in s0 because the string "bba" cannot start with 'a'. On seeing a 'b', it moves to **s1**.
- **s1** is the state after seeing a 'b'. On seeing another 'b', it moves to **s2** (as we now have "bb"). On seeing an 'a', it goes back to s0 because the string "bba" cannot start with "ba".
- **s2** is the state after seeing "bb". On seeing an 'a', it moves to **s3** (as we now have "bba"). On seeing a 'b', it stays in **s2** because we're still in the middle of seeing a sequence of 'b's.
- **s3** is the trap state. Once the DFA sees the string "bba", it moves to **s3** and stays there on any input. This is because the string already contains "bba", so any further input cannot change this.

The DFA accepts a string if and only if it is not in **s3** at the end of the string. This is because being in **s3** means that the string contains "bba", and we want to accept exactly those strings that do not contain "bba".

2. DIT084

2.2

Answer: See text file 2-2.txt

Motivation:

• **s0** is the start state and also the state for an even number of 'a's. If it sees an 'a', it moves to **s1** (indicating that it has now seen an odd number of 'a's). If it sees a 'b', it stays in **s0** because 'b' does not affect the count of 'a's.

• **s1** is the state for an odd number of 'a's. If it sees an 'a', it moves back to **s0** (indicating that it has now seen an even number of 'a's). If it sees a 'b', it stays in **s1** because 'b' does not affect the count of 'a's.

The DFA accepts a string if and only if it is in s0 at the end of the string. This is because being in s0 means that the string contains an even number of 'a's, and we want to accept exactly those strings.

2.3

Answer: See text file 2.3.txt

Motivation: Given two DFAs $A_1=(Q_1,\Sigma,\delta_1,q_{01},F_1)$ and $A_2=(Q_2,\Sigma,\delta_2,q_{02},F_2)$ with the same alphabet, we can construct a DFA $A_1\otimes A_2$ that satisfies the following property:

$$L(A_1 \otimes A_2) = L(A_1) \cap L(A_2) \tag{1}$$

Construction:

$$A_1 \otimes A_2 = (Q_1 \times Q_2, \Sigma, \delta, (q_{01}, q_{02}), F_1 \times F_2)$$
 [2]

where

$$\delta((s_1,s_2),a) = (\delta_1(s_1,a),\delta_2(s_2,a)) \tag{3}$$

3

3.1

Answer: See text file 3-1.txt

Motivation:

Given two DFAs $A_1=(Q_1,\Sigma,\delta_1,q_{01},F_1)$ and $A_2=(Q_2,\Sigma,\delta_2,q_{02},F_2)$ with the same alphabet, we can construct a DFA $A_1\oplus A_2$ that satisfies the following property:

$$L(A_1 \oplus A_2) = L(A_1) \cup L(A_2) \tag{4}$$

Construction:

$$A_1 \oplus A_2 = (Q_1 \times Q_2, \Sigma, \delta, (q_{01}, q_{02}), F_1 \cup F_2) \tag{5}$$

where

$$\delta((s_1, s_2), a) = (\delta_1(s_1, a), \delta_2(s_2, a))$$
[6]

Important: this also applies for NFAs.

3.2

Answer: See text file 3-2.txt

Motivation:

First we omitted the inaccessable states in the NFA, and then we construced the DFA with subset construction. The DFA is constructed by creating a state for each subset of the states of the NFA.