Week 2 CS-312 Homework

Cory Ness Jack Engledow James Sgrazzutti

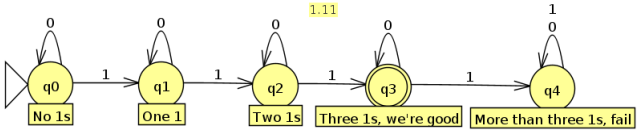
October 16, 2020

1 Problem 1.11

1.1 Question

Construct an FA that accepts all binary strings with precisely three 1's.

1.2 Answer

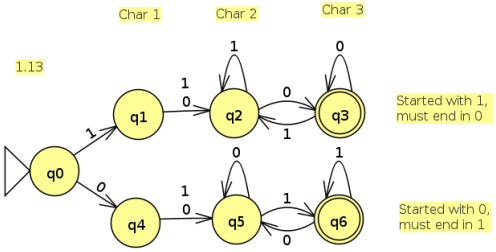


2 Problem 1.13

2.1 Question

Give an FA for the language of all binary strings that have at least three symbols and whose first and last symbols are different.

2.2 Answer

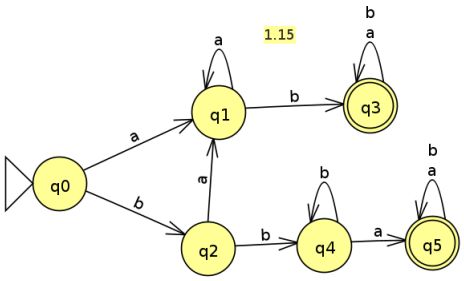


3 Problem 1.15

3.1 Question

Construct an FA that accepts all strings of $\{a, b\}$ that contain either ab or bba (or both) as substrings.

3.2 Answer

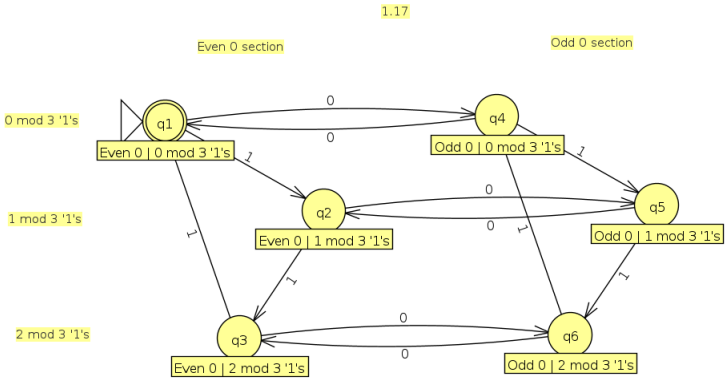


4 Problem 1.17

4.1 Question

Construct an FA that accepts all binary strings with an even number of 0's and the number of 1's is a multiple of 3.

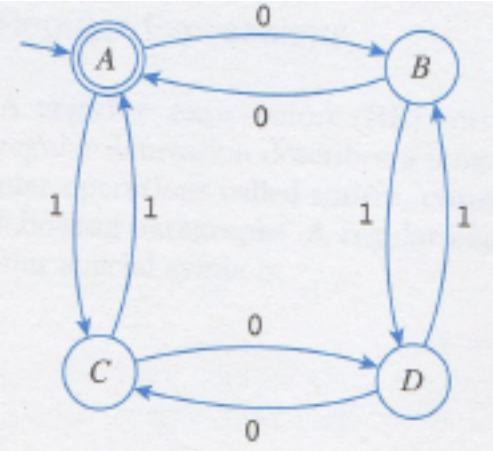
4.2 Answer



5 Problem 1.19

5.1 Question

Explain in English what the following FA accepts:



5.2 Answer

The FA accepts any binary string with an even number of 0's and an even number of 1's.

6 Problem 2.3

Give RE's for:

6.1 All binary strings with exactly two 1's

```
0*10*10*
```

6.2 All binary strings with a double symbol (contains 00 or 11) somewhere

```
(0+1)*(00+11)(0+1)*
```

6.3 All binary strings that contain both 00 and 11 as substrings

```
(0+1)*((00(0+1)*11)+(11(0+1)*00))(0+1)*
```

6.4 All binary strings without a double symbol anywhere

```
((01)*(!+0))+((10)*(!+1))+0+1+!
```

7 Problem 2.7

7.1 Question

Give an RE for the language of Exercise 1.12

7.2 Answer

```
((b+c)*+((b+c)*a(b+c)*a(b+c)*)*)a(b+c)*
```

8 Problem 2.10

8.1 Question

Give an RE for the language of Exercise 1.15

8.2 Answer

`(a+b)*((ab)+(bba))(a+b)*`

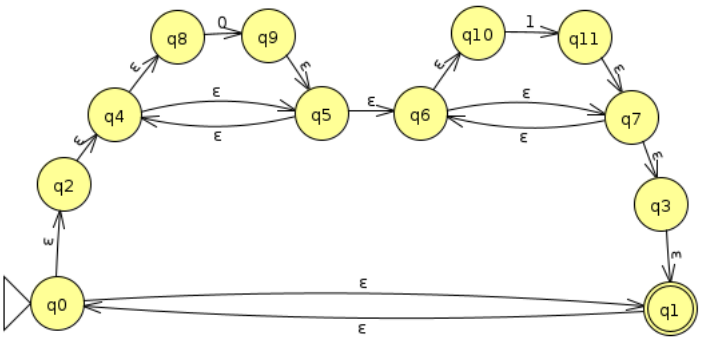
9 Problem 2.15

9.1 Question

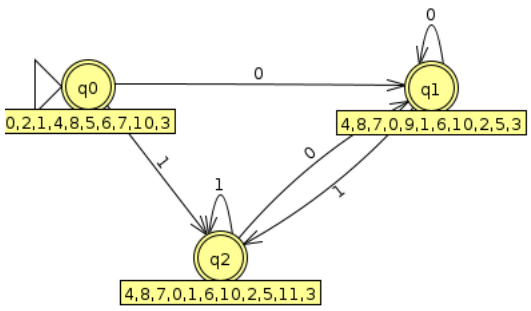
Show that the language of RE`(0*1*)*` is all binary strings.

9.2 Answer

We can trivially prove that the Regular Expression accepts all binary strings if we were to look at the Discrete Finite Automata equivalent. In order to get there, we first perform a Nondiscrete Finite Automata conversion, creating the graph shown below.



From here, we can convert it into a Discrete Finite Automata, seen below.



And from here we can trivially see the DFA accepts any binary string, including the empty string. This is because it initializes to a final state, and transitions to a final state on 0 or 1, regardless of the previous state, hence any binary string is accepted by this DFA.