CS 3120 - Module 2 HW 1

Tyler Kim

October 05

1 DFA or NFA for Two Languages

- $\{x|x$ has at least three a's and at least two b's$\}$

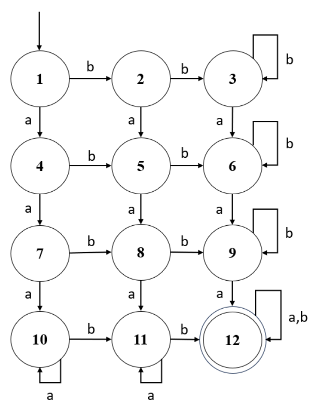

Figure 1: DFA/NFA for $\{x|x$ has at least three a's and at least two b's$\}$

	a	b
1	{4}	{2}
2	{5}	{3}
3	{6}	{3}
4	{7}	{5}
5	{8}	{6}
6	{9}	{6}
7	{10}	{8}
8	{11}	{9}
9	{12}	{9}
10	{10}	{11}
11	{11}	{12}
12	{12}	{12}

Figure 2: δ_1

The automaton is $(\{x|x \in \mathbf{N} \cap x \leq 12\}, \{\text{a, b}\}, \delta_1, \{1\}, \{12\})$

- { $x|x$ has an even number of a's and exactly one or two b's}

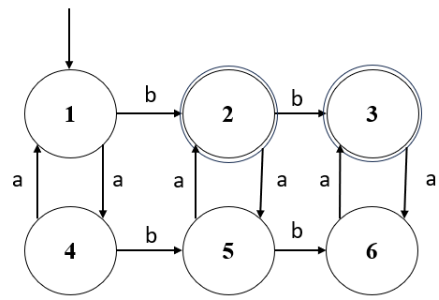

Figure 3: DFA/NFA for { $x|x$ has an even number of a's and exactly one or two b's}

	a	**b**
1	{4}	{2}
2	{5}	{3}
3	{6}	{}
4	{1}	{5}
5	{2}	{6}
6	{3}	{}

Figure 4: δ_2

The automaton is $(\{x|x \in \mathbf{N} \cap x \leq 6\}$, {a, b}, $\delta_2, \{1\}, \{2, 3\})$

2 Prove A^R

Given a regular language A, I need to prove that A^R is also a regular language. I will prove this using construction.

Since A is a regular language, this means that there exists some DFA/NFA, D, such that D recognizes A. Using the D that recognizes A as input, I can construct a new DFA, D', such that D' recognizes A^R. In order to construct D' using D, four steps must be followed:

1. A dummy start node will epsilon to the original accept states.
2. All original accept states will become normal states.
3. All start states become accept states.
4. All transitions between states point in the opposite direction

In addition, the transformation will still yield a a valid DFA/NFA because D' has same finite set of states as D, the alphabets are the same, the transitions are still valid, and D' will have one start state and at least one accept state.

3 Prove $_1$ is irregular using Pumping Lemma

Assume that the language $A_1 = \{0^n 1^n 2^n | n \geq 0\}$. Let $s' = 000001111122222$ such that $s' \in A_1$. Since A_1 is regular, that means s' can be split into three components xyz such that y can be pumped. There are 4 possible cases for y:

1. y **is all** 0**'s or all** 1**'s or all** 2**'s**: If y were to be pumped, this would not follow the language since there would be an unequal number of 0's, 1's, and/or 2's.

2. y **is a combination of equal number of** 0**'s and** 1**'s**: If y were to be pumped, the number of 0's and 1's would be the same but there would not be enough 2's. Therefore, the language would break.

3. y **is a combination of equal number of** 1**'s and** 2**'s**: If y were to be pumped, the number of 1's and 2's would be the same but there would not be enough 0's. Therefore, the language would break.

4. y **is a combination of equal number of** 0**'s,** 1**'s, and** 2**'s**: If y were to be pumped, the number of 0's, 1's, and 2's would be the same but it would not follow the same pattern as provided in the language. Therefore, the language would break.

Since there is no way to pump s', language A_1 is not regular.

4 Incorrect Proof

The biggest problem with the given proof attempt is that 0^*1^* does not mean that the number of 0's and the number of 1's have to be equal. In other words, you can have different number of 0's and different number of 1's. The proof seems to imply that the number of 0's and the number of 1's must be equal when that is not the case. The proof would have worked if the language was 0^n1^n but for this case, you can pick a y of all 0's or all 1's and it will successfully pump.