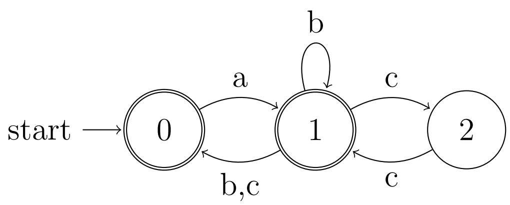
ECS 120 Problem Set 3

Hardy Jones
999397426
Professor Rogaway
Spring 2014



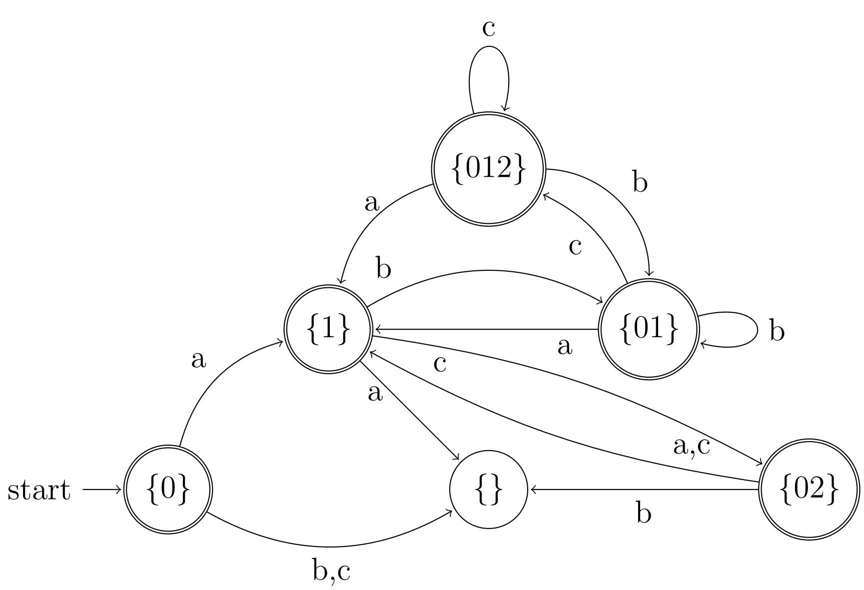
Problem 1

We need to start by enumerating all sets of states, and their transitions for each input.

State	a	b	c
{0}	{1}	{}	{}
{1}	{}	{0,1}	{0,2}
{0,1}	{1}	{0,1}	{0,1,2}
{0,2}	{1}	{}	{1}
{0,1,2}	{1}	{0,1}	{0,1,2}

Now we need to generate new final states. These are any state sets that contain the original final states. In this case, all of the states in the first column are final states.

Finally, we can generate our DFA.



Problem 2 We have ε-arrows from 1 to 2, from 2 to 5, and from 3 to 1.

From 1 to 2, we can get to state 4 following the path "εa". So, we can eliminate this ε-arrow by creating a new path from 1 to 4 via a.

From 1 to 2, we can also get to state 5 through an ε-arrow to state 2 following the path "$\varepsilon\varepsilon$". Since state 5 is a final state, we can eliminate these ε-arrows by making state 1 a final state.

We can get from 2 to 5 via an ε-arrow and state 5 is a final state. So, we can eliminate this ε-arrow by making state 2 a final state.

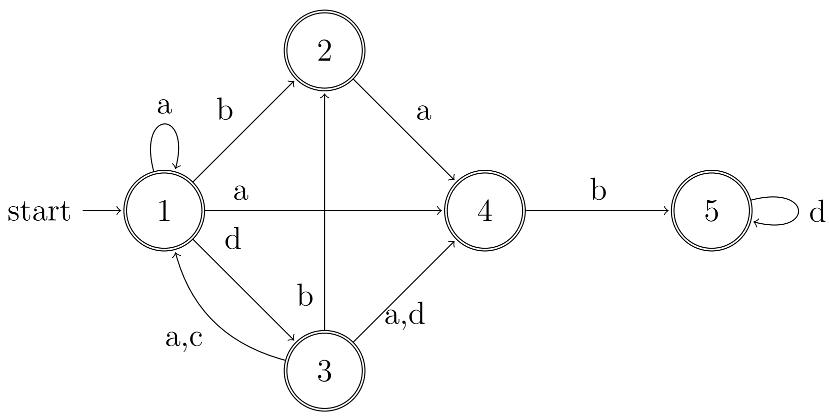
From 3 to 1, we can get to 1 again following the path "εa". So, we can eliminate this ε-arrow by creating a new path from 3 to 1 via a.

From 3 to 1, we can get to 1 again following the path "εc". So, we can eliminate this ε-arrow by creating a new path from 3 to 1 via c.

From 3 to 1, we can get to state 4 through an ε-arrow to state 2 following the path "$\varepsilon\varepsilon$a". So, we can eliminate this ε-arrow by creating a new path from 3 to 4 via a.

Finally, from 3 to 1, we can also get to state 5 through an ε-arrow to state 2 and another ε-arrow to state 5 following the path "$\varepsilon\varepsilon\varepsilon$". Since state 5 is a final state, we can eliminate these ε-arrows by making state 3 a final state.

Our ε-arrow-free NFA is:



Problem 3 *Proof.* If L_1, L_2, L_3 are DFA-acceptable languages, then the intersection of these languages is also acceptable. We can reformulate the definition of **maj**:

$\mathbf{maj}(L_1, L_2, L_3) =$

$\{x \in \Sigma^* | (x \in L_1 \wedge x \in L_2) \vee (x \in L_1 \wedge x \in L_3) \vee (x \in L_2 \wedge x \in L_3) \vee (x \in L_1 \wedge x \in L_2 \wedge x \in L_3)\}$

Or put more succinctly:

$\mathbf{maj}(L_1, L_2, L_3) =$

$\{x \in \Sigma^* | x \in (L_1 \cap L_2) \vee x \in (L_1 \cap L_3) \vee x \in (L_2 \cap L_3) \vee x \in (L_1 \cap L_2 \cap L_3)\}$

But we know that each of these intersections is DFA-acceptable, so $\mathbf{maj}(L_1, L_2, L_3)$ is DFA-acceptable. $\square$

Problem 4 *Proof.* Given some accepting DFA $M = (Q, \Sigma, \delta, q_0, F)$ for L, we can construct a new accepting DFA $M' = (Q', \Sigma', \delta', q_0, F')$ for $\mathcal{Z}(L)$, where:

$$\begin{aligned}
Q' &= Q \cup n \text{ additional states, where } n = |Q| \\
\Sigma' &= \Sigma \cup \{0\} \\
\delta' &: Q' \times \Sigma' \to Q' \\
F' &= F \cup q_f
\end{aligned}$$

Where we define

$$\begin{aligned}
\delta'(q, \varepsilon) &= \delta(q, \varepsilon) \\
\delta'(q_i, a0x) &= \delta(q_i, ax)
\end{aligned}$$

In other words, we "consume" the 0 by sending the DFA to an intermediate state q_{i_0} which needs a 0 in order to continue accepting the string.

$q_f =$ The state with an arrow for the character 0 coming from all final states of M and an arrow for every character in Σ' going to a rejection state.

By construction, M' accepts all strings from M with 0 interspersed and also suffixed by 0. So, M' is also DFA-acceptable.

Thus, DFA-acceptable languages are closed under $\mathcal{Z}$. □

Proof. We can also prove this in another way.

Given some accepting DFA M for L and the fact that DFA's are closed under concatenation, we can decompose M into smaller DFA's of a single character. I.e. $M = M_1 M_2 \cdots M_n$. We can also construct a DFA M_0 that accepts the singleton $\{0\}$. Now we can intersperse M_0 through our decomposed M. We end up with $M' = M_1 M_0 M_2 M_0 \cdots M_n M_0$.

And since DFA-acceptable languages are closed under concatenation. Our new machine is an accepting DFA.

If we define $\mathcal{Z}$ as this deconstruction/reconstruction operation, then we see that DFA-acceptable languages are closed under $\mathcal{Z}$. □