| Prob. 1 | Prob. 2 | Prob. 3 |
|---|---|---|
|  |  |  |

## Team members: Küng, Pirelli, Schubert, Dousse, Vu

### Problem 1.

A string in A cannot begin with a zero; it needs to switch to 1s eventually in order to have an even number of 1s, but ”01” is banned. Also, since ”01” is banned, all 1s must precede all 0s. Therefore, A can be redefined as the concatenation of

$$A_1 \subseteq \{1\}^* := \{\omega \mid \omega \text{ contains an odd number of 1s}\}$$
$$A_2 \subseteq \{0\}^* := \{\omega \mid \omega \text{ contains an even number of 0s}\}.$$

Both $A_1$ and $A_2$ are regular, since we can construct the following DFAs for them:

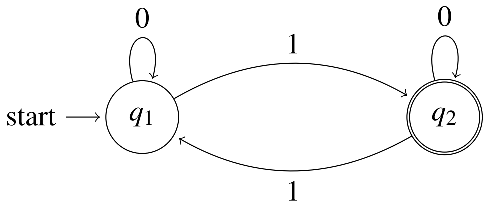

Figure 1: DFA $A_1$

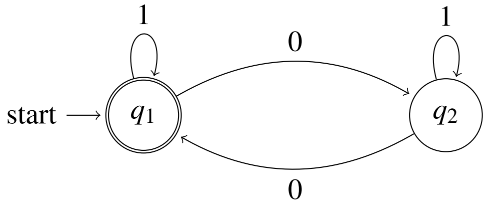

Figure 2: DFA $A_2$

Since A is the concatenation of the two regular languages ($A_1 \circ A_2$), it is also regular, as we saw in class.

## Problem 2.

Since $L$ is regular, there is a DFA $M$ that can match $L$. Let us create an NFA $M'$ that can match $L^R$, which will prove that $L^R$ is regular. $M'$ is composed of all of $M$'s states, plus one new state. This new state is the start state of the NFA, and it contains only $\varepsilon$ transitions to all states that were accepting states of $M$. The accepting state of $M'$ is the start state of $M$. The transition function of $M'$ is the opposite of $M$; if an input $x$ caused $M$ to go from $q_0$ to $q_1$, it causes $M'$ to go from $q_1$ to $q_0$.

To express it more formally:

$$\begin{aligned}
M &= (Q, \Sigma, \delta, s, F) \\
M' &= (Q', \Sigma, \delta;, s', F') \\
s' &= \text{new state, connected to the end of all the states } F \\
Q' &= Q + \{s'\} \\
F' &= \{s\} \\
\delta' &= \delta^R
\end{aligned}$$

$$\delta'(q,x) = \begin{cases} F & \text{if } q = s' \\ \{q_2 | \delta(q_2,x)\} = q & \text{otherwise} \end{cases}$$

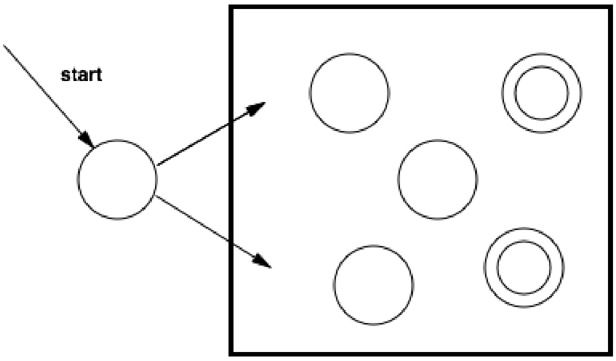

Figure 3: Original DFA $M$

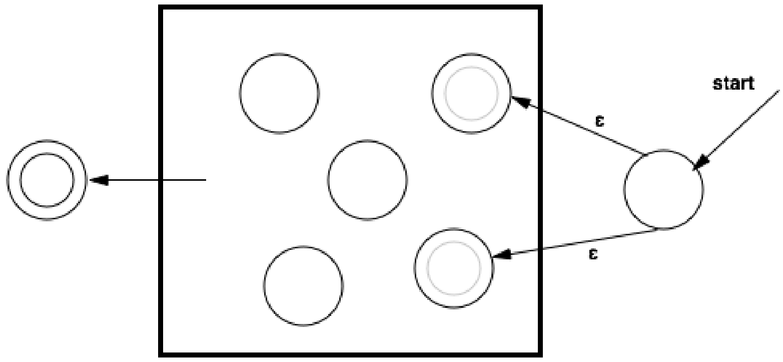

Figure 4: New NFA $M'$

This new machine $M'$ will be able to read every word the original machine $M$ was able to read, in the opposite direction; indeed, for every word $M$ was able to read it ended up in one of the accepting states to which $M'$ can transition to at no cost ($\varepsilon$) from the start and therefore will end up after the reverse read of the word in the initial state of $M$, meaning in the accepting state of $M'$.

Problem 3.

We want to prove the following:

$$\frac{L}{2} := \{x \in \Sigma * | \exists y : yx \in L, |x| = |y|\}$$

We haven proven in problem 2 that the inverse of a regular language is regular. Let's assume we have some way of proving that the first half of a language is regular; we could prove that the first half of a reversed language is regular, which would also mean that the reverse of the first half of a reversed language is regular. A bit tricky, but it does correspond to the second half.

Let us prove our assumption; that the first half of a regular language is regular.

Let $L$ be a regular language. Let $half(L) := \{x | \exists y : xy \in L, |x| = |y|\}$. We want to prove that $half(L)$ is regular.

Let $M < Q, \Sigma, \delta, q_0, F >$ be a DFA that accepts $L$, which must exist because $L$ is regular. Supposing that $\hat{\delta}(q_0, x) = q_i$, i.e. the input $x$ leads $M$ to the state $q_i$, we must check if $\exists y : |y| = |x|$ that satisfies $\hat{\delta}(q_i, y) \in F$.

Let $S_n \subset Q$ be the set of states that lead to an accepting state for some input (not necessarily any input) of length n. It is clear that for $x$ of length $n$ and $\hat{\delta}(q_0, x) = q_i$, $q_i \in S_n \rightarrow x \in \frac{L}{2}$. $(*)$

$S_0 = F$, since by definition of $S_n$ there are no more steps needed. $S_{n+1}$ can be easily computed from $S_n$ and $\delta$; it is the set of states that can transition to a state in $S_n$. We need a DFA that will keep track of $S_n$ as we go through $L$. Let $M'$ be a new DFA whose states are in $(Q, Q*)$ i.e. they're pairs of one state in $Q$ and a set of states in $Q$. The transition function $\delta'$ of $M'$ takes an input $x$ of length $n$ and yields $(\hat{\delta}(q_0, x), S_n)$.

The start state of $M'$ is $(q_0, F)$, and the accepting states are $(q, S) \in (Q, Q^*)$ where $q \in S$; in common English, "the states reached after n inputs that can also reach an accepting state after n inputs".

$(*)$ above allows us to show that $M'$ indeed accepts $half(L)$, and therefore $half(L)$ is regular.

(source/inspiration for the $half(L)$ proof: `http://www-bcf.usc.edu/~breichar/teaching/2011cs360/half%28L%29example.pdf`)