CS 381 – Prelim1 Solutions

1. For a word $w \in \{0,1\}^*$, define a language $L_{sub(w)} = \{xwy \mid x, y \in \{0,1\}^*\}$. That is, $L_{sub(w)}$ is the set of all strings that contain w as a sub-string.

Problem i:

Prove that for every pair of strings x, y: $L_{sub(xy)} \subseteq L_{sub(x)} \cap L_{sub(y)}$.

Solution i:

Take any string $w \in L_{sub(xy)}$. We must show that $w \in L_{sub(x)} \cap L_{sub(y)}$. w is in this intersection if and only if w is in each of the intersected languages. Thus we must show that $w \in L_{sub(x)}$ and also that $w \in L_{sub(y)}$. Because $w \in L_{sub(xy)}$ we know that $w = x'\ xy\ y'$ for some $x', y' \in \{0,1\}^*$. We can show that $w \in L_{sub(x)}$ because $w = x_x\ x\ y_x$ for $x_x = x'$ and $y_x = yy'$. We can show that $w \in L_{sub(y)}$ because $w = x_y\ y\ y_y$ for $x_y = xx'$ and $y_y = y'$. Thus $w \in L_{sub(x)} \cap L_{sub(y)}$ and therefore, because we picked w arbitrarily, we conclude $L_{sub(xy)} \subseteq L_{sub(x)} \cap L_{sub(y)}$.

Problem ii:

Prove that for $x = 00$ and $y = 11$, it is not the case that $L_{sub(xy)} \supseteq L_{sub(x)} \cap L_{sub(y)}$.

Solution ii:

We prove this by constructing a counter example – a word $w \in L_{sub(x)} \cap L_{sub(y)}$ but $w \notin L_{sub(xy)}$. Take $w = 001011$. Note $w = x_{00}00y_{00}$ for $x_{00} = \epsilon \in \{0,1\}^*$ and $y_{00} = 1011 \in \{0,1\}^*$. Therefore $w \in L_{sub(x=00)}$. Note $w = x_{11}11y_{11}$ for $x_{11} = 0010 \in \{0,1\}^*$ and $y_{11} = \epsilon \in \{0,1\}^*$. Therefore $w \in L_{sub(y=11)}$. So $w \in L_{sub(x)} \cap L_{sub(y)}$. But ... $w \notin L_{sub(xy=0011)}$ because 0011 is not a substring of $w = 001011$. Thus we conclude $L_{sub(xy)} \not\supseteq L_{sub(x)} \cap L_{sub(y)}$.

Problem iii:

Prove that for every $w \in \{0,1\}^*$, $L_{sub(w)}$ is a regular language.

Solution iii:

We know that regular languages are closed under concatenation. We know that $\{0,1\}^*$ is regular because we can easily construct a DFA that accepts all strings. We know that $\{w\}$ is regular for any specific $w \in \{0,1\}^*$ because all finite languages are regular. Finally, observe that the definition of $L_{sub(w)}$ is simply the triple concatenation of $\{0,1\}^*$ and $\{w\}$ and $\{0,1\}^*$. Therefore, $L_{sub(w)}$ is regular.

Alternatively, one could directly construct a NFA accepting $L_{sub(w)}$ as shown in figures 1 and 2.

Comments: Those who claimed that the closure of regular languages under concatenation implied the regularity of $L_{sub(w)}$ without identifying the languages being concatenated did not receive much credit. There were many who claimed that $L_{sub(w)} = \{0,1\}^*\{0,1\}^*\{0,1\}^*$, apparently because w could be any string in $\{0,1\}^*$. However, as the problem stated, $L_{sub(w)}$ is the set of all strings

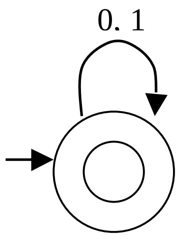

Figure 1: NFA accepting $L_{sub(w)}$, where $w = \varepsilon$

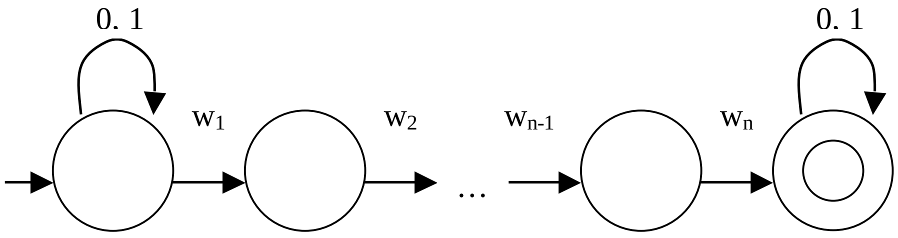

Figure 2: NFA accepting $L_{sub(w)}$, where $w = w_1w_2...w_n$, with $w_i \in \{0,1\}$ for $1 \le i \le n$.

over $\{0,1\}^*$ which contain w as a substring; i.e., given any single string x in $\{0,1\}^*$, there is a unique corresponding language $L_{sub(x)}$ consisting of all words in $\{0,1\}^*$ containing x as a substring.

Important note: It does **not** suffice to prove that any element of $L_{sub(w)}$ is in $\{0,1\}^*$, and then to argue that since $L_{sub(w)} \subseteq \{0,1\}^*$ and $\{0,1\}^*$ is regular, $L_{sub(w)}$ must be regular. **It is not the case that every subset of a regular language is regular!!** If this were true, every language would be regular, whereas the truth is that we have proven existence and seen examples of lanuages that are not regular.

The incorrect reasoning that seems to have been used by those who made this mistake is that"if language L is regular, all the strings in L must be computable, so therefore all of the strings in any subset of L are computable." However, this does not make sense! Languages are computable, not strings! **For a language L to be computable by DFA M, M must accept all strings in L *and reject all strings not in L.***

Problem iv (BONUS):

Prove that for every set K of $\{0,1\}^*$ strings, the following language is regular:

$$\bigcap_{w \in K} L_{sub(w)}$$

Solution iv:

If K is finite, the resulting intersection is regular because each for each $w \in K$, $L_{sub(w)}$ is regular and the intersection of finitely many regular languages is regular.

If K is infinite... I claim that the above language is regular because it is the empty. Observe that for all words $x \in L_{sub(w)}$, it must be the case that $|x| \geq |w|$. Thus $L_{sub(w)} \subseteq L_{|w|}$ where we define $L_{|w|}$ as the set of all words of length greater than or equal to $|w|$. There are a finite number of words of any specific length, so for K to be infinite there must be no bound on the length of words in K. Thus if I take any word y, I can show that $y \notin \bigcap_{w \in K} L_{sub(w)}$ because we can find a string $k \in K$ with length $|k| > |w|$ and thus $w \notin L_{|k|}$ which in turn means that $w \notin L_{sub(w)}$.

We have handled the case when K is infinite and when K is finite and shown our language to be regular in both cases. Hence, the language is regular.

2. Consider the following NFA $N = (Q, \Sigma, q_0, \Delta, F) = (\{a, b, c\}, \{0, 1\}, a, \Delta, \{c\})$ where Δ is shown by the following NFA diagram (converted from the exam table).

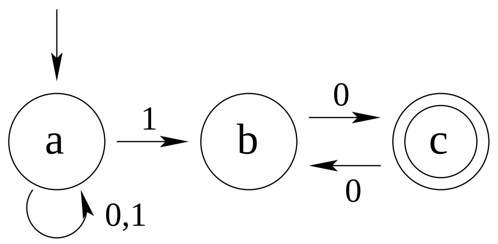

Figure 3: NFA N states and transitions.

Problem i:

Describe in words what language is computed by N.

Solution i:

This machine accepts all strings that end with a 1 followed by an odd number of 0s. This is clear because the only way to get to state c, the accepting state, is to read a 1 to go from a to b and then to read an odd number of 0s (an even number of 0s returns to b). Any leading 0s and 1s are ignored because state a can transition to itself on either a 0 or a 1.

An equally correct way of describing this solution is all strings that end in 10 and are folowed by an even number of 0s.

Comments i:

Most people got this problem correct. Of the most common errors, one was not noting the 1 character which must be present in all strings accepted by this automaton. The other was merely stating the strings accepted by the automaton with no supporting explanation to the answer. People that incorrectly transcribed the automaton from table to graph form lost 2 points/error, but the rest of their solution was graded with regard to the NFA they had drawn.

Problem ii:

Construct a deterministic automaton M such that $L(M) = L(N)$. (See the following figure for a drawn solution.)

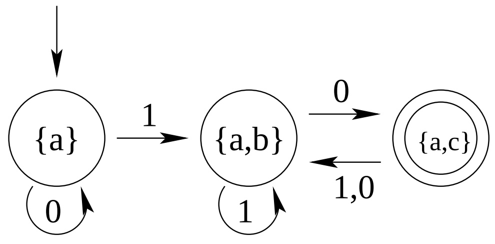

Figure 4: We have constructed this machine using the power set of states method. Each state is thus labeled as a set of the states of N. Elements of the power set that were not needed as states have been omitted.

Grading Comments: You must argue or prove correctness of your DFA. If you used the subset construction, that's enough. If not, you must show that your DFA accepts the correct language. Be sure to indicate start state and accepting states.