CS 3100 – Models of Computation – Fall 2011

Assignment 8, Posted on: 10/20, Due 10/27 midnight

This assignment is worth 8% of the total points for assignments

100 points total

1 Grading Key for Asg8

Submit your answers as simple writeups in English. Be brief, yet complete. To make grading easier, please provide (or summarize) the question and then your answer. Submit as PDF or TXT into asg8 via handin.

1. **(10 points) for each language** Let the computer go first. Select these languages:

   $$L = \{a^n b^k c^{(n+k)} \ \mid \ n \geq 0, \ k \geq 0\}$$

   A simple pump-down approach works, and takes the string outside the language for a choice of $n = m$, where m is the number of DFA states. *Note that we will be doing more rigorous proofs later.*

   $$L = \{a^n b^l a^k \ \mid \ n > 5, \ l > 3, \ k \leq l\}$$

   I chose m=10. The machine offered me 'aaaaaa bbbb aa.' I pumped down on the 'a's to win. *Note that we will be doing more rigorous proofs later.*

   Show how you won (which means you could prove that the language is not regular). Then understand the explanation provided, and copy it down in your answer.

   Look for clear answers; then grade accordingly. Must be CFGs, so they could pump and get out.

2. **(10 points) for each language** Let the computer go first. Select two languages from the JFLAP Pumping Lemma tutor for which you are guaranteed to get "Please try again." Show that to be the case for at least three tries for each language. Then understand the explanation provided, and copy it down in your answer.

   Look for clear answers; then grade accordingly. Must be regular languages, so they could not 'pump out.'

3. **(20 points)** Do 12.6 from my book chapter.

4. **(3 points) for each part** Do 12.7, parts 1, 2, 3, 4, 5, 8, 9, 10, 11, 12 from my book chapter.

   Answers must be similar to this:

   1 False. **We can have $L_1 = \{0^i 1^j \ \mid \ i = j\}$ and $L_2 = \{0^i 1^j \ \mid \ i \neq j\}$. Their union is the language denoted by $0^* 1^*$ even though L_1 and L_2 are non-regular.**

   2 False. $L_1 \cap L_2 = \emptyset$ **which is regular.**

3 True. **$\emptyset$ is a subset of *any* language L.**

4 True. **I mis-read the question. It says "a regular set." So it could be any regular set. Obviously, all non-regular sets are proper supersets of the empty set $\emptyset$.**

5 True. **Choose $\emptyset$ which is the subset of any regular set one may choose.**

8 True. **This is because reversal preserves regularity.**

9 True. **Choose Σ^* for the regular set.**

10 True. **Choose $\emptyset^*$ for the regular set.**

11 True. **Choose $\emptyset^*$ for the regular set.**

12 False. **All finite sets are regular.**

5. **(10 points)** Write a summary of the proof of "U" versus "Y" given in my book. What would have been a simpler proof for that problem? Explain in a few sentences.

   Answers must be similar to this: **The proof shows how *all* decompositions into "y" (the middle of the pump) are being covered by the choice of $m + m!$. A simpler proof would have been to take the complement of the given set and then prove it is not regular.**

   The complement is $\{0^n 1^n \mid n \geq 0\} \cup (0+1)^*1(0+1)^*0(0+1)^*$, i.e. the familiar 0^n1^n set and all those strings where a 1 may appear before a 0. Call these parts $L_{0n1n} = \{0^n 1^n \mid n \geq 0\}$ and $L_{.1.0.} = (0+1)^*1(0+1)^*0(0+1)^*$.

   Now, $L_{.1.0.}$ is regular. *However there is no overlap with L_{0n1n}. They are disjoint sets. Hence, by the theorem Theorem-1 below, we can prove that L_{0n1n} is non-regular.*

Theorem-1: Suppose $A \cap B = \emptyset$ and A is non-regular, and B is regular. Then $A \cup B$ is non-regular.

Proof: Suppose not. Then there is a DFA for $A \cup B$—call it $D_{A\cup B}$. Now build a DFA for A by taking $D_{A\cup B} \cap (\overline{(DFA_B)})$. This is a DFA for A (call it D_A) because it accepts exactly when the "first part" accepts. This is a contradiction because A is non-regular.

2 Extra Credit Problem - 15 points extra

Prove that the following language in the Pumping Lemma tutor is regular (draw a DFA for it). (Note: this is not as hard as it sounds—simplify the language first.)

$$L = \{b^5 w \mid w \in \{a,b\}^*,\ (2n_a(w) + 5n_b(w))\ mod\ 3 = 0\}$$

Here, $n_a(w)$ denotes the number of as in string w, and so on.

Let us manipulate the condition $(2n_a(w) + 5n_b(w))\ mod\ 3 = 0$ and see what it simplifies to. Let's use % for mod.

$$
\begin{aligned}
& (2n_a(w) + 5n_b(w)) \;\%\; 3 = 0 \\
= \;& (2.(n_a(w)\%3) + 2.(n_b(w)\%3)) \;\%\; 3 = 0 \\
= \;& (n_a(w)\%3 + n_b(w)\%3)\%3 = 0 \\
= \;& (n_a(w) + n_b(w))\%3 = 0
\end{aligned}
$$

This is clearly regular. In fact, it is our favorite mod-3 machine which does not care about whether it is stepping on a or b! The b^5 that precedes is also regular. So their concatenation is regular.