CSE322: Formal Models in Computer Science Spring 2006

Puzzles for fun

At the beginning of each class, I will give out a "puzzle" (or at the very least, one every week). The problems are not going to be graded and are intended purely for moments like when you are on the bus and have nothing to do and are dying to solve (what else but) a puzzle. They also would be a good source for extra practice, though be warned that more often than not these would be harder than a typical homework problem.

Solutions will not be handled out but feel free to talk to the instructor/TAs if you want to verify your solutions. All the puzzles would be recorded in this file for future reference.

---

1. (Mar 29) Is the following statement true ?

   For any deterministic finite automaton (DFA) M, there is another DFA M' (M' can be the same as M) such that the following two conditions hold:

   (a) M' accepts exactly the same set of string as M, that is, $L(M) = L(M')$; and

   (b) M' has *at most* one final state.

   If your answer is yes, then you need to show how to convert any M to an equivalent M' with the above properties. If your answer is no, then you need to show an M (with at least two final states) and prove that such an M' cannot exist.

2. (Mar 31) For any $k \geq 1$ and ℓ such that $0 \leq \ell \leq k-1$, define the following language:

   $$C_{\ell,k} = \{w \mid w \in \{0,1\}^*, w \text{ is the binary representation of a number } N_w \text{,where } N_w \bmod k = \ell\}.$$

   Prove that for every fixed k and ℓ, $C_{\ell,k}$ is regular. (Thinking about the formal definition of a DFA would be helpful for this. If you need a hint, take a look at Example 1.5 in Sipser's book).

3. (Apr 3) Design a DFA with as few states as possible for the following language (think of k as an arbitrary but fixed (constant) number given to you [1]):

   $$L_k = \{w \mid w \in \{0,1\}^* \text{ and the } k^{\text{th}} \text{ symbol from the right is a } 1\}.$$

   Also prove that there is no DFA that has fewer states than the one you constructed above.

4. (Apr 5) Design an NFA with as few states as possible for the language L_k (in the puzzle above). Also prove that there is no NFA with a fewer number of states.

   (As a hint, the minimum number of states for an NFA and a DFA for L_k is $k+1$ and 2^k respectively. These two puzzles show that the exponential blowup in the number of states in the construction of DFAs from NFAs we covered in class, in general, is inevitable.)

---

[1] In particular this, means that you will have a different DFA for each k.

5. (Apr 10) Prove that if L is regular, then so is the following language:

$$\log_2(L) = \{x \mid xy \in L, 2^{|x|} = |y|\}.$$

6. (Apr 14) Let r and s be regular expressions where the language described by r does not contain the empty string ϵ. Consider the equation

   $$X = r \circ X \cup s,$$

   (where $\circ$ stands for the concatenation of regular expressions and $\cup$ for the union) with unknown variable X.

   Find a solution for X (i.e., a regular expression) that satisfies the above equation and prove that this solution is *unique*[2].

7. (Apr 17) Prove the following.

   (a) If a language is described by a regular expression with m symbols, then it is also accepted by an NFA with $O(m)$ states.

   (b) If a language is accepted by an NFA with k states, then it also described by a regular expression with $O(4^k)$ symbols.

8. (Apr 19) Prove that every regular language is accepted by a *planar* NFA.

   An NFA is planar if it can be embedded in the plane (i.e. one can "draw" it) such that there are no crossings.

9. (Apr 24) Use the pumping lemma to prove that the following language is not regular:

   $L_1 = \{w \mid w \in \{0,1\}^*, w$ starts with a 1 and is the binary representation of a prime number $\}$.

   For example 101 is in L as it is the binary representation of 5 while 110, which is the binary representation of 6, is not in L.

10. (Apr 28) Use the Myhill-Nerode theorem to show that the following language is not regular:

    $L_2 = \{0^n \mid n$ is a prime number $\}$.

11. (May 3) Design a context-free grammar for the following language:

    $L_3 = \{w \mid w \in \{(,)\}^*$ such that w is a string of balanced parens except for *exactly one* extra $(\}$.

    For example $((()(())) \in L_3$ but $(()(()))$ and $(((()$ are not in L_3.

12. (May 8) Consider the following language:

    $L_4 = \{a^i b^j c^k \mid i = j$ or $j = k$ and $i, j, k \geq 0\}$.

    Prove that

    - L_4 is context free, and

[2] In other words, there cannot be two regular expression that describe different languages *and* satisfy the equation.

- L_4 is inherently ambiguous.

13. (May 10) Design a context free grammar for the following language:

    $L_5 = \{xy \mid x, y \in \{0,1\}^* \text{ and } |x| = |y| \text{ but } x \neq y\}$.

14. (May 12) Design a PDA for the language L_5 above.

15. (May 19) Prove that the language L_2 (In # 10 handed out on Apr 28) is also not context-free.

16. (May 24) Show that PDAs with *two* stacks are equivalent to Turing Machines.

    (*Hint:* Try to think of what you need in order to simulate a TM and use the two stack to move around stuff like a "slinky".)

17. (May 26) Show that the following language is decidable.

    $L_6 = \{\langle G \rangle \mid G \text{ is a CFG such that } 1^* \subseteq L(G)\}$.

18. (May 31) Let Σ be a fixed alphabet. Prove the following:

    (a) Σ^* is countable.

    (b) The set of all languages $L \subseteq \Sigma^*$ is uncountable.

    (c) Conclude that there is a language that is not Turing-recognizable.