CS360 Fall 2013 -Assignment 5

Due Wednesday, November 27th, 11:59am

1. [**20 points**] We define a new model of computation, *Two-Stack Push Down Automata* (2SPDA)as machine that is exactly like a PDA, except that is has two stacks (at any point in time the machine read-write head is pointed to the top of one of these stacks), and for every state and stack letter there is an ϵ transition that takes the head from its current stack to the other stack. A 2SPDA, $M = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$ accepts a word, w, if there is a path of machine transitions that starts with w on the machine input tape, the machine in state q_0 both stacks having Z_0, and reaches an accepting state when it reaches the end of w. Prove 2SPDA's have the same computing power as Turing machines. Namely, a language can be computed by some 2SPDA if and only if it can be computed by some Turing machine. More concretely, given any Turing machine, T, construct a 2SODA that accepts the same language that T does, and given any 2SPDA, $M = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$, describe in detail a Turing machine that accepts the language $L(M)$.

2. Consider the following languages:

   - $L_1 = \{a^n b^k : \frac{n}{k}$ *is an integer* $\}$,
   - $L_2 = \{a^n b^k :$ *n is an even number and k is a prime number, or, n is an odd number and k is not a prime number* $\}$.

   For each of these languages,

   (a) [**2 × 10 points**] Describe a Turing machine M that decides it. Namely, a machine M that halts on every input and $L(M) = L$.

   (b) [**2 × 5 points**] Prove that L is not a regular language.

   (c) [**2 × 10 points**] Find out whether L a CFL and prove your claim.

3. [**Bonus 10 points**]Prove that each of the following languages, L, is decidable. That is, that there exist a Turing machine M that halts on every input, such that $L(M) = L$. $L_\pi = \{a^n :$ *in the decimal expansion of π there is a run of at least n consecutive 7's* $\}$.

4. [**2 × 15 points**] Prove that each of the following problems is not decidable:

   (a) Input: A code P for a program. Decision: Does P halt on infinitely many inputs?

   (b) Input: Codes for two programs P_1, P_2. Decision: Do these two programs compute functions with the same domain? (Namely, is it the case that for every input I, P_1 halts of I if and only if P_2 halts on I?).