CS 3100 – Models of Computation – Fall 2010

Practice Midterm-2

This being a practice test, there are more longer answer questions and lesser multiple-choice questions than in the real exam.

1 Multiple Choice (Closed Book), 30 points

You earn 2 points for each right answer and -0.5 for each wrong. If *all of the above* or *none of the above* can be selected, you must do so.

1. *Select all that are true by putting a check mark (√).* Every context-free grammar
   - Is regular
   - Generates a context-free language
   - Can be converted to the Chomsky Normal Form
   - All of the above
2. Using a suitable Pumping Lemma, one can
   - Prove that a language is regular
   - Prove that a language is context-free
   - Prove that a language is context-free but not regular
   - Disprove that a language is context-free
3. The language $\{00, 110\}$
   - Is regular
   - Is context-free
   - Has a left-linear grammar
   - All of the above
4. The Post Correspondence Problem states that there is a domino of tiles with bit-strings written on the top and bottom (PCP instance), and
   - There is no LBA that decides an arrangement of the tiles, for all PCP instances
   - There is no TM that decides an arrangement (without tile repetitions) of the tiles, for all PCP instances
   - There is no TM that decides an arrangement (with tile repetitions allowed) of the tiles, for all PCP instances
   - The top and bottom are strings over a unary alphabet and such instances are RE

2 Longer Answer Questions (Open Book), 30 points

1. Convert the following left-linear grammar into an equivalent right-linear grammar.

```
S -> S a | T b | U c | f

T -> T a | c
```

```
U -> S d | e
```

2. Show using the Pumping Lemma that $\{ww \mid w \in \{0,1\}^*\}$ is not context-free.
3. Simple PDA design problems
4. Simple CFG design problems
5. Simple TM design problems
6. Problems about unary PCP - recursive or RE
7. Problems about binary PCP - recursive or RE
8. Review these proofs:
   - That S_{TM} is not RE
   - Hence not recursive
   - Hence if there is an A_{TM} decider called H, then there is a contradiction because H can decide S_{TM}
   - Direct proof: Suppose we have an H that decides A_{TM}. Build a machine $D(x)$ as follows. Find the machine encoding x (call it M_x). See if $H(M_x, x)$ says M_x accepts x; if so D rejects x; else D accepts. Now consider what happens under the application $D(\langle D \rangle)$. If this accepts then it does not, according to H; else it does not accept, but then accepts according to H.
9. Compute the cardinality of the set of all functions from Nat to $\{0,1\}$
10. Compute the cardinality of the set of all functions from Nat to Nat
11. Derive an exact bound for an LBA to be deemed to be looping. Its input is P long, the LBA has Q states, and G tape symbols.
12. Show that the set of RE languages is closed under Kleene star.
13. Show that RE languages are not closed under complement.
14. Show that if there is a printer TM that prints strings in lexicographic order, then the language is recursive.
15. Show that Turing machines with multiple tapes are equivalent to Turing machines with a single tape
16. Show that a non-deterministic TM can be simulated by a deterministic TM
17. Convert a PDA to a CFG for an extremely simple PDA.
18. Convert a CFG to a PDA.
19. Show that a PDA that accepts by empty stack is equivalent to a PDA that accepts by final state.
20. Show that a PDA that accepts by final state is equivalent to a PDA that accepts by empty stack.
21. Show that given two PDAs P1 and P2, whether P1's language is contained in P2's language is not decidable. Thus what can you say about the nature (RE, recursive) of P1 and P2 pairs such that L(P1) is included in L(P2)?