UNID Name:

CS 3100 – Open book/notes – Midterm-2 – TOTAL 75 points
PART-1 – Total 30 points (6 qns, 5 pts each)

The multiple choice problems given here can earn you positive points (correct answer) or and negative points (incorrect). We show it as [**+m/-n**]. **You must write a compact two-sentence (approx.) explanation in support of your answer, without which you won't gain any points. This explanation may help you earn partial credits for incorrect answers.**

1. [**+5/-1**] **Choose from various answers below.**

   A. Language $\{0, 11\}$ is context-free. **(True; it is even regular.)**

   B. There is a deterministic Push-down automaton for the language of even length palindromes. **(False; need to guess the midpoint.)**

   C. There are some ambiguous context-free grammars that can be disambiguated (rewritten to become unambiguous). **(True; we saw the if/then grammar or the expression grammar.)**

   D. Every context-free grammar can be disambiguated. **(False; there are inherently ambiguous languages)**

- ☐ All these assertions are true.
- ☐ Assertions A, B, and C are true.
- ☐ Assertion A and C alone are true. **(This one)**
- ☐ Assertions A and D are true.

2. [**+5/-1**] **Choose from various answers below.**

   A. For infinite sets A and B, if $A \subset B$, the cardinality of A is different from that of B. **(False; we saw how Odds are a subset of Nat; yet they have the same cardinality.)**

   B. The cardinality of the set of all natural numbers Nat is the same as that of set of all C programs. **(True; we even proved this in class. Remember the trick of adding ; and the use of the Schröder-Bernstein Theorem?)**

   C. The cardinality of Nat is the same as the cardinality of the set of all $Real$ numbers **(False; this was shown by diagonaization.)**

   D. The cardinality of the set $[0, 1) \subset Real$ is the same as that of $[1, \infty) \subset Real$. **(True; We showed this in class. Remember the function $1/x + 1$ that one of you proposed?)**

- ☐ None of these assertions are true.
- ☐ Assertions B and D alone are true. **(This one.)**

- ☐ Assertion C and D alone are true.
- ☐ Assertion A alone is true.

3. **[+5/-1] The following assertions (true or false) have been made about an arbitrary context-free grammar. Choose from various answers below.**

   For every context-free grammar, there is a

   A. language equivalent left-linear context-free grammar **(False; not every CFG can be turned regular which is what left-linear grammars are!)**

   B. language equivalent deterministic finite automaton **(False; same reason as above.)**

   C. language equivalent push down automaton **(True; we can build a PDA for a CFG)**

   D. language equivalent Turing machine **(True; we can build a TM that matches the language of any CFG.)**

   - ☐ Assertions A, B, and C are true
   - ☐ Assertions C and D are true **(This one.)**
   - ☐ Assertion C alone is true
   - ☐ Assertions A, C, and D are true

4. **[+5/-1] The following assertions (true or false) have been made about an arbitrary context-free language. Choose from the answers below.** *Every context-free language*

   A. is regular **(False; CFLs are not always regular.)**

   B. is recursive enumerable **(True; CFLs are RE.)**

   C. is recursively enumerable but not recursive **(False; this is nonsense.)**

   D. is recursive **(True; we can decide membership in a CFP.)**

   - ☐ Assertions B and D are true **(This one.)**
   - ☐ Assertions A and B are true
   - ☐ Assertion C alone is true
   - ☐ Assertion D alone is true

   Explanation:

5. **[+5/-1] Consider the following assertions.**

   A. For every recursive language L, there is a Turing machine M with $\overline{L}$ as its language. **(This is true. Recursive languages have TMs for them and their complement.)**

B. For every non-determinstic push-down automaton, there is an equivalent deterministic push-down automaton. **(This is false; these are incomparable in power.)**

C. The language S_{TM} of codes of self-denying Turing machines is recursively enumerable. **(This is false; we showed it through diagonalization.)**

D. Non-determinstic Turing machines are strictly more powerful than Determinstic Turing machines. **(This is false.)**

- ☐ Assertions A and B are true
- ☐ Assertions B and D are true
- ☐ Assertions A and C are true
- ☐ Assertion A alone is true **(This one.)**

6. **[+5/-1]** Consider the CFG where `e` stands for ε ("Epsilon")

```
S -> 0 S | 1 S | e | T | U

T -> 0 T 1 | e

U -> U 0 | 2 U
```

A. The language of S is infinite **(This is true; nature of the productions.)**

B. There is/are non-terminal(s) that is/are not generating **(This is true; U is the non-terminal.)**

C. The language of S is regular. **(This is true because the S production generates Σ^* that absorbs and makes "disappear" the CFL generated by T.)**

D. The language of S is context-free and not regular **(This is false.)**

- ☐ Assertions A, B, and C are true. **(This one.)**
- ☐ Assertions A, B, and D are true.
- ☐ Assertion B and D alone are true.
- ☐ Assertion D alone is true.

PART-2 – Total 45 points – You can earn up to 50 points (extra built-in)

1. (7.5 points) Design a context-free grammar for the language

$$L = \{a^i b^j c^k \ \mid \ i, j, k \geq 0, \ and \ if \ i = 1 \ then \ j = k\}$$

Explain in about three sentences why this CFG is correct.

```
S -> a M | a a R N | N

M -> b M c | e

R -> a R | e

N -> P Q

P -> b P | e

Q -> c Q | e
```

We go through the different cases.

2. (7.5 points) Design a PDA for L directly—i.e., without converting the CFG to a PDA Explain in about three sentences why this PDA is correct.

   Answer: This PDA will simply non-determinstically branch into three possible futures. The first (when there are no as or two as) will merge to look for b^* and then c^*. The second (when there is a single a) will use the stack to match.

3. (10 points) Design a deterministic Turing machine that recognizes the language $10(0+1)^*$

   ```
   Make the TM read a 1 and a 0, then accept, ignoring the
   rest of the string.
   ```

4. (5 points) Solve the following PCP instance. Express your solution in terms of a sequence of tile numbers with repetitions allowed.

   ```
   Tile numbers: 1   2
                 00  0000
                 000 0

   1 1 1 2  will do
   ```

5. (10 points) Why is the language

   $$\{\langle G_1, G_2\rangle \mid L(G_1) \neq L(G_2)\}$$

   recursively enumerable? Clearly explain, sketching the workings of the TM whose language this is.

   ```
   We can enum strings and find out the difference in the grammars
   for all grammar pairs that exhibit such a difference.
   ```

6. (5 points) Why is

   $$\{\langle G_1, G_2\rangle \mid L(G_1) = L(G_2)\}$$

   not recursively enumerable? Clearly explain.

```
Then this language will then become recursive.
But it is known that we cannot decide
if a CFG generates $\Sigma^*$---and we can make $G_2$ be one that
encodes $\Sigma^*$.
```

7. (10 points) Someone wanted to directly show that $Halt_{TM} = \{\langle M, w\rangle \mid M \textit{ is a } TM \textit{ that halts on input } w\}$ is not recursive. They wrote a proof as follows. Please help them finish the proof.

```
Let H be a decider for the above language.
H(x,x) is asking if x halts when applied to x.
Define a machine D as follows.

D(x) {
 If H(x,x) accepts, D goes into an infinite loop;
 Else D accepts;
}
```

Now help the person finish the proof by arguing what the call $D(D)$ results in. Provide the full case analysis.

ANSWER:

```
If D(D) loops as per D's defn, that is when D halts on D as per
H's defn, because H(D,D) accepts, or D halts (does not loop) on D.

If D(D) halts as per D's defn, that is when D loops as per
H's defn, because H(D,D) rejects, or D does not halt (loops) on D.

Both cases result in a contradiction.
```