Name: ______________________________

1 True/False

Answer whether the following statements are true or false and briefly explain your answer.

a) [TRUE/FALSE] If A is Turing-recognizable and $\overline{A}$ is Turing-recognizable, then $\overline{A}$ is Turing-decidable. [5 pts]

True. We could create a decider for $\overline{A}$ by running the recognizers for A and $\overline{A}$ in parallel.

b) [TRUE/FALSE] Let A be an NP-Hard problem. A poly-time solution to A means that *all* NP-Hard problems are solvable in polynomial time. [5 pts]

False. This would solve all problems in NP in poly-time, but some NP-hard problems may be harder than (and therefore not poly-time reducible to) others.

c) [TRUE/FALSE] A decision problem A is NP-Hard if and only if SAT $\leq_p A$. [5 pts]

True. Problems are NP-hard if every problem in NP is reducible to them, and SAT $\in$ NP. Furthermore, since every problem in NP is poly-time reducible to SAT, there is no way for SAT to be poly-time reducible to a problem that was not NP-hard.

d) [TRUE/FALSE] There exists a pushdown automaton to decide every context-free language. [5 pts]

True. By definition, a language is context-free if and only if some pushdown automaton exists to decide it.

2 Proofs

a) Let the operator $\diamond$ be defined as follows:

$$A \diamond B = \{st \mid s \in A \text{ and } t \in B \text{ and } |s| = |t|\}$$

Show that Turing-recognizable languages are closed under the $\diamond$ operator. [10 pts]

Assume that A and B are Turing-recognizable languages. Then there exist machines M_A and M_B that recognize them. We can construct a two-tape machine that decides $A \diamond B$ as follows:

M = "On input w:

(a) If w contains an odd number of characters, REJECT

(b) Copy every other character — starting with the second — to tape 2, marking each as we go.

(c) Remove all marked characters from tape 1

(d) Nondeterministically run M_A on the contents of tape 1 and M_B on the contents of tape 2 in parallel.

  i. If both machines accept, ACCEPT

  ii. If either machine rejects, REJECT"

b) Let $A = \{\langle D \rangle \mid D$ is a DFA that doesn't accept any string containing an odd number of 1s$\}$. Show that A is decidable. [10 pts]

$M =$ "On input $\langle D \rangle$:

(a) Let $C = \{w \mid w$ contains an odd number of ones$\}$. Construct a machine D' that decides $L(D) \cap C$

(b) Mark the start state of D'.

(c) Repeat until no new states are marked:

   i. Mark any state in D' reachable from a marked state

(d) If any final state in D' is marked, Reject

(e) Accept"

c) Let $\textsc{Comp}_{\mathrm{TM}} = \{\langle M_1, M_2\rangle \mid L(M_1) = \overline{L(M_2)}\}$. Show that $\textsc{Comp}_{TM}$ is undecidable. [10 pts]

We can create a mapping reduction from A_{TM} as follows:

$F =$ "On input $\langle M, w\rangle$:

(a) Construct a machine M_1 that decides $\emptyset$

(b) Construct a machine M_2 as follows: $M_2 =$ "On input x:

  i. Simulate M on w.

    A. If M accepts w, Accept x

    B. If M rejects w, Reject x"

(c) Output $\langle M_1, M_2\rangle$"

d) Let DOUBLESAT $= \{\langle\Phi\rangle \mid \Phi$ is a Boolean formula with (at least) two different satisfying assignments$\}$. Show that DOUBLESAT $\in$ NP. [10 pts]

We can show that DOUBLESAT $\in$ NP by constructing a poly-time deterministic verifier for it.

Let c_1 and c_2 be truth assignments for the variables in Φ.

We know that SAT is NP-Complete. This means that SAT $\in$ NP, and there exists a poly-time deterministic verifier for SAT. Let n be the number of variables in Φ.

$V =$ "On input $\langle Phi, c_1, c_2\rangle$:

(a) If $c_1 = c_2$, REJECT $O(n)$

(b) Run $\langle\Phi, c_1\rangle$ through a verifier for SAT. If this verifier rejects, REJECT Poly-time

(c) Run $\langle\Phi c_2\rangle$ through a verifier for SAT. If this verifier rejects, REJECT Poly-time

(d) ACCEPT" $O(1)$