Name ______________________ Student No. ______________________

No aids allowed. Answer all questions on test paper. Use backs of sheets for scratch work.

Total Marks: 50

[10] 1. The *parity* of a binary string is 1 if the string has an odd number of 1s, and 0 if the string has an even number of 1s. For example, parity(001011) = 1, while parity(11) = 0 and parity(0000) = 0. Formally, $\text{parity}(w) = \left(\sum_{i=1}^{|w|} w_i\right) \pmod 2$.

Let $L_{\text{par}} = \{w | \text{parity}(w) = 1\}$.

Design a Turing machine which decides the language L_{par}. (Note that your Turing machine should be simple; this question should take you little time.)

Solution: Let M be a Turing machine with $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, B\}$, and with $Q = \{q_{\text{init}}, q_0, q_1, \text{yes}, \text{no}\}$, where q_{init} is the initial state, $F = \{\text{yes}\}$, and both "yes" and "no" are halting states—thus "no" is a rejecting state.

The transition function δ is:

$$\begin{aligned}
&\delta(q_{\text{init}}, 0) = (q_0, 0, \rightarrow)\\
&\delta(q_{\text{init}}, 1) = (q_1, 1, \rightarrow)\\
&\delta(q_0, 0) = (q_0, 0, \rightarrow)\\
&\delta(q_0, 1) = (q_1, 1, \rightarrow)\\
&\delta(q_1, 0) = (q_1, 0, \rightarrow)\\
&\delta(q_1, 1) = (q_0, 1, \rightarrow)\\
&\delta(q_0, B) = \text{no}\\
&\delta(q_1, B) = \text{yes}
\end{aligned}$$

[10] 2. Recall from class that the language $L_U = \{\langle M, w\rangle |\ M \text{ accepts } w\ \}$ is *not* recursive. Here $\langle M, w\rangle$ denotes an encoding of the pair M (a Turing machine) and w (an input).

Let $L_H = \{\langle M, w\rangle |\ M \text{ halts on } w\}$ be the *Halting Language*, i.e., the set of encodings of pairs M, w where M is a Turing machine that halts on input w.

Using the fact that L_U is not recursive, show directly (i.e., without Rice's Theorem) that L_H is also not recursive.

Solution: Suppose that L_H is recursive. Then, there exists a Turing machine M_1 that decides L_H, i.e., $L_H = L(M_1)$ and M_1 always halts.

We can therefore design a Turing machine M_2 which decides L_U as follows: on input $\langle M, w\rangle$, M_2 simulates M_1 on $\langle M, w\rangle$. If M_1 accepts, it means that M halts on w, so M_2 now simulates M on w to find out if it accepts. If it accepts, M_2 also accepts. If, on the other hand, M rejects w, then M_2 also rejects.

If M_1 does not accept, it means that M does not halt on w, so M_2 directly rejects.

Note that M_2 always halts, and $L_U = L(M_2)$, which means that L_U is recursive. Contradiction.

3. Let $L_{\text{ne}} = \{w | w = w_i \text{ and } L(M_i) \neq \emptyset\}$, that is, L_{ne} is the set of strings that encode Turing machines which accept at least one string.

[5] (a) Show that L_{ne} is RE.

Solution: Let N be a non-deterministic Turing machine, which on input $w = w_i$, "guesses" a string x, and simulates M_i on x (using the Universal Turing machine U as a subroutine). Clearly, $L(N) = L_{\text{ne}}$, and so L_{ne} is RE.

[5] (b) Use Rice's Theorem to show that L_{ne} is undecidable (i.e., not recursive).

Solution: Let $\mathcal{P} = \text{RE} - \{\emptyset\}$, so $L_{\mathcal{P}}$ is the set of encodings of Turing machines which accept at least one string.
Note that $\mathcal{P}$ is non-trivial, because it is not everything (it does not contain $\emptyset$) and it is not empty because there are Turing machines which accept at least one string, and their languages are in $\mathcal{P}$. Also, $\mathcal{P} \subseteq \text{RE}$.
Therefore, we can use Rice's Theorem to conclude that $L_{\mathcal{P}}$ is undecidable. But $L_{\mathcal{P}} = L_{\text{ne}}$, and hence L_{ne} is also undecidable.

[5] 4. (a) Show that a language L is recursive $\iff$ both L and $\overline{L}$ are RE.

Solution: If L is recursive, so is $\overline{L}$, and Recursive $\subseteq$ RE. For the other direction, assume that both L and $\overline{L}$ are RE. Then there exist TMs M_1 and M_2 such that $L = L(M_1)$ and $\overline{L} = L(M_2)$. Let M be a TM which on input w simulates M_1 on one tape and M_2 on the second tape, both on input w. Since $L(M_1) \cup L(M_2) = \Sigma^*$, it follows that at some point M_1 or M_2 accepts (and not both!). If M_1 accepts, so does M. If M_2 accepts, M rejects. Note that $L(M) = L$, and M always halts, so L is recursive.

[5] (b) Let $L_{\mathrm{e}} = \overline{L_{\mathrm{ne}}}$, that is, L_{e} is the complement of the language L_{ne} defined in question 3. Use questions 3 and 4a to show that L_{e} is *not* RE.

Solution: By question 3 L_{ne} is RE but not recursive. Therefore, L_{e} cannot be RE as that would imply by 4a that L_{ne} is recursive.

[10] 5. Suppose that we limit the Post's Correspondence Problem (PCP) to a one symbol alphabet, for example $\Sigma = \{0\}$. Would this restricted case of PCP still be undecidable? (No marks for an answer without justification.)

Solution: The problem *is decidable* by the following algorithm: First, if all the A-strings are strictly longer than their corresponding B-strings, then there can be no solution. Neither is there a solution in the opposite case, where all the B-strings are strictly longer than their corresponding A-strings.

In all other cases, there is a solution:

If any corresponding pair of strings are the same length, then they are identical, and so just that pair is a solution.

If there is no pair of the same length, then there exists at least one pair (w_i, x_i) and at least one pair (w_j, x_j) such that $|w_i| - |x_i| = m > 0$ and $|x_j| - |w_j| = n > 0$.

Then, $|w_i^n w_j^m| = n|w_i| + m|w_j| = n(m + |x_i|) + m(|x_j| - n) = n|x_i| + m|x_j| = |x_i^n x_j^m|$. Since there is only one symbol, i.e., $\Sigma = \{0\}$, it follows that $w_i^n w_j^m = x_i^n x_j^m$ and we have a solution.