Name ____________________ Student No. ____________________

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

Total Marks: 50

[10] 1. Design a Turing machine which decides the set of strings over $\{0, 1\}$ which have more 1s than 0s. For example, 01101 would be in this set, but 001 or 10 wouldn't.

Give an informal description of your Turing machine in English; you do *not* have to give a formal description with δ's.

A significant portion of the marks will be awarded for the clarity of your solution.

Solution: Use a 3 tape TM. The first head scans the input left to right. Every time it sees a 0 (1) it writes a 1 on the second (third) tape.

When the first head reaches the end of the input, it has a string on 1s on the second (third) tape, and the second (third) head is positioned on the last 1.

It now scans the 1s on the second and third tape left to right. If the second head reaches a blank sooner (or at the same time) as the third head, it rejects. Otherwise, it accepts.

[10] 2. Consider the nondeterministic Turing machine

$$M = (\{q_0, q_1, q_2, q_f\}, \{0,1\}, \{0,1,B\}, \delta, q_0, B, \{q_f\})$$

Describe the language $L(M)$ if δ consists of the following sets of rules:

$$\begin{aligned}
\delta(q_0, 0) &= \{(q_0, 1, R), (q_1, 1, R)\} \\
\delta(q_1, 1) &= \{(q_2, 0, L)\} \\
\delta(q_2, 1) &= \{(q_0, 1, R)\} \\
\delta(q_1, B) &= \{(q_f, B, R)\}
\end{aligned}$$

and prove (using induction on the length of strings) that your language is the correct one.

Solution: The language of M is the set of strings that start with zero, i.e., $0(0+1)^*$.

Here is the proof by induction. The Basis Case is $w = 0$ or $w = 1$. In the first case, $q_0 0 \vdash 1q_1 \vdash 1Bq_f$. On the other hand, if $w = 1$, then there is no transition, so this branch dies.

Induction Step: consider $0w_1w_2 \dots w_n$ (if the string starts with 1, again there is no transition out of q_0 on 1, so this branch dies). Starting with $q_0 0 w_1 w_2 \dots w_n$, look for the largest i such that $w_1 w_2 \dots w_i = 0^i$. Then, this yields $11 \dots 1q_0 01 w_{i+2} w_{i+3} \dots w_n$. Now we do the following:

$$\begin{aligned}
&11 \dots 1q_0 01 w_{i+2} w_{i+3} \dots w_n && \text{on } (q_1, 1, R) \\
\vdash &11 \dots 11 q_1 1 w_{i+2} w_{i+3} \dots w_n && \text{on } (q_2, 0, L) \\
\vdash &11 \dots 1 q_2 10 w_{i+2} w_{i+3} \dots w_n && \text{on } (q_0, 1, R) \\
\vdash &11 \dots 11 q_0 0 w_{i+2} w_{i+3} \dots w_n &&
\end{aligned}$$

We know by induction hypothesis that $q_0 0 w_{i+2} w_{i+3} \dots w_n$ yields an accepting configuration, and so we are done.

[10] 3. Let L be the language consisting of pairs of Turing machine codes plus an integer, (M_1, M_2, k), such that $L(M_1) \cap L(M_2)$ contains at least k strings. Show that L is RE, but not recursive.

Solution: It is RE because the following TM M accepts it: on input (M_1, M_2, k) it enters the following loop:

For $i = 1 \ldots \infty$
For $j = 1 \ldots i$
Simulate M_1 on w_j for i many steps,
and then simulate M_2 on w_j also for i many steps.
If both accept, write w_j on another tape.
After the j loop ends, it checks the number of *different*
w_j's on the other tape.
If their number is $\geq k$, accept.

Note that $w_1, w_2, w_3, \ldots$ is the list of all strings over the input alphabet (in, say, lexicographic order).

To see that it is *not* recursive, we reduce it to L_{ne} (the language of TMs whose languages are not empty, seen in class to be not recursive).

Let E be a very simple TM that on any input, just accepts, so $L(E) = \Sigma^*$.

Given any TM M, $L(M) = \emptyset$ iff $|L(M) \cap L(E)| \geq 1$.

Therefore, $(M, E, 1) \in L$ iff $L(M) \neq \emptyset$.

[10] 4. Recall the Post Correspondence Problem (PCP).

(a) State precisely the definition of PCP.

(b) Give an outline of the proof that PCP is undecidable.

Solution: See course slides.

[10] 5. Define the problem of testing for the ambiguity of grammars as a language. Use the undecidability of PCP to show that this language is not recursive.

Solution: See course slides.