This homework is *optional* and is due Wednesday December 6. Please place it in the box marked "CSc 473" in the department mailroom, Gould-Simpson 713, by 4:00pm that day. Questions are drawn from the material in Chapters 3 and 4 of the text on Turing machines and decidability.

In problems that ask you to show that a task can be performed by a Turing machine, unless you are explicitly asked to give a transition diagram you may describe your machine by giving a very-high level description in prose of the algorithm that it carries out.

The homework is worth 100 points. When questions with several parts do not specify the points for each part, each part has equal weight.

Remember to write on just one side of a page, do not use scrap paper, put your answers in the correct order, and staple your pages together. If you can't solve a problem, state this, and write only what you know to be correct. Neatness and conciseness count.

(1) **(Move-then-read Turing machines)** (20 points) In our definition of a Turing machine, a transition is labeled by

$$a \to b,\, D$$

where $a$ and $b$ are tape symbols and $D \in \{\mathtt{L}, \mathtt{R}\}$ is a head direction. The action on such a transition proceeds in three successive steps:

(1) read the symbol at the current head location; if it is $a$, then
(2) write $b$ at the current head location, and
(3) move the head in direction $D$.

The transition is taken only if symbol $a$ is at the current head location.

Suppose we define a Turing-machine *variant* called a **move-then-read** Turing machine, where a transition is instead labeled by

$$D,\, a \to b$$

and the action on a transition proceeds according to the following sequence of steps:

(1′) first move the head in direction $D$, and then
(2′) read the symbol at the new head location; if it is $a$,
(3′) write $b$ at the new location.

Note that now the transition is taken only if symbol $a$ is at the new head location; if $a$ is not at the new location, the transition is not taken and the original head position is not changed.

The following questions ask you to show that move-then-read Turing machines and standard Turing machines are equivalent.

(a) Show how to *simulate* a standard Turing machine $M$ by a move-then-read Turing machine $\widetilde{M}$ that follows this new sequence of actions on a transition. In particular, by drawing a *transition diagram*, describe in detail how to map a transition of $M$ into equivalent transitions performed by $\widetilde{M}$.

(b) Now show how to simulate a move-then-read Turing machine $\widetilde{M}$ by a standard Turing machine $M$, again by drawing a transition diagram.

(2) **(Two-way infinite tapes)** (20 points) A Turing machine with a *two-way infinite tape* is a standard Turing machine except its tape extends infinitely to the left, as well as to the right. Initially the tape is filled with blanks except where the input is written, and the head is at the leftmost input symbol. Computation proceeds as usual, except the head never encounters a left end on moving left.

Prove that with a two-way infinite tape, Turing machines recognize no more than the standard class of Turing-recognizable languages.

(3) **(Two stacks are better than one)** (20 points) Consider a pushdown automaton that has $k$ stacks, where $k \geq 0$. Such a machine behaves like a normal PDA, except each transition independently updates all $k$ of its stacks. A 0-stack PDA is equivalent to an NFA, while a 1-stack PDA is just an ordinary PDA.

(a) Prove that a 2-*stack* pushdown automaton is equivalent to a Turing machine.

(Hint: Show how to simulate a Turing machine with one tape by a PDA with two stacks.)

(b) Prove that a 3-*stack* pushdown automaton is no more powerful than a 2-stack pushdown automaton.

(Hint: Argue that by Part (a), it suffices to show how to simulate a 3-stack PDA by a Turing machine.)

(4) **(Turing machines with rewind)** (20 points) Suppose we redefine the way the tape head is moved on a Turing machine so that after reading and writing a symbol the head can be moved *right* one cell (R), or can be moved all the way *back* to the left end of the tape (B). (So now the head cannot be directly moved one cell to the left.) Show that every Turing recognizable language can also be recognized by this new type of Turing machine.

(Hint: Show how to simulate moving the head one cell to the left, making use of the operation that rewinds the head back to the left end of the tape.)

(5) **(Deciding containment of regular expressions)** (20 points) Show that the problem of testing whether $L(R) \subseteq L(S)$ for two given regular expressions $R$ and $S$ is decidable.

(Hint: Test whether $L(R) - L(S)$ is empty, by first converting $R$ and $S$ to DFAs.)

(6) **(Bonus) (Separating co-Turing-recognizable languages)** (10 points) Let $A$ and $B$ be any two disjoint languages whose complements are Turing-recognizable. Prove there always exists a decidable language $C$ that *separates* $A$ and $B$, in the sense that $A \subseteq C$ and $B \subseteq \overline{C}$.

Note that Problem (6) is a bonus question. It is not required (except for the honors section) and its points are not added to regular points.