Each of the problems is worth 10 points. Please write your solutions clearly and concisely. If you do not explain your answer you will be given no credit. You are encouraged to collaborate on the homework, but you *must* write your own solutions and list your collaborators on your solution sheet. Copying someone else's solution will be considered plagiarism and may result in failing the whole course.

Please turn in the solutions by 11.59pm on Thursday 13 November. The homework should be dropped off in the box labeled CSC 3130 on the 9th floor of SHB. Late homeworks will not be accepted.

Problem 1

Argue that the following variants of Turing Machines are equivalent to the ordinary Turing Machine (namely, they recognize the same class of languages). Assume for simplicity that the input alphabet is $\Sigma = \{0, 1\}$ and the tape alphabet Γ is arbitrary.

(a) Turing Machines with a "delete key". Apart from moving its head left (L) and right (R), this kind of machine can also execute a delete (D) transition which erases the current tape symbol and moves everything to the right of it one square to the left. For example, if the tape contents are $\mathtt{00\check{1}100}\square\square\cdots$ upon a delete transition they become $\mathtt{00\check{1}00}\square\square\cdots$. (Here $\check{\ }$ indicates the tape head position.)

(b) Turing Machines where the tape alphabet is $\Gamma' = \{0, 1, \square\}$.

Problem 2

Prove that the following statements are true.

(a) If L_1 and L_2 are recognizable, then $L_1 \cap L_2$ is recognizable.

(b) If L_1 and L_2 are recognizable, then $L_1 L_2$ is recognizable.

(c) If L is recognizable, then L^* is recognizable.

(d) If L_1 is recognizable and L_2 is decidable, then $L_1 - L_2$ is recognizable. Recall that

$$L_1 - L_2 = \{x : x \in L_1 \text{ and } x \notin L_2\}.$$

Problem 3

For each of these languages, say whether it is (i) decidable, (ii) recognizable but not decidable, or (iii) not recognizable. Justify your answer using methods from class.

(a) $L_1 = \{(\langle M \rangle, w) : \text{TM } M \text{ accepts all strings shorter than } w.\}$

(b) $L_2 = \{(\langle M_1 \rangle, \langle M_2 \rangle) : \text{TM } M_1 \text{ accepts input } \langle M_2 \rangle \text{ or TM } M_2 \text{ accepts input } \langle M_1 \rangle \text{ (or both).}\}$

(c) $L_3 = \{\langle M \rangle : M \text{ accepts finitely many inputs.}\}$

Problem 4

In class we showed that any language recognized by a non-deterministic Turing Machine is also recognized by an ordinary Turing Machine. More precisely, for every nondeterministic Turing Machine N we gave an ordinary Turing Machine M with an input tape x, a simulation tape z and an address tape a that does the following:

> Initially, x contains the input of N, and z and a are empty.
> For all possible strings a:
>   Copy x to z
>   Simulate N on input z using a as the nondeterminism.
>   If a specified an invalid choice or
>   simulation runs out of choices, abandon simulation.
>   If N enters its accept state, accept and halt.
>   If N enters its reject state, abandon simulation.

(a) Give an example of a language L and machine N such that N *decides* L, but M does not decide L.

(b) Describe how to modify M such that if N decides L, then M also decides L.

(c) Now consider the following argument that all context-free languages are decidable:

> Every context-free language L has a pushdown automaton. This pushdown automaton can be simulated by a nondeterministic Turing Machine. Since nondeterministic Turing Machines are just as powerful as ordinary Turing Machines, it follows that L is decidable.

What is wrong with this argument?

(d) Give a sound argument that context-free languages are indeed decidable. You may use the Church-Turing Thesis.