Problem Set 2 (due Friday, February 6)

1. ($4 \times 5 = 20$ points) Classification of languages

Each of the following three parts gives a definition, description, or some properties of a language. In each case, tell whether the language is:

A Turing-decidable

B Turing-recognizable but not Turing-decidable

C not Turing-recognizable

Justify your answers with proofs, constructions, algorithms, or examples as needed.

(a) The set of descriptions of all Turing machines that halt on every input.

(b) The set of descriptions of all Turing machines that halt on at least one input.

(c) The set of descriptions of all Turing machines that ever write the symbol a on some input; i.e., the set of all Turing Machines M for which there exists an input w such that M writes the symbol a at some point during its computation on w.

(d) The set of descriptions of all Turing machines which ever write a symbol different than the one currently scanned by the head on some input; i.e., the set of all Turing Machines M for which there exists an input w such that M writes a symbol different than the one currently scanned by the head at some point during its computation on w.

2. ($3 \times 5 = 15$ points) Context-sensitive and context-free languages

Recall that context-sensitive grammars are grammars in which the length of the left-hand side of any production rule is no more than the length of the right-hand side.

(a) Give a context-sensitive grammar for the language $\{ww : w \in \{0,1\}^*\} - \{\varepsilon\}$.

We briefly discussed in class that every context-sensitive language is accepted by a (nondeterministic) linear-bounded Turing machine. Answer the following two questions, giving formal justifications.

(b) Is every context-free language accepted by a *deterministic* linear-bounded Turing machine?

(c) Is every language accepted by a *deterministic* linear-bounded Turing machine context-free?

3. ($3 \times 5 = 15$ points) Turing machines, languages, and partial functions

Consider a function f from nonnegative integers to nonnegative integers. The *graph* of f is defined as the set

$$\{(x, f(x)) : x \text{ is a nonnegative integer}\}.$$

We say that a Turing machine *computes* f if for every input x (in binary), it halts with $f(x)$ on its tape, written in binary.

(a) Given a TM that computes f, show how to construct a decider that accepts the graph of f.

(b) Given a decider for the graph of f, show how to construct a TM that computes f.

(c) We say that a function is *partial* if it may be undefined on some arguments. The graph of f is then defined to be the set of all pairs $(x, f(x))$ over all x on which f is defined. We now do not require for a TM computing f to halt on the inputs for which f is undefined. Do your constructions for parts (a) and (b) hold for partial functions? If not, explain how you would modify the constructions to make them work with recognizers instead; i.e., do (a) and (b) with "decider" replaced by "recognizer".

4. (10 points) Length-ordered enumeration of strings

Show that a language L is decidable if and only if there exists an enumerator for L that enumerates the strings of L in length-increasing order. (That is, the enumerator prints all strings of L of length ℓ before printing any string of L of length greater than ℓ.)

Does the claim continue to hold if we require enumeration in lexicographic order?

5. (10 points) Post Correspondence Problem with small strings

Show that the Post Correspondence Problem is undecidable if strings are restricted to be of length one or two. What happens if the strings are restricted to have length exactly two?