Midterm Exam 2 Solutions

1. Define a language B as follows:

$$B = \left\{ \langle M, x \rangle \; : \; \begin{array}{l} M \text{ is a DTM, } x \text{ is a string over the input alphabet of } M \text{, and} \\ M \text{ accepts at least one string } y \text{ with } y \leq x \end{array} \right\}$$

As usual, the notation $y \leq x$ means that either $y = x$ or y comes before x with respect to the lexicographic ordering (of strings over the input alphabet of M).

(a) Prove that B is Turing-recognizable.

Solution. Define a DTM K as follows.

On input $\langle M, x \rangle$, where M is a DTM and x is a string over the input alphabet of M:

1. Set $t \leftarrow 1$.
2. Simulate M for t steps on every string $y \leq x$.
3. If M accepts in any of the simulations in step 2, then accept.
4. Set $t \leftarrow t + 1$ and goto 2.

It is clear that $L(K) = B$, and therefore B is Turing-recognizable.

(b) Prove that B is undecidable. To do this, it may help you to recall that the following languages have all been proved to be undecidable in class:

$$\begin{aligned} \text{A} &= \{\langle M, x \rangle \; : \; M \text{ is a DTM and } x \in L(M)\}\,, \\ \text{E} &= \{\langle M \rangle \; : \; M \text{ is a DTM and } L(M) = \varnothing\}\,, \\ \text{AE} &= \{\langle M \rangle \; : \; M \text{ is a DTM and } \varepsilon \in L(M)\}\,, \\ \text{EQ} &= \{\langle M_1, M_2 \rangle \; : \; M_1 \text{ and } M_2 \text{ are DTMs with } L(M_1) = L(M_2)\}\,. \end{aligned}$$

Solution. We will reduce AE to B, using the observation that $\langle M, \varepsilon \rangle \in B$ if and only if M accepts ε. With this in mind, let f be a function defined as $f(\langle M \rangle) = \langle M, \varepsilon \rangle$ for every DTM M; and as usual we assume $f(y)$ is defined to be some fixed element of $\overline{B}$ whenever $y \neq \langle M \rangle$ for all DTMs M. It is clear that f is computable. We have

$$\langle M \rangle \in \text{AE} \Leftrightarrow f(\langle M \rangle) = \langle M, \varepsilon \rangle \in B,$$

and therefore AE $\leq_m B$. As AE is undecidable, it follows that B is undecidable as well.

2. For both of the following two statements, decide whether the statement is **true** or **false**. If you decide the statement is **true**, give a proof, and if you decide the statement is **false**, give a counter-example. You may assume that $\Sigma = \{0, 1\}$ for both statements.

(a) For every undecidable language $A \subseteq \Sigma^*$, the language

$$B = \{0x \; : \; x \in A\} \cup \{1x \; : \; x \notin A\}$$

is not Turing-recognizable.

Solution. True. Consider the functions $f_0(x) = 0x$ and $f_1(x) = 1x$ (for all $x \in \Sigma^*$), which are obviously computable. It holds that f_0 is a mapping reduction from A to B and f_1 is a mapping reduction from $\overline{A}$ to B, and therefore

$$A \leq_m B \qquad \text{and} \qquad \overline{A} \leq_m B.$$

Under the assumption that B is Turing-recognizable it therefore holds that both A and $\overline{A}$ are Turing-recognizable, and thus A is decidable. Given that A is assumed not to be decidable, it follows that B cannot be Turing-recognizable.

(b) For every language $A \subseteq \Sigma^*$ satisfying $A \leq_m \overline{A}$, it holds that A is decidable.

Solution. False. Consider the language B from part (a) for any choice of A that is not decidable. As proved in part (a), any such B is not Turing-recognizable, and is therefore undecidable. It remains to prove that $B \leq_m \overline{B}$. Define f as $f(0x) = 1x$ and $f(1x) = 0x$ for all $x \in \Sigma^*$, and let

$$f(\varepsilon) = \begin{cases} 0 & \text{if } \varepsilon \in A \\ 1 & \text{if } \varepsilon \notin A. \end{cases}$$

Then f is computable and it holds that $x \in B \Leftrightarrow f(x) \in \overline{B}$. Thus $B \leq_m \overline{B}$ as required.

3. Assume that $\Sigma = \{0, 1\}$, and recall that we proved the following fact in class:

   For any non-empty language $B \subseteq \Sigma^*$, we have that B is Turing-recognizable if and only if $B = \text{range}(f)$ for some computable function $f : \Sigma^* \to \Sigma^*$.

   Prove the following variation of this fact:

   For any *infinite* language $B \subseteq \Sigma^*$, we have that B is Turing-recognizable if and only if $B = \text{range}(g)$ for some computable *one-to-one* (or *injective*) function $g : \Sigma^* \to \Sigma^*$.

   You may (and should) use the first fact when proving the second; as always in this class, there is no need for you to re-prove facts on exams that we have already proved in class.

Solution. Suppose first that $B = \text{range}(g)$ for a computable one-to-one function g. Given that B is infinite it is certainly non-empty. By the fact proved in class it follows that B is Turing-recognizable.

Now assume that B is Turing-recognizable. It holds that B is non-empty, given that it is infinite, and therefore by the fact proved in class there exists a computable function f such that $B = \text{range}(f)$. Define a DTM M as follows:

On input x:

1. Let n be the position of x in the lexicographic ordering of Σ^* (starting with $n = 1$ when $x = \varepsilon$).
2. Let $y = \varepsilon$.
3. Compute $f(z)$ for every string $z \leq y$, and count the number of distinct output strings that result. If this number is n, i.e., if $|\{f(z) : z \leq y\}| = n$, then halt and output $f(z)$.
4. Increment y (with respect to the lexicographic ordering of Σ^*) and goto 3.

If follows from the assumption that B is infinite that M halts on all inputs. Taking g to be the function computed by M, it holds that g is one-to-one and $B = \text{range}(g)$ as required.

4. Prove that there exist Turing-recognizable languages $A, B \subseteq \Sigma^*$ that simultaneously satisfy these two properties:

   (a) $A \cap B = \varnothing$, and

   (b) there does <u>not</u> exist a decidable language $C \subseteq \Sigma^*$ such that $A \subseteq C$ and $B \subseteq \overline{C}$.

   Hint: use diagonalization directly for well-chosen languages A and B.

Solution. Define

$$A = \{\langle M \rangle \,:\, M \text{ rejects } \langle M \rangle\},$$
$$B = \{\langle M \rangle \,:\, M \text{ accepts } \langle M \rangle\}.$$

It is clear that $A \cap B = \varnothing$. Now assume toward contradiction that there exists a decidable language C such that $A \subseteq C$ and $B \subseteq \overline{C}$. The language C is decidable, so there must exist a DTM M that decides C. For this choice of M we have

$$\langle M \rangle \in C \;\Rightarrow\; M \text{ accepts } \langle M \rangle \;\Rightarrow\; \langle M \rangle \in B \;\Rightarrow\; \langle M \rangle \in \overline{C} \;\Rightarrow\; \langle M \rangle \notin C$$

and

$$\langle M \rangle \notin C \;\Rightarrow\; M \text{ rejects } \langle M \rangle \;\Rightarrow\; \langle M \rangle \in A \;\Rightarrow\; \langle M \rangle \in C.$$

Therefore $\langle M \rangle \in C$ if and only if $\langle M \rangle \notin C$, which is a contradiction. It follows that there does not exist a decidable language C as claimed.