CIS 511: Spring 2012
Midterm Solutions

(1) Let \( B \) be any language over the alphabet \( \Sigma \). Prove that \( B = B^* \) iff \( BB \subseteq B \) and \( \varepsilon \in B \).

**Answer:** One direction is easy. If \( B = B^* \), then \( BB \subseteq B^* = B \) and \( \varepsilon \in B^* = B \).

Suppose now that \( BB \subseteq B \) and \( \varepsilon \in B \). We prove by induction that \( B^n \subseteq B \) for \( n \geq 2 \).

The base case is \( n = 2 \) is by hypothesis. Then \( B^2 = B^1B \subseteq BB \subseteq B \) (where the first inclusion is by the induction hypothesis). Noting that by hypothesis, \( \varepsilon \in B \), we conclude that for all \( n \geq 0 \), \( B^n \subseteq B \) and thus \( B^* = B \) (since \( B \subseteq B^* \) is trivial).

(2) Let \( \Sigma = \{0, 1, +, =\} \) and \( ADD = \{x = y + z \mid x, y, z \text{ are binary integers, and } x \text{ is the sum of } y \text{ and } z\} \)

**Answer:** Suppose \( ADD \) is regular. Let \( p \) be its pumping length. Take \( w \) to be the string \( 1^p = 1^p + 0 \). Then let \( w = xyz \) be a partition of \( w \) with \( |xy| \leq p \) and \( |y| > 0 \). Note that \( y = 1^k \) for some \( 1 \leq k \leq p \) and that \( 1^p + 0 \) is contained fully in \( z \). Thus \( xy^2z \) is \( 1^{p+k} = 1^k + 0 \), which is not in the language, contradicting the pumping lemma. Thus, \( ADD \) is not regular.

(3) Let \( L = \left\{ u_1 \# u_2 \# \cdots \# u_k \mid u_i \in \{0, 1\}^*, k \geq 2 \text{ and for some } i \text{ between } 1 \text{ and } k-1, u_i+1 \neq u_i^R \right\} \)

Show that \( L \) is context-free by describing a PDA for it.

**Answer:** The PDA will non-deterministically guess the index \( i \) for which \( u_i^R \neq u_{i+1} \). For the guessed \( i \), it will enter a state where it remembers \( u_i \) by pushing each symbol of \( u_i \) onto the stack. It knows what \( u_i \) ends since it is delimited by a \( \# \), so when it sees the \( \# \), it moves to the next state.

In this state, it compares \( u_{i+1} \) to the top symbol of the stack, discarding both, and repeating. If we see 0 from the input and 1 or \( \$ \) on the stack, we go to an accept state – regardless of what else happens, \( u_{i+1} \neq u_i^R \). Similarly for 1. If we see \( \# \) from the input and \( \$ \) on the stack, this means that \( u_i^R = u_{i+1} \) and this branch of computation should end. If we see \( \# \) from the input and 0 or 1 on the stack, this means that \( u_{i+1} \) is shorter than \( u_i^R \), and in particular not equal.

The accept state has a loop to itself on any symbol from the input.

(4) Consider the problem of deciding whether a two-tape Turing Machine ever writes a non-blank symbol on its second tape when it runs on input \( w \). Formulate this problem as a language and show that it is undecidable by reducing \( A_{TM} \) to it.

**Answer:**

\[ L = \left\{ \langle M, w \rangle \mid M \text{ is a Turing Machine, which on input } w, \text{ writes a non-blank symbol on its second tape} \right\} \]

We define the following two-tape TM \( M_1 \). It takes as input a pair \( \langle M, w \rangle \) where \( M \) is a TM and \( w \) is some string. \( M_1 \) will simulate \( M \) using just its first tape (it will ignore the second tape until the end – this is possible since a one-tape TM can simulate any one-tape TM) on input \( w \). If \( M \) accepts \( w \), then \( M_1 \) writes a non-blank symbol from its alphabet on its second tape. If the simulation halts or rejects, we simply stop.

Now, suppose that \( T \) decides \( L \). Then we construct \( T' \) which decides \( A_{TM} \). On input \( \langle M, w \rangle \), it simulates the machine \( M_1 \) as described above, and then run \( T \) on \( M_1 \), and accept if \( T \) accepts, and reject if \( T \) rejects.
We see that $T'$ accepts iff $T$ accepts $\langle M_1, \langle M, w \rangle \rangle$ iff $M_1$ writes on its second tape iff $M$ accepts $w$. Thus, $T'$ decides $A_{TM}$. Since $A_{TM}$ cannot have a decider, we see that there is no $T$ deciding $L$, and so $L$ is undecidable.

(5) Let 
\[ L = \{ \langle M, k \rangle \mid M \text{ accepts some string of length at most } k \} \]
Prove that $L$ is undecidable.

**Answer:** Suppose $L$ is decidable, and let $R$ be a decider for $L$. Let $M_1(M, w)$ be a machine which operates as follows on input $x$: if $x \neq w$, we reject, otherwise, we run $M$ on $w = x$ and accept if $M$ accepts.

We then build the following decider, $S$, for $A_{TM}$: on input $\langle M, w \rangle$, construct $M_1$, and run $R$ on $M_1$. If $R$ accepts, we accept, and if $R$ rejects, we reject.

Now note that $S$ accepts $\langle M, w \rangle$ iff $R$ accepts $M_1(M, w)$ iff $M$ accepts $w$. That is, $S$ decides $A_{TM}$: a contradiction. Thus, no such $R$ exists, and $L$ is undecidable.