Assignment 3 solutions

---

1. Prove that the following languages are decidable, by giving high-level descriptions of DTMs that decide them.

$$A = \left\{ \langle D_0, D_1 \rangle \,:\, \begin{array}{l} D_0 \text{ and } D_1 \text{ are DFAs, and there exists at least one} \\ \text{string that is accepted by both } D_0 \text{ and } D_1 \end{array} \right\},$$

$$B = \big\{ \langle G \rangle \,:\, G \text{ is a CFG for which } \mathrm{L}(G) \text{ is infinite} \big\},$$

$$C = \big\{ \langle D \rangle \,:\, D \text{ is a DFA, and every string accepted by } D \text{ is a palindrome} \big\}.$$

   In each case, you are to assume that reasonable encoding schemes for the objects involved have been selected—and while the languages do indeed depend on the chosen encoding schemes, the fact that they are decidable does not.

Solution.

Proof that A is decidable: Let M be a DTM operating as follows:

On input $\langle D_0, D_1 \rangle$, where D_0 and D_1 are DFAs:

1. Construct a DFA F such that $\mathrm{L}(F) = \mathrm{L}(D_0) \cap \mathrm{L}(D_1)$ using one of the methods for doing this covered in lecture. (The Cartesian product construction is simplest.)
2. Decide whether or not $\langle F \rangle \in \mathrm{E}_{\mathrm{DFA}}$ using the DTM for this language discussed in lecture. *Reject* if $\langle F \rangle \in \mathrm{E}_{\mathrm{DFA}}$, *accept* if not.

It is evident from an inspection of M that this DTM decides A, and therefore A is decidable.

Proof that B is decidable. We will make use of the following fact:

> If G is a context-free grammar in Chomsky normal form having m variables, then $\mathrm{L}(G)$ is infinite if and only if there exists a string $x \in \mathrm{L}(G)$ such that $|x| \geq n = 2^m$.

This fact follows from the proof of the pumping lemma for context-free languages from Lecture 10. With this fact in mind, the following DTM decides B:

On input $\langle G \rangle$ where G is a CFG:

1. Convert G to an equivalent CFG H in Chomsky normal form, and let m be the number of variables in H.
2. Construct a DFA D, having the same alphabet as H, that recognizes the language of all strings having length at least 2^m (which is a regular language).
3. Construct a new CFG K for the language $\mathrm{L}(H) \cap \mathrm{L}(D)$. This is possible through the construction we discussed in Lecture 9 when proving that the intersection of a context-free language and a regular language is context-free.
4. Decide whether or not $\langle K \rangle \in \mathrm{E}_{\mathrm{CFG}}$ using the DTM for this language discussed in Lecture 14. *Reject* if $\langle K \rangle \in \mathrm{E}_{\mathrm{CFG}}$, and *accept* otherwise.

In more detail, if the CFG K generates any strings, then H must generate a string having length at least 2^m, and therefore H generates an infinite number of strings by the fact above (and so does G because H and G are equivalent). If K does not generate any strings, then H must generate a finite number of strings, because there are only finitely many strings in total having length less than 2^m.

An alternative approach to this problem is to directly design an algorithm that inspects the rules of a CFG G (or an equivalent CFG in Chomsky normal form) in various ways. For instance, your solution might have involved a search for a cycle among a collection of variables reachable from the start variable. By using the same algorithm used to decide $\mathrm{E}_{\mathrm{CFG}}$, you could throw away all variables that are not capable of generating any strings, so as to potentially simplify the algorithm.

Proof that C is decidable. Define a DTM M as follows:

On input $\langle D \rangle$, where D is a DFA:

1. Construct a CFG G for the language consisting of all non-palindromes over the alphabet of D. (This was done on Assignment 2 for the alphabet $\{0, 1\}$, and that CFG can easily be generalized for any alphabet.)
2. Construct a CFG H for the language $\mathrm{L}(G) \cap \mathrm{L}(D)$ using the construction discussed in lecture (when we proved that the intersection of a context-free language and a regular language is context-free).
2. *Accept* if $\langle H \rangle \in \mathrm{E}_{\mathrm{CFG}}$, and *reject* otherwise.

It holds that $\mathrm{L}(H)$ is empty if and only if every string accepted by D is a palindrome, and therefore M decides the given language.

2. Let Σ be an alphabet, and suppose that $A \subseteq \Sigma^*$ is Turing recognizable. Prove that these languages are also Turing recognizable:

$$\begin{aligned}
\text{Prefix}(A) &= \{x \in \Sigma^* \,:\, \text{there exists } v \in \Sigma^* \text{ such that } xv \in A\}, \\
\text{Suffix}(A) &= \{x \in \Sigma^* \,:\, \text{there exists } u \in \Sigma^* \text{ such that } ux \in A\}, \\
\text{Substring}(A) &= \{x \in \Sigma^* \,:\, \text{there exists } u, v \in \Sigma^* \text{ such that } uxv \in A\}.
\end{aligned}$$

Solution. All three languages can be recognized by DTMs using the same basic idea. For all three, we will assume that M_A is a DTM such that $\mathrm{L}(M_A) = A$. Here is a DTM that recognizes $\text{Prefix}(A)$:

On input x:

1. Set $t \leftarrow 1$.
2. For every string $v \in \Sigma^*$ satisfying $|v| \leq t$, run M_A for t steps on input xv.
   If M_A accepts xv within t steps, then *accept*.
3. Set $t \leftarrow t + 1$ and goto 2.

Here is a DTM that recognizes $\text{Suffix}(A)$:

On input x:

1. Set $t \leftarrow 1$.
2. For every string $u \in \Sigma^*$ satisfying $|u| \leq t$, run M_A for t steps on input ux.
   If M_A accepts ux within t steps, then *accept*.
3. Set $t \leftarrow t + 1$ and goto 2.

And finally, here is a DTM that recognizes $\text{Substring}(A)$:

On input x:

1. Set $t \leftarrow 1$.
2. For every pair of strings $u, v \in \Sigma^*$ satisfying $|u| \leq t$ and $|v| \leq t$, run M_A for t steps on input uxv. If M_A accepts uxv within t steps, then *accept*.
3. Set $t \leftarrow t + 1$ and goto 2.

3. Let Σ be an alphabet, and suppose that $A, B \subseteq \Sigma^*$ are Turing-recognizable languages for which both $A \cap B$ and $A \cup B$ are decidable. Prove that A is decidable.

Solution. Given that A and B are Turing-recognizable, there exist DTMs M_A and M_B such that $\mathrm{L}(M_A) = A$ and $\mathrm{L}(M_B) = B$. Define a DTM M as follows:

On input $x \in \Sigma^*$:

1. If $x \in A \cap B$, then *accept*.
2. If $x \notin A \cup B$, then *reject*.
3. Set $t \leftarrow 1$.
4. Run M_A for t steps on input x: if M_A accepts x within t steps, then *accept*.
5. Run M_B for t steps on input x: if M_B accepts x within t steps, then *reject*.
6. Set $t \leftarrow t + 1$ and goto 4.

Steps 1 and 2 can be performed by M under the assumption that $A \cap B$ and $A \cup B$ are decidable. If the DTM reaches step 3, then it must be the case that x is contained in the symmetric difference $A \triangle B$, so we are guaranteed that exactly one of the DTMs M_A or M_B will eventually accept x. When we reach a large enough value of t for this to happen, we are then able to correctly decide whether or not x is in A. Because M decides A, we have that A is decidable.

---

4. Suppose that we have chosen an encoding scheme whereby each DTM M is encoded as a string $\langle M \rangle$ over some alphabet Σ. Define two languages over Σ as follows:

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

For the sake of this problem, you should assume that the statements "M accepts the input $\langle M \rangle$" and "M rejects the input $\langle M \rangle$" are both false if it so happens that there are symbols in the string $\langle M \rangle$ that are not contained in the input alphabet of M.

Prove that there does not exist a decidable language $C \subseteq \Sigma^*$ for which both of the containments $A \subseteq C$ and $B \subseteq \overline{C}$ hold.

Hint: it is possible to prove the required fact directly through the use of diagonalization. You are, however, free to prove the fact through whatever method you choose.

Solution. Assume toward contradiction that there exists a decidable language C such that $A \subseteq C$ and $B \subseteq \overline{C}$. By the assumption that C is decidable, there exists a DTM M that decides C. For this DTM we have

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

and

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

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 such that $A \subseteq C$ and $B \subseteq \overline{C}$.