Solutions for Problem Set 5
cs172

Allison Coates

4.4, 4.5, 4.18, 4.21

4.4 Let $A\epsilon_{\text{CFG}} = \{< G > \mid$ G is a CFG that generates ϵ . Show that $A\epsilon_{CFG}$ is decidable.

We need to ensure that we test all derivations, but we also need the derivations not to be infinite, or to loop forever. Using the Chomsky Normal Form we can convert the given grammar into a form where only a finite number of steps are needed to determine whether G generates the empty string. In Chomsky Normal Form, if a grammar excepts the empty string, it does so from the first/start rule. So, we can build a Turing machine M for $A\epsilon_{\text{CFG}}$ as follows:

$M =$ " On input $< G >$ where G is a CFG,

1. Convert G to an equivalent grammar using Chomsky Normal Form.
2. Examine the start rule of this grammar. If it generates ϵ, then accept. Else, reject.

4.5 Let $INFINITE_{\text{DFA}} = \{< A > |L(A)$ is an infinite language$\}$. Show that $INFINITE_{\text{DFA}}$ is decidable.

Given A, a DFA, A accepts an infinite language if it accepts a string of length $> n$ and $< 2n$, where n is the number of finite states in A.

We can build a Turing machine T that decides $INFINITE_{\text{DFA}}$ as follows:

$S =$ " On input A,

1. Generate all possible strings W over A's alphabet of lengths $n + 1$ to $2n$.
2. For all $w \in W$ above, simulate A on w.
3. If any such string is accepted by A, accept. Else, reject.

4.18 Let A and B be two disjoint languages. Say that language C **separates** A and B if $A \subseteq C$ and $B \subseteq \bar{C}$. Show that any two disjoint co-Turing-recognizable languages are separable by some decidable languages.

That is, if A and B are two disjoint co-Turing-recognizable languages, then A and B are separated by a decidable language C.

A is co-r.e. Therefore, there is an enumerator for $\bar{A}$. Call this enumerator $E_{\bar{A}}$. B is co-r.e. Therefore, there is an enumerator for $\bar{B}$. Call this enumerator $E_{\bar{B}}$. A and B are disjoint; i.e. $A \cap B = \emptyset$. This means that $\bar{A} \cup \bar{B} = \Sigma^*$. So, every string is in the union of $\bar{A}$ and $\bar{B}$. Further, every string in B is in $\bar{A}$. Every string in A is in $\bar{B}$.

Using these facts, we can construct a language D by constructing a Turing machine $M, L(M) = D$ as follows:

$M =$" on input w,

1. Run both $E_{\bar{B}}$ and $E_{\bar{A}}$ in parallel.
2. Alternating between the enumerators, and beginning with $E_{\bar{B}}$, compare the outputs of each of the enumerators, one string at a time against w.
3. If some output of $E_{\bar{B}}$ matches w, accept. If some output of $E_{\bar{A}}$ matches w, reject.

Since $\bar{A} \cup \bar{B} = \Sigma^*$, every string is enumerated by $E_{\bar{A}}$ or $E_{\bar{B}}$ (or both). Hence, M will halt on all inputs. Hence, M is a decider for some language C. Since every string in A is in $\bar{B}$, the output of $E_{\bar{B}}$ contains all strings of A. M accepts all strings that are output by only $E_{\bar{B}}$. Hence, M accepts all strings of A. Likewise, since every string of B is in $\bar{A}$, the output of $E_{\bar{A}}$ contains all strings of B. M rejects all strings that are output by only $E_{\bar{A}}$. Hence, M rejects all strings in B. We have now shown that M halts on all inputs, that it accepts all strings in A and that it rejects all strings in B. Hence, M is a decider for C, and C separates A and B.

Note that we did not prove which set of strings C accepted.The particular language of C in fact depends on the order of the outputs of the enumerators. But, the only strings in questions are the strings that are in $\bar{A} \cap \bar{B}$. Whether these strings are in C or in $\bar{C}$ is not relevant to the question of separating A and B.

4.21 Let A be a Turing-recognizable language consisting of descriptions of Turing machines, $\{< M_1 >, < M_2 >, \ldots\}$, where every M_i is a decider. Prove that some decidable language D is not decided by any decider M_i whose description appears in A.

Proof by construction. I will construct a decidable language D that is not decided by any M_i whose description is in A. I begin by noting that all strings in Σ^* can be enumerated in lexicographic-length order. (There are a countable number of strings in Σ^*.) I can associate with each string w an index i, which is the position of w in the enumerated list. I then wish to construct a Turing machine whose language differs from each language $L(M_i)$ on (at least) one string. In that case, this Turing machine will need to accept when M_i rejects, and reject when M_i accepts. Further, I wish to construct this machine to halt on all inputs. I will also use an enumerator E_A for A, which prints descriptions of Turing deciders. I construct a Turing machine M^*, where $L(M^*) = D$ as follows:

$M^* =$“ on input w,

1. Enumerate enough strings in lexicographic order to determine the index of w. Call this index i.
2. Run the enumerator E_A to retrieve the description of M_i, the ith Turing decider output by E_A. If there is no ith decider, reject and halt.
3. Simulate M_i on w.
4. If M_i accepts, reject. If M_i rejects, accept.

First, we need to show that M^* halts on all inputs. First, if the enumerator does not have an ith enumerator, then the enumerator halts, and so does M^*. If the enumerator does not halt, then there is some Turing decider M_i. Since M_i is a decider, it halts on all inputs. M^* halts every time M_i halts; hence, M^* halts on all inputs. (Note that it does not matter whether the enumerator is repeating Turing machines, or what order it prints out the descriptions. The enumerator is still printing out the descriptions in a deterministic order, so M_i will always be M_i, even if it is also M_j, $i \neq j$.)

Now we show that M^* decides a language that is not decided by any M_i in A. For all i, there is at least one string on which M^* disagrees with the outcome of M_i. Therefore, the language $L(M^*)$ contains one string not in M_i, or does not contain a string in M_i, for each M_i. Therefore, M^* decides a language that is not decided by any M_i in A. Therefore, there exists a decidable language D that is not decided by any decider M_i whose description appears in A.