Introduction to the Theory of Computation Solutions

Ryan Dougherty

Contents

Chapter 1

Chapter 2

Chapter 3

1 Chapter 4

4.2

Consider the problem of determining whether a DFA and a regular expression are equivalent. Express this problem as a language and show that it is decidable.
Solution: We formulate the problem $EQ_{DFA,REX} = \{\langle A, R\rangle \mid A$ is a DFA, R is a regular expression, and $L(A) = L(R)\}$. We will design a TM T that decides $EQ_{DFA,REX}$:
$T =$ "On input $\langle A, R\rangle$ where A is a DFA, R is a regular expression:

1. Use Theorem 1.54 to convert R into an equivalent DFA B. Therefore, $L(B) = L(R)$.
2. Run EQ_{DFA} on input $\langle A, B\rangle$. Output what EQ_{DFA} outputs."

Since EQ_{DFA} is decidable, and the conversion from regular expressions to DFAs takes finite time, $EQ_{DFA,REX}$ is decidable.

4.3

Let $ALL_{DFA} = \{\langle A\rangle \mid A$ is a DFA and $L(A) = \Sigma^*\}$. Show that ALL_{DFA} is decidable.
Solution: We will design a TM T that decides ALL_{DFA}:
$T =$ "On input ⟨A⟩ where A is a DFA:

1. Construct a DFA B such that $L(A) = \overline{L(B)}$.
2. Run E_{DFA} on input ⟨B⟩. Output what E_{DFA} outputs."

Since E_{DFA} is decidable, ALL_{DFA} is decidable.

4.4

Let $A\epsilon_{CFG} = \{\langle G\rangle \mid G$ is a CFG that generates ϵ.$\}$. Show that $A\epsilon_{CFG}$ is decidable.
Solution: We will design a TM T that decides $A\epsilon_{CFG}$:
$T =$ "On input ⟨G⟩ where G is a CFG:

1. Convert G into an equivalent CFG $C = (V, \Sigma, R, S)$ in Chomsky Normal Form.
2. *Accept* if C includes the rule $S \to \epsilon$, reject otherwise."

Since converting a CFG into CNF is decidable, $A\epsilon_{CFG}$ is decidable.

4.11

Let $INFINITE_{PDA} = \{\langle M\rangle \mid M$ is a PDA and $L(M)$ is an infinite language$\}$. Show that $INFINITE_{PDA}$ is decidable.
Solution: We will design a TM T that decides $INFINITE_{PDA}$:
$T =$ "On input $\langle M\rangle$ where M is a PDA:

1. Construct an equivalent CFG G from M.
2. Convert G into an equivalent CFG $C = (V, \Sigma, R, S)$ in Chomsky Normal Form.
3. *Accept* $\langle M\rangle$ if there exists a derivation $A \overset{+}{\Rightarrow} uAv$ for some $u, v \in \Sigma^*$. Otherwise, *reject* $\langle M\rangle$.

Since all of the algorithms in this machine are decidable, $INFINITE_{PDA}$ is decidable.

4.13

Let $A = \{\langle R, S\rangle \mid R$ and S are regular expressions and $L(R) \subseteq L(S)\}$. Show that A is decidable.
Solution: We will design a TM T that decides A:
$T =$ "On input $\langle R, S\rangle$ where R and S are regular expressions:

1. Construct a DFA B such that $L(B) = \overline{L(S)} \cap L(R)$.
2. Run E_{DFA} on input ⟨B⟩. Output what E_{DFA} outputs."

Since E_{DFA} is decidable, A is decidable. This construction is correct because $L(R) \subseteq L(S) \Leftrightarrow \overline{L(S)} \cap L(R) = \varnothing$.

4.15

Show that the problem of determining whether a CFG generates all strings in 1* is decidable. In other words, show that $\{\langle G\rangle \mid G$ is a CFG over $\{0, 1\}$ and $1^* \subset L(G)\}$ is a decidable language.
Solution: Let f be a computable function. Construct a decider D:
$D =$ "On input ⟨G⟩ where G is a CFG:

1. Convert G into an equivalent CFG C in Chomsky Normal Form.
2. Let p be the pumping length of C.
3. Repeat $\forall i \leq f(p)$:
   a. Check whether $1^i \in L(C)$.
   b. If not, *reject* ⟨G⟩.
4. *Accept* ⟨G⟩.

We can check $1^i \in L(C)$ using the Cocke-Younger-Kasami (CYK) algorithm for CFGs, which has a running time of $\Theta(n^3|G|)$. Therefore, the loop has a running time of $\Theta(pn^3|G|)$. Since Steps 1, 2, 3, and 4 take finite time, this language is decidable.

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9