Problem 4.3

$$\text{Let } ALL_{DFA} = \{\langle A \rangle \mid \text{A is a DFA and L(A)} = \Sigma^*\}$$

Want to show that ALL_{DFA} is decidable.

Note: A language L is Turing Decidable (or just decidable in short) if there exists a halting TM M (decider) such that $L = L(M)$. L is decidable if there exists a decider that decides the language.

⇒ Want to construct a TM M that decides ALL$_{DFA}$

Note: If D is a DFA and $L(D) = \Sigma^*$ then the complement of the language D can be obtained by converting all the accept states in D to reject states, and all the reject states to accept states. (Not true for NFAs)

Note: The complement of a language is defined in terms of set difference from Σ^*:

$$\begin{aligned} \overline{L} &= \Sigma^* - L \\ \Rightarrow \overline{L(D)} &= \Sigma^* - L(D) \\ &= \Sigma^* - \Sigma^* \\ &= \emptyset \end{aligned}$$

Given a DFA A and $L(A) = \Sigma^*$ we can construct the complement of the DFA A called DFA B with $L(B) = \emptyset$. Then we can use TM T from Theorem 4.4 on input $\langle B \rangle$ to check if L(B)=$\emptyset$. If T accepts, M accepts. If T rejects, M rejects.

The following TM M decides ALL_{DFA}:

M = "On input $\langle A \rangle$ where A is a DFA and L(A)=Σ^*

1. Let B be the DFA obtained by swapping accept and reject states of A.
2. Run TM T from Theorem 4.4 on input $\langle B \rangle$ to see if L(B)=$\emptyset$.
3. If so accept, else reject."

⇒ALL$_{DFA}$ is decidable

Problem 4.4

$$\text{Let } ALL_{\epsilon CFG} = \{\langle G\rangle \,|\text{G is a CFG that generates } \epsilon\}$$
$$= \{\langle G\rangle \,|\text{G is a CFG and } \epsilon \in \text{L(G)}\}$$

Want to show that $ALL_{\epsilon CFG}$ is decidable.
$\Rightarrow$ Construct a halting TM M that decides $\text{ALL}_{\epsilon CFG}$

Note: Given the CFG G, we can convert it into an equivalent CFG $G' = (V, \Sigma, R, S)$ in Chomsky Normal Form. By definition since G' is in Chomsky Normal Form, we know that the only possible ϵ rule in G' will be $S \to \epsilon$ where S is the start variable.

If $S \to \epsilon$ is a rule in R $\Rightarrow \epsilon \in L(G')$
Else $\epsilon \notin L(G')$

The following TM M decides $ALL_{\epsilon CFG}$:

M = "On input $\langle G\rangle$ where G is a CFG:
1. Convert G into an equivalent CFG G' in Chomsky Normal Form.
2. If G' includes the rule S $\to \epsilon$, accept. Else reject."

$\Rightarrow$$\text{ALL}_{\epsilon CFG}$ is decidable

Problem 4.8

$$\text{Let } T = \{(i,j,k) | i,j,k \in \mathbb{N}\}$$

Note: A set A is countable if it is finite or has the same size as $\mathbb{N}$.
Want to show that the set T is countable

For each triple (i,j,k) where $i,j,k \in \mathbb{N}$, let $i+j+k$ be the sum S.
We can enumerate all triples in T since there are only a finite number of triples whose sum equal to S such that $S = 0, 1, 2, ...$

$\Rightarrow T = \{(i,j,k) | i,j,k \in \mathbb{N}\}$ is countable.

Problem 4.11

Let $INFINITE_{PDA} = \{\langle M \rangle \,|$M is a PDA and L(M) is an infinite language$\}$

Want to show that $INFINITE_{PDA}$ is decidable.
$\Rightarrow$ Construct a halting TM M that decides INFINITE$_{PDA}$

Note: Given the PDA M, we can convert it into an equivalent CFG.
We can then convert the CFG into an equivalent CFG G' in Chomsky Normal Form.
We can check CFG G' if there exists a derivation $A \Rightarrow uAv$ where $u, v \in \Sigma^*$

If $A \Rightarrow uAv$ is a derivation in CFG G' then L(M) is an infinite language.
If $A \Rightarrow uAv$ is not a derivation in CFG G' then L(M) is not an infinite language.

The following TM M decides $INFINITE_{PDA}$:

M = "On input $\langle M \rangle$ where M is a PDA:
1. Convert M into an equivalent CFG G
2. Convert G into an equivalent CFG G' in Chomsky Normal Form.
3. If G' includes the derivation A $\Rightarrow uAv$, accept. Else, reject."

$\Rightarrow$INFINITE$_{PDA}$ is decidable

Problem 4.13

Construct a Turing machine T to show that A is decidable. First convert the regular expression R and S into DFA D_R and D_S. Then construct DFA D where $L(D) = \overline{L(D_S)} \cap \overline{L(D_R)}$. Run T on input $\langle D \rangle$. If DFA D accepts, T accepts. Else, reject.

Problem 4.21

Construct a Turing machine T to show that S is decidable. Let M^R be the DFA that accepts the reverse of strings that are accepted by M. Then, $L(M^R) = L(M)$. Now the Turing machine T constructs a DFA D on input $\langle M \rangle$ that accepts the reverse of strings accepted by M. T sends $\langle M, D \rangle$ as input to the decider.
If DFA D accepts then T also accepts. Else, reject. T accepts M iff $L(M^R) = L(M)$.
Since T decides S, S is decidable.

Problem 4.24

Let P be the set of all pushdown automata. Let $U = \{x | x \in P \text{ where x has a useless state}\}$
To show that U is decidable we construct a Turning machine T that accepts strings that only belong to U.
We use the fact whether a PDA having an empty language is decidable. We can reduce the problem (whether given state q is useless) by making q the only accepting state. The Turing machine can solve this question by performing this for each state.

Problem 4.29

Let $C_{CFG} = \{\langle G, k\rangle$ |G is a CFG and L(G) contains exactly k strings where k $\geq$ 0 or k = $\infty\}$

Consider a decider M that decides the following language:

$INFINITE_{CFG} = \{\langle G\rangle$ | G is a CFG and L(G) is an infinite language}

Want to show that C_{CFG} is decidable.
$\Rightarrow$ Construct a decider T that decides C_{CFG}.

The following TM T decides C_{CFG}:

T = "On input $\langle G, k\rangle$ where G is a CFG and k is string:
1. Check if L(G) is infinite using decider M.
   If L(G) is infinite and k = ∞, accept.
   If L(G) is finite and k $\neq$ ∞, reject.
   If L(G) is finite and k = ∞, reject.
   If L(G) is finite and k $\neq$ ∞, continue.
2. Calculate the pumping length p for CFG G.
3. Let Count = 0
4. For $i = 1, 2, ..., p$:
   For all strings s with length i.
   If $s \in L(G), Count++$.
5. If $Count = k$, accept. If $Count \neq k$, reject."

First we check if the language of CFG G is infinite using decider T. If L(G) is finite and $k \neq \infty$, then we need to check if L(G) contains exactly k strings. Since we know that L(G) is finite we know that none of the strings in L(G) may have length greater than the pumping length p for grammar G.

$\Rightarrow$C$_{CFG}$ is decidable