Theory of Computation, Homework 4

5.1
Assume $E_{C\text{FG}}$ is decidable and there exists a decider $D_{E_{C\text{FG}}}$ deciding it. That is, constructing a TM $M_{\text{ALL} \cdot \text{CFG}}$ to decide $\text{ALL}_{C\text{FG}}$ is possible as follows:

$M_{\text{ALL} \cdot \text{CFG}}$ = "On input CFG $(G)$

1. derive a CFG $G_{\text{ALL}}$ generating all possible strings
2. run $D_{E_{C\text{FG}}}$ to decide whether $(G, \text{ALL})$ and $(G)$ are equivalent
3. accept if it accepts; reject, otherwise."

However, $\text{ALL}_{C\text{FG}}$ is undecidable, which contradicts the assumption that $E_{C\text{FG}}$ is decidable.

5.2
Since $A_{C\text{FG}}$ is decidable, $M_{\text{A} \cdot \text{CFG}}$ can be used to test if a string $w \in G$, where $G$ is a CFL. A Turing-recognizer $M_{\text{to-\text{EQ}_{C\text{FG}}}}$ for the complement of $E_{C\text{FG}}$ can be constructed in the following to show $\neg \text{EQ}_{C\text{FG}}$ is Turing-recognizable.

$M_{\text{to-\text{EQ}_{C\text{FG}}}}$ = "On input CFGs $(G_1, G_2)$

1. enumerate strings with lengths in an ascending order
2. for each string $w$ enumerated
   a. run $M_{\text{A} \cdot \text{CFG}}$ to decide if $w \in L(G_1)$ and if $w \in L(G_2)$
   b. accept if either $w \in L(G_1)$ and $w \notin L(G_2)$, or $w \notin L(G_1)$ and $w \in L(G_2)$
   c. continue otherwise"

5.3
$\begin{bmatrix} a & b \\ a & a \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} b \\ a \end{bmatrix}$

5.9
Assume $T$ is decidable. Then there exists a decider for $T$ which can be used to decide $\text{A} \cdot \text{T} \cdot M$.

The following TM $M_T$ is constructed to be an instance of $T$.

$M_T$ = "given $(M_{\text{A} \cdot \text{T}}, w)$, on input $x$

1. accept if $x \neq 0$
2. reject if $x \neq 0$
3. accept if $M_{\text{A} \cdot \text{T}}$ accepts $w$. Otherwise, reject.

Obviously, when $L(M_T) = \{0, 0\}$, $(M_T, x) \in T$ if $M_{\text{A} \cdot \text{T}}$ accepts $w$. That is, by using the decider for $T$, $(M_T)$ can be decided such that $(M_{\text{A} \cdot \text{T}}, w)$ can also be decided. Therefore, $\text{A} \cdot \text{T} \cdot M$ is decidable. However, $\text{A} \cdot \text{T} \cdot M$ is not decidable, contradicting the assumption $T$ is decidable.
5.12
Let $B = \{M_{\text{blank}} \mid M_{\text{blank}}$ is a single-tape TM which writes a blank symbol over a nonblank symbol during the course of its computation on any input string$\}$ be a language. Assume there is a decider $D_B$ deciding $B$. The goal is to reduce $\overline{A_{TM}}$ to $B$. Therefore, we construct a TM $M_S$ which decides $\overline{A_{TM}}$ using $D_B$.

$$M_S = \begin{cases} 
\text{on input } (M, \omega) \\
1. \text{ construct TM } M_B \text{ by modifying } M \\
a. \text{ for each transition which writes a blank symbol, replace the blank symbol with a new nonblank symbol not in } \Sigma \\
b. \text{ for each transition which reads a blank symbol, replace the blank symbol with the new nonblank symbol that is previously written} \\
c. \text{ for each transition to the accept state, write a nonblank symbol, overwrite with a blank symbol and then go to the accept state} \\
2. \text{ run } D_B \text{ to decide } M_B \\
3. \text{ accept if } D_B \text{ accepts; otherwise, reject} 
\end{cases}$$

Obviously, $\overline{A_{TM}}$ can be decided by TM $M_S$, leading to a contradiction because $\overline{A_{TM}}$ is undecidable. Hence, $B$ is undecidable.

5.14
Let $L = \{ (M, \omega) \mid M$ is a TM which ever attempts to move its head left when its head is on the left-most tape cell on input $\omega \}$ be a language. Assume there is a decider $D_L$ deciding $L$. The goal is to reduce $\overline{A_{TM}}$ to $L$. Therefore, we construct a TM $M_S$ which decides $\overline{A_{TM}}$ using $D_L$.

$$M_S = \begin{cases} 
\text{on input } (M, \omega) \\
1. \text{ construct TM } M_L \text{ by modifying } M \\
a. \text{ shift the input on the tape one cell right} \\
b. \text{ write the cell at the left-hand end (using a mark to avoid moving left from the leftmost cell) with a new symbol, say '@,' not in } \Sigma \\
c. \text{ add a transition for the head to move one cell right on reading '@'} \\
d. \text{ when the accept state is reached, move the head left off the leftmost cell} \\
2. \text{ run } D_L \text{ to decide } M_L \\
3. \text{ accept if } D_L \text{ accepts; otherwise, reject} 
\end{cases}$$

Obviously, $\overline{A_{TM}}$ can be decided by TM $M_S$, leading to a contradiction because $\overline{A_{TM}}$ is undecidable. Hence, $L$ is undecidable.
The goal is to reduce $PCP$ to $AMBIG_{CFG}$. Given an instance of $PCP$ as follows

$$P = \left\{ \left[ \begin{array}{c} t_1 \\ l_1 \\ b_1 \\ l_2 \\ b_2 \\ l_k \\ b_k \end{array} \right] \right\},$$

a corresponding instance $G$ of $AMBIG_{CFG}$ can be constructed in the following.

$$S \rightarrow T \mid B \mid$$

$$T \rightarrow t_1^3T_1a_1 \mid t_2^3T_2a_2 \mid \cdots \mid t_k^3T_ka_k \mid b_1^3a_1 \mid b_2^3a_2 \mid \cdots \mid b_k^3a_k \mid$$

$$B \rightarrow b_1^3T_1a_1 \mid b_2^3T_2a_2 \mid \cdots \mid b_k^3T_ka_k \mid b_1^3a_1 \mid b_2^3a_2 \mid \cdots \mid b_k^3a_k \mid$$

If $P$ has a match $i_1, i_2, \ldots, i_j = b_1^3b_2^3b_3^3$ with indices $i_1, i_2, \ldots, i_j$, then a string $b_1^3b_2^3b_3^3a_i^3a_j^3 \vdash a_i^3a_j^3$ produced by $G$ can be derived from both $T$ and $B$. That is, $G$ is an ambiguous CFG.

On the other hand, if $G$ is ambiguous, all the generated strings of the form $wa_i^3a_j^3 \vdash a_i^3a_j^3$ where $w$ is a string containing symbols in $\{t_1^3, t_2^3, \ldots, t_k^3, b_1^3, b_2^3, \ldots, b_k^3\}$ must have at most two derivations as follows.

One is $S \Rightarrow T \Rightarrow \ast \Rightarrow t_1^3b_1^3t_2^3b_2^3t_3^3b_3^3a_i^3a_j^3 \vdash a_i^3a_j^3 = b_1^3b_2^3b_3^3a_i^3a_j^3 \vdash a_i^3a_j^3$.

The other is $S \Rightarrow B \Rightarrow \ast \Rightarrow t_1^3b_1^3t_2^3b_2^3t_3^3b_3^3a_i^3a_j^3 \vdash a_i^3a_j^3 = b_1^3b_2^3b_3^3a_i^3a_j^3 \vdash a_i^3a_j^3$.

As a result, $t_1^3b_1^3t_2^3b_2^3t_3^3b_3^3a_i^3a_j^3 \vdash a_i^3a_j^3$ implies $t_1^3b_1^3t_2^3b_2^3t_3^3b_3^3a_i^3a_j^3 \vdash a_i^3a_j^3$.

Therefore, if the instance $G$ of $AMBIG_{CFG}$ can be decided, then the instance $P$ of $PCP$ can also be decided. However, it’s impossible since $PCP$ is undecidable.

5.22

(→) If $A \leq_{TM} A_{TM}$, then by the Theorem 5.39, $A$ is also Turing-recognizable because $A_{TM}$ is Turing-recognizable.

(←) If $A$ is Turing-recognizable, there exists a TM $M_A$ recognizing $A$. That is, $A = \{w \mid w \text{ is accepted by } M_A\}$. Now we define function $f(M_A, w) = (M_A, w)$ to be the reduction of $A$ to $A_{TM}$. Obviously, for every input $w, \forall w \in A \Leftrightarrow f(w) \in A_{TM}$. Hence, $A \leq_{TM} A_{TM}$.
5.23

(←) If $A \leq_m 0^*1^*$, then there exists a computable function $f : \Sigma^* \to \Sigma^*$, where $w \in A \iff f(w) \in 0^*1^*$. Since $0^*1^*$ is regular, there is a decider $R$ for $0^*1^*$. Obviously, $A$ can be decided by running $R$ on $f(w)$, where $w$ is an input string in $A$.

(→) If $A$ is decidable, there is a decider $R_A$ that decides $A$. Now we define function as $f(R_A, w) = 01$ if $R_A$ accepts $w$, or 10 otherwise. Obviously, $f$ is the reduction of $A$ to $0^*1^*$ since for every input $(R_A, w), (R_A', w) \in A \iff f(R_A, w) \in 0^*1^*$. Hence, $A \leq_m 0^*1^*$.

5.30(b)

Rice’s theorem holds when $P$ is a nontrivial property of the language of a Turing machine. In this problem, $P$ is the property that $1011 \in L(M)$. Besides, $P$ has to fulfill the following two conditions.

1. $P$ is nontrivial - it contains some, but not all, TM descriptions. Let $L(M_1) = \{0,1\}^*$ and $L(M_2) = \emptyset$. Obviously, $P$ contains $M_1$ but not $M_2$.

2. $P$ is a property of the TM's language - whenever $L(M_1) = L(M_2)$, we have $M_1 \in P$ iff $M_2 \in P$. Here, $M_1$ and $M_2$ are any TMs. By definition, $P$ is a property of the language of TM $M$, $\{(M) \mid M \text{ is a TM and } 1011 \in L(M)\}$.

Since $P$ satisfies the two conditions above, $P$ is a nontrivial property of $L(M)$. Hence, by Rice’s theorem, the language $\{(M) \mid M \text{ is a TM and } 1011 \in L(M)\}$ is undecidable.