Question:

Prove that the following two languages are undecidable.

a. $OVERLAP_{\mathsf{CFG}} = \{\langle G, H\rangle |\ G \text{ and } H \text{ are CFGs where } L(G) \cap L(H) \neq \emptyset\}$.
(Hint: Adapt the hint in Problem 5.21.)

b. $PREFIX\text{-}FREE_{\mathsf{CFG}} = \{\langle G\rangle |\ G \text{ is a CFG where } L(G) \text{ is prefix-free}\}$.

Answer:

----SETP1----

The given languages have to be proven to be un-decidable.

a)

$OVERLAP_{CFG} = \{\langle G, H\rangle \mid G \text{ and } H \text{ are CFGs where } L(G) \cap L(H) \neq \phi\}$

----SETP2----

Assume that $OVERLAP_{CFG}$ is decidable. Given an instance for the problem of Post Correspondence $P = \left\{\left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \ldots, \left[\frac{t_n}{b_n}\right]\right\}$, introduce unique new terminals $a_1, a_2, \ldots, a_n$ for the CFGs.

----SETP3----

• Define the CFG G as:

$G = t_1, t_2, \ldots, t_n$
$L(G) = \{s \mid s = t_i t_j \ldots t_k a_k \ldots a_j a_i\}$
$S_G \rightarrow t_1 S_G a_1 \mid \ldots \mid t_n S_G a_n \mid t_1 a_1 \mid \ldots \mid t_n a_n$

• Similarly, define the CFG H as follows:

$H = b_1, b_2, \ldots, b_n$
$L(H) = \{s \mid s = b_i b_j \ldots b_k a_k \ldots a_j a_i\}$
$S_H \rightarrow b_1 S_H a_1 \mid \ldots \mid b_n S_H a_n \mid b_1 a_1 \mid \ldots \mid b_n a_n$

As $L(G) \cap L(H) \neq \phi$, we get $t_i t_j \ldots t_k a_k \ldots a_j a_i = b_i b_j \ldots b_k a_k \ldots a_j a_i$.

----SETP4----

• Since the new terminals $a_1, a_2, \ldots, a_n$ are unique, which can be cancelled from both sides resulting in:

$t_i t_j \ldots t_k = b_i b_j \ldots b_k$.

This is a way to solve for **the Post Correspondence Problem** P**.** This is a contradiction as the Post Correspondence **Problem is un-decidable**. Therefore, the assumption taken that $OVERLAP_{CFG}$ **is decidable, is incorrect**.

----SETP5----

b)

$PREFIX\text{-}FREE_{CFG} = \{\langle G\rangle \mid G \text{ is a CFG where } L(G) \text{ is prefix-free}\}$

A mapping reducibility from the language A_{TM} (does a Turing machine accept a string?) to the language $PREFIX\text{-}FREE_{CFG}$ is given by the function f. Here, $\langle M, w\rangle$ is taken as an input of the computable functions f and $\langle M', w'\rangle$ is returned in such a way that:

$\langle M, w\rangle \in A_{TM}$ iff $\langle M', w'\rangle \in PREFIX\text{-}FREE_{CFG}$

----SETP6----

The machine F to compute the function f is:

F ="when $\langle M, w \rangle$ is taken as an input:

1. The machine M' is constructed

M' ="On input x:

1. For all proper prefixes y of x:

1. M will be run on y.

2. then *reject*.

2. Run M on x.

3. If it is accepted by M, then *accept*.

4. If it is rejected by M,, then *reject*."

2. Output $\langle M', w' \rangle$.

The output machine M' only accepts an input string w if the language $L(M)$ is prefix-free. It does so by checking if any of the proper prefixes of w do not lie in $L(M)$ and w lies in it.

----SETP7----

• It has been shown that A_{TM} **is mapping reducible to** $PREFIX\text{-FREE}_{CFG}$, that is:

$A_{TM} \leq_m PREFIX\text{-}FREE_{CFG}$

Thus as A_{TM} is un-decidable, the language $PREFIX\text{-FREE}_{CFG}$ is also un-decidable.