Guidelines: Solve all problems in the class. Do not search for solutions online.

1. Consider a silly variant of PCP called SPCP where corresponding strings in both lists are restricted to have the same length. Show that this variant is decidable.

   Solution: Let $A = \{w_1, \ldots, w_n\}$ and $B = \{x_1, \ldots, x_n\}$ denote an instance of SPCP. If there exists a solution, then there is an index $j \in [1,n]$ such that the solution starts with $w_j, x_j$. Let $|w_j| = |x_j| = \ell$. Since there is a match in the first $\ell$ positions, it must be the case that $w_j = x_j$. Also, if there is an index $j$ such that $w_j = x_j$, then $j$ is a solution to the SPCP instance $(A, B)$. Therefore, SPCP can be decided by just checking whether for each $j \in [1,n]$, $w_j = x_j$.

2. Prove that $\{G_1, G_2 \mid G_1, G_2 \text{ are CFGs and } L(G_1) \cap L(G_2) \neq \emptyset\}$ is undecidable via a reduction from PCP.

   Solution: Let $A = \{w_1, \ldots, w_k\}$ and $B = \{x_1, \ldots, x_k\}$ denote an instance of PCP over alphabet $\Sigma$. Let $\Sigma' = \Sigma \cup \{a_1, \ldots, a_k\}$ for some new symbols $a_1, \ldots, a_k \notin \Sigma$. Define two CFGs $G_A = (\{S_A\}, \Sigma', P_A, S_A)$ and $G_B = (\{S_B\}, \Sigma', P_B, S_B)$ where $P_A$ consists of the productions
   
   $$S_A \rightarrow w_i S_A a_i \mid w_i a_i \text{ for } 1 \leq i \leq k,$$

   and $P_B$ consists of
   
   $$S_B \rightarrow x_i S_B a_i \mid x_i a_i \text{ for } 1 \leq i \leq k.$$

   Suppose the PCP instance $(A, B)$ has a solution $i_1, \ldots, i_m$. Let $y = w_{i_1} \cdots w_{i_m} = x_{i_1} \cdots x_{i_m}$. Then $ya_{i_1} \cdots a_{i_m} \in L(G_A)$ and $ya_{i_1} \cdots a_{i_m} \in L(G_B)$. As a result $L(G_A) \cap L(G_B) \neq \emptyset$.

   Now, suppose that $L(G_A) \cap L(G_B) \neq \emptyset$. Then there is a string $ya_{i_1} \cdots a_{i_m} \in L(G_A) \cap L(G_B)$. Since $ya_{i_1} \cdots a_{i_m} \in L(G_A)$ it must be the case that $y = w_{i_1} w_{i_2} \cdots w_{i_m}$. Also, $y$ must be equal to $x_{i_1} x_{i_2} \cdots x_{i_m}$ since $ya_{i_1} \cdots a_{i_m} \in L(G_B)$. Then $i_1, \ldots, i_m$ forms a solution to PCP instance $(A, B)$.

3. Show that PCP is undecidable over the binary alphabet $\{0,1\}$.

   Solution: Denote PCP over alphabet $\{0,1\}$ as BPCP. We show that PCP $\leq_m$ BPCP. Let $A = \{w_1, \ldots, w_n\}$ and $B = \{x_1, \ldots, x_n\}$ denote an instance of PCP over some alphabet $\Sigma$. Let $s = |\Sigma|$ and $\Sigma = \{a_1, \ldots, a_s\}$. Define a map $f : \Sigma \rightarrow \{0,1\}^*$ as $f(a_i) = 0^i1$ for $i \in [1,s]$. Extend this map to strings over $\Sigma^*$ as $F : \Sigma^* \rightarrow \{0,1\}^*$ where for any string $y = a_{i_1} a_{i_2} \cdots a_{i_k}, F(y) = f(a_{i_1}) f(a_{i_2}) \cdots f(a_{i_k})$. Observe that PCP instance $(A, B)$ has a solution iff the BPCP instance $\{(F(w_1), \ldots, F(w_n)), \{F(x_1), \ldots, F(x_n)\}\}$ has a solution, when $F$ is one-one. Suppose that $F(y) = F(z)$ for some $y, z \in \Sigma^*$. Let $y = a_{i_1} \cdots a_{i_k}$ and $z = a_{j_1} \cdots a_{j_l}$ for some $i_1, \ldots, i_k, j_1, \ldots, j_l \in [1,n]$, then $F(y) = f(a_{i_1}) f(a_{i_2}) \cdots f(a_{i_k}) = 0^{i_1} 10^{i_2} 1 \cdots 0^{i_k} 1 = f(a_{j_1}) f(a_{j_2}) \cdots f(a_{j_l}) = F(z)$. If $y \neq z$ then let $r$ be the minimum integer such that $a_{i_r} \neq a_{j_r}$. Then $f(a_{i_r}) = 0^{i_r} 1 \neq 0^{j_r} 1 = f(a_{j_r})$ but then this implies that $F(y) \neq F(z)$ contradicting our assumption. Therefore $x = z$ and as a consequence $F$ is 1-1.
4. Show that the language $PF = \{ G | G$ is a CFG and $L(G)$ is prefix-free $\}$ is undecidable.

**Solution:** We know that PCP is undecidable. This implies $\neg \text{PCP}$ is undecidable as well.

We describe a reduction $\neg \text{PCP} \leq_m \text{PF}$. Let $A = (w_1, w_2, \ldots, w_k)$ and $B = (x_1, x_2, \ldots, x_k)$ be an instance of $\neg \text{PCP}$ defined over alphabet $\Sigma$. Let $a_1, a_2, \ldots, a_k, \# \notin \Sigma$ be $k + 1$ new distinct symbols and let $\Sigma' = \Sigma \cup \{a_1, \ldots, a_k, \# \}$. Define a context-free grammar $G = (N = \{S, S_A, S_B\}, \Sigma', P, S)$ where $P$ consists of the following productions:

$$
S \rightarrow S_A \# \mid S_B \#,
$$

$$
S_A \rightarrow w_i S_A a_i \mid w_i a_i \quad \text{for } 1 \leq i \leq k,
$$

$$
S_B \rightarrow x_i S_A a_i \mid x_i a_i \quad \text{for } 1 \leq i \leq k.
$$

Observe that all strings derived from $S \rightarrow S_A \# \mid S_B \#$ end with $\# \$ and all strings derived from $S \rightarrow S_B \# \mid S_A \#$ end with $\#$. Suppose there are distinct strings $u, v \in L(G)$ such that $u$ is a prefix of $v$. Then all symbols in $u$ and $v$ up to and including $\#$ must match. That is, we can write $u = u' \#, v = v' \#$ such that $u' = v'$, $S_A \overset{\gamma}{\rightarrow} v'$ and $S_B \overset{\gamma}{\rightarrow} u'$. We now show that $(A, B) \in \neg \text{PCP}$ iff $L(G)$ is prefix-free. Suppose that $(A, B) \notin \neg \text{PCP}$. For a solution $i_1, i_2, \ldots, i_m$, we have $w_{i_1} w_{i_2} \cdots w_{i_m} = x_{i_1} x_{i_2} \cdots x_{i_m}$. Let $z = w_{i_1} w_{i_2} \cdots w_{i_m} a_{i_1} \cdots a_{i_m} = x_{i_1} x_{i_2} \cdots x_{i_m} a_{i_1} \cdots a_{i_m}$. By definition, $L(G)$ contains both $z \# \mid$ and $\#z$ and hence is not prefix-free. Suppose that $\exists u, v \in L(G)$ such that $u$ is a prefix of $v$. Then, $u = u' \#, v = v' \#$ such that $u' = v'$, $S_A \overset{\gamma}{\rightarrow} v'$ and $S_B \overset{\gamma}{\rightarrow} u'$. The string $v'$, derived from $S_A$, must be of the form $w_{i_1} w_{i_2} \cdots w_{i_m} a_{i_1} \cdots a_{i_m} a_{i_1} \cdots a_{i_m}$. Similarly, $u'$ has the form $x_{i_1} x_{i_2} \cdots x_{i_m} a_{i_1} \cdots a_{i_m} a_{i_1} \cdots a_{i_m}$. The $a_i$’s at the end must all match since $u' = v'$. As a result, $w_{i_1} w_{i_2} \cdots w_{i_m} = x_{i_1} x_{i_2} \cdots x_{i_m}$ implying that $i_1, i_2, \ldots, i_m$ is a solution for $(A, B)$ i.e., $(A, B) \notin \neg \text{PCP}$. We have shown that $(A, B) \notin \neg \text{PCP} \iff G \notin \text{PF}$ from which it follows that $(A, B) \notin \neg \text{PCP} \iff G \in \text{PF}$ and therefore $\text{PF}$ is undecidable.

5. For $A, B \subseteq \Sigma^*$, define $A/B = \{ x \in \Sigma^* | \exists y \in B \quad xy \in A \}$.

(a) Show that if $A$ and $B$ are recursively enumerable, then so is $A/B$.

**Solution:** Let $M_A, M_B$ be Turing machines accepting $A$, $B$ respectively. Define a TM $\mathcal{N}'$ that on input $x$ does the following.

- For each $y \in \Sigma^*$, simulate $M_B$ on $y$ on a time-shared basis. That is, simulate $M_B$ on $y_1$ for one step and then simulate it on $y_2$ for one step and continue simulations for some fixed ordering $y_1, y_2, \ldots$ of strings in $\Sigma^*$.

- If $M_B$ accepts, then simulate $M_A$ on $xy$.

- Halt and accept if $M_A$ accepts.

If $x \in A/B$, then for some $y \in \Sigma^*$, $M_B$ accepts $y$ eventually and $M_A$ accepts $xy$. Hence $A/B$ is recursively enumerable.

(b) Show that every r.e. set can be represented as $A/B$ with $A$ and $B$ being context-free languages.

**Solution:** Let $R$ be an r.e. set and let $M = (Q, \Sigma, \Gamma, \delta, \iota, s, t, r)$ be a Turing machine accepting it. Recall that we defined $\text{VALCOMPS}_{M,c}$ over the alphabet $\Delta = \{\#\} \cup (\Gamma \times (Q \cup \{-\}))$. For $x = a_1 a_2 \cdots a_n$, let $S_{M,x}$ be the starting configuration, given by $\iota \downarrow a_1 a_2 \cdots a_n$.
We now define the sets $A$ and $B$ over the alphabet $\Sigma \cup \Delta$ as follows:

$$A = \left\{ x#S_{M,x}#\alpha_1^R#\alpha_2#\cdots#\alpha'_N \mid \alpha_i \xrightarrow{\text{M}} \alpha_{i+1} \text{ for all odd } i \right\}$$

$$B = \left\{ #\alpha_0#\alpha_1^R#\alpha_2#\cdots#\alpha'_N \mid \alpha_i \xrightarrow{\text{M}} \alpha_{i+1} \text{ for all even } i \text{ and } \alpha_N \text{ contains } t \right\}$$

Here, $\alpha'_N = \alpha^R_N$ if $N$ is odd and $\alpha'_N = \alpha_N$ otherwise.

Convince yourself that $R = A/B$!

6. Let

$$f(x) = \begin{cases} 3x + 1 & \text{if } x \text{ is odd} \\ x/2 & \text{if } x \text{ is even} \end{cases}$$

for any natural number $x$. Define $C(x)$ as the sequence $x, f(x), f(f(x)), \ldots$, which terminates if and when it hits 1. For example, if $x = 7$, then

$$C(x) = (7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1).$$

Computer tests have shown that $C(x)$ hits 1 eventually for $x$ ranging from 1 to $87 \times 2^{60}$ (as of 2017). But, the question of whether $C(x)$ ends at 1 for all $x \in \mathbb{N}$ is not proven. This is believed to be true and known as the Collatz conjecture. Suppose that $MP$ were decidable by a Turing machine $K$. Use $K$ to describe a TM that is guaranteed to prove or disprove Collatz conjecture.

**Solution:** Let $N$ be a TM, that on input $x$, does the following.

- while $x \neq 1$,
  - if $x$ is even, $x \leftarrow x/2$
  - otherwise $x \leftarrow 3x + 1$
- accept and halt

Let $M$ be a TM, that on input $y$, does the following.

- erase $y$
- set $x \leftarrow 1$ and repeat the following
  - use $K$ to determine whether $N$ accepts $x$
  - if not, accept and halt
  - otherwise, $x \leftarrow x + 1$

If the Collatz conjecture is true, then $M$ runs forever; otherwise $M$ halts after finding a counter example (in fact, the smallest counter example).

We now use $K$ to determine whether or not $M$ accepts an arbitrary input $y$ to decide whether or not Collatz conjecture is true.