6.1. In each case below, consider \( R1 \) to be Regular, \( R2 \) to be finite, and \( L1 \) and \( L2 \) to be non-regular CFLs. Fill in the three columns with \( Y \) or \( N \), indicating what kind of language \( L \) can be. No proofs are required. Read \( \subseteq \) as “contained in and may equal.”
Put \( Y \) in all that are possible and \( N \) in all that are not.

<table>
<thead>
<tr>
<th>Definition of ( L )</th>
<th>Regular?</th>
<th>CFL, non-Regular?</th>
<th>Not even a CFL?</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L = L1 \cap L2 )</td>
<td>( Y )</td>
<td>( Y )</td>
<td>( Y )</td>
</tr>
<tr>
<td>( L = L1 - R2 )</td>
<td>( Y )</td>
<td>( Y )</td>
<td>( N )</td>
</tr>
<tr>
<td>( L = \Sigma^* - R1 )</td>
<td>( Y )</td>
<td>( N )</td>
<td>( N )</td>
</tr>
<tr>
<td>( L \subseteq R1 )</td>
<td>( Y )</td>
<td>( Y )</td>
<td>( Y )</td>
</tr>
</tbody>
</table>

3.2. Choosing from among (D) decidable, (U) undecidable, categorize each of the following decision problems. No proofs are required. \( L \) is a language over \( \Sigma \).

<table>
<thead>
<tr>
<th>Problem / Language Class</th>
<th>Regular</th>
<th>Context Free</th>
<th>Context Sensitive</th>
<th>Phrase Structured</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L ) contains ( \Sigma ) ?</td>
<td>( D )</td>
<td>( D )</td>
<td>( D )</td>
<td>( U )</td>
</tr>
<tr>
<td>(</td>
<td>L</td>
<td>) is infinite ?</td>
<td>( D )</td>
<td>( D )</td>
</tr>
</tbody>
</table>

4.3. Prove that any class of languages, \( C \), closed under union, concatenation, intersection with regular languages, homomorphism and substitution (e.g., the Context-Free Languages) is closed under **Cut with Regular Sets**, denoted by the operator \( || \), where \( L \in C \), \( R \) is Regular, \( L \) and \( R \) are both over the alphabet \( \Sigma \), and
\[
L||R = \{ x_1 x_2 \ldots x_k \mid k \geq 1, \forall i, y_i \in R \text{ and } x_1 y_1 x_2 y_2 \ldots x_k y_k \in L \text{ where each } x_i \in \Sigma^+ \}.
\]
You may assume substitution \( f(a) = \{ a, a' \} \), and homomorphisms \( g(a) = a' \) and \( h(a) = a, h(a') = \lambda \). Here \( a \in \Sigma \) and \( a' \) is a new character associated with each such \( a \in \Sigma \).
You only need give me the definition of \( L||R \) in an expression that obeys the above closure properties; you do not need to prove or even justify your expression.

\[
L||R = h( f(L) \cap ( \Sigma^+ g(R) )^+ )
\]

4.4. Specify True (T) or False (F) for each statement.

<table>
<thead>
<tr>
<th>Statement</th>
<th>T or F</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Context Sensitive Languages are closed under union</td>
<td>( T )</td>
</tr>
<tr>
<td>The Post Correspondence Problem is undecidable if (</td>
<td>\Sigma</td>
</tr>
<tr>
<td>The function ( \text{Univ}(f,x) = \phi(x) ) is primitive recursive</td>
<td>( F )</td>
</tr>
<tr>
<td>If ( \text{Halt} \leq_m P ) then ( P ) must be RE</td>
<td>( F )</td>
</tr>
<tr>
<td>The RE sets are closed under complement</td>
<td>( F )</td>
</tr>
<tr>
<td>Myhill-Nerode proves that every regular language has a minimum state DFA</td>
<td>( T )</td>
</tr>
<tr>
<td>If ( P ) is solvable then Rice’s Theorem cannot apply to ( P )</td>
<td>( T )</td>
</tr>
<tr>
<td>The incorrect traces of a Turing Machine’s Computations form a CFL</td>
<td>( T )</td>
</tr>
</tbody>
</table>
Let $P = \langle x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n \rangle$, $x_i, y_1 \in \Sigma^+$, be an arbitrary instance of PCP. We can use PCP’s undecidability to show the undecidability of the problem to determine if the language associated with a Context Sensitive Grammar is non-empty. Present a grammar, $G$, associated with an arbitrary instance of PCP, $P$, such that $L(G)$ is non-empty if and only if there is a solution to $P$.

Define $G = (\{S, T\} \cup \Sigma, \{\ast\}, R, S)$ where $R$ is the set of rules (this is your job):

$$
\begin{align*}
S & \rightarrow x_i S y_i^R \mid x_i T y_i^R \\
a \ T \ a & \rightarrow \ast \ T \ast \\
a \ast & \rightarrow \ast \ \alpha \\
\ast \ a & \rightarrow \alpha \ast \\
T & \rightarrow \ast
\end{align*}
$$

Choosing from among (REC) recursive, (RE) re non-recursive, (coRE) co-re non-recursive, (NRNC) non-re/non-co-re, categorize each of the sets in a) through d). Justify your answer by showing some minimal quantification of some known recursive predicate.

a.) $D = \{ <f, x> \mid \varphi(x) \text{ converges in at most } 10 \text{ steps} \}$

$$
\begin{array}{c}
\text{STP}(f, x, 10) \quad \text{REC}
\end{array}
$$

b.) $B = \{ f \mid \varphi f \text{ is not total} \}$

$$
\begin{array}{c}
\exists x \ \forall t \ \neg \text{STP}(f, x, t) \quad \text{NRNC}
\end{array}
$$

c.) $C = \{ f \mid \text{domain}(\varphi f) \text{ is non-empty} \}$

$$
\begin{array}{c}
\exists <x, t> \ \text{STP}(f, x, t) \quad \text{RE}
\end{array}
$$

d.) $A = \{ f \mid |\text{range}(\varphi f)| \leq 1 \}$

$$
\begin{array}{c}
\forall <x, y, t> \ [\text{STP}(f, x, t) \& \text{STP}(f, y, t) \Rightarrow \text{VALUE}(f, x, t) = \text{VALUE}(f, y, t)] \quad \text{CO-RE}
\end{array}
$$

Looking back at Question 6, which of these are candidates for using Rice’s Theorem to show their unsolvability? Check all for which Rice Theorem might apply.

a) ____ b) $\times$ c) $\times$ d) $\times$
8. We wish to prove that, if $S$ and its complement $S'$ are both be non-empty and recursively enumerable, then $S$ is recursive (decidable). There are two approaches. The first is based on the fact that there are algorithms, $f_S$ and $f_{S'}$, that enumerate $S$ and $S'$, respectively. The second is based on the fact that there are procedures (partial recursive functions), $g_S$ and $g_{S'}$, whose domains are $S$ and $S'$, respectively.

3a.) Define a characteristic function for $S$ based on the existence of $f_S$ and $f_{S'}$.

$$\chi_S(x) = f_S (\mu y | f_S(y) = x || f_{S'}(y) = x ) = x$$

3b.) Define a characteristic function for $S$ based on the existence of $g_S$ and $g_{S'}$.

$$\chi_S(x) = \text{STP} ( g_S, x, \mu t | \text{STP}( g_S, x, t ) || \text{STP}( g_{S'}, x, t ) )$$

6 9. Let sets $A$ be a non-empty recursive (decidable) set and let $B$ be re non-recursive (undecidable). Consider $C = \{ z | z = \min(x,y), \text{where } x \in A \text{ and } y \in B \}$. For (a)-(c), either show sets $A$ and $B$ and the resulting set $C$, such that $C$ has the specified property (argue convincingly that it has the correct property) or demonstrate (prove by construction) that this property cannot hold.

a. Can $C$ be recursive? Circle $Y$ or $N$.

$A = \{ 0 \} \quad B = K$

$C = A = \{ 0 \} \text{ which is recursive}$

b. Can $C$ be re non-recursive? Circle $Y$ or $N$.

$A = \{ 2x \mid x \in \mathbb{N} \} \quad B = \{ 2x + 1 \mid x \in K \}$

$C = A \cup B \text{ as they are disjoint sets and each elements un each set is the minimum of some pair with the other set. The membership problem for K reduces to that of C and vice versa. Thus, C is re, non-recursive}$

c. Can $C$ be non-re? Circle $Y$ or $N$.

You may assume $A = \text{range}(f_A), B = \text{range}(f_B)$, for some algorithms $f_A, f_B$.

$$f_{\min(A,B)} (<x, y>) = \min ( f_A(x), f_B(y) )$$

The range of $f_{\min(A,B)}$ is then the set of minimums of the sets $A$ and $B$. This shows $C$ is the range of some procedure and so is RE
10. Define PseudoFIB (PF) = \{ f \mid \text{for some input } x, f(x+2) = f(x+1)+f(x) \}.

2 a.) Show some minimal quantification of some known recursive predicate that provides an upper bound for the complexity of this set. (Hint: Look at c.) and d.) to get a clue as to what this must be.

\[ \exists <x, t> \{ \text{STP}(f, x+2, t) \& \text{STP}(f, x+1, t) \& \text{STP}(f, x, t) \& \]
\[ \text{VALUE}(f, x+2, t) = \text{VALUE}(f, x+2, t) + \text{VALUE}(f, x, t) \} \]

5 b.) Use Rice’s Theorem to prove that PF is undecidable.

First, PF is non-trivial as C0 \in PF and C1 \notin PF

Next, let f and g, be arbitrary function indices such that \forall x f(x) = g(x)

\[ f \in PF \iff \exists x [ f(x+2) = f(x)+g(x+1) ] \iff \]
\[ \exists x [ g(x+2) = g(x)+g(x+1) \text{ since } \forall x f(x) = g(x) \]
\[ \iff g \in PF \]

By the Strong Form of Rice’s Theorem, PF is thus shown to be undecidable

5 c.) Show that K \leq_m PF, where K = \{ f \mid f(f) \uparrow \}.

Let f be an arbitrary function index

Define \forall x F_f (x) = f(f) – f(f)

\[ f \in K \iff \forall x F_f (x) = 0 \Rightarrow F_f \in PF \]
\[ f \notin K \iff \forall x F_f (x) \uparrow \Rightarrow F_f \notin PF \]

Thus K \leq_m PF as was to be shown

1 d.) From a.) through c.) what can you conclude about the complexity of PF (Recursive, RE, RE-Complete, CO-RE, CO-RE-Complete, NON-RE/NON-CO-RE)?

RE-Complete since PF is RE, K \leq_m PF, and K is RE-Complete