Post’s Correspondence Problem (PCP)

**Input:** Two sequences, \( A = w_1, \ldots, w_k \) and \( B = x_1, \ldots, x_k \), where each \( w_i \) and \( x_i \) is a string over some alphabet \( \Sigma \).

**Question:** Is there a sequence \( i_1, \ldots, i_m \) such that 
\[1 \leq i_j \leq k \text{ for } 1 \leq j \leq m \text{ and } w_{i_1} \cdots w_{i_m} = x_{i_1} \cdots x_{i_m}\]?

**Example:**
\[
\begin{align*}
A &= 1, 10111, 10 \\
B &= 111, 10, 0
\end{align*}
\]

**Solution:** 2, 1, 1, 3:
**Defn:** Let $L_{PCP}$ be the set of all strings of the form $0^k \# w_1 \# \cdots \# w_k \# x_1 \# \cdots \# x_k$ such that $w_1, \ldots, w_k, x_1, \ldots, x_k \in \{0, 1\}^*$ and for some $i_1, \ldots, i_m$, $w_{i_1} \cdots w_{i_m} = x_{i_1} \cdots x_{i_m}$.

**Claim:** $L_{PCP}$ is not recursive; i.e., PCP is undecidable.

**Modified PCP (MPCP):** Same as PCP, except that a solution is required to start with index 1.

**Claim:** $L_{MPCP}$ is not recursive.

We will show that $L_U \leq L_{MPCP} \leq L_{PCP}$.

$L_U \leq L_{MPCP}$ (outline):

We will describe:

1. for arbitrary $v \in \{0, 1\}^*$, an instance of MPCP that has a solution iff $v \in L_U$; and

2. a TM $M$, that, given $v \in \{0, 1\}^*$, produces a string $y \in \{0, 1, \#\}^*$ such that $y \in L_{MPCP}$ iff $v \in L_U$.

Let $U = (Q, \{0, 1\}, \Gamma, \delta, q_0, B, F)$ be a universal TM, and let $\$$ be a symbol not in $Q \cup \Gamma$. 
Given \( v \in \{0,1\}^* \), consider the following instance of MPCP:

- Let \( w_1 = \$ \) and \( x_1 = \$ q_0 v \$ \).
- For each \( X \in \Gamma \cup \{\$\} \), include the pair \( \langle X, X \rangle \).
- For all \( q \in Q - F, p \in Q, X, Y, Z \in \Gamma \), include
  - \( \langle qX, Yp \rangle \) if \( \delta(q, X) = (p, Y, R) \);
  - \( \langle ZqX, pZY \rangle \) and \( \langle qX, qBY \rangle \) if \( \delta(q, X) = (p, Y, L) \);
  - \( \langle q\$, Yp$ \rangle \) if \( \delta(q, B) = (p, Y, R) \);
  - \( \langle Zq\$, pZY$ \rangle \) and \( \langle q\$, qBY$ \rangle \) if \( \delta(q, B) = (p, Y, L) \).
- For each \( q \in F, X \in \Gamma \), include \( \langle Xq, q \rangle \), \( \langle qX, q \rangle \), and \( \langle q\$$, $ \rangle \).

It can be shown that this instance has a solution iff \( v \in L_U \).

Note that this instance has alphabet \( Q \cup \Gamma \cup \{\$\} \), which is independent of \( v \).

The TM \( M \):

- We encode the symbols of the MPCP instance as fixed-length strings over \( \{0,1\} \).
- Only \( x_1 \) in the MPCP instance depends on \( v \).
- All of \( y \) except the encoding of \( v \) is stored in the finite control of \( M \).
- \( M \) simply encodes its input \( v \) and inserts it into \( y \).
MPCP ≤ PCP:

Let \((A, B)\) be an instance of MPCP over \(\Sigma\), and let \(*\) and \(\$\) be distinct symbols not in \(\Sigma\).

From \(A = w_1, \ldots, w_k\), we construct \(A' = w'_1, \ldots, w'_{k+1}\) as follows:

- Insert \(*\) after each symbol in \(w_1, \ldots, w_k\).
- Also, insert \(*\) before the first symbol in \(w_1\).
- Let \(w'_{k+1} = \$\).

From \(B = x_1, \ldots, x_k\), we construct \(B' = x'_1, \ldots, x'_{k+1}\) as follows:

- Insert \(*\) before each symbol in \(x_1, \ldots, x_k\).
- Let \(x'_{k+1} = \*\$\).

It is easily seen that \((A, B)\) has a solution for MPCP iff \((A', B')\) has a solution for PCP.

The construction is clearly computable.
Undecidable CFG Problems

**Theorem 9.20:** The problem of deciding whether a given CFG is ambiguous is undecidable.

**Proof sketch:** We will reduce PCP to this problem.

**Arbitrary PCP Instance:** $w_1, \ldots, w_k, x_1, \ldots, x_k$ over $\Sigma$.

Let $a_1, \ldots, a_k$ be symbols not in $\Sigma$.

Let $G$ be the CFG given by:

$$S \rightarrow A \mid B$$
$$A \rightarrow w_iAa_i \mid w_ia_i \quad \text{for } 1 \leq i \leq k$$
$$B \rightarrow x_iBa_i \mid x_ia_i \quad \text{for } 1 \leq i \leq k$$

where $S$, $A$, and $B$ are distinct symbols not in $\Sigma \cup \{a_1, \ldots, a_k\}$.

If the instance of PCP has a solution, then $G$ is clearly ambiguous.
Suppose $G$ is ambiguous, and let $z \in (\Sigma \cup \{a_1, \ldots, a_k\})^*$, 
$\alpha, \beta, \gamma_1, \gamma_2 \in (\Sigma \cup \{a_1, \ldots, a_k, S, A, B\})^*$, and 
$C \in \{S, A, B\}$ such that 
\begin{itemize}
  \item $S \xrightarrow{*} \alpha C \beta \Rightarrow \alpha \gamma_1 \beta \xrightarrow{*} z$;
  \item $S \xrightarrow{*} \alpha C \beta \Rightarrow \alpha \gamma_2 \beta \xrightarrow{*} z$; and
  \item $\gamma_1 \neq \gamma_2$.
\end{itemize}
It is easily seen that there is at most one derivation $A \Rightarrow \cdots \Rightarrow z$ and at most one derivation $B \Rightarrow \cdots \Rightarrow z$; hence $\alpha C \beta = S$.

The string obtained by removing all $a_i$s from $z$ is a solution to the instance of PCP. The construction is clearly computable. Therefore, the ambiguity problem is undecidable.

\textbf{Lemma:} Let $w_1, \ldots, w_k$ be strings over $\Sigma$, and let 
$a_1, \ldots, a_k$ and $S$ be distinct symbols not in $\Sigma$. Let 
$G = (\{S\}, \Sigma \cup \{a_1, \ldots, a_k\}, P, S)$ be the CFG such that 
$P$ is given by 
\begin{align*}
  S &\rightarrow w_i S a_i \mid w_i a_i \quad \text{for } 1 \leq i \leq k.
\end{align*}
Then we can effectively construct a CFG $G'$ such that 
$L(G') = \overline{L(G)}$. 

Proof sketch:

- We can easily construct a PDA $M$ such that $L(M) = \overline{L(G)}^R$.
- We can construct a CFG $G''$ such that $L(G'') = L(M) = \overline{L(G)}^R$.
- By reversing the right-hand-sides of all productions in $G''$, we obtain a CFG $G'$ such that $L(G') = \overline{L(G)}$.

Theorem 9.22: The following problems are undecidable, where $G_1$ and $G_2$ denote given CFGs and $R$ denotes a given regular expression:

- Is $L(G_1) \cap L(G_2) \neq \emptyset$?
- Is $L(G_1) \neq L(G_2)$?
- Is $L(G_1) \neq L(R)$?
- Is $L(G_1) \neq T^*$ for every alphabet $T$?
- Is $L(G_2) - L(G_1) \neq \emptyset$?
- Is $L(R) - L(G_1) \neq \emptyset$?
We will reduce PCP to each of these problems.

**Aribtrary PCP Instance:** \( w_1, \ldots, w_k, x_1, \ldots, x_k \) over \( \Sigma \).

Let \( I = \{a_1, \ldots, a_k\} \) be an alphabet such that \( \Sigma \cap I = \emptyset \).

Define

\[
G_A: S \rightarrow w_iSa_i \mid w_ia_i \text{ for } 1 \leq i \leq k.
\]

\[
G_B: S \rightarrow x_iSa_i \mid x_ia_i \text{ for } 1 \leq i \leq k.
\]

Then \( L(G_A) \cap L(G_B) \neq \emptyset \) iff the PCP instance has a solution.

We can construct CFGs \( G'_A \) and \( G'_B \) such that \( L(G'_A) = \overline{L(G_A)} \) and \( L(G'_B) = \overline{L(G_B)} \).

Furthermore, we can construct the following:

- a CFG \( G_1 \) such that \( L(G_1) = L(G'_A) \cup L(G'_B) = \overline{L(G_A)} \cap \overline{L(G_B)} \);
- a CFG \( G_2 \) such that \( L(G_2) = (\Sigma \cup I)^* \); and
- a regular expression \( R \) such that \( L(R) = (\Sigma \cup I)^* \).
Each of the following holds iff the PCP instance has a solution:

- $L(G_1) \neq L(G_2)$
- $L(G_1) \neq L(R)$
- $L(G_1) \neq T$ for every alphabet $T$
- $L(G_2) - L(G_1) \neq \emptyset$
- $L(R) - L(G_1) \neq \emptyset$