Outline

1. PCP

2. Decision problems about CFGs
PCP reduction

**Given:** \( \langle M, w \rangle \) in encoded form

**To construct:** an instance \((A, B)\) of MPCP such that \(M\) accepts \(w\) if and only if \((A, B)\) has a solution.

**Preliminaries:**

- Assume w.l.o.g. that
  - \(M\) never prints a blank;
  - \(M\) never moves left from the first tape square (i.e. never hangs).
- Starting configuration: \((q_0, \#w)\)
- Representation of configuration: \(\alpha q \beta\)
  (If head is on a blank, i.e. at the end of the written portion of the tape, \(\beta = \varepsilon\).)
<table>
<thead>
<tr>
<th>Group</th>
<th>List $A$</th>
<th>List $B$</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group I</strong></td>
<td>(\vdash)</td>
<td>(\vdash q_o w \vdash)</td>
<td></td>
</tr>
<tr>
<td><strong>Group II</strong></td>
<td>$a$</td>
<td>$a$</td>
<td>for any $a \in \Sigma$</td>
</tr>
<tr>
<td></td>
<td>(\vdash)</td>
<td>(\vdash)</td>
<td></td>
</tr>
<tr>
<td><strong>Group IIIA</strong></td>
<td>(qa)</td>
<td>(pb)</td>
<td>if $b \in \Sigma$</td>
</tr>
<tr>
<td>when $\delta(q, a) = (p, b)$</td>
<td>(qa)</td>
<td>(ap)</td>
<td>if $b = R$</td>
</tr>
<tr>
<td></td>
<td>(cqa)</td>
<td>(pca)</td>
<td>if $b = L$</td>
</tr>
<tr>
<td><strong>Group IIIB</strong></td>
<td>$q \vdash$</td>
<td>(pb \vdash)</td>
<td>if $b \in \Sigma$</td>
</tr>
<tr>
<td>when $\delta(q, #) = (p, b)$</td>
<td>$q \vdash$</td>
<td>#$'p \vdash$</td>
<td>if $b = R$</td>
</tr>
<tr>
<td></td>
<td>(cq \vdash)</td>
<td>(pc \vdash)</td>
<td>if $b = L$</td>
</tr>
</tbody>
</table>
## PCP instance

<table>
<thead>
<tr>
<th></th>
<th>List $A$</th>
<th>List $B$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group IV</strong></td>
<td>$abh$  $h$</td>
<td>$h$</td>
</tr>
<tr>
<td></td>
<td>$ah$  $h$</td>
<td>$h$</td>
</tr>
<tr>
<td></td>
<td>$hb$  $h$</td>
<td>$h$</td>
</tr>
<tr>
<td><strong>Group V</strong></td>
<td>$h$  $</td>
<td>$  $</td>
</tr>
</tbody>
</table>
Lemma: $M$ accepts $w$ if and only if the MPCP instance $(A, B)$ above has a solution.

Proof sketch: ($\Rightarrow$)

1. Start with pair from group I.

2. Use pairs from group III to simulate one move in the region around the head position; use pairs from group II to copy rest of the tape contents.

3. If $M$ halts, use pairs from group IV to progressively erase the “tape” in the $B$ string.

   $A$ string: $\ldots \vdash a_1 \ldots a_i ha_{i+1} \ldots \vdash$

   $B$ string: $\ldots \vdash a_1 \ldots a_i ha_{i+1} \ldots \vdash a_1 \ldots a_{i-1} ha_{i+2} \ldots \vdash$

4. Terminate with pair from group V.
PCP is undecidable

Proof sketch (contd.): \((\iff)\)

Solution must begin with: \(A\) string = \(\vdash\) \(B\) string = \(\vdash q_0 w \vdash\).

Partial solutions have the form: \(A\) string = \(x\) \(B\) string = \(xy\)

where

\[
x = \text{sequence of configurations separated by } \vdash, \text{ last one } (w_C, \text{ say}) \\
y = \text{computation upto and including } w_C \text{ followed by prefix of a configuration that follows } w_C
\]

Partial solution can be completed only if \(M\) halts; otherwise, \(B\) string is always longer.
Outline

1. PCP

2. Decision problems about CFGs
Ambiguity of CFGs

Given: \( A = w_1, w_2, \ldots, w_k, \quad B = x_1, x_2, \ldots, x_k. \)

Reduction:
Let \( G_A : A \to w_1Aa_1 \mid \ldots \mid w_kAa_k \mid w_1a_1 \mid \ldots \mid w_ka_k \)
and \( G_B : B \to x_1Ba_1 \mid \ldots \mid x_kBa_k \mid x_1a_1 \mid \ldots \mid x_ka_k \)
Let \( G_{AB} : S \to A \mid B \ldots \)
Then \( G_{AB} \) is ambiguous if and only if \((A, B)\) has a solution.
Ambiguity of CFGs

Given: \( A = w_1, w_2, \ldots, w_k, \quad B = x_1, x_2, \ldots, x_k. \)

Reduction:
Let \( G_A : A \rightarrow w_1 A a_1 \mid \ldots \mid w_k A a_k \mid w_1 a_1 \mid \ldots \mid w_k a_k \)
and \( G_B : B \rightarrow x_1 B a_1 \mid \ldots \mid x_k B a_k \mid x_1 a_1 \mid \ldots \mid x_k a_k \)
Let \( G_{AB} : S \rightarrow A \mid B \ldots \)
Then \( G_{AB} \) is ambiguous if and only if \((A, B)\) has a solution.

List languages
\( L(G_A) \) and \( L(G_B) \) are called list languages.
**Lemma:** If $L_A$ is a list language, then $\overline{L_A}$ is a CFL.

Proof: Let $P$ be a (D)PDA that works as follows.

1. On symbols in $\Sigma$, $P$ pushes the input symbol on stack.
2. On symbols $a_1, \ldots, a_k$, it pops the stack to see if the top of the stack contains $w_i^R$.
3. If input and stack are both empty, reject.
4. In all other cases, accept.
Theorem: Let $G_1$ and $G_2$ be CFGs, and $R$ a regular expression. Then the following are undecidable.

- $L(G_1) \cap L(G_2) = \emptyset$ ?
Theorem: Let $G_1$ and $G_2$ be CFGs, and $R$ a regular expression. Then the following are undecidable.

- $L(G_1) \cap L(G_2) = \emptyset$?

Let $L(G_1) = L_A$, $L(G_2) = L_B$. 

Let $L(G_2) = L_A [L_B]$, $L(G_1) = [I]$. 

M.Mitra (ISI)
Theorem: Let $G_1$ and $G_2$ be CFGs, and $R$ a regular expression. Then the following are undecidable.

- $L(G_1) \cap L(G_2) = \emptyset$ ?
  - Let $L(G_1) = L_A$, $L(G_2) = L_B$.
- $L(G_1) = L(G_2)$ ?
Theorem: Let $G_1$ and $G_2$ be CFGs, and $R$ a regular expression. Then the following are undecidable.

- $L(G_1) \cap L(G_2) = \emptyset$ ?
- $L(G_1) = L(G_2)$ ?

Let $L(G_1) = L_A$, $L(G_2) = L_B$.

Let $L(G_1) = \overline{L_A} \cup \overline{L_B}$, $L(G_2) = (\Sigma \cup I)^*$.
Theorem: Let $G_1$ and $G_2$ be CFGs, and $R$ a regular expression. Then the following are undecidable.

- $L(G_1) \cap L(G_2) = \emptyset$?
- $L(G_1) = L(G_2)$?
- $L(G_1) = L(R)$?

Let $L(G_1) = L_A$, $L(G_2) = L_B$.

Let $L(G_1) = \overline{L_A} \cup \overline{L_B}$, $L(G_2) = (\Sigma \cup I)^*$.
Theorem: Let $G_1$ and $G_2$ be CFGs, and $R$ a regular expression. Then the following are undecidable.

- $L(G_1) \cap L(G_2) = \emptyset$?
- $L(G_1) = L(G_2)$?
- $L(G_1) = L(R)$?

Let $L(G_1) = L_A$, $L(G_2) = L_B$.

Let $L(G_1) = \overline{L_A} \cup \overline{L_B}$, $L(G_2) = (\Sigma \cup I)^*$.

Let $L(G_1) = \overline{L_A} \cup \overline{L_B}$, $R = (\Sigma \cup I)^*$.
Theorem: Let $G_1$ and $G_2$ be CFGs, and $R$ a regular expression. Then the following are undecidable.

- $L(G_1) \cap L(G_2) = \emptyset$?
- $L(G_1) = L(G_2)$?
- $L(G_1) = L(R)$?
- $L(G_1) \subseteq L(G_2)$?

Let $L(G_1) = L_A$, $L(G_2) = L_B$.

Let $L(G_1) = \overline{L_A} \cup \overline{L_B}$, $L(G_2) = (\Sigma \cup I)^*$.

Let $L(G_1) = \overline{L_A} \cup \overline{L_B}$, $R = (\Sigma \cup I)^*$. 

M.Mitra (ISI)
**Theorem:** Let $G_1$ and $G_2$ be CFGs, and $R$ a regular expression. Then the following are undecidable.

- $L(G_1) \cap L(G_2) = \emptyset$?
- $L(G_1) = L(G_2)$?
- $L(G_1) = L(R)$?
- $L(G_1) \subseteq L(G_2)$?

Let $L(G_1) = L_A$, $L(G_2) = L_B$.

Let $L(G_1) = \overline{L_A} \cup \overline{L_B}$, $L(G_2) = (\Sigma \cup I)^*$.

Let $L(G_1) = \overline{L_A} \cup \overline{L_B}$, $R = (\Sigma \cup I)^*$.

Let $L(G_2) = \overline{L_A} \cup \overline{L_B}$, $L(G_1) = (\Sigma \cup I)^*$. 
CFG related problems

**Theorem:** Let $G_1$ and $G_2$ be CFGs, and $R$ a regular expression. Then the following are undecidable.

- $L(G_1) \cap L(G_2) = \emptyset$ ?
- $L(G_1) = L(G_2)$ ?
- $L(G_1) = L(R)$ ?
- $L(G_1) \subseteq L(G_2)$ ?
- $L(R) \subseteq L(G_1)$ ?

Let $L(G_1) = L_A$, $L(G_2) = L_B$.

Let $L(G_1) = \overline{L_A} \cup \overline{L_B}$, $L(G_2) = (\Sigma \cup I)^*$.  

Let $L(G_1) = \overline{L_A} \cup \overline{L_B}$, $R = (\Sigma \cup I)^*$.  

Let $L(G_2) = \overline{L_A} \cup \overline{L_B}$, $L(G_1) = (\Sigma \cup I)^*$.  

M.Mitra (ISI)