COMS W3261
Computer Science Theory
Lecture 16
Post's Correspondence Problem

Alexander Roth

2014 – 10 – 29

Outline

1. Review
2. Post's correspondence problem
3. Modified PCP
4. Undecidability of the ambiguity problem for CFG's

1 Review

- The diagonal language L_d is not recursively enumerable.
- The universal language L_u is recursively enumerable but not recursive.
- The complement of the diagonal language is recursively enumerable but not recursive.
- The complement of the universal language is not recursively enumerable.

2 Post's Correspondence Problem (PCP)

- An instance of Post's correspondence problem consists of two lists of strings over some alphabet where the two lists have the same number of strings. Let $A = (w_1, w_2, \ldots, w_k)$ and $B = (x_1, x_2, \ldots, x_k)$ be the two lists.
- A solution to this instance of PCP, if one exists, is a sequence of one or more integers $i_1, i_2, \ldots, i_m$ such that $w_{i_1} w_{i_2} \ldots w_{i_m} = x_{i_1} x_{i_2} \ldots x_{i_m}$.

- Example: Let $A = (a, b, ca, abc)$ and $B = (ab, ca, a, c)$. The sequence $1, 2, 3, 1, 4$ is a solution because the same string $abcaaabc$ is obtained by concatenating the corresponding strings from either list A $[(a)(b)(ca)(a)(abc)]$.
- Post's correspondence problem is to determine whether an instance has a solution.
- We will show that Post's correspondence problem is undecidable by reducing the universal language to PCP.
- We will then show that the ambiguity problem for CFG's is undecidable by reducing PCP to the ambiguity problem for CFG's.

3 Modified PCP

- The Modified PCP has the additional requirement that the first string from list A and the first string from list B has to be the first string in the solution. The example above has this property.
- Formally, a solution to an instance of the MPCP is a sequence of zero or more integers $i_1, i_2, \ldots, i_m$ such that $w_1 w_{i_1} w_{i_2} \ldots w_{i_m} = x_1 x_{i_1} x_{i_2} \ldots x_{i_m}$.
- We can show that the PCP problem can be reduced to the modified PCP problem as follows:
  - We are given an instance of MPCP with lists $A = (w_1, w_2, \ldots, w_k)$ and $B = (x_1, x_2, \ldots, x_k)$.
  - Assume * and $ are new symbols.
  - From (A, B) we construct a PCP instance (C, D) with $C = (y_0, y_1, \ldots, y_{k+1})$ and $D = (z_0, z_1, \ldots, z_{k+1})$ where
    1. y_i is w_i with a * after each symbol in w_i, for $i = 1, 2, \ldots, k$.
    2. z_i is x_i with a * before each symbol in x_i, for $i = 1, 2, \ldots, k$.
    3. $y_0 = *y_1$ and $y_{k+1} = \$$.
    4. $z_0 = z_1$ and $z_{k+1} = *\$$.
  - We can show $i_1, i_2, \ldots, i_m$ is a solution to the given (A, B)-MPCP instance iff $0, i_1, i_2, \ldots, i_m, i_{k+1}$ is a solution to this constructed (C, D)-PCP instance.

4 Reducing the Universal Language to MPCP

- We can show that given (M, w), an instance of L_u, we can reduce this instance of L_u to an instance (A, B) of the MPCP such that M accepts w iff (A, B) has a solution. We do this by showing that (A, B) simulates the computation of M on w.
- This shows that both the MPCP and the PCP problems are undecidable.

5 Undecidability of the Ambiguity Problem for CFG's

- We can reduce an instance of the PCP problem to an instance of determining whether a CFG is ambiguous, thereby showing it is undecidable to determine whether a CFG is ambiguous.
- We will illustrate the reduction with the following example. Let (A, B) be an instance of the PCP problem with $A = (a, b, ca, abc)$ and $B = (ab, ca, a, c)$. Let G be the CFG with the productions

$$
\begin{aligned}
&S \to A \mid B \\
&A \to aA1 \mid bA2 \mid caA3 \mid abcA4 \mid a1 \mid b2 \mid ca3 \mid abc4 \\
&B \to abB1 \mid caB2 \mid aB3 \mid cB4 \mid ab1 \mid ca2 \mid a3 \mid c4
\end{aligned}
$$

- There are two distinct leftmost derivations for the string $abcaaabc41321$ because this instance of the PCP problem has a solution.