The Post Correspondence Problem

Mark Greenstreet, CpSc 421, Term 1, 2008/09

- The Post Correspondence Problem (PCP)
  - Definition
  - Examples
  - Demonstrating undecidability of PCP
- Reductions summary

The Post Correspondence Problem

- Given a set, P of pairs of strings:

$$P = \left\{ \begin{bmatrix} \frac{t_1}{b_1} \end{bmatrix}, \begin{bmatrix} \frac{t_2}{b_2} \end{bmatrix}, \ldots \begin{bmatrix} \frac{t_k}{b_k} \end{bmatrix} \right\}$$

  where each $t_i, b_i \in \Sigma^*$,

- Question: Does there exist a sequence i_1, i_2, ... i_n such that:

$$t_{i_1} t_{i_2} \cdots t_{i_n} = b_{i_1} b_{i_2} \cdots b_{i_n} \quad ?$$

  Note: the same pair can occur multiple times, i.e. there can be $j \neq m$ s.t. $i_j = i_m$.

A PCP Example

- Let $P = \left\{ \left[\frac{\text{a}}{\text{ab}}\right]_1, \left[\frac{\text{ab}}{\text{bb}}\right]_2, \left[\frac{\text{ba}}{\text{aa}}\right]_3, \left[\frac{\text{bc}}{\text{cc}}\right]_4, \left[\frac{\text{ca}}{\text{aa}}\right]_5, \left[\frac{\text{cd}}{\text{d}}\right]_6 \right\}$.

  (I've numbered the tiles to make it easier to talk about them.)

- Does the PCP problem P have a solution?

Another PCP Example

- Let $P = \left\{ \begin{bmatrix} \text{a} \\ \text{ab} \end{bmatrix}_1, \begin{bmatrix} \text{b} \\ \text{cc} \end{bmatrix}_2, \begin{bmatrix} \text{c} \\ \text{b} \end{bmatrix}_3, \begin{bmatrix} \text{c} \\ \text{d} \end{bmatrix}_4, \begin{bmatrix} \text{ddddd} \\ \end{bmatrix}_5, \begin{bmatrix} \text{ddde} \\ \text{e} \end{bmatrix}_6 \right\}$.
- Does the PCP problem P have a solution?

Another PCP Example

- Let $P = \left\{ \left[\frac{\text{a}}{\text{ab}}\right]_1, \left[\frac{\text{b}}{\text{cc}}\right]_2, \left[\frac{\text{c}}{\text{b}}\right]_3, \left[\frac{\text{c}}{\text{d}}\right]_4, \left[\frac{\text{ddddd}}{}\right]_5, \left[\frac{\text{ddde}}{\text{e}}\right]_6 \right\}$.
- Does the PCP problem P have a solution?
  - P has a solution iff $\exists n.\ (2^n \bmod 5) = 3$.
  - Yes (let $n = 3$).

PCP is undecidable

- Proof by computational histories.
- Sketch:
  - Start with a pair that has the initial configuration for a TM on the bottom and an empty string on top.
  - Include pairs in P whose top strings match the current configuration, and whose bottom strings build the next configuration.
  - A bunch of details to:
    - Account for moving the tape head.
    - Extend the tape with blanks when needed.
    - Force the first pair of a solution to be the one that gives the initial configuration.
    - ...
- A Simplifying Assumption:
  - We'll assume that any solution must start with tile 1 –
    we'll call this the "Modified Post Correspondence Problem" (MPCP).
  - (Don't worry.) We'll remove this assumption later.

Tile 1

- We'll reduce A_{TM} to MPCP.
- Let $M\#w$ be a string where M describes a TM and w describes an input string to M.
- The first tile will give the initial TM configuration as the bottom string, and an empty string on top. We'll use $\#$ (with $\# \notin \Gamma$) as the end marker for configurations.

$$\boxed{\begin{array}{c} \# \\ \hline \#q_0w\# \end{array}}_1 \quad \in \quad P$$

From one configuration to the next

- At each step, we copy the current configuration from the bottom string to the upper string, and build the next configuration on the lower string:

$$\left[\frac{\#C_0\#C_1\#\ldots C_{k-1}\#}{\#C_0\#C_1\#\ldots C_{k-1}\#C_k}\right] \longrightarrow \left[\frac{\#C_0\#C_1\#\ldots C_{k-1}\#C_k\#}{\#C_0\#C_1\#\ldots C_{k-1}\#C_k\#C_{k+1}}\right]$$

- A configuration looks like $\alpha bqc\beta$.
- To calculate the next configuration, we
  - Copy α to the upper and lower strings.
  - Copy αbqc to the upper string and write its successor to the lower string.
  - Copy β to the upper and lower strings.
- To copy α and β we include the following tile in P for each $c \in \Gamma$: $\left[\frac{c}{c}\right]$.
- The next two slides describe how to handle transitions.

All the Right Moves

For each transition $\delta(q, c) = (q', c', R)$:

- We add the tile $\begin{array}{|c|}\hline qc \\ \hline c'q' \\ \hline\end{array}$ to P. This enables the move:

$$\begin{array}{|l|}\hline \# \ldots \# \alpha \\ \hline \# \ldots \# \alpha q c \beta \# \alpha \\ \hline\end{array} \longrightarrow \begin{array}{|l|}\hline \# \ldots \# \alpha q c \\ \hline \# \ldots \# \alpha q c \beta \# \alpha c' q' \\ \hline\end{array}$$

- If $c = \square$, we also add the tile $\begin{array}{|c|}\hline q\# \\ \hline c'q'\# \\ \hline\end{array}$ to handle the case when the head is moving further into the infinite string of blanks at the end of the tape.

All the Left Moves

For each transition $\delta(q, c) = (q', c', L)$:

- for each $b \in \Gamma$ we add the tile $\boxed{\begin{array}{c} bqc \\ \hline q'bc' \end{array}}$ to P. This enables the move:

$$\boxed{\begin{array}{l} \# \ldots \#\alpha \\ \hline \# \ldots \#\alpha bqc\beta\#\alpha \end{array}} \quad \longrightarrow \quad \boxed{\begin{array}{l} \# \ldots \#\alpha bqc \\ \hline \# \ldots \#\alpha bqc\beta\#\alpha q'bc' \end{array}}$$

- We also add the tile $\boxed{\begin{array}{c} \#qc \\ \hline \#q'c' \end{array}}$ to P to handle the case when the head is at the left end of the tape.

The End Game

- M accepts w iff we can reach a configuration for our MPCP

$$\left[\begin{array}{|l|}\hline \#C_0 \ldots \#C_{n-1}\# \\ \hline \#C_0 \ldots \#C_{n-1}\#\alpha q_{accept}\beta\# \\ \hline\end{array}\right]$$

- Now we have to "fix" the problem that we've got one more configuration on the lower tape than the upper one. For each $c \in \Gamma$ we add the tiles:

$$\left[\begin{array}{|c|}\hline c q_{accept} \\ \hline q_{accept} \\ \hline\end{array}\right], \left[\begin{array}{|c|}\hline q_{accept} c \\ \hline q_{accept} \\ \hline\end{array}\right].$$

- These allow us to discard one tape symbol each time we copy the configurations until we get to:

$$\left[\begin{array}{|l|}\hline \#C_0 \ldots \#q_{accept}c\# \\ \hline \#C_0 \ldots \#q_{accept}c\#q_{accept}\# \\ \hline\end{array}\right]$$

So, we add one more tile to our set: $\left[\begin{array}{|c|}\hline q_{accept}\#\# \\ \hline \# \\ \hline\end{array}\right]$.

- Now, we have an instance of MPCP that has a solution iff M accepts

A Star is Born

- We need to force our $tile_1$ (see slide 6) to be the first tile of any solution.
- Let $\star$ be a new symbol (i.e. not in $\Gamma \cup \{\#\}$).
- For any string, s, let $\star s$ be the string obtained by inserting a $\star$ before each symbol of s. For example, $\star(abc) = \star a \star b \star c$.
- For any string, s, let $s\star$ be the string obtained by adding a $\star$ before each symbol of s. For example, $(abc)\star = a \star b \star c\star$.
- Finally, $\star s \star$, puts on star between each pair of symbols of s and one star at the beginning of s and one at the end. For example, $\star(abc)\star = \star a \star b \star c\star$.

From MPCP to PCP

- Given a set of tiles, P for MPCP as described above:
  - Replace the initial tile, $\left[\frac{\#}{\#q_0w\#\star}\right]$ with $\left[\frac{\star\#}{\star\#q_0w\#\star}\right]$.
  - Replace the final tile, $\left[\frac{q_{accept}\#\#}{\#}\right]$ with $\left[\frac{\star q_{accept}\#\star\#}{\#}\right]$ with
  - For every other tile, $\left[\frac{t}{b}\right]$, replace it with $\left[\frac{\star t}{b\star}\right]$
- Now, $\left[\frac{\star\#}{\star\#q_0w\#\star}\right]$ must be the first tile of any solution because it is the only tile that starts and ends with the same symbol.
- We have reduced computational histories for A_{TM} to PCP.
  $\therefore$ PCP is undecidable.

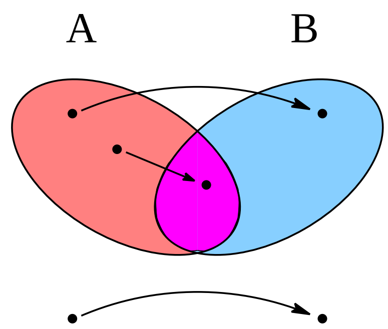
Summarizing Reductions

- Turing computable functions.
- Mapping reductions.
- Using reductions to show non-decidability.
- Examples

Turing computable functions

- $f : \Sigma^* \to \Sigma^*$ be a function over strings of Σ, a finite alphabet.
- f is Turing computable (henceforth, "computable") iff there is some TM that on every input w halts with $f(w)$ (and nothing but $f(w)$) on its tape.
- Examples of computable functions:
  - addition, subtract, multiplication of integers encoded as binary (or unary, or decimal, or any base you like) strings.
  - Sorting a list of strings into lexigraphical order.
  - solution of the Traveling Salesman Problem.
- Examples we've seen in this class
  - Transforming a description of a TM (and possibly its input) into the description of another TM (and possibly its input).
  - Transforing the description of a TM (and possibly its input) into a string describing another kind of machine such as a PDA, CFG, PCP problem, etc.

Mapping Reductions



- Language A is mapping reducible to language B iff there is a computable function, f such that for every w:

$$w \in A \quad \Leftrightarrow \quad f(w) \in B$$

- We write $A \leq_M B$ to indicate that A is mapping reducible to B.
- Mapping reducibility is a reflexive and transitive relation:

$$A \leq_M A$$
$$(A \leq_M B) \wedge (B \leq_M C) \quad \Rightarrow \quad A \leq_M C$$

Mapping and Decidability

- If $A \leq_M B$ and B is Turing decidable, then A is decidable.
  - Likewise if B is Turing recognizable so is A.
  - And so on for co-recognizable, and any other complexity class you want to name.
- If $A \leq_M B$ and A is not Turing decidable, then B is not Turing decidable either.

Mapping Examples

- We've shown $\overline{A_{TM}} \leq_M E_{TM}$ to show that E_{TM} is undecidable (Oct. 31).
- We've shown $A_{TM} \leq_M REGULAR$ and $\overline{A_{TM}} \leq_M REGULAR$ to show that $REGULAR$ is undecidable (in fact it is neither Turing recognizable nor Turing co-recognizable) (Nov. 7).
- We've shown $\overline{A_{TM}} \leq_M E_{LBA}$ (using computational histories) to show that E_{LBA} is undecidable (Nov. 10).
- Let $CFALL = \{G \mid G$ describes a CFG and $L(G) = \Sigma^*\}$. We've shown $\overline{A_{TM}} \leq_M CFALL$ (using computational histories) to show that $CFALL$ is undecidable (Nov. 10).
- We've shown $A_{TM} \leq_M PCP$ (using computational histories) to show that the Post Correspondence Problem is undecidable (today).

This coming week (and beyond)

- Reading
  - Today: Sipser 5.3
  - Nov. 14 (Friday): Sipser 7.1
  - Nov. 17 (Monday): Sipser 7.2
  - Nov. 19 (A week from today): Tutorial by Brad Bingham
- Homework
  - Nov. 14 (Friday): HW 10 goes out.
  - Nov. 17 (Monday): HW 9 due.