CS 483 - Introduction to the Theory of Computation

Lecture 16

More Undecidable Languages

Recall that Turing machine M accepts w if and only if there exists a sequence of configurations $C_0, C_1, C_2, \ldots, C_t$ (called computation history) such that:

- C_0 is the initial configuration $q_0 w$
- C_t is an accepting configuration (contains q_{acc})
- C_i yields C_{i+1}: if $C_i = \cdots abqcd \cdots$ then
  - $C_{i+1} = \cdots aq'bc'd \cdots$, if $\delta(q, c) = (q', c', \leftarrow)$
  - $C_{i+1} = \cdots abc'q'd \cdots$, if $\delta(q, c) = (q', c', \rightarrow)$

More Undecidable Languages

Theorem. $\mathrm{ALL}_{\mathrm{PDA}} = \{ M \mid M \text{ is a PDA}, L(M) = \Sigma^* \}$ is **coRE**-complete.

More Undecidable Languages

Theorem. $\mathrm{ALL}_{\mathrm{PDA}} = \{ M \mid M \text{ is a PDA}, L(M) = \Sigma^* \}$ is **coRE**-complete.

Proof. $\mathrm{ALL}_{\mathrm{PDA}} \in$ **coRE** because

$$\overline{\mathrm{ALL}_{\mathrm{PDA}}} = \{ M \mid \exists w \in \Sigma^*, w \notin L(M) \}$$

while A_{PDA} is decidable.

To prove **coRE**-hardness, we will show $\overline{A_{\mathrm{TM}}} \leq_{\mathrm{m}} \mathrm{ALL}_{\mathrm{PDA}}$.

More Undecidable Languages

Let $L_{M,w}$ be the language:

$$\{ C_0 \# C_1^R \# C_2 \# C_3^R \cdots \# C_t \mid C_0, \ldots, C_t \text{ is not an accepting history of } M(w) \}$$

which can be computed by a PDA $M_{M,w}$ that nondeterministically accepts if any of the following conditions holds:

- C_i is not a valid configuration for some i
- $C_0 \neq q_0 w$
- C_t does not contain q_{acc}
- C_i does not yield C_{i+1} for some i

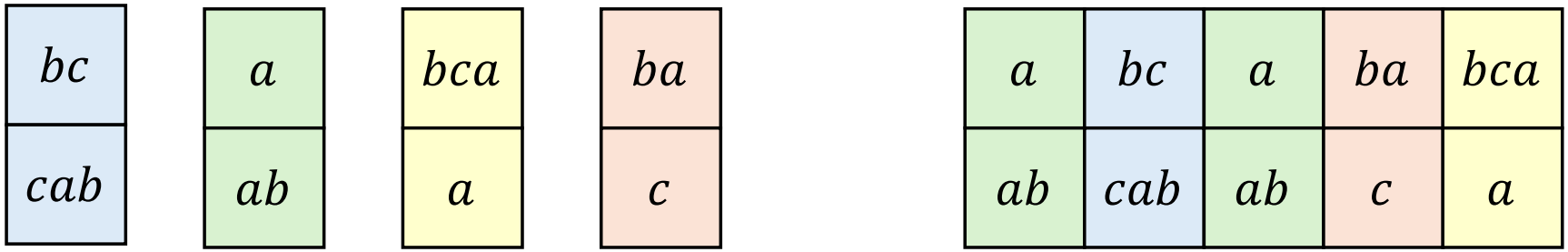
Then $(M, w) \in \overline{A_{\text{TM}}} \Leftrightarrow M_{M,w} \in \text{ALL}_{\text{PDA}}$.

Post Correspondence Problem

The input is a list of pairs $(x_1, y_1), \ldots, (x_n, y_n)$, each $x_k, y_k \in \Sigma^*$.

Question: Are there $m \geq 1$ and $1 \leq k_1, \ldots, k_m \leq n$ such that

$$x_{k_1} \cdots x_{k_m} = y_{k_1} \cdots y_{k_m}?$$

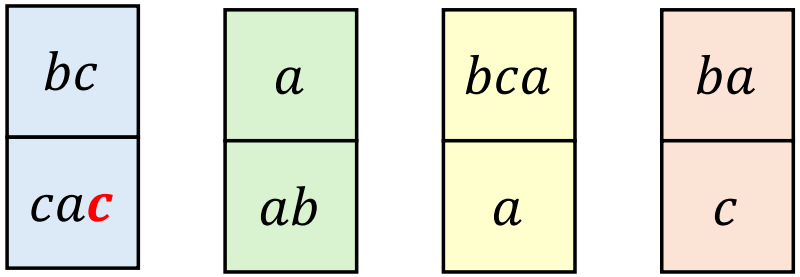


Post Correspondence Problem

The input is a list of pairs $(x_1, y_1), \ldots, (x_n, y_n)$, each $x_k, y_k \in \Sigma^*$.

Question: Are there $m \geq 1$ and $1 \leq k_1, \ldots, k_m \leq n$ such that

$$x_{k_1} \cdots x_{k_m} = y_{k_1} \cdots y_{k_m}?$$

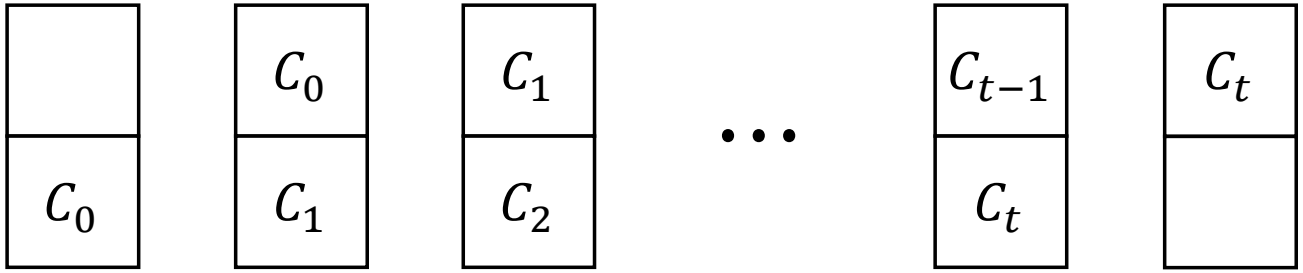


?

Post Correspondence Problem

Theorem. The Post Correspondence Problem (PCP) is **RE**-complete.

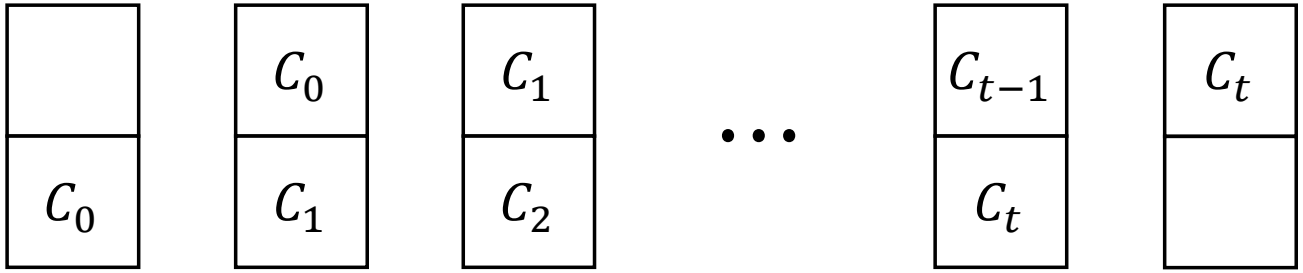
We will show that $A_{\mathrm{TM}} \leq_{\mathrm{m}} \mathrm{PCP}$. Intuitively, in $f(M, w)$ we want pairs:



Post Correspondence Problem

Theorem. The Post Correspondence Problem (PCP) is **RE**-complete.

We will show that $A_{\mathrm{TM}} \leq_{\mathrm{m}} \mathrm{PCP}$. Intuitively, in $f(M, w)$ we want pairs:



Problems: - The number of pairs (C, C') that C yields C' is not finite.

- Need to ensure every match corresponds to a history.

Post Correspondence Problem

Transitions:

bqc
$q'bc'$

if $\delta(q,c) = (q', c', \leftarrow)$

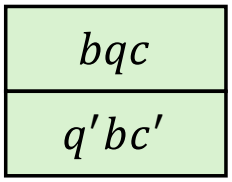
bqc
$bc'q'$

if $\delta(q,c) = (q', c', \rightarrow)$

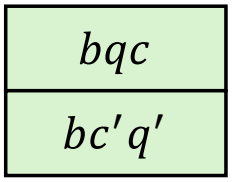
for every $q \in Q,\ b, c \in \Gamma$

Post Correspondence Problem

Transitions:



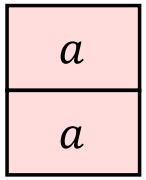
if $\delta(q,c) = (q', c', \leftarrow)$



if $\delta(q,c) = (q', c', \rightarrow)$

for every $q \in Q,\ b, c \in \Gamma$

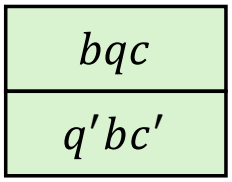
Untouched cells:



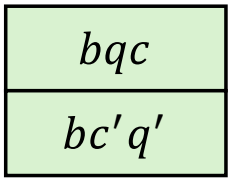
for every $a \in \Gamma$,

Post Correspondence Problem

Transitions:



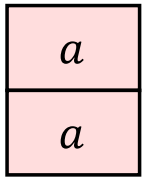
if $\delta(q,c) = (q',c',\leftarrow)$



if $\delta(q,c) = (q',c',\rightarrow)$

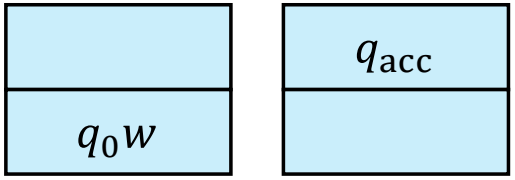
for every $q \in Q,\ b,c \in \Gamma$

Untouched cells:



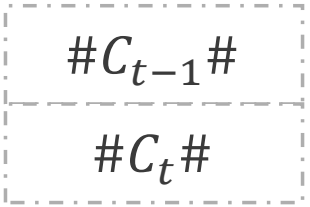
for every $a \in \Gamma$,

Initial and accepting configuration:

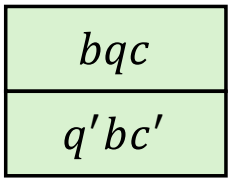


(assume that M clears the tape when accepting)

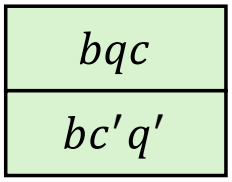
Post Correspondence Problem



Transitions:

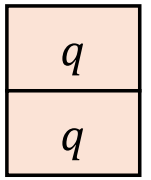


if $\delta(q,c) = (q', c', \leftarrow)$

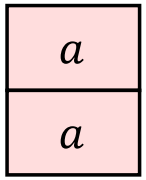


if $\delta(q,c) = (q', c', \rightarrow)$

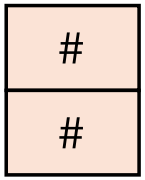
for every $q \in Q,\ b, c \in \Gamma$

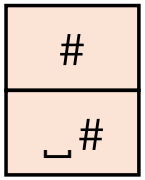


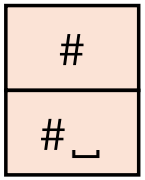
Untouched cells:

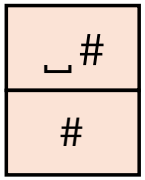


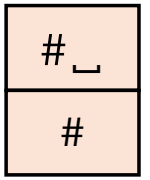
for every $a \in \Gamma$,



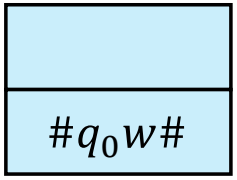


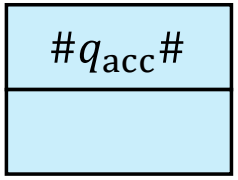






Initial and accepting configuration:





(assume that M clears the tape when accepting)

Post Correspondence Problem

Transitions:

bqc
$q'bc'$

if $\delta(q,c) = (q', c', \leftarrow)$

bqc
$bc'q'$

if $\delta(q,c) = (q', c', \rightarrow)$

for every $q \in Q,\ b, c \in \Gamma$

q
q

Untouched cells:

$!a$
$a!$

for every $a \in \Gamma$,

#
#

#
$\sqcup$#

#
#$\sqcup$

$\sqcup$#
#

#$\sqcup$
#

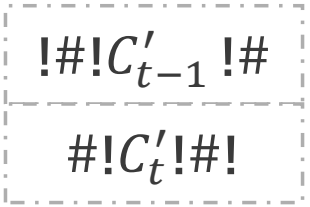
Initial and accepting configuration:

$\#q_0w\#$

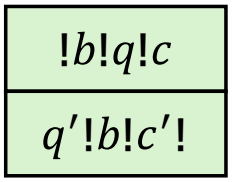
$\#q_{\text{acc}}\#$

(assume that M clears the tape when accepting)

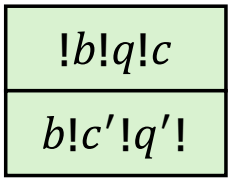
Post Correspondence Problem



Transitions:

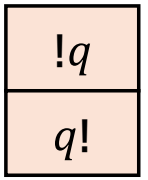


if $\delta(q,c) = (q', c', \leftarrow)$

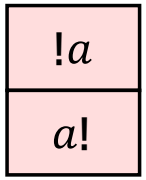


if $\delta(q,c) = (q', c', \rightarrow)$

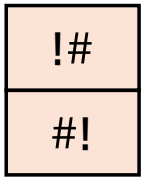
for every $q \in Q,\ b, c \in \Gamma$

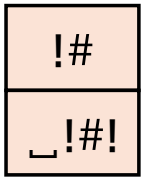


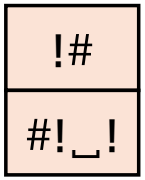
Untouched cells:

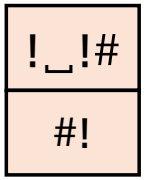


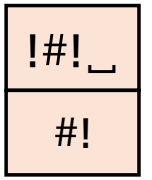
for every $a \in \Gamma$,



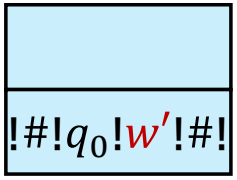


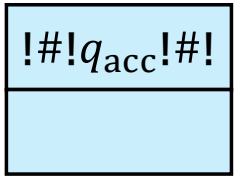






Initial and accepting configuration:





(assume that M clears the tape when accepting)

Post Correspondence Problem

If $M(w) = 1$, there exists a match that looks like:

	$!\#!C_0'!\#$	$!\#!C_1'!\#$	…	$!\#!C_{t-1}'!\#$	$!\#!C_t'!\#!$
$!\#!C_0'!\#!$	$\#!C_1'!\#!$	$\#!C_2'!\#!$		$\#!C_t'!\#!$	

and every match must look like one above which implies $M(w) = 1$.

Therefore $A_{\mathrm{TM}} \leq_{\mathrm{m}} \mathrm{PCP}$.

General Grammars

Recall that in general, a grammar $G = (N, \Sigma, P, S)$ have production rules

$$\alpha \to \beta$$

where $\alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^*$ and $\beta \in (N \cup \Sigma)^*$.

General Grammars

Recall that in general, a grammar $G = (N, \Sigma, P, S)$ have production rules

$$\alpha \to \beta$$

where $\alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^*$ and $\beta \in (N \cup \Sigma)^*$.

Lemma. Given a Turing machine M, there is a grammar G_M such that $L(G_M) = L(M)$, and the map $M \to G_M$ is computable.

General Grammars

Recall that in general, a grammar $G = (N, \Sigma, P, S)$ have production rules

$$\alpha \to \beta$$

where $\alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^*$ and $\beta \in (N \cup \Sigma)^*$.

Lemma. Given a Turing machine M, there is a grammar G_M such that $L(G_M) = L(M)$, and the map $M \to G_M$ is computable.

Corollary 1. $L \in$ **RE** if and only if L is generated by a grammar.

Corollary 2. $A_{\mathrm{G}} = \{ (G, w) \mid \text{grammar } G \text{ generates } w \}$ is **RE**-complete.

General Grammars

Let $G_M = (N, \Sigma, P, S)$ where $N = Q \cup \{S, \#\} \cup (\Gamma \setminus \Sigma)$, and P contains:

$S \to \#q_{\text{acc}}\#$

$q'bc \to bqc$ if $\delta(q, c) = (q', c', \leftarrow)$

$bc'q' \to bqc$ if $\delta(q, c) = (q', c', \rightarrow)$

$q_0 \to \varepsilon$

$\# \to \#\textvisiblespace \mid \textvisiblespace\# \mid \varepsilon$

$\#\textvisiblespace \to \#$

$\textvisiblespace\# \to \#$

Chomsky Hierarchy

