CMSC 28100

Introduction to Complexity Theory

Spring 2025
Instructor: William Hoza

Which problems

can be solved

through computation?

Which languages are decidable?

The rejection problem is undecidable

- $\text{REJECT} = \{\langle M, w \rangle : M \text{ is a Turing machine that rejects } w\}$

Theorem: REJECT is undecidable.

Proof that REJECT is undecidable

- Assume for the sake of contradiction that there is some Turing machine R that decides REJECT
- Let's construct a new TM S that decides $\mathrm{SELF\text{-}REJECTORS}$

S

Given the input $\langle M \rangle$:

1. "Copy and paste" to construct the string $\langle M, \langle M \rangle \rangle$
2. Simulate R on $\langle M, \langle M \rangle \rangle$
3. If R accepts, accept. If R rejects, reject.

- If $\langle M \rangle \in \mathrm{SELF\text{-}REJECTORS}$, then R accepts $\langle M, \langle M \rangle \rangle$, and therefore S accepts $\langle M \rangle$ ✔
- If $\langle M \rangle \notin \mathrm{SELF\text{-}REJECTORS}$, then R rejects $\langle M, \langle M \rangle \rangle$, and therefore S rejects $\langle M \rangle$ ✔

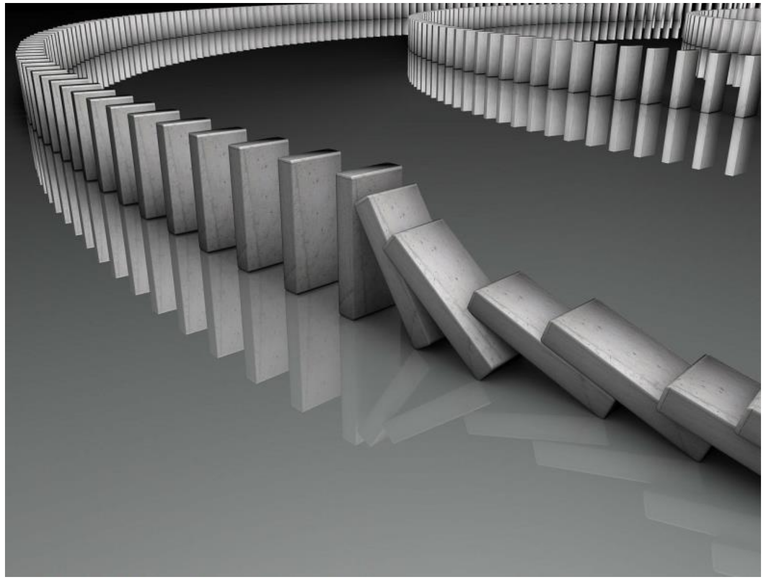
Reductions

- Amazing thing about reductions: The existence of one algorithm implies the non-existence of another!
- Our goal was to prove that REJECT is undecidable
- Our strategy was to design an algorithm for deciding $\mathrm{SELF\text{-}REJECTORS}$! (using a hypothetical Turing machine that decides REJECT)

Note on standards of rigor

- Going forward, when we want to construct a Turing machine (e.g., for a reduction), we will simply describe what it does in plain English
  - As if we were giving instructions to a human being
  - Each plain English description can be formalized as a Turing machine, but this is tedious
  - You should follow this convention on Exercise 9 and beyond

Undecidability



- Now we have two examples of undecidable languages
- SELF-REJECTORS and REJECT
- Next, we will see an example of an undecidable language that (seemingly) isn't about Turing machines

Post's Correspondence Problem

- **Given:** Two sequences of strings $b_1, \dots, b_k, t_1, \dots, t_k \in \Gamma^*$ for some alphabet Γ
- **Goal:** Determine whether there exists a sequence of indices $i_1, \dots, i_n$, where $n \geq 1$, such that

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

Post’s Correspondence Problem

- Helpful picture: We are given a set of “dominos”

$$\boxed{\begin{array}{c} t_1 \\ b_1 \end{array}} \quad \boxed{\begin{array}{c} t_2 \\ b_2 \end{array}} \quad \boxed{\begin{array}{c} t_3 \\ b_3 \end{array}} \quad \cdots \quad \boxed{\begin{array}{c} t_k \\ b_k \end{array}}$$

- Goal: Determine whether it is possible to generate a “match”

$$\boxed{\begin{array}{c} t_{i_1} \\ b_{i_1} \end{array}}\boxed{\begin{array}{c} t_{i_2} \\ b_{i_2} \end{array}}\boxed{\begin{array}{c} t_{i_3} \\ b_{i_3} \end{array}}\boxed{\begin{array}{c} t_{i_4} \\ b_{i_4} \end{array}}\boxed{\begin{array}{c} t_{i_5} \\ b_{i_5} \end{array}} \quad \cdots \quad \boxed{\begin{array}{c} t_{i_n} \\ b_{i_n} \end{array}}$$

in which the sequence of symbols on top equals the sequence of symbols on the bottom

- Using the same domino multiple times is permitted

Post's Correspondence Problem: Example 1

- Suppose we are given

$$\left[\frac{0}{1}\right] \quad \left[\frac{1}{0}\right] \quad \left[\frac{11}{\epsilon}\right] \quad \left[\frac{\epsilon}{00}\right]$$

- This is a YES case. Match:

$$\left[\frac{11}{\epsilon}\right]\left[\frac{0}{1}\right]\left[\frac{0}{1}\right]\left[\frac{\epsilon}{00}\right] \quad \begin{matrix} \leftarrow 1100 \\ \leftarrow 1100 \end{matrix}$$

Post's Correspondence Problem: Example 2

- Suppose we are given

#	#$##	$
#$	$#	#$

- This is a NO case
- **Proof:** A match would have to start with $\begin{bmatrix}\#\\ \#\$\end{bmatrix}$ …
- But a sequence containing $\begin{bmatrix}\#\\ \#\$\end{bmatrix}$ cannot be a match, because such a sequence has more \$ symbols on the bottom than on the top

Post’s Correspondence Problem is undecidable

- **Post’s correspondence problem, formulated as a language:**

$$\mathrm{PCP} = \{\langle t_1, \dots, t_k, b_1, \dots, b_k \rangle : \exists i_1, \dots, i_n \text{ such that } t_{i_1} \cdots t_{i_n} = b_{i_1} \cdots b_{i_n}\}$$

Theorem: PCP is undecidable

- Proof on the upcoming 18 slides. Outline:
  - Step 1: Reduce REJECT to a modified version (“MPCP”)
  - Step 2: Reduce MPCP to PCP

Modified PCP

$$\text{MPCP} = \{\langle t_1, \ldots, t_k, b_1, \ldots, b_k \rangle : \exists i_1, \ldots, i_n \text{ such that } t_1 t_{i_1} \cdots t_{i_n} = b_1 b_{i_1} \cdots b_{i_n}\}$$

- The difference between PCP and MPCP: In MPCP, matches must start with the first domino
- We'll use a double outline to indicate the special first domino: $\boxed{\begin{matrix} t_1 \\ b_1 \end{matrix}}$

Lemma: MPCP is undecidable

Proof that MPCP is undecidable

- Assume there is a TM P that decides MPCP

- Let's construct a new TM R that decides REJECT

R

Given $\langle M, w \rangle$:

1. Construct dominos $t_1, \ldots, t_k, b_1, \ldots b_k$ based on M and w

   (details on upcoming slides)

2. Simulate P on $\langle t_1, \ldots, t_k, b_1, \ldots, b_k \rangle$

3. If P accepts, accept. If P rejects, reject.

Reducing $\mathbf{REJECT}$ to $\mathbf{MPCP}$

- We are given $\langle M, w \rangle$, where

$$M = (Q, q_0, q_{\text{accept}}, q_{\text{reject}}, \Sigma, \sqcup, \delta)$$

- Our job is to produce a sequence of dominos
- Plan: Produce dominos such that constructing a match is equivalent to constructing a rejecting computation history

Reducing REJECT

- Let C_0 be the initial configurat
- $\boxed{\dfrac{\epsilon}{(C_0)}}$, $\dfrac{(}{(}$, $\dfrac{)}{)}$, and $\dfrac{(q}{}$

Given $\langle M, w\rangle$, how does one construct these dominos?

A: Simulate M on w. If it accepts, accept; if it rejects, reject

B: Simulate M on w and copy whatever dominos it produces

C: There does not exist an algorithm that computes f

D: Inspect the transition function of M to figure out the dominos

Respond at PollEv.com/whoza or text “whoza” to 22333

- For every $q \in Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}$ and every $b \in \Sigma$:
  - If $\delta(q, b) = (q', b', \mathrm{R})$, we include $\dfrac{qb)}{b'q' \sqcup)}$, and we include $\dfrac{qba}{b'q'a}$ for every $a \in \Sigma$
  - If $\delta(q, b) = (q', b', \mathrm{L})$, we include $\dfrac{(qb}{(q' \sqcup b'}$, and we include $\dfrac{aqb}{q'ab'}$ for every $a \in \Sigma$
- $\dfrac{b}{b}$, $\dfrac{bq_{\text{reject}}}{q_{\text{reject}}}$, and $\dfrac{q_{\text{reject}}b}{q_{\text{reject}}}$ for every $b \in \Sigma$

Proof that MPCP is undecidable

- Assume there is a TM P that decides MPCP
- Let's construct a new TM R that decides REJECT

R

Given $\langle M, w \rangle$:

1. Construct dominos $t_1, \dots, t_k, b_1, \dots b_k$ based on M and w (details on preceding slides)
2. Simulate P on $\langle t_1, \dots, t_k, b_1, \dots, b_k \rangle$
3. If P accepts, accept. If P rejects, reject.

Need to show:

- If M rejects w, then there is a match
- If there is a match, then M rejects w

Domino Feature 1

- **Domino Feature 1:** For every non-halting configuration C of M, there is a sequence of dominos such that the top string is (C) and bottom string is $\left(\mathrm{NEXT}(C)\right)$
- Example:

$($	0	1	$\#$	0	$0q_1 0$	0	$\sqcup$	$)$
$($	0	1	$\#$	0	$q_2 01$	1	$\sqcup$	$)$

- Think of this sequence as one "super-domino" $\dfrac{(C)}{\left(\mathrm{NEXT}(C)\right)}$

If M rejects w, then there is a match

- Let $C_0, \ldots, C_T$ be the rejecting computation history of M on w
- Partial match:

ϵ	(C_0)	(C_1)	$\cdots$	(C_{T-1})
(C_0)	(C_1)	(C_2)		(C_T)

- At this point, we have an extra (C_T) on the bottom

Domino Feature 2

- **Domino Feature 2:** For every rejecting configuration D, there is a sequence of dominos such that the top string is (D) and the bottom string is (D'), where D' is a rejecting configuration* of length $|D| - 1$
  - *Possibly $D' = q_{\text{reject}}$
- Example:

(	0	1	#	0	$0q_{\text{reject}}$	0	⊔	)
(	0	1	#	0	q_{reject}	1	⊔	)

- Think of this sequence as one "super domino" $\dfrac{(D)}{(D')}$

If M rejects w, then there is a match

- We construct a sequence of shorter and shorter rejecting configurations $C_T = D_0, D_1, \ldots, D_n = q_{\text{reject}}$ such that we have a super-domino $\boxed{\begin{matrix}(D_{i-1})\\(D_i)\end{matrix}}$ for every i
- Full match:

ϵ	(C_0)	(C_1)	…	(C_{T-1})	(C_T)	(D_1)	(D_2)	…	(D_{n-1})	(q_{reject})
(C_0)	(C_1)	(C_2)		(C_T)	(D_1)	(D_2)	(D_3)		(q_{reject})	ϵ

Proof that MPCP is undecidable

- Assume there is a TM P that decides MPCP
- Let's construct a new TM R that decides REJECT

R

Given $\langle M, w \rangle$:

1. Construct dominos $t_1, \ldots, t_k, b_1, \ldots b_k$ based on M and w (details on preceding slides)
2. Simulate P on $\langle t_1, \ldots, t_k, b_1, \ldots, b_k \rangle$
3. If P accepts, accept. If P rejects, reject.

Need to show:

- If M rejects w, then there is a match ✔
- If there is a match, then M rejects w

Domino Features 3 and 4

- **Domino Feature 3:** If C is a non-halting configuration, then every sequence of dominos in which the top string starts with (C) must begin with the following super-domino: $\boxed{\begin{matrix}(C)\\ (\mathrm{NEXT}(C))\end{matrix}}$
- **Domino Feature 4:** If C is an accepting configuration, then there does not exist a sequence of dominos in which the top string begins with (C)

If there is a match, then M rejects w

- Assume there is a match

- By Domino Feature 3, it must have the form

ϵ	(C_0)	(C_1)	$\cdots$	(C_{T-1})	$(C_T)x$
(C_0)	(C_1)	(C_2)		(C_T)	x

where C_T is a halting configuration and $x \in \Gamma^*$

- By Domino Feature 4, C_T cannot be accepting, so it must be rejecting

Proof that MPCP is undecidable

- Assume there is a TM P that decides MPCP
- Let's construct a new TM R that decides REJECT

R

Given $\langle M, w \rangle$:

1. Construct dominos $t_1, \ldots, t_k, b_1, \ldots b_k$ based on M and w (details on preceding slides)
2. Simulate P on $\langle t_1, \ldots, t_k, b_1, \ldots, b_k \rangle$
3. If P accepts, accept. If P rejects, reject.

Need to show:

- If M rejects w, then there is a match
- If there is a match, then M rejects w

Post's Correspondence Problem is undecidable

- **Post's correspondence problem, formulated as a language:**

$$\text{PCP} = \{\langle t_1, \dots, t_k, b_1, \dots, b_k \rangle : \exists i_1, \dots, i_n \text{ such that } t_{i_1} \cdots t_{i_n} = b_{i_1} \cdots b_{i_n}\}$$

Theorem: PCP is undecidable

- Proof outline:
  - Step 1: Reduce REJECT to a modified version ("MPCP") ✔
  - Step 2: Reduce MPCP to PCP