CMSC 28100

Introduction to Complexity Theory

Spring 2025
Instructor: William Hoza

Post’s Correspondence Problem

- **Given**: a set of “dominos”

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

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

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

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 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 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: $\begin{bmatrix} t_1 \\ b_1 \end{bmatrix}$

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 $t_1, \dots, t_k, b_1, \dots b_k$ based on M and w

   (details on next slide)

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.

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

- Let C_0 be the initial configuration of M on w. Dominos:
- $\boxed{\boxed{\dfrac{\epsilon}{(C_0)}}}$, $\boxed{\dfrac{(}{(}}$, $\boxed{\dfrac{)}{)}}$, and $\boxed{\dfrac{(q_{\text{reject}})}{\epsilon}}$
- 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 $\boxed{\dfrac{qb)}{b'q' \sqcup)}}$, and we include $\boxed{\dfrac{qba}{b'q'a}}$ for every $a \in \Sigma$
  - If $\delta(q, b) = (q', b', \mathrm{L})$, we include $\boxed{\dfrac{(qb}{(q' \sqcup b'}}$, and we include $\boxed{\dfrac{aqb}{q'ab'}}$ for every $a \in \Sigma$
- $\boxed{\dfrac{b}{b}}$, $\boxed{\dfrac{bq_{\text{reject}}}{q_{\text{reject}}}}$, and $\boxed{\dfrac{q_{\text{reject}}b}{q_{\text{reject}}}}$ 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 $t_1, \ldots, t_k, b_1, \ldots b_k$ based on M and w (details on previous slide)
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.

Last class:

- 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:**

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

Theorem: PCP is undecidable

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

Proof that PCP is undecidable

- Assume there is a TM P that decides PCP
- Let’s construct a new TM M that decides MPCP
- For a string $u = u_1 u_2 \dots u_n$, define $\overline{u} = u_1 \star u_2 \star \cdots \star u_n$

M:

Given $\boxed{\left[\frac{t_1}{b_1}\right]}$ $\left[\frac{t_2}{b_2}\right]$ $\left[\frac{t_3}{b_3}\right]$ … $\left[\frac{t_k}{b_k}\right]$:

1. Simulate P on $\left[\frac{\star\,\overline{t_1}}{\star\,\overline{b_1}\,\star}\right]$ $\left[\frac{\star\,\overline{t_1}}{\overline{b_1}\,\star}\right]$ $\left[\frac{\star\,\overline{t_2}}{\overline{b_2}\,\star}\right]$ $\left[\frac{\star\,\overline{t_3}}{\overline{b_3}\,\star}\right]$ … $\left[\frac{\star\,\overline{t_k}}{\overline{b_k}\,\star}\right]$ $\left[\frac{\star}{\epsilon}\right]$
2. If P accepts, accept. If P rejects, reject.

Given $\boxed{\boxed{\begin{matrix} t_1 \\ b_1 \end{matrix}}}$ $\boxed{\begin{matrix} t_2 \\ b_2 \end{matrix}}$ $\boxed{\begin{matrix} t_3 \\ b_3 \end{matrix}}$ … $\boxed{\begin{matrix} t_k \\ b_k \end{matrix}}$:

1. Simulate P on $\boxed{\begin{matrix} \star\,\overline{t_1} \\ \star\,\overline{b_1}\,\star \end{matrix}}$ $\boxed{\begin{matrix} \star\,\overline{t_1} \\ \overline{b_1}\,\star \end{matrix}}$ $\boxed{\begin{matrix} \star\,\overline{t_2} \\ \overline{b_2}\,\star \end{matrix}}$ $\boxed{\begin{matrix} \star\,\overline{t_3} \\ \overline{b_3}\,\star \end{matrix}}$ … $\boxed{\begin{matrix} \star\,\overline{t_k} \\ \overline{b_k}\,\star \end{matrix}}$ $\boxed{\begin{matrix} \star \\ \epsilon \end{matrix}}$
2. If P accepts, accept. If P rejects, reject.

- Suppose the MPCP instance has a match: $\boxed{\boxed{\begin{matrix} t_1 \\ b_1 \end{matrix}}}\boxed{\begin{matrix} t_{i_1} \\ b_{i_1} \end{matrix}}\boxed{\begin{matrix} t_{i_2} \\ b_{i_2} \end{matrix}}\boxed{\begin{matrix} t_{i_3} \\ b_{i_3} \end{matrix}}\boxed{\begin{matrix} t_{i_4} \\ b_{i_4} \end{matrix}}$ … $\boxed{\begin{matrix} t_{i_n} \\ b_{i_n} \end{matrix}}$

- Then the PCP instance also has a match: $\boxed{\begin{matrix} \star\,\overline{t_1} \\ \star\,\overline{b_1}\,\star \end{matrix}}\boxed{\begin{matrix} \star\,\overline{t_{i_1}} \\ \overline{b_{i_1}}\,\star \end{matrix}}\boxed{\begin{matrix} \star\,\overline{t_{i_2}} \\ \overline{b_{i_2}}\,\star \end{matrix}}$ … $\boxed{\begin{matrix} \star\,\overline{t_{i_n}} \\ \overline{b_{i_n}}\,\star \end{matrix}}\boxed{\begin{matrix} \star \\ \epsilon \end{matrix}}$

Given $\boxed{\left[\frac{t_1}{b_1}\right]}$ $\left[\frac{t_2}{b_2}\right]$ $\left[\frac{t_3}{b_3}\right]$ $\cdots$ $\left[\frac{t_k}{b_k}\right]$:

1. Simulate P on $\left[\frac{\star\,\overline{t_1}}{\star\,\overline{b_1}\,\star}\right]$ $\left[\frac{\star\,\overline{t_1}}{\overline{b_1}\,\star}\right]$ $\left[\frac{\star\,\overline{t_2}}{\overline{b_2}\,\star}\right]$ $\left[\frac{\star\,\overline{t_3}}{\overline{b_3}\,\star}\right]$ $\cdots$ $\left[\frac{\star\,\overline{t_k}}{\overline{b_k}\,\star}\right]$ $\left[\frac{\star}{\epsilon}\right]$
2. If P accepts, accept. If P rejects, reject.

- Conversely, suppose the PCP instance has a match
- Must start with $\left[\frac{\star\,\overline{t_1}}{\star\,\overline{b_1}\,\star}\right]$, because that's the only domino in which the top string and bottom string start with the same symbol
- Delete all $\star$ symbols $\Rightarrow$ MPCP match

Post's Correspondence Problem is undecidable

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

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

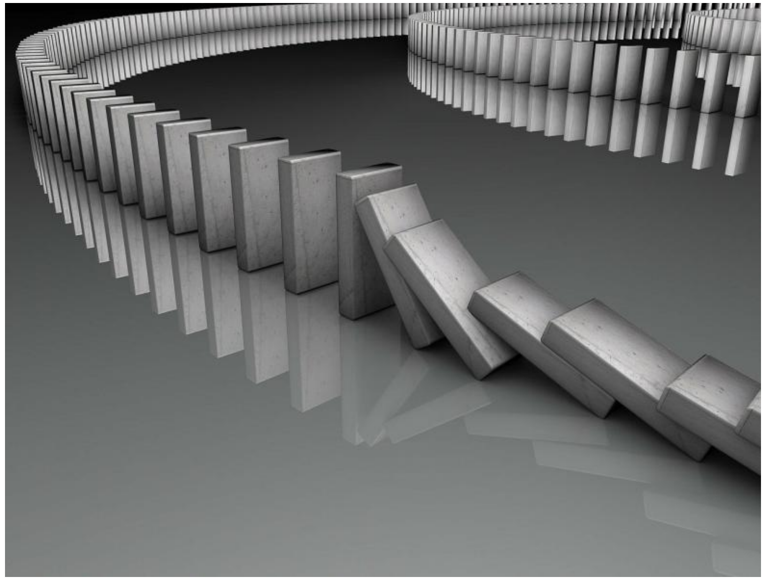
Theorem: PCP is undecidable

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

Post's Correspondence Problem: Recap

- Post's Correspondence Problem seems like "just a domino puzzle"
- However, we showed how to build a computer out of dominos!
- PCP was secretly a problem about Turing machines all along!

Undecidability



- Known undecidable languages:
  - SELF-REJECTORS
  - REJECT
  - PCP and MPCP
- Next: The "halting problem"

The halting problem

- **Informal problem statement:** Given a Turing machine M and an input w, determine whether M halts on w.
- **The same problem, formulated as a language:**

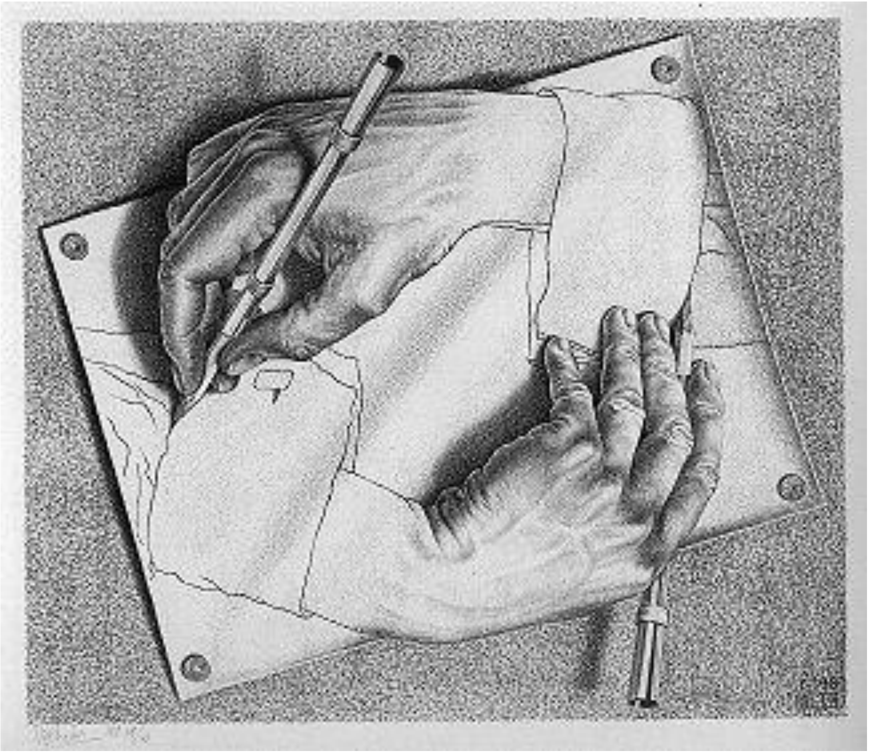
$$\mathrm{HALT} = \{\langle M, w\rangle : M \text{ is a Turing machine that halts on input } w\}$$

- It's the problem of identifying bugs in someone else's code! 🐞

Theorem: HALT is undecidable

Code as data II

- The proof that HALT is undecidable involves Turing machines constructing Turing machines
- Turing machines can both read and write descriptions $\langle M \rangle$ where M is a Turing machine



"Drawing Hands."
(1948 lithograph by M. C. Escher)

Proof that HALT is undecidable

- Assume there is a TM H that decides HALT

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

R:

Given $\langle M, w \rangle$:

1. Construct $\langle M' \rangle$, where M' is the following TM:

   M':

   Given x:
   1. Simulate M on x
   2. If M rejects, reject. If M accepts, loop.

2. Simulate H on $\langle M', w \rangle$
3. If H accepts, accept. If H rejects, reject.

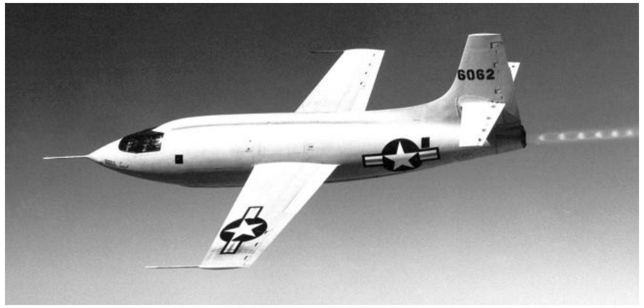
If M rejects w…

- Then M' rejects w
- Therefore, H accepts $\langle M', w \rangle$
- Therefore, R accepts $\langle M, w \rangle$

If M does not reject w…

- Then M' loops on w
- Therefore, H rejects $\langle M', w \rangle$
- Therefore, R rejects $\langle M, w \rangle$

The Church-Turing thesis, revisited



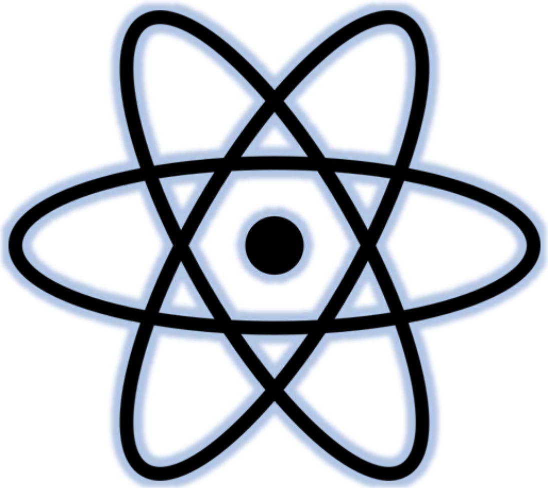
- Let $Y \subseteq \{0,1\}^*$

Church-Turing Thesis:

There exists an "algorithm" / "procedure" for figuring out whether a given string is in Y if and only if there exists a Turing machine that decides Y.

- Computation is an intuitive notion rooted in everyday human experience
- Could it be possible to solve the halting problem using science and technology?

Hypercomputers



- A hypercomputer is a hypothetical device that can solve some computational problem that cannot be solved by Turing machines, such as the halting problem
- Could it be possible to build a hypercomputer?
- We could try using quantum physics, antimatter, black holes, dark energy, superconductors, wormholes, closed timelike curves, …

The Physical Church-Turing Thesis

- Let Y be a language

Physical Church-Turing Thesis:

It is physically possible to build a device that decides Y if and only if there exists a Turing machine that decides Y.

The Physical Church-Turing Thesis

- The standard Church-Turing thesis is a philosophical statement
- The Physical Church-Turing thesis is a scientific law
- Conceivably, it could be disproven by future discoveries… but that would be very surprising
- Analogy: Second Law of Thermodynamics
- Analogy: Cannot travel faster than the speed of light

Which problems

can be solved

through computation?

Which languages are decidable?

Some more undecidable problems

- We have seen several interesting examples of undecidable languages
  - SELF-REJECTORS, REJECT, PCP, MPCP, HALT
- I'll describe a few more examples
- Each can be proven undecidable via reduction from HALT
- But we will not do the proofs
- (This material will not be on exercises or exams)

Hilbert's 10th problem

- **Informal problem statement:** Given a polynomial equation with integer coefficients such as

$$x^2 + 3xz + y^3 + z^2x^2 = 4xy^2 + 6yz + 2,$$

determine whether there is an integer solution

- **As a language:** $\text{HILBERT10} = \{\langle p, q\rangle : \exists \vec{x} \ \text{ such that } \ p(\vec{x}) = q(\vec{x})\}$

Theorem: HILBERT10 is undecidable

Matrix mortality

- Let M_0 and M_1 be $n \times n$ matrices with integer entries
- We say that (M_0, M_1) is a mortal pair if there exists $k \in \mathbb{N}$ and there exist $i_1, \ldots, i_k \in \{0, 1\}$ such that $M_{i_1} \cdot M_{i_2} \cdot \cdots \cdot M_{i_k} = 0$ (the all-zeroes matrix).
- Let $\mathrm{MORTAL} = \{\langle M_0, M_1 \rangle : (M_0, M_1) \text{ is a mortal pair}\}$

Theorem: MORTAL is undecidable

Derivatives vs. Integrals

- Recall: Calculus
- Computing derivatives is mechanistic
  - Sum rule $(f+g)' = f' + g'$, product rule $(fg)' = f'g + fg'$, chain rule $(f \circ g)' = (f' \circ g) \cdot g'$, etc.
- In contrast, computing integrals seems to involve creativity
  - u-substitutions, integration by parts, etc.

Elementary functions

- Definition: A function $f: \mathbb{R} \to \mathbb{R}$ is elementary if it can be defined by a formula using addition, multiplication, rational constants, powers, exponentials, logarithms, trigonometric functions, and π
- E.g. $f(x) = x \cdot \sin(x^4) - 3\pi \cdot e^{e^{\sqrt{x}}}$

Integration is undecidable

- Fact: There exist elementary functions that do not have elementary antiderivatives, such as $f(x) = e^{-x^2}$
- Let $\text{INTEGRABLE} = \{\langle f \rangle : f \text{ is an elementary function with an elementary antiderivative}\}$

Theorem: INTEGRABLE is undecidable