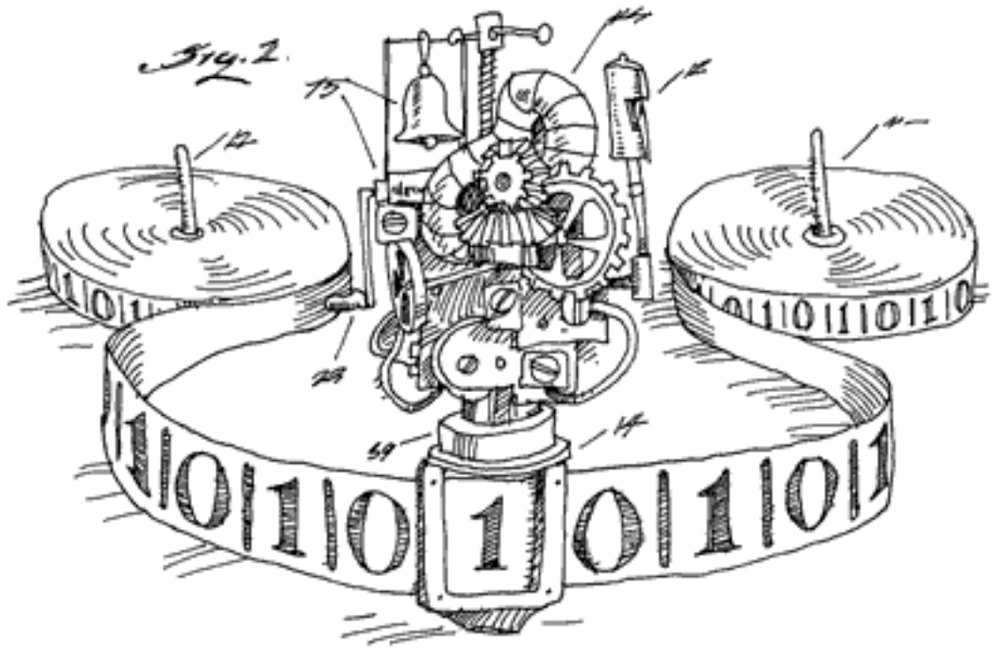
CSC4010: Introduction to Theoretical Computer Science



Lecture 9

Adapted from Sipser's MIT OCW Course on TOC

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

SCHOOL OF DATA SCIENCE
數據科學學院

SDS

TOPICAL SEMINAR SERIES



Speaker

He SUN

Director, Center for Algorithms and Learning Theory
Shenzhen Institute of Advanced Technology, CAS

Associate Director, State Key Laboratory of Biomedical Imaging Science and System

Faster Dynamic Graph Clustering with Hierarchical Graph Contraction

Abstract

We study clustering algorithm for dynamic evolving graphs G_t, in which new edges (and potentially new vertices) are added into a graph, and the underlying cluster structure of the graph gradually changes over time. By designing a hierarchical data structure for dynamically maintaining the cluster structure of G_t we prove that the cluster structure of G_T can be maintained with O(d) amortised update time and O(n_T^{1/d}) amortised query time, for an arbitrary large constant d. We further demonstrate this improvement with experiments on both synthetic and real-world datasets.

The talk is based on joint work with Suranjan De (University of Edinburgh) and Steinar Leanen (Google Zurich). The work appears at NeurIPS 2026.

About the Speaker

Professor He Sun is the Director of Center of Algorithms and Learning Theory, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, and the Associate Director of the State Key Laboratory of Biomedical Imaging Science and System with over 150 researchers. He received his Ph.D. from Fudan University in 2010 and worked at the Max Planck Institute for Informatics, UC Berkeley, University of Bristol, and University of Edinburgh. He is a recipient of the Chinese High-Level Talent Recruitment Program for Overseas Experts and is a fellow selected under the CAS 100 Talents Program (Category A).

He has served as an area chair and PC member of leading international conferences like ICML (2026), NeurIPS (2026), ICLR (2027), and STOC (2026), and as regular reviewer for research councils including ERC, UKRI, and ISF.

Host / Aditi Dudeja

Assistant Professor
School of Data Science
CUHK-Shenzhen

10.15 2026

BEIJING TIME	**11:00** AM– **12:00** NOON
VENUE	**Room 413** **Zhi Xin Building**
LANGUAGE	**English**

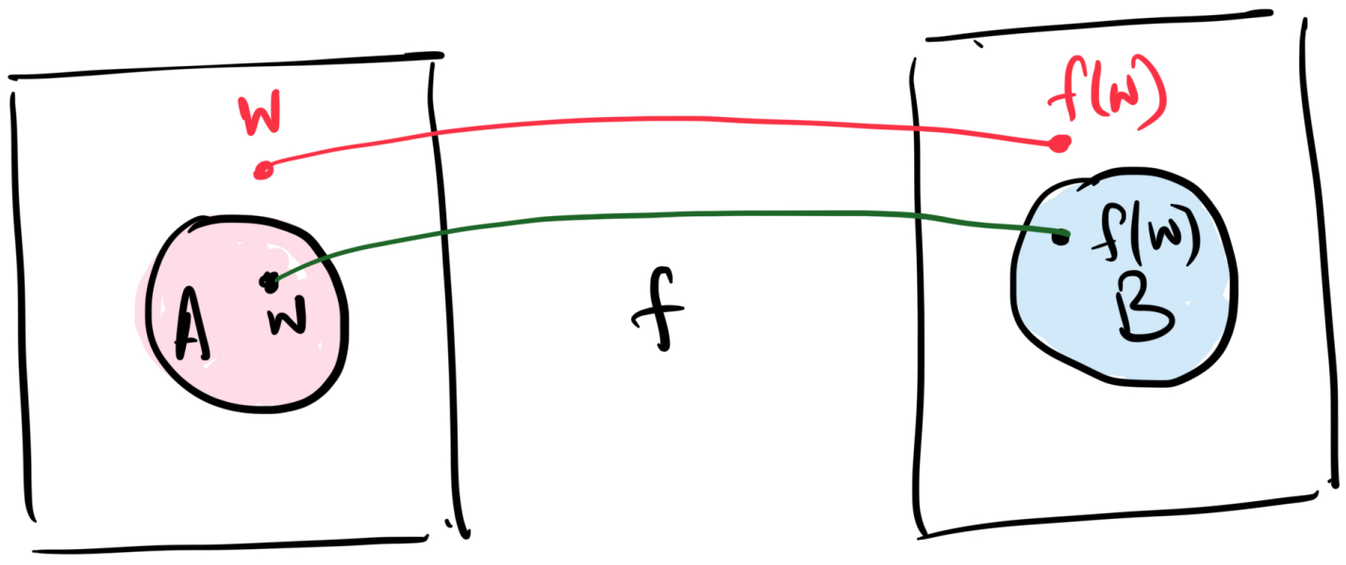
SDS Wechat


Recap

Mapping Reducibility

- **Definition:** A function $f : \Sigma^* \to \Sigma^*$ is said to be **computable** if there exists a Turing Machine F, which on input $w \in \Sigma^*$ terminates with just $f(w)$ written on its tape, for any w .

- **Definition:** A language A is **mapping reducible** to a language B (denoted $A \leq_m B$) if and only if there exists a computable function $f : \Sigma^* \to \Sigma^*$ such that for every $w \in \Sigma^*$, $w \in A \iff f(w) \in B$.



f is called a mapping reduction from A to B .

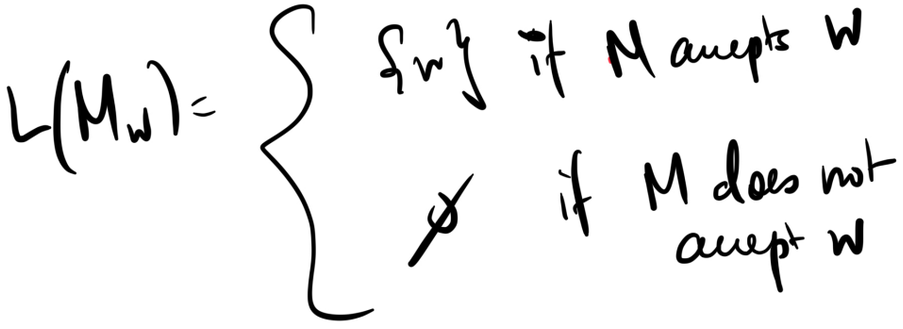
$A_{TM} \leq_m \overline{\text{EMPTY}}_{TM}$

Solution: Given input $\langle M, w \rangle$ for A_{TM}, $f(\langle M, w \rangle) = \langle M_w \rangle$ defined as follows:

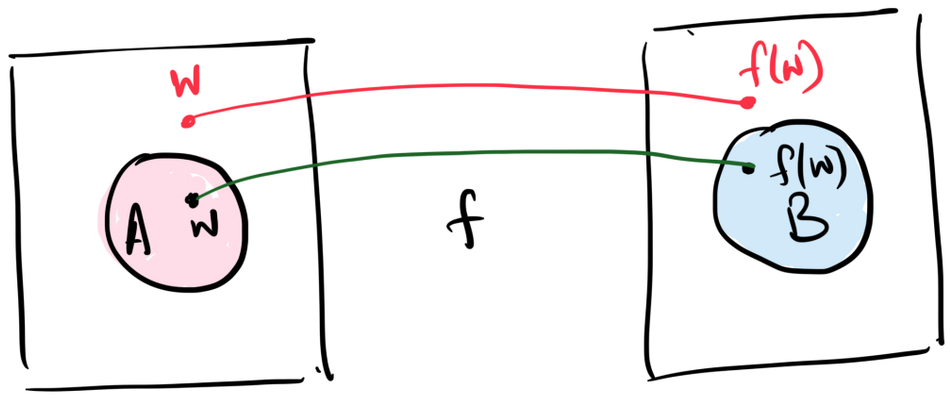
$M_w =$ "On input $x \in \{0,1\}^*$

1. If $x \neq w$, then reject.
2. Simulate M on w .

Claim: $f(\langle M, w \rangle) \in \overline{\text{EMPTY}}_{TM} \iff \langle M, w \rangle \in A_{TM}$.



Mapping Reducibility: Properties



Theorem: If $A \leq_m B$ and B is decidable (recognizable), then A is decidable (respectively, recognizable).

Corollary: If $A \leq_m B$ and A is undecidable (unrecognizable), then B is undecidable (respectively, unrecognizable).

- **A recipe for undecidability/unrecognizability:** To show B is undecidable (unrecognizable), find undecidable (respectively, unrecognizable) A and a mapping reduction from A to B.

Theorem: If $A \leq_m B$ then $\bar{A} \leq_m \bar{B}$.

Using Mapping Reduction for Unrecognizability

EMPTY$_{TM}$ is unrecognizable

- Let EMPTY$_{TM} = \{\langle M \rangle \mid L(M) = \varnothing\}$.

Theorem: EMPTY$_{TM}$ is unrecognizable.

- **Proof:** Proof by contradiction. We will show $\bar{A}_{TM} \leq_m$ EMPTY$_{TM}$.

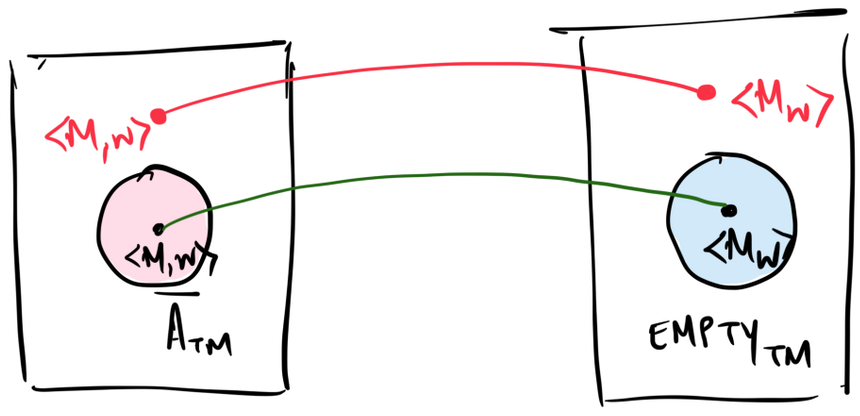
Reduction Function: $f(\langle M, w \rangle) = \langle M_w \rangle$

$M_w =$ "On input $x \in \{0,1\}^*$

1. If $x \neq w$, then reject.
2. Simulate M on w.

$$L(M_w) = \begin{cases} \{w\} & \text{if } M \text{ accepts } w \\ \varnothing & \text{if } M \text{ does not accept } w \end{cases}$$

$f(\langle M, w \rangle) \in$ EMPTY$_{TM}$ $\iff$ $\langle M_w \rangle \in$ EMPTY$_{TM}$
$\iff$ $L(M_w) = \varnothing$
$\iff$ M does not accept w
$\iff$ $\langle M, w \rangle \in \bar{A}_{TM}$



More Examples

EQ_{TM} and $\overline{\text{EQ}}_{TM}$ are unrecognizable

- Let $\text{EQ}_{TM} = \{\langle M_1, M_2\rangle \mid L(M_1) = L(M_2)\}$.

Theorem: EQ_{TM} is unrecognizable.

- **Proof:** Proof by contradiction. We will show $\bar{A}_{TM} \leq_m \text{EQ}_{TM}$.

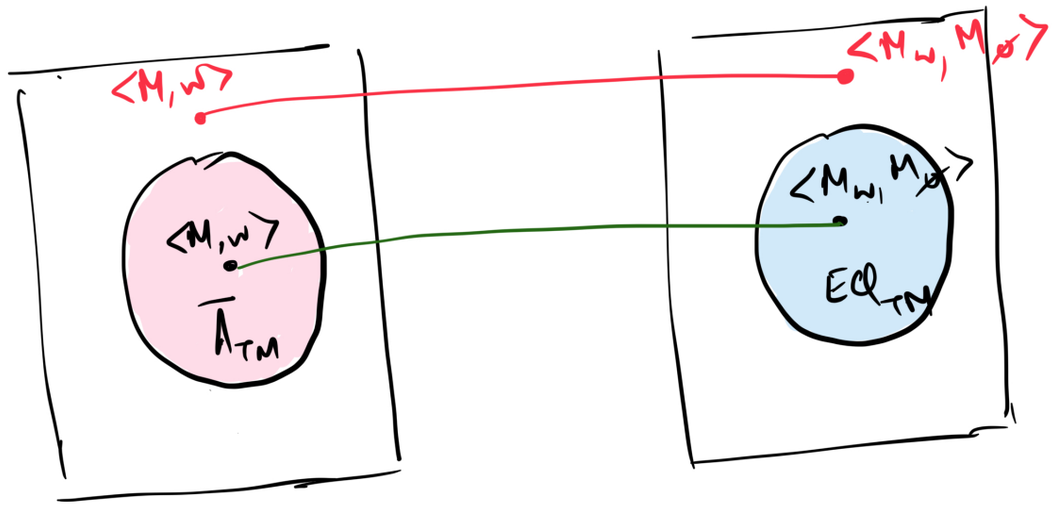
Reduction Function: $f(\langle M, w\rangle) = \langle T_w, M_\emptyset\rangle$

$T_w =$ "On input $x \in \{0,1\}^*$

1. Ignore x .
2. Simulate M on w .

$$L(T_w) = \begin{cases} \{0,1\}^* & \text{if } M \text{ accepts } w \\ \emptyset & \text{if } M \text{ rejects } w \end{cases}$$

$$\langle T_w, M_\emptyset\rangle \in \text{EQ}_{TM} \iff L(T_w) = L(M_\emptyset)$$
$$\iff L(T_w) = \emptyset$$
$$\iff M \text{ does not accept } w$$
$$\iff \langle M, w\rangle \in \bar{A}_{TM}$$



EQ_{TM} and $\overline{\mathrm{EQ}}_{TM}$ are unrecognizable

- Let $\overline{\mathrm{EQ}}_{TM} = \{\langle M_1, M_2\rangle \mid L(M_1) \neq L(M_2)\}$.

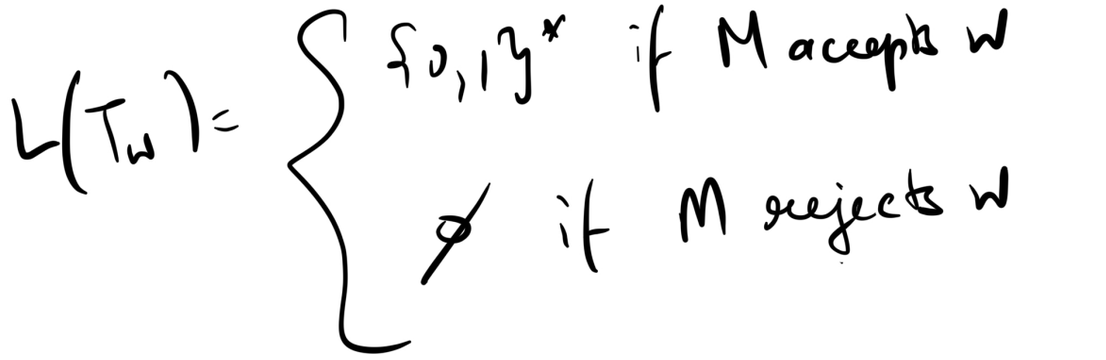
Theorem: $\overline{\mathrm{EQ}}_{TM}$ is unrecognizable.

- **Proof:** Proof by contradiction. We will show $\overline{A}_{TM} \leq_m \overline{\mathrm{EQ}}_{TM}$.

Reduction Function: $f(\langle M, w\rangle) = \langle T_w, M_{acc}\rangle$

$T_w =$ "On input $x \in \{0,1\}^*$

1. Ignore x.
2. Simulate M on w.

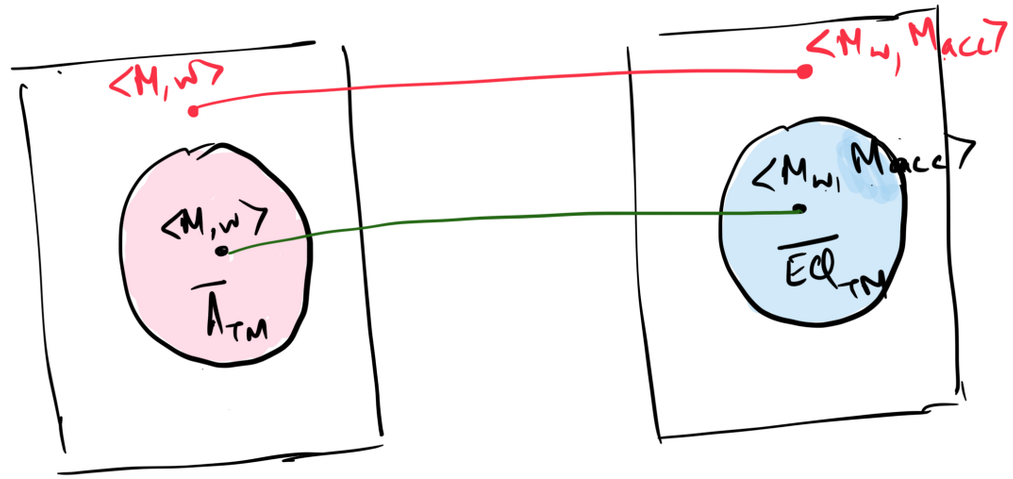


$f(\langle T_w, M_{acc}\rangle) \in \overline{\mathrm{EQ}}_{TM} \iff L(T_w) \neq L(M_{acc})$

$\iff L(T_w) \neq \{0,1\}^*$

$\iff M$ does not accept w

$\iff \langle M, w\rangle \in \bar{A}_{TM}$



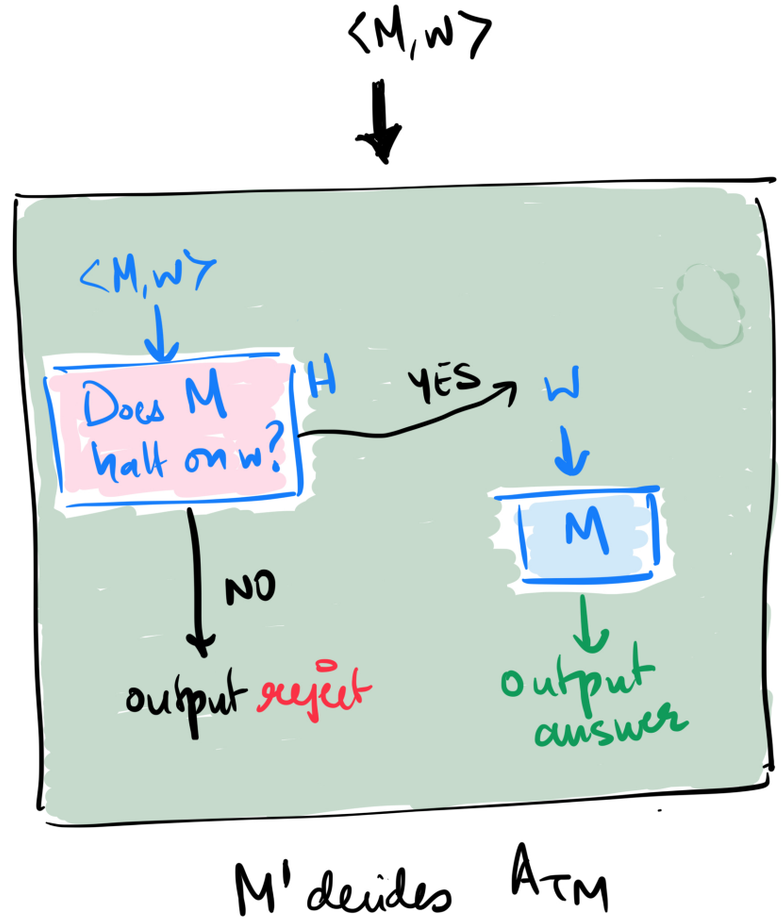
L_{01} is unrecognizable

- Let $L_{01} = \{\langle M \rangle \mid L(M) = \{01\}\}$.

Theorem: L_{01} is unrecognizable.

Turing Reducibility vs Mapping Reducibility

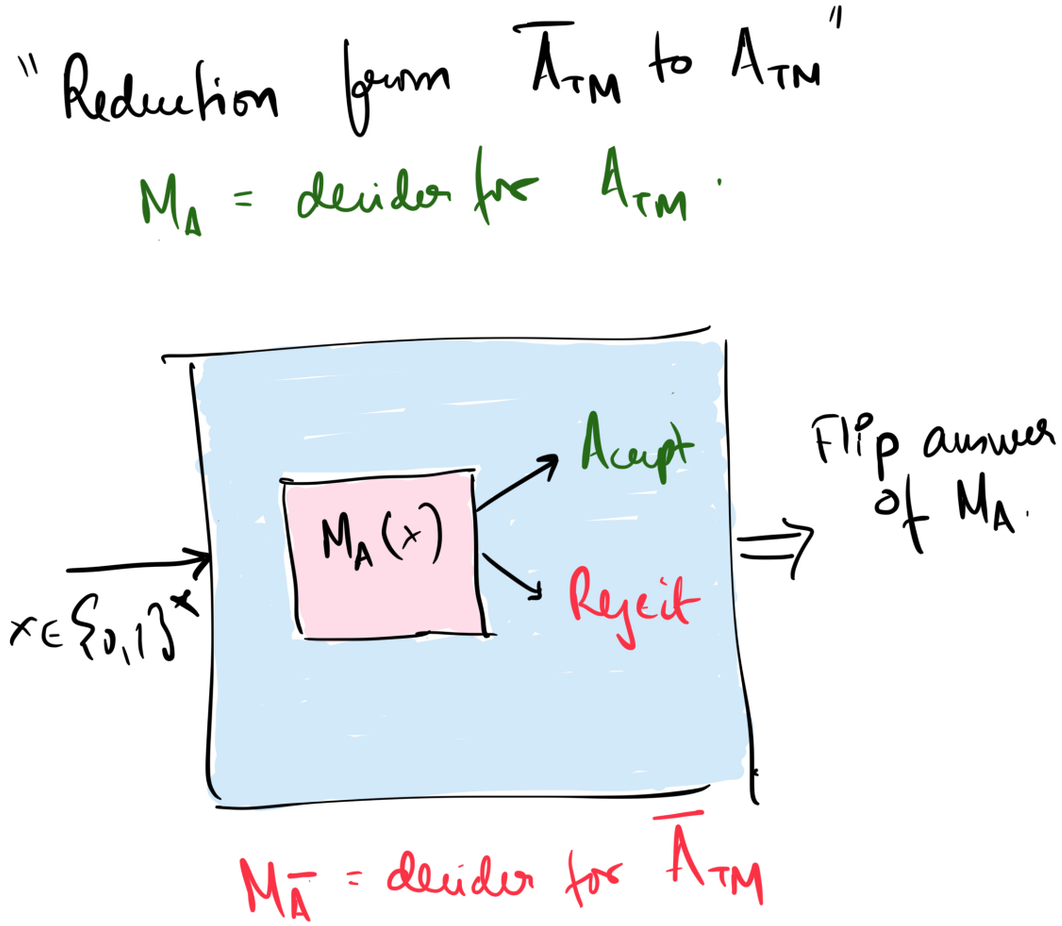
- We already prove that A_{TM} is undecidable (by diagonalization).
- Our goal was to show a candidate language L is undecidable.
- We showed if L is decidable, then so is A_{TM} .
  - Design an algorithm for A_{TM} using the hypothetical algorithm for L .
- "Reduction from A_{TM} to L."



Proving HALT$_{TM}$ is undecidable.

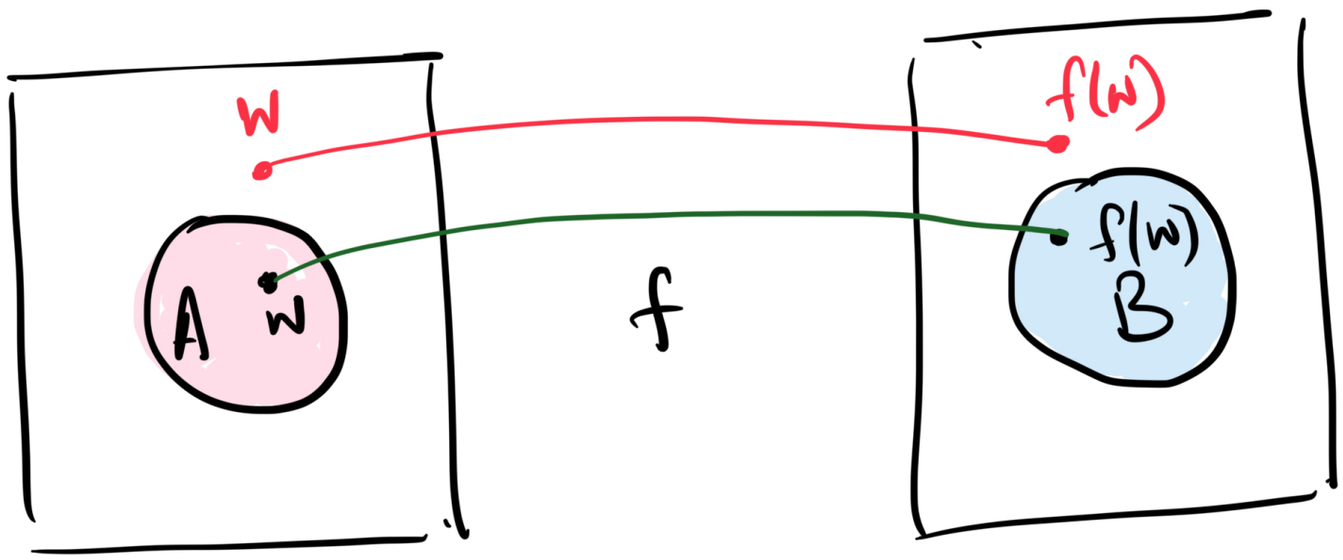
Turing Reducibility vs Mapping Reducibility

- We already prove that A_{TM} is undecidable (by diagonalization).
- Our goal was to show a candidate language L is undecidable.
- We showed if L is decidable, then so is A_{TM}.
  - Design an algorithm for A_{TM} using the hypothetical algorithm for L.
- "Reduction from A_{TM} to L."



Doesn't let us conclude unrecognizability: $\bar{A}_{TM}$ is unrecognisable, A_{TM} is recognisable.

Mapping Reducibility: More Refined



$$A \leq_m B$$

- Translate from A to B, while preserving membership and non-membership.
- So, we can directly return the answer of the recogniser for B without modifying the answer.

Reducibility Terminology

Why do we use the term “reduce”?

- When we reduce A to B, we show how to solve A by using B. We conclude that A is no harder than B. (This suggests the $\leq_m$ notation).
  - Here, reduce means to “boil down a question about A to a question about B”.

- **Possibility 1 (Algorithmic perspective):** We bring A’s difficulty down to B’s difficulty. (Using the solver for B to solve A.)

- **Possibility 2 (Hardness perspective):** We bring B’s difficulty up to to A’s difficulty. (Showing B is as hard as A.)

Summary

- Introduced the mapping reducibility to show undecidability/ unrecognisability.
- Showed EMPTY$_{TM}$ is undecidable and unrecognisable.
- Showed $\overline{\text{EQ}}_{TM}$ and EQ$_{TM}$ are unrecognisable.
- Showed L_{01} is unrecognisable.