COL 352 Introduction to Automata and Theory of Computation

Nikhil Balaji

Bharti 420
Indian Institute of Technology, Delhi
nbalaji@cse.iitd.ac.in

April 16, 2023

Lecture 31: Beyond Undecidability

Beyond Undecidability

Beyond Undecidability

- Virtually all interesting questions about Turing machines - A_{TM}, $HALT$, REG, FIN are undecidable.

Beyond Undecidability

- Virtually all interesting questions about Turing machines - A_{TM}, $HALT$, REG, FIN are undecidable.
- Are they all equally hard?
- Suppose by some magic we were given the power to decide $HALT$. Can we use that to decide REG?

Beyond Undecidability

- Virtually all interesting questions about Turing machines - A_{TM}, $HALT$, REG, FIN are undecidable.
- Are they all equally hard?
- Suppose by some magic we were given the power to decide $HALT$. Can we use that to decide REG?
- In other words, relative to the halting problem, is regularity decidable?

Beyond Undecidability

- Virtually all interesting questions about Turing machines - A_{TM}, $HALT$, REG, FIN are undecidable.
- Are they all equally hard?
- Suppose by some magic we were given the power to decide $HALT$. Can we use that to decide REG?
- In other words, relative to the halting problem, is regularity decidable?

Beyond Undecidability

- Virtually all interesting questions about Turing machines - A_{TM}, $HALT$, REG, FIN are undecidable.
- Are they all equally hard?
- Suppose by some magic we were given the power to decide $HALT$. Can we use that to decide REG?
- In other words, relative to the halting problem, is regularity decidable? How to model such questions?

Beyond Undecidability

- Virtually all interesting questions about Turing machines - A_{TM}, $HALT$, REG, FIN are undecidable.
- Are they all equally hard?
- Suppose by some magic we were given the power to decide $HALT$. Can we use that to decide REG?
- In other words, relative to the halting problem, is regularity decidable? How to model such questions?
- Oracle Turing machines!
- In addition to its ordinary read/write tape, there is a special one-way-infinite read-only input tape ("oracle tape") on which some infinite string ("oracle") is written.

Beyond Undecidability

- Virtually all interesting questions about Turing machines - A_{TM}, $HALT$, REG, FIN are undecidable.
- Are they all equally hard?
- Suppose by some magic we were given the power to decide $HALT$. Can we use that to decide REG?
- In other words, relative to the halting problem, is regularity decidable? How to model such questions?
- Oracle Turing machines!
- In addition to its ordinary read/write tape, there is a special one-way-infinite read-only input tape ("oracle tape") on which some infinite string ("oracle") is written.
- Machine can move its oracle tape head one cell in either direction in each step and make decisions based on the symbols written on the oracle tape.

Relative hardness

Definition

For $A, B \subseteq \Sigma^*$, we say that A is recursively enumerable (Turing recognizable) in B if there is an oracle TM M with oracle B such that $A = L(M)$. In addition, if M halts on all inputs, we write $A \leq_T B$ and say that A is recursive (decidable) in B or that A Turing reduces to B.

Relative hardness

Definition

For $A, B \subseteq \Sigma^*$, we say that A is recursively enumerable (Turing recognizable) in B if there is an oracle TM M with oracle B such that $A = L(M)$. In addition, if M halts on all inputs, we write $A \leq_T B$ and say that A is recursive (decidable) in B or that A Turing reduces to B.

Lemma

Halting problem is recursive in the membership problem.

Relative hardness

Definition

For $A, B \subseteq \Sigma^*$, we say that A is recursively enumerable (Turing recognizable) in B if there is an oracle TM M with oracle B such that $A = L(M)$. In addition, if M halts on all inputs, we write $A \leq_T B$ and say that A is recursive (decidable) in B or that A Turing reduces to B.

Lemma

Halting problem is recursive in the membership problem.

- Given a TM M and input x, first ask the oracle whether M accepts x.

Relative hardness

Definition

For $A, B \subseteq \Sigma^*$, we say that A is recursively enumerable (Turing recognizable) in B if there is an oracle TM M with oracle B such that $A = L(M)$. In addition, if M halts on all inputs, we write $A \leq_T B$ and say that A is recursive (decidable) in B or that A Turing reduces to B.

Lemma

Halting problem is recursive in the membership problem.

- Given a TM M and input x, first ask the oracle whether M accepts x.
- Given M, x, query oracle whether M accepts x. If the answer is yes, then output "yes".

Relative hardness

Definition

For $A, B \subseteq \Sigma^*$, we say that A is recursively enumerable (Turing recognizable) in B if there is an oracle TM M with oracle B such that $A = L(M)$. In addition, if M halts on all inputs, we write $A \leq_T B$ and say that A is recursive (decidable) in B or that A Turing reduces to B.

Lemma

Halting problem is recursive in the membership problem.

- Given a TM M and input x, first ask the oracle whether M accepts x.
- Given M, x, query oracle whether M accepts x. If the answer is yes, then output "yes".
- If the answer is no, switch accept and reject states of M (M') and query the oracle whether M' accepts x. If the answer is yes, output "yes". If the answer is still no, output "no" (M should loop on x).

Relative hardness

Lemma

Membership problem is recursive in the Halting problem.

Relative hardness

Lemma

Membership problem is recursive in the Halting problem.

- On input M, x ask the oracle if M halts on x. If it answers no, output "no".

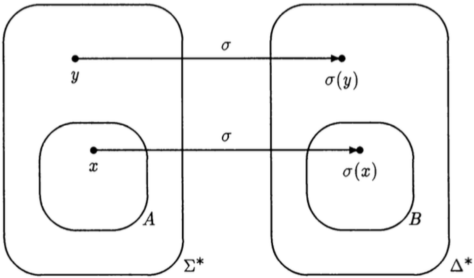
Relative hardness

Lemma

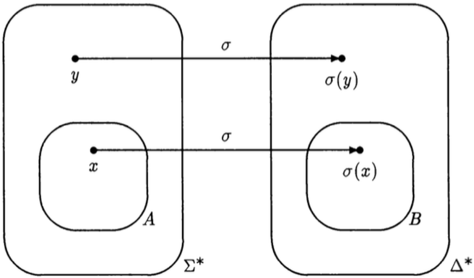
Membership problem is recursive in the Halting problem.

- On input M, x ask the oracle if M halts on x. If it answers no, output "no".
- If it answers "yes", run M on x and output the answer.

Many-one vs Turing reducibility

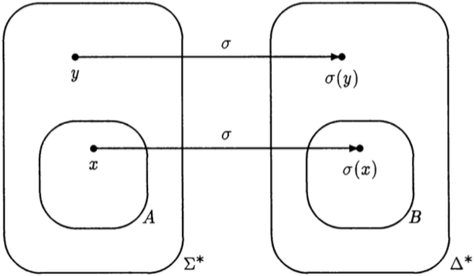


Many-one vs Turing reducibility



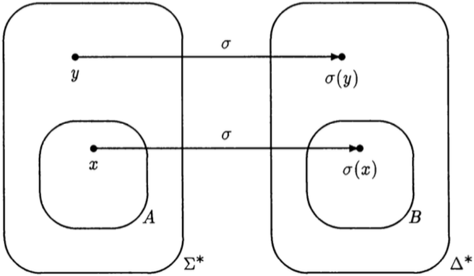
- $\leq_T$ is coarser than $\leq_m$.

Many-one vs Turing reducibility



- $\leq_T$ is coarser than $\leq_m$.
- $\leq_T$ is strictly coarser than $\leq_m$: $\overline{HALT} \not\leq_m HALT$, but $\overline{HALT} \leq_T HALT$.

Many-one vs Turing reducibility



- $\leq_T$ is coarser than $\leq_m$.
- $\leq_T$ is strictly coarser than $\leq_m$: $\overline{HALT} \not\leq_m HALT$, but $\overline{HALT} \leq_T HALT$.
- $\leq_T$ is transitive!

A hierarchy of hardness?

- Is everything computable with oracle access to $HALT$?

A hierarchy of hardness?

- Is everything computable with oracle access to $HALT$?
- What about $A_{TM}, HALT$ for oracle machines?

A hierarchy of hardness?

- Is everything computable with oracle access to $HALT$?
- What about $A_{TM}, HALT$ for oracle machines?
- Σ_1^0 = $\{$Turing Recognizable (r.e.) languages$\}$.

A hierarchy of hardness?

- Is everything computable with oracle access to $HALT$?
- What about $A_{TM}, HALT$ for oracle machines?
- Σ_1^0 = $\{$Turing Recognizable (r.e.) languages$\}$.
- Δ_1^0 = $\{$Decidable languages$\}$

A hierarchy of hardness?

- Is everything computable with oracle access to $HALT$?
- What about $A_{TM}, HALT$ for oracle machines?
- Σ_1^0 = {Turing Recognizable (r.e.) languages}.
- Δ_1^0 = {Decidable languages}
- Π_1^0 = {Complement of Turing Recognizable (co-r.e.) languages}

A hierarchy of hardness?

- Is everything computable with oracle access to $HALT$?
- What about $A_{TM}, HALT$ for oracle machines?
- $\Sigma_1^0 = \{$Turing Recognizable (r.e.) languages$\}$.
- $\Delta_1^0 = \{$Decidable languages$\}$
- $\Pi_1^0 = \{$Complement of Turing Recognizable (co-r.e.) languages$\}$
- $\Sigma_{n+1}^0 = \{$Languages r.e. in some $B \in \Sigma_n^0\}$

A hierarchy of hardness?

- Is everything computable with oracle access to $HALT$?
- What about $A_{TM}, HALT$ for oracle machines?
- $\Sigma_1^0 = \{$Turing Recognizable (r.e.) languages$\}$.
- $\Delta_1^0 = \{$Decidable languages$\}$
- $\Pi_1^0 = \{$Complement of Turing Recognizable (co-r.e.) languages$\}$
- $\Sigma_{n+1}^0 = \{$Languages r.e. in some $B \in \Sigma_n^0\}$
- $\Delta_{n+1}^0 = \{$Languages recursive in some $B \in \Sigma_n^0\}$

A hierarchy of hardness?

- Is everything computable with oracle access to $HALT$?
- What about $A_{TM}, HALT$ for oracle machines?
- Σ_1^0 = {Turing Recognizable (r.e.) languages}.
- Δ_1^0 = {Decidable languages}
- Π_1^0 = {Complement of Turing Recognizable (co-r.e.) languages}
- Σ_{n+1}^0 = {Languages r.e. in some $B \in \Sigma_n^0$}
- Δ_{n+1}^0 = {Languages recursive in some $B \in \Sigma_n^0$}
- Π_{n+1}^0 = {Complements of languages in Σ_n^0}

A hierarchy of hardness?

- Is everything computable with oracle access to $HALT$?
- What about $A_{TM}, HALT$ for oracle machines?
- $\Sigma_1^0 = \{$Turing Recognizable (r.e.) languages$\}$.
- $\Delta_1^0 = \{$Decidable languages$\}$
- $\Pi_1^0 = \{$Complement of Turing Recognizable (co-r.e.) languages$\}$
- $\Sigma_{n+1}^0 = \{$Languages r.e. in some $B \in \Sigma_n^0\}$
- $\Delta_{n+1}^0 = \{$Languages recursive in some $B \in \Sigma_n^0\}$
- $\Pi_{n+1}^0 = \{$Complements of languages in $\Sigma_n^0\}$

A hierarchy of harder and harder problems!

Arithmetic hierarchy via quantifiers

Arithmetic hierarchy via quantifiers

- $A_{TM} = \{(M, x) \mid \exists t M$ accepts x in t steps$\}$

Arithmetic hierarchy via quantifiers

- $A_{TM} = \{(M, x) \mid \exists t M \text{ accepts } x \text{ in } t \text{ steps}\}$
- $HALT = \{(M, x) \mid \exists t M \text{ halts on } x \text{ in } t \text{ steps}\}$

Arithmetic hierarchy via quantifiers

- $A_{TM} = \{(M,x) \mid \exists t M \text{ accepts } x \text{ in } t \text{ steps}\}$
- $HALT = \{(M,x) \mid \exists t M \text{ halts on } x \text{ in } t \text{ steps}\}$
- M halts on x is not a decidable predicate, but M halts on x in t steps is!

Arithmetic hierarchy via quantifiers

- $A_{TM} = \{(M, x) \mid \exists t M \text{ accepts } x \text{ in } t \text{ steps}\}$
- $HALT = \{(M, x) \mid \exists t M \text{ halts on } x \text{ in } t \text{ steps}\}$
- M halts on x is not a decidable predicate, but M halts on x in t steps is!
- $A_{TM} = \{(M, x) \mid \exists v, v \text{ is an accepting comp. history of } M \text{ on } x\}$
- $HALT = \{(M, x) \mid \exists v, v \text{ is a halting comp. history of } M \text{ on } x\}$

Arithmetic hierarchy via quantifiers

- $A_{TM} = \{(M, x) \mid \exists t M \text{ accepts } x \text{ in } t \text{ steps}\}$
- $HALT = \{(M, x) \mid \exists t M \text{ halts on } x \text{ in } t \text{ steps}\}$
- M halts on x is not a decidable predicate, but M halts on x in t steps is!
- $A_{TM} = \{(M, x) \mid \exists v, v \text{ is an accepting comp. history of } M \text{ on } x\}$
- $HALT = \{(M, x) \mid \exists v, v \text{ is a halting comp. history of } M \text{ on } x\}$
- $A \in \Sigma_1^0$ if and only if $A = \{x \mid \exists y R(x, y)\}$

Arithmetic hierarchy via quantifiers

- $A_{TM} = \{(M, x) \mid \exists t M$ accepts x in t steps$\}$
- $HALT = \{(M, x) \mid \exists t M$ halts on x in t steps$\}$
- M halts on x is not a decidable predicate, but M halts on x in t steps is!
- $A_{TM} = \{(M, x) \mid \exists v, v$ is an accepting comp. history of M on $x\}$
- $HALT = \{(M, x) \mid \exists v, v$ is a halting comp. history of M on $x\}$
- $A \in \Sigma_1^0$ if and only if $A = \{x \mid \exists y R(x, y)\}$
- $B \in \Pi_1^0$ if and only if $A = \{x \mid \forall y R(x, y)\}$

Arithmetic hierarchy via quantifiers

- $A_{TM} = \{(M, x) \mid \exists t M \text{ accepts } x \text{ in } t \text{ steps}\}$
- $HALT = \{(M, x) \mid \exists t M \text{ halts on } x \text{ in } t \text{ steps}\}$

- M halts on x is not a decidable predicate, but M halts on x in t steps is!

- $A_{TM} = \{(M, x) \mid \exists v, v \text{ is an accepting comp. history of } M \text{ on } x\}$
- $HALT = \{(M, x) \mid \exists v, v \text{ is a halting comp. history of } M \text{ on } x\}$

- $A \in \Sigma_1^0$ if and only if $A = \{x \mid \exists y R(x, y)\}$
- $B \in \Pi_1^0$ if and only if $A = \{x \mid \forall y R(x, y)\}$
- $\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0$

Arithmetic Hierarchy

Arithmetic Hierarchy

Theorem

1. *A set* A *is in* Σ^0_n *if there exists a decidable* $(n+1)$*-ary predicate* R *such that*

$$A = \{x \mid \exists y_1, \forall y_2, \ldots Q y_n R(x, y_1, \ldots, y_n)\}$$

where $Q = \exists$ *if* n *is odd and* $\forall$ *if* n *is even.*

2. *A set* A *is in* Π^0_n *iff there exists a decidable* $(n+1)$*-ary predicate* R *such that*

$$A = \{x \mid y_1, \forall y_2, \ldots Q y_n R(x, y_1, \ldots, y_n)\}$$

where $Q = \forall$ *if* n *is odd and* $\exists$ *if* n *is even.*

3. $\Delta^0_n = \Sigma^0_n \cap \Pi^0_n$.

Examples

$$EMPTY = \{M \mid L(M) = \varnothing\}$$

Examples

$$EMPTY = \{M \mid L(M) = \varnothing\}$$

$$EMPTY = \{M \mid \forall x \forall t, M \text{ does not accept in } t \text{ steps}\}$$

Examples

$$EMPTY = \{M \mid L(M) = \varnothing\}$$

$$EMPTY = \{M \mid \forall x \forall t, M \text{ does not accept in } t \text{ steps}\}$$

- Is this in Π_1^0?

Examples

$$EMPTY = \{M \mid L(M) = \emptyset\}$$

$$EMPTY = \{M \mid \forall x \forall t, M \text{ does not accept in } t \text{ steps}\}$$

- Is this in Π_1^0?
- $(i,j) \to \binom{i+j+1}{2} + i$ (Exercise: verify that this is a computable one-one pairing function!)

Examples

$$EMPTY = \{M \mid L(M) = \varnothing\}$$

$$EMPTY = \{M \mid \forall x \forall t, M \text{ does not accept in } t \text{ steps}\}$$

- Is this in Π_1^0?
- $(i,j) \to \binom{i+j+1}{2} + i$ (Exercise: verify that this is a computable one-one pairing function!)

$$TOTAL = \{M \mid M \text{ halts on all inputs}\}$$

Examples

$$EMPTY = \{M \mid L(M) = \varnothing\}$$

$$EMPTY = \{M \mid \forall x \forall t, M \text{ does not accept in } t \text{ steps}\}$$

- Is this in Π_1^0?
- $(i,j) \to \binom{i+j+1}{2} + i$ (Exercise: verify that this is a computable one-one pairing function!)

$$TOTAL = \{M \mid M \text{ halts on all inputs}\}$$

$$TOTAL = \{M \mid \forall x \exists t \text{ M halts on } x \text{ in } t \text{ steps}\}$$

Examples

$$EMPTY = \{M \mid L(M) = \varnothing\}$$

$$EMPTY = \{M \mid \forall x \forall t, M \text{ does not accept in } t \text{ steps}\}$$

- Is this in Π_1^0?
- $(i,j) \to \binom{i+j+1}{2} + i$ (Exercise: verify that this is a computable one-one pairing function!)

$$TOTAL = \{M \mid M \text{ halts on all inputs}\}$$

$$TOTAL = \{M \mid \forall x \exists t \text{ M halts on } x \text{ in } t \text{ steps}\}$$

$TOTAL \in \Pi_2^0$.

Examples

$$FIN = \{M \mid L(M) \text{ is finite}\}$$

Examples

$$FIN = \{M \mid L(M) \text{ is finite}\}$$

$$FIN = \{M \mid \exists n, \forall x, \text{ if } |x| > n, x \notin L(M)\}$$

Examples

$$FIN = \{M \mid L(M) \text{ is finite}\}$$

$$FIN = \{M \mid \exists n, \forall x, \text{ if } |x| > n,\ x \notin L(M)\}$$

$$FIN = \{M \mid \exists n, \forall x, \forall t, |x| \leq n \text{ or } M \text{ does not accept } x \text{ in } t \text{ steps}\}$$

Examples

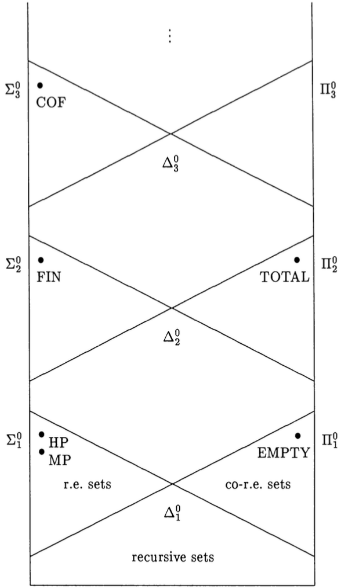
$$FIN = \{M \mid L(M) \text{ is finite}\}$$

$$FIN = \{M \mid \exists n, \forall x, \text{ if } |x| > n, \ x \notin L(M)\}$$

$$FIN = \{M \mid \exists n, \forall x, \forall t, |x| \le n \text{ or } M \text{ does not accept } x \text{ in } t \text{ steps}\}$$

$$FIN \in \Sigma_2^0$$

Arithmetic Hierarchy



Completeness

Completeness

- $A_{TM} \in r.e.$, and for every $L \in r.e$, $L \leq_m A_{TM}$.

Completeness

- $A_{TM} \in r.e.$, and for every $L \in r.e$, $L \leq_m A_{TM}$.
- A_{TM} is the "hardest" problem in Σ_1^0!

Completeness

- $A_{TM} \in r.e.$, and for every $L \in r.e$, $L \leq_m A_{TM}$.
- A_{TM} is the "hardest" problem in Σ_1^0!
- A set B is r.e.-complete, if it is both r.e.-hard and is in r.e.

Completeness

- $A_{TM} \in r.e.$, and for every $L \in r.e$, $L \leq_m A_{TM}$.
- A_{TM} is the "hardest" problem in Σ_1^0!
- A set B is r.e.-complete, if it is both r.e.-hard and is in r.e.
- For any class of languages $\mathcal{C}$, we say B is $\mathcal{C}$-complete if for all $L \in \mathcal{C}$, $L \leq_m B$ and $B \in \mathcal{C}$.

Completeness

- $A_{TM} \in r.e.$, and for every $L \in r.e$, $L \leq_m A_{TM}$.
- A_{TM} is the "hardest" problem in Σ_1^0!
- A set B is r.e.-complete, if it is both r.e.-hard and is in r.e.
- For any class of languages $\mathcal{C}$, we say B is $\mathcal{C}$-complete if for all $L \in \mathcal{C}$, $L \leq_m B$ and $B \in \mathcal{C}$.
- Exercise: All the problems we saw before are hard for the respective Σ_k^0, Π_k^0 classes.

Completeness

- $A_{TM} \in r.e.$, and for every $L \in r.e$, $L \leq_m A_{TM}$.
- A_{TM} is the "hardest" problem in Σ_1^0!
- A set B is r.e.-complete, if it is both r.e.-hard and is in r.e.
- For any class of languages $\mathcal{C}$, we say B is $\mathcal{C}$-complete if for all $L \in \mathcal{C}$, $L \leq_m B$ and $B \in \mathcal{C}$.
- Exercise: All the problems we saw before are hard for the respective Σ_k^0, Π_k^0 classes.
- $COFIN$ is Σ_3^0-complete!

and Beyond!

- What is 0 in Σ_1^0?

and Beyond!

- What is 0 in Σ_1^0?
- Arithmetic hierarchy: based on first order quantifications (quantification over natural numbers or strings).

and Beyond!

- What is 0 in Σ_1^0?
- Arithmetic hierarchy: based on first order quantifications (quantification over natural numbers or strings).
- What if you are allowed to quantify over functions and relations? (sets of numbers, strings)
- Analytic hierarchy: $\Sigma_1^1, \Pi_1^1, \ldots$ etc.

and Beyond!

- What is 0 in Σ_1^0?
- Arithmetic hierarchy: based on first order quantifications (quantification over natural numbers or strings).
- What if you are allowed to quantify over functions and relations? (sets of numbers, strings)
- Analytic hierarchy: $\Sigma_1^1, \Pi_1^1, \ldots$ etc.
- These classes do have natural complete problems!