Foundations of Computing

Lecture 16

Arkady Yerukhimovich

March 19, 2024

Outline

1. Lecture 15 Review
2. Proof by Reduction
3. Where Are We Now?
4. Reduction Types

Lecture 15 Review

- Countable and Uncountable Sets
  - Diagonalization
- Proving A_{TM} is Undecidable

Outline

1. Lecture 15 Review
2. Proof by Reduction
3. Where Are We Now?
4. Reduction Types

Another Way to Prove Undecidability

Reductions Between Problems

There is a reduction from a problem A to a problem B if we can use a solution to problem B to solve problem A

$$A \leq B$$

Another Way to Prove Undecidability

Reductions Between Problems

There is a reduction from a problem A to a problem B if we can use a solution to problem B to solve problem A

$$A \leq B$$

Intuition

$A \leq B$ means that:

- problem A is no harder than problem B.

Another Way to Prove Undecidability

Reductions Between Problems

There is a reduction from a problem A to a problem B if we can use a solution to problem B to solve problem A

$$A \leq B$$

Intuition

$A \leq B$ means that:

- problem A is no harder than problem B.
- Equivalently, problem B is no easier than problem A

Main Observation

Suppose that $A \leq B$, then:

- If A is undecidable
- B must also be undecidable

Reductions and Undecidability

Main Observation

Suppose that $A \leq B$, then:

- If A is undecidable
- B must also be undecidable

Proof: (by contradiction)

Reductions and Undecidability

Main Observation

Suppose that $A \leq B$, then:

- If A is undecidable
- B must also be undecidable

Proof: (by contradiction)

- Suppose that B is decidable

Reductions and Undecidability

Main Observation

Suppose that $A \leq B$, then:

- If A is undecidable
- B must also be undecidable

Proof: (by contradiction)

- Suppose that B is decidable
- Since $A \leq B$, there exists an algorithm (i.e., a reduction) that uses a solution to B to solve A

Reductions and Undecidability

Main Observation

Suppose that $A \leq B$, then:

- If A is undecidable
- B must also be undecidable

Proof: (by contradiction)

- Suppose that B is decidable
- Since $A \leq B$, there exists an algorithm (i.e., a reduction) that uses a solution to B to solve A
- But, this means that A is decidable by running the reduction using the decider machine for B.

Undecidability of $HALT_{TM}$

$$HALT_{TM} = \{\langle M, w \rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$$

Undecidability of $HALT_{TM}$

$$HALT_{TM} = \{\langle M, w \rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$$

Theorem: $HALT$ is undecidable

Undecidability of $HALT_{TM}$

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$$

Theorem: $HALT$ is undecidable

Proof Sketch:

- We show that $A_{TM} \leq HALT_{TM}$

Undecidability of $HALT_{TM}$

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$$

Theorem: $HALT$ is undecidable

Proof Sketch:

- We show that $A_{TM} \leq HALT_{TM}$
- Since we know that A_{TM} is undecidable, this shows that $HALT_{TM}$ is also undecidable

Undecidability of $HALT_{TM}$

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$$

Theorem: $HALT$ is undecidable

Proof Sketch:

- We show that $A_{TM} \leq HALT_{TM}$
- Since we know that A_{TM} is undecidable, this shows that $HALT_{TM}$ is also undecidable

Proof:

Construct reduction R that decides A_{TM} given a TM D that decides $HALT$

Undecidability of $HALT_{TM}$

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$$

Theorem: $HALT$ is undecidable

Proof Sketch:

- We show that $A_{TM} \leq HALT_{TM}$
- Since we know that A_{TM} is undecidable, this shows that $HALT_{TM}$ is also undecidable

Proof:

Construct reduction R that decides A_{TM} given a TM D that decides $HALT$

On input $\langle M, w\rangle$, R does the following: R is trying to decide if

$M(w) = 1$

Undecidability of $HALT_{TM}$

$$HALT_{TM} = \{\langle M, w \rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$$

Theorem: $HALT$ is undecidable

Proof Sketch:

- We show that $A_{TM} \leq HALT_{TM}$
- Since we know that A_{TM} is undecidable, this shows that $HALT_{TM}$ is also undecidable

Proof:

Construct reduction R that decides A_{TM} given a TM D that decides $HALT$

On input $\langle M, w \rangle$, R does the following:

- Run $D(\langle M, w \rangle)$

Undecidability of $HALT_{TM}$

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$$

Theorem: $HALT$ is undecidable

Proof Sketch:

- We show that $A_{TM} \leq HALT_{TM}$
- Since we know that A_{TM} is undecidable, this shows that $HALT_{TM}$ is also undecidable

Proof:

Construct reduction R that decides A_{TM} given a TM D that decides $HALT$

On input $\langle M, w\rangle$, R does the following:

- Run $D(\langle M, w\rangle)$
- If D rejects – $M(w)$ doesn't halt – halt and reject

Undecidability of $HALT_{TM}$

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$$

Theorem: $HALT$ is undecidable

Proof Sketch:

- We show that $A_{TM} \leq HALT_{TM}$
- Since we know that A_{TM} is undecidable, this shows that $HALT_{TM}$ is also undecidable

Proof:

Construct reduction R that decides A_{TM} given a TM D that decides $HALT$

On input $\langle M, w\rangle$, R does the following:

- Run $D(\langle M, w\rangle)$
- If D rejects – $M(w)$ doesn't halt – halt and reject
- if D accepts – $M(w)$ halts – Simulate $M(w)$ until it halts, and output whatever M outputs

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable
Proof Sketch:

- We show that $A_{TM} \leq REGULAR_{TM}$

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable

Proof Sketch:

- We show that $A_{TM} \leq REGULAR_{TM}$
- Specifically, reduction builds another TM M' s.t.

$\langle M, w\rangle$ Does $M(w) = 1$

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable
Proof Sketch:

- We show that $A_{TM} \leq REGULAR_{TM}$
- Specifically, reduction builds another TM M' s.t.
  - If M accepts w, M' recognizes Σ^* – regular language

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable
Proof Sketch:

- We show that $A_{TM} \leq REGULAR_{TM}$
- Specifically, reduction builds another TM M' s.t.
  - If M accepts w, M' recognizes Σ^* – regular language
  - If M does not accept w, M' recognizes $\{0^n 1^n\}$ – not regular

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable
Proof Sketch:

- We show that $A_{TM} \leq REGULAR_{TM}$
- Specifically, reduction builds another TM M' s.t.
  - If M accepts w, M' recognizes Σ^* – regular language
  - If M does not accept w, M' recognizes $\{0^n 1^n\}$ – not regular
- If we can decide whether M' recognizes a regular language or not, can use that to decide whether M accepts w or not

1. On input $\langle M, w \rangle$ build M'
2. Run $D(M')$
3. if $D(M') = 1 \Rightarrow M(w) = 1$, if $D(M') = 0 \Rightarrow M$ does not accept w

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable
Proof:

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable
Proof:
Reduction R that decides A_{TM} given a TM D that decides $REGULAR_{TM}$

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable
Proof:
Reduction R that decides A_{TM} given a TM D that decides $REGULAR_{TM}$
On input $\langle M, w\rangle$:

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable
Proof:
Reduction R that decides A_{TM} given a TM D that decides $REGULAR_{TM}$
On input $\langle M, w \rangle$:

1. Construct TM $M'_{\langle M,w \rangle}$ s.t. $M'_{\langle M,w \rangle}(x)$ is as follows:

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable
Proof:
Reduction R that decides A_{TM} given a TM D that decides $REGULAR_{TM}$
On input $\langle M, w \rangle$:

1. Construct TM $M'_{\langle M,w \rangle}$ s.t. $M'_{\langle M,w \rangle}(x)$ is as follows:
   - If $x = 0^n 1^n$, accept

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable
Proof:
Reduction R that decides A_{TM} given a TM D that decides $REGULAR_{TM}$
On input $\langle M, w \rangle$:

1. Construct TM $M'_{\langle M,w \rangle}$ s.t. $M'_{\langle M,w \rangle}(x)$ is as follows:
    - If $x = 0^n 1^n$, accept
    - If x does not have this form, run $M(w)$ and accept if it accepts

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable

Proof:

Reduction R that decides A_{TM} given a TM D that decides $REGULAR_{TM}$

On input $\langle M, w\rangle$:

1. Construct TM $M'_{\langle M,w\rangle}$ s.t. $M'_{\langle M,w\rangle}(x)$ is as follows:
   - If $x = 0^n1^n$, accept
   - If x does not have this form, run $M(w)$ and accept if it accepts
2. Run D on input $\langle M'\rangle$

Other Undecidable Languages

$$REGULAR_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Theorem: $REGULAR_{TM}$ is undecidable
Proof:
Reduction R that decides A_{TM} given a TM D that decides $REGULAR_{TM}$
On input $\langle M, w \rangle$:

1. Construct TM $M'_{\langle M,w \rangle}$ s.t. $M'_{\langle M,w \rangle}(x)$ is as follows:
   - If $x = 0^n 1^n$, accept
   - If x does not have this form, run $M(w)$ and accept if it accepts
2. Run D on input $\langle M' \rangle$
3. Output what D outputs

$M(w) = 1$: M' accepts all strings $L(M') = \Sigma^* -$ Reg

M does not accept w : M' accepts $0^n 1^n$ $L(M')$ – Not Reg.

Other Undecidable Languages – Exercise

$$EMPTY - STRING_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } M(\epsilon) = 1\}$$