CSCI 361 Lecture 15: Reducibility

Shikha Singh

Announcements & Logistics

- Happy Halloween:
  - Grab a candy from the candy bowl
- Hand in **reading assignment # 10**
- Pick up **reading assignment # 11**
  - Due start of class on Tues Nov 5
- **HW 6** released, due next Wed Nov 6
- My elective!

Last Time

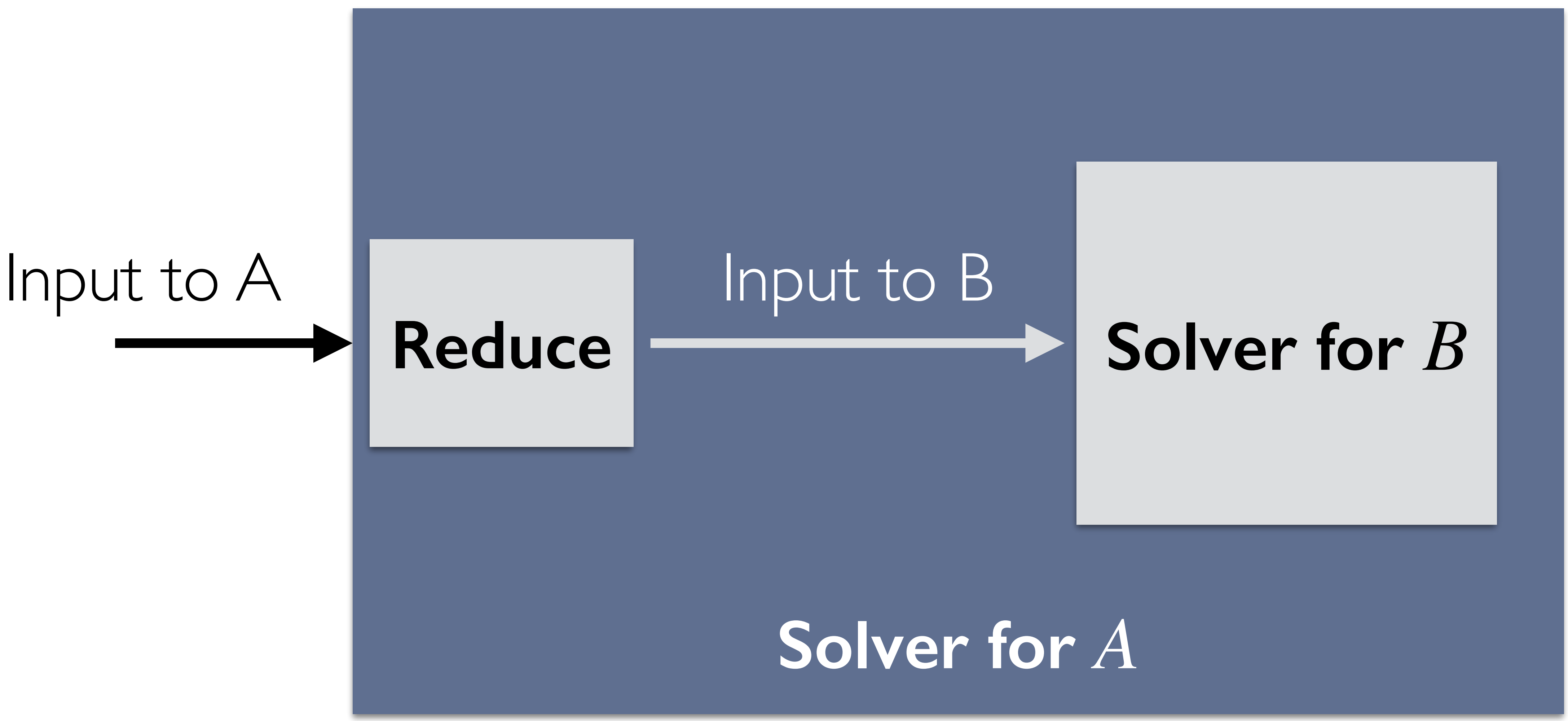
- Proved some problems are undecidable:
  - Given a TM and an input, does it accept it?
    - $A_{\text{TM}} = \{\langle M, w\rangle \mid T \text{ is a TM and } w \in L(M)\}$
  - Given a TM and an input, does it halt on it (accept/reject it)?
    - $\text{HALT}_{\text{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ halts on } w\}$
  - Given a TM, is its language empty?
    - $E_{\text{TM}} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) = \varnothing\}$
- Introduced Turing reductions

Today

- Do more practice with reductions to prove undecidability
- Identify many more TM undecidable and TM unrecognizable problems

Reduction Review

- Informally, problem A ***reduces*** to Problem B if we can use the solution of B to solve A



Review Reduction: A_{TM} to E_{TM}

Suppose TM R decides E_{TM}. Consider the following decider:

- $D =$ "On input $\langle M, w \rangle$
  - Encode a TM M_w that does the following:
    - $M_w =$ "On input x,
      - If $x \neq w$, reject.
      - If $x = w$, then run M on w and accept if M does, else reject.
  - Run R on $\langle M_w \rangle$. If R accepts, reject; if R rejects, accept.
- **Correctness**: If R is a decider for E_{TM} then D a decider for A_{TM}.

More Practice with Reductions

Definition. $\text{REGULAR}_{\text{TM}} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$. M is a TM.

Question. Show that $\text{REGULAR}_{\text{TM}}$ is undecidable.

Proof. Reduce A_{TM} to $\text{REGULAR}_{\text{TM}}$.

Goal: Given $\langle M, w \rangle$, convert it to a new Turing machine M_{new} s.t.

M accepts w if and only if $L(M_{\text{new}})$ is regular.

Idea: Let's try a similar idea as the last reduction

$\text{REGULAR}_{\text{TM}}$ is undecidable

Proof. Let R be a decider for $\text{REGULAR}_{\text{TM}}$. Then consider TM D:

D = "On input $\langle M, w \rangle$

1. Create M_{new} = "On input x,
    1. If x has the form $0^n 1^n$, then accept.
    2. Otherwise, run M on w and accept if M accepts."
2. Run R on M_{new}.
3. If R accepts, accept. If R rejects, reject.

- What is $L(M_{\text{new}})$?

$\text{REGULAR}_{\text{TM}}$ is undecidable

Proof. Let R be a decider for $\text{REGULAR}_{\text{TM}}$. Then consider TM D:

D = "On input $\langle M, w \rangle$

1. Create M_{new} = "On input x,
    1. If x has the form $0^n 1^n$, then accept.
    2. Otherwise, run M on w and accept if M accepts."
2. Run R on M_{new}.
3. If R accepts, accept. If R rejects, reject.

- Suppose M accepts w, then M_{new} accepts all x, $L(M_{\text{new}}) = \Sigma^*$
- Suppose M does not accept w, then M_{new} only accepts $0^n 1^n$

$\text{REGULAR}_{\text{TM}}$ is undecidable

Proof. Let R be a decider for $\text{REGULAR}_{\text{TM}}$. Then consider TM D:

D = "On input $\langle M, w \rangle$

1. Create M_{new} = "On input x,
   1. If x has the form $0^n 1^n$, then accept.
   2. Otherwise, run M on w and accept if M accepts."
2. Run R on M_{new}.
3. If R accepts, accept. If R rejects, reject.

- M accepts w if and only if $L(M_{\text{new}})$ is regular. ∎

Exercise

Problem. Show that

$EQ_{\mathsf{TM}} = \{\langle M, N \rangle \mid M, N$ are TMs and $L(M) = L(N)\}$ is undecidable.

Hint. Reduce E_{TM} to it.

Mapping Reducibility

- A technical formulation of reducibility that lets us prove more things
- **Definition.** Language A is mapping reducible to language B, denoted $A \leq_m B$, if there exists a computable function $f : \Sigma^* \to \Sigma^*$, such that

$$w \in A \iff f(w) \in B \quad \text{for every } w$$

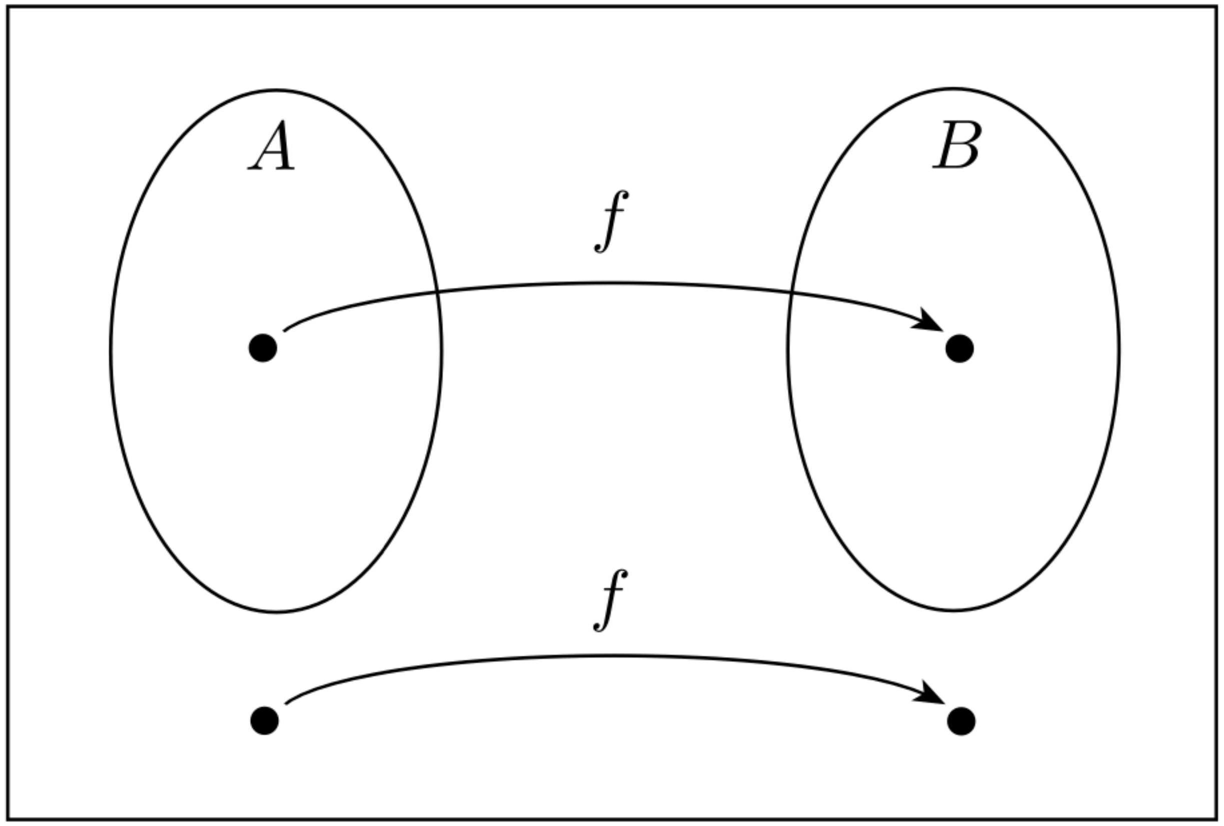
- The function f is called the ***reduction*** from A to B
- A function $f : \Sigma^* \to \Sigma^*$ is ***computable*** if some Turing machine M when given any input w, halts with just the output $f(w)$ on its tape.

Mapping Reducibility

- **Definition.** Language A is mapping reducible to language B, denoted $A \leq_m B$, if there exists a computable function $f : \Sigma^* \to \Sigma^*$, such that

$$w \in A \iff f(w) \in B \quad \text{for every } w$$

- **Remark.** If $A \leq_m B$ then $\overline{A} \leq_m \overline{B}$



Mapping Reducibility

- Using reductions to prove **decidability**:
  - **Theorem.** If $A \leq_m B$ and B is decidable, then A is decidable.
  - Why is this true?
- Using reductions to prove **undecidability**:
  - **Corollary.** If $A \leq_m B$ and A is undecidable, then B is undecidable.

Revisit Past Reductions

Reduction from A_{TM} to HALT_{TM} from last lecture:

- Suppose TM R decides HALT_{TM}.
- Construct a decider S for A_{TM} =
  "On input $\langle M, w \rangle$,
  - Run R on $\langle M, w \rangle$.
  - If R rejects, then reject.
  - If R accepts, then simulate M on w. If M enters accept state, then accept; if M enters reject state, then reject.

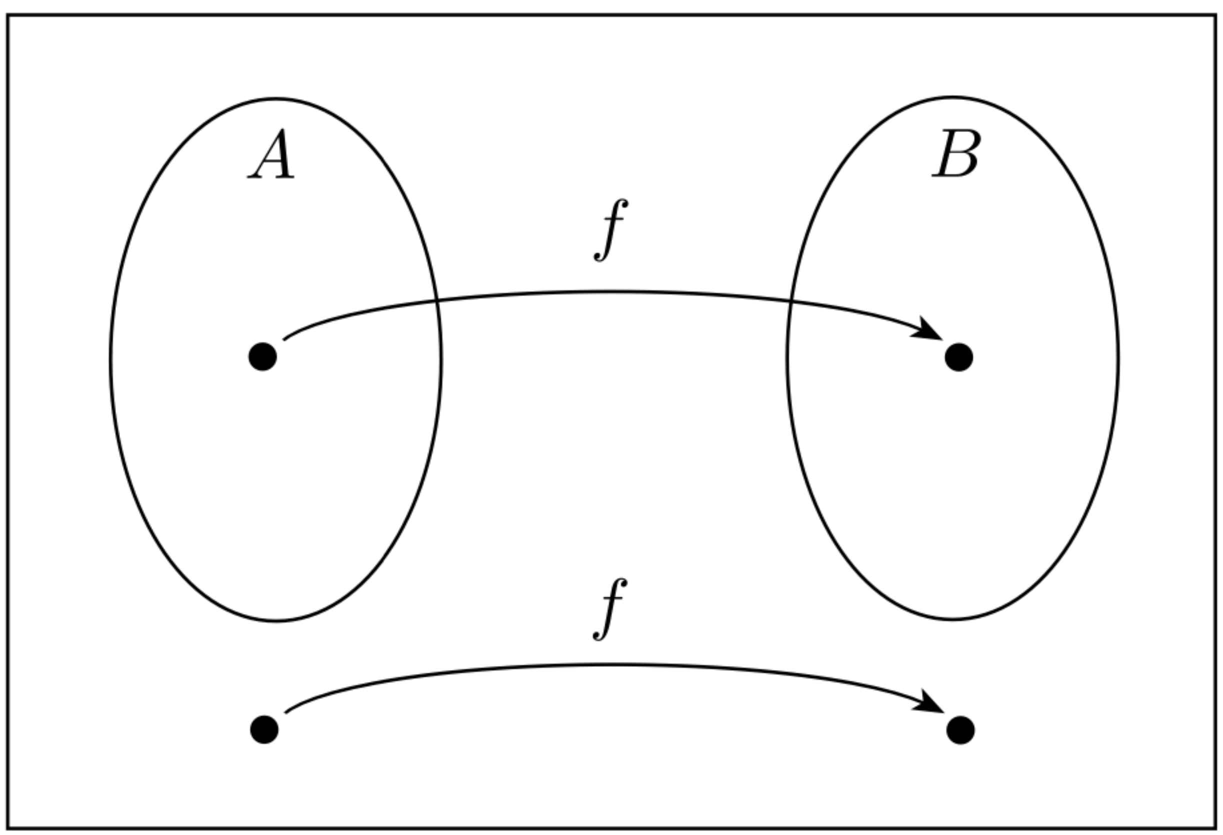
Question. Is this a mapping reduction from A_{TM} to HALT_{TM}?

Revisit Past Reductions

Key difference: Need to map "yes" instances to "yes" and "no" to "no".

Need a computable function f that maps $\langle M, w\rangle$ to $\langle M', w'\rangle$ such that

$$\langle M, w\rangle \in A_{\mathsf{TM}} \text{ iff } \langle M', w'\rangle \in \mathsf{HALT}_{\mathsf{TM}}$$



Mapping Reduction: A_{TM} to $\mathrm{HALT}_{\mathrm{TM}}$

Reduction function computed by the following Turing machine:

F = "On input $\langle M, w \rangle$:

1. Construct the machine M' = "On input x:
   1. Run M on x.
   2. If M accepts, *accept.*
   3. If M rejects, go into an infinite loop.
4. Output $\langle M', w \rangle$"

Why Mapping Reductions?

- Seem unnecessarily strict, can use informal reductions just fine to prove undecidability
- Why force mapping from yes instances to yes, no to no?
  - Useful to reason about Turing recognizability and unrecognizability
- Mapping reductions to prove **recognizability**:
  - **Theorem.** If $A \leq_m B$ and B is recognizable, then A is recognizable.
- Mapping reductions to prove **unrecognizability**:
  - **Corollary.** $A \leq_m B$ and A is unrecognizable, then B is unrecognizable.

Exercise

- Review the reductions from earlier:
  - From A_{TM} to E_{TM}
  - From E_{TM} to EQ_{TM}
  - From A_{TM} to $\mathrm{REGULAR}_{\mathsf{TM}}$
- **Questions.**
  - Which of these are mapping reductions?
  - Is it possible to have a mapping reduction in all these cases?

No Mapping Reduction: A_{TM} to E_{TM}

- Earlier reduction is mapping reduction from A_{TM} to $\overline{E_{\mathsf{TM}}}$
    - That is, $\overline{A_{\mathsf{TM}}} \leq_m E_{\mathsf{TM}}$
- What can we say about E_{TM}?
    - Since $\overline{A_{\mathsf{TM}}}$ is not Turing recognizable, E_{TM} is also not Turing recognizable.
- Found another example of a language that is not recognizable!
- **Exercise.** Show that A_{TM} is not mapping reducible to E_{TM}.

Undecidability Summary

Question. Which of these are decidable?

- Acceptance problems for DFA, CFG, TM
- Emptiness problems for DFA, CFG, TM
- Accepts all strings problem for DFA, CFG, TM
- Equivalence problems for DFA, CFG, TM

Rice's Theorem

Any nontrivial **property of the languages** recognized by Turing machines is undecidable.

- Is the language empty? Is it finite? Is it infinite? Is it regular?
- Does the language contain strings in Σ^*
- Is the language the same as the language of another TM?

We proved many such examples in class.

HW 6 will have more practice with these.

Rice's Theorem

Any nontrivial **property of the languages** recognized by Turing machines is undecidable.

- Is the language empty? Is it finite? Is it infinite? Is it regular?
- Does the language contain strings in Σ^*
- Is the language the same as the language of another TM?

In contrast, questions about the TM's structure are decidable

- Has more than 15 states, has no transitions into its reject state, etc

Decidable or Not

Questions about behavior of TM's computation on inputs may or may not be decidable.

(HW 6 Problem): One of these is decidable, one is not:

- Does a given TM M and input w, does it ever move its head left when running on w?
- Does a given TM M and input w, does it ever move its head three times in a row when running on w?