18.404/6.840 Lecture 9

Last time:
- is undecidable
- The diagonalization method
- is T-unrecognizable
- The Reducibility Method, preview

Today:
- The Reducibility Method for proving undecidability
  and T-unrecognizability.
- General reducibility
- Mapping reducibility

Posted: Problem Set 2 solutions and Problem Set 3.
TAs are available to answer questions during chat-breaks!

The Reducibility Method

If we know that some problem (say) is undecidable,
we can use that to show other problems are undecidable.

Defn: halts on input

Recall Theorem: is undecidable
Proof by contradiction, showing that is reducible to :

Assume that is decidable and show that is decidable (false!).
Let TM decide .
Construct TM deciding .

"On input
1. Use to test if on halts. If not, *reject*.
2. Simulate on until it halts (as guaranteed by).
3. If has accepted then *accept*.
   If has rejected then *reject*.

TM decides , a contradiction. Therefore is undecidable.

Reducibility – Concept

If we have two languages (or problems) and , then
 is reducible to means that we can use to solve .

Example 1: Measuring the area of a rectangle
is reducible to measuring the lengths of its sides.

Example 2: We showed that is reducible to .

Example 3: From Pset 2, *PUSHER* is reducible to .
(Idea- Convert push states to accept states.)

If is reducible to then solving gives a solution to .
- then is easy   is easy.
- then is hard   is hard.

this is the form we will use

Check-in 9.1

Is Biology reducible to Physics?

(a) Yes, all aspects of the physical world may be explained in terms of Physics, at least in principle.

(b) No, some things in the world, maybe life, the brain, or consciousness, are beyond the realm pf Physics.

(c) I’m on the fence on this question!

is undecidable

Let is a TM and

Theorem: is undecidable

Proof by contradiction. Show that is reducible to .

Assume that is decidable and show that is decidable (false!).
Let TM decide .
Construct TM deciding .

"On input

1. Transform to new TM "On input
    1. If , *reject*.
    2. else run on
    3. *Accept* if accepts."
2. Use to test whether
3. If YES [so rejects] then *reject*.
   If NO [so accepts] then *accept*.

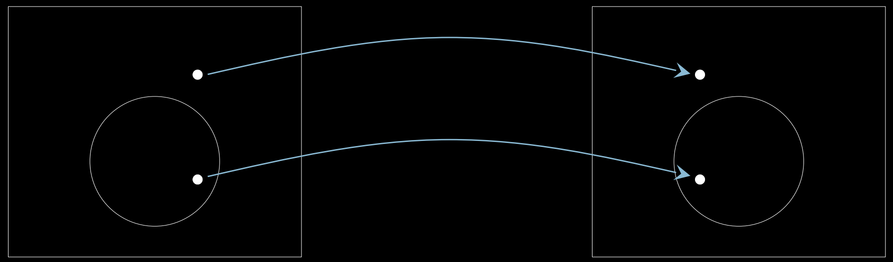
works like except that it always rejects strings where .

So

Mapping Reducibility

Defn: Function is <u>computable</u> if there is a TM
where on input halts with on its tape, for all strings .

Defn: <u> is mapping-reducible to () </u>if there is
a computable function where iff .



Example:

The computable reduction function is) =

Because iff

(accepts iff)

Recall TM “On input

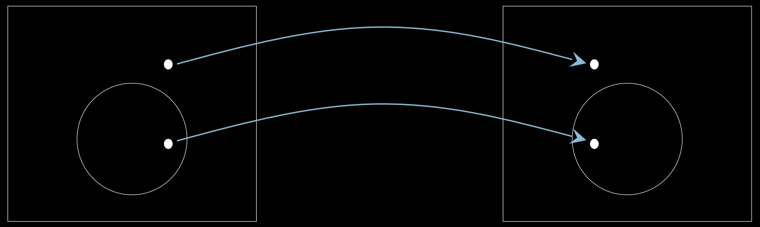
1. If , *reject*.
2. else run on
3. *Accept* if accepts.”

Mapping Reductions - properties

Theorem: If and is decidable then so is

Proof: Say TM decides .

Construct TM deciding :



"On input

1. Compute
2. Run on to test if
3. If halts then output same result."

Corollary: If and is undecidable then so is

Theorem: If and is T-recognizable then so is

Proof: Same as above.

Corollary: If and is T-unrecognizable then so is

Check-in 9.2

Suppose .

What can we conclude?

Check all that apply.

(a)

(b)

(c) None of the above

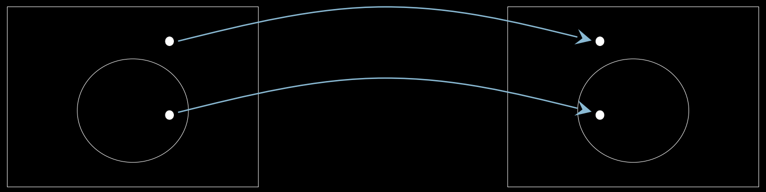
Check-in 9.2


Coffee
Break

Mapping vs General Reducibility

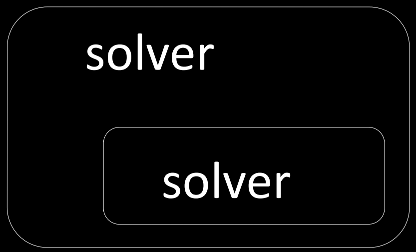
Mapping Reducibility of to : Translate -questions to -questions.

- A special type of reducibility
- Useful to prove T-unrecognizability



(General) Reducibility of to : Use solver to solve .

- May be conceptually simpler
- Useful to prove undecidability



Noteworthy difference:

- is reducible to
- may not be mapping reducible to .

For example

Check-in 9.3

We showed that if and is T-recognizable then so is .

Is the same true if we use general reducibility instead of mapping reducibility?

(a) Yes

(b) No

Check-in 9.3

Reducibility – Templates

To prove is undecidable:

- Show undecidable is reducible to . (often is)
- Template: Assume TM decides .
  Construct TM deciding . Contradiction.

To prove is T-unrecognizable:

- Show T-unrecognizable is mapping reducible to . (often is)
- Template: give reduction function .

is T-unrecognizable

Recall is a TM and

Theorem: is T-unrecognizable

Proof: Show

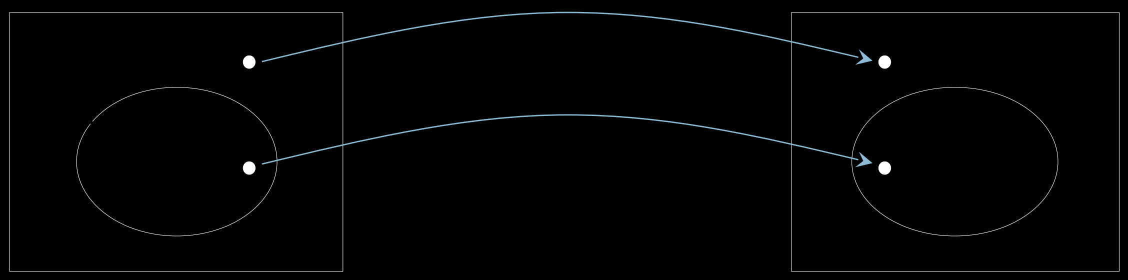
Reduction function:

Explanation: iff

rejects iff

Recall TM "On input

1. If , *reject*.
2. else run on
3. *Accept* if accepts."



and are T-unrecognizable

and are TMs and

Theorem: Both and are T-unrecognizable

Proof: (1)

(2)

For any let “On input

acts on all inputs the way acts on .

1. Ignore .
2. Simulate on .”

(1) Here we give which maps problems (of the form) to problems (of the form).

is a TM that always rejects.

(2) Similarly always accepts.

Reducibility terminology

Why do we use the term “reduce”?

When we reduce to , we show how to solve by using
and conclude that is no harder than . (suggests the notation)

Possibility 1: We bring ’s difficulty down to ’s difficulty.

Possibility 2: We bring ’s difficulty up to ’s difficulty.

Quick review of today

1. Introduced The Reducibility Method to prove undecidability and T-unrecognizability.
2. Defined mapping reducibility as a type of reducibility.
3. is undecidable.
4. is T-unrecognizable.
5. and are T-unrecognizable.