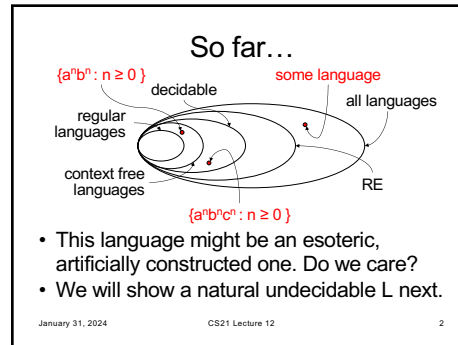




1



2

The Halting Problem

- Definition of the "Halting Problem":
 $HALT = \{ \langle M, x \rangle : TM M \text{ halts on input } x \}$
- HALT is recursively enumerable.
– proof?
- Is HALT decidable?

January 31, 2024 CS21 Lecture 12 3

3

The Halting Problem

Theorem: HALT is not decidable (undecidable).

Proof:

- Suppose TM H decides HALT
- Define new TM H': on input $\langle M \rangle$
 - if H accepts $\langle M, \langle M \rangle \rangle$ then loop
 - if H rejects $\langle M, \langle M \rangle \rangle$ then halt

January 31, 2024 CS21 Lecture 12 4

4

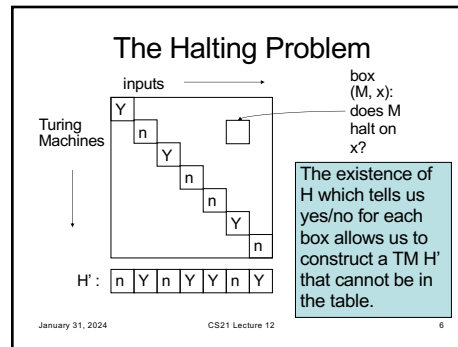
The Halting Problem

Proof:

- define new TM H': on input $\langle M \rangle$
 - if H accepts $\langle M, \langle M \rangle \rangle$ then loop
 - if H rejects $\langle M, \langle M \rangle \rangle$ then halt
- consider H' on input $\langle H' \rangle$:
 - if it halts, then H rejects $\langle H', \langle H' \rangle \rangle$, which implies it cannot halt
 - if it loops, then H accepts $\langle H', \langle H' \rangle \rangle$ which implies it must halt
- contradiction.

January 31, 2024 CS21 Lecture 12 5

5



6

So far...

$\{a^n b^n : n \geq 0\}$ decidable
 regular languages
 context free languages
 $\{a^n b^n c^n : n \geq 0\}$ RE
 HALT
 some language
 all languages

- Can we exhibit a natural language that is non-RE?

January 31, 2024 CS21 Lecture 12 7

7

RE and co-RE

- The complement of a RE language is called a co-RE language

$\{a^n b^n : n \geq 0\}$ decidable co-RE some language
 regular languages
 context free languages
 $\{a^n b^n c^n : n \geq 0\}$ RE
 HALT
 all languages

January 31, 2024 CS21 Lecture 12 8

8

RE and co-RE

Theorem: a language L is decidable if and only if L is RE and L is co-RE.

Proof:

(\Rightarrow) we already know decidable implies RE

- if L is decidable, then complement of L is decidable by flipping accept/reject.
- so L is in co-RE.

January 31, 2024 CS21 Lecture 12 9

9

RE and co-RE

Theorem: a language L is decidable if and only if L is RE and L is co-RE.

Proof:

(\Leftarrow) we have TM M that recognizes L , and TM M' recognizes complement of L .

- on input x , simulate M, M' in parallel
- if M accepts, accept; if M' accepts, reject.

January 31, 2024 CS21 Lecture 12 10

10

A natural non-RE language

Theorem: the complement of HALT is not recursively enumerable.

Proof:

- we know that HALT is RE
- suppose complement of HALT is RE
- then HALT is co-RE
- implies HALT is decidable. Contradiction.

January 31, 2024 CS21 Lecture 12 11

11

Summary

$\{a^n b^n : n \geq 0\}$ decidable co-RE some language
 regular languages
 context free languages
 $\{a^n b^n c^n : n \geq 0\}$ RE
 HALT
 all languages

Main point: some problems have no algorithms, HALT in particular.

January 31, 2024 CS21 Lecture 12 12

12

Reductions

- Given a new problem NEW, want to determine if it is easy or hard
 - right now, easy typically means decidable
 - right now, hard typically means undecidable
- One option:
 - prove from scratch that the problem is decidable, or
 - prove from scratch that the problem is undecidable (dream up a diag. argument)

January 31, 2024 CS21 Lecture 12 13

13

Reductions

- A better option:
 - to prove NEW is decidable, show how to transform it into a known decidable problem OLD so that solution to OLD can be used to solve NEW.
 - to prove NEW is undecidable, show how to transform a known undecidable problem OLD into NEW so that solution to NEW can be used to solve OLD.
- called a **reduction**

January 31, 2024 CS21 Lecture 12 14

14

Reductions

Reductions are one of the most important and widely used techniques in theoretical Computer Science.

- especially for proving problems "hard"
 - often difficult to do "from scratch"
 - sometimes not known how to do from scratch
 - reductions allow proof by giving an algorithm to perform the transformation

January 31, 2024 CS21 Lecture 12 15

15

Example reduction

- Try to prove undecidable:
 - $A_{TM} = \{ \langle M, w \rangle : M \text{ accepts input } w \}$
- We know this language is undecidable:
 - $HALT = \{ \langle M, w \rangle : M \text{ halts on input } w \}$
- Idea:
 - suppose A_{TM} is decidable
 - show that we can use A_{TM} to decide HALT
 - conclude HALT is decidable. Contradiction.

reduction

January 31, 2024 CS21 Lecture 12 16

16

Example reduction

- How could we use procedure that decides A_{TM} to decide HALT?
 - given input to HALT: $\langle M, w \rangle$
- Some things we can do:
 - check if $\langle M, w \rangle \in A_{TM}$
 - construct another TM M' and check if $\langle M', w \rangle \in A_{TM}$

January 31, 2024 CS21 Lecture 12 17

17

Example reduction

- Deciding HALT using a procedure that decides A_{TM} ("reducing HALT to A_{TM} ").
 - on input $\langle M, w \rangle$
 - check if $\langle M, w \rangle \in A_{TM}$
 - if yes, the M halts on w; **ACCEPT**
 - if no, then M either rejects w or it loops on w
 - construct M' by swapping q_{accept}/q_{reject} in M
 - check if $\langle M', w \rangle \in A_{TM}$
 - if yes, then M' accepts w, so M rejects w; **ACCEPT**
 - if no, then M neither accepts nor rejects w; **REJECT**

January 31, 2024 CS21 Lecture 12 18

18

Example reduction

- Preceding reduction proved:

Theorem: A_{TM} is undecidable.

Proof (recap):

- suppose A_{TM} is decidable
- we showed how to use A_{TM} to decide HALT
- conclude HALT is decidable. Contradiction.

January 31, 2024 CS21 Lecture 12 19

19

Another example

- Try to prove undecidable:
 - $E_{TM} = \{ \langle M \rangle : L(M) = \emptyset \}$
- which problem should we **reduce from**?
 - HALT = $\{ \langle M, w \rangle : M \text{ halts on input } w \}$
 - $A_{TM} = \{ \langle M, w \rangle : M \text{ accepts input } w \}$
- Some things we can do:
 - check if $\langle M \rangle \in E_{TM}$
 - construct another TM M' and check if $\langle M' \rangle \in E_{TM}$

January 31, 2024 CS21 Lecture 12 20

20

Another example

- We are given input $\langle M, w \rangle$
- We want to use a procedure that decides E_{TM} to decide if $\langle M, w \rangle \in A_{TM}$
- Idea:
 - check if $\langle M \rangle \in E_{TM}$
 - if not?
 - helpful if could make M reject everything except possibly w.

January 31, 2024 CS21 Lecture 12 21

21

Another example

- Construct TM M' :
 - on input x, if $x \neq w$, then reject.
 - else simulate M on x, and accept if M does.
- on input $\langle M, w \rangle$
 - construct M' from description of M
 - check if $\langle M' \rangle \in E_{TM}$
 - if no, M must accept w; **ACCEPT**
 - if yes, M cannot accept w; **REJECT**

January 31, 2024 CS21 Lecture 12 22

22

Another example

- Preceding reduction proved:

Theorem: E_{TM} is undecidable.

Proof (recap):

- suppose E_{TM} is decidable
- we showed how to use E_{TM} to decide A_{TM}
- conclude A_{TM} is decidable. Contradiction.

January 31, 2024 CS21 Lecture 12 23

23

Example reduction

- We proved
 - $A_{TM} = \{ \langle M, w \rangle : M \text{ accepts input } w \}$
 - undecidable, by reduction from
 - HALT = $\{ \langle M, w \rangle : M \text{ halts on input } w \}$
- We proved
 - $E_{TM} = \{ \langle M \rangle : L(M) = \emptyset \}$
 - undecidable by reduction from A_{TM}

January 31, 2024 CS21 Lecture 12 24

24