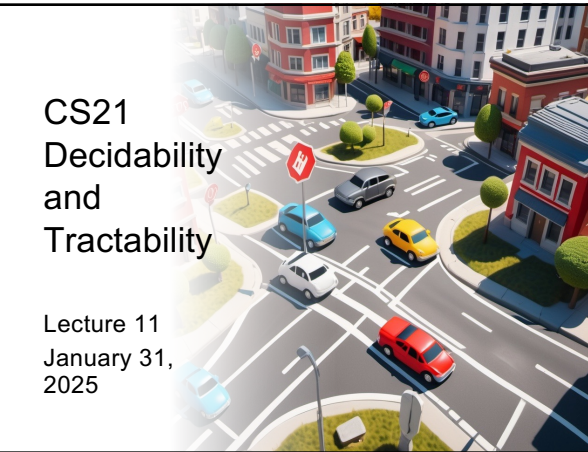


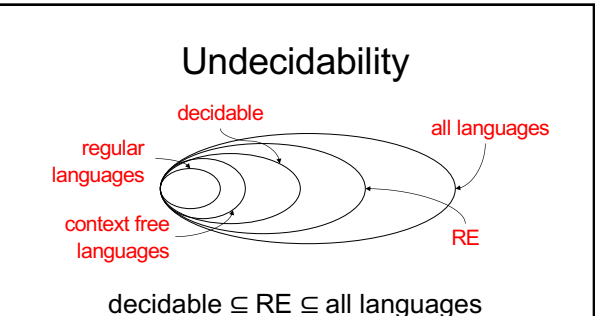
CS21 Decidability and Tractability

Lecture 11
January 31,
2025



1

Undecidability



decidable

regular languages

context free languages

RE

all languages

decidable \subseteq RE \subseteq all languages

our goal: prove these containments proper

January 31, 2025 CS21 Lecture 11

2

Countable and Uncountable Sets

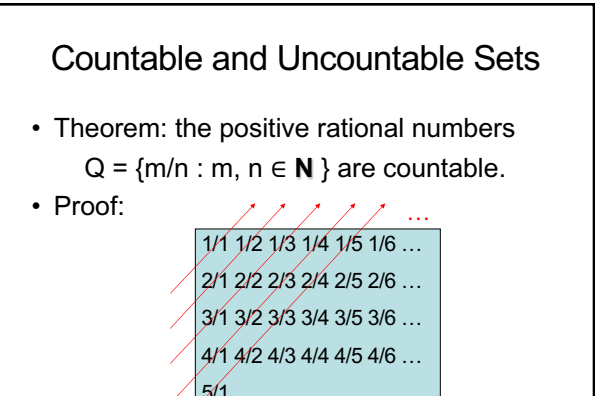
- the natural numbers $\mathbf{N} = \{1,2,3,\dots\}$ are **countable**
- Definition: a set S is **countable** if it is finite, or it is infinite and there is a bijection $f: \mathbf{N} \rightarrow S$

January 31, 2025 CS21 Lecture 11

3

Countable and Uncountable Sets

- Theorem: the positive rational numbers $\mathbf{Q} = \{m/n : m, n \in \mathbf{N}\}$ are countable.
- Proof:



1/1	1/2	1/3	1/4	1/5	1/6	...
2/1	2/2	2/3	2/4	2/5	2/6	...
3/1	3/2	3/3	3/4	3/5	3/6	...
4/1	4/2	4/3	4/4	4/5	4/6	...
5/1	...					

January 31, 2025 CS21 Lecture 11

4

Countable and Uncountable Sets

Theorem: the real numbers \mathbf{R} are NOT countable (they are “uncountable”).

- How do you prove such a statement?
 - assume countable (so there exists bijection f)
 - derive contradiction (some element not mapped to by f)
 - technique is called diagonalization (Cantor)

January 31, 2025 CS21 Lecture 11

5

Countable and Uncountable Sets

- Proof:
 - suppose \mathbf{R} is countable
 - list \mathbf{R} according to the bijection f :

n	$f(n)$
1	3.14159...
2	5.55555...
3	0.12345...
4	0.50000...
...	...

January 31, 2025 CS21 Lecture 11

6

Countable and Uncountable Sets

- Proof:
 - suppose \mathbf{R} is countable
 - list \mathbf{R} according to the bijection f :

n	$f(n)$
1	3.14159...
2	5.55555...
3	0.12345...
4	0.50000...
...	...

 set $x = 0.a_1a_2a_3a_4\dots$
 where digit $a_i \neq i^{\text{th}}$ digit after decimal point of $f(i)$ (not 0, 9)
 e.g. $x = 0.2312\dots$
x cannot be in the list!

January 31, 2025

CS21 Lecture 11

7

non-RE languages

Theorem: there exist languages that are not Recursively Enumerable.

Proof outline:

- the set of all TMs is **countable**
- the set of all languages is **uncountable**
- the function $L: \{\text{TMs}\} \rightarrow \{\text{languages}\}$ cannot be onto

January 31, 2025

CS21 Lecture 11

8

non-RE languages

- Lemma: the set of all TMs is **countable**.
- Proof:
 - each TM M can be described by a finite-length string $\langle M \rangle$
 - can enumerate these strings, and give the natural bijection with \mathbf{N}

January 31, 2025

CS21 Lecture 11

9

non-RE languages

- Lemma: the set of all languages is **uncountable**
- Proof:
 - fix an enumeration of all strings s_1, s_2, s_3, \dots (for example, lexicographic order)
 - a language L is described by its **characteristic vector** χ_L whose i^{th} element is 0 if s_i is not in L and 1 if s_i is in L

January 31, 2025

CS21 Lecture 11

10

non-RE languages

- suppose the set of all languages is countable
- list characteristic vectors of all languages according to the bijection f :

n	$f(n)$
1	0101010...
2	1010011...
3	1110001...
4	0100011...
...	...

January 31, 2025

CS21 Lecture 11

11

non-RE languages

- suppose the set of all languages is countable
- list characteristic vectors of all languages according to the bijection f :

n	$f(n)$
1	0101010...
2	1010011...
3	1110001...
4	0100011...
...	...

January 31, 2025

CS21 Lecture 11

12

So far...

$\{a^n b^n : n \geq 0\}$ (some language)
 regular languages
 context free languages
 $\{a^n b^n c^n : n \geq 0\}$ (some language)
 RE

- This language might be an esoteric, artificially constructed one. Do we care?
- We will show a natural undecidable L next.

January 31, 2025 CS21 Lecture 11

13

The Halting Problem

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

January 31, 2025 CS21 Lecture 11

14

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, 2025 CS21 Lecture 11

15

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, 2025 CS21 Lecture 11

16

The Halting Problem

Turing Machines
 inputs
 box (M, x): does M halt on x?
 The existence of H which tells us yes/no for each box allows us to construct a TM H' that cannot be in the table.
 H': n Y n Y Y n Y

January 31, 2025 CS21 Lecture 11

17

So far...

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

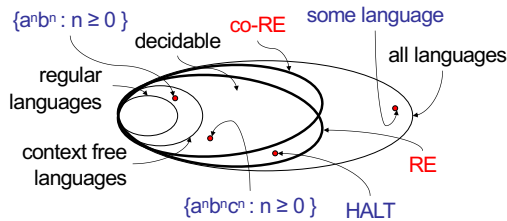
- Can we exhibit a natural language that is non-RE?

January 31, 2025 CS21 Lecture 11

18

RE and co-RE

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



January 31, 2025

CS21 Lecture 11

19

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, 2025

CS21 Lecture 11

20

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, 2025

CS21 Lecture 11

21

A natural non-RE language

Theorem: the complement of HALT is not recursively enumerable.

Proof:

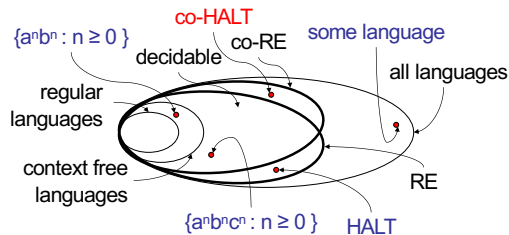
- we know that HALT is RE
- suppose complement of HALT is RE
- then HALT is co-RE
- implies HALT is decidable. Contradiction.

January 31, 2025

CS21 Lecture 11

22

Summary



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

January 31, 2025

CS21 Lecture 11

23

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, 2025

CS21 Lecture 11

24

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, 2025

CS21 Lecture 11

25

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, 2025

CS21 Lecture 11

26

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, 2025

CS21 Lecture 11

27

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, 2025

CS21 Lecture 11

28

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 $q_{\text{accept}}/q_{\text{reject}}$ 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, 2025

CS21 Lecture 11

29

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, 2025

CS21 Lecture 11

30