Computability and Complexity

Lecture 4

Reals are uncountable
Undecidability of the acceptance problem for TM
Complement of the acceptance problem for TM is unrecognizable

given by Jiri Srba

Bijection (Correspondence)

Definition

A function $f : A \to B$ is called bijection (correspondence) iff

- f is injective (one-to-one), i.e., $f(a) \neq f(b)$ whenever $a \neq b$, and
- f is surjective (onto), i.e., for every $b \in B$ there is $a \in A$ such that $f(a) = b$.

Examples:

- There is a bijection between natural numbers and even natural numbers.
- There is a bijection between natural and rational numbers.

Countable Sets

Definition

A set A is countable if it is either finite or there is a bijection between the set A and the set of natural numbers.

Facts:

- All even natural numbers are countable.
- The set of rational numbers is countable.

Countable Sets

Definition

A set A is countable if it is either finite or there is a bijection between the set A and the set of natural numbers.

Facts:

- All even natural numbers are countable.
- The set of rational numbers is countable.

Theorem

The set of real numbers is uncountable.

Proof: By contradiction via the diagonalization method.

There Exist Nonrecognizable Languages

Observation

There are only countably many Turing machines.

There Exist Nonrecognizable Languages

Observation

There are only countably many Turing machines.

Observation

There are uncountably many different languages.

There Exist Nonrecognizable Languages

Observation

There are only countably many Turing machines.

Observation

There are uncountably many different languages.

Conclusion

There exist languages that are not recognizable.

Acceptance Problem for Turing Machines

Problem: "Given a TM M and a string w, does M accept w?

Language Formulation (Acceptance Problem for TM)

$$A_{TM} \stackrel{\text{def}}{=} \{\langle M, w\rangle \mid \ M \text{ is a TM}, w \text{ a string, and } M \text{ accepts } w\}$$

Acceptance Problem for Turing Machines

Problem: "Given a TM M and a string w, does M accept w?

Language Formulation (Acceptance Problem for TM)

$$A_{TM} \stackrel{\text{def}}{=} \{\langle M, w\rangle \mid \ M \text{ is a TM, } w \text{ a string, and } M \text{ accepts } w\}$$

Theorem (Turing, 1936)

The language A_{TM} is undecidable.

Proof of Undecidability of A_{TM}

By contradiction assume that there is a decider H for A_{TM}:

$$H(\langle M, w\rangle) = \begin{cases} \underline{\text{accept}} & \text{if } M \text{ accepts } w \\ \underline{\text{reject}} & \text{if } M \text{ does not accept } w \end{cases}$$

Proof of Undecidability of A_{TM}

By contradiction assume that there is a decider H for A_{TM}:

$$H(\langle M, w \rangle) = \begin{cases} \underline{\text{accept}} & \text{if } M \text{ accepts } w \\ \underline{\text{reject}} & \text{if } M \text{ does not accept } w \end{cases}$$

From H we can build a machine D, which is also a decider:

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

1. Run H on $\langle M, \langle M \rangle \rangle$.
2. If H accepted then D <u>rejects</u>. If H rejected then D <u>accepts</u>."

Proof of Undecidability of A_{TM}

By contradiction assume that there is a decider H for A_{TM}:

$$H(\langle M, w\rangle) = \begin{cases} \underline{\text{accept}} & \text{if } M \text{ accepts } w \\ \underline{\text{reject}} & \text{if } M \text{ does not accept } w \end{cases}$$

From H we can build a machine D, which is also a decider:

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

1. Run H on $\langle M, \langle M\rangle\rangle$.
2. If H accepted then D rejects. If H rejected then D accepts."

What happens if we run D on $\langle D\rangle$?

1. D accepts $\langle D\rangle$, but then H rejected $\langle D, \langle D\rangle\rangle$ and hence D did not accept $\langle D\rangle$, contradiction!

Proof of Undecidability of A_{TM}

By contradiction assume that there is a decider H for A_{TM}:

$$H(\langle M, w\rangle) = \begin{cases} \underline{\text{accept}} & \text{if } M \text{ accepts } w \\ \underline{\text{reject}} & \text{if } M \text{ does not accept } w \end{cases}$$

From H we can build a machine D, which is also a decider:

> $D =$ "On input $\langle M\rangle$
> 1. Run H on $\langle M, \langle M\rangle\rangle$.
> 2. If H accepted then D rejects. If H rejected then D accepts."

What happens if we run D on $\langle D\rangle$?

1. D accepts $\langle D\rangle$, but then H rejected $\langle D, \langle D\rangle\rangle$ and hence D did not accept $\langle D\rangle$, contradiction!
2. D rejects $\langle D\rangle$, but then H accepted $\langle D, \langle D\rangle\rangle$ and hence D accepted $\langle D\rangle$, contradiction!

Proof of Undecidability of A_{TM}

By contradiction assume that there is a decider H for A_{TM}:

$$H(\langle M, w\rangle) = \begin{cases} \underline{\text{accept}} & \text{if } M \text{ accepts } w \\ \underline{\text{reject}} & \text{if } M \text{ does not accept } w \end{cases}$$

From H we can build a machine D, which is also a decider:

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

1. Run H on $\langle M, \langle M\rangle\rangle$.
2. If H accepted then D rejects. If H rejected then D accepts."

What happens if we run D on $\langle D\rangle$?

1. D accepts $\langle D\rangle$, but then H rejected $\langle D, \langle D\rangle\rangle$ and hence D did not accept $\langle D\rangle$, contradiction!
2. D rejects $\langle D\rangle$, but then H accepted $\langle D, \langle D\rangle\rangle$ and hence D accepted $\langle D\rangle$, contradiction!

So D cannot exist, so H cannot exist either (D was built from H).

This means that A_{TM} is undecidable. $\square$

The Language A_{TM} Is Recognizable

$$A_{TM} \stackrel{\text{def}}{=} \{\langle M, w\rangle \mid \ M \text{ is a TM, } w \text{ a string, and } M \text{ accepts } w\}$$

Theorem

The language A_{TM} is recognizable.

The Language A_{TM} Is Recognizable

$A_{TM} \stackrel{\text{def}}{=} \{\langle M, w\rangle \mid M \text{ is a TM, } w \text{ a string, and } M \text{ accepts } w\}$

Theorem

The language A_{TM} is recognizable.

Proof: Construct a recognizer U for A_{TM}.

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

1. Simulate M on w.
2. If M accepted then U <u>accepts</u>.
   If M rejected then U <u>rejects</u>."

The Language A_{TM} Is Recognizable

$A_{TM} \stackrel{\text{def}}{=} \{\langle M, w\rangle \mid$ M is a TM, w a string, and M accepts $w\}$

Theorem

The language A_{TM} is recognizable.

Proof: Construct a recognizer U for A_{TM}.

$U =$ "On input $\langle M, w\rangle$:
1. Simulate M on w.
2. If M accepted then U accepts.
   If M rejected then U rejects."

Remarks:

- The machine U (also called the universal TM) is not a decider.
- It is a "programmable" TM, think of $\langle M, w\rangle$ as a program with input data.

Co-Recognizable Languages

Intuition for a recognizable language L (recognized by TM M):

- If $w \in L$ then M run on w will halt in accept state.
- If $w \notin L$ then M run on w will reject or loop.

Co-Recognizable Languages

Intuition for a recognizable language L (recognized by TM M):

- If $w \in L$ then M run on w will halt in accept state.
- If $w \notin L$ then M run on w will reject or loop.

Intuition for a co-recognizable language L (recognized by TM M):

- If $w \notin L$ then M run on w will halt in accept state.
- If $w \in L$ then M run on w will reject or loop.

Co-Recognizable Languages

Intuition for a recognizable language L (recognized by TM M):

- If $w \in L$ then M run on w will halt in accept state.
- If $w \notin L$ then M run on w will reject or loop.

Intuition for a co-recognizable language L (recognized by TM M):

- If $w \notin L$ then M run on w will halt in accept state.
- If $w \in L$ then M run on w will reject or loop.

Definition (Co-Recognizable Language)

A language L is co-recognizable if $\overline{L}$ is recognizable.

Theorem

Theorem

A language is decidable if and only if it is recognizable and co-recognizable.

Proof:

Theorem

Theorem

A language is decidable if and only if it is recognizable and co-recognizable.

Proof:
"$\Rightarrow$": Every decidable language is recognizable and decidable languages are closed under complement.

Theorem

Theorem

A language is decidable if and only if it is recognizable and co-recognizable.

Proof:
"$\Rightarrow$": Every decidable language is recognizable and decidable languages are closed under complement.
"$\Leftarrow$": Assume that L is recognizable and co-recognizable. Hence there exists a recognizer M_1 for L and a recognizer M_2 for $\overline{L}$. We construct a decider M for L.
$M =$ "On input x:

1. Run M_1 on x and M_2 on x in parallel.
2. If M_1 accepted then M <u>accepts</u>.
   If M_2 accepted then M <u>rejects</u>."

Notice that M terminates on any input x as either $x \in L$ or $x \in \overline{L}$.

$\overline{A_{TM}}$ Is not Recognizable

Corollary

The language $\overline{A_{TM}}$ is co-recognizable but not recognizable.

$\overline{A_{TM}}$ Is not Recognizable

Corollary

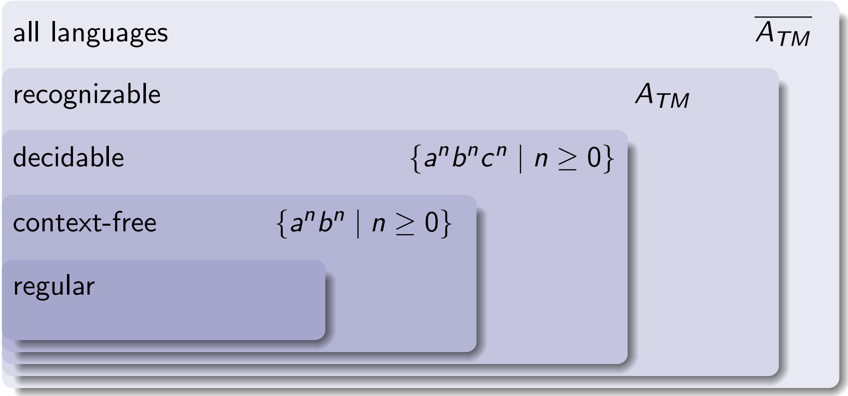
The language $\overline{A_{TM}}$ is co-recognizable but not recognizable.

Proof: Co-recognizability follows directly from the definition. Unrecognazibility is proved by contradiction:

- Assume that $\overline{A_{TM}}$ is recognizable.
- Because we know that A_{TM} is recognizable, our theorem implies that $\overline{A_{TM}}$ and A_{TM} are both decidable.
- But we know that A_{TM} is not decidable. This is a contradiction, hence $\overline{A_{TM}}$ cannot be recognizable.

$\square$

Overview (Strict Hierarchy of Language Classes)



Exam Questions

- The language A_{TM} and its undecidability (including proof).
- Definition of co-recognizable languages.
- Proof that A_{TM} is recognizable and $\overline{A_{TM}}$ is co-recognizable.
- Theorem that language if decidable iff it is recognizable and co-recognizable.
- The language $\overline{A_{TM}}$ is not recognizable.