An Undecidable Language

$A_{TM} = \{ <M, w> \mid M \text{ is a TM that accepts string } w \}$

- Turing-recognizable?
- Turing-decidable?

We will show that $A_{TM}$ is undecidable.

Acceptance$_{TM}$ ($TM, M, w$):

1. run $M$ on $w$
2. if $M$ accepts, then accept

($T$-recognizable bec. we will accept if $M$ accepts $w$ but we do not know how to reject if $M$ goes into an infinite computation on $w$)
A bit about infinite sets and their sizes (diagonalization):

Def 4.12: Let $A,B$ be sets and let $f:A \to B$. We say that $f$ is
- **one-to-one** if $f(a) \neq f(b)$ for every $a \neq b$
- **onto** if for every $b \in B$ there exists $a \in A$ such that $f(a) = b$

If $f$ is one-to-one and onto, then $A,B$ are the **same size** and $f$ is called **correspondence**.

Example: \( \mathbb{N} = \{1,2,3,4,5,\ldots\} \) and \( \{2,4,6,8,\ldots\} \)}
A bit about infinite sets and their sizes (diagonalization):

**Def 4.12:** Let \( A, B \) be sets and let \( f : A \rightarrow B \). We say that \( f \) is
- **one-to-one** if \( f(a) \neq f(b) \) for every \( a \neq b \)
- **onto** if for every \( b \in B \) there exists \( a \in A \) such that \( f(a) = b \)

If \( f \) is one-to-one and onto, then \( A, B \) are the **same size** and \( f \) is called **correspondence**.

**Example:** \( \mathcal{N} = \{1,2,3,4,5,...\} \) and \( \{2,4,6,8,...\} \)

**Def 4.14:** A set is **countable** if it is finite or has the same size as \( \mathcal{N} \).
An Undecidable Language

[Section 4.2]

Are \( \mathbb{Q} \) (rational numbers) and \( \mathbb{R} \) (real numbers) countable?
Cor 4.18: There is a language that is not Turing-recognizable.
An Undecidable Language

Thm 4.11: $A_{TM}$ is not decidable.

Recall: $A_{TM} = \{ <M,w> | M \text{ is a TM that accepts string } w \}$

By contradiction, assume $A_{TM}$ is decidable. Then, there is an algorithm for the question "Does $M$ accept $w$?". Let $H$ be such an algo.

\[ H(<M,w>): \]
- accept if $M$ accepts $w$
- reject if $M$ does not accept $w$

Let us create the following TM: $D(<M>)$ on input TM $M$:

1. Run $H$ on $<M,<M>)$
2. if $H(<M,<M>)$ accepts, then reject
3. else (i.e. $H(<M,<M>)$ rejects), then accept

Let's run:

\[ D(<D>): \]
- accept if $H(<D,<D>)$ rejects iff $D$ does not accept $<D$
- reject if $H(<D,<D>)$ accepts iff $D$ accepts $<D$

\[ \Box \]
Thm 4.22: A language $L$ is decidable iff $L$ is Turing-recognizable and $\overline{L}$ is Turing-recognizable (we say that $L$ is co-Turing-recognizable).

\[ \Rightarrow \quad \text{if } L \text{ is decidable, then } \exists \text{ a TM-decider } M \text{ for } L, \text{ thus } M \text{ is also a TM, i.e. } L \text{ is T-recognizable.} \]

\[ \Leftarrow \quad \text{if } L \text{ and } \overline{L} \text{ are both T-recognizable, then we can create a TM-decider } M \text{ for } L: \]

Let $M_1$ be a TM for $L$, then $M$: run $M_1$ and $M_2$ in parallel (both on input $w$)

If $M_1$ accepts, accept
If $M_2$ accepts, reject

Cor 4.23: $A_{TM}$ is not Turing-recognizable.

Otherwise, since $A_{TM}$ is T-recognizable, $A_{TM}$ would be decidable!