The Halting Problem

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

- Turing-recognizable?  - \textbf{YES} \\
- Turing-decidable?  \\
  - No, we'll see why \\

with a function-type view:

\[
\begin{align*}
\text{def} & \quad \text{acceptance} \ (\text{fnc M, string w}): \\
& \quad \text{return} \quad M(w)
\end{align*}
\]

i.e. the TM for \( A_{TM} \) simulates \( M \) on \( w \),

if \( M \) accepts, this TM accepts
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: \( \mathbb{N} = \{1, 2, 3, 4, 5, \ldots\} \) and \( \{2, 4, 6, 8, \ldots\} \)

\[ f(n) = 2n \]
The Halting Problem

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:** $\mathbb{N} = \{1, 2, 3, 4, 5, \ldots\}$ and $\{2, 4, 6, 8, \ldots\}$

**Def 4.14:** A set is **countable** if it is finite or has the same size as $\mathbb{N}$. 
The Halting Problem

Are \( \mathbb{Q} \) (rational numbers) and \( \mathbb{R} \) (real numbers) countable?

\( \mathbb{Q} \) can be ordered, i.e., it is countable.

\( \mathbb{R} \) by contradiction, suppose there is an ordering of the numbers in \( \mathbb{R} \).

E.g.,

\[
\begin{align*}
0.\hspace{1em}000123 & \hspace{1em} 0.1341 \ldots \\
0.\hspace{1em}1201395 & \\
1.235 & \\
2.106 & \\
& \\
& \\
&
\end{align*}
\]

For the \( i \)-th number, look at the \( i \)-th position after the decimal point, choose a different digit, create a number of all these digits.

\( \) differs from all listed numbers.
Cor 4.18: There is a language that is not Turing-recognizable.

We will show that the set of all languages is uncountable.

Lexicographical ordering:

$L_1 = \{ \epsilon, a, aa \}$
$L_2 = \{ a, a^2, a^3 \}$
$L_3 = \{ \epsilon, a^4, a^5, a^6 \}$

Remark:

Any one language is countable because lexicographic ordering.

Create $L$ s.t. looking at the $i$-th language, the $i$-th string:

- If the string is in $L_i$, we do not include it in $L$.
- If it is not in $L_i$, we include it in $L$.

Now $L$ differs from all $L_i$'s $\Rightarrow$ not on the list $\nexists$.

Notice that every TM can be described (encoded) as a string $\Rightarrow$ #TM is countable.

#Languages is uncountable.
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 that $A_{TM}$ is decidable.

Argument with functions:
- We have a function $\text{acceptance}(M,w)$ that returns true/false.
  - returns true if $M(w)$ returns true.
  - returns false otherwise (i.e., if $M(w)$ returns false, or keeps running).
- Note: acceptance cannot simulate $M$, it is assumed to be doing some magic.

```haskell
def crazy (func X):
    run acceptance (X,X)
    return true if returns false
    return false if true
```

we can not have a func for acceptance
(undecidable)
**Thm 4.22:** A language $L$ is decidable iff $L$ is Turing-recognizable and $L^c$ is Turing-recognizable (we say that $L$ is **co-Turing-recognizable**).

Let $M_1$ be a TM for $L$, 
$M_2$ for $L^c$

Then we create $M$ that simulates $M_1, M_2$ simultaneously (e.g. on 2 tapes)
- when $M_2$ accepts, $M$ accepts
- $M_2$ accepts, $M$ rejects

For any input, either $M_2$ or $M_2$ accepts.

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

Recall $A_{TM}$ is T-recognizable and undecidable, thus $A_{TM}^c$ cannot be T-recognizable.
because if it were then $A_{TM}$ decidable by Thm 4.22