

The Halting problem

- **Definition:** $A_{\text{TM}} = \{\langle M, w \rangle \mid M \text{ accepts } w\}$
- **Theorem:** $\overline{A_{\text{TM}}}$ is not recognizable
- **Proof:** Suppose, for contradiction, that a TM $M_{\overline{A_{\text{TM}}}}$ recognizes $\overline{A_{\text{TM}}}$. We will use $M_{\overline{A_{\text{TM}}}}$ to define a TM M_{DIAG} that recognizes DIAG, which is a contradiction since DIAG is not recognizable. Define $M_{\text{DIAG}} =$ “on input $\langle M \rangle$...
 1. Run $M_{\overline{A_{\text{TM}}}}$ on input $\langle M, M \rangle$
 2. If $M_{\overline{A_{\text{TM}}}}$ accepts then accept
 3. If $M_{\overline{A_{\text{TM}}}}$ rejects then reject

We need to show that $L(M_{\text{DIAG}}) = \text{DIAG}$.

$$\begin{aligned}
 M_{\text{DIAG}} \text{ accepts } \langle M \rangle &\iff M_{\overline{A_{\text{TM}}}} \text{ accepts } \langle M, M \rangle \\
 &\iff \langle M, M \rangle \in \overline{A_{\text{TM}}} \\
 &\iff M \text{ does not accept } \langle M \rangle \\
 &\iff \langle M \rangle \in \text{DIAG}
 \end{aligned}$$

So $M_{\text{DIAG}} \text{ accepts } \langle M \rangle \iff \langle M \rangle \in \text{DIAG}$, and thus $L(M_{\text{DIAG}}) = \text{DIAG}$.

- **Corollary:** A_{TM} is not decidable.
- **Proof:** the complement of a decidable language is decidable; so if A_{TM} is decidable then so is $\overline{A_{\text{TM}}}$. But we just proved that $\overline{A_{\text{TM}}}$ is not even recognizable, so A_{TM} must not be decidable.
- **Theorem:** A_{TM} is recognizable.
- **Proof:** $A_{\text{TM}} = L(U)$, where U is the universal TM.
- So A_{TM} is recognizable but not decidable; so we have three (non-empty) categories of languages:
 - Decidable languages (e.g. $\{0^n 1^n \mid n \in \mathbb{N}\}$)
 - Languages that are recognizable but not decidable (e.g. A_{TM})
 - Languages that are not recognizable (e.g. DIAG or $\overline{A_{\text{TM}}}$)
- **Definition:** $\text{HALT} = \{\langle M, w \rangle \mid M \text{ halts on input } w\}$
- **Theorem:** HALT is not decidable
- **Proof:** Suppose, for contradiction, that a TM M_{HALT} decides HALT. We will use M_{HALT} to construct a TM $M_{A_{\text{TM}}}$ that decides A_{TM} . This is a contradiction since A_{TM} is not decidable. Define $M_{A_{\text{TM}}} =$ “on input $\langle M, w \rangle$...

1. Modify M so that instead of rejecting, it goes into an infinite loop
2. Call the modified machine M'
3. Run M_{HALT} on input $\langle M', w \rangle$
4. If M_{HALT} accepts then accept
5. If M_{HALT} rejects then reject

We need to show two things: 1) That $M_{A_{\text{TM}}}$ halts on every input; and 2) That $L(M_{A_{\text{TM}}}) = A_{\text{TM}}$.

1. Every step of $M_{A_{\text{TM}}}$ ends (since in particular M_{HALT} always halts). The machine M' can be obtained by a simple modification that replaces every transition that leads to q_{reject} with a transition leading to a new state q_{loop} , so that every transition from q_{loop} moves right and returns to state q_{loop} .
2. Consider which strings are accepted by $M_{A_{\text{TM}}}$.

$$\begin{aligned}
 M_{A_{\text{TM}}} \text{ accepts } \langle M, w \rangle &\iff M_{\text{HALT}} \text{ accepts } \langle M', w \rangle \\
 &\iff M' \text{ halts on input } w \\
 &\iff M \text{ accepts input } w \\
 &\iff \langle M, w \rangle \in A_{\text{TM}}
 \end{aligned}$$

Reductions

- **Definition:** Let A, B be languages. Then $A \leq_m B$ ("A reduces to B") if there exists a computable function $f : \Sigma^* \mapsto \Sigma^*$ such that

$$\forall x \in \Sigma^* : x \in A \iff f(x) \in B$$

- Recall that a function f is computable if there is a Turing machine that, on input x , halts with $f(x)$ on its tape. In terms of high-level descriptions, it is enough to describe an algorithm that, on input x , returns $f(x)$.
- **Theorem:** If $A \leq_m B$ and A is undecidable, then B is undecidable.
- **Proof:** Assume that $A \leq_m B$ and A is undecidable, and for contradiction assume that B is decidable. Let M_B decide B , and let $f : \Sigma^* \mapsto \Sigma^*$ be a computable function such that $x \in A \iff f(x) \in B$ for all $x \in \Sigma^*$. We will use M_B to construct a TM M_A that decides A , which is a contradiction since we assumed that A is undecidable. Define $M_A =$ " on input $x \dots$
 1. Compute $y = f(x)$
 2. Run M_B on input y
 3. If M_B accepts then accept
 4. If M_B rejects then reject

Each stage of this algorithm terminates, since M_B always halts and f is computable (note: saying "compute $y = f(x)$ " is shorthand for saying "run M_f on input x and store its output, where M_f is the TM that computes f "). We need to show that $L(M_A) = A$.

$$\begin{aligned}M_A \text{ accepts } x &\iff M_B \text{ accepts } f(x) \\ &\iff f(x) \in B \\ &\iff x \in A\end{aligned}$$

So M_A decides A , contradicting the assumption that A is undecidable.

- **Theorem:** If $A \leq_m B$ and A is unrecognizable, then B is unrecognizable
- **Proof:** The proof is nearly identical to the proof of the previous theorem.