Undecidability

Oliver Chubet, October 30, 2024

Decidable Languages

Decidable Languages

A language L is **decidable** if there exists a TM M such that

Decidable Languages

A language L is **decidable** if there exists a TM M such that

1. M accepts an input x **iff** $x \in L$ **and**

Decidable Languages

A language L is **decidable** if there exists a TM M such that

1. M accepts an input x **iff** $x \in L$ **and**

2. M rejects an input x **iff** $x \notin L$**.**

Decidable Languages

A language L is **decidable** if there exists a TM M such that

1. M accepts an input x **iff** $x \in L$ **and**
2. M rejects an input x **iff** $x \notin L$.

Decidable Languages

A language L is **decidable** if there exists a TM M such that

1. M accepts an input x **iff** $x \in L$ **and**
2. M rejects an input x **iff** $x \notin L$.

Or Equivalently:

Decidable Languages

A language L is **decidable** if there exists a TM M such that

1. M accepts an input x **iff** $x \in L$ **and**
2. M rejects an input x **iff** $x \notin L$**.**

Or Equivalently:

M *halts on every input and accepts an input* x ***iff*** $x \in L$*.*

The Language A_{TM}

The Language A_{TM}

Q: *Is the language of decidable languages decidable?*

The Language A_{TM}

Q: *Is the language of decidable languages decidable?*

… [crickets] …

The Language A_{TM}

Q: *Is the language of decidable languages decidable?*

… [crickets] …

Q: *How do I know if a TM M accepts an input x?*

The Language A_{TM}

Q: *Is the language of decidable languages decidable?*

… [crickets] …

Q: *How do I know if a TM* M *accepts an input* x*?*

Q: *Is the language of pairs* (M, x) *such that* M *accepts* x *decidable?*

The Language A_{TM}

Q: *Is the language of decidable languages decidable?*

… [crickets] …

Q: *How do I know if a TM M accepts an input x?*

Q: *Is the language of pairs (M, x) such that M accepts x decidable?*

Idea: *Write a program to simulate M running on input x.*

The Language A_{TM}

$$A_{TM} = \{(M, x) \mid M \text{ accepts } x\}$$

The Universal Turing Machine

(A TM that simulates TMs)

The Universal Turing Machine

(A TM that simulates TMs)

Input: Turing machine M and input x

The Universal Turing Machine
(A TM that simulates TMs)

Input: Turing machine M and input x

- TMs take string input, so we need to encode M as a string

The Universal Turing Machine

(A TM that simulates TMs)

Input: Turing machine M and input x

- TMs take string input, so we need to encode M as a string
- There are many encoding possibilities

The Universal Turing Machine
(A TM that simulates TMs)

Input: Turing machine M and input x

- TMs take string input, so we need to encode M as a string
- There are many encoding possibilities
- (Similar to the binary for a computer program)

Is A_{TM} Turing recognizable?

Is A_{TM} Turing recognizable?

On input $\langle M, x \rangle$:

1. Simulate M on x.
2. If M accepts, **accept**. If M rejects, **reject**.

Is A_{TM} Turing recognizable?

On input $\langle M, x \rangle$:

1. Simulate M on x.
2. If M accepts, **accept**. If M rejects, **reject**.

(Ie. Use the Universal Turing machine.)

Is A_{TM} decidable?

Is A_{TM} decidable?

If it was, then…

Is A_{TM} decidable?

If it was, then…

… we would have a TM H that decides it; that is,

Is A_{TM} decidable?

If it was, then…

… we would have a TM H that decides it; that is,

On input (M, x):

- H accepts if M accepts x
- H rejects if M does not accept x

A Diagonalization using H

A Diagonalization using H

Define TM D that simulates H running a TM on the encoding of itself and outputs the opposite of H.

A Diagonalization using H

Define TM D that simulates H running a TM on the encoding of itself and outputs the opposite of H.

On input $\langle M \rangle$:

1. Run H on input $\langle M, \langle M \rangle \rangle$
2. If H accepts, **reject**. If H rejects, **accept**.

A Diagonalization using H

Define TM D that simulates H running a TM on the encoding of itself and outputs the opposite of H.

On input $\langle M \rangle$:

1. Run H on input $\langle M, \langle M \rangle \rangle$
2. If H accepts, **reject**. If H rejects, **accept**.

Q: *What if we run D on the description of itself?*

Running $D(\langle D \rangle)$

Running $D(\langle D \rangle)$

Suppose D **accepts** $\langle D \rangle$:

Running $D(\langle D \rangle)$

Suppose D **accepts** $\langle D \rangle$:

- Then H accepts $\langle D, \langle D \rangle \rangle$…

Running $D(\langle D \rangle)$

Suppose D **accepts** $\langle D \rangle$:

- Then H accepts $\langle D, \langle D \rangle \rangle$…
- *But that means that* D *should reject* $\langle D \rangle$*!!!*

Running $D(\langle D \rangle)$

Suppose D **accepts** $\langle D \rangle$:

- Then H accepts $\langle D, \langle D \rangle \rangle$…
- *But that means that* D *should reject* $\langle D \rangle$*!!!*

Suppose D **rejects** $\langle D \rangle$:

Running $D(\langle D \rangle)$

Suppose D **accepts** $\langle D \rangle$:

- Then H accepts $\langle D, \langle D \rangle \rangle$…
- *But that means that* D *should reject* $\langle D \rangle$*!!!*

Suppose D **rejects** $\langle D \rangle$:

- Then H rejects $\langle D, \langle D \rangle \rangle$…

Running $D(\langle D \rangle)$

Suppose D **accepts** $\langle D \rangle$:

- Then H accepts $\langle D, \langle D \rangle \rangle$…
- *But that means that* D *should reject* $\langle D \rangle$*!!!*

Suppose D **rejects** $\langle D \rangle$:

- Then H rejects $\langle D, \langle D \rangle \rangle$…
- *But that means that* D *should accept* $\langle D \rangle$*!!!*

Running $D(\langle D \rangle)$

Suppose D **accepts** $\langle D \rangle$:

- Then H accepts $\langle D, \langle D \rangle \rangle$…
- *But that means that* D *should reject* $\langle D \rangle$*!!!*

Suppose D **rejects** $\langle D \rangle$:

- Then H rejects $\langle D, \langle D \rangle \rangle$…
- *But that means that* D *should accept* $\langle D \rangle$*!!!*

Running $D(\langle D \rangle)$

Suppose D **accepts** $\langle D \rangle$:

- Then H accepts $\langle D, \langle D \rangle \rangle$…
- *But that means that* D *should reject* $\langle D \rangle$*!!!*

Suppose D **rejects** $\langle D \rangle$:

- Then H rejects $\langle D, \langle D \rangle \rangle$…
- *But that means that* D *should accept* $\langle D \rangle$*!!!*

Therefore A_{TM} ***must not be decidable…***

The Halting Problem

$$L_{HALT} = \{(M, x) \mid M \text{ halts on input } x\}$$

Is $HALT$ Turing recognizable?

Is $HALT$ Turing recognizable?

On input $\langle M, x \rangle$:

1. Simulate M on x.
2. If M accepts, **accept**. If M rejects, **accept**.

Is $HALT$ Turing recognizable?

On input $\langle M, x \rangle$:

1. Simulate M on x.
2. If M accepts, **accept**. If M rejects, **accept**.

(Ie. Use the Universal Turing machine.)

Is $HALT$ decidable?

Is $HALT$ decidable?

Suppose so… can we use the TM that decides $HALT$ to decide A_{TM}?