5CCS2FC2: Foundations of Computing II

The Limits of Computation

Week 10

Dr Christopher Hampson

Department of Informatics

King's College London

Undecidable Problems

Undecidable Problems

Theorem There is a language L that is *undecidable*.

Proof:

Step 1) First, note that we can **enumerate** all words over our chosen alphabet

$\varepsilon,\ 1,\ 0,\ 00,\ 01,\ 10,\ 11,\ 000,\ 001,\ 010,\ 011,\ 100,\ \ldots$

Undecidable Problems

Theorem There is a language L that is *undecidable*.

Proof:

Step 1) First, note that we can **enumerate** all words over our chosen alphabet

$$w_0,\ w_1,\ w_2,\ w_3,\ w_4,\ w_5,\ w_6,\ w_7,\ w_8,\ w_9,\ w_{10},\ w_{11},\ \ldots$$

Undecidable Problems

Theorem There is a language L that is *undecidable*.

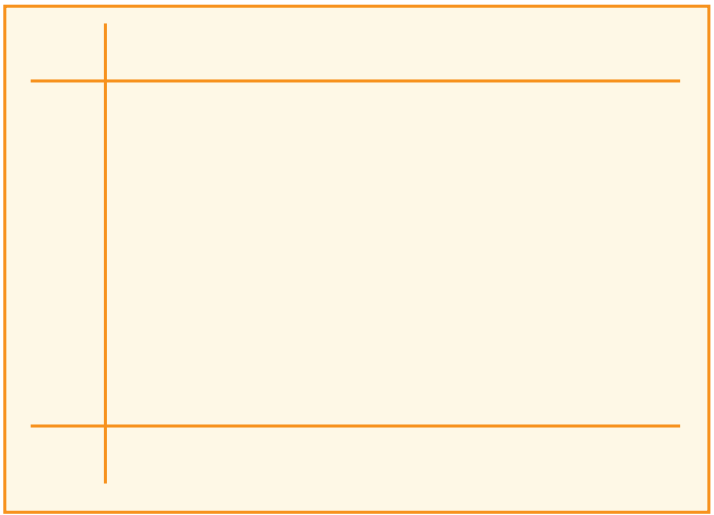
Proof:

Step 2) We can also **enumerate** every possible Turing Machine

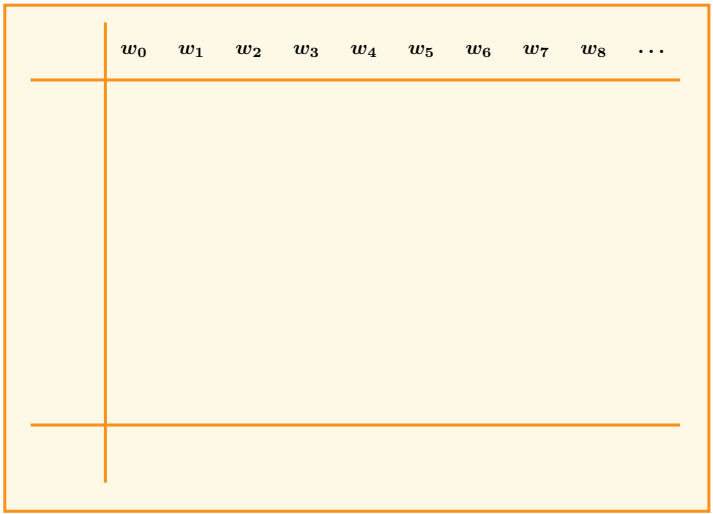
(there can be only as many machines as there are *codes* for machines)

$$M_0, \quad M_1, \quad M_2, \quad M_3, \quad M_4, \quad M_5, \quad M_6, \quad M_7, \quad M_8, \quad \ldots$$

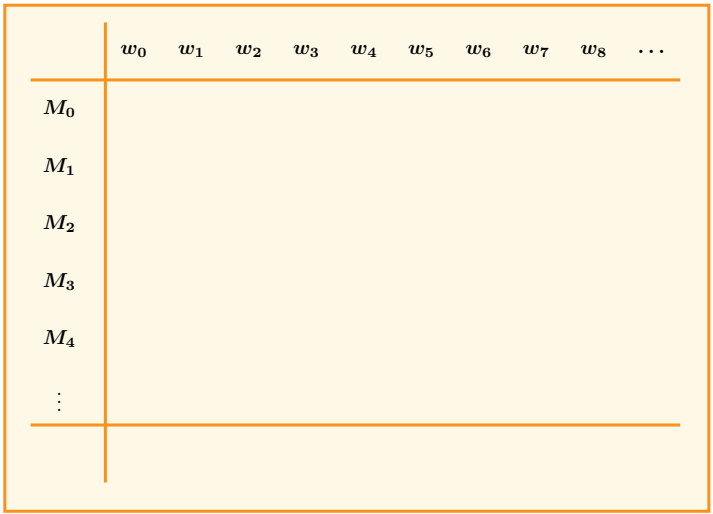
Undecidable Problems



Undecidable Problems



Undecidable Problems



Undecidable Problems

	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	$\cdots$
M_0	✓	✗	✓	✓	✗	✓	✗	✗	✓	$\cdots$
M_1	✗	✗	✓	✗	✗	✓	✓	✗	✗	$\cdots$
M_2	✓	✗	✗	✓	✓	✗	✓	✓	✗	$\cdots$
M_3	✗	✓	✓	✓	✗	✓	✗	✗	✓	$\cdots$
M_4	✗	✓	✗	✓	✗	✗	✗	✓	✗	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Undecidable Problems

	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	$\cdots$
M_0	✓	✗	✓	✓	✗	✓	✗	✗	✓	$\cdots$
M_1	✗	✗	✓	✗	✗	✓	✓	✗	✗	$\cdots$
M_2	✓	✗	✗	✓	✓	✗	✓	✓	✗	$\cdots$
M_3	✗	✓	✓	✓	✗	✓	✗	✗	✓	$\cdots$
M_4	✗	✓	✗	✓	✗	✗	✗	✓	✗	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
	✓	✗	✗	✓	✗	✗	✓	✓	✗	$\cdots$

Undecidable Problems

	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	$\cdots$
M_0	✓	✗	✓	✓	✗	✓	✗	✗	✓	$\cdots$
M_1	✗	✗	✓	✗	✗	✓	✓	✗	✗	$\cdots$
M_2	✓	✗	✗	✓	✓	✗	✓	✓	✗	$\cdots$
M_3	✗	✓	✓	✓	✗	✓	✗	✗	✓	$\cdots$
M_4	✗	✓	✗	✓	✗	✗	✗	✓	✗	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
	✗	✓	✓	✗	✓	✓	✗	✗	✓	$\cdots$

Undecidable Problems

	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	$\cdots$
M_0	✓	✗	✓	✓	✗	✓	✗	✗	✓	$\cdots$
M_1	✗	✗	✓	✗	✗	✓	✓	✗	✗	$\cdots$
M_2	✓	✗	✗	✓	✓	✗	✓	✓	✗	$\cdots$
M_3	✗	✓	✓	✓	✗	✓	✗	✗	✓	$\cdots$
M_4	✗	✓	✗	✓	✗	✗	✗	✓	✗	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
		w_1	w_2		w_4	w_5			w_8	$\cdots$

Undecidable Problems

	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	$\cdots$
M_0	✓	✗	✓	✓	✗	✓	✗	✗	✓	$\cdots$
M_1	✗	✗	✓	✗	✗	✓	✓	✗	✗	$\cdots$
M_2	✓	✗	✗	✓	✓	✗	✓	✓	✗	$\cdots$
M_3	✗	✓	✓	✓	✗	✓	✗	✗	✓	$\cdots$
M_4	✗	✓	✗	✓	✗	✗	✗	✓	✗	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
$L = \{$		$w_1,$	$w_2,$		$w_4,$	$w_5,$			$w_8,$	$\cdots$ $\}$

Undecidable Problems

$$L = \{ w_1, w_2, w_4, w_5, w_8, \ldots \}$$

Step 3) Let L denote the set of words w_i not accepted by the corresponding machine M_i

$$L = \{w_i \in \Sigma^* : M_i \text{ does not accept } w_i\}$$

Q.E.D.

Step 3) Let L denote the set of words w_i not accepted by the corresponding machine M_i

$$w_i \in L \iff M_i \text{ does not accept } w_i$$

Q.E.D.

Step 4) If L were **decidable** then it would be the language of some **terminating** machine M_j from our list. However,

$$w_j \in \textbf{Language}\,(M_j) \iff w_j \notin L$$

Q.E.D.

Step 4) If L were **decidable** then it would be the language of some **terminating** machine M_j from our list. However,

$$\text{Language}\,(M_j) \neq L$$

Q.E.D.

The Halting Problem

The Halting Problem

The Halting Problem $\mathbf{HALT}_{TM}$

Input) — An encoding of a Machine $\mathbf{code}(M)$,
— an input word $w \in \Sigma^*$,

Output) **True** if and only if M terminates on input w.

$$\mathbf{HALT}_{TM} = \{ \langle \mathbf{code}(M), w \rangle \ : \ M \text{ terminates on input } w \}$$

The Halting Problem

Theorem The Halting Problem $\mathbf{HALT}_{TM}$ is Undecidable.

Proof:

Step 1) Suppose, to that contrary that there is some **sound**, **complete** and **terminating** algorithm **halt** such that

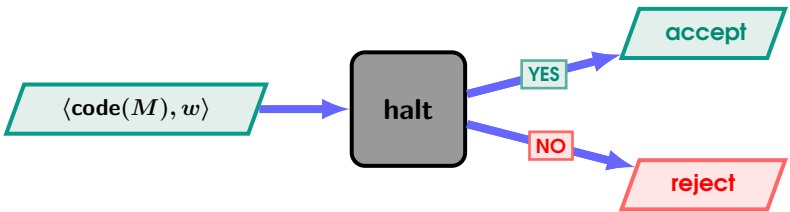
$$\mathbf{halt}\left(\langle \mathbf{code}(M), w\rangle\right) = \begin{cases} 1 & \text{if } M(w) = 1 \text{ or } M(w) = 0 \\ 0 & \text{if } M(w) = \uparrow \end{cases}$$

The Halting Problem

Theorem The Halting Problem $\mathbf{HALT}_{TM}$ is Undecidable.

Proof:

Step 1) Suppose, to that contrary that there is some **sound**, **complete** and **terminating** algorithm **halt** such that

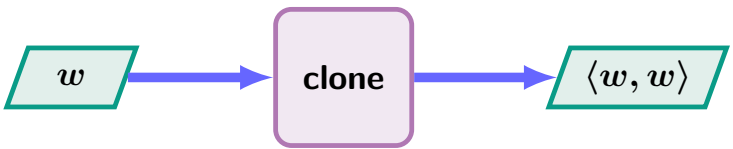


The Halting Problem

Theorem The Halting Problem $\mathbf{HALT}_{TM}$ is Undecidable.

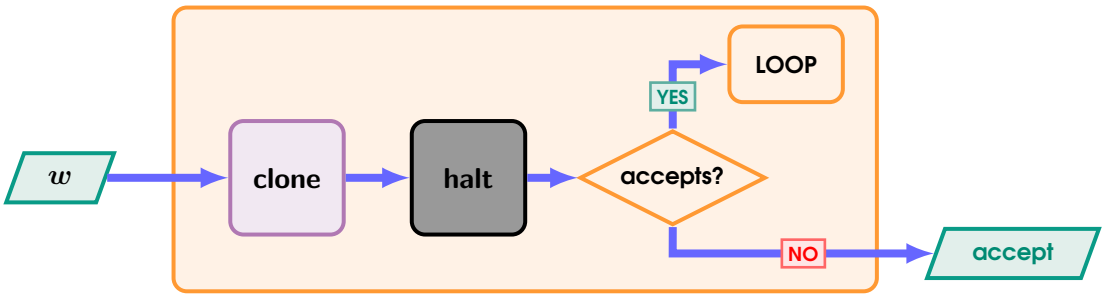
Proof:

Step 2) First, construct a **computable function** **clone** that takes a single input w and outputs the encoding of the pair $\langle w, w \rangle$ containing **two copies** of the input word



The Halting Problem

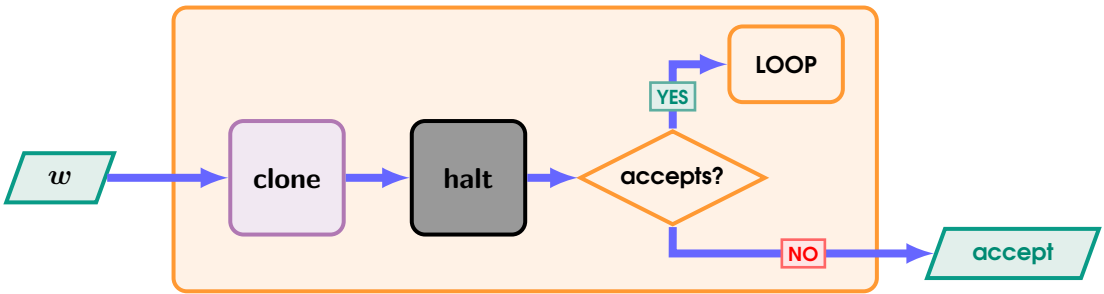
Step 3) We construct a **new machine** $M^\star$ as follows:



$$M^\star(w) = 1 \iff \textbf{halt}\left(\textbf{clone}(w)\right) = 0$$

The Halting Problem

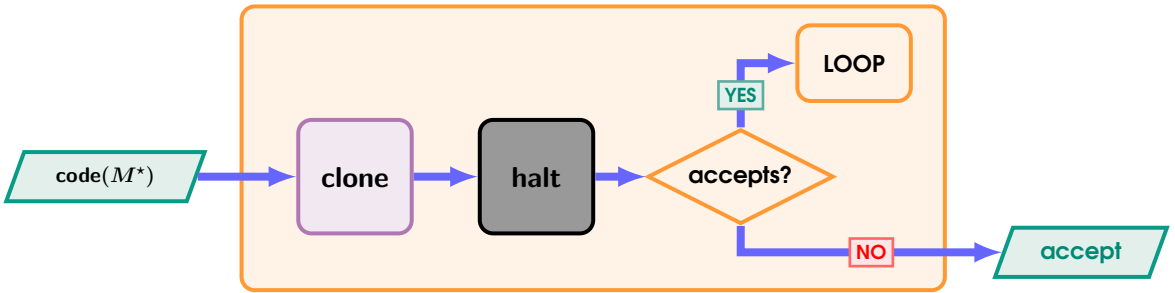
Step 3) We construct a **new machine** $M^\star$ as follows:



$$M^\star(w) = 1 \iff \textbf{halt}\left(\langle w, w\rangle\right) = 0$$

The Halting Problem

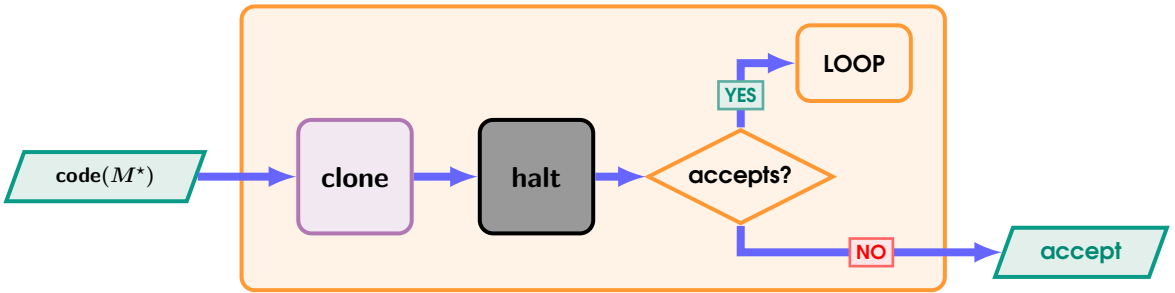
Step 4) But what if we use $\mathbf{code}(M^\star)$ as the input string in place of w?



$$M^\star\left(\mathbf{code}(M^\star)\right) = 1 \iff \mathbf{halt}\left(\langle \mathbf{code}(M^\star), \mathbf{code}(M^\star)\rangle\right) = 0$$

The Halting Problem

Step 4) But what if we use $\mathbf{code}(M^\star)$ as the input string in place of w?



$$M^\star\left(\mathbf{code}(M^\star)\right) = 1 \iff M^\star\left(\mathbf{code}(M^\star)\right) = \uparrow$$

The Halting Problem

Step 5) **Contradiction!** Hence our assumption that **halt** exists must be wrong!

Q.E.D.

Other Undecidable Problems

Other Undecidable Problems

- Some other Undecidable Languages
  - The Accepting Problem

$$\mathbf{A}_{TM} = \{ \langle \mathbf{code}(M), w \rangle : M(w) = 1 \}$$

  - The Rejecting Problem

$$\mathbf{R}_{TM} = \{ \langle \mathbf{code}(M), w \rangle : M(w) = 0 \}$$

Other Undecidable Problems

- Some other Undecidable Languages
  - The Emptiness Problem

$$\mathbf{E}_{TM} = \{ \textbf{code}(M) : \textbf{Language}(M) = \emptyset \}$$

  - The Equivalence Problem

$$\mathbf{EQ}_{TM} = \{ \langle \textbf{code}(M_1), \textbf{code}(M_2) \rangle : \textbf{Language}(M_1) = \textbf{Language}(M_2) \}$$

Other Undecidable Problems

The Accepting Problem A_{TM}

Input) — An encoding of a Machine $\mathbf{code}(M)$,
— an input word $w \in \Sigma^*$,

Output) **True** if and only if M accepts the input w.

$$\mathbf{A}_{TM} = \{ \langle \mathbf{code}(M), w \rangle : M(w) = 1 \}$$

Other Undecidable Problems

The Rejecting Problem $\mathbf{R}_{TM}$

Input) — An encoding of a Machine $\mathbf{code}(M)$,
— an input word $w \in \Sigma^*$,

Output) **True** if and only if M rejects the input w.

$$\mathbf{R}_{TM} = \{ \langle \mathbf{code}(M), w \rangle : M(w) = 0 \}$$

Other Undecidable Problems

The Emptiness Problem E_{TM}

Input) An encoding of a Machine $\textbf{code}(M)$,

Output) **True** if and only if **Language** $(M) = \emptyset$.

$$E_{TM} = \{\textbf{code}(M) : \textbf{Language}(M) = \emptyset\}$$

Other Undecidable Problems

The Equivalence Problem EQ_{TM}

Input) — An encoding of a Machine $\mathbf{code}(M_1)$,
— An encoding of a second Machine $\mathbf{code}(M_2)$,

Output) **True** if and only if M_1 and M_2 accept the same language.

$$\mathbf{EQ}_{TM} = \{ \langle \mathbf{code}(M_1), \mathbf{code}(M_2) \rangle : \mathbf{Language}(M_1) = \mathbf{Language}(M_2) \}$$

Other Undecidable Problems

The Regular Language Problem REGULAR$_{TM}$

Input) An encoding of a Machine **code**(M),

Output) **True** if and only if **Language** (M) is regular.

REGULAR$_{TM}$ $=$ $\{$**code**(M) : **Language** (M) is regular $\}$

Other Undecidable Problems

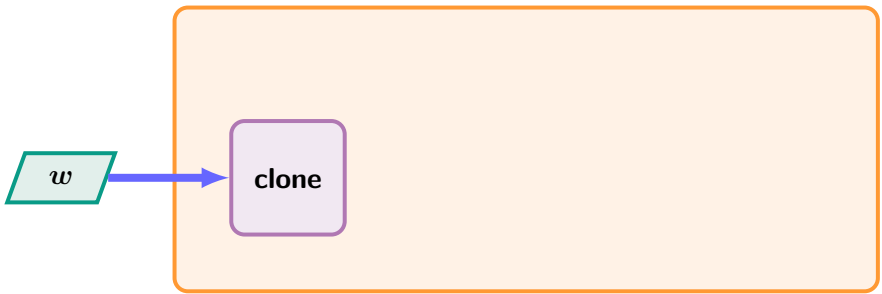
Step 3) We construct a **new machine** $M^\star$ as follows:

Other Undecidable Problems

Step 3) We construct a **new machine** $M^\star$ as follows:

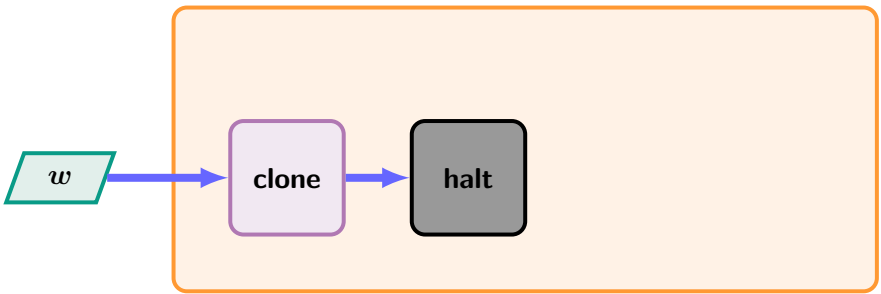
Other Undecidable Problems

Step 3) We construct a **new machine** $M^{\star}$ as follows:



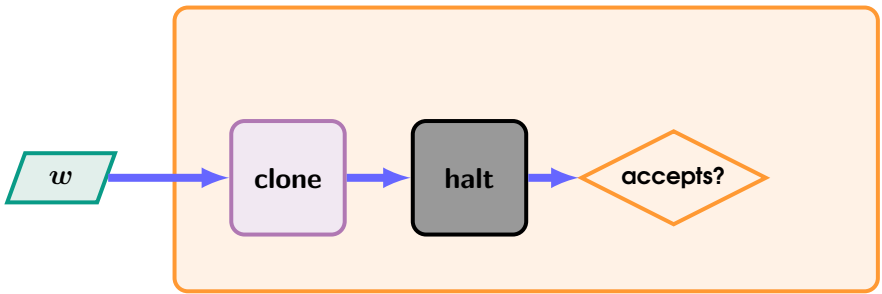
Other Undecidable Problems

Step 3) We construct a **new machine** $M^{\star}$ as follows:



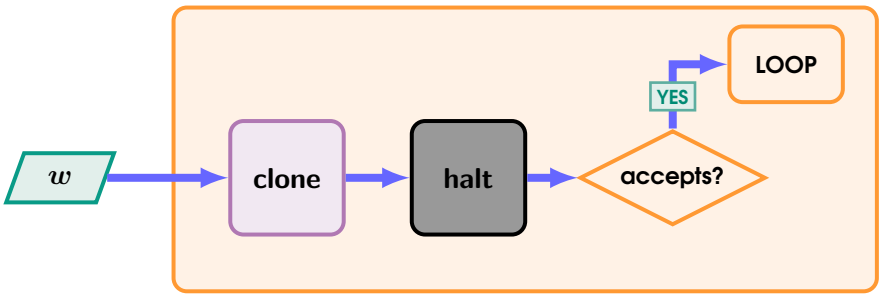
Other Undecidable Problems

Step 3) We construct a **new machine** $M^{\star}$ as follows:



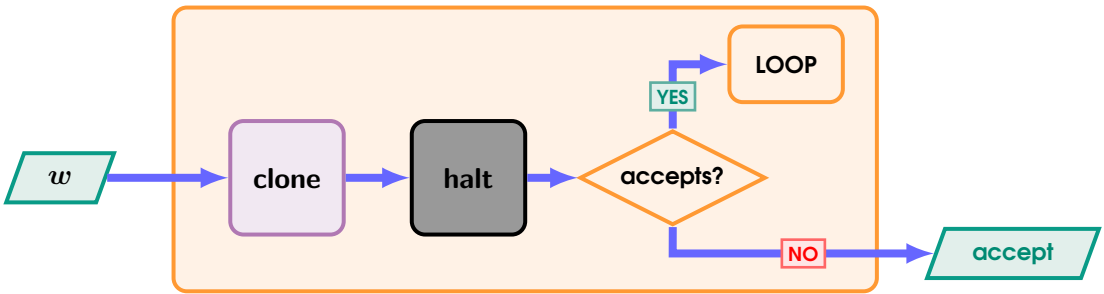
Other Undecidable Problems

Step 3) We construct a **new machine** $M^{\star}$ as follows:



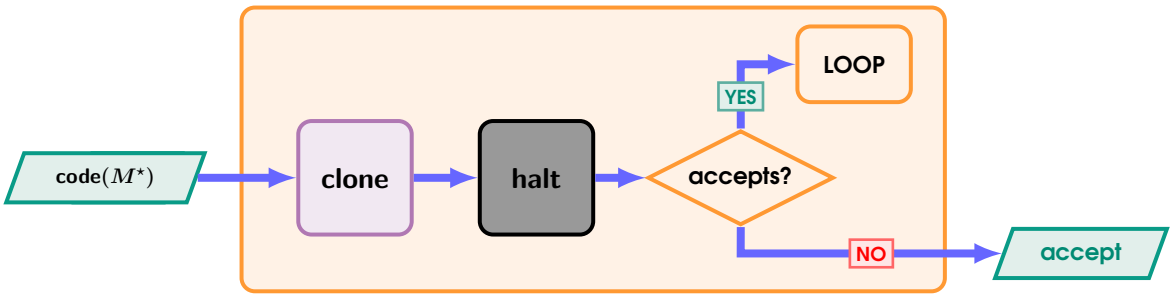
Other Undecidable Problems

Step 3) We construct a **new machine** $M^{\star}$ as follows:



Other Undecidable Problems

Step 3) We construct a **new machine** $M^{\star}$ as follows:



End of Slides!