CSC 339 – Theory of Computation
Fall 2023

13. Decidable and Undecidable Languages

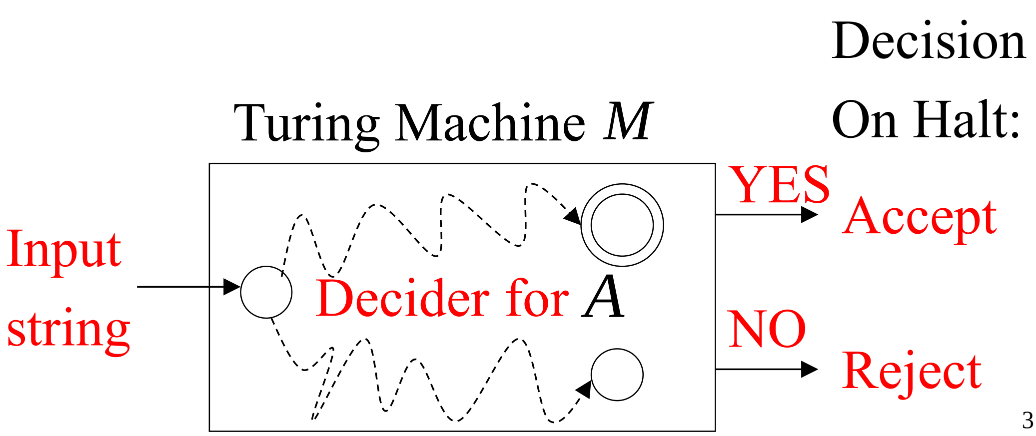
Outline

- Decidable languages
- Undecidable languages
- Halting problem
- Chomsky Hierarchy

2

Decidable Languages

A language A is decidable, if there is a Turing machine M (decider) that accepts the language A and halts on every input string.



A computational problem is decidable if the corresponding language is decidable.

We also say that the problem is solvable.

Problem 1: Does DFA M accept the empty language $L(M) = \emptyset$?

Corresponding Language: (Decidable)

$EMPTY_{DFA} =$

$\{\langle M \rangle : M$ is a DFA that accepts empty language $\emptyset\}$

Description of DFA M as a string
(For example, we can represent M as a binary string)

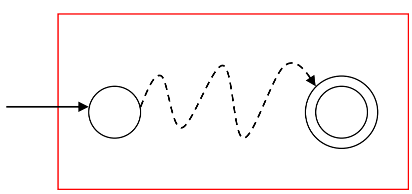
5

Decider for $EMPTY_{DFA}$:

On input $\langle M \rangle$:

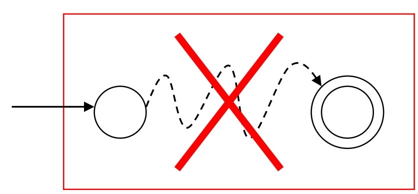
Determine whether there is a path from the initial state to any accepting state.

DFA M



$L(M) \neq \varnothing$

DFA M



$L(M) = \varnothing$

Decision:

Reject $\langle M \rangle$ (NO)

Accept $\langle M \rangle$ (YES)

Problem 2: Does DFA M accept a finite language?

Corresponding Language: (Decidable)

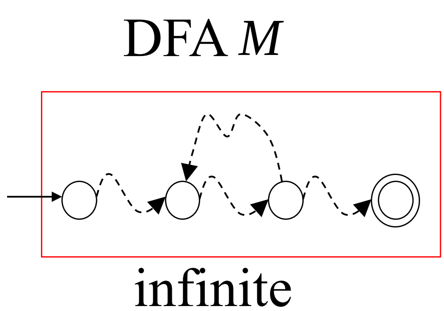
$FINITE_{DFA} =$

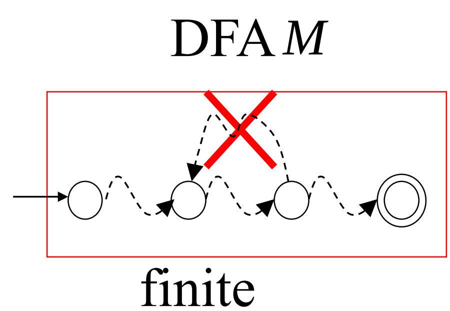
$\{\langle M \rangle : M \text{ is a DFA that accepts a finite language}\}$

Decider for $FINITE_{DFA}$:

On input $\langle M \rangle$:

Check if there is a walk with a cycle from the initial state to an accepting state.





Decision: Reject $\langle M \rangle$ (NO) Accept $\langle M \rangle$ (YES)

8

Problem 3: Does DFA M accept string w?

Corresponding Language: (Decidable)

$$A_{DFA} = \{\langle M, w \rangle : M \text{ is a DFA that accepts string } w\}$$

9

Decider for A_{DFA} :

On input string $\langle M, w \rangle$:

Run DFA M on input string w

If M accepts w
Then accept $\langle M, w \rangle$ (and halt)
Else reject $\langle M, w \rangle$ (and halt)

10

Problem 4: Do DFAs M_1 and M_2 accept the same language?

Corresponding Language: (Decidable)

$$EQUAL_{DFA} = \{\langle M_1, M_2 \rangle : M_1 \text{ and } M_2 \text{ are DFAs that accept the same language}\}$$

Decider for $EQUAL_{DFA}$:

On input $\langle M_1, M_2 \rangle$:

Let L_1 be the language of DFA M_1
Let L_2 be the language of DFA M_2

Construct DFA M such that:

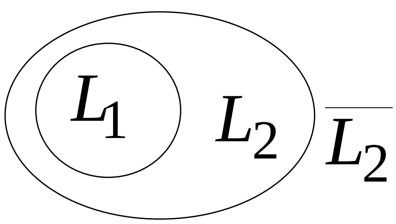
$$L(M) = (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2)$$

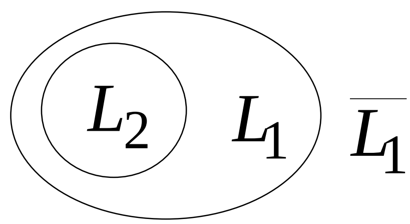
(combination of DFAs)

12

$$(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) = \varnothing$$

$$L_1 \cap \overline{L_2} = \varnothing \quad \text{and} \quad \overline{L_1} \cap L_2 = \varnothing$$





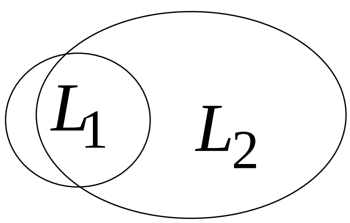
$$L_1 \subseteq L_2 \qquad\qquad L_2 \subseteq L_1$$

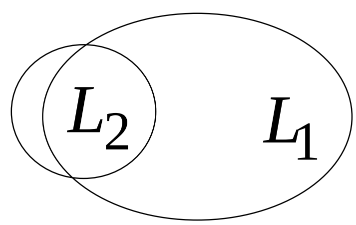
$$L_1 = L_2$$

13

$$(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) \neq \varnothing$$

$L_1 \cap \overline{L_2} \neq \varnothing$ or $\overline{L_1} \cap L_2 \neq \varnothing$





$L_1 \not\subset L_2$ $\qquad$ $L_2 \not\subset L_1$

$$L_1 \neq L_2$$

Therefore, we only need to determine whether

$$L(M) = (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) = \varnothing$$

which is a solvable problem for DFAs:
Problem 1: $EMPTY_{DFA}$

Undecidable Languages

undecidable language = not decidable language

There is no decider:

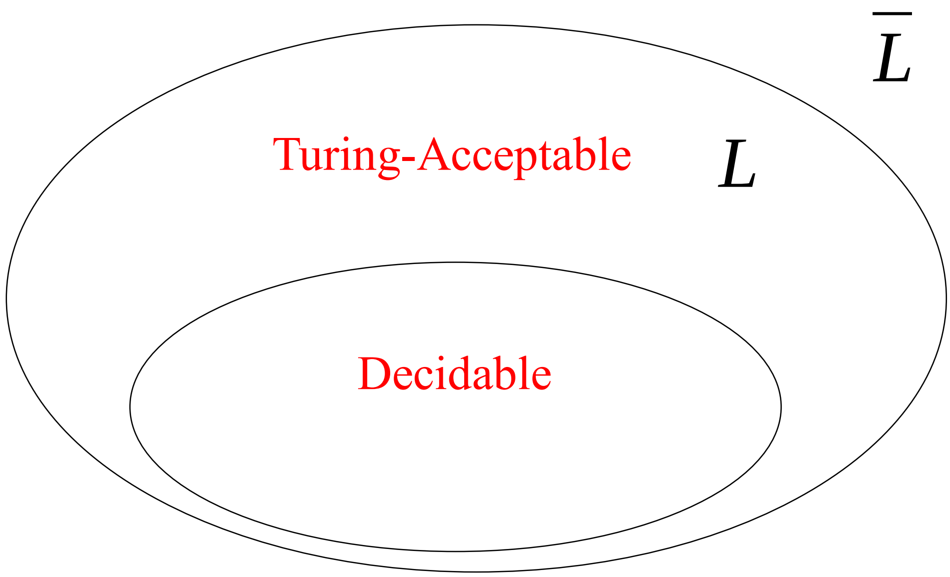
There is no Turing Machine which accepts the language and makes a decision (halts) for every input string (a machine may make decision for some input strings).

16

For an undecidable language, the corresponding problem is undecidable (unsolvable):

There is no Turing Machine (Algorithm) that gives an answer (yes or no) for every input instance (answer may be given for some input instances).

We have shown before that there are undecidable languages:



L is Turing-Acceptable and undecidable.

18

We will prove that the halting problem is unsolvable.

Halting Problem

Input: • Turing Machine M
• String w

Question: Does M halt while processing input string w ?

Corresponding language:

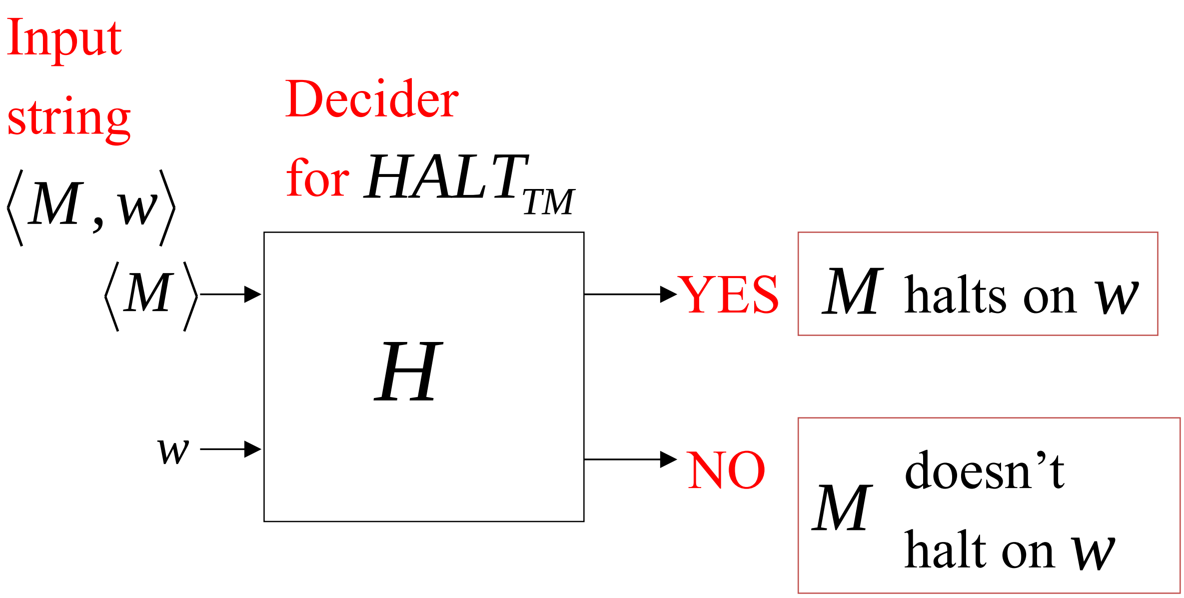
$$HALT_{TM} = \{\langle M, w \rangle : M \text{ is a Turing machine that halts on input string } w\}$$

Theorem:

$HALT_{TM}$ is undecidable.

(This means that the halting problem is unsolvable).

Suppose that $HALT_{TM}$ is decidable



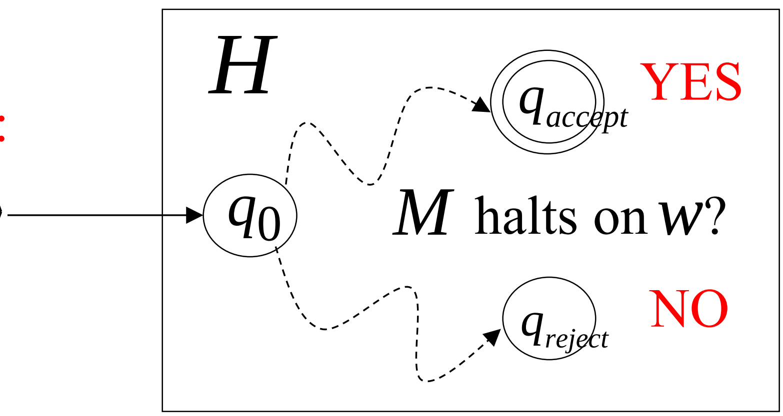
22

Looking inside H

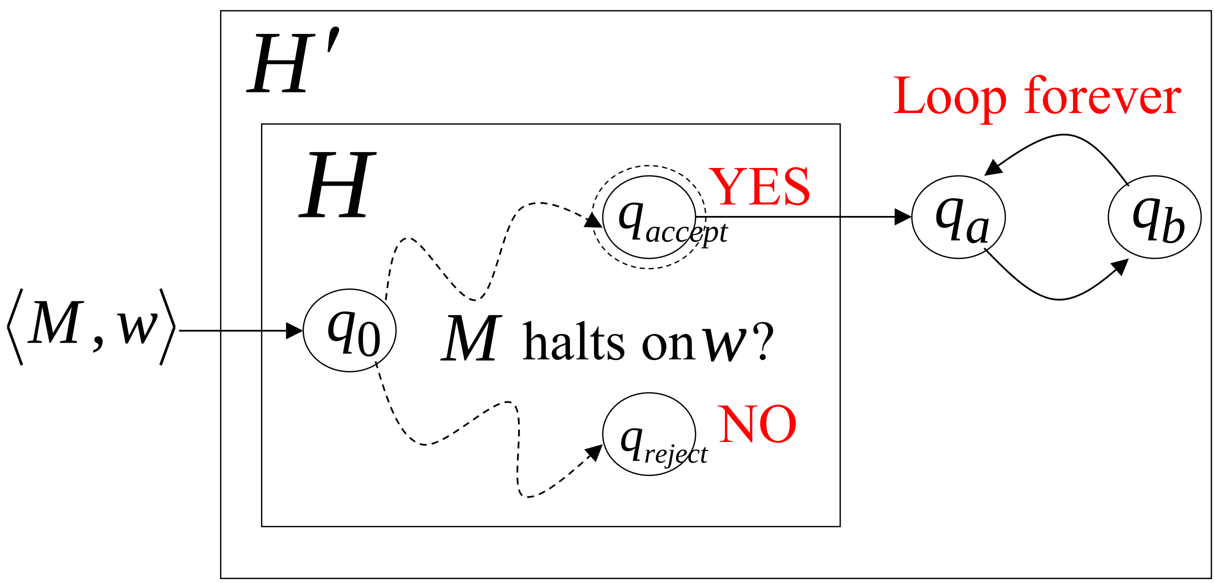
Decider for $HALT_{TM}$

Input string:

$\langle M, w \rangle$



Construct machine H':

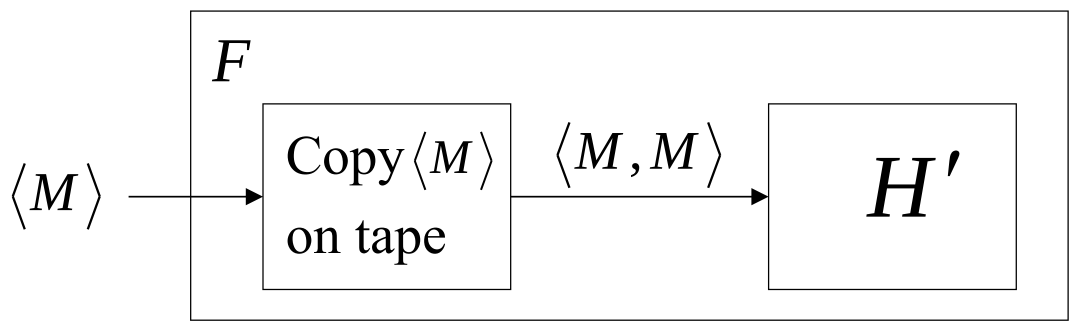


If M halts on input w Then Loop Forever

Else Halt

24

Construct machine F:

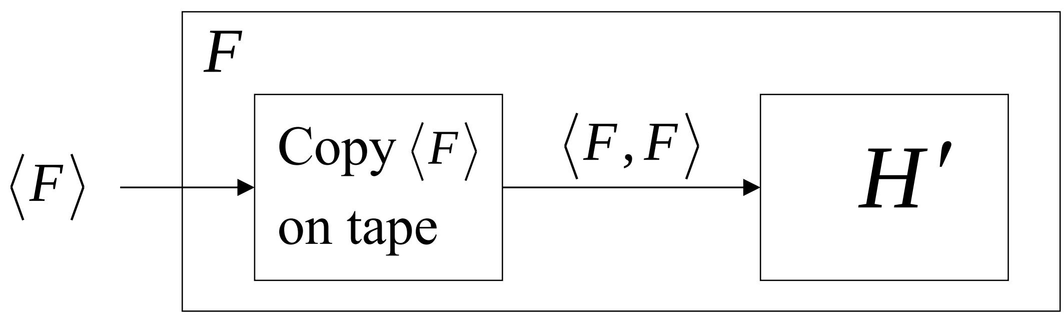


If M halts on input $\langle M \rangle$

Then Loop forever

Else Halt

Run F with input itself



If F halts on input $\langle F \rangle$

<u>Then</u> F loops forever on input $\langle F \rangle$

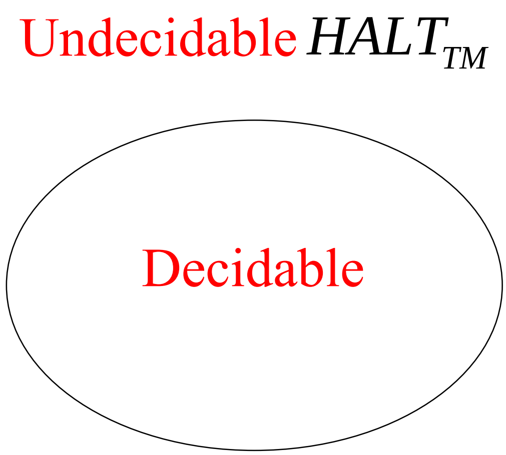
<u>Else</u> F halts on input $\langle F \rangle$

Contradiction!

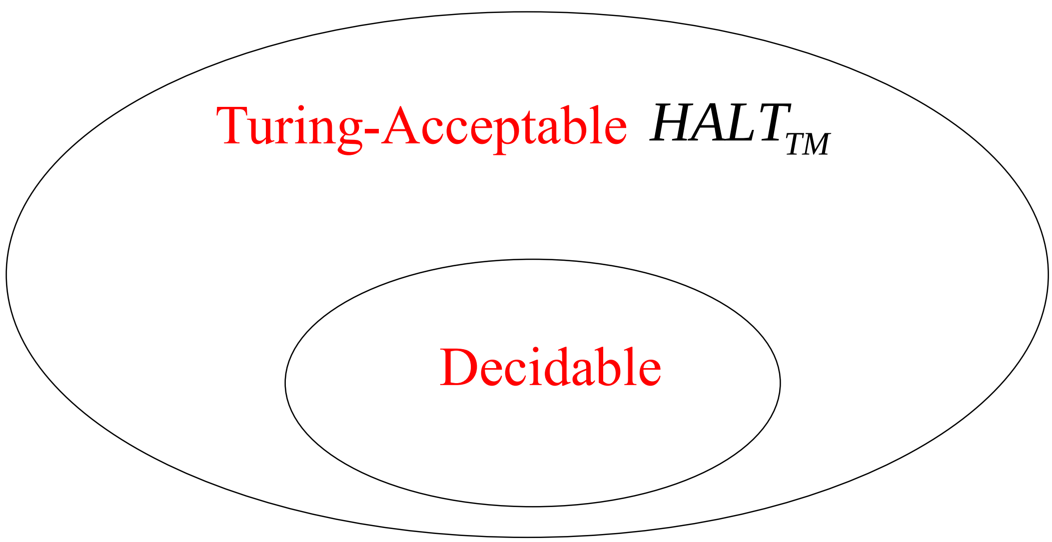
$HALT_{TM}$ is undecidable.

26

We then have:



Let's show:



28

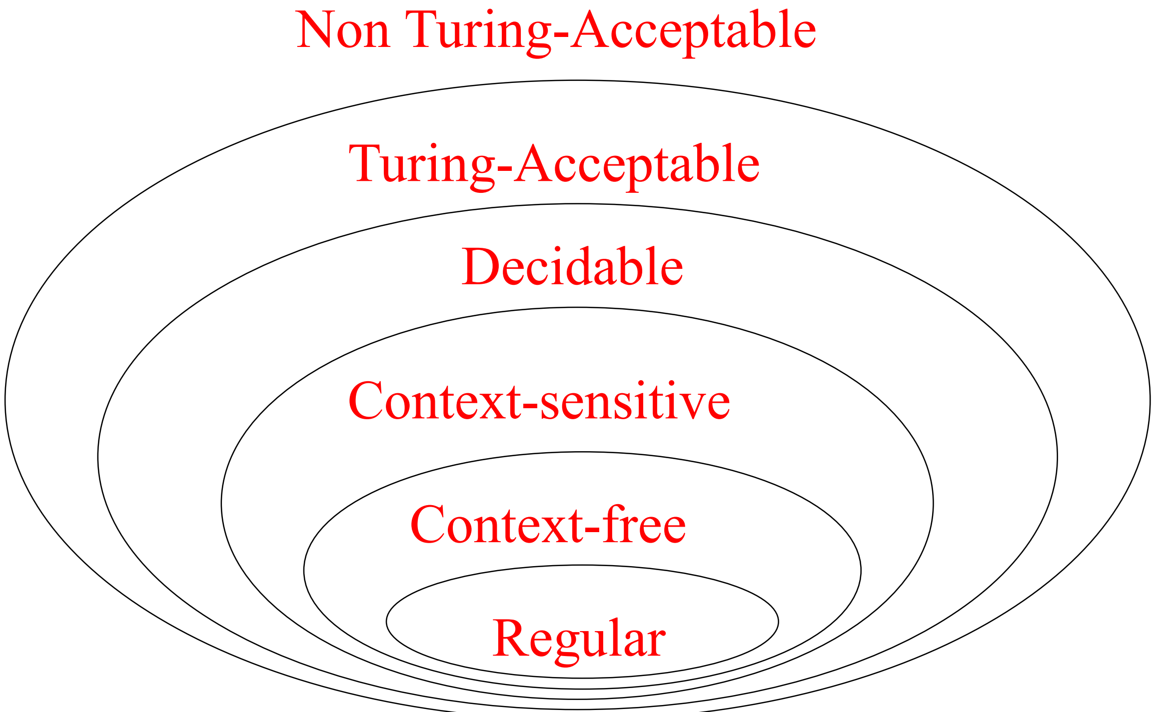
$HALT_{TM}$ is Turing-Acceptable

Turing machine that accepts $HALT_{TM}$:

$\langle M, w \rangle \longrightarrow$

1. Run M on input w
2. If M halts on w then accept $\langle M, w \rangle$

Chomsky Hierarchy



30