

# Theory of Computer Science

## B10. Turing Machines I

Gabriele Röger

University of Basel

March 29, 2023

# Turing Machines

# Automata for Type-1 and Type-0 Languages?



Finite automata recognize exactly the regular languages, push-down automata exactly the context-free languages. Are there automata models for context-sensitive and type-0 languages?

# Automata for Type-1 and Type-0 Languages?



Finite automata recognize exactly the regular languages, push-down automata exactly the context-free languages. Are there automata models for context-sensitive and type-0 languages?

Yes!  $\rightsquigarrow$  Turing machines

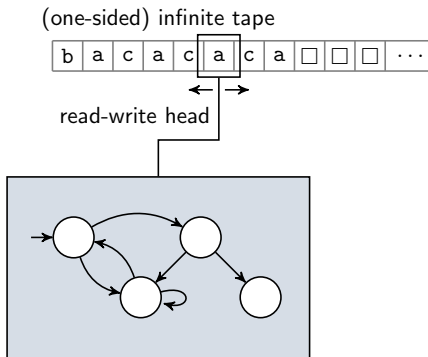
# Alan Turing (1912–1954)



Picture courtesy of Jon Callas /  
wikimedia commons

- British logician, mathematician, cryptanalyst and computer scientist
- most important work (for us):  
On Computable Numbers,  
with an Application to the  
Entscheidungsproblem  
↪ Turing machines
- collaboration on Enigma decryption
- conviction due to homosexuality;  
pardoned by Elizabeth II in Dec. 2013
- Turing award most important  
science award in computer science

# Turing Machines: Conceptually



# Turing Machine: Definition

## Definition (Deterministic Turing Machine)

A (deterministic) **Turing machine (DTM)** is given by a 7-tuple  $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \rangle$ , where  $Q, \Sigma, \Gamma$  are all finite sets and

- $Q$  is the set of **states**,
- $\Sigma$  is the **input alphabet**, not containing the **blank symbol**  $\square$ ,
- $\Gamma$  is the **tape alphabet**, where  $\square \in \Gamma$  and  $\Sigma \subseteq \Gamma$ ,
- $\delta : (Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$  is the **transition function**,
- $q_0 \in Q$  is the **start state**,
- $q_{\text{accept}} \in Q$  is the **accept state**,
- $q_{\text{reject}} \in Q$  is the **reject state**, where  $q_{\text{accept}} \neq q_{\text{reject}}$ .

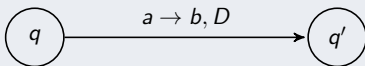
# Turing Machine: Transition Function

Let  $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \rangle$  be a DTM.

What is the Intuitive Meaning of the Transition Function  $\delta$ ?

$\delta(q, a) = \langle q', b, D \rangle$ :

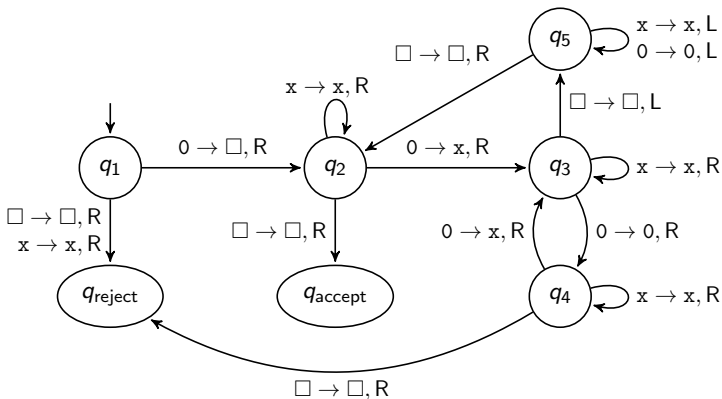
- If  $M$  is in state  $q$  and reads  $a$ , then
- $M$  transitions to state  $q'$  in the next step,
- replacing  $a$  with  $b$ ,
- and moving the head in direction  $D \in \{L, R\}$ , where:
  - **R**: one step to the **right**,
  - **L**: one step to the **left**, except if the head is on the left-most cell of the tape in which case there is no movement





# Deterministic Turing Machine: Example

$\langle \{q_1, \dots, q_5, q_{\text{accept}}, q_{\text{reject}}\}, \{0\}, \{0, x, \square\}, \delta, q_1, q_{\text{accept}}, q_{\text{reject}} \rangle$



# Turing Machine: Configuration

## Definition (Configuration of a Turing Machine)

A **configuration** of a Turing machine

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \rangle$$

is given by a triple  $c \in \Gamma^* \times Q \times \Gamma^+$ .

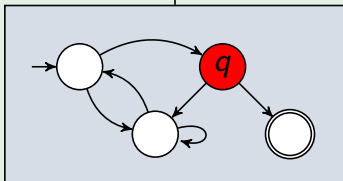
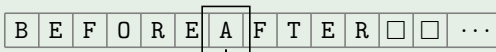
Configuration  $\langle w_1, q, w_2 \rangle$  intuitively means that

- the non-empty or already visited part of the tape contains the word  $w_1 w_2$ ,
- the read-write head is on the first symbol of  $w_2$ , and
- the TM is in state  $q$ .

# Turing Machine Configurations: Example

## Example

configuration  $\langle \text{BEFORE}, q, \text{AFTER} \square\square \rangle$ .



# Turing Machine Configurations: Start Configuration

## Initially

- the TM is in start state  $q_0$ ,
- the head is on the first tape cell, and
- the tape contains the input word  $w$  followed by an infinite number of  $\square$  entries.

The corresponding start configuration is  $\langle \varepsilon, q_0, w \rangle$  if  $w \neq \varepsilon$  and  $\langle \varepsilon, q_0, \square \rangle$  if  $w = \varepsilon$ .

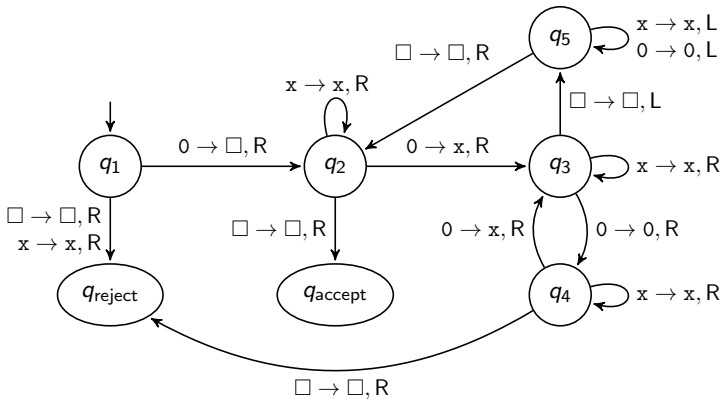
# Turing Machine: Step

## Definition (Transition/Step of a Turing Machine)

A DTM  $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \rangle$  transitions from configuration  $c$  to configuration  $c'$  in one step ( $c \vdash_M c'$ ) according to the following rules:

- $\langle a_1 \dots a_m, q, b_1 \dots b_n \rangle \vdash_M \langle a_1 \dots a_m c, q', b_2 \dots b_n \rangle$   
if  $\delta(q, b_1) = \langle q', c, R \rangle$ ,  $m \geq 0$ ,  $n \geq 2$
- $\langle a_1 \dots a_m, q, b_1 \rangle \vdash_M \langle a_1 \dots a_m c, q', \square \rangle$   
if  $\delta(q, b_1) = \langle q', c, R \rangle$ ,  $m \geq 0$
- $\langle a_1 \dots a_m, q, b_1 \dots b_n \rangle \vdash_M \langle a_1 \dots a_{m-1}, q', a_m c b_2 \dots b_n \rangle$   
if  $\delta(q, b_1) = \langle q', c, L \rangle$ ,  $m \geq 1$ ,  $n \geq 1$
- $\langle \varepsilon, q, b_1 \dots b_n \rangle \vdash_M \langle \varepsilon, q', c b_2 \dots b_n \rangle$   
if  $\delta(q, b_1) = \langle q', c, L \rangle$ ,  $n \geq 1$

# Step: Exercise (Slido)



$\langle \square x, q_3, 00 \rangle \vdash ?$



## DTM: Accepted Words

Intuitively, a DTM **accepts a word** if its computation terminates in the **accept state**.

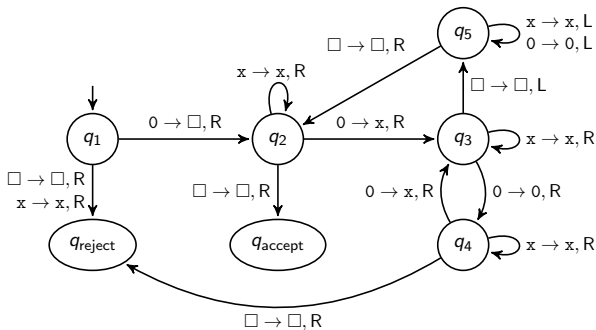
### Definition (Words Accepted by a DTM)

DTM  $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \rangle$  **accepts the word**  $w = a_1 \dots a_n$  if there is a sequence of configurations  $c_0, \dots, c_k$  with

- 1  $c_0$  is the start configuration of  $M$  on input  $w$ ,
- 2  $c_i \vdash_M c_{i+1}$  for all  $i \in \{0, \dots, k-1\}$ , and
- 3  $c_k$  is an accepting configuration, i. e., a configuration with state  $q_{\text{accept}}$ .

# Accepted Word: Example (Slido)

Does this Turing machine accept input 0000?





## DTM: Recognized Language

### Definition (Language Recognized by a DTM)

Let  $M$  be a deterministic Turing Machine

The **language recognized by  $M$**  (or **the language of  $M$** ) is defined as  $\mathcal{L}(M) = \{w \in \Sigma^* \mid w \text{ is accepted by } M\}$ .

## DTM: Recognized Language

### Definition (Language Recognized by a DTM)

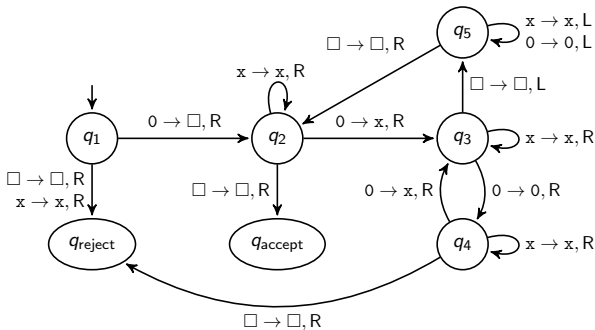
Let  $M$  be a deterministic Turing Machine

The **language recognized by  $M$**  (or **the language of  $M$** ) is defined as  $\mathcal{L}(M) = \{w \in \Sigma^* \mid w \text{ is accepted by } M\}$ .

### Definition (Turing-recognizable Language)

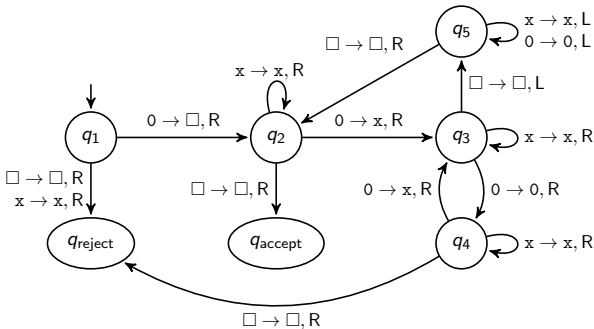
We call a language **Turing-recognizable** if some deterministic Turing machine recognizes it.

# Turing Machine: Example



- 1 Sweep left to right across the tape, crossing off every other 0.
- 2 If in stage 1 the tape contained a single 0, accept.
- 3 If in stage 1 the tape contained more than one 0 and the number of 0s was odd, reject.
- 4 Return the head to the left end of the tape and go to stage 1.

# Recognized Language: Example



What language does the Turing machine recognize?



# Deciders

- A Turing machine either fails to accept an input
  - because it **rejects** it (entering  $q_{\text{reject}}$ ) or
  - because it **loops** (= does not halt).
- A Turing machine that halts on all inputs (entering  $q_{\text{reject}}$  or  $q_{\text{accept}}$ ) is called a **decider**.
- A decider that recognizes some language also is said to **decide** the language.

## Definition (Turing-decidable Language)

We call a language **Turing-decidable** (or **decidable**) if some deterministic Turing machine decides it.

## Exercise (if time)

Specify the state diagram of a DTM that decides language

$$L = \{w\#w \mid w \in \{0, 1\}^*\}.$$



Feel free to solve this together with your neighbour.

# Questions



Questions?

# Summary



# Summary

- Turing machines only have finitely many states but an **unbounded tape** as “memory”.
- Alan Turing proposed them as a mathematical model for arbitrary algorithmic computations.
- In this role, we will revisit them in the parts on computability and complexity theory.