Computación y Estructuras Discretas III

Andrés A. Aristizábal P.
aaaristizabal@icesi.edu.co
Ángela Villota
apvillota@icesi.edu.co

Departamento de Computación y Sistemas Inteligentes

2024-2

Agenda del día

1. **Turing machines**
   - Topic presentation
   - Exercises

Agenda del día

1. **Turing machines**
   - Topic presentation
   - Exercises

Turing machines

What is a Turing machine?

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, b\!\!/, \delta)$, consists of seven components:

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, b\!\!/, \delta)$, consists of seven components:

1. Q is the (finite) set of internal states of the control unit.

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, \not{b}, \delta)$, consists of seven components:

1. Q is the (finite) set of internal states of the control unit.
2. $q_0 \in Q$ is the initial state.

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, b\!\!\!/, \delta)$, consists of seven components:

1. Q is the (finite) set of internal states of the control unit.
2. $q_0 \in Q$ is the initial state.
3. F is the set of final or accepting states, $\emptyset \neq F \subseteq Q$.

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, b\!\!\!/, \delta)$, consists of seven components:

1. Q is the (finite) set of internal states of the control unit.
2. $q_0 \in Q$ is the initial state.
3. F is the set of final or accepting states, $\emptyset \neq F \subseteq Q$.
4. Σ is the input alphabet.

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, \not b, \delta)$, consists of seven components:

1. Q is the (finite) set of internal states of the control unit.
2. $q_0 \in Q$ is the initial state.
3. F is the set of final or accepting states, $\emptyset \neq F \subseteq Q$.
4. Σ is the input alphabet.
5. Γ is the tape alphabet, which includes Σ, i.e., $\Sigma \subseteq \Gamma$.

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, b\!\!\!/, \delta)$, consists of seven components:

1. Q is the (finite) set of internal states of the control unit.
2. $q_0 \in Q$ is the initial state.
3. F is the set of final or accepting states, $\emptyset \neq F \subseteq Q$.
4. Σ is the input alphabet.
5. Γ is the tape alphabet, which includes Σ, i.e., $\Sigma \subseteq \Gamma$.
6. $b\!\!\!/ \in \Gamma$ is the “blank” symbol (the symbol $b\!\!\!/$ cannot be part of the input alphabet Σ).

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, \not b, \delta)$, consists of seven components:

1. Q is the (finite) set of internal states of the control unit.
2. $q_0 \in Q$ is the initial state.
3. F is the set of final or accepting states, $\emptyset \neq F \subseteq Q$.
4. Σ is the input alphabet.
5. Γ is the tape alphabet, which includes Σ, i.e., $\Sigma \subseteq \Gamma$.
6. $\not b \in \Gamma$ is the "blank" symbol (the symbol $\not b$ cannot be part of the input alphabet Σ).
7. δ is the transition function of the machine:

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, \not{b}, \delta)$, consists of seven components:

1. Q is the (finite) set of internal states of the control unit.
2. $q_0 \in Q$ is the initial state.
3. F is the set of final or accepting states, $\emptyset \neq F \subseteq Q$.
4. Σ is the input alphabet.
5. Γ is the tape alphabet, which includes Σ, i.e., $\Sigma \subseteq \Gamma$.
6. $\not{b} \in \Gamma$ is the "blank" symbol (the symbol $\not{b}$ cannot be part of the input alphabet Σ).
7. δ is the transition function of the machine:

$$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow\}$$

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, b\!\!\!/, \delta)$, consists of seven components:

1. Q is the (finite) set of internal states of the control unit.
2. $q_0 \in Q$ is the initial state.
3. F is the set of final or accepting states, $\emptyset \neq F \subseteq Q$.
4. Σ is the input alphabet.
5. Γ is the tape alphabet, which includes Σ, i.e., $\Sigma \subseteq \Gamma$.
6. $b\!\!\!/ \in \Gamma$ is the “blank” symbol (the symbol $b\!\!\!/$ cannot be part of the input alphabet Σ).
7. δ is the transition function of the machine:

$$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow\}$$

   - δ is a partial function, meaning it may not be defined for some elements of the domain.

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, b\!\!\!/, \delta)$, consists of seven components:

1. Q is the (finite) set of internal states of the control unit.
2. $q_0 \in Q$ is the initial state.
3. F is the set of final or accepting states, $\emptyset \neq F \subseteq Q$.
4. Σ is the input alphabet.
5. Γ is the tape alphabet, which includes Σ, i.e., $\Sigma \subseteq \Gamma$.
6. $b\!\!\!/ \in \Gamma$ is the “blank” symbol (the symbol $b\!\!\!/$ cannot be part of the input alphabet Σ).
7. δ is the transition function of the machine:

$$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow\}$$

- δ is a partial function, meaning it may not be defined for some elements of the domain.
- The arrow $\leftarrow$ denotes a move to the left, while $\rightarrow$ denotes a move to the right.

Turing machines

What is a Turing machine?

A Turing Machine (TM), $M = (Q, q_0, F, \Sigma, \Gamma, \not{b}, \delta)$, consists of seven components:

1. Q is the (finite) set of internal states of the control unit.
2. $q_0 \in Q$ is the initial state.
3. F is the set of final or accepting states, $\emptyset \neq F \subseteq Q$.
4. Σ is the input alphabet.
5. Γ is the tape alphabet, which includes Σ, i.e., $\Sigma \subseteq \Gamma$.
6. $\not{b} \in \Gamma$ is the "blank" symbol (the symbol $\not{b}$ cannot be part of the input alphabet Σ).
7. δ is the transition function of the machine:

$$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow\}$$

- δ is a partial function, meaning it may not be defined for some elements of the domain.
- The arrow $\leftarrow$ denotes a move to the left, while $\rightarrow$ denotes a move to the right.
- The transition $\delta(q, a) = (p, b, D)$ means: while in state q, scanning symbol a, the control unit erases a, writes b and moves to state p, either to the left (if the move D is $\longleftarrow$) or to the right (if D is $\longrightarrow$).

Turing machines

How does a Turing machine work?

Turing machines

How does a Turing machine work?

- A Turing Machine M processes input strings $w \in \Sigma^*$ placed on an infinite tape in both directions.
- To process an input string w, the control unit of M is in the initial state q_0 scanning the first symbol of w.
- The remaining cells or squares of the tape contain the blank symbol $\not{b}$.

Turing machines

How does a Turing machine work?

- A Turing Machine M processes input strings $w \in \Sigma^*$ placed on an infinite tape in both directions.
- To process an input string w, the control unit of M is in the initial state q_0 scanning the first symbol of w.
- The remaining cells or squares of the tape contain the blank symbol $\not{b}$.

What is an instantaneous configuration?

Turing machines

How does a Turing machine work?

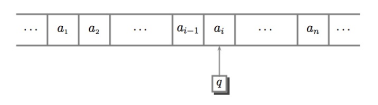
- A Turing Machine M processes input strings $w \in \Sigma^*$ placed on an infinite tape in both directions.
- To process an input string w, the control unit of M is in the initial state q_0 scanning the first symbol of w.
- The remaining cells or squares of the tape contain the blank symbol b.

What is an instantaneous configuration?

A description or instantaneous configuration is an expression of the form:

$$a_1 a_2 \cdots a_{i-1} q a_i \cdots a_n$$

where the symbols $a_1, \ldots, a_n$ belong to the tape alphabet Γ y $q \in Q$. This expression represents the current state of the computation:



Turing machines

What does the instantaneous configuration indicate?

Turing machines

What does the instantaneous configuration indicate?

The instantaneous description $a_1 a_2 \cdots a_{i-1} q a_i \cdots a_n$ indicates that the control unit of M is in state q scanning symbol a_i. It is assumed that all cells to the left of a_1 and to the right of a_n, contain the blank symbol $\not b$.

Example

Examples of instantaneous descriptions are:

$$aabq_2baaa$$
$$q_5ababba$$
$$ab\not b\not bbaabq_0bba$$

Turing machines

What does the instantaneous configuration indicate?

The instantaneous description $a_1 a_2 \cdots a_{i-1} q a_i \cdots a_n$ indicates that the control unit of M is in state q scanning symbol a_i. It is assumed that all cells to the left of a_1 and to the right of a_n, contain the blank symbol $\not{b}$.

Example

Examples of instantaneous descriptions are:

$$aabq_2 baaa$$
$$q_5 ababba$$
$$ab\not{b}\not{b}baabq_0 bba$$

What is the initial instantaneous configuration?

Turing machines

What does the instantaneous configuration indicate?

The instantaneous description $a_1 a_2 \cdots a_{i-1} q a_i \cdots a_n$ indicates that the control unit of M is in state q scanning symbol a_i. It is assumed that all cells to the left of a_1 and to the right of a_n, contain the blank symbol $\not b$.

Example

Examples of instantaneous descriptions are:

$$aabq_2baaa$$
$$q_5ababba$$
$$ab\not b\not bbaabq_0bba$$

What is the initial instantaneous configuration?

The initial instantaneous configuration, or simply the initial configuration, is $q_0 w$, where w is the input string. This is placed anywhere on the input tape.

What is a computational step?

Turing machines

What is a computational step?

The transition from one instantaneous description to another, through a transition defined by δ, is called a computational step and is denoted by:

$$u_1 q u_2 \vdash v_1 p v_2.$$

Here $u_1, u_2, v_1, v_2 \in \Gamma^*$ and $p, q \in Q$.

Turing machines

What is a computational step?

The transition from one instantaneous description to another, through a transition defined by δ, is called a computational step and is denoted by:

$$u_1 q u_2 \vdash v_1 p v_2 .$$

Here $u_1, u_2, v_1, v_2 \in \Gamma^*$ and $p, q \in Q$.

Example

A concrete example is:

$$abbaq_2ba \vdash abbq_1aca$$

in which the machine used the transition $\delta(q_2, b) = (q_1, c, \leftarrow)$.

Turing machines

What is a computational step?

The transition from one instantaneous description to another, through a transition defined by δ, is called a computational step and is denoted by:

$$u_1 q u_2 \vdash v_1 p v_2.$$

Here $u_1, u_2, v_1, v_2 \in \Gamma^*$ and $p, q \in Q$.

Example

A concrete example is:

$$abbaq_2ba \vdash abbq_1aca$$

in which the machine used the transition $\delta(q_2, b) = (q_1, c, \leftarrow)$.

What does the notation $u_1 q u_2 \overset{*}{\vdash} v_1 p v_2$ indicate?

Turing machines

What is a computational step?

The transition from one instantaneous description to another, through a transition defined by δ, is called a computational step and is denoted by:

$$u_1 q u_2 \vdash v_1 p v_2 .$$

Here $u_1, u_2, v_1, v_2 \in \Gamma^*$ and $p, q \in Q$.

Example

A concrete example is:

$$abbaq_2 ba \vdash abbq_1 aca$$

in which the machine used the transition $\delta(q_2, b) = (q_1, c, \leftarrow)$.

What does the notation $u_1 q u_2 \overset{*}{\vdash} v_1 p v_2$ indicate?

It means that M can transition from the instantaneous description $u_1 q u_2$ to the instantaneous description $v_1 p v_2$ in zero or more computational steps.

Turing machines

What are special computations?

What are special computations?

During the computation or processing of an input string, there are two special situations that can occur:

1. The computation terminates because at some point there is no defined transition.
2. The computation does not terminate; this is what is called an "infinite loop" or an "infinite cycle". This situation is represented by the notation

$$u_1 q u_2 \overset{*}{\vdash} \infty$$

   which indicates that the computation starting from the instantaneous description $u_1 q u_2$ never stops.

Turing machines

Are transitions with the symbol $\not b$, $\delta(q, \not b) = (p, s, D)$, the same as the λ transitions used in automata?

Are transitions with the symbol $\not{b}$, $\delta(q, \not{b}) = (p, s, D)$, the same as the λ transitions used in automata?

- They are not the same.
- A λ transition en un autómata transition in an automaton occurs regardless of the symbol read, and the control unit remains stationary, whereas a $\delta(q, \not{b}) = (p, s, D)$ transition in a TM requires the blank symbol $\not{b}$ to be written on the scanned cell.
- The control unit overwrites the blank and performs a shift.

Turing machines

What is the language accepted by a Turing machine?

Turing machines

What is the language accepted by a Turing machine?

- An input string w is accepted by a Turing machine M if the computation that starts in the initial configuration $q_0 w$ ends in an instantaneous configuration $w_1 p w_2$, where p is an accepting state, and in which M halts completely.
- The language $L(M)$ accepted by a Turing machine M is then defined as

$$L(M) := \{w \in \Sigma^* \mid q_0 w \overset{*}{\vdash} w_1 p w_2, p \in F, w_1, w_2 \in \Gamma^*, M \text{ halts in the configuration } w_1 p w_2\}.$$

Turing machines

What is the language accepted by a Turing machine?

- An input string w is accepted by a Turing machine M if the computation that starts in the initial configuration $q_0 w$ ends in an instantaneous configuration $w_1 p w_2$, where p is an accepting state, and in which M halts completely.
- The language $L(M)$ accepted by a Turing machine M is then defined as

$$L(M) := \{w \in \Sigma^* \mid q_0 w \overset{*}{\vdash} w_1 p w_2, p \in F, w_1, w_2 \in \Gamma^*, M \text{ halts in the configuration } w_1 p w_2\}.$$

What difference exists between the notion of acceptance for Turing machines and for automata?

Turing machines

What is the language accepted by a Turing machine?

- An input string w is accepted by a Turing machine M if the computation that starts in the initial configuration $q_0 w$ ends in an instantaneous configuration $w_1 p w_2$, where p is an accepting state, and in which M halts completely.
- The language $L(M)$ accepted by a Turing machine M is then defined as

$$L(M) := \{w \in \Sigma^* \mid q_0 w \overset{*}{\vdash} w_1 p w_2, p \in F, w_1, w_2 \in \Gamma^*, M \text{ halts in the configuration } w_1 p w_2\}.$$

What difference exists between the notion of acceptance for Turing machines and for automata?

- The notion of acceptance for Turing machines is more flexible than for automata.
- An input string does not have to be read entirely to be accepted.
- It is only required that the machine halts completely, at a certain point, in an accepting state.

When does the control unit stop in a Turing machine?

Turing machines

When does the control unit stop in a Turing machine?

- In the standard model, the control unit always stops upon entering *a* in an accepting state.
- That is, transitions $\delta(q, a)$ are not allowed when $q \in F$.

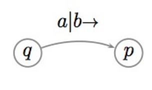
Turing machines

How are transition diagrams for Turing machines?

Turing machines

How are transition diagrams for Turing machines?

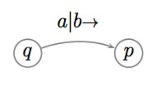
The transition function δ of a Turing machine can be presented as a labeled digraph. Thus, the transition $\delta(q, a) = (p, b, \rightarrow)$ s represented by



Turing machines

How are transition diagrams for Turing machines?

The transition function δ of a Turing machine can be presented as a labeled digraph. Thus, the transition $\delta(q, a) = (p, b, \rightarrow)$ s represented by

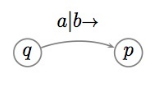


What is the stationary control unit?

Turing machines

How are transition diagrams for Turing machines?

The transition function δ of a Turing machine can be presented as a labeled digraph. Thus, the transition $\delta(q, a) = (p, b, \rightarrow)$ s represented by



What is the stationary control unit?

- To simplify the description of the standard model, it has been required that the control unit moves left or right on each transition.
- However, it is possible to allow the control unit to stay stationary during a certain computational step, meaning it does not move at all.

Turing machines

- This type of transition is written in the form:

$$\delta(q, a) = (p, b, -)$$

where $a, b \in \Gamma$; $p, q \in Q$ and $-$ means that the control unit does not move.
  - Such a transition can be simulated by moving right followed by a return to the left.
  - Thus, to simulate the transition $\delta(q, a) = (p, b, -)$ we use a new auxiliary state q' and the transitions:

$$\begin{aligned}&\delta(q, a) = (q', b, \rightarrow),\\&\delta(q', s) = (p, s, \leftarrow), \text{ for every symbol } s \in \Gamma.\end{aligned}$$

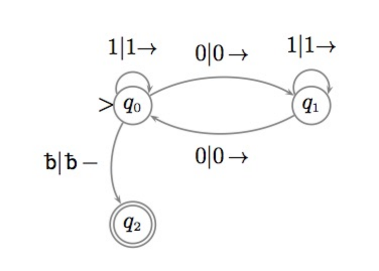
  - Stationary movement can also be simulated with a move to the left followed by a return to the right.
- It is concluded that Turing machines with stationary transitions, $\delta(q, a) = (p, b, -)$, in addition to normal transitions, accept the same languages as standard Turing machines.
- It is worth noting that stationary transitions resemble λ transitions in automata, except for the fact that Turing machines have the ability to overwrite scanned symbols.

Example

The following Turing machine accepts the language of strings with an even number of zeros, over the alphabet $0, 1$.

Example

The following Turing machine accepts the language of strings with an even number of zeros, over the alphabet $0, 1$.



Turing machines

Example

In this example, we built a Turing machine M that accepts the language $L = \{a^i b^i c^i \mid i \geq 0\}$ and halts when processing all inputs. Therefore, L eis a recursive language although it is not context-free, meaning, L cannot be accepted by any pushdown automaton.

Example

In this example, we built a Turing machine M that accepts the language $L = \{a^i b^i c^i \mid i \geq 0\}$ *and halts when processing all inputs. Therefore, L eis a recursive language although it is not context-free, meaning, L cannot be accepted by any pushdown automaton.*

Solution

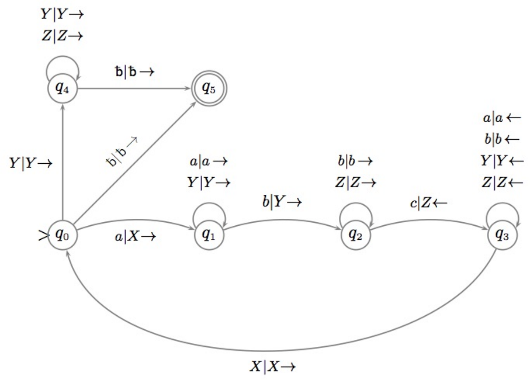
Let M be the Turing machine with parameters

$$\begin{aligned} \Sigma &= \{a, b, c\}, \\ \Gamma &= \{a, b, c, X, Y, Z, b\!\!\!/\}, \\ Q &= \{q_0, q_1, q_2, q_3, q_4, q_5\}, \\ F &= \{q_5\}, \end{aligned}$$

and whose transition function is represented by the following diagram:

Turing machines

Solution



Turing machines

Solution

The idea used for the design of this Turing machine can be described as follows:

Solution

The idea used for the design of this Turing machine can be described as follows:

- *The control unit changes the first a to X and moves to the right until it finds the first b, which is overwritten by a Y.*
- *Then it moves to the right until it finds the first c, which is changed to Z.*
- *The control then moves back to the left in search of the first X it encounters on its way; this return is done in state q_3.*
- *The machine then moves to the right, until the first a left on the tape, and the whole process is repeated.*
- *If the input string has the required form, all a's will be replaced by Xs, b's by Ys anc c's by Zs.*
- *Once the transformation is complete, the control moves to the right, in state q_4, until it finds the first cell marked with the blank symbol $\flat$.*
- *The Turing machine is designed in such a way that if the input string does not have the required form, the processing will end in a state other than the accepting state q_5.*

Solution

Next, we process the input string $w = aabbcc \in L$.

$$
\begin{array}{rcl}
q_0 aabbcc & \vdash & Xq_1 abbcc \vdash Xaq_1 bbcc \vdash XaYq_2 bcc \vdash XaYbq_2 cc \vdash XaYq_3 bZc \\
& \overset{*}{\vdash} & q_3 XaYbZc \vdash Xq_0 aYbZc \vdash XXq_1 YbZc \vdash XXYq_1 bZc \\
& \vdash & XXYYq_2 Zc \vdash XXYZq_2 c \vdash XXYq_3 ZZ \overset{*}{\vdash} Xq_3 XYZZ \\
& \vdash & XXq_0 YYZZ \vdash XXYq_4 YZZ \overset{*}{\vdash} XXYYZZq_4 \text{b} \\
& \vdash & XXYYZZ\text{b}q_5\text{b}.
\end{array}
$$

The input string $w = aaabbcc$*, which is not in* L*, is processed as follows:*

$$
\begin{array}{rcl}
q_0 aaabbcc & \vdash & Xq_1 aabbcc \vdash Xaaq_1 bbcc \vdash XaaYq_2 bcc \vdash XaaYbq_2 cc \\
& \vdash & XaaYbq_3 bZc \overset{*}{\vdash} q_3 XaaYbZc \vdash Xq_0 aaYbZc \vdash XXaq_1 YbZc \\
& \vdash & XXaYq_1 bZc \vdash XXaYYq_2 Zc \vdash XXaYZq_2 c \vdash XXaYq_3 ZZ \\
& \overset{*}{\vdash} & Xq_3 XaYZZ \vdash XXq_0 aYYZZ \vdash XXXq_1 YYZZ\text{b} \\
& \overset{*}{\vdash} & XXXYYq_1 ZZ \text{ (Cómputo abortado).}
\end{array}
$$

Agenda del día

1. **Turing machines**
   - Topic presentation
   - Exercises