Fundamental Computer Science

Giorgio Lucarelli (revisited by Denis Trystram)

2019

Organization

Classes

- 30 hours in total
- Theory: 50%
- Exercises: 50%

Evaluation

- Exams: 70%
- Tests during Exercises sessions: 30%

References

1. Harry R. Lewis and Christos H. Papadimitriou, *Elements of the Theory of Computation*, Prentice-Hall
2. Christos H. Papadimitriou, *Computational Complexity*, Pearson
3. S. Arora and B. Barak, *Computational complexity – a modern approach*, Cambridge
4. Vijay V. Vazirani, *Approximation Algorithms*, Springer

What is an *Algorithm* ?

Informally: a procedure composed by a set of steps that solves a problem

Desired properties

What is an *Algorithm* ?

Informally: a procedure composed by a set of steps that solves a problem

Desired properties

- clearly defined steps (formalization)
- efficiency (complexity - how many steps?)
- termination

History

- Etymology:
  - Al-Khwārizmī – a Persian mathematician of the 9th century
  - $\alpha\rho\iota\theta\mu$ός – the Greek word that means "number"

- Euclid's algorithm for computing the *greatest common divisor* (3rd century BC)
- End of 19th century - beginning of 20th century: mathematical formalizations (proof systems, axioms, etc). Is there an algorithm for any problem?
- Entscheidungsproblem (a challenge proposed by David Hilbert 1928): create an algorithm which is able to decide if a mathematical statement is true in a finite number of operations
- Church-Turing thesis (1930's): provides a formal definition of an algorithm (λ-calculus, Turing machine) and show that a solution to Entscheidungsproblem does not exist

Preliminaries

alphabet: a finite set of symbols

- examples: Roman alphabet $\{a, b, \ldots, z\}$, binary alphabet $\{0, 1\}$

Preliminaries

alphabet: a finite set of symbols

- examples: Roman alphabet $\{a, b, \ldots, z\}$, binary alphabet $\{0, 1\}$

string: a finite sequence of symbols over an alphabet

- examples: $science$, 0011101
- ϵ: the empty string
- Σ^*: the set of all strings over an alphabet Σ (including ϵ)

Preliminaries

alphabet: a finite set of symbols

- examples: Roman alphabet $\{a, b, \ldots, z\}$, binary alphabet $\{0, 1\}$

string: a finite sequence of symbols over an alphabet

- examples: $science$, 0011101
- ϵ: the empty string
- Σ^*: the set of all strings over an alphabet Σ (including ϵ)

language: a set strings over an alphabet Σ (i.e., a subset of Σ^*)

- examples: $\emptyset$, Σ, Σ^*
- more examples:

$$L = \{w \in \Sigma^* : w \text{ has some property } P\}$$
$$L = \{w \in \Sigma^* : w = w^R\} \quad (w^R = \text{reverse of } w)$$
$$L = \{w \in \{0,1\}^* : w \text{ has an equal number of 0's and 1's}\}$$

Preliminaries

alphabet: a finite set of symbols

- examples: Roman alphabet $\{a, b, \ldots, z\}$, binary alphabet $\{0, 1\}$

string: a finite sequence of symbols over an alphabet

- examples: $science$, 0011101
- ϵ: the empty string
- Σ^*: the set of all strings over an alphabet Σ (including ϵ)

language: a set strings over an alphabet Σ (i.e., a subset of Σ^*)

- examples: $\emptyset$, Σ, Σ^*
- more examples:

  $L = \{w \in \Sigma^* : w$ has some property $P\}$
  $L = \{w \in \Sigma^* : w = w^R\}$ $\quad$ (w^R = reverse of w)
  $L = \{w \in \{0, 1\}^* : w$ has an equal number of 0's and 1's$\}$
  $L = \{w \in \{1, 2, \ldots, n\} : w$ is a permutation of $\{1, 2, \ldots, n\}$
  corresponding to a Hamiltonian Path$\}$

Preliminaries

Decision problem: a problem that can be posed as an yes/no question

- example: Given a number n, is n prime?

Preliminaries

Decision problem: a problem that can be posed as an yes/no question

- example: Given a number n, is n prime?
- other examples???

Preliminaries

Decision problem: a problem that can be posed as an yes/no question

- example: Given a number n, is n prime?
- other examples:
  - Given a set of numbers $A = \{a_1, a_2, \ldots, a_n\}$, does the permutation π correspond to an increasing ordering of the numbers in A? (i.e., is it true that $a_{\pi(1)} < a_{\pi(2)} < \ldots < a_{\pi(n)}$)

Preliminaries

Decision problem: a problem that can be posed as an yes/no question

- example: Given a number n, is n prime?
- other examples:
  - Given a set of numbers $A = \{a_1, a_2, \ldots, a_n\}$, does the permutation π correspond to an increasing ordering of the numbers in A? (i.e., is it true that $a_{\pi(1)} < a_{\pi(2)} < \ldots < a_{\pi(n)}$)
  - Given a graph $G = (V, E)$, is there a permutation π of the vertex set such that $(v_{\pi(i)}, v_{\pi(i+1)}) \in E$ for all i, $1 \leq i \leq |V - 1|$?

Preliminaries

Decision problem: a problem that can be posed as an yes/no question

- example: Given a number n, is n prime?
- other examples:
  - Given a set of numbers $A = \{a_1, a_2, \ldots, a_n\}$, does the permutation π correspond to an increasing ordering of the numbers in A? (i.e., is it true that $a_{\pi(1)} < a_{\pi(2)} < \ldots < a_{\pi(n)}$)
  - Given a graph $G = (V, E)$, is there a permutation π of the vertex set such that $(v_{\pi(i)}, v_{\pi(i+1)}) \in E$ for all i, $1 \leq i \leq |V - 1|$? (Hamiltonian Path)

Preliminaries

Decision problem: a problem that can be posed as an yes/no question

- example: Given a number n, is n prime?
- other examples:
  - Given a set of numbers $A = \{a_1, a_2, \ldots, a_n\}$, does the permutation π correspond to an increasing ordering of the numbers in A? (i.e., is it true that $a_{\pi(1)} < a_{\pi(2)} < \ldots < a_{\pi(n)}$)
  - Given a graph $G = (V, E)$, is there a permutation π of the vertex set such that $(v_{\pi(i)}, v_{\pi(i+1)}) \in E$ for all i, $1 \leq i \leq |V - 1|$? (Hamiltonian Path)

Optimization problem: a problem of searching for the best answer

Preliminaries

Decision problem: a problem that can be posed as an yes/no question

- example: Given a number n, is n prime?
- other examples:
  - Given a set of numbers $A = \{a_1, a_2, \ldots, a_n\}$, does the permutation π correspond to an increasing ordering of the numbers in A? (i.e., is it true that $a_{\pi(1)} < a_{\pi(2)} < \ldots < a_{\pi(n)}$)
  - Given a graph $G = (V, E)$, is there a permutation π of the vertex set such that $(v_{\pi(i)}, v_{\pi(i+1)}) \in E$ for all i, $1 \leq i \leq |V - 1|$? (Hamiltonian Path)

Optimization problem: a problem of searching for the best answer

- example: Given a graph $G = (V, E)$, two vertices $s, t \in V$ and an integer distance $d(e)$ for each $e \in E$, find the path p between s and t such that the sum of distances of the edges in p is minimized.

Preliminaries

Decision problem: a problem that can be posed as an yes/no question

- example: Given a number n, is n prime?
- other examples:
  - Given a set of numbers $A = \{a_1, a_2, \ldots, a_n\}$, does the permutation π correspond to an increasing ordering of the numbers in A? (i.e., is it true that $a_{\pi(1)} < a_{\pi(2)} < \ldots < a_{\pi(n)}$)
  - Given a graph $G = (V, E)$, is there a permutation π of the vertex set such that $(v_{\pi(i)}, v_{\pi(i+1)}) \in E$ for all i, $1 \leq i \leq |V - 1|$? (Hamiltonian Path)

Optimization problem: a problem of searching for the best answer

- example: Given a graph $G = (V, E)$, two vertices $s, t \in V$ and an integer distance $d(e)$ for each $e \in E$, find the path p between s and t such that the sum of distances of the edges in p is minimized.
- decision version: Given a graph $G = (V, E)$, two vertices $s, t \in V$, an integer distance $d(e)$ for each $e \in E$ **and an integer** D, is there a path p between s and t such that the sum of distances of the edges in p is at most D?

Preliminaries

Observation 1:
In most of these lectures we will deal with decision problems

Preliminaries

Observation 1:

In most of these lectures we will deal with decision problems

Observation 2:

A decision problem is defined by the input and the yes/no question

- examples of input:
  - Given a set of numbers $A = \{a_1, a_2, \ldots, a_n\}$
  - Given a graph $G = (V, E)$
  - Given a graph $G = (V, E)$ and a positive weight $w(e)$ for each $e \in E$

Preliminaries

Observation 1:
In most of these lectures we will deal with decision problems

Observation 2:
A decision problem is defined by the input and the yes/no question

- examples of input:
  - Given a set of numbers $A = \{a_1, a_2, \ldots, a_n\}$
  - Given a graph $G = (V, E)$
  - Given a graph $G = (V, E)$ and a positive weight $w(e)$ for each $e \in E$
- $< I >$: string encoding of the input
  - $< a_1, a_2, \ldots, a_n >$
  - $<$ adjacency matrix of $G >$
  - $<$ adjacency matrix of $G, w(e)\ \forall e \in E >$

Preliminaries

Observation 1:

In most of these lectures we will deal with decision problems

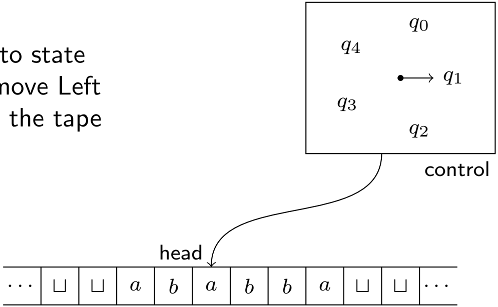
Observation 2:

A decision problem is defined by the input and the yes/no question

- examples of input:
  - Given a set of numbers $A = \{a_1, a_2, \ldots, a_n\}$
  - Given a graph $G = (V, E)$
  - Given a graph $G = (V, E)$ and a positive weight $w(e)$ for each $e \in E$
- $< I >$: string encoding of the input
  - $< a_1, a_2, \ldots, a_n >$
  - $<$ adjacency matrix of $G >$
  - $<$ adjacency matrix of $G, w(e)\ \forall e \in E >$
- $|I|$: size of the input (in binary)
  - $\log_2 a_1 + \log_2 a_2 + \ldots \log_2 a_n$
  - $|V|^2$
  - $|V|^2 + \sum_{e \in E} \log_2 w(e)$

Turing machine

- memory: an infinite tape
  - initially, it contains the input string
  - move the head left or right
  - read and/or write to current cell
- control states
  - finite number of them
  - one current state
- At each step:
  – move from state to state
  – read or write or move Left
  or move Right in the tape



Turing machine: formal definition

A Turing Machine (M) is a sextuple $(K, \Sigma, \Gamma, \delta, s, H)$, where

- K is a finite set of states
- Σ is the input alphabet not containing the *blank* symbol $\sqcup$
- Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$
- $s \in K$: the initial state
- $H \subseteq K$: the set of halting states
- δ: the transition function from $(K \setminus H) \times \Gamma$ to $K \times (\Gamma \cup \{\leftarrow, \rightarrow\})$

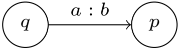
Turing machine: formal definition

A Turing Machine (M) is a sextuple $(K, \Sigma, \Gamma, \delta, s, H)$, where

- K is a finite set of states
- Σ is the input alphabet not containing the *blank* symbol $\sqcup$
- Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$
- $s \in K$: the initial state
- $H \subseteq K$: the set of halting states
- δ: the transition function from $(K \setminus H) \times \Gamma$ to $K \times (\Gamma \cup \{\leftarrow, \rightarrow\})$

In general, $\delta(q, a) = (p, b)$ means that when M is in the state q and reads a in the tape, it goes to the state p and

– if $b \in \Sigma$, writes b in the place of a
– if $b \in \{\leftarrow, \rightarrow\}$, moves the head either Left or Right



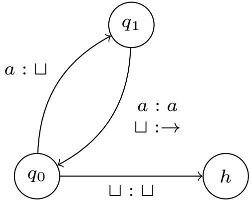
A first example

Consider the Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ where

$$K = \{q_0, q_1, h\}, \quad \Sigma = \{a\}, \quad \Gamma = \{a, \sqcup\}, \quad s = q_0, \quad H = \{h\},$$

and δ is given by the table. How does M proceed?

q	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$



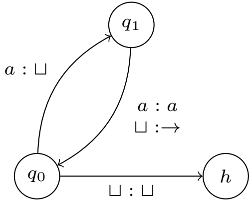
A first example

Consider the Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ where

$$K = \{q_0, q_1, h\}, \quad \Sigma = \{a\}, \quad \Gamma = \{a, \sqcup\}, \quad s = q_0, \quad H = \{h\},$$

and δ is given by the table. How does M proceed?

q	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$



$$(q_0, \underline{a}aa)$$

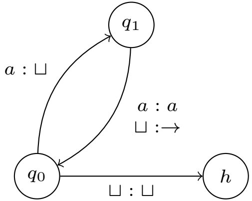
A first example

Consider the Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ where

$$K = \{q_0, q_1, h\}, \quad \Sigma = \{a\}, \quad \Gamma = \{a, \sqcup\}, \quad s = q_0, \quad H = \{h\},$$

and δ is given by the table. How does M proceed?

q	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$



$$(q_0, \underline{a}aa) \quad \vdash_M \quad (q_1, \underline{\sqcup}aa)$$

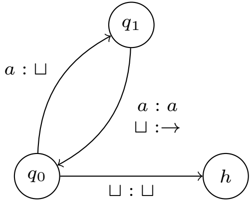
A first example

Consider the Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ where

$$K = \{q_0, q_1, h\}, \quad \Sigma = \{a\}, \quad \Gamma = \{a, \sqcup\}, \quad s = q_0, \quad H = \{h\},$$

and δ is given by the table. How does M proceed?

q	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$



$$(q_0, \underline{a}aa) \vdash_M (q_1, \underline{\sqcup}aa) \vdash_M (q_0, \sqcup\underline{a}a)$$

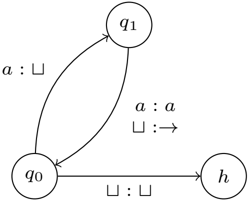
A first example

Consider the Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ where

$$K = \{q_0, q_1, h\}, \quad \Sigma = \{a\}, \quad \Gamma = \{a, \sqcup\}, \quad s = q_0, \quad H = \{h\},$$

and δ is given by the table. How does M proceed?

q	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$



$$\begin{aligned} (q_0, \underline{a}aa) \quad &\vdash_M \quad (q_1, \underline{\sqcup}aa) \;\vdash_M\; (q_0, \sqcup\underline{a}a) \\ &\vdash_M \quad (q_1, \sqcup\underline{\sqcup}a) \end{aligned}$$

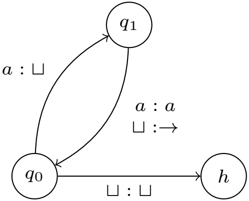
A first example

Consider the Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ where

$$K = \{q_0, q_1, h\}, \quad \Sigma = \{a\}, \quad \Gamma = \{a, \sqcup\}, \quad s = q_0, \quad H = \{h\},$$

and δ is given by the table. How does M proceed?

q	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$



$$\begin{aligned}(q_0, \underline{a}aa) &\vdash_M (q_1, \underline{\sqcup}aa) \vdash_M (q_0, \sqcup\underline{a}a) \\ &\vdash_M (q_1, \sqcup\underline{\sqcup}a) \vdash_M (q_0, \sqcup \sqcup \underline{a})\end{aligned}$$

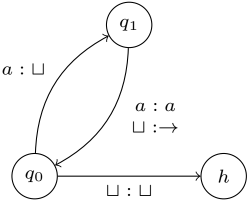
A first example

Consider the Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ where

$$K = \{q_0, q_1, h\}, \quad \Sigma = \{a\}, \quad \Gamma = \{a, \sqcup\}, \quad s = q_0, \quad H = \{h\},$$

and δ is given by the table. How does M proceed?

q	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$



$$\begin{aligned}
(q_0, \underline{a}aa) \;&\vdash_M\; (q_1, \underline{\sqcup}aa) \vdash_M (q_0, \sqcup\underline{a}a) \\
&\vdash_M\; (q_1, \sqcup\underline{\sqcup}a) \vdash_M (q_0, \sqcup\sqcup\underline{a}) \\
&\vdash_M\; (q_1, \sqcup\sqcup\underline{\sqcup})
\end{aligned}$$

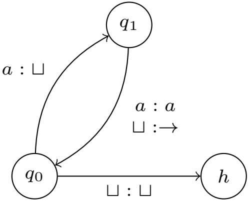
A first example

Consider the Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ where

$$K = \{q_0, q_1, h\}, \quad \Sigma = \{a\}, \quad \Gamma = \{a, \sqcup\}, \quad s = q_0, \quad H = \{h\},$$

and δ is given by the table. How does M proceed?

q	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$



$$\begin{aligned}
(q_0, \underline{a}aa) \quad &\vdash_M \quad (q_1, \underline{\sqcup}aa) \ \vdash_M \ (q_0, \sqcup\underline{a}a) \\
&\vdash_M \quad (q_1, \sqcup\underline{\sqcup}a) \ \vdash_M \ (q_0, \sqcup\sqcup\underline{a}) \\
&\vdash_M \quad (q_1, \sqcup\sqcup\underline{\sqcup}) \ \vdash_M \ (q_0, \sqcup\sqcup\sqcup\underline{\sqcup})
\end{aligned}$$

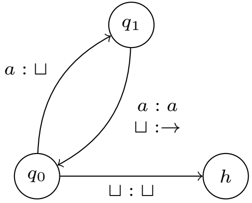
A first example

Consider the Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ where

$$K = \{q_0, q_1, h\}, \quad \Sigma = \{a\}, \quad \Gamma = \{a, \sqcup\}, \quad s = q_0, \quad H = \{h\},$$

and δ is given by the table. How does M proceed?

q	σ	$\delta(q,\sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$



$$\begin{aligned}
(q_0, \underline{a}aa) &\vdash_M (q_1, \underline{\sqcup}aa) \vdash_M (q_0, \sqcup\underline{a}a) \\
&\vdash_M (q_1, \sqcup\underline{\sqcup}a) \vdash_M (q_0, \sqcup\sqcup\underline{a}) \\
&\vdash_M (q_1, \sqcup\sqcup\underline{\sqcup}) \vdash_M (q_0, \sqcup\sqcup\sqcup\underline{\sqcup}) \\
&\vdash_M (h, \sqcup\sqcup\sqcup\underline{\sqcup})
\end{aligned}$$

Formalize the notation

Definition

A **configuration** of a Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ is a member of $K \times \Gamma^* \times \Gamma^*((\Gamma \setminus \{\sqcup\}) \cup \{\epsilon\})$.

- informally: a triplet describing
  - the current state
  - the contents of the tape on the left of the head (including head's position)
  - the contents of the tape on the right of the head

Formalize the notation

Definition

A **configuration** of a Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ is a member of $K \times \Gamma^* \times \Gamma^*((\Gamma \setminus \{\sqcup\}) \cup \{\epsilon\})$.

- informally: a triplet describing
  - the current state
  - the contents of the tape on the left of the head (including head's position)
  - the contents of the tape on the right of the head
- example: $(q_1, \sqcup a, a)$ or simply $(q_1, \sqcup\underline{a}a)$ or simply $(q_1, \underline{a}a)$

Formalize the notation

Definition

A **configuration** of a Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ is a member of $K \times \Gamma^* \times \Gamma^*((\Gamma \setminus \{\sqcup\}) \cup \{\epsilon\})$.

- informally: a triplet describing
  - the current state
  - the contents of the tape on the left of the head (including head's position)
  - the contents of the tape on the right of the head
- example: $(q_1, \sqcup a, a)$ or simply $(q_1, \sqcup\underline{a}a)$ or simply $(q_1, \underline{a}a)$

Initial configuration: $(s, \underline{a}w)$ where $M = (K, \Sigma, \Gamma, \delta, s, H)$ is a Turing Machine, $a \in \Sigma$, $w \in \Sigma^*$ and aw is the *input string*

Formalize the notation

Definition

A **configuration** of a Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ is a member of $K \times \Gamma^* \times \Gamma^*((\Gamma \setminus \{\sqcup\}) \cup \{\epsilon\})$.

- informally: a triplet describing
  - the current state
  - the contents of the tape on the left of the head (including head's position)
  - the contents of the tape on the right of the head
- example: $(q_1, \sqcup a, a)$ or simply $(q_1, \sqcup\underline{a}a)$ or simply $(q_1, \underline{a}a)$

Initial configuration: $(s, \underline{a}w)$ where $M = (K, \Sigma, \Gamma, \delta, s, H)$ is a Turing Machine, $a \in \Sigma$, $w \in \Sigma^*$ and aw is the *input string*

Halted configuration: a configuration whose state belongs to H

- example: $(h, \sqcup \sqcup \sqcup\sqcup, \epsilon)$ or simply $(h, \sqcup \sqcup \sqcup\underline{\sqcup})$ or simply $(h, \underline{\sqcup})$

Formalize the notation

Definition

Consider a Turing Machine M and two configurations C_1 and C_2 of M. If M can go from C_1 to C_2 in a *single step*, then we write

$$C_1 \vdash_M C_2$$

Formalize the notation

Definition

Consider a Turing Machine M and two configurations C_1 and C_2 of M. If M can go from C_1 to C_2 in a *single step*, then we write

$$C_1 \vdash_M C_2$$

Definition

Consider a Turing Machine M and two configurations C_1 and C_2 of M. If M can go from C_1 to C_2 using a *sequence* of configurations, then we say that C_1 **yields** C_2 and we write

$$C_1 \vdash_M^* C_2$$

Formalize the notation

Definition

Consider a Turing Machine M and two configurations C_1 and C_2 of M. If M can go from C_1 to C_2 in a *single step*, then we write

$$C_1 \vdash_M C_2$$

Definition

Consider a Turing Machine M and two configurations C_1 and C_2 of M. If M can go from C_1 to C_2 using a *sequence* of configurations, then we say that C_1 **yields** C_2 and we write

$$C_1 \vdash_M^* C_2$$

Definition

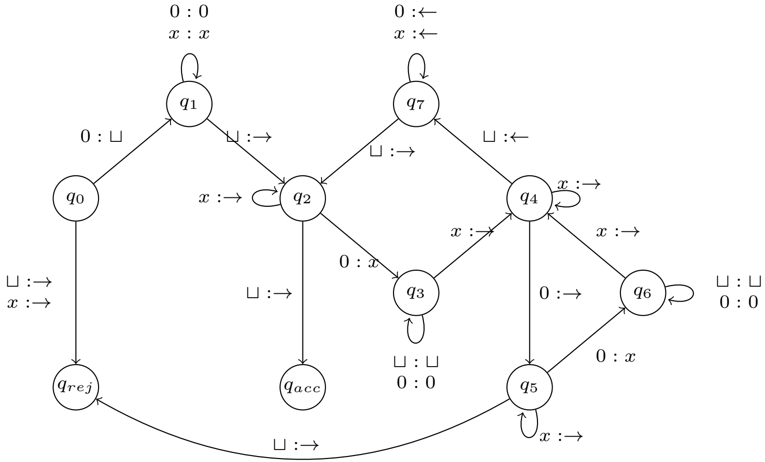
A **computation** of a Turing Machine M is a sequence of configurations $C_0, C_1, \ldots, C_n$, for some $n \geq 0$, such that

$$C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M \ldots \vdash_M C_n$$

The **length** of the computation is n (or it performs n steps).

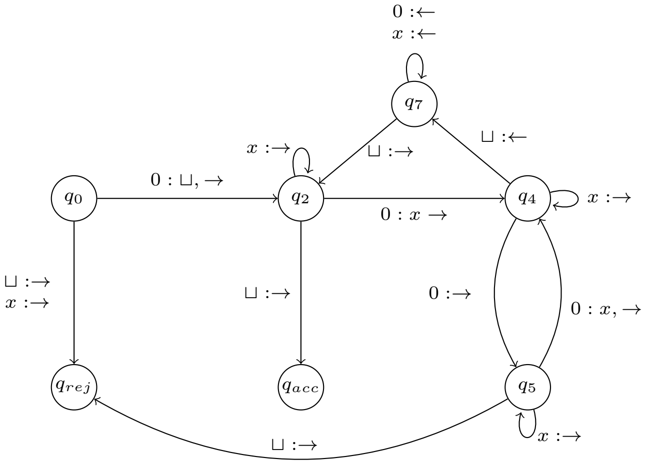
A second example

$$\Sigma = \{0\}, \quad \Gamma = \{0, x, \sqcup\}, \quad s = q_0, \quad H = \{q_{acc}, q_{rej}\}$$



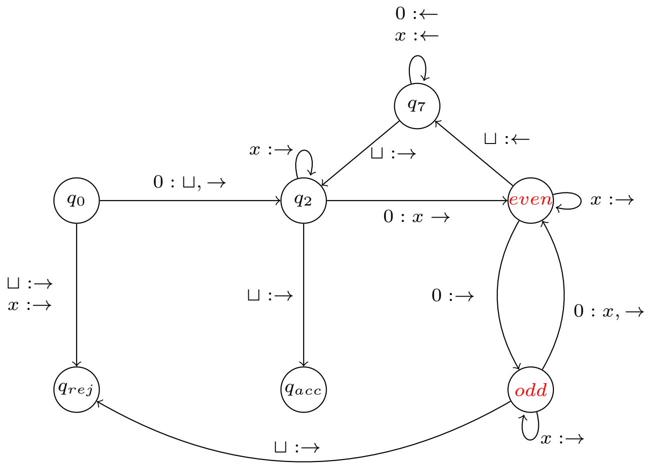
A second example

$$\Sigma = \{0\}, \quad \Gamma = \{0, x, \sqcup\}, \quad s = q_0, \quad H = \{q_{acc}, q_{rej}\}$$



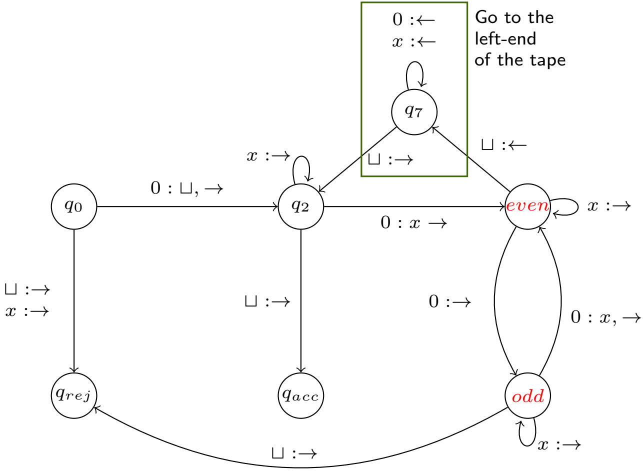
A second example

$$\Sigma = \{0\}, \quad \Gamma = \{0, x, \sqcup\}, \quad s = q_0, \quad H = \{q_{acc}, q_{rej}\}$$



A second example

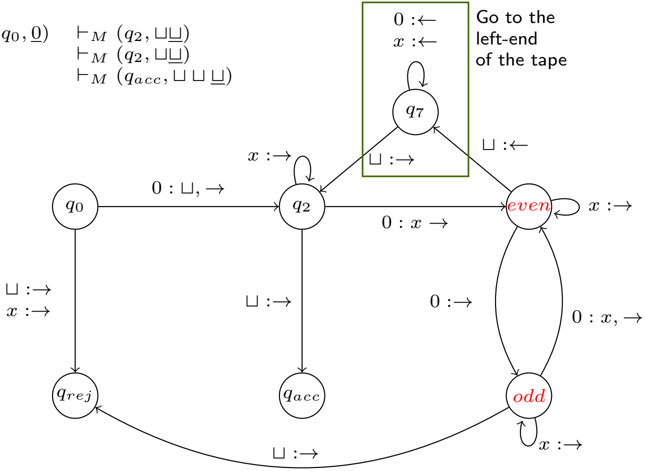
$$\Sigma = \{0\}, \quad \Gamma = \{0, x, \sqcup\}, \quad s = q_0, \quad H = \{q_{acc}, q_{rej}\}$$



A second example

$$\Sigma = \{0\}, \quad \Gamma = \{0, x, \sqcup\}, \quad s = q_0, \quad H = \{q_{acc}, q_{rej}\}$$

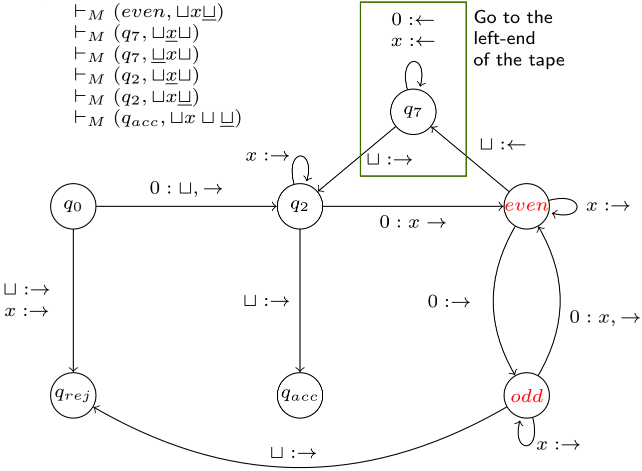
$$\begin{aligned} (q_0, \underline{0}) &\vdash_M (q_2, \sqcup\underline{\sqcup}) \\ &\vdash_M (q_2, \sqcup\underline{\sqcup}) \\ &\vdash_M (q_{acc}, \sqcup \sqcup \underline{\sqcup}) \end{aligned}$$



A second example

$$\Sigma = \{0\}, \quad \Gamma = \{0, x, \sqcup\}, \quad s = q_0, \quad H = \{q_{acc}, q_{rej}\}$$

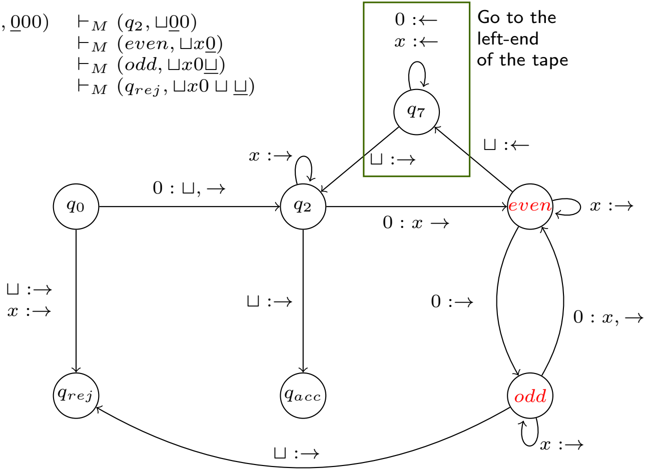
$$\begin{aligned}
(q_0, \underline{0}0) &\vdash_M (q_2, \sqcup\underline{0}) \\
&\vdash_M (even, \sqcup x\underline{\sqcup}) \\
&\vdash_M (q_7, \sqcup\underline{x}\sqcup) \\
&\vdash_M (q_7, \underline{\sqcup}x\sqcup) \\
&\vdash_M (q_2, \sqcup\underline{x}\sqcup) \\
&\vdash_M (q_2, \sqcup x\underline{\sqcup}) \\
&\vdash_M (q_{acc}, \sqcup x \sqcup \underline{\sqcup})
\end{aligned}$$



A second example

$$\Sigma = \{0\}, \quad \Gamma = \{0, x, \sqcup\}, \quad s = q_0, \quad H = \{q_{acc}, q_{rej}\}$$

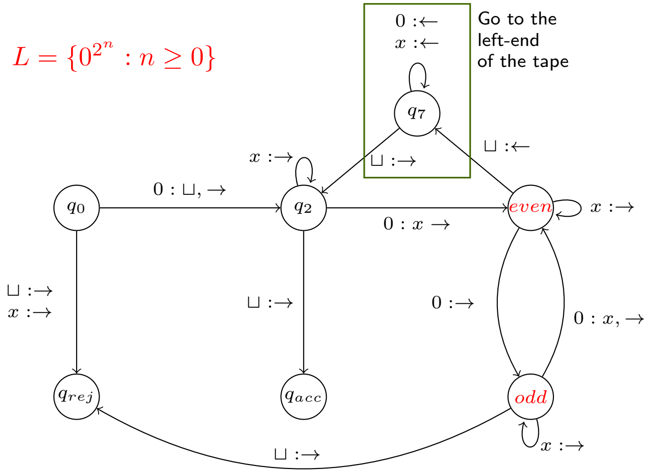
$$\begin{aligned}
(q_0, \underline{0}00) &\vdash_M (q_2, \sqcup\underline{0}0) \\
&\vdash_M (even, \sqcup x\underline{0}) \\
&\vdash_M (odd, \sqcup x0\underline{\sqcup}) \\
&\vdash_M (q_{rej}, \sqcup x0 \sqcup \underline{\sqcup})
\end{aligned}$$



A second example

$$\Sigma = \{0\}, \quad \Gamma = \{0, x, \sqcup\}, \quad s = q_0, \quad H = \{q_{acc}, q_{rej}\}$$

$L = \{0^{2^n} : n \geq 0\}$

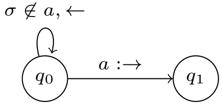


Exercise

Construct the Turing Machine that *accepts* the language

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

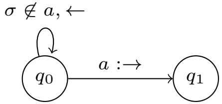
A more general notation for Turing Machines



Turing Machine $L_a = (K, \Sigma, \Gamma, \delta, s, H)$ where:

- $K = \{q_0, q_1\}$
- $a \in \Sigma$
- $s = q_0$
- $H = \{q_1\}$

A more general notation for Turing Machines

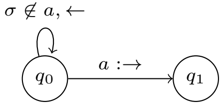


Turing Machine $L_a = (K, \Sigma, \Gamma, \delta, s, H)$ where:

- $K = \{q_0, q_1\}$
- $a \in \Sigma$
- $s = q_0$
- $H = \{q_1\}$

- Define similar simple Turing Machines
  - examples: L, R, L_a, R_a, L^2, R^2, a, $\sqcup$, etc

A more general notation for Turing Machines



Turing Machine $L_a = (K, \Sigma, \Gamma, \delta, s, H)$ where:

- $K = \{q_0, q_1\}$
- $a \in \Sigma$
- $s = q_0$
- $H = \{q_1\}$

- Define similar simple Turing Machines
  - examples: L, R, L_a, R_a, L^2, R^2, a, $\sqcup$, etc
- Combine simple machines to construct more complicated ones

$$\begin{tikzcd} M_3 & \\ M_1 \arrow[u, "b"'] \arrow[r, "a"] & M_2 \end{tikzcd}$$

1. Run M_1
2. If M_1 finishes and the head reads a then run M_2 starting from this a
3. Else run M_3 starting from this b

Example

What is the goal of the following Turing Machine?

$$
\begin{tikzcd}
> L_{\sqcup} \arrow[r] & R \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R^2_{\sqcup} a L^2_{\sqcup} a \arrow[l, bend right] \\
& R_{\sqcup} &
\end{tikzcd}
$$

Example

What is the goal of the following Turing Machine?

$$
\begin{tikzcd}
> L_{\sqcup} \arrow[r] & R \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R_{\sqcup}^{2} a L_{\sqcup}^{2} a \arrow[l, bend right] \\
 & R_{\sqcup} &
\end{tikzcd}
$$

$$(\sqcup abc\underline{\sqcup}) \quad \vdash_M^* \quad (\underline{\sqcup}abc\sqcup) \qquad\qquad (L_{\sqcup})$$

Example

What is the goal of the following Turing Machine?

$$\begin{tikzcd} > L_{\sqcup} \arrow[r] & R \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R^2_{\sqcup} a L^2_{\sqcup} a \arrow[l, bend right] \\ & R_{\sqcup} & \end{tikzcd}$$

$$\begin{array}{llll} (\sqcup abc\underline{\sqcup}) & \vdash_M^* & (\underline{\sqcup}abc\sqcup) & (L_{\sqcup}) \\ & \vdash_M & (\sqcup\underline{a}bc\sqcup) & (R) \end{array}$$

Example

What is the goal of the following Turing Machine?

$$
\begin{tikzcd}
> L_{\sqcup} \arrow[r] & R \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R^2_{\sqcup} a L^2_{\sqcup} a \arrow[l, bend right] \\
& R_{\sqcup} &
\end{tikzcd}
$$

$$
\begin{array}{llll}
(\sqcup abc\underline{\sqcup}) & \vdash^*_M & (\underline{\sqcup}abc\sqcup) & (L_{\sqcup}) \\
& \vdash_M & (\sqcup\underline{a}bc\sqcup) & (R) \\
& \vdash_M & (\sqcup\underline{\sqcup}bc\sqcup) & (\sqcup)
\end{array}
$$

Example

What is the goal of the following Turing Machine?

$$
\begin{tikzcd}
> L_{\sqcup} \arrow[r] & R \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R_{\sqcup}^{2} a L_{\sqcup}^{2} a \arrow[l, bend right, to path={|- ([yshift=3em]\tikztostart.north) -| (\tikztotarget)}] \\
 & R_{\sqcup} &
\end{tikzcd}
$$

$$
\begin{aligned}
(\sqcup abc\underline{\sqcup}) \quad & \vdash_M^* \quad (\underline{\sqcup}abc\sqcup) && (L_{\sqcup}) \\
& \vdash_M \quad (\sqcup\underline{a}bc\sqcup) && (R) \\
& \vdash_M \quad (\sqcup\underline{\sqcup}bc\sqcup) && (\sqcup) \\
& \vdash_M^* \quad (\sqcup \sqcup bc \sqcup \underline{\sqcup}) && (R_{\sqcup}^{2})
\end{aligned}
$$

Example

What is the goal of the following Turing Machine?

$$\begin{tikzcd}
> L_{\sqcup} \arrow[r] & R \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R_{\sqcup}^{2} a L_{\sqcup}^{2} a \arrow[l, bend right] \\
& R_{\sqcup} &
\end{tikzcd}$$

$$\begin{array}{llll}
(\sqcup abc\underline{\sqcup}) & \vdash_M^* & (\underline{\sqcup}abc\sqcup) & (L_{\sqcup}) \\
& \vdash_M & (\sqcup\underline{a}bc\sqcup) & (R) \\
& \vdash_M & (\sqcup\underline{\sqcup}bc\sqcup) & (\sqcup) \\
& \vdash_M^* & (\sqcup \sqcup bc \sqcup \underline{\sqcup}) & (R_{\sqcup}^{2}) \\
& \vdash_M & (\sqcup \sqcup bc \sqcup \underline{a}) & (a)
\end{array}$$

Example

What is the goal of the following Turing Machine?

$$\begin{tikzcd}
> L_{\sqcup} \arrow[r] & R \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R^2_{\sqcup} a L^2_{\sqcup} a \arrow[l, bend right] \\
& R_{\sqcup} &
\end{tikzcd}$$

$$\begin{array}{llll}
(\sqcup abc\underline{\sqcup}) & \vdash_M^* & (\underline{\sqcup}abc\sqcup) & (L_{\sqcup}) \\
& \vdash_M & (\sqcup\underline{a}bc\sqcup) & (R) \\
& \vdash_M & (\sqcup\underline{\sqcup}bc\sqcup) & (\sqcup) \\
& \vdash_M^* & (\sqcup \sqcup bc \sqcup \underline{\sqcup}) & (R^2_{\sqcup}) \\
& \vdash_M & (\sqcup \sqcup bc \sqcup \underline{a}) & (a) \\
& \vdash_M^* & (\sqcup\underline{\sqcup}bc \sqcup a) & (L^2_{\sqcup})
\end{array}$$

Example

What is the goal of the following Turing Machine?

$$
\begin{tikzcd}
> L_{\sqcup} \arrow[r] & R \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R_{\sqcup}^{2} a L_{\sqcup}^{2} a \arrow[l, bend right] \\
 & R_{\sqcup} &
\end{tikzcd}
$$

$$
\begin{array}{llll}
(\sqcup abc\underline{\sqcup}) & \vdash_M^* & (\underline{\sqcup}abc\sqcup) & (L_{\sqcup}) \\
 & \vdash_M & (\sqcup\underline{a}bc\sqcup) & (R) \\
 & \vdash_M & (\sqcup\underline{\sqcup}bc\sqcup) & (\sqcup) \\
 & \vdash_M^* & (\sqcup \sqcup bc \sqcup \underline{\sqcup}) & (R_{\sqcup}^2) \\
 & \vdash_M & (\sqcup \sqcup bc \sqcup \underline{a}) & (a) \\
 & \vdash_M^* & (\sqcup\underline{\sqcup}bc \sqcup a) & (L_{\sqcup}^2) \\
 & \vdash_M & (\sqcup\underline{a}bc \sqcup a) & (a)
\end{array}
$$

Example

What is the goal of the following Turing Machine?

$$\begin{tikzcd} > L_{\sqcup} \arrow[r] & R \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R^2_{\sqcup} a L^2_{\sqcup} a \arrow[l, bend right] \\ & R_{\sqcup} & \end{tikzcd}$$

$$\begin{array}{llll}
(\sqcup abc\underline{\sqcup}) & \vdash_M^* & (\underline{\sqcup}abc\sqcup) & (L_{\sqcup}) \\
 & \vdash_M & (\sqcup\underline{a}bc\sqcup) & (R) \\
 & \vdash_M & (\sqcup\underline{\sqcup}bc\sqcup) & (\sqcup) \\
 & \vdash_M^* & (\sqcup \sqcup bc \sqcup \underline{\sqcup}) & (R^2_{\sqcup}) \\
 & \vdash_M & (\sqcup \sqcup bc \sqcup \underline{a}) & (a) \\
 & \vdash_M^* & (\sqcup\underline{\sqcup}bc \sqcup a) & (L^2_{\sqcup}) \\
 & \vdash_M & (\sqcup\underline{a}bc \sqcup a) & (a) \\
 & \vdash_M & (\sqcup a\underline{b}c \sqcup a) & (R)
\end{array}$$

Example

What is the goal of the following Turing Machine?

$$
\begin{tikzcd}
> L_{\sqcup} \arrow[r] & R \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R^2_{\sqcup} a L^2_{\sqcup} a \arrow[l, bend right] \\
 & R_{\sqcup} &
\end{tikzcd}
$$

$$
\begin{array}{llll}
(\sqcup abc\underline{\sqcup}) & \vdash^*_M & (\underline{\sqcup}abc\sqcup) & (L_{\sqcup}) \\
 & \vdash_M & (\sqcup\underline{a}bc\sqcup) & (R) \\
 & \vdash_M & (\sqcup\underline{\sqcup}bc\sqcup) & (\sqcup) \\
 & \vdash^*_M & (\sqcup \sqcup bc \sqcup \underline{\sqcup}) & (R^2_{\sqcup}) \\
 & \vdash_M & (\sqcup \sqcup bc \sqcup \underline{a}) & (a) \\
 & \vdash^*_M & (\sqcup\underline{\sqcup}bc \sqcup a) & (L^2_{\sqcup}) \\
 & \vdash_M & (\sqcup\underline{a}bc \sqcup a) & (a) \\
 & \vdash_M & (\sqcup a\underline{b}c \sqcup a) & (R)
\end{array}
$$

Solution:
transforms $\sqcup w\underline{\sqcup}$ to $\sqcup w \sqcup w\underline{\sqcup}$

Exercises

Construct the Turing Machines that implement the following operations

1. copy reversed (from $\sqcup w \underline{\sqcup}$ to $\sqcup w w^R \underline{\sqcup}$)
2. right shift (from $\sqcup w \underline{\sqcup}$ to $\sqcup \sqcup w \underline{\sqcup}$)
3. left shift (from $\sqcup w \underline{\sqcup}$ to $w \underline{\sqcup}$)
4. delete w (from $\sqcup w \underline{\sqcup}$ to $\sqcup \underline{\sqcup}$)

Generalize more the notation ...

High-level description

- give an algorithmic description of how the Turing Machine works in finite and discrete steps
- what is allowed?

Generalize more the notation ...

High-level description

- give an algorithmic description of how the Turing Machine works in finite and discrete steps
- what is allowed? $\quad\rightarrow\quad$ almost everything!!

Generalize more the notation ...

High-level description

- give an algorithmic description of how the Turing Machine works in finite and discrete steps
- what is allowed? $\rightarrow$ almost everything!!

Example

$M =$ "On input w:

1. scan the input from left to right to be sure that is member of $a^*b^*c^*$ and *reject* if not
2. find the leftmost a in the tape and if such an a does not exist, then
   - scan the input for a c and if such a c exists then *reject* else *accept*
3. replace a by $\hat{a}$
4. scan the input for the leftmost b and if such a b does not exist, then restore all b's (replace all $\hat{b}$ by b) and goto 2
5. replace b by $\hat{b}$
6. scan to the right for the first c and if such a c does not exist, then *reject*
7. replace c by $\hat{c}$ and goto 4"

Generalize more the notation ...

High-level description

- give an algorithmic description of how the Turing Machine works in finite and discrete steps
- what is allowed? $\rightarrow$ almost everything!!

Example

$L = \{a^i b^j c^k : i \times j = k\}$

$M =$ "On input w:

1. scan the input from left to right to be sure that is member of $a^*b^*c^*$ and *reject* if not
2. find the leftmost a in the tape and if such an a does not exist, then
   - scan the input for a c and if such a c exists then *reject* else *accept*
3. replace a by $\hat{a}$
4. scan the input for the leftmost b and if such a b does not exist, then restore all b's (replace all $\hat{b}$ by b) and goto 2
5. replace b by $\hat{b}$
6. scan to the right for the first c and if such a c does not exist, then *reject*
7. replace c by $\hat{c}$ and goto 4"

Exercise

Give the high-level description for a Turing Machine that accepts the following language

$$L = \{\#x_1\#x_2\# \ldots \#x_\ell : \text{ each } x_i \in \{0,1\}^* \text{ and } x_i \neq x_j \text{ for each } i \neq j\}$$

More exercises

Ex. 1 Consider the Turing Machine $M = (K, \Sigma, \Gamma, \delta, s, H)$ where $K = \{q_0, q_1, q_2, h\}$, $\Sigma = \{a\}$, $\Gamma = \{a, \sqcup, \#\}$, $s = q_0$, $H = \{h\}$ and δ is given by the following table. Let $n \geq 0$. Describe what M does when started in the configuration $(q_0, \#a^n a)$.

q	q_0	q_0	q_0	q_1	q_1	q_1	q_2	q_2	q_2
σ	a	$\sqcup$	$\#$	a	$\sqcup$	$\#$	a	$\sqcup$	$\#$
$\delta(q, \sigma)$	$(q_1, \leftarrow)$	$(q_0, \sqcup)$	$(q_0, \rightarrow)$	$(q_2, \sqcup)$	$(h, \sqcup)$	$(q_1, \rightarrow)$	(q_2, a)	$(q_0, \leftarrow)$	$(q_2, \rightarrow)$

Ex. 2 Give the full details of the following three Turing Machines.

$$> LL \qquad\qquad > R \qquad\qquad > L \xrightarrow{\sqcup} R$$

Ex. 3 Explain what the following Turing Machine does.

$$> R \xrightarrow{a \neq \sqcup} > R \xrightarrow{b \neq \sqcup} > R_\sqcup a R_\sqcup b$$

Ex. 4 Give the high-level definition of a Turing Machine that finds the maximum between three integers encoded in *unary*. Which is the length of the computation?

An application to draw and play with Turing Machines

http://www.jflap.org/