CS310 Automata Theory – 2016-2017

Nutan Limaye

Indian Institute of Technology, Bombay
nutan@cse.iitb.ac.in

Lecture 20: Turing machines, computability

March 06, 2017

Last two modules

Regular languages, NFA/DFA, Regular expressions, Myhill-Nerode relations.

Last two modules

Regular languages, NFA/DFA, Regular expressions, Myhill-Nerode relations.

2DFA: DFA + two-way head movement.

Last two modules

Regular languages, NFA/DFA, Regular expressions, Myhill-Nerode relations.

2DFA: DFA $+$ two-way head movement. They recognize exactly regular languages.

Last two modules

Regular languages, NFA/DFA, Regular expressions, Myhill-Nerode relations.

2DFA: DFA + two-way head movement. They recognize exactly regular languages.

Pushdown automata: NFA + Stack.

Last two modules

Regular languages, NFA/DFA, Regular expressions, Myhill-Nerode relations.

2DFA: DFA + two-way head movement. They recognize exactly regular languages.

Pushdown automata: NFA + Stack. The class of languages recognized by these is called Context-free languages (CFLs).

Last two modules

Regular languages, NFA/DFA, Regular expressions, Myhill-Nerode relations.

2DFA: DFA + two-way head movement. They recognize exactly regular languages.

Pushdown automata: NFA + Stack. The class of languages recognized by these is called Context-free languages (CFLs).

Context-free grammars: Recursive programs.

Last two modules

Regular languages, NFA/DFA, Regular expressions, Myhill-Nerode relations.

2DFA: DFA + two-way head movement. They recognize exactly regular languages.

Pushdown automata: NFA + Stack. The class of languages recognized by these is called Context-free languages (CFLs).

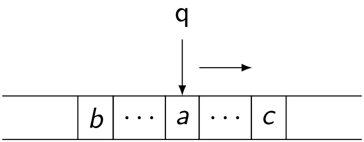
Context-free grammars: Recursive programs. The class of languages generated by these grammars is CFLs.

Turing machines

What is a Turing machine? (Informal description.)

Turing machines

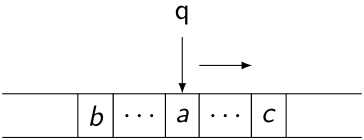
What is a Turing machine? (Informal description.)



Read and write on the input tape.

Turing machines

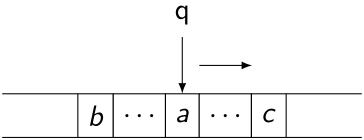
What is a Turing machine? (Informal description.)



Read and write on the input tape. Head moves left/right.

Turing machines

What is a Turing machine? (Informal description.)

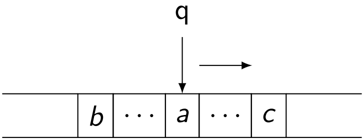


Read and write on the input tape. Head moves left/right.

The tape is infinite.

Turing machines

What is a Turing machine? (Informal description.)



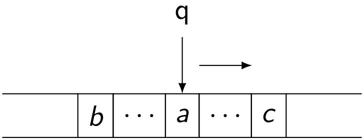
Read and write on the input tape. Head moves left/right.

The tape is infinite.

A special symbol & to indicate blank cells.

Turing machines

What is a Turing machine? (Informal description.)



Read and write on the input tape. Head moves left/right.

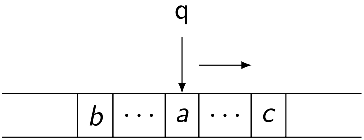
The tape is infinite.

A special symbol & to indicate blank cells.

Initially all cells blank except the part where the input is written.

Turing machines

What is a Turing machine? (Informal description.)



Read and write on the input tape. Head moves left/right.

The tape is infinite.

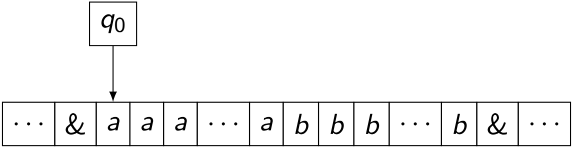
A special symbol & to indicate blank cells.

Initially all cells blank except the part where the input is written.

Special states for accepting and rejecting.

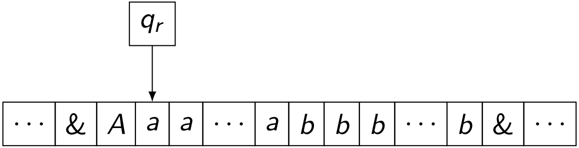
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



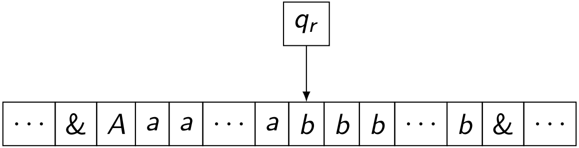
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



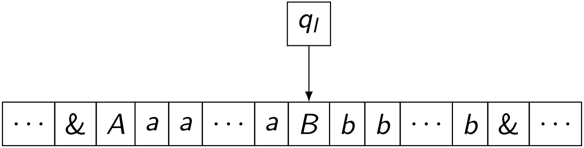
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



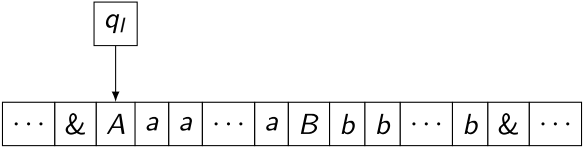
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



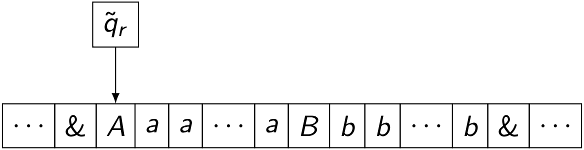
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



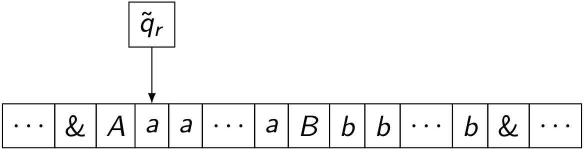
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



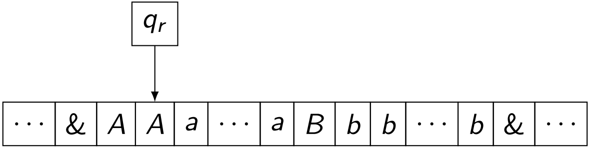
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



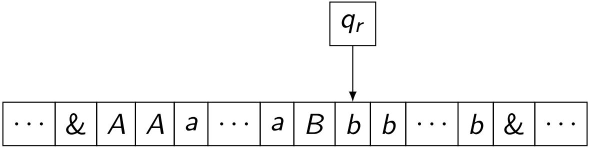
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



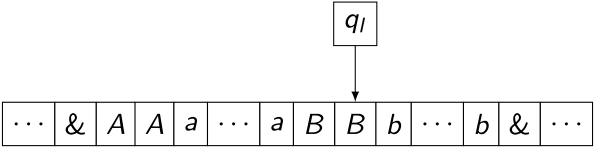
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



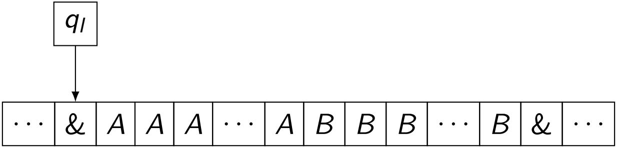
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



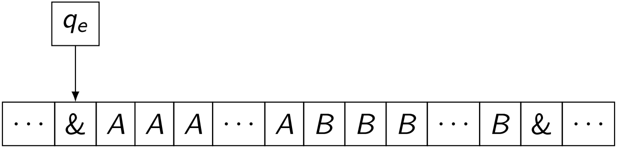
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



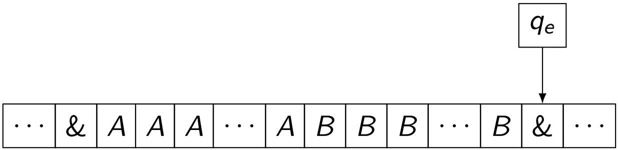
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



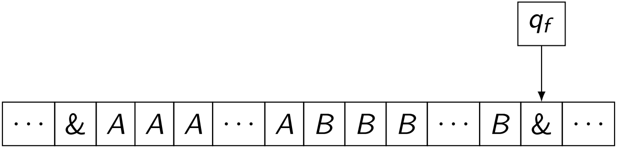
Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



Example

$L_{a,b} = \{a^n b^n \mid n \geq 0\}$.



Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q: set of states

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q: set of states Σ: input alphabet

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state		

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state	Γ:	tape alphabet

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state	Γ:	tape alphabet, $\Sigma \subseteq \Gamma$

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state	Γ:	tape alphabet, $\Sigma \subseteq \Gamma$, & $\in \Gamma$

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state	Γ:	tape alphabet, $\Sigma \subseteq \Gamma$, & $\in \Gamma$
q_{acc}:	accept state		

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state	Γ:	tape alphabet, $\Sigma \subseteq \Gamma$, & $\in \Gamma$
q_{acc}:	accept state	q_{rej}:	reject state

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state	Γ:	tape alphabet, $\Sigma \subseteq \Gamma$, & $\in \Gamma$
q_{acc}:	accept state	q_{rej}:	reject state

$$\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R\}.$$

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state	Γ:	tape alphabet, $\Sigma \subseteq \Gamma$, $\& \in \Gamma$
q_{acc}:	accept state	q_{rej}:	reject state

$$\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R\}.$$

Understanding δ

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state	Γ:	tape alphabet, $\Sigma \subseteq \Gamma$, & $\in \Gamma$
q_{acc}:	accept state	q_{rej}:	reject state

$$\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R\}.$$

Understanding δ

For a $q \in Q, a \in \Gamma$ if $\delta(q, a) = (p, b, L)$

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state	Γ:	tape alphabet, $\Sigma \subseteq \Gamma$, & $\in \Gamma$
q_{acc}:	accept state	q_{rej}:	reject state

$$\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R\}.$$

Understanding δ

For a $q \in Q, a \in \Gamma$ if $\delta(q, a) = (p, b, L)$,

then p is the new state of the machine

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state	Γ:	tape alphabet, $\Sigma \subseteq \Gamma$, $\& \in \Gamma$
q_{acc}:	accept state	q_{rej}:	reject state

$$\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R\}.$$

Understanding δ

For a $q \in Q, a \in \Gamma$ if $\delta(q, a) = (p, b, L)$,

then p is the new state of the machine,

b is the letter with which a gets overwritten

Formal definition

Definition

A Turing machine (TM) is given by $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f, q_{rej})$

Q:	set of states	Σ:	input alphabet
q_0:	start state	Γ:	tape alphabet, $\Sigma \subseteq \Gamma$, & $\in \Gamma$
q_{acc}:	accept state	q_{rej}:	reject state

$$\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R\}.$$

Understanding δ

For a $q \in Q, a \in \Gamma$ if $\delta(q, a) = (p, b, L)$,

then p is the new state of the machine,

b is the letter with which a gets overwritten,

the head moves to the left of the current position.

Turing machine for a non-context free language

Example

Turing machine for a non-context free language

Example

$$\mathrm{EQ} = \{w \cdot \# \cdot w \mid w \in \Sigma^*\}.$$

Turing machine for a non-context free language

Example

$$\text{EQ} = \{w \cdot \# \cdot w \mid w \in \Sigma^*\}.$$

Give a full description of a Turing machine for the above language.