Session 18

Pushdown Automata

Pushdown Automata (PDA)

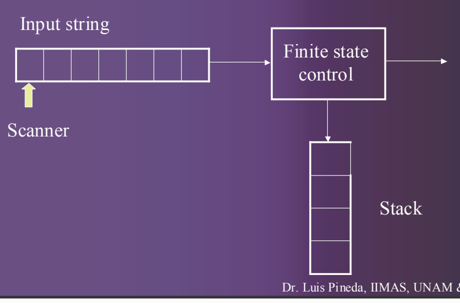
- PDA is a machine that defines CFG
- Extension of NFA-Λ with the addition of a stack.

A machine for accepting even *Pal*

- $L = \{xx^r \mid x \in \{a, b\}^* \}$
- $G_{pal\text{-}even} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid \Lambda\}$
  - $x = 101101$
- How can we tell if the string is in the language?

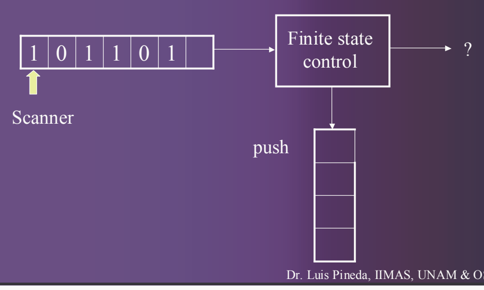
A machine for accepting *Pal*

- Initial state: Reading first part of the string



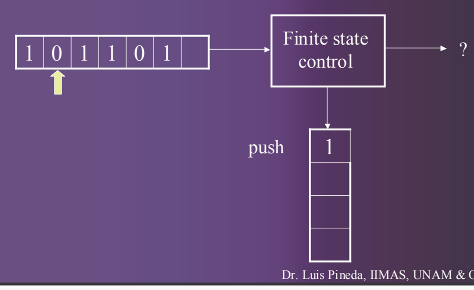
A machine for accepting *Pal*

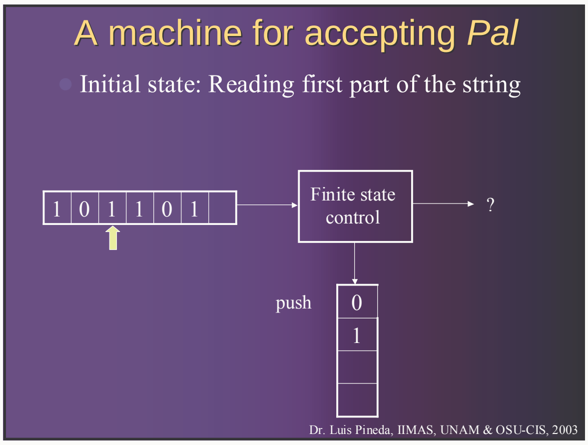
- Initial state: Reading first part of the string

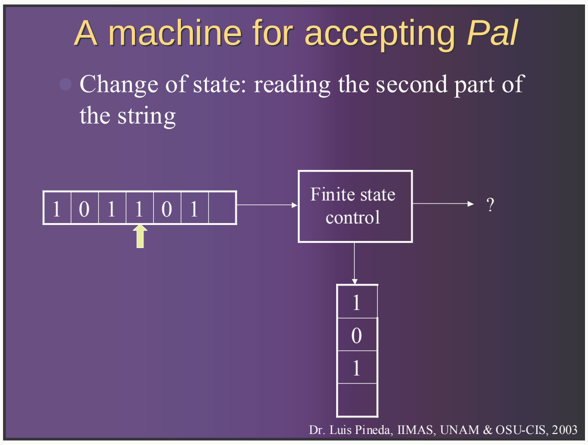


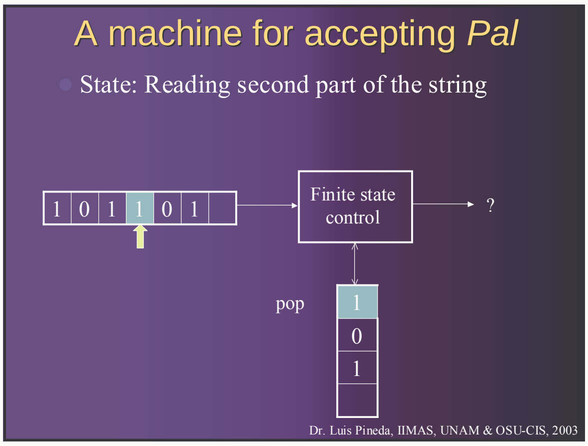
A machine for accepting *Pal*

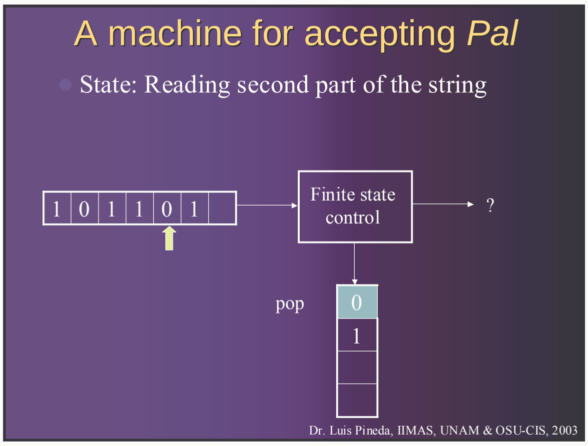
- Initial state: Reading first part of the string

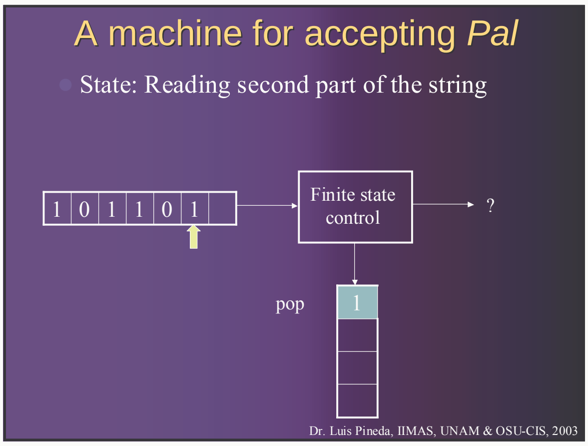


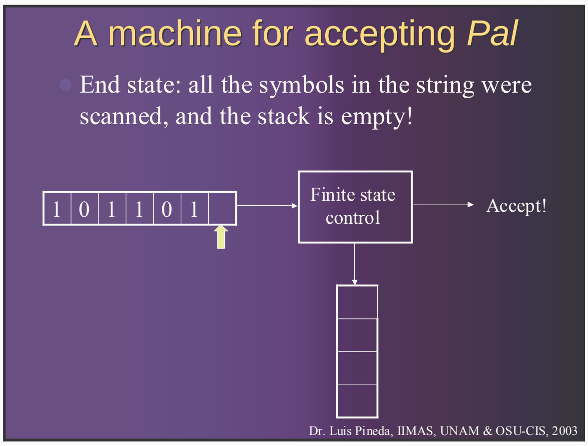

A machine for accepting Pal
Initial state: Reading first part of the string
1 0 1 1 0 1
Finite state control
?
push
0
1
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


A machine for accepting Pal
Change of state: reading the second part of the string
1 0 1 1 0 1
Finite state control
?
1
0
1
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


A machine for accepting Pal
State: Reading second part of the string
1 0 1 1 0 1
Finite state control
?
pop
1
0
1
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


A machine for accepting Pal
State: Reading second part of the string
1 0 1 1 0 1
Finite state control
?
pop
0
1
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


A machine for accepting Pal
State: Reading second part of the string
1 0 1 1 0 1
Finite state control
?
pop
1
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


A machine for accepting Pal
End state: all the symbols in the string were scanned, and the stack is empty!
1 0 1 1 0 1
Finite state control
Accept!
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Notion of a state

- In FA (DFA, NFA and NFA-Λ) the next state depends on the current state and the symbol on the input: the state is a full picture of the machine at a given point of the computation!
- In PDA, the full picture depends, in addition, of the content of the stack!

The machine is not deterministic

- How does the machine know when it has reached the end of the first half of the string?
  – It does not: it makes a guess!
- When every symbol is scanned, the machine either
  – Reads the symbol and do the corresponding op.
  – Makes a lambda transition and changes to the state for reading the second part of the string!
- How does it now that the stack is empty?
  – We use a special symbol in the stack!

Machine operation

- Depending:
  – Symbol currently been scanned
  – Current state
  – Symbol on the top of the stack
- Action:
  – Select next state (can be the same)
  – Action on the stack (push or pop or nothing)
- Accept:
  – All symbols in the string have been scanned
  – An accepting state is reached or the stack is empty

Definition of PDA

- A Pushdown Automaton is a 7-tuple:

$$(Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$$

where

– Q is a finite set (of states)
– Σ is the alphabet of the language rec. by the PDA
– Γ is the alphabet for symbols in the stack
– $q_0 \in Q$ (the initial state)
– $Z_0 \in \Gamma$ (the initial stack symbol)
– $A \subseteq Q$ (the set of accepting states)
– δ is a transition function of type:

$$Q \times (\Sigma \cup \{\Lambda\}) \times \Gamma \rightarrow \text{finite subsets of } Q \times \Gamma^*$$

Transition function

- Transition function for DFA:

$$\delta: Q \times \Sigma \rightarrow Q$$

- Transition function for NFA:

$$\delta: Q \times \Sigma \rightarrow 2^Q$$

- Transition function for NFA-Λ

$$\delta: Q \times (\Sigma \cup \{\Lambda\}) \rightarrow 2^Q$$

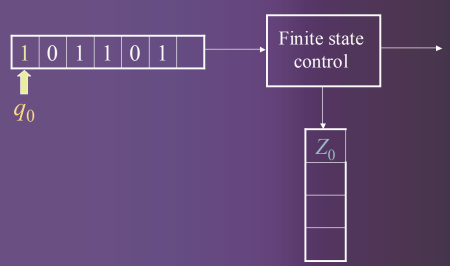
- Transition function for PDA:

$$\delta: Q \times (\Sigma \cup \{\Lambda\}) \times \Gamma \rightarrow \text{f.s.s. } Q \times \Gamma^*$$

A transition

- A transition:

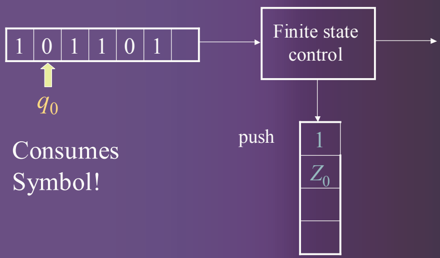
$$\delta(q_0, 1, Z_0)$$



A transition

- A Transition: Push!

  $\delta(q_0, 1, Z_0) = (q_0, 1Z_0)$

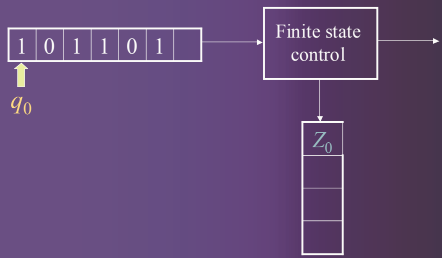


Consumes Symbol!

It is non-deterministic!

- A transition:

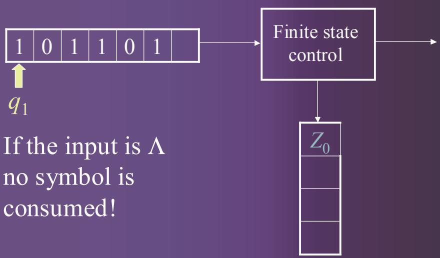
  $\delta(q_0, \Lambda, Z_0)$



It is non-deterministic!

- A Λ-transition:

  $\delta(q_0, \Lambda, Z_0) = (q_1, Z_0)$



If the input is Λ no symbol is consumed!

Transition table $L = \{xx^r \mid x \in \{a, b\}^*\}$

Id	State	Input	Stack symbol	Move(s)
1	q_0	0	Z_0	$(q_0, 0Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0)$
3	q_0	0	0	$(q_0, 00)$
4	q_0	1	0	$(q_0, 10)$
5	q_0	0	1	$(q_0, 01)$
6	q_0	1	1	$(q_0, 11)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Other combinations				non

Transition table $L = \{xx^r \mid x \in \{a, b\}^*\}$

Push (rows 1–6)

Id	State	Input	Stack symbol	Move(s)
1	q_0	0	Z_0	$(q_0, 0Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0)$
3	q_0	0	0	$(q_0, 00)$
4	q_0	1	0	$(q_0, 10)$
5	q_0	0	1	$(q_0, 01)$
6	q_0	1	1	$(q_0, 11)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Other combinations				non

Transition table $L = \{xx^r \mid x \in \{a, b\}^*\}$

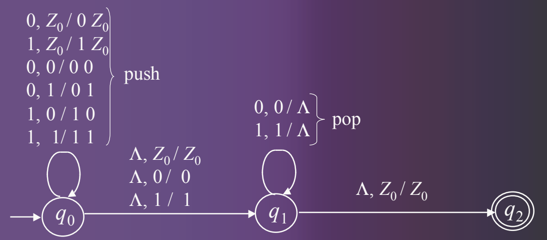
Middle (rows 7–9)

Id	State	Input	Stack symbol	Move(s)
1	q_0	0	Z_0	$(q_0, 0Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0)$
3	q_0	0	0	$(q_0, 00)$
4	q_0	1	0	$(q_0, 10)$
5	q_0	0	1	$(q_0, 01)$
6	q_0	1	1	$(q_0, 11)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Other combinations				non

Transition table $L = \{xx^r \mid x \in \{a, b\}^*\}$

	Id	State	Input	Stack symbol	Move(s)
	1	q_0	0	Z_0	$(q_0, 0Z_0)$
	2	q_0	1	Z_0	$(q_0, 1Z_0)$
	3	q_0	0	0	$(q_0, 00)$
	4	q_0	1	0	$(q_0, 10)$
	5	q_0	0	1	$(q_0, 01)$
	6	q_0	1	1	$(q_0, 11)$
	7	q_0	Λ	Z_0	(q_1, Z_0)
	8	q_0	Λ	0	$(q_1, 0)$
	9	q_0	Λ	1	$(q_1, 1)$
Pop	10	q_1	0	0	(q_1, Λ)
	11	q_1	1	1	(q_1, Λ)
	12	q_1	Λ	Z_0	(q_2, Z_0)
	Other combinations				non

A Graphical notation for PDA

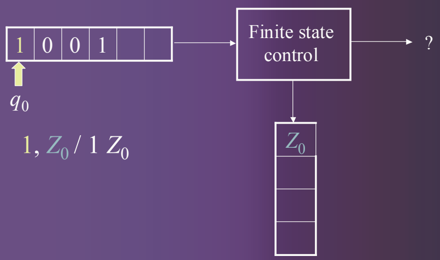


- A label on an arc of form a, X/α from p to q means that on the input a, the top of the stack X is substituted by α:

$$\delta(p, a, X) = (q, \alpha)$$

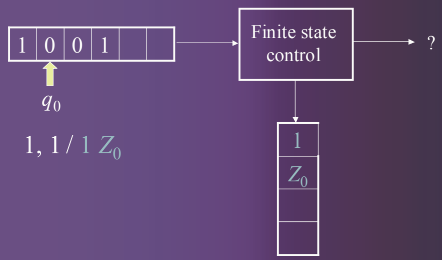
A machine for accepting *Pal*

$\delta(q_0, 1, Z_0) = (q_0, 1Z_0)$



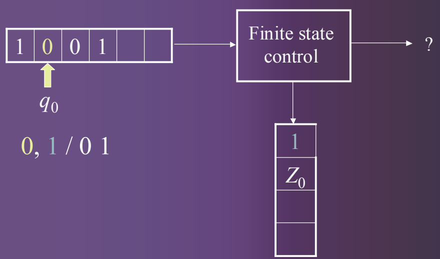
A machine for accepting *Pal*

$\delta(q_0, 1, Z_0) = (q_0, 1Z_0)$



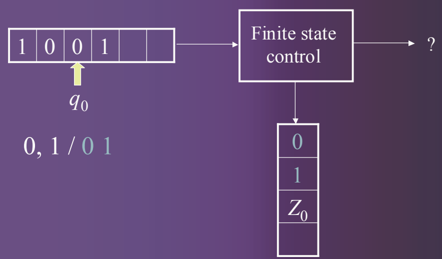
A machine for accepting *Pal*

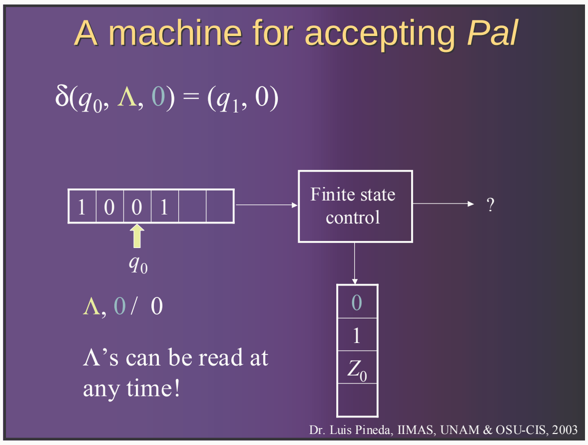
$\delta(q_0, 0, 1) = (q_0, 01)$

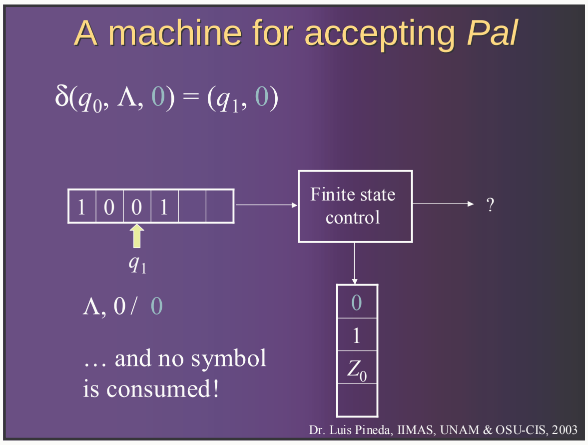


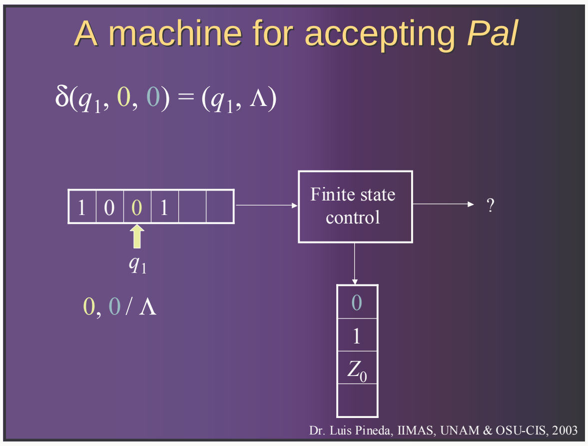
A machine for accepting *Pal*

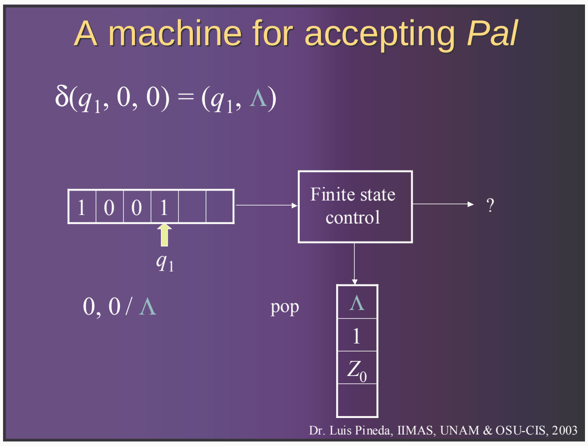
$\delta(q_0, 0, 1) = (q_0, 01)$

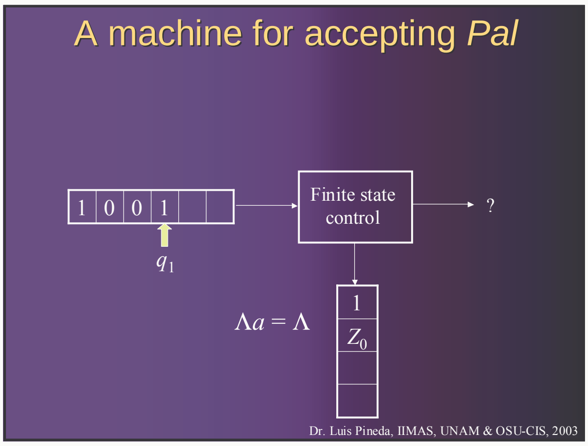


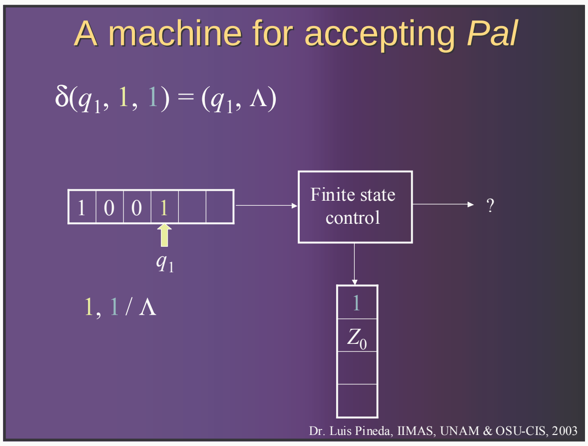

A machine for accepting Pal
δ(q0, Λ, 0) = (q1, 0)
1 0 0 1
Finite state control
?
q0
0
1
Z0
Λ, 0 / 0
Λ's can be read at any time!
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


A machine for accepting Pal
δ(q0, Λ, 0) = (q1, 0)
1 0 0 1
Finite state control
?
q1
0
1
Z0
Λ, 0 / 0
… and no symbol is consumed!
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


A machine for accepting Pal
δ(q1, 0, 0) = (q1, Λ)
1 0 0 1
Finite state control
?
q1
0
1
Z0
0, 0 / Λ
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

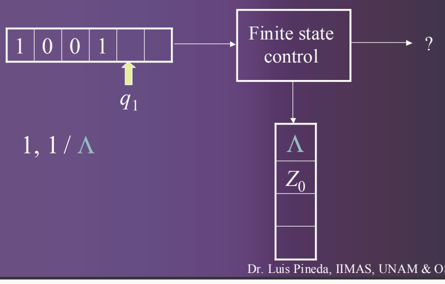

A machine for accepting Pal
δ(q1, 0, 0) = (q1, Λ)
1 0 0 1
Finite state control
?
q1
0, 0 / Λ
pop
Λ
1
Z0
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003


A machine for accepting Pal
1 0 0 1
Finite state control
?
q1
1
Z0
Λa = Λ
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

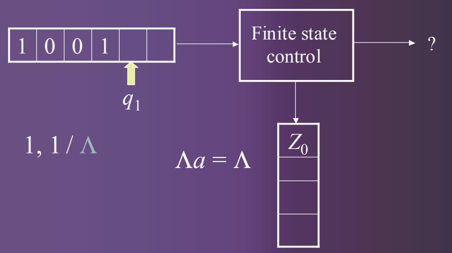

A machine for accepting Pal
δ(q1, 1, 1) = (q1, Λ)
1 0 0 1
Finite state control
?
q1
1, 1 / Λ
1
Z0
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

A machine for accepting *Pal*

$\delta(q_1, 1, 1) = (q_1, \Lambda)$

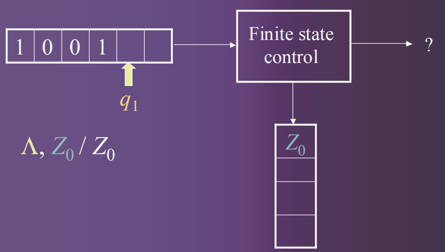


A machine for accepting *Pal*



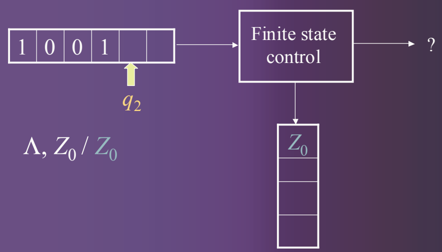
A machine for accepting *Pal*

$\delta(q_1, \Lambda, Z_0) = (q_2, Z_0)$



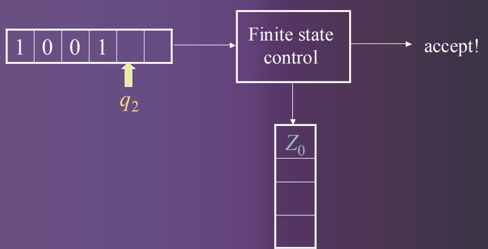
A machine for accepting *Pal*

$\delta(q_1, \Lambda, Z_0) = (q_2, Z_0)$



A machine for accepting *Pal*

- The final state q_2 is accepting
- The whole of the string has been read!
- Z_0 is on the top of the stack!



Configurations of a PDA

- A configuration of a PDA

  $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$

  is a triple

  (q, x, α)

  where
  - $q \in Q$ is the current state
  - $x \in \Sigma^*$ is the part of the string remaining unread
  - $\alpha \in \Gamma^*$ the current content of the stack
- A move from one configuration to the next:

  $(p, x, \alpha) \Rightarrow_M (q, y, \beta)$
- A sequence of cero or more moves:

  $(p, x, \alpha) \Rightarrow^*_M (q, y, \beta)$

Acceptance by a PDA

- Acceptance by final state:
  - If $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ is a PDA. Then, $L(M)$, the language accepted by M by *final state,* is:

    $L(M) = \{w \mid (q_0, w, Z_0) \Rightarrow^*_M (q, \Lambda, \alpha)\}$

    for some $\alpha \in \Gamma^*$ and some $q \in A$. The stack may or may not be empty when w is accepted, because α may or may not be Λ.
- A Language $L \subseteq \Sigma^*$ is said to be accepted by M if L is precisely the set of strings accepted by M, and we write $L = L(M)$

Acceptance by a PDA

- Acceptance by empty stack:
  - If $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ is a PDA. Then, $N(M)$, the language accepted by M by *empty stack,* is:

    $N(M) = \{w \mid (q_0, w, Z_0) \Rightarrow^*_M (q, \Lambda, \Lambda)\}$

    Here, the final state is irrelevant!
- If $L \subseteq \Sigma^*$ is accepted by a PDA M_F by *final state*, there is a PDA M_N that accepts L by empty stack
- For a given PDA M, the language it accepts by final state is usually different from the language it accepts by empty stack!

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

A machine for accepting *Pal* (by final state)

- The language:

  $Pal = \{x \mid x = x^r \in \{a, b\}^*\}$
- The grammar:

  $G_{pal} = (\{P\}, \{0, 1\}, P, P \rightarrow 0P0 \mid 1P1 \mid 1 \mid 0 \mid \Lambda\}$
- Define M_{pal}:

  $M = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, q_0, Z_0, \{q_2\}, \delta)$

  Guesses computing the first part of the input string:
  - Still in there: stay in state, consume symbol and push
  - The first half of an even-length pal. is reached: change state, with a Λ-transition
  - The middle symbol of an odd-length pal is reached: change the state, consume symbol but do not push it into the stack!

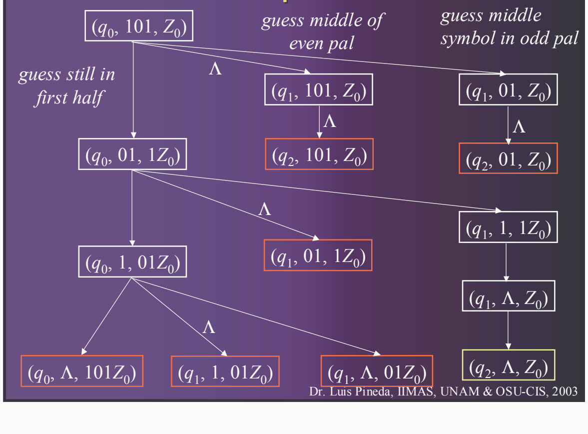
Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Transition table

Id	State	Input	Stack symbol	Move(s)
1	q_0	0	Z_0	$(q_0, 0Z_0), (q_1, Z_0)$
2	q_0	1	Z_0	$(q_0, 1Z_0), (q_1, Z_0)$
3	q_0	0	0	$(q_0, 00), (q_1, 0)$
4	q_0	1	0	$(q_0, 10), (q_1, 0)$
5	q_0	0	1	$(q_0, 01), (q_1, 1)$
6	q_0	1	1	$(q_0, 11), (q_1, 1)$
7	q_0	Λ	Z_0	(q_1, Z_0)
8	q_0	Λ	0	$(q_1, 0)$
9	q_0	Λ	1	$(q_1, 1)$
10	q_1	0	0	(q_1, Λ)
11	q_1	1	1	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
Other combinations				non

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

Accept 101



Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003