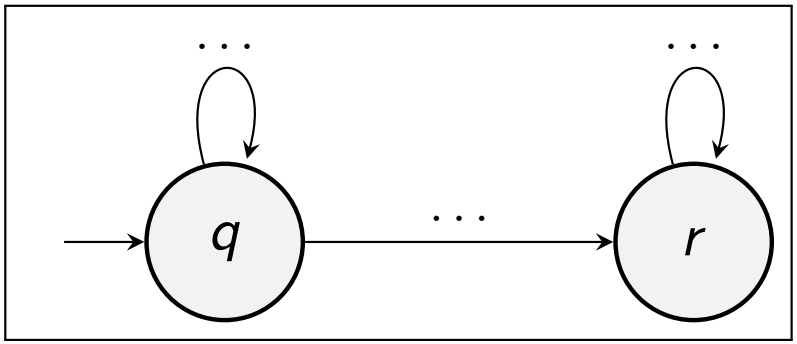
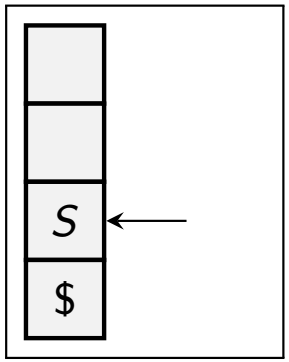
COS210 - Theoretical Computer Science
Pushdown Automata (Part 2)

Deterministic vs Non-Deterministic Pushdown Automata

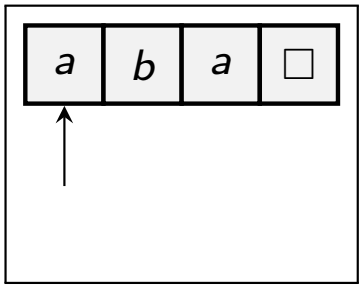
state control



stack



tape



Deterministic PDA:

- for each configuration there is a **unique** transition instruction, e.g. $qaS \to qRSS$
- for each input string there is a **unique** run over the string

Non-deterministic PDA:

- for each configuration there may exist **multiple** transition instructions, e.g. $qaS \to qRSS$ $qaS \to rR\epsilon$
- for each input string there may exist **multiple** runs over the string

Non-deterministic PDAs can describe **more languages** than deterministic PDAs

Non-Deterministic Pushdown Automata – Example

Construct a NPDA M that accepts the language

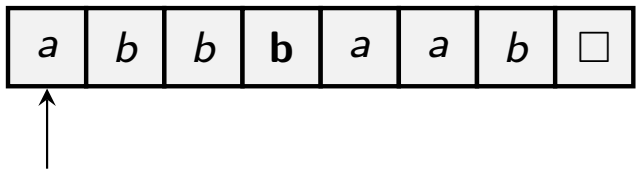
$$L = \{v\mathbf{b}w : v, w \in \Sigma^* \text{ and } |v| = |w|\} \text{ where } \Sigma = \{a, b\}$$

Strings where the symbol in the **middle** is equal to **b**

Examples: **b**, a**b**a, ba**b**aa, bbb**b**bab

A PDA can only scan the input string once, from left to right

- the automaton will **not know** when the symbol in the middle is reached
- but a non-deterministic PDA can **guess** when the symbol in the middle is reached
- guessing via **non-deterministic branches**: either middle reached, or not reached

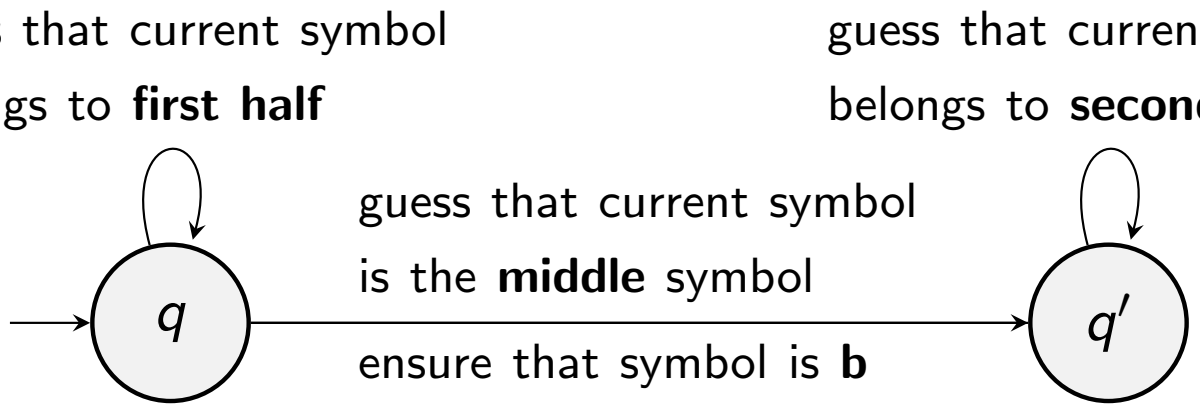


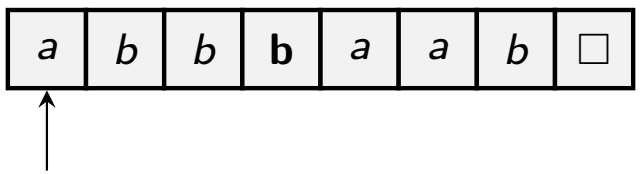
- each branch creates another run over the input string, if at least one run over the string is accepting then PDA will accept the string

Non-Deterministic Pushdown Automata – Example

Construction idea:

- use two states q and q' that shall represent scanning the first and the second half of the input string, respectively
- switching from q to q' corresponds to guess that middle symbol has been scanned



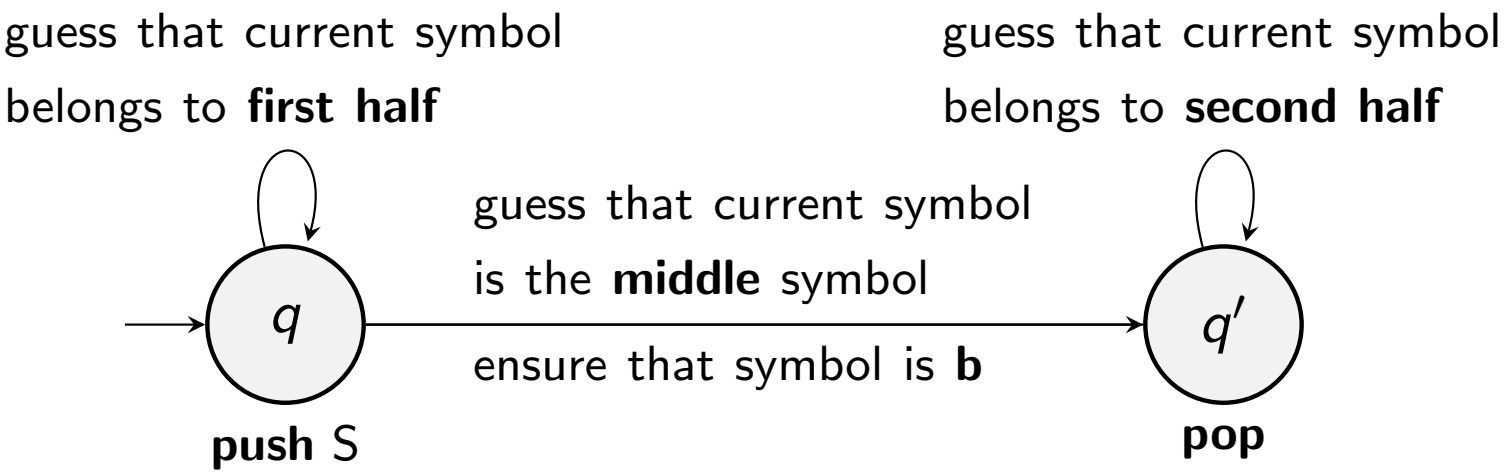


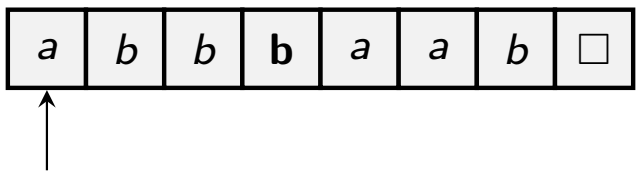
We still need to validate in the end that the middle was correctly guessed

Non-Deterministic Pushdown Automata – Example

Construction idea (continued):

- for each symbol that is guessed to belong to the **first half**, **push** an S onto stack
- for each symbol that is guessed to belong to the **second half**, **pop** a symbol from the stack





If the guess was correct, then the stack only contain \$ when □ is read

Non-Deterministic Pushdown Automata – Example

$M = (\Sigma, \Gamma, Q, \delta, q)$ where $\Sigma = \{a, b\}$, $\Gamma = \{\$, S\}$, $Q = \{q, q'\}$, q is the initial state, and δ is defined by the following instructions:

1)	$qa\$ \to qR\$S$	**guess first half, push S**
2)	$qaS \to qRSS$	**guess first half, push S**
3)	$qb\$ \to q'R\$$	**guess middle, switch to q'**
4)	$qb\$ \to qR\$S$	**guess first half, push S**
5)	$qbS \to q'RS$	**guess middle, switch to q'**
6)	$qbS \to qRSS$	**guess first half, push S**
7)	$q\square\$ \to qN\$$	**empty input string, loop forever**
8)	$q\square S \to qNS$	**string ends while first half is guessed, loop forever**
9)	$q'a\$ \to q'N\epsilon$	**middle was guessed incorrectly, terminate reject**
10)	$q'aS \to q'R\epsilon$	**guess second half, pop**
11)	$q'b\$ \to q'N\epsilon$	**middle was guessed incorrectly, terminate reject**
12)	$q'bS \to q'R\epsilon$	**guess second half, pop**
13)	$q'\square\$ \to q'N\epsilon$	**middle was guessed correctly, terminate accept**
14)	$q'\square S \to q'NS$	**middle was guessed incorrectly, loop forever**

Non-Deterministic Pushdown Automata – Example 2

Construct a NPDA M that accepts the language

$$L = \{v\mathbf{b}v^R : v \in \Sigma^*\} \text{ where } \Sigma = \{a, b\}$$

(if $v = v_1 \dots v_n$, then $v^R = v_n \dots v_1$)

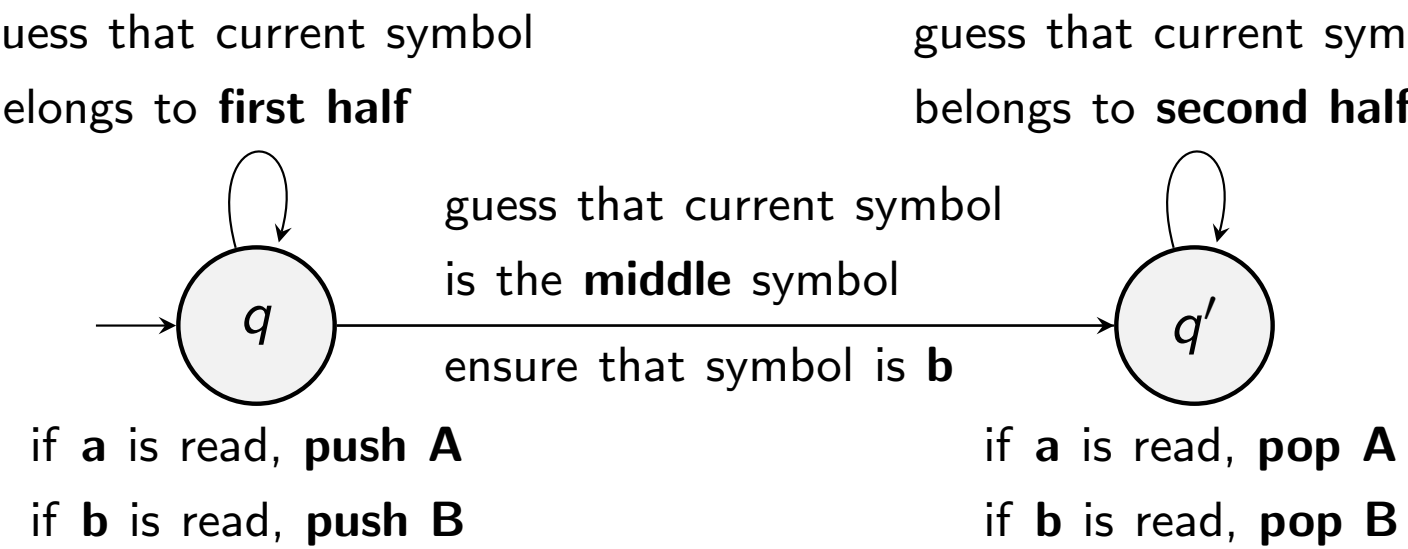
Strings where the symbol in the **middle** is equal to **b**
and the **second half** of the string is the **reverse of** the **first half**.

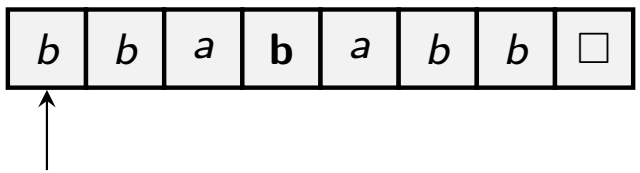
Examples: **b**, a**b**a, ab**b**ba, bba**b**abb

Non-Deterministic Pushdown Automata – Example 2

Construction idea:

- use the stack alphabet $\Gamma = \{\$, A, B\}$





Non-Deterministic Pushdown Automata – Example 2

$M = (\Sigma, \Gamma, Q, \delta, q)$ where $\Sigma = \{a, b\}$, $\Gamma = \{\$, A, B\}$, $Q = \{q, q'\}$, δ:

1) $qa\$ \to qR\$A$ **guess first half, read a, push A**
2) $qaA \to qRAA$ **guess first half, read a, push A**
3) $qaB \to qRBA$ **guess first half, read a, push A**

4) $qb\$ \to qR\$B$ **guess first half, read b, push B**
5) $qbA \to qRAB$ **guess first half, read b, push B**
6) $qbB \to qRBB$ **guess first half, read b, push B**

7) $q\square\$ \to qN\$$ **empty input string, loop forever**
8) $q\square A \to qNA$ **string ends while first half is guessed, loop forever**
9) $q\square B \to qNB$ **string ends while first half is guessed, loop forever**

10) $qb\$ \to q'R\$$ **guess middle, switch to q'**
11) $qbA \to q'RA$ **guess middle, switch to q'**
12) $qbB \to q'RB$ **guess middle, switch to q'**

Non-Deterministic Pushdown Automata – Example 2

$M = (\Sigma, \Gamma, Q, \delta, q)$ where $\Sigma = \{a, b\}$, $\Gamma = \{\$, A, B\}$, $Q = \{q, q'\}$, δ:

13) $q'a\$ \to q'N\epsilon$ **middle was guessed incorrectly, terminate reject**

14) $q'aA \to q'R\epsilon$ **read** a**, pop** A

15) $q'aB \to q'NB$ **second half is not reverse of first half, loop forever**

16) $q'b\$ \to q'N\epsilon$ **middle was guessed incorrectly, terminate reject**

17) $q'bA \to q'NA$ **second half is not reverse of first half, loop forever**

18) $q'bB \to q'R\epsilon$ **read** b**, pop** B

19) $q'\square\$ \to q'N\epsilon$ **correct string, terminate accept**

20) $q'\square A \to q'NA$ **middle was guessed incorrectly, loop forever**

21) $q'\square B \to q'NB$ **middle was guessed incorrectly, loop forever**