Universidad Carlos III de Madrid
www.uc3m.es

AUTOMATA THEORY AND FORMAL LANGUAGES

UNIT 6: PUSH-DOWN AUTOMATA

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

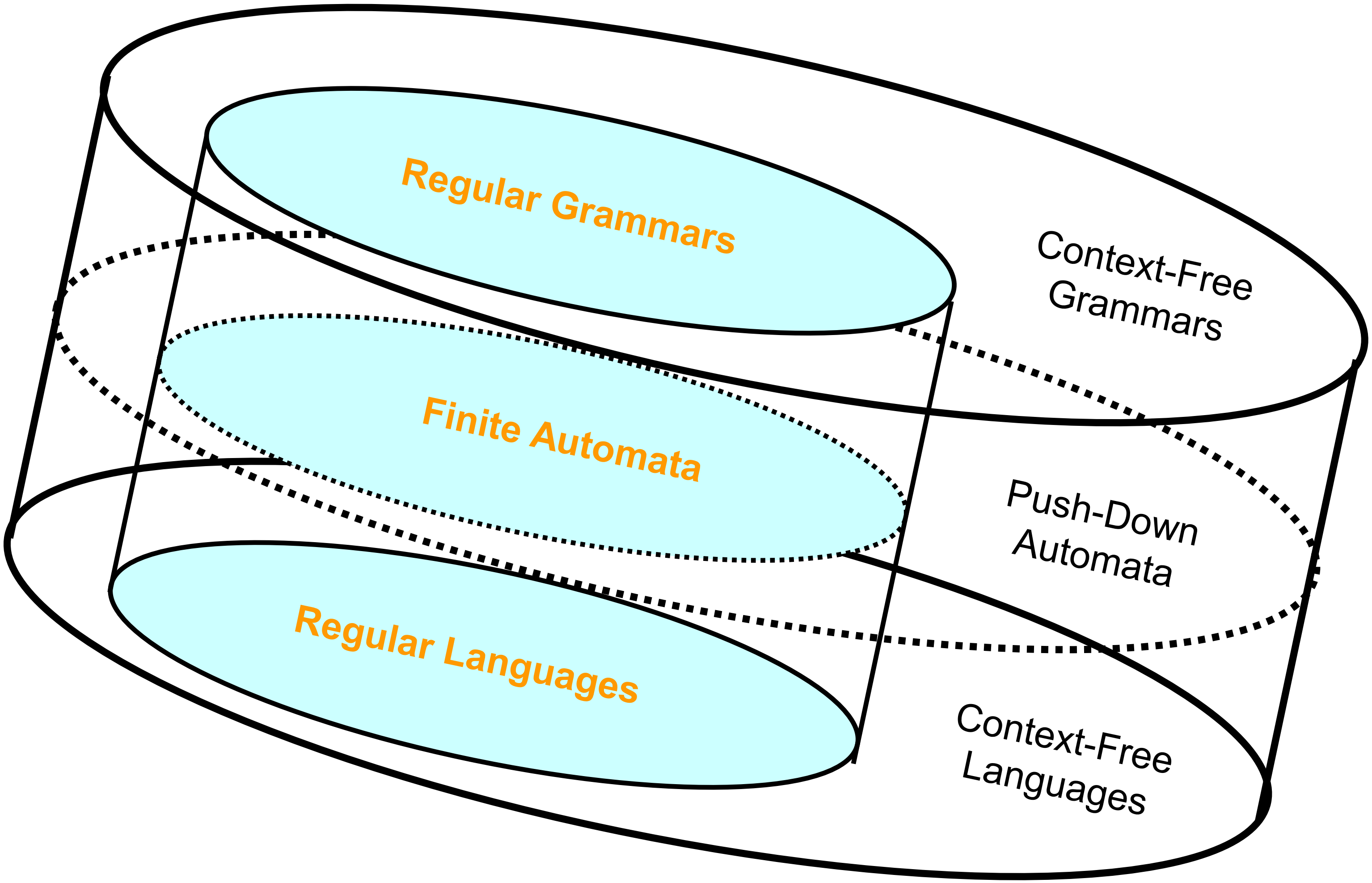
OUTLINE

- Introduction
- Definition of Push-Down Automata
  - Acceptance in final states or when the stack is empty
  - Formal definition
  - Transitions
  - Instantaneous Description, Movement
  - Deterministic Push-Down Automata
  - Language Accepted by a Push-Down Automaton
  - Examples
- Equivalence between PD Automata and Context-Free Languages

OUTLINE

- **Introduction**
- Definition of Push-Down Automata
  - Acceptance in final states or when the stack is empty
  - Formal definition
  - Transitions
  - Instantaneous Description, Movement
  - Deterministic Push-Down Automata
  - Language Accepted by a Push-Down Automaton
  - Examples
- Equivalence between PD Automata and Context-Free Languages

Introduction



David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Introduction

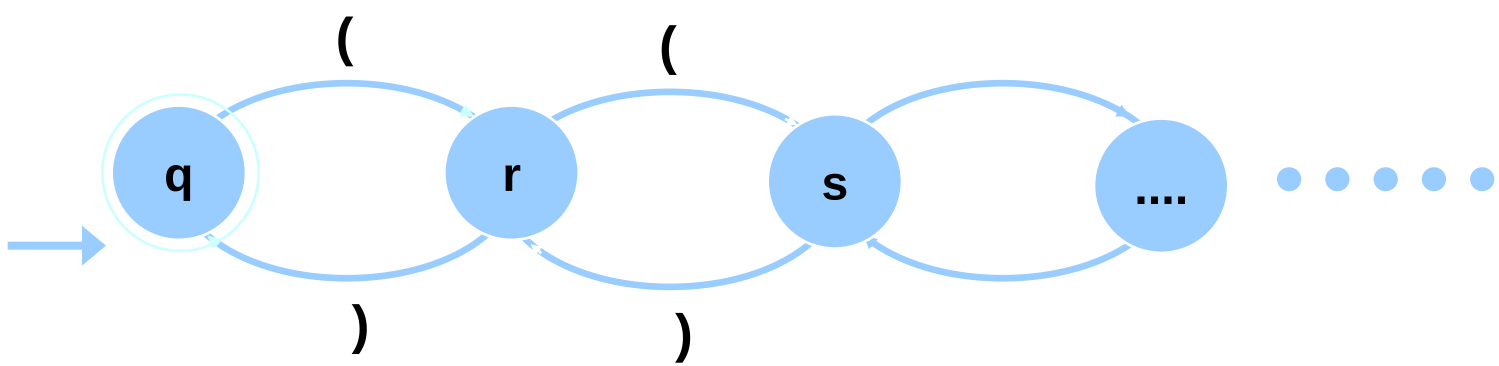
- **Limitations of FA's:**
  - Only repetition sentences can be recognized.
    - E. g. $a^n b^n$, $a^n b^n c^n$
  - It is not possible to determine if a program is correct.
  - It is not possible to determine syntax errors present in natural language.

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

- **Limitations of FA's: Explanation.**

Lack of Memory

Mathematical expressions cannot be recognized,
e.g. "(2x+(2+n/25))", nested paired brackets, language X^nY^n



- Function: Analyze words to know if they belong to Type-2 languages: **Accept or not accept.**

- Same structure that a finite automata adding a stack (auxiliary memory).

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Push-Down Automata and Languages

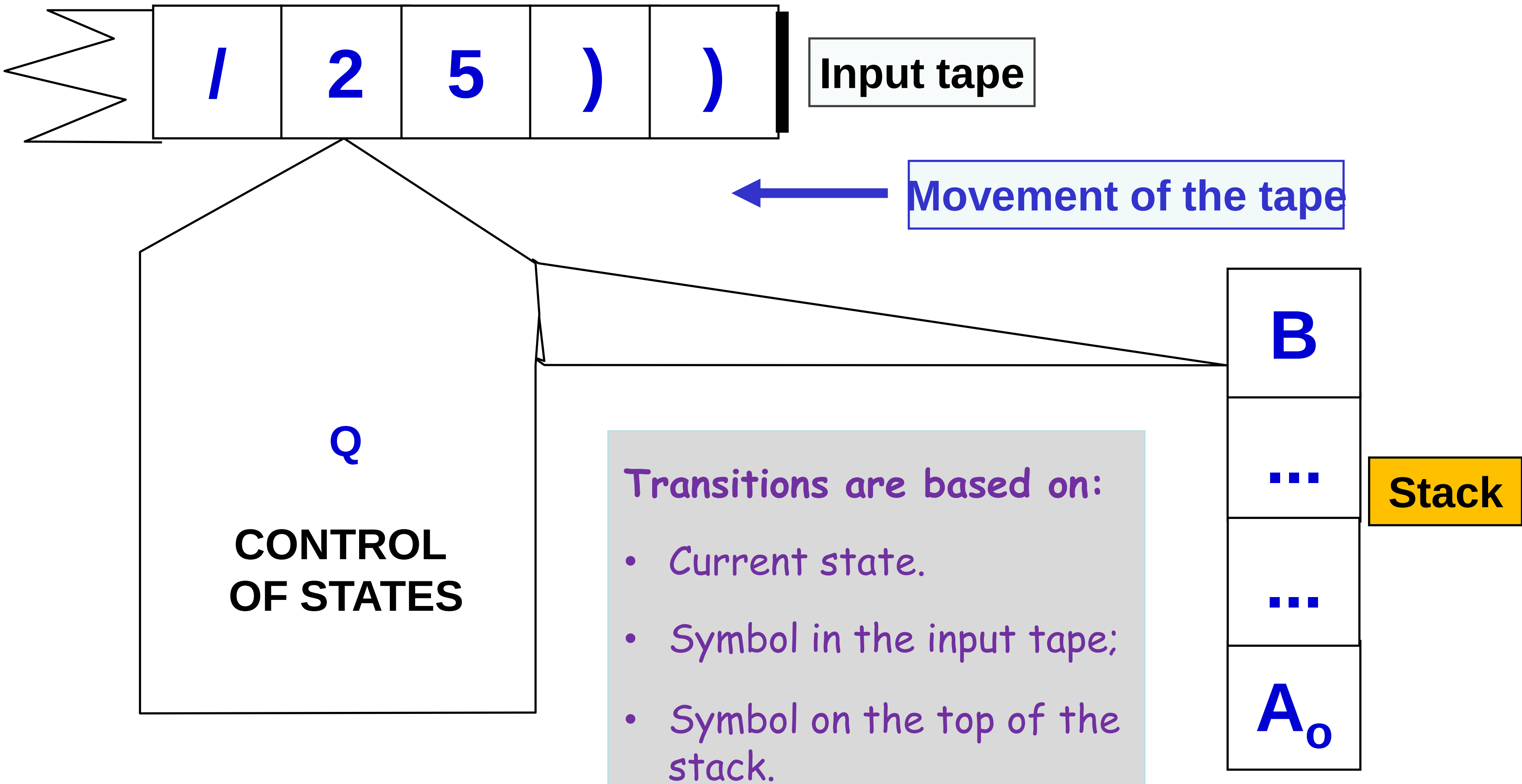
- Theorems:
  - For each context-free grammar G, there is a push-down automaton M that fulfills L(G)=L(M)
  - For each push-down automata M, there is a context-free grammar G that fulfills L(M)=L(G)
  - There are context-free languages that cannot be recognized by any deterministic push-down automaton.

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

OUTLINE

- Introduction
- **Definition of Push-Down Automata**
  - Acceptance in final states
  - Acceptance when the stack is empty
  - Formal definition
  - Transitions
  - Instantaneous Description, Movement
  - Deterministic Push-Down Automata
  - Language Accepted by a Push-Down Automaton
  - Examples
- Equivalence between PD Automata and Context-Free Languages

Definition of Push-Down Automaton

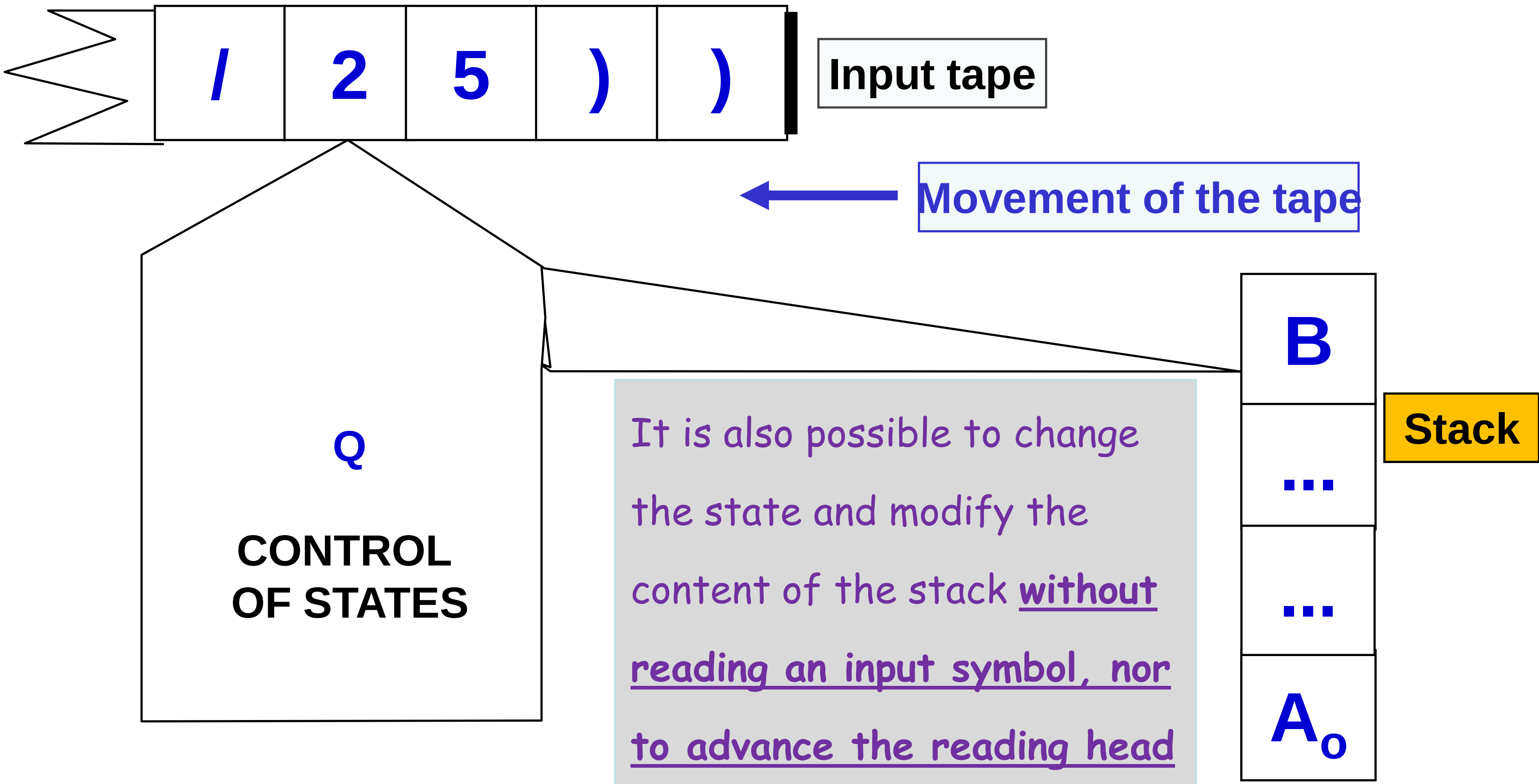


Transitions are based on:

- Current state.
- Symbol in the input tape;
- Symbol on the top of the stack.

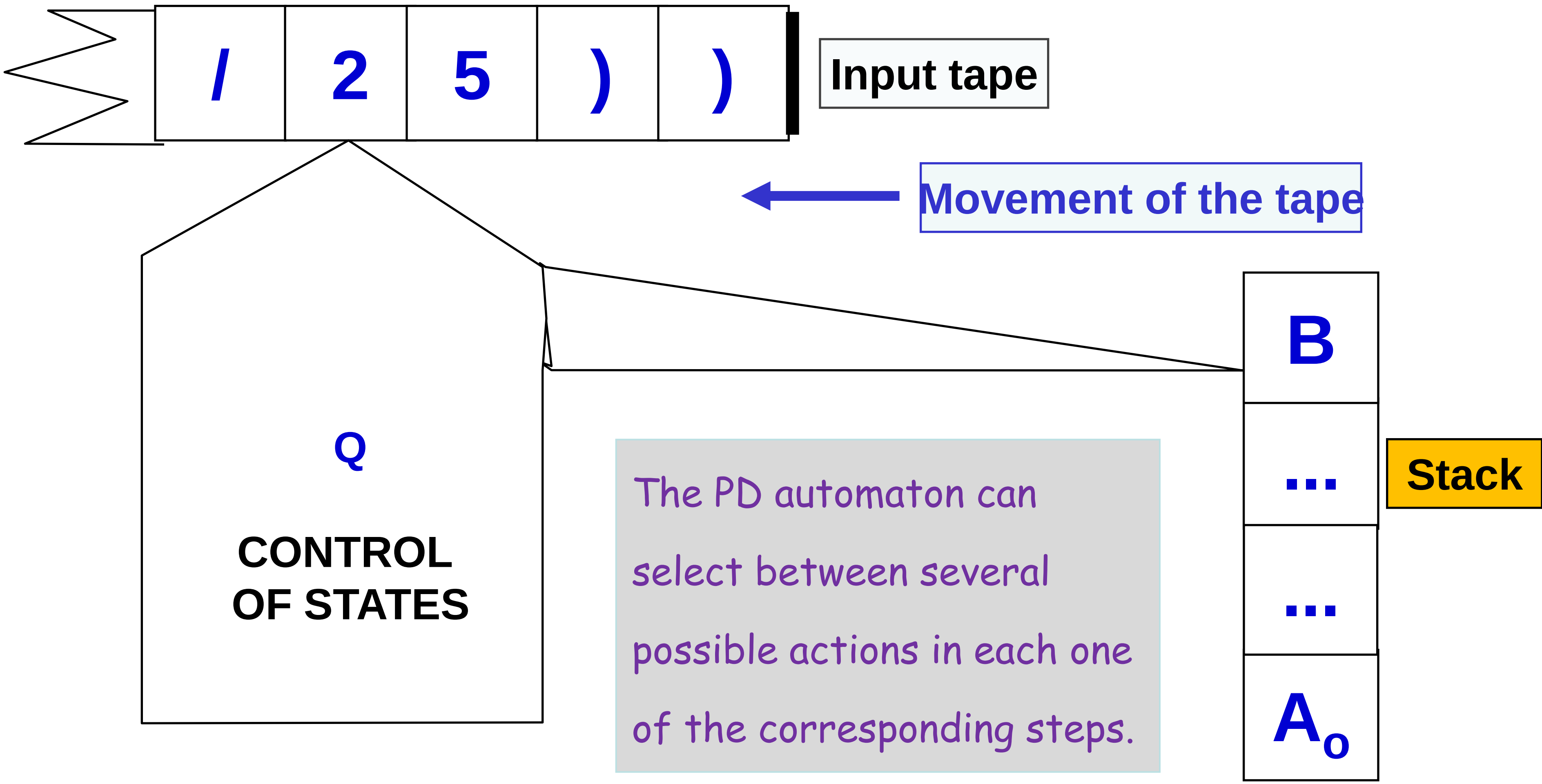
David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Definition of Push-Down Automaton



It is also possible to change the state and modify the content of the stack **<u>without reading an input symbol, nor to advance the reading head of the tape.</u>**

Definition of Push-Down Automaton



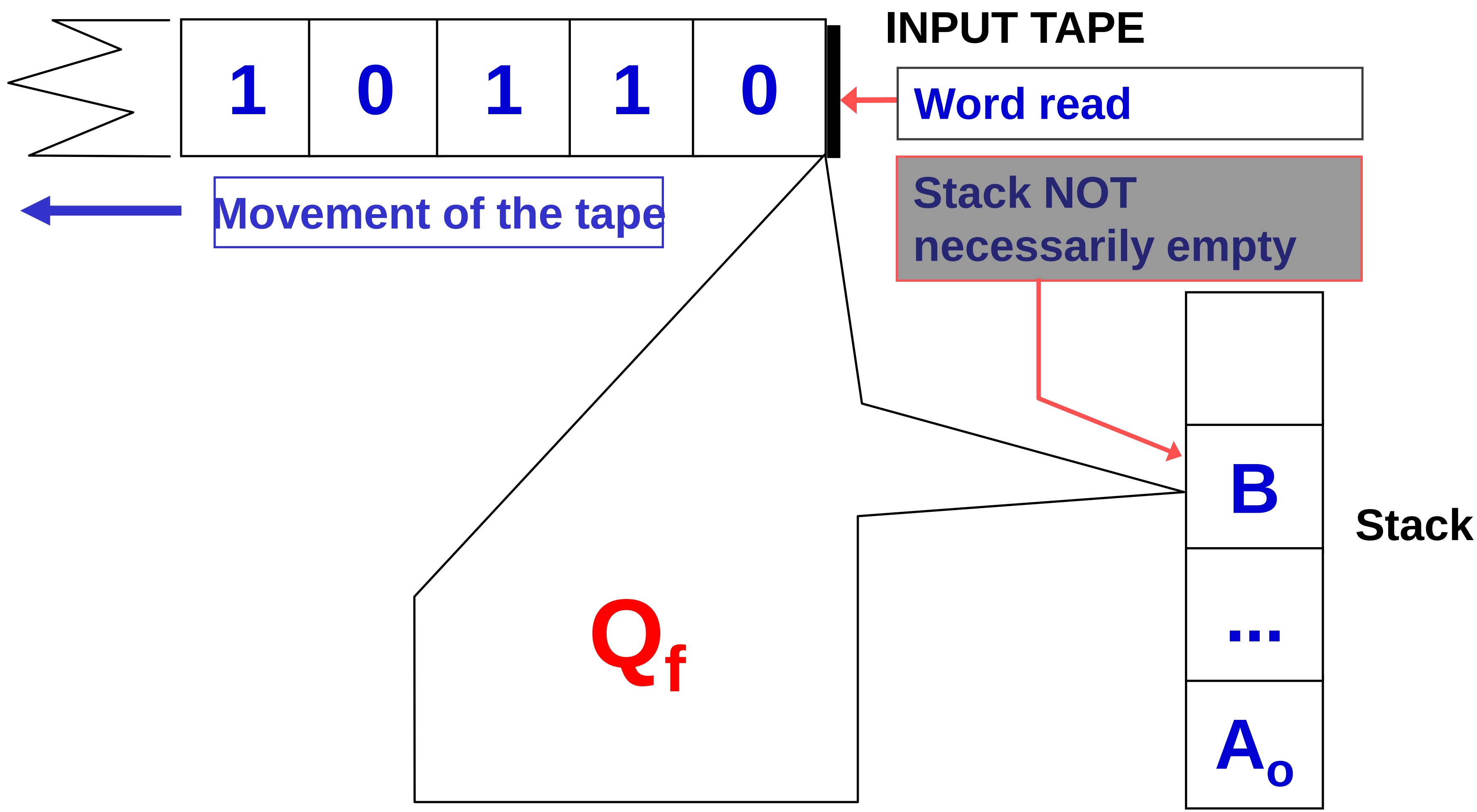
The PD automaton can select between several possible actions in each one of the corresponding steps.

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

OUTLINE

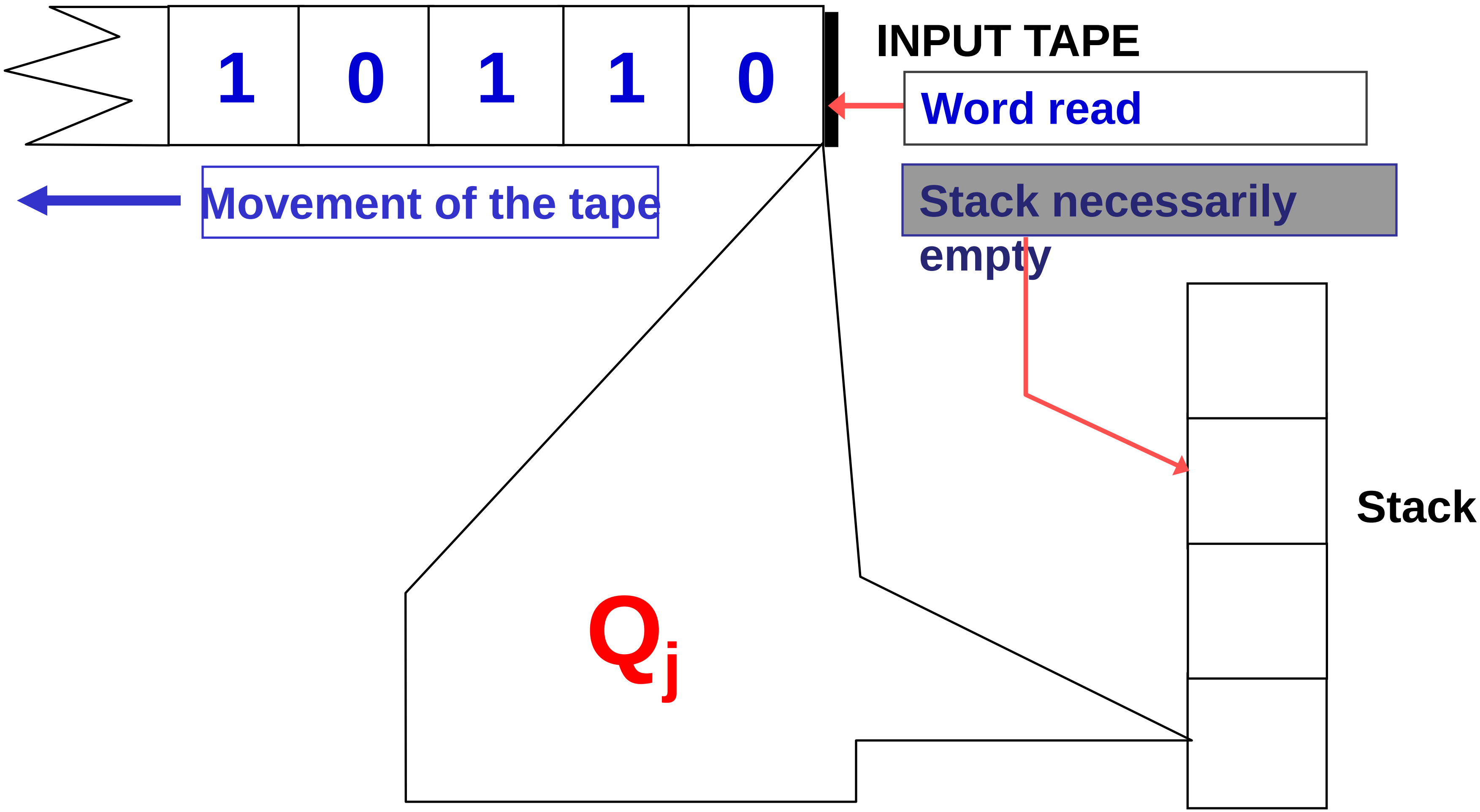
- Introduction
- **Definition of Push-Down Automata**
  - **Acceptance in final states or when the stack is empty**
  - Formal definition
  - Transitions
  - Instantaneous Description, Movement
  - Deterministic Push-Down Automata
  - Language Accepted by a Push-Down Automaton
  - Examples
- Equivalence between PD Automata and Context-Free Languages

Acceptation in final states



David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

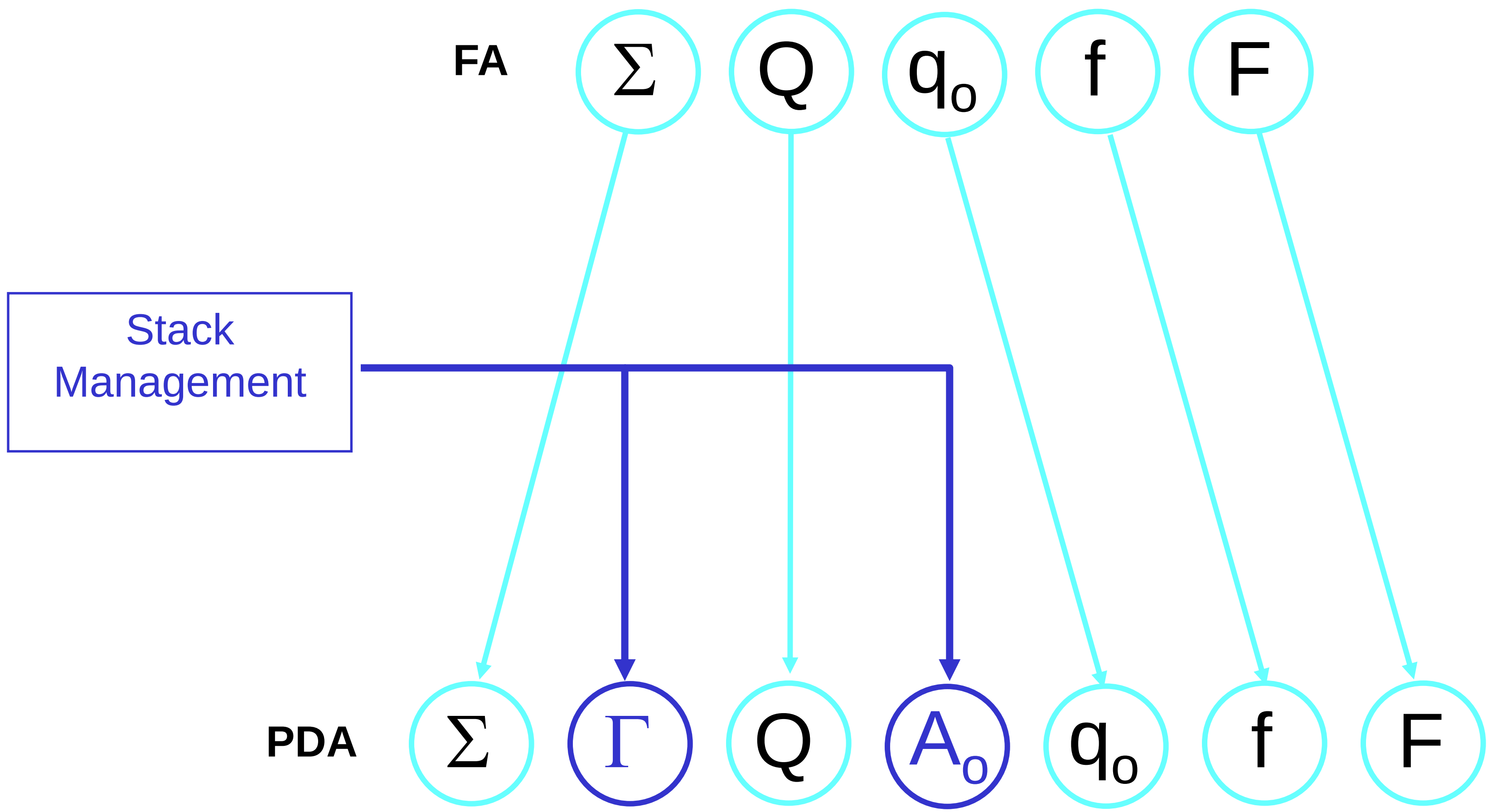
Acceptation when the stack is empty



OUTLINE

- Introduction
- **Definition of Push-Down Automata**
  - Acceptance in final states or when the stack is empty
  - **Formal definition**
  - Transitions
  - Instantaneous Description, Movement
  - Deterministic Push-Down Automata
  - Language Accepted by a Push-Down Automaton
  - Examples
- Equivalence between PD Automata and Context-Free Languages

Formal definition of Push-Down Automaton



Formal definition of Push-Down Automaton

PDA: $(\Sigma, \Gamma, Q, A_o, q_o, f, F)$

- Σ : input alphabet (tape) **Input Words:** x, y, z, ax, ay... $\in \Sigma^*$.
- Γ : stack alphabet **Words in the stack:** X, Y, Z, AX, AY... $\in \Gamma^*$
- **Q : finite set of states** **Q =** {p,q,r,...}
- $A_o \in \Gamma$: **initial symbol in the stack**
- $q_o \in Q$: **initial state of the automaton**
- **f : transition function**
- $F \subset Q$: **set of final states**

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

OUTLINE

- Introduction
- **Definition of Push-Down Automata**
  - Acceptance in final states or when the stack is empty
  - Formal definition
  - **Transitions**
  - Instantaneous Description, Movement
  - Deterministic Push-Down Automata
  - Language Accepted by a Push-Down Automaton
  - Examples
- Equivalence between PD Automata and Context-Free Languages

Push-Down Automaton. Transitions

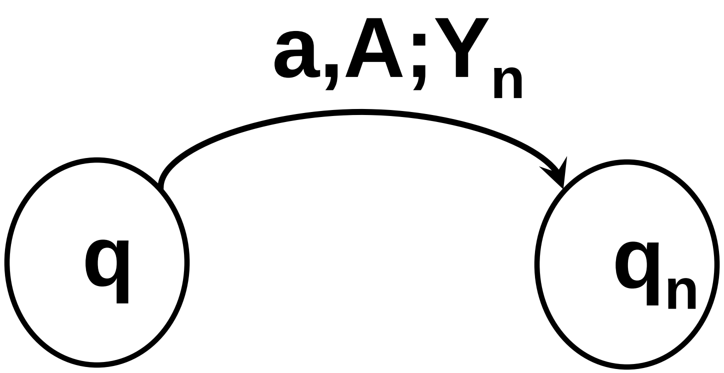
- **Transition function**:

$$f : Q \times (\Sigma \cup \{\lambda\}) \times \mathbf{t} \rightarrow \mathcal{P}(Q \times \mathbf{t}^*)$$

For each state, input symbol in the tape or empty word, and symbol on the top of the stack → the automaton determines the transition to another state and decides the symbols to be written in the stack.

Push-Down Automaton. Transitions

- Transitions in a push-down automaton follow the following sequence:
  - Read an input symbol.
  - Extract a symbol from the stack.
  - Insert a word in the stack.
  - Transit to a new state.



- Definition:
  - $f(q,a,A)=\{(q_1,Z_1),(q_2, Z_2),...,(q_n, Z_n)\}$
  - Another notation: $(q,a,A;q_n,Y_n)$

  where q, qi $\in$ Q, a $\in \Sigma$, A $\in \Gamma$, Zi $\in \Gamma^*$

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Push-Down Automaton. Transitions

Universidad Carlos III de Madrid
www.uc3m.es

$$f : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow P(Q \times \Gamma^*)$$

Transitions that depend on the input

$Q \times \Sigma \times \Gamma$

Transitions that do not depend on the input

$Q \times \lambda \times \Gamma$

Deterministic Push-Down Automata

$Q \times \Gamma^*$

Non-Deterministic Push-Down Automata

$P(Q \times \Gamma^*)$

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Push-Down Automaton. Transitions

Transitions that do not depend on the input

- Given the transition:

$$f(q, \lambda, A) = \{(q_1, Z_1), (q_2, Z_2), \ldots, (q_n, Z_n)\}$$

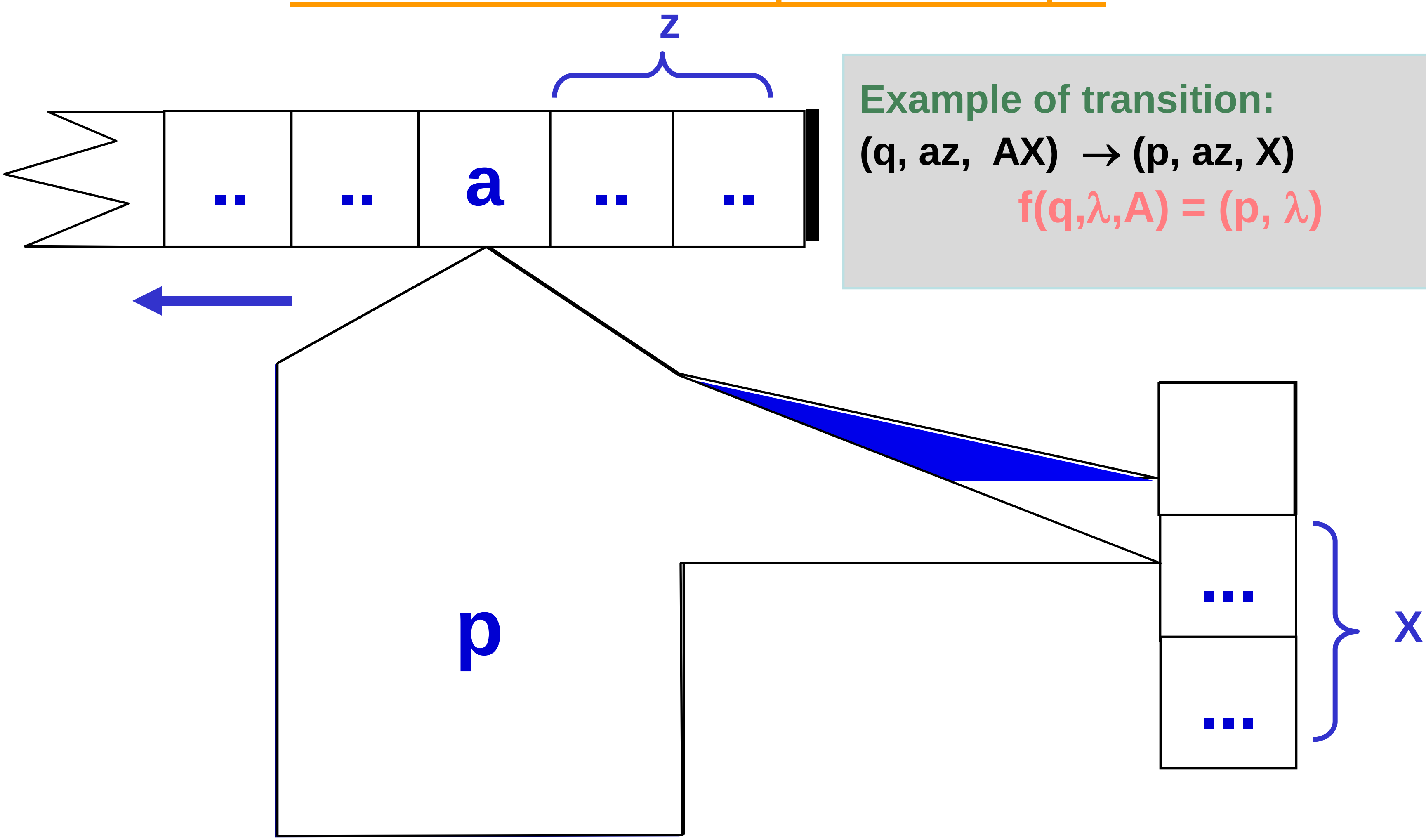
where:

- $q, q_i \in Q$
- $A \in \Gamma$
- $Z_i \in \Gamma^*$

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Push-Down Automaton. Transitions

Transitions that do not depend on the input



Example of transition:

(q, az, AX) → (p, az, X)

$f(q,\lambda,A) = (p, \lambda)$

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Push-Down Automaton. Transitions

Transitions that depend on the input

- **Given the transition:**

$$f(q,a,A) = \{(q_1,Z_1), (q_2,Z_2),\ldots,(q_n,Z_n)\}$$

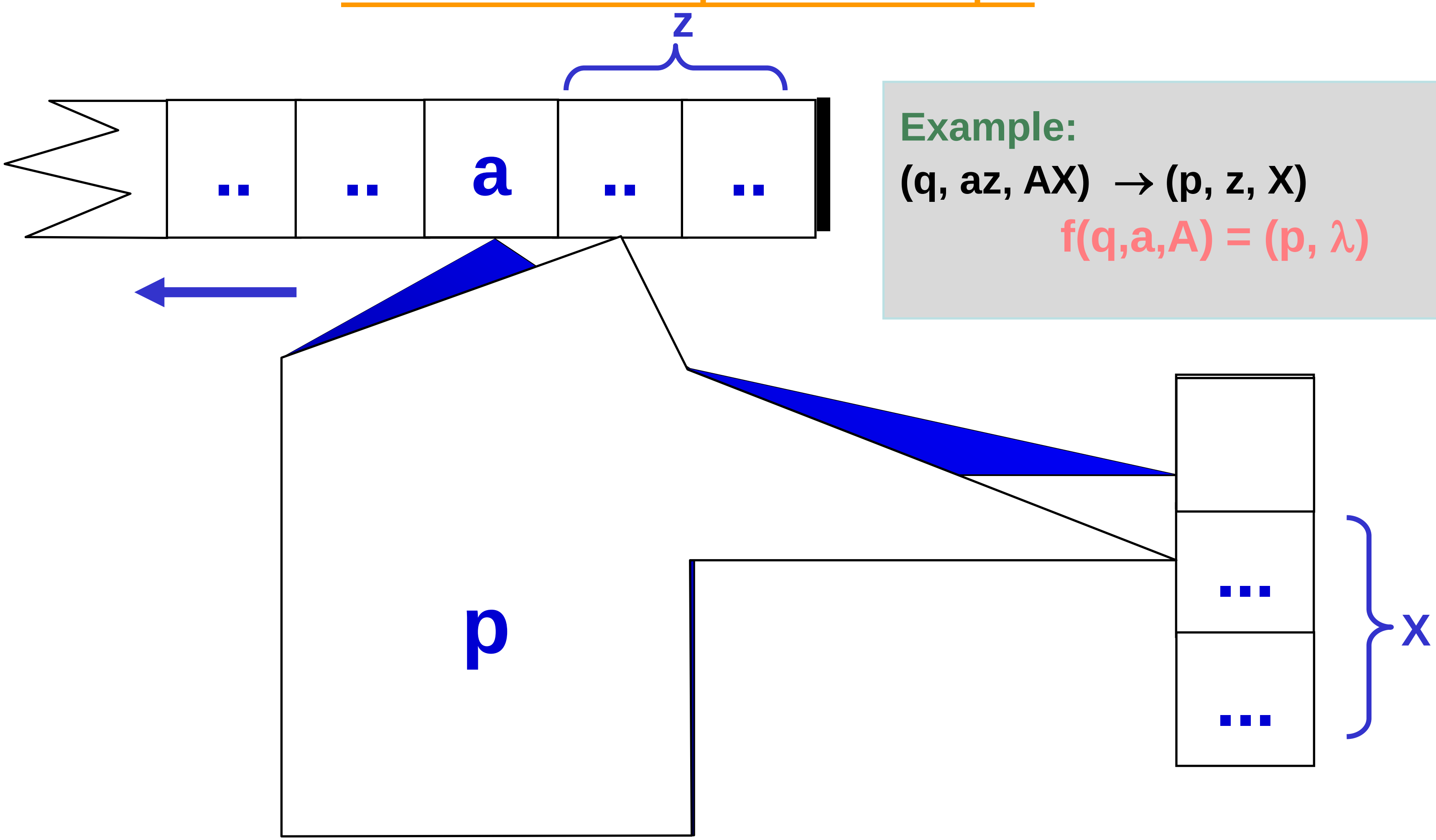
where:

- $q, q_i \in Q$
- $a \in \Sigma$
- $A \in \Gamma$
- $Z_i \in \Gamma^*$

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Push-Down Automaton. Transitions

Transitions that depend on the input



Example:

(q, az, AX) → (p, z, X)

f(q,a,A) = (p, λ)

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

OUTLINE

- Introduction
- **Definition of Push-Down Automata**
  - Acceptance in Final States or when the stack is empty
  - Formal definition
  - Transitions
  - **Instantaneous Description, Movement**
  - Deterministic Push-Down Automata
  - Language Accepted by a PD Automaton
  - Examples
- Equivalence between PD Automata and Context-Free Languages

Instantaneous description

- It is used to easily describe the configuration of a Push-Down automaton in each moment.
  - Group of three (q,x,z)
    where $q \in Q$, $x \in \Sigma^*$, $z \in \Gamma^*$
  - It contains:
    - the current state (q);
    - the part of the input word that is still to be read (x);
    - the symbol on the top of the stack (z).

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Instantaneous description

- **Instantaneous description (q,x,z)** where $q \in Q$, $x \in \Sigma^*$, $z \in \Gamma^*$

- **Movement $(q,ay,AX) \vdash (p,y,YX)$** describes the transition from an instantaneous description to another.

  (p,y,YX) precedes (q,ay,AX) if $(p,Y) \in f(q,a,A)$

- **Succession of movements: $(q,ay,AX) \vdash^* (p,y,YX)$** represents that the second instantaneous description can be reached from the first one.

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

OUTLINE

- Introduction
- **Definition of Push-Down automata**
  - Acceptance in Final States or when the stack is empty
  - Formal definition
  - Transitions
  - Instantaneous Description, Movement
  - **Deterministic Push-Down Automata**
  - Language Accepted by a PD Automaton
  - Examples
- Equivalence between PD Automata and Context-Free Languages

Deterministic Push-Down Automaton

- $(\Sigma,\Gamma,Q,A_0,q_0,f,F)$ is deterministic if verifies:
  - $\forall q \in Q,\ A \in \Gamma,\ |f(q,\lambda,A)| > 0 \Rightarrow f(q,a,A) = \Phi\ \forall a \in \Sigma$
    - If there is a λ-transition, given a state *q* and a stack symbol *A*, then there is not any λ-transition with any other input symbol and state.
  - $\forall q \in Q,\ A \in \Gamma,\ \forall a \in \Sigma \cup \{\lambda\},\ |f(q,a,A)| < 2$
    - There is only one transition given a state and a symbol on the top of the stack: f (q,a,A) = (p,X)
    - If (p, x, y; q, z) and (p, x, y; r, w) are transitions of a deterministic push-down automaton, then:

$$q \equiv r, \quad z = w$$

OUTLINE

- Introduction
- **Definition of Push-Down Automata**
  - Acceptance in final states or when the stack is empty
  - Formal definition
  - Transitions
  - Instantaneous Description, Movement
  - Deterministic Push-Down Automata
  - **Language Accepted by a Push-Down Automaton**
  - Examples
- Equivalence between PD Automata and Context-Free Languages

Language accepted by a Push-Down Automaton

- **Given that the stack is empty**:
  - $LE_{PDA}=\{x \mid (q_0,x,A_0) \vdash_* (p,\lambda,\lambda),\ p \in Q,\ x \in \Sigma^*\}$
  - When the acceptance is when the stack is empty, the set of final states is irrelevant, and usually it is empty (F=Ø).

- **Given an acceptance state**:
  - $LF_{PDA}=\{x \mid (q_0,x,A_0) \vdash_* (p,\lambda,X),\ p \in F,\ x \in \Sigma^*, X \in \Gamma^*\}$

OUTLINE

- Introduction
- **Definition of Push-Down Automata**
  - Acceptance in final states or when the stack is empty
  - Formal definition
  - Transitions
  - Instantaneous Description, Movement
  - Deterministic Push-Down Automata
  - Language Accepted by a Push-Down Automaton
  - **Examples**
- Equivalence between PD Automata and Context-Free Languages

Definition of Push-Down Automaton. Example

LANGUAGE: set of sentences

var ::= num;

if cond
then
Assignation or IF

if cond
then
Assignation or IF
else
Assignation or IF

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Definition of Push-Down Automaton. Example

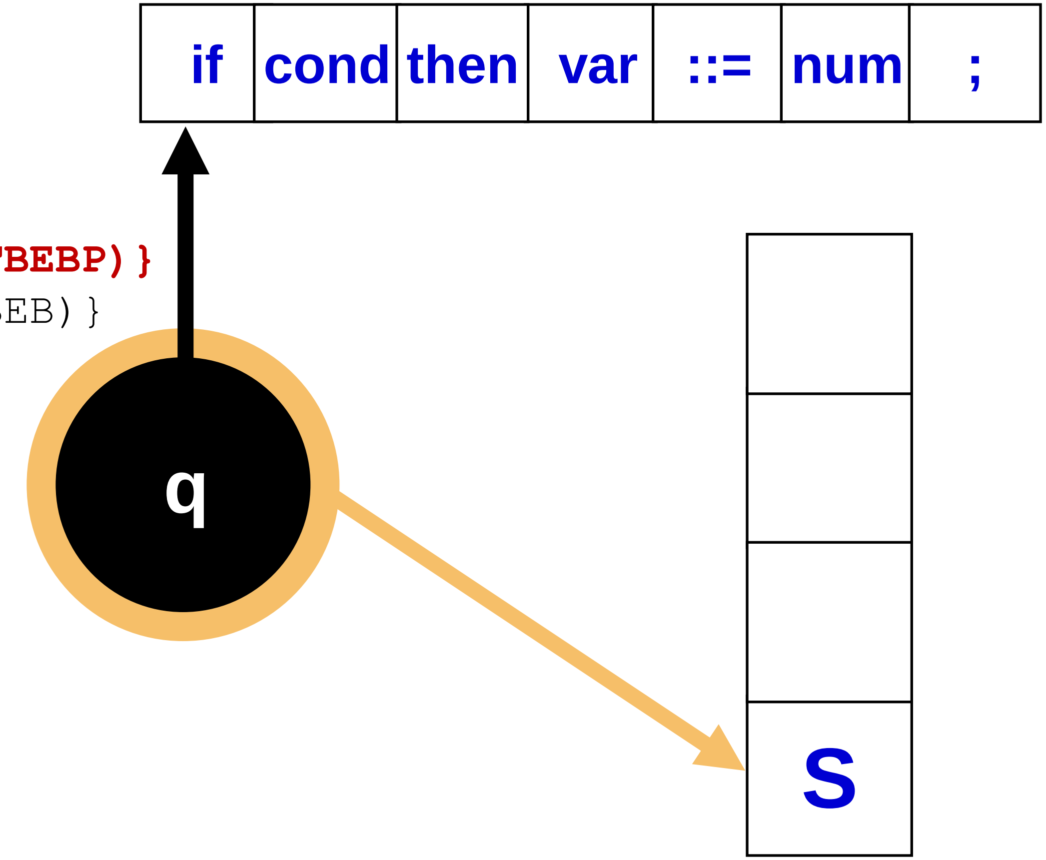
AP= ({if, then, else, ::=, var, num, cond, ;}, {S, B, C, F, N, P, T, E}, {q}, q, S, f, ϕ)

f(q, var, S) = {(q, FNP)}
f(q, if, S) = {(q, CTBP), (q, CTBEBP)}
f(q, if, B) = {(q, CTB), (q, CTBEB)}
f(q, var, B) = {(q, FN)}
f(q, cond, C) = {(q, λ)}
f(q, ::=, F) = {(q, λ)}
f(q, num, N) = {(q, λ)}
f(q, ;, P) = {(q, λ)}
f(q, then, T) = {(q, λ)}
f(q, else, E) = {(q, λ)}

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Definition of Push-Down Automaton. Example

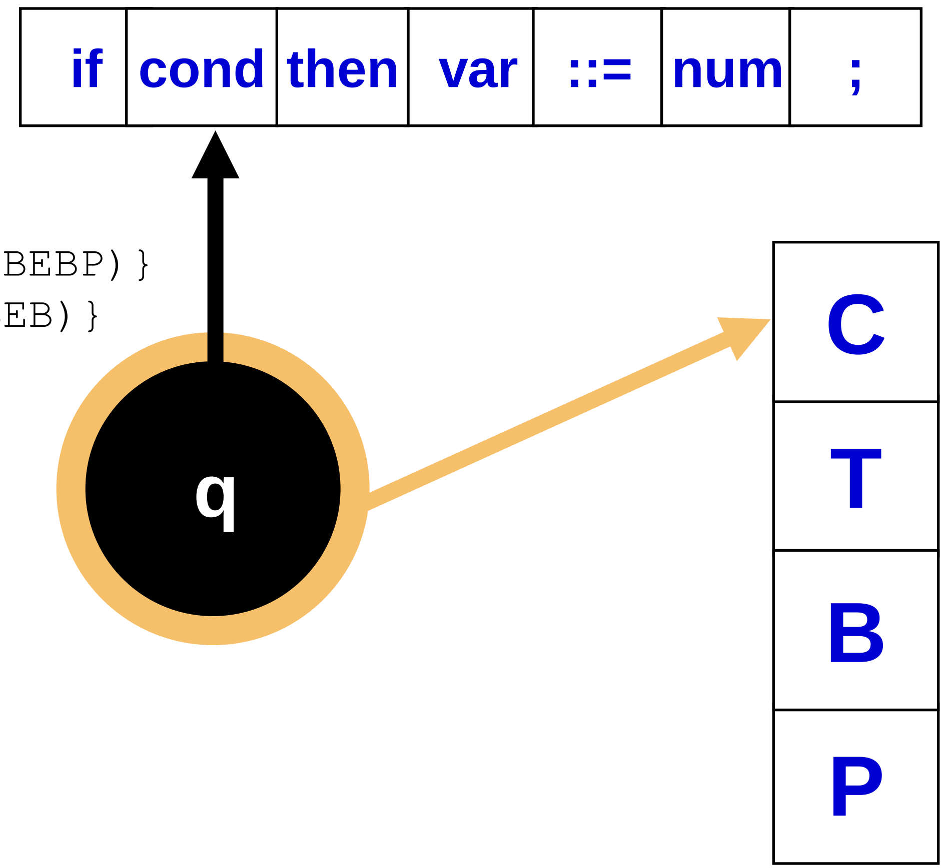
```
f(q,var,S)   = {(q,FNP)}
f(q,if,S)    = {(q,CTBP),(q,CTBEBP)}
f(q,if,B)    = {(q,CTB),(q,CTBEB)}
f(q,var,B)   = {(q,FN)}
f(q,cond,C)  = {(q,λ)}
f(q,::=,F)   = {(q,λ)}
f(q,num,N)   = {(q,λ)}
f(q,;,P)     = {(q,λ)}
f(q,then,T)  = {(q,λ)}
f(q,else,E)  = {(q,λ)}
```



David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

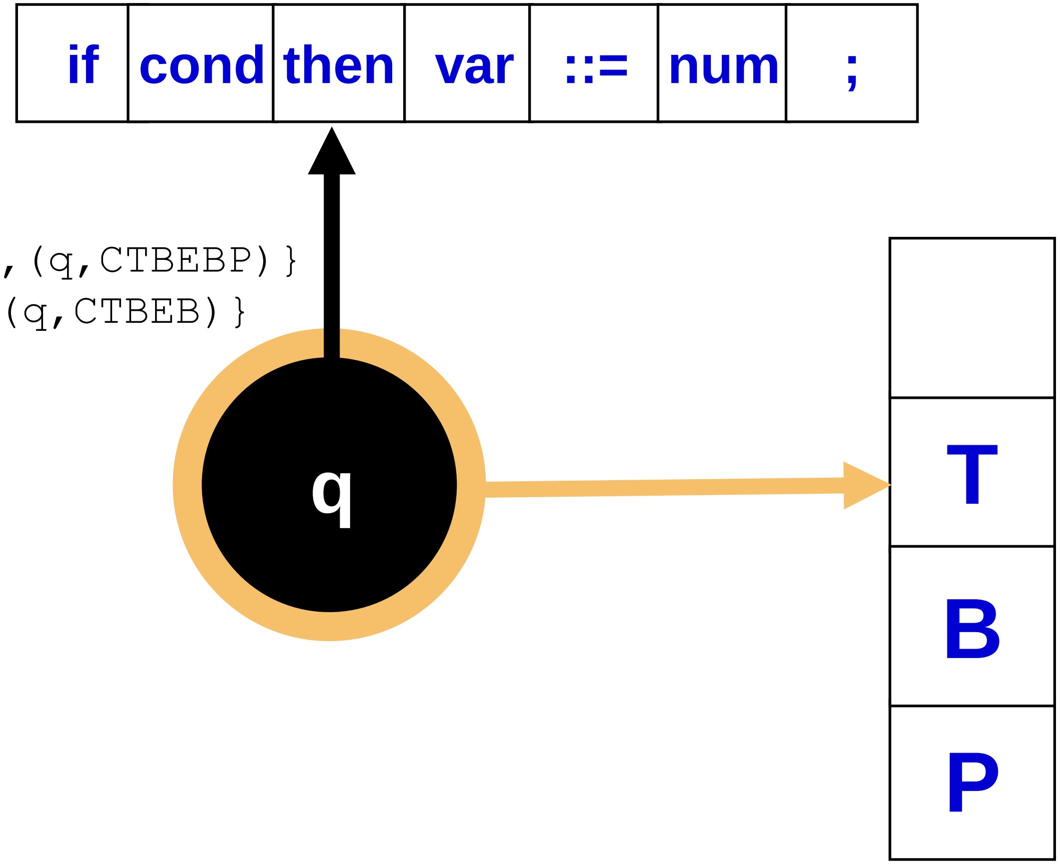
Definition of Push-Down Automaton. Example

```
f(q,var,S)   = {(q,FNP)}
f(q,if,S)    = {(q,CTBP),(q,CTBEBP)}
f(q,if,B)    = {(q,CTB),(q,CTBEB)}
f(q,var,B)   = {(q,FN)}
f(q,cond,C)  = {(q,λ)}
f(q,::=,F)   = {(q,λ)}
f(q,num,N)   = {(q,λ)}
f(q,;,P)     = {(q,λ)}
f(q,then,T)  = {(q,λ)}
f(q,else,E)  = {(q,λ)}
```



David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

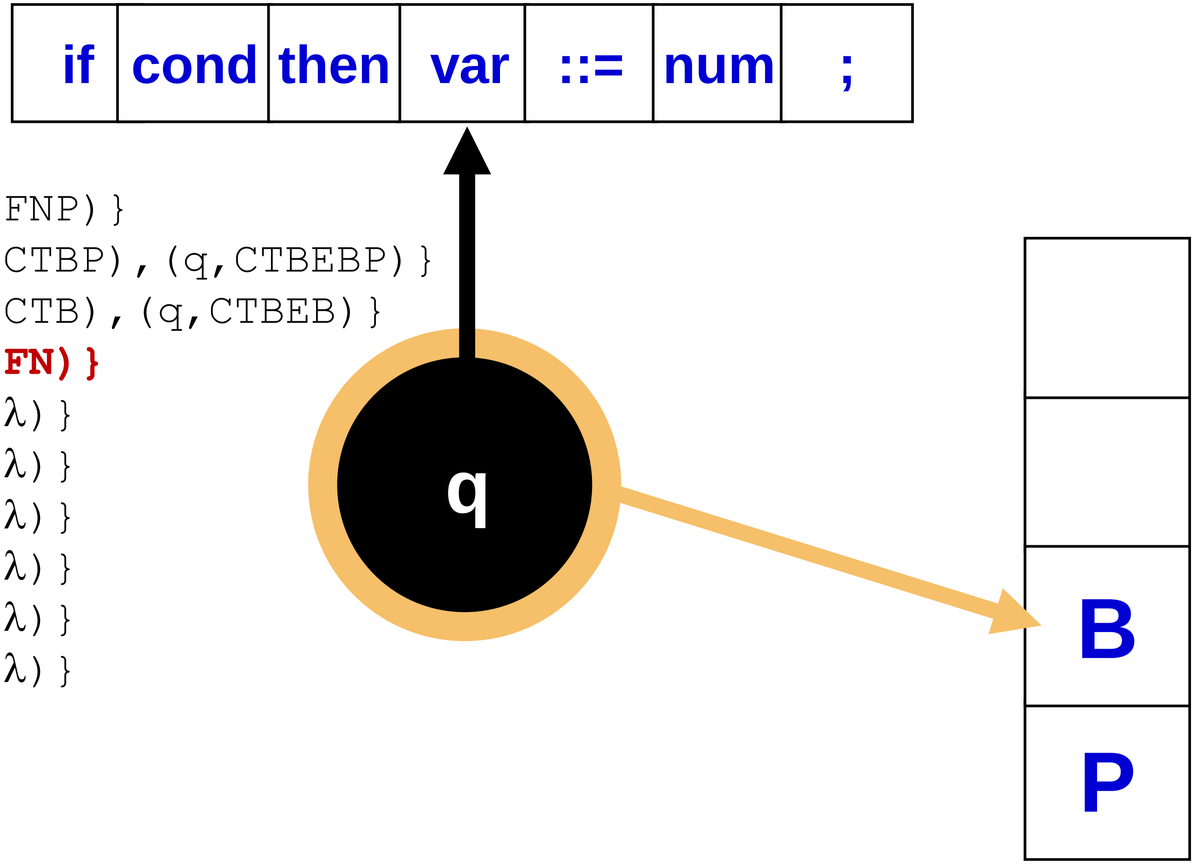
Definition of Push-Down Automaton. Example




```
f(q,var,S)   = {(q,FNP)}
f(q,if,S)    = {(q,CTBP),(q,CTBEBP)}
f(q,if,B)    = {(q,CTB),(q,CTBEB)}
f(q,var,B)   = {(q,FN)}
f(q,cond,C)  = {(q,λ)}
f(q,::=,F)   = {(q,λ)}
f(q,num,N)   = {(q,λ)}
f(q,;,P)     = {(q,λ)}
f(q,then,T)  = {(q,λ)}
f(q,else,E)  = {(q,λ)}
```


David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

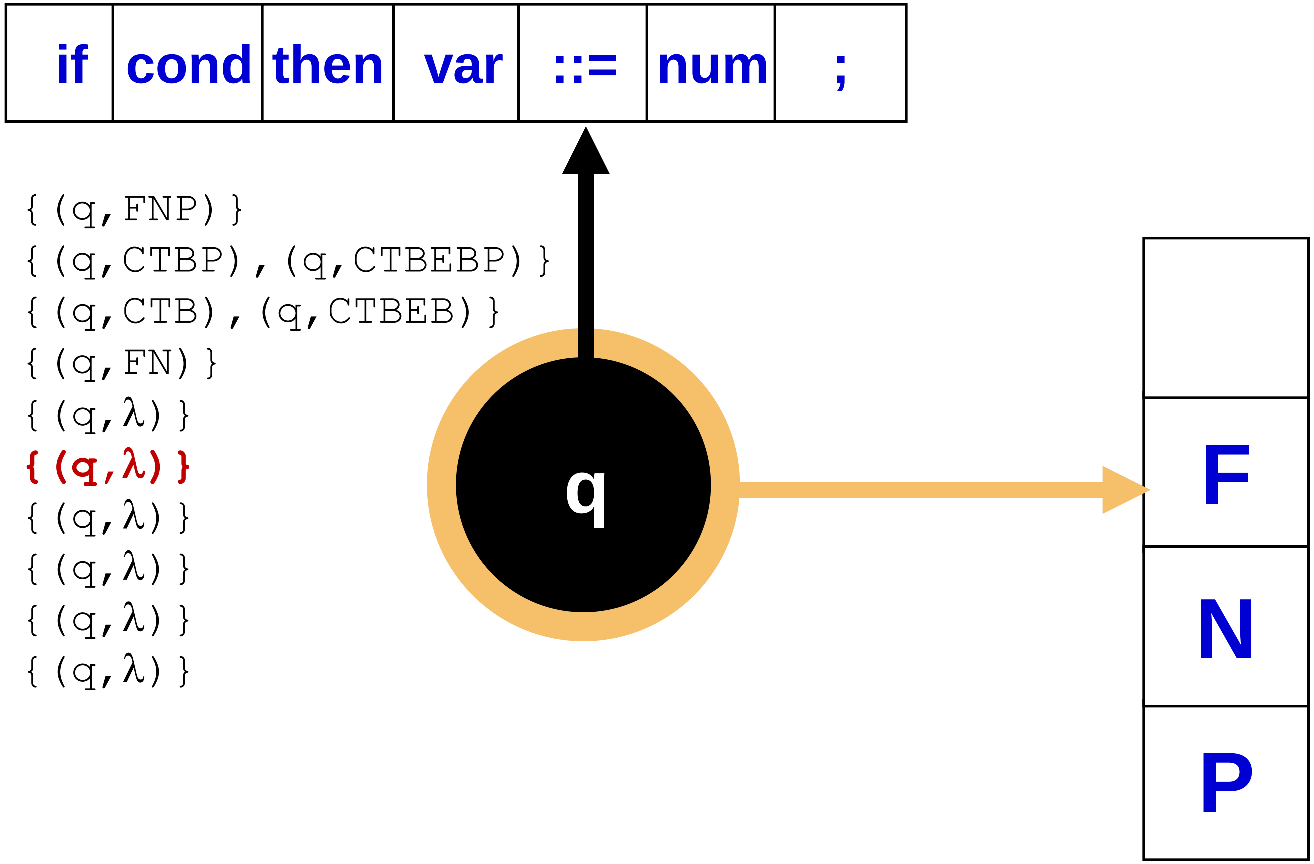
Definition of Push-Down Automaton. Example




```
f(q,var,S)   = {(q,FNP)}
f(q,if,S)    = {(q,CTBP),(q,CTBEBP)}
f(q,if,B)    = {(q,CTB),(q,CTBEB)}
```
f(q,var,B)   = {(q,FN)}
```
f(q,cond,C)  = {(q,λ)}
f(q,::=,F)   = {(q,λ)}
f(q,num,N)   = {(q,λ)}
f(q,;,P)     = {(q,λ)}
f(q,then,T)  = {(q,λ)}
f(q,else,E)  = {(q,λ)}
```


David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

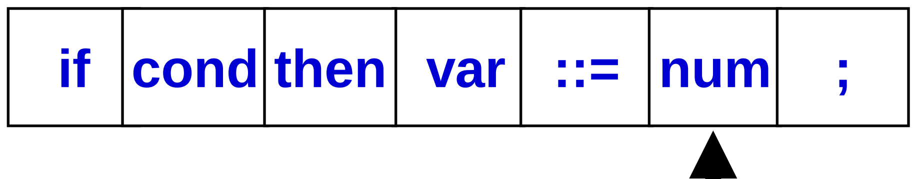
Definition of Push-Down Automaton. Example



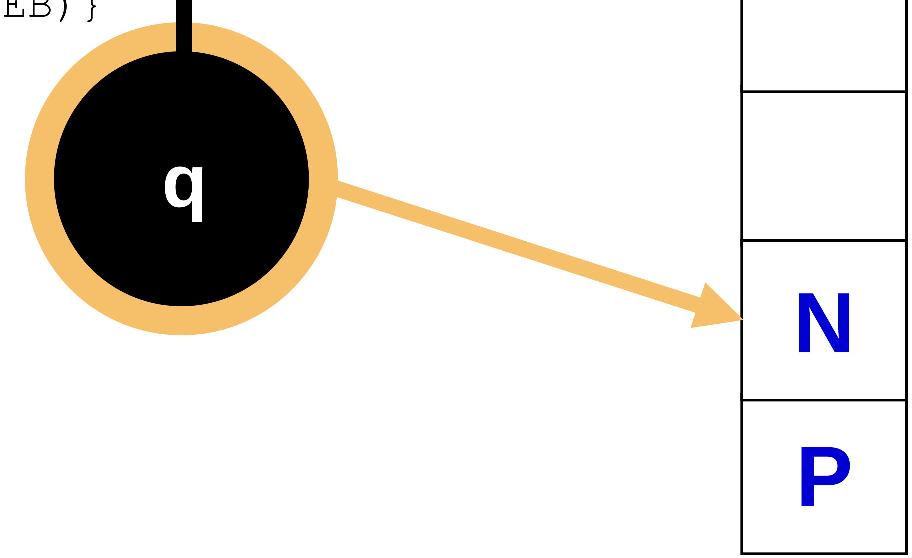

```
f(q,var,S)  = {(q,FNP)}
f(q,if,S)   = {(q,CTBP),(q,CTBEBP)}
f(q,if,B)   = {(q,CTB),(q,CTBEB)}
f(q,var,B)  = {(q,FN)}
f(q,cond,C) = {(q,λ)}
f(q,::=,F)  = {(q,λ)}
f(q,num,N)  = {(q,λ)}
f(q,;,P)    = {(q,λ)}
f(q,then,T) = {(q,λ)}
f(q,else,E) = {(q,λ)}
```


David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Definition of Push-Down Automaton. Example



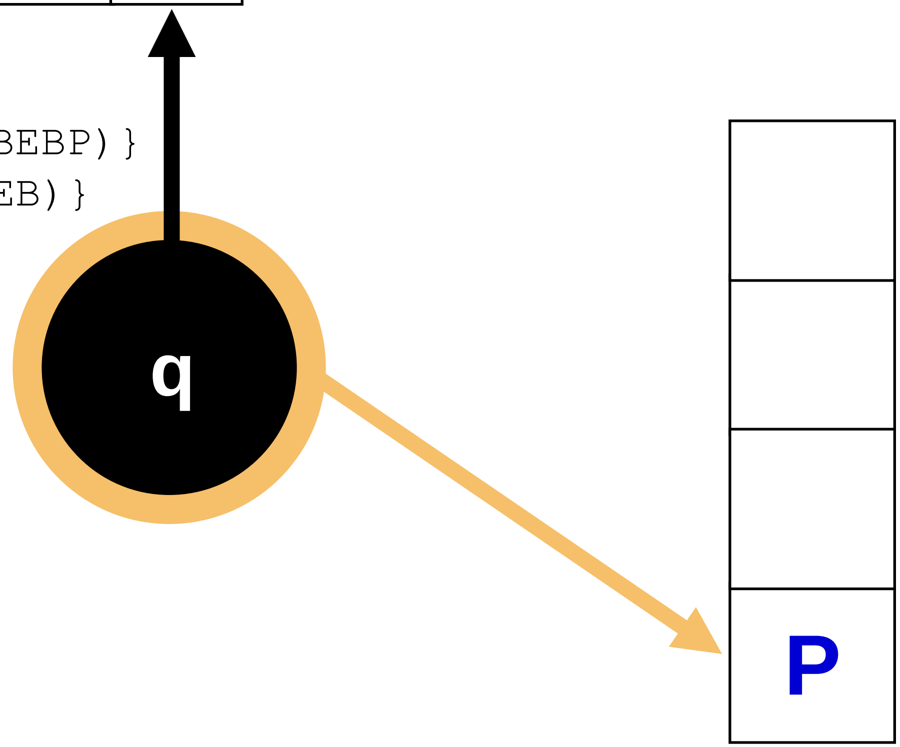

```
f(q,var,S)   = {(q,FNP)}
f(q,if,S)    = {(q,CTBP),(q,CTBEBP)}
f(q,if,B)    = {(q,CTB),(q,CTBEB)}
f(q,var,B)   = {(q,FN)}
f(q,cond,C)  = {(q,λ)}
f(q,::=,F)   = {(q,λ)}
f(q,num,N)   = {(q,λ)}
f(q,;,P)     = {(q,λ)}
f(q,then,T)  = {(q,λ)}
f(q,else,E)  = {(q,λ)}
```



David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

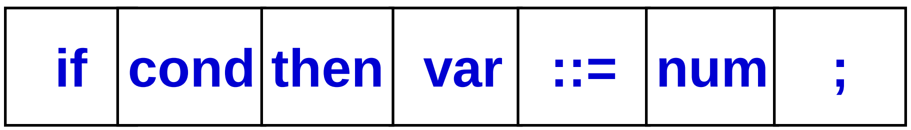
Definition of Push-Down Automaton. Example

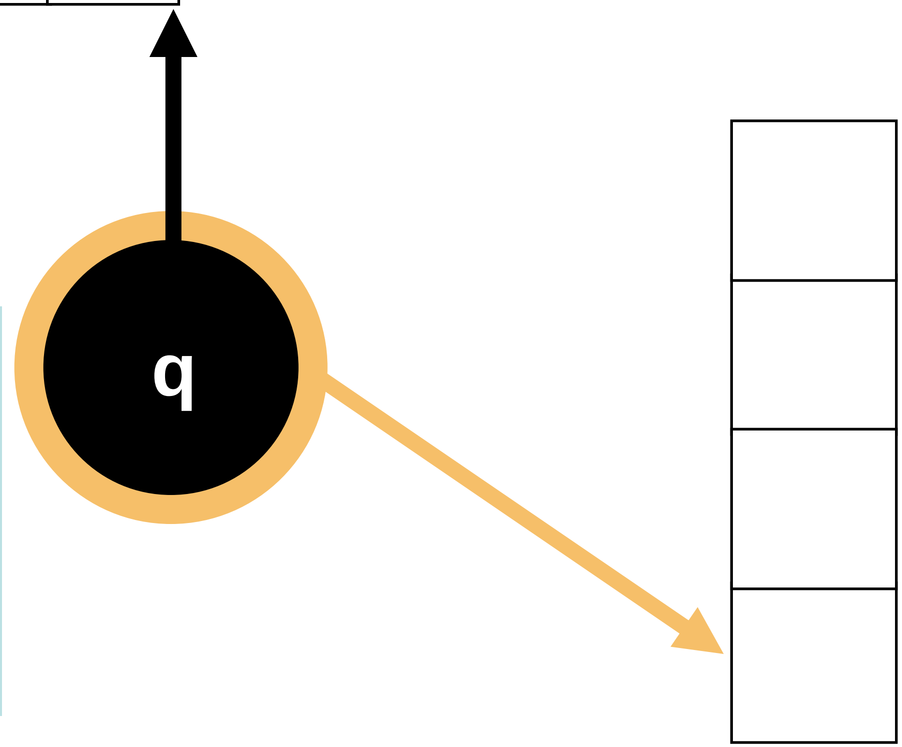

```
f(q,var,S)   = {(q,FNP)}
f(q,if,S)    = {(q,CTBP),(q,CTBEBP)}
f(q,if,B)    = {(q,CTB),(q,CTBEB)}
f(q,var,B)   = {(q,FN)}
f(q,cond,C)  = {(q,λ)}
f(q,::=,F)   = {(q,λ)}
f(q,num,N)   = {(q,λ)}
f(q,;,P)     = {(q,λ)}
f(q,then,T)  = {(q,λ)}
f(q,else,E)  = {(q,λ)}
```



David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Definition of Push-Down Automaton. Example





Word read

Empty stack

Sentence recognized

OUTLINE

- Introduction
- Definition of Push-Down Automata
  - Acceptance in final states or when the stack is empty
  - Formal definition
  - Transitions
  - Instantaneous Description, Movement
  - Deterministic Push-Down Automata
  - Language Accepted by a Push-Down Automaton
  - Examples
- **Equivalence between PD Automata and Context-Free Languages**

Push-Down Automaton. Theorem

- For each push-down automaton accepting strings without emptying the stack (**PDA_F**), there is an equivalent automaton that empties the stack before an accepting state (**PDA_E**).

$$L(PDA_F) = L(PDA_E)$$

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Equivalence PDA_E and PDA_F

From PDA_F to PDA_E

$$PDA_F = (\Sigma, \Gamma, Q, A_0, q_0, f, F)$$

$$PDA_E = (\Sigma, \Gamma \cup \{B\}, Q \cup \{p, r\}, B, p, f', \phi)$$

- $\Gamma \cup \{B\}$: New symbol for the stack
- $Q \cup \{p, r\}$: Two new states
- B: Initial value on the stack
- p: New initial state
- ϕ: WITHOUT FINAL STATES

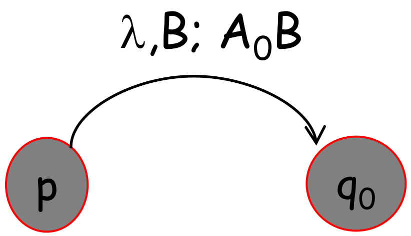
David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Equivalence PDA_E and PDA_F

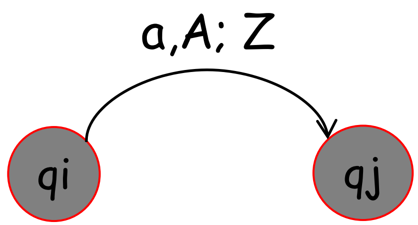
$PDA_F = (\Sigma, \Gamma, Q, A_0, q_0, f, F)$

$PDA_E=(\Sigma, \Gamma \cup \{B\}, Q \cup \{p,r\}, B, p, f', \phi)$

f' is defined as following:



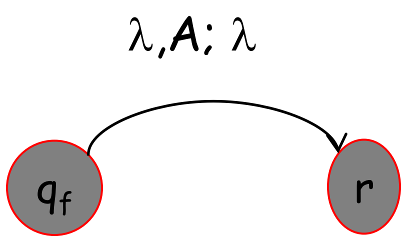
Transition independent of the input of the PDA_E with the first symbol of the stack transiting to the state q_0 of the PDA_F and putting A_0 on the stack.



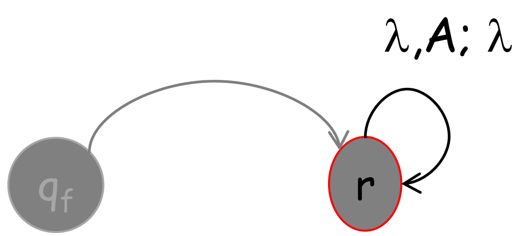
$q_i, q_j \in Q$, $a \in \Sigma \cup \{\lambda\}$, $A \in \Gamma$, $Z \in \Gamma^*$,

The transitions in the PDA_F are kept.

The characteristics of acceptance of this state are removed.



$\forall q_f \in F$, $A \in \Gamma \cup \{B\}$



$\forall A \in \Gamma \cup \{B\}$

Equivalence PDA_E and PDA_F

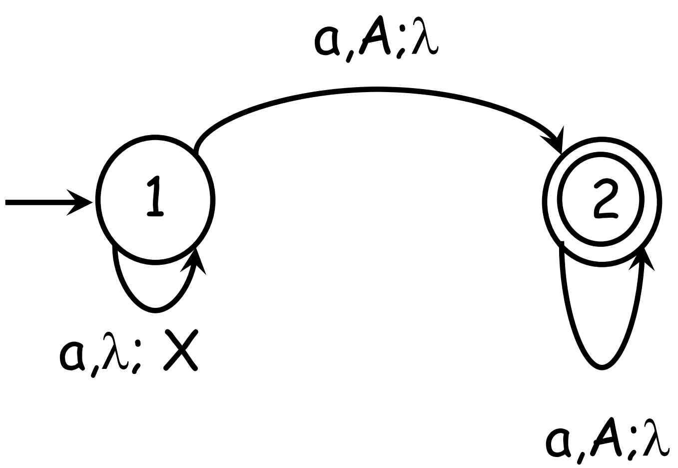
$PDA_F = (\Sigma, \Gamma, Q, A_0, q_0, f, F)$
$PDA_E = (\Sigma, \Gamma \cup \{B\}, Q \cup \{p,r\}, B, p, f', \phi)$ **f' is defined as following:**

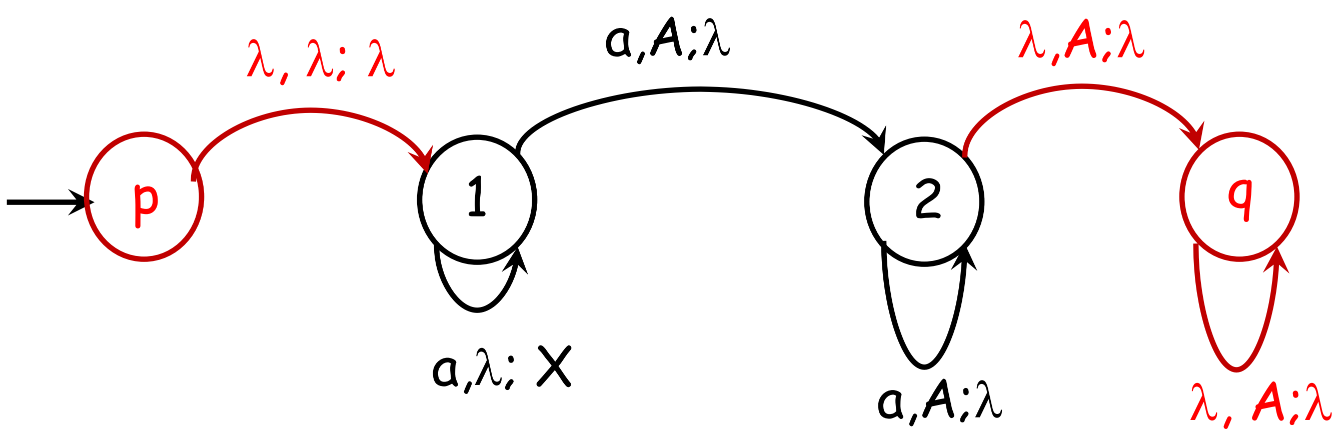
- $f'(p, \lambda, B) = (q_0, A_0 \cdot B)$
    - A new initial state is incorporated and a transition from this new state to the original initial state of the PDA_F, the transition inserts A to which already existed: A_0B
- $f'(q, a, A) = f(q, a, A) \ \forall q \in Q, \ a \in \Sigma \cup \{\lambda\}, A \in \Gamma$
    - Eliminate the characteristics of acceptance of each state.
- $(r, \lambda) \in f'(q, a, A) \ \forall q \in F, A \in \Gamma \cup \{B\}$
    - A new state p is added along with the transitions to acceptance states of acceptance to q_f, without reading, extracting or inserting symbols.
- $(r, \lambda) \in f'(r, \lambda, A) \ \forall A \in \Gamma \cup \{B\}$
    - For each $A \in \Gamma$, add the transition $(r,\lambda,A;r,\lambda)$

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Equivalence PDA_E and PDA_F

From PDA_F to PDA_E





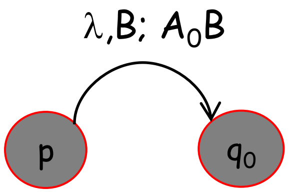
David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Equivalence PDA_E and PDA_F

$PDA_E=(\Sigma, \Gamma, Q, A_0, q_0, f, \phi) \rightarrow PDA_F=(\Sigma, \Gamma \cup \{B\}, Q \cup \{p, r\}, B, p, f', \{r\})$

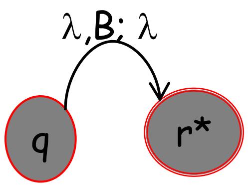
f' is defined as following:

- $f'(p, \lambda, B) = (q_0, A_0 \cdot B)$



- $f(q, a, A) = f'(q, a, A) \quad \forall q \in Q, \; a \in \Sigma \cup \{\lambda\}, A \in \Gamma$

- $(r, \lambda) \in f'(q, \lambda, B) \quad \forall q \in Q,$



David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Equivalence PDA_E and PDA_F

$PDA_E=(\Sigma, \Gamma, Q, A_0, q_0, f, \phi) \rightarrow PDA_F=(\Sigma, \Gamma \cup \{B\}, Q \cup \{p, r\}, B, p, f', \{r\})$

f' is defined as following:

- $f'(p, \lambda, B) = (q_0, A_0 \cdot B)$
  - The first transition of the PDA_F is to go to q0 of the PDA_V and write A0B on the stack, verifying that B is down on the stack.

- $f(q, a, A) = f'(q, a, A) \quad \forall\, q \in Q,\ a \in \Sigma \cup \{\lambda\},\ A \in \Gamma$
  - The transitions of the PDA_E are kept (the original PDA)

- $(r, \lambda) \in f'(q, \lambda, B) \quad \forall\, q \in Q,$
  - When there is no input, it goes to the final state of the PDA_F: in the stack only remains B (that was introduced at the beginning) . Therefore, the word x is accepted and it goes to the final state.

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

Given a G2 in GNF, construct a PDA_E:

$G = (\Sigma_T, \Sigma_N, S, P)$

- $PDA_E = (\Sigma_T, \Sigma_N, \{q\}, S, q, f, \phi)$ We obtain an PDA_E with only one state.

$(q, Z) \in f(q, a, A)$

i.e., $f(q, a, A) = (q, Z)$ if there is a production with the form A ::= aZ.
$f(q, a, A) = (q, \lambda)$ if there is a production with the form A ::= a

$f(q, a, A) = \{(q, Z), (q, \lambda)\}$

Given a production $A ::= aZ \mid aD \mid b \Rightarrow$ $f(q, a, A)= \{(q, Z), (q, D)\}$
$f(q, b, A) = (q, \lambda)$

- **If $S ::= \lambda \Rightarrow (q, \lambda) \in f(q, \lambda, S)$**

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es

- **Given a G2, construct a PDA_F:**
  - $G = (\Sigma_T, \Sigma_N, S, P)$
  - $PDA_E = (\Sigma_T, \Gamma, Q, A_0, q_0, f, \{q_2\})$
- Where:
  - $\Gamma = \Sigma_T \cup \Sigma_N \cup \{A_0\}$, where $A_0 \notin \Sigma_T \cup \Sigma_N$
  - $Q=\{q_0, q_1, q_2\}$, q_0 is the initial state, q_1 is the state from which transitions are carried out and q_2 is the final state.

- *f* is defined as follows:
  - $f(q_0, \lambda, A_0)=\{q_1, SA_0\}$
  - $\forall A \in \Sigma_N$, if $A ::= \alpha \in P$, $(\alpha \in \Sigma^*) \Rightarrow (q_1, \alpha) \in f(q_1, \lambda, A)$
  - $\forall\ a \in \Sigma_T$, $(q_1, \lambda) \in f(q_1, a, a)$
  - $f(q_1, \lambda, A_0) = \{q_2, A\}$

- **Given a PDA_E, construct a G2 that fulfills $L(G2) = L(PDA_E)$**
  - $PDA_E = (\Sigma, \Gamma, Q, A_0, q_o, f, \phi)$
  - $G = (\Sigma_T, \Sigma_N, S, P)$

- $\Sigma_N = \{S\} \cup \{ (p,A,q) \mid p,q \in Q, \; A \in \Gamma\}$

- **To construct P:**
  1. $S ::= (q_0, A_0, q) \quad \forall q \in Q$ (select those that begins with q_0A_0)
  2. From each transition $f(p,a,A) = (q, BB'B''....B''')$ where $A,B,B',B'',...,B''' \in \Gamma$; $a \in \Sigma \cup \{\lambda\}$
     - $(p\ A\ z) ::= a\ (q\ B\ r)\ (r\ B'\ s)\ s\ ...\ y\ (y\ B'''\ z)$
  3. From each transition $f(p, a, A) = (q, \lambda)$, we obtain: $(p,A,q) ::= a$

David Griol Barres - Computer Science Department – UC3M - dgriol@inf.uc3m.es