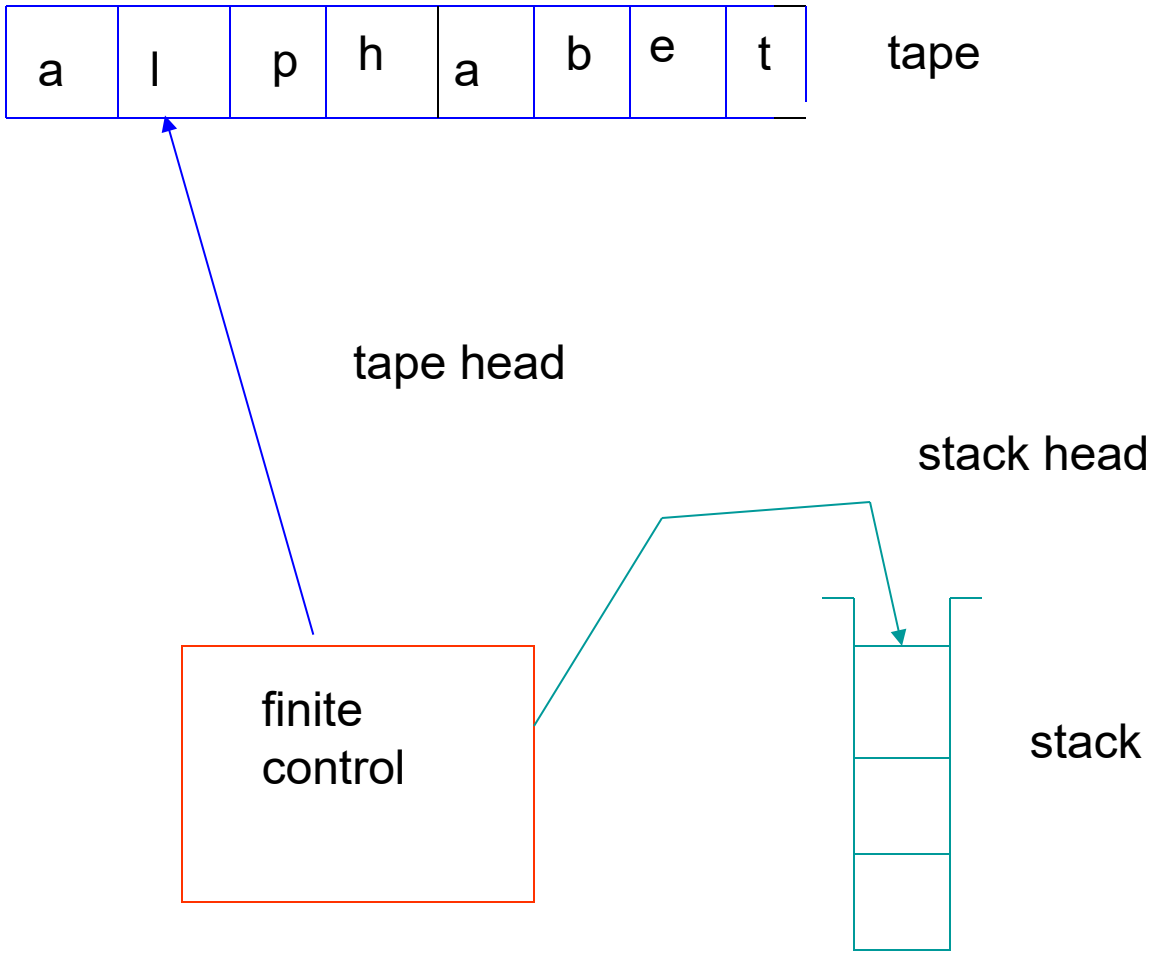
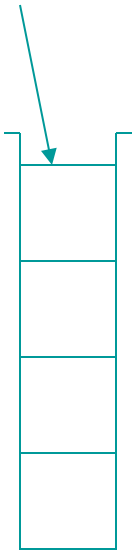
Pushdown Automata –PDA



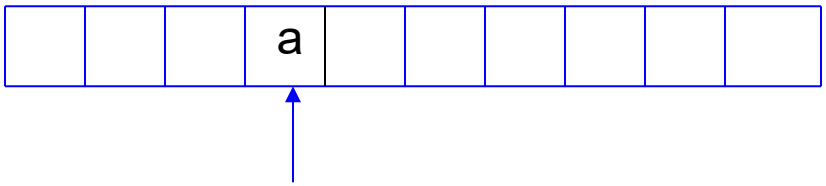
The tape is divided into finitely many cells. Each cell contains a symbol in an alphabet Σ.



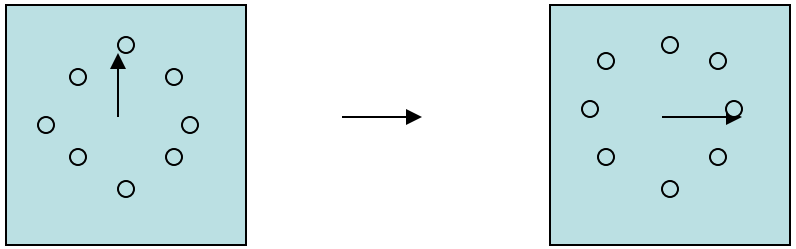
The stack head always scans the top symbol of the stack.
It performs two basic operations:

Push: add a new symbol at the top.
Pop: read and remove the top symbol.

Alphabet of stack symbols: Γ



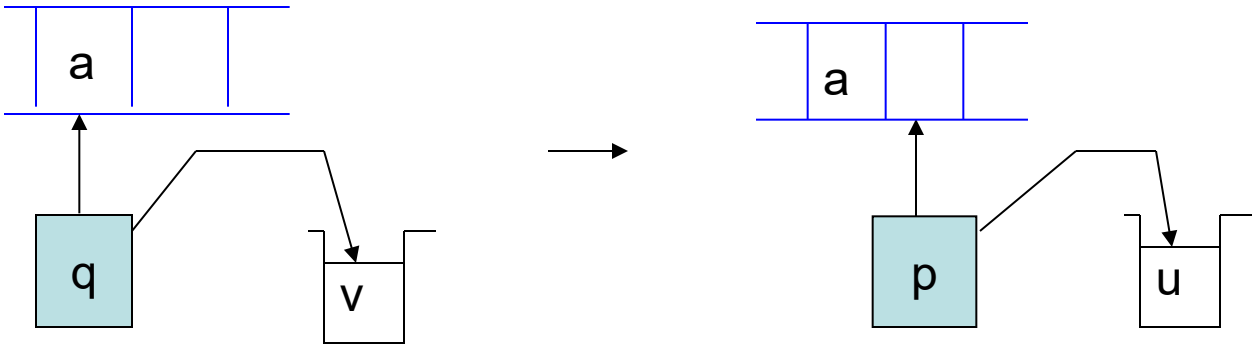
The head scans at a cell on the tape and can read a symbol on the cell.
In each move, the head can move to the right cell.



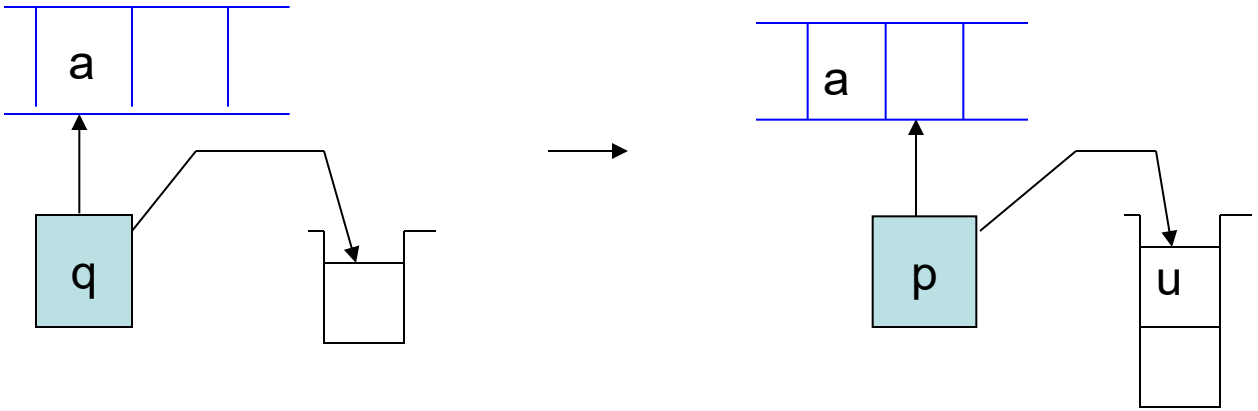
The finite control has finitely many states which form a set Q.

For each move, the state is changed according to the evaluation of a ***transition function***

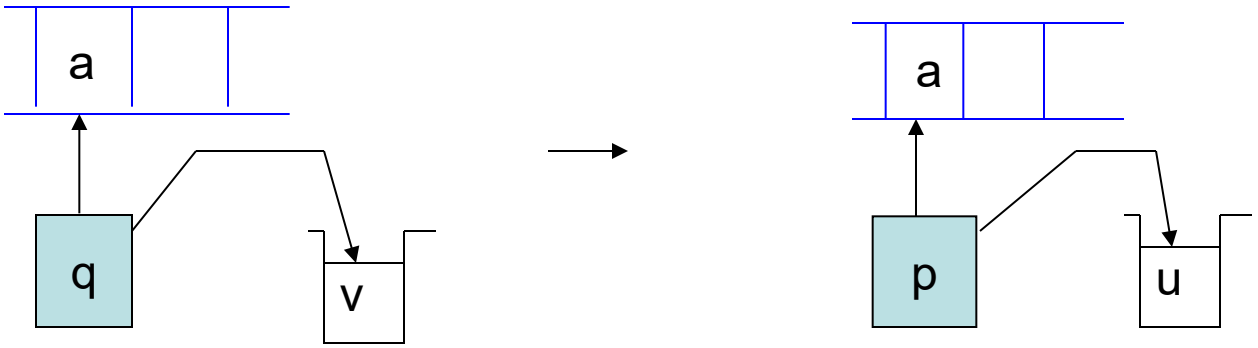
$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \rightarrow 2^{Q \times (\Gamma \cup \{\varepsilon\})}.$$



$(p, u) \in \delta(q, a, v)$ means that if the tape head reads a, the stack head read v, and the finite control is in the state q, then one of possible moves is that the next state is p, v is replaced by u at stack, and the tape head moves one cell to the right.

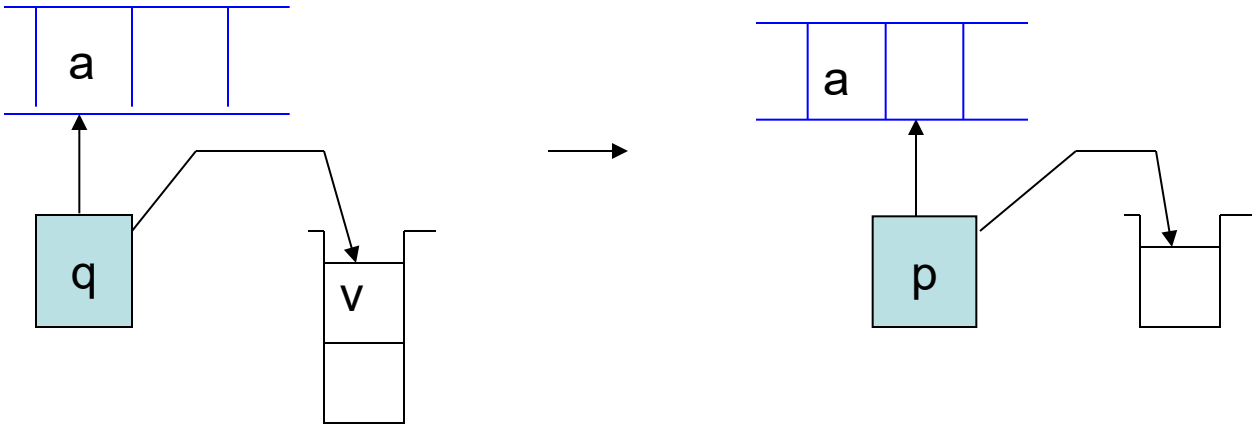


$(p, u) \in \delta(q, a, \varepsilon)$ means that a push operation performs at stack.

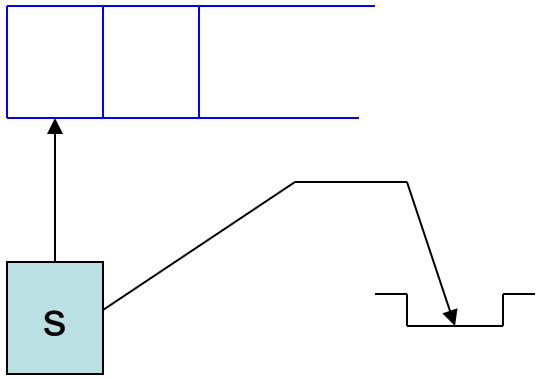


$\in$

(p, u) *δ(q, ε, v)* means that this a ε-move.



$(p, \varepsilon) \in \delta(q, a, v)$ means that a pop operation performs at stack

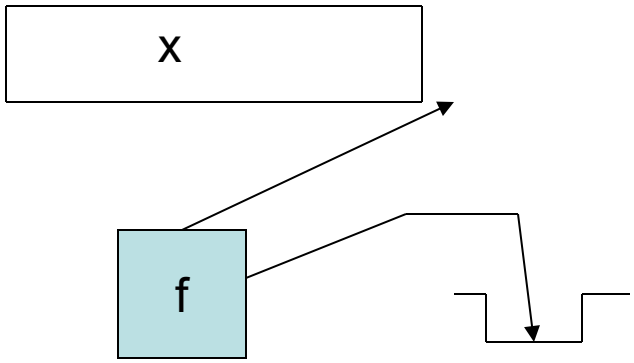


There are some special states: an initial state *s* and a final set F of final states.

Initially, the PDA is in the initial state s and the head scans the leftmost cell.

The tape holds an input string.

The stack is empty.



When the head gets off the tape, the PDA stops.

An input string *x* is accepted by the PDA if the PDA stops at a final state and the stack is empty.

Otherwise, the input string is rejected.

The PDA can be represented by

$M = (Q, \Sigma, \Gamma, \delta, s, F)$

where Σ is the alphabet of input symbols

Γ is the alphabet of stack symbols.

The set of all strings accepted by a PDA M is denoted by $L(M)$.

We also say that the language $L(M)$ is accepted by M.

Push Down Automata Formal Definition

Definition: A Pushdown Automaton (PDA) is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$

Σ input alphabet, Γ stack alphabet ,

δ: $\mathrm{Q} \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$, $\delta(q, \mathrm{a}, \mathrm{c}) = \{(r_1, \mathrm{d}), (r_2, \mathrm{e})\}$

Sample input:

0	1	1	1	1	0

Accept if some thread is in the accept state at the end of the input string.

Example: PDA for $B = \{ww^{\mathcal{R}} | \, w \in \{0,1\}^* \}$

1) Read and push input symbols.
   Nondeterministically either repeat or go to (2).
2) Read input symbols and pop stack symbols, compare.
   If ever ≠ then thread rejects.
3) Enter accept state if stack is empty. (do in "software")

Conventions

a, b, … are input symbols.

But sometimes we allow ε as a possible value.

…, X, Y, Z are stack symbols.

…, w, x, y, z are strings of input symbols.

α, β,… are strings of stack symbols.

PDA Formalism

A PDA is described by:

A finite set of *states* (Q, typically).

An *input alphabet* (Σ, typically).

A *stack alphabet* (Γ, typically).

A *transition function* (δ, typically).

A *start state* (q_0, in Q, typically).

A *start symbol* (Z_0, in Γ, typically).

A set of *final states* (F ⊆ Q, typically).

The Transition Function

Takes three arguments:

A state, in Q.

An input, which is either a symbol in Σ or ε.

A stack symbol in Γ.

δ(q, a, Z) is a set of zero or more actions of the form (p, α).

p is a state; α is a string of stack symbols.

Actions of the PDA

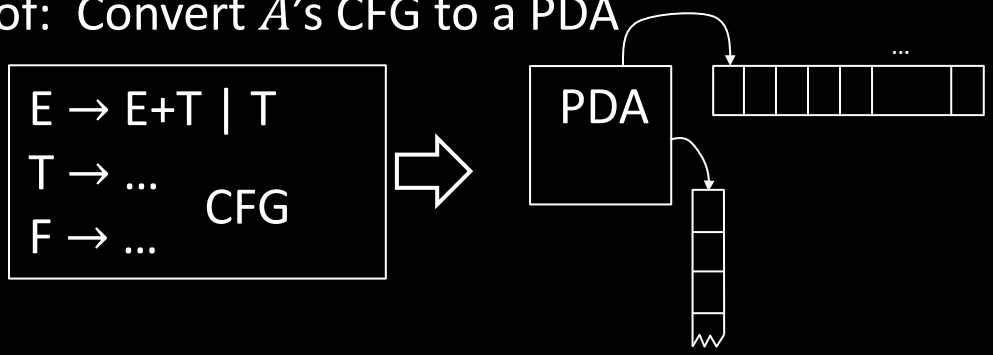
If $\delta(q, a, Z)$ contains (p, α) among its actions, then one thing the PDA can do in state q, with *a* at the front of the input, and Z on top of the stack is:

1. Change the state to p.
2. Remove *a* from the front of the input (but *a* may be ε).
3. Replace Z on the top of the stack by α.

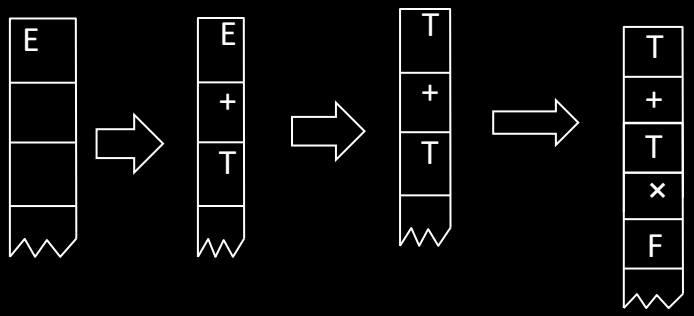
Converting CFGs to PDAs

Theorem: If A is a CFL then some PDA recognizes A

Proof: Convert A's CFG to a PDA



DEA: PDA begins with starting variable and guesses substitutions.
It keeps intermediate generated strings on stack. When done, compare with input.



Input: | a | + | a | × | a |

G_2

E → E+T | T
T → T×F | F
F → (E) | a

E ⇒ E+T ⇒ T+T×F

⇒F+Fxa⇒ a+a×a

Problem! Access below the top of stack is cheating!

Instead, only substitute variables when on the top of stack.

If a terminal is on the top of stack, pop it and compare with input. Reject if ≠.

Converting CFGs to PDAs (contd)

Theorem: If A is a CFL then some PDA recognizes A

Proof construction: Convert the CFG for A to the following PDA.

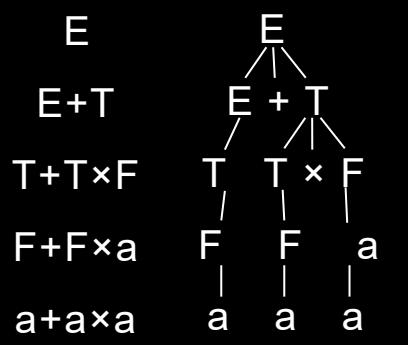
G_2

E → E+T | T
T → T×F | F
F → (E) | a

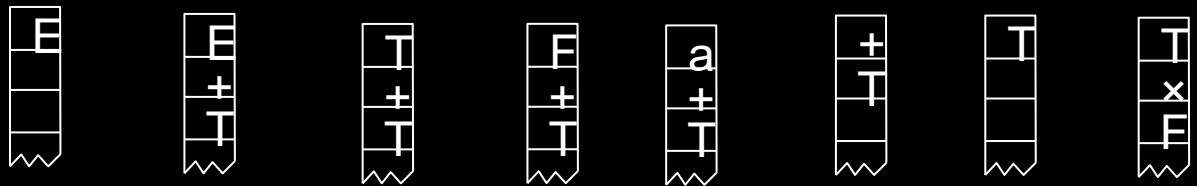
1) Push the start symbol on the stack.
2) If the top of stack is

   Variable: replace with right hand side of rule (nondeterministic choice).

   Terminal: pop it and match with next input symbol.
3) If the stack is empty, *accept.*



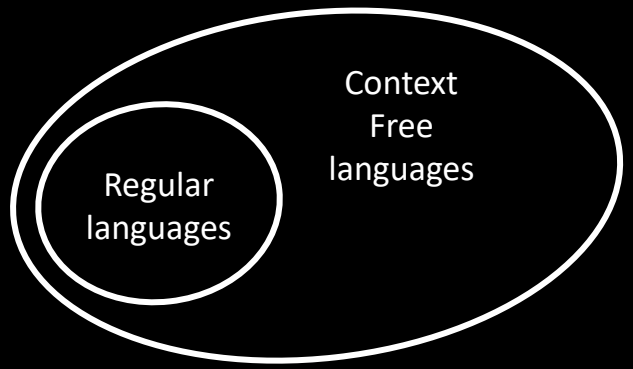
Example:



Equivalence of CFGs and PDAs

	Recognizer	Generator
Regular language	DFA or NFA	Regular expression
Context Free language	PDA	Context Free Grammar

Theorem: A is a Context Free Language iff (if and onlu if) some PDA recognizes A



Is every Regular Language also a Context Free Language?

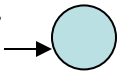
(a) Yes

(b) No

(c) Not sure

The transition diagram of a PDA is an alternative way to represent the PDA.

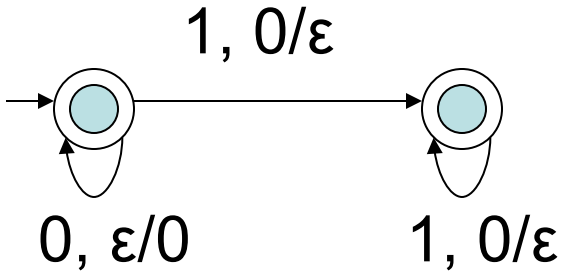
For $M = (Q, \Sigma, \Gamma, \delta, s, F)$, the transition diagram of M is an edge-labeled digraph $G=(V, E)$ satisfying the following:

$V = Q$ ($s =$  $f =$  for $f \in F$)

$E = \{ q \xrightarrow{a,\ v/u} p \mid (p,u) \in \delta(q, a, v) \}$

Example 1. Construct PDA to accept $L = \{0^n 1^n \mid n \geq 0\}$

Solution 1.



Example: PDA

Design a PDA to accept $\{0^n1^n \mid n \geq 1\}$.

The states:

q = start state. We are in state q if we have seen only 0's so far.

p = we've seen at least one 1 and may now proceed only if the inputs are 1's.

f = final state; accept.

Example: PDA

The stack symbols:

Z_0 = start symbol. Also marks the bottom of the stack, so we know when we have counted the same number of 1's as 0's.

X = marker, used to count the number of 0's seen on the input.

Example: PDA

The transitions:

$\delta(q, 0, Z_0) = \{(q, XZ_0)\}$.

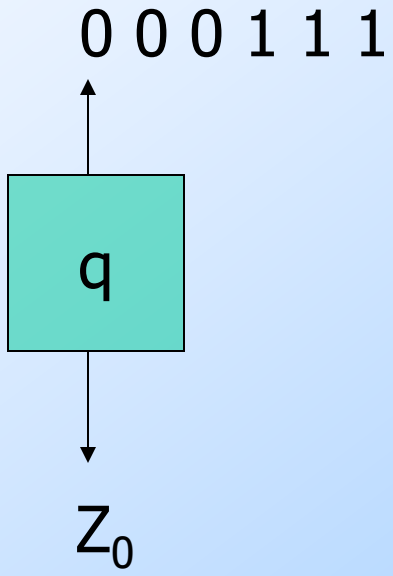
$\delta(q, 0, X) = \{(q, XX)\}$. These two rules cause one X to be pushed onto the stack for each 0 read from the input.

$\delta(q, 1, X) = \{(p, \varepsilon)\}$. When we see a 1, go to state p and pop one X.

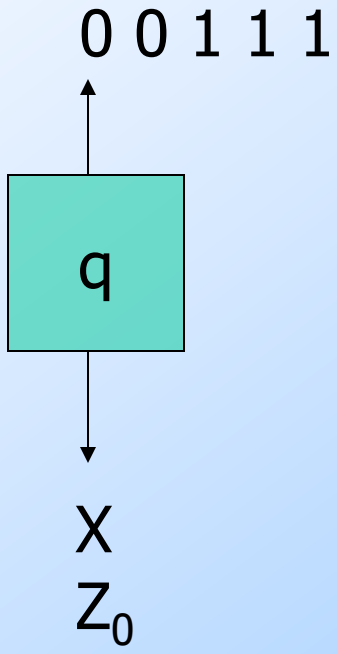
$\delta(p, 1, X) = \{(p, \varepsilon)\}$. Pop one X per 1.

$\delta(p, \varepsilon, Z_0) = \{(f, Z_0)\}$. Accept at bottom.

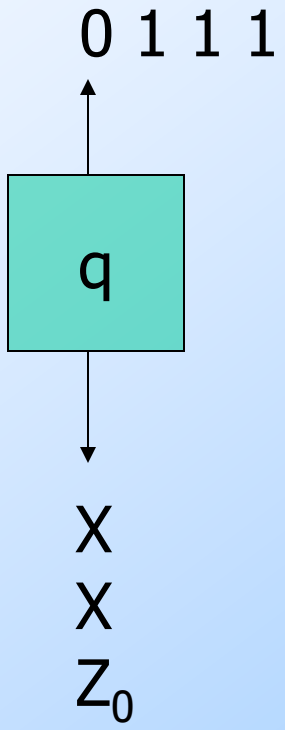
Actions of the Example PDA



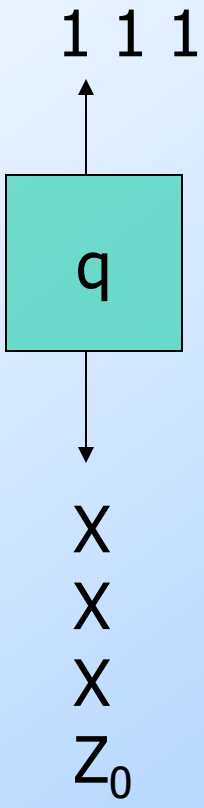
Actions of the Example PDA



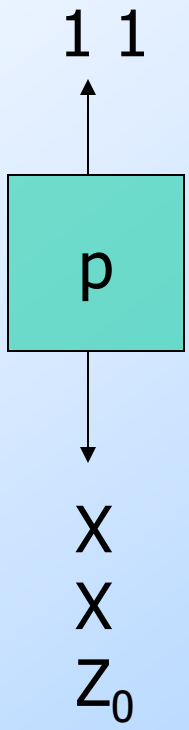
Actions of the Example PDA



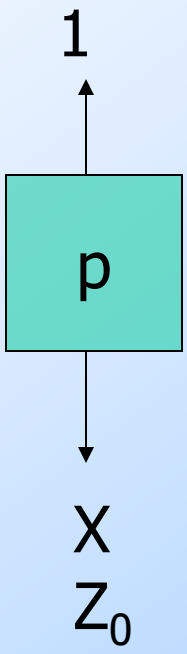
Actions of the Example PDA



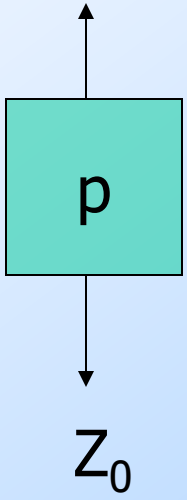
Actions of the Example PDA



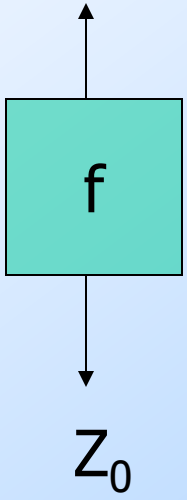
Actions of the Example PDA



Actions of the Example PDA



Actions of the Example PDA



Instantaneous Descriptions

We can formalize the pictures just seen with an *instantaneous description* (ID).

A ID is a triple (q, w, α), where:

q is the current state.

w is the remaining input.

α is the stack contents, top at the left.

Example

Using the previous example PDA, we can describe the sequence of moves by:

$(q, 000111, Z_0) \vdash (q, 00111, XZ_0) \vdash (q, 0111, XXZ_0)$

$\vdash (q, 111, XXXZ_0) \vdash (p, 11, XXZ_0)$

$\vdash (p, 1, XZ_0) \vdash (p, \varepsilon, Z_0) \vdash (f, \varepsilon, Z_0)$

Thus, $(q, 000111, Z_0) \vdash^* (f, \varepsilon, Z_0)$.

What would happen on input 0001111?

Answer

$(q, 0001111, Z_0) \vdash (q, 001111, XZ_0)$

$\vdash (q, 01111, XXZ_0) \vdash (q, 1111, XXXZ_0)$

$\vdash (p, 111, XXZ_0) \vdash (p, 11, XZ_0)$

$\vdash (p, 1, Z_0) \vdash (f, 1, Z_0)$

Legal because a PDA can use ε input even if input remains.

Note the last ID (Instantaneous Descriptions) has no move.

0001111 is not accepted, because the input is not completely consumed.

Language of a PDA

The common way to define the language of a PDA is by *final state*.

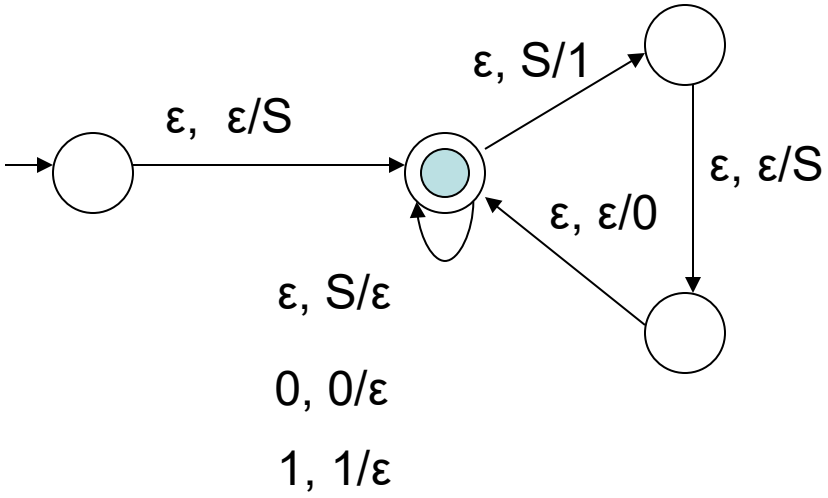
If P is a PDA, then L(P) is the set of strings w such that

$(q_0, w, Z_0) \vdash (f, \varepsilon, \alpha)$

for final state f and any α.

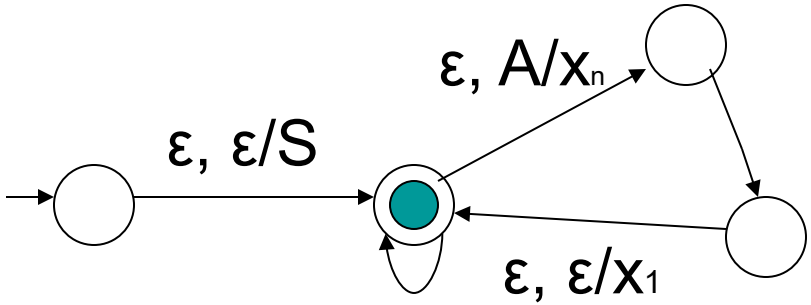
Solution 2.

Consider a CFG
G = ({S}, {0,1}, {S → ε | 0S1}, S).



Theorem: Every CFL can be accepted by a PDA

Proof Consider a Context Free Language L = L(G) for a CFG G = (V, Σ, R, S)



$A \rightarrow x_1 x_2 \ldots x_n$ in R

a, a/ε for a in Σ

Theorem

A language L is CFL $\Leftrightarrow$ it can be accepted by a PDA.

Sometimes, constructing the PDA is easier than constructing CFG.

Example 2

Show that $\{x \in \{a,b\} \mid \#_b(x) \leq \#_a(x) \leq 2\#_b(x)\}$ is a CFL

Construct a PDA or a CFG?

PDA!!!

Sometimes, constructing the PDA is easier than constructing CFG.