

Equivalence of PDAs and CFGs

A language is context iff some PDA recognize it

If $L=L(G)$ for some CFG G , then $L=L(M)$ for some PDA M .

Proof idea

* Need to show that given CFG G , we can find PDA M that recognize the same language

G generates.

* Basic idea is to build PDA that simulates a leftmost derivation.

* For example, consider CFG $G=(V,\Sigma,R,S)$

Variables $V= \{S\}$

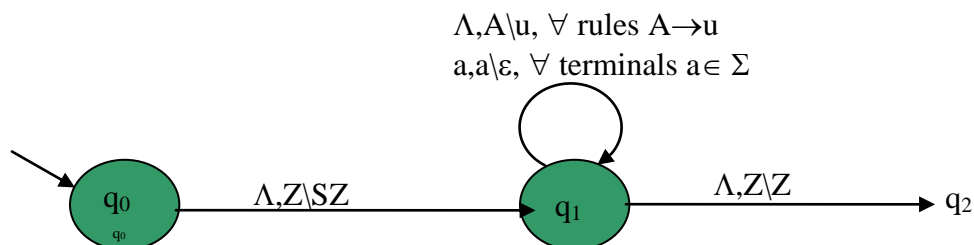
Terminals $\Sigma=\{0,1\}$

Rules $R= S \rightarrow 0S0S \mid 1S0 \mid 1$

* Leftmost derivation of string $010110 \in L(G)$

$S \rightarrow 0S0S \rightarrow 010S \rightarrow 0101S0 \rightarrow 010110$

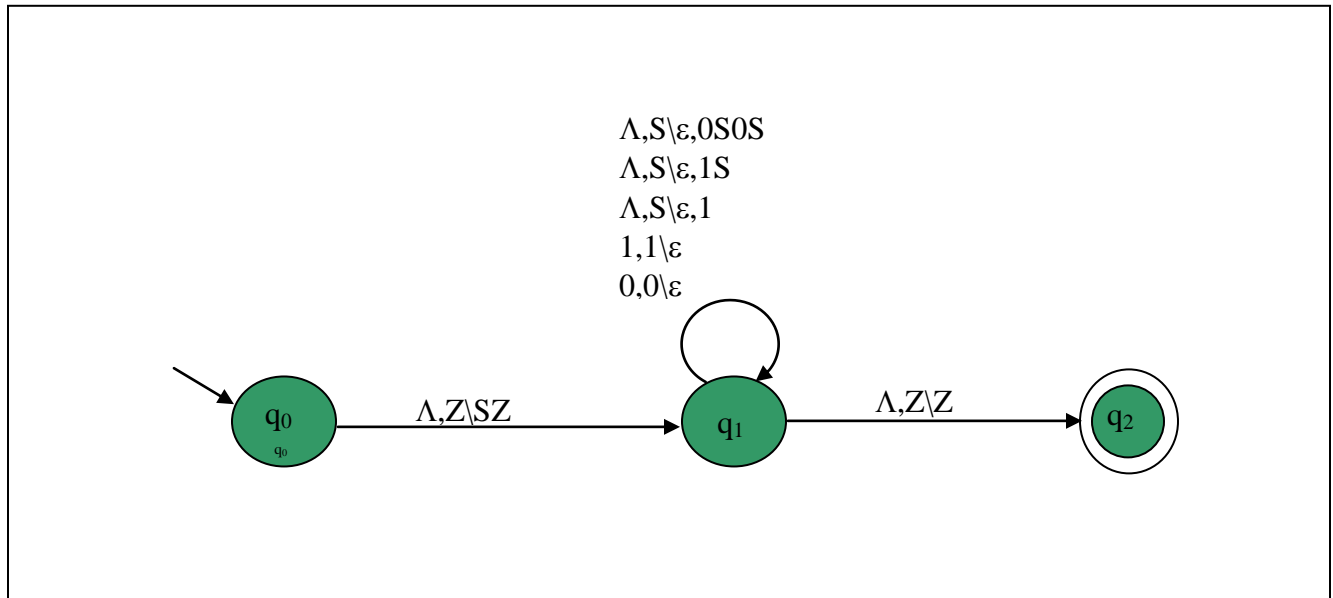
Creat PDA for CFG as follow:



*** PDA works as follow:**

- 1- push S on the stack, where S is start variable.
- 2- Repeat following until stack empty
 - (a) if top of stack is variable $A \in V$, then replace A by some $u \in (\Sigma \cup V)^*$, where $A \rightarrow u$ is a rule in R .
 - (b) if top of stack is terminal $a \in \Sigma$ and current input symbol is a , then pop.
 - (c) if top of stack is Z , then accept.

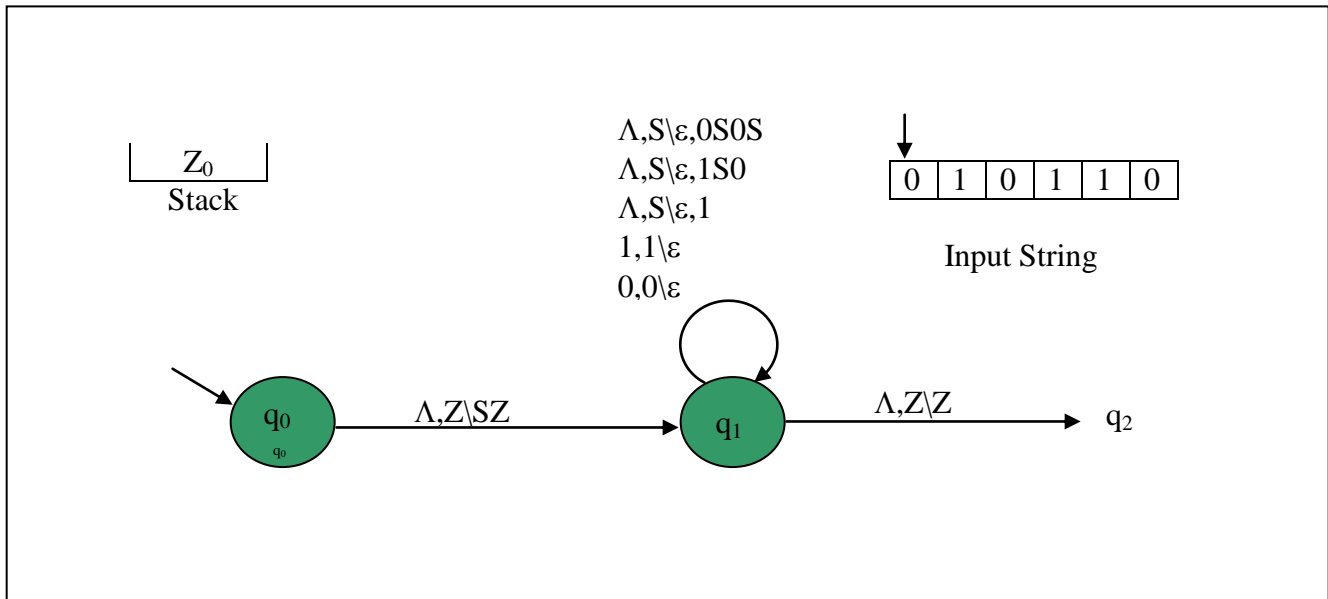
Recall CFG $S \rightarrow 0S0S0 \mid 1S0 \mid 1$
Corresponding PDA



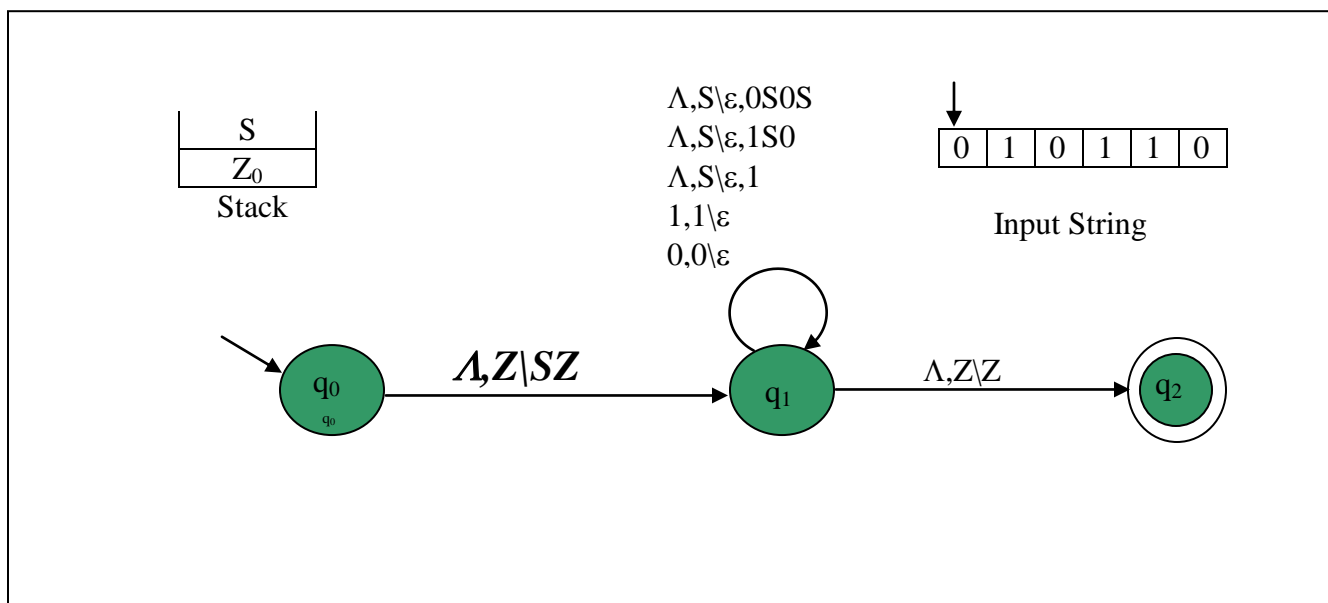
- * PDA is non-deterministic
- * Input alphabet of PDA is the terminal of CFG $\Sigma = \{0, 1\}$
- * Stack alphabet of PDA consists of all variables, terminals and Z
 $\Gamma = \{S, 0, 1, Z\}$
- * PDA simulates a leftmost derivation using CFG

Let's now process string 010110 on PDA

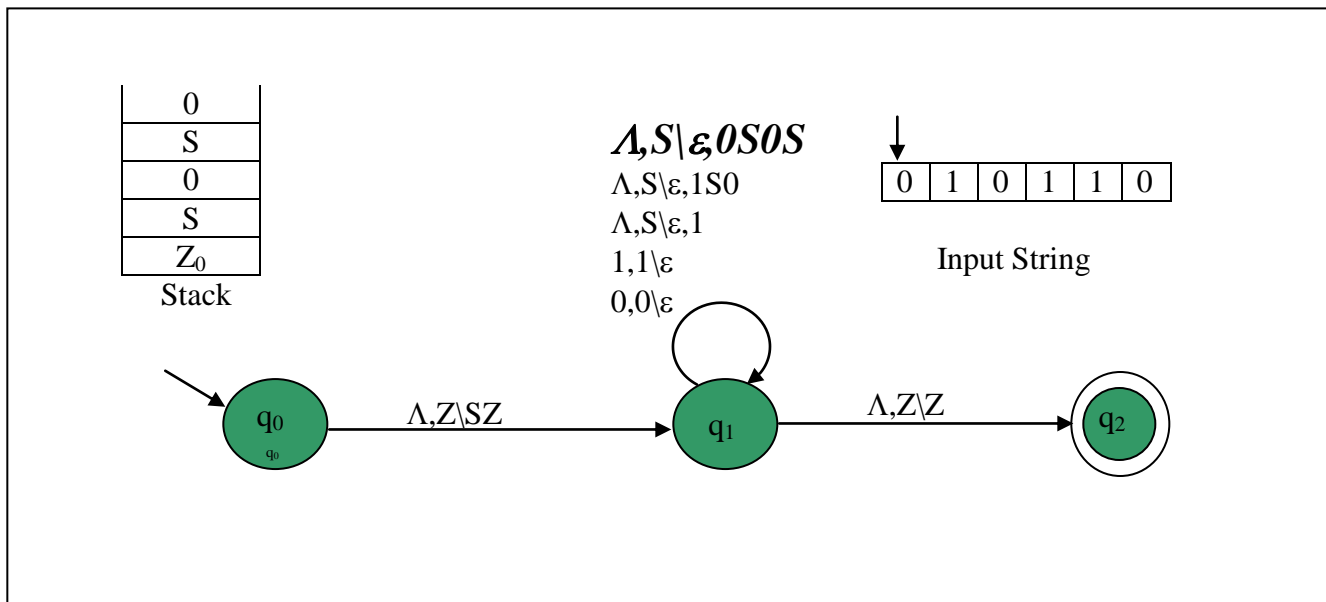
0- start in state q_0 with 010110 on input tape and empty stack



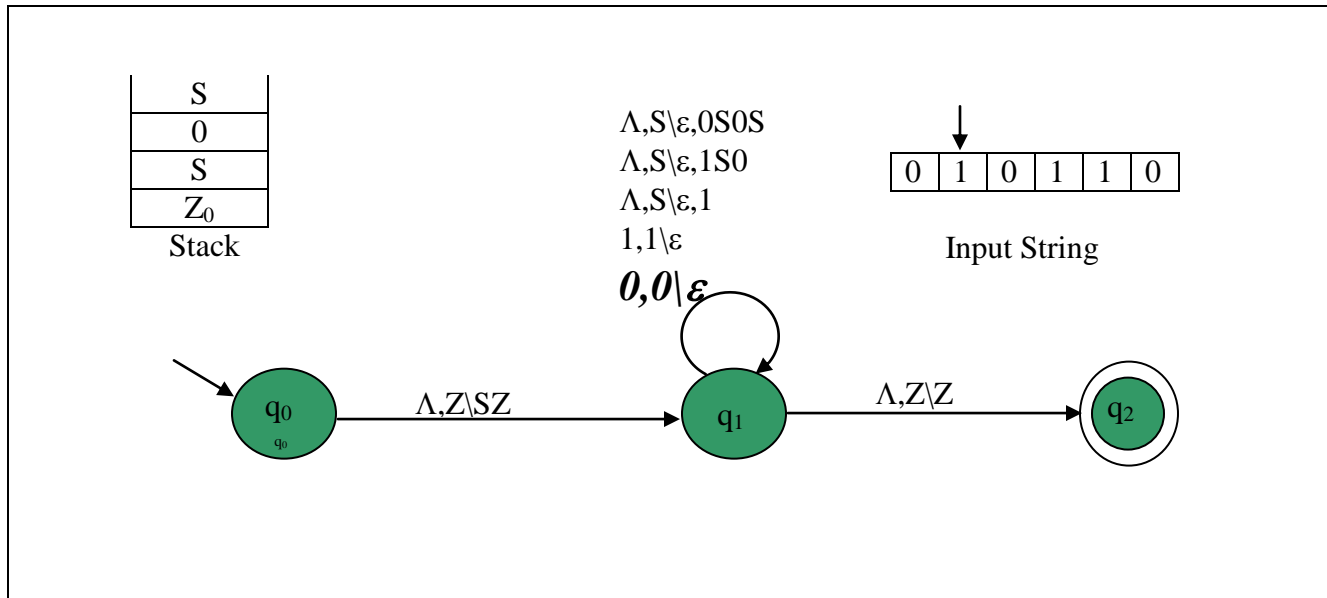
1- read nothing, pop nothing, push S and move to q_1



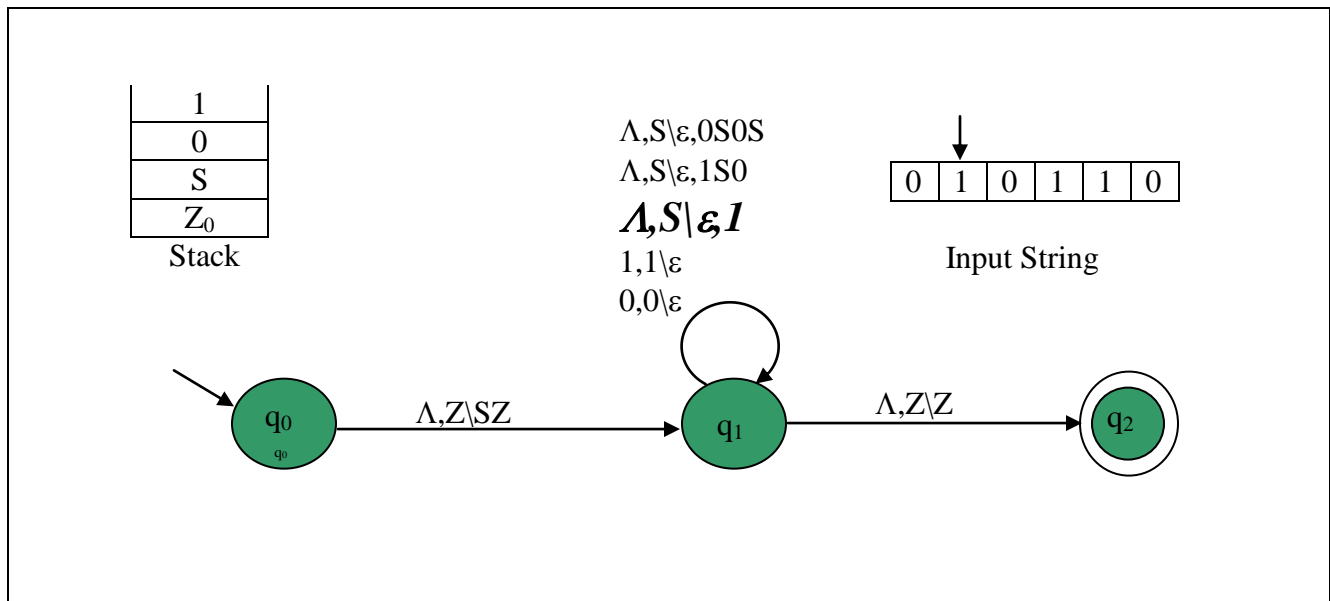
2- read nothing, pop S , push 0S0S, and return to q1



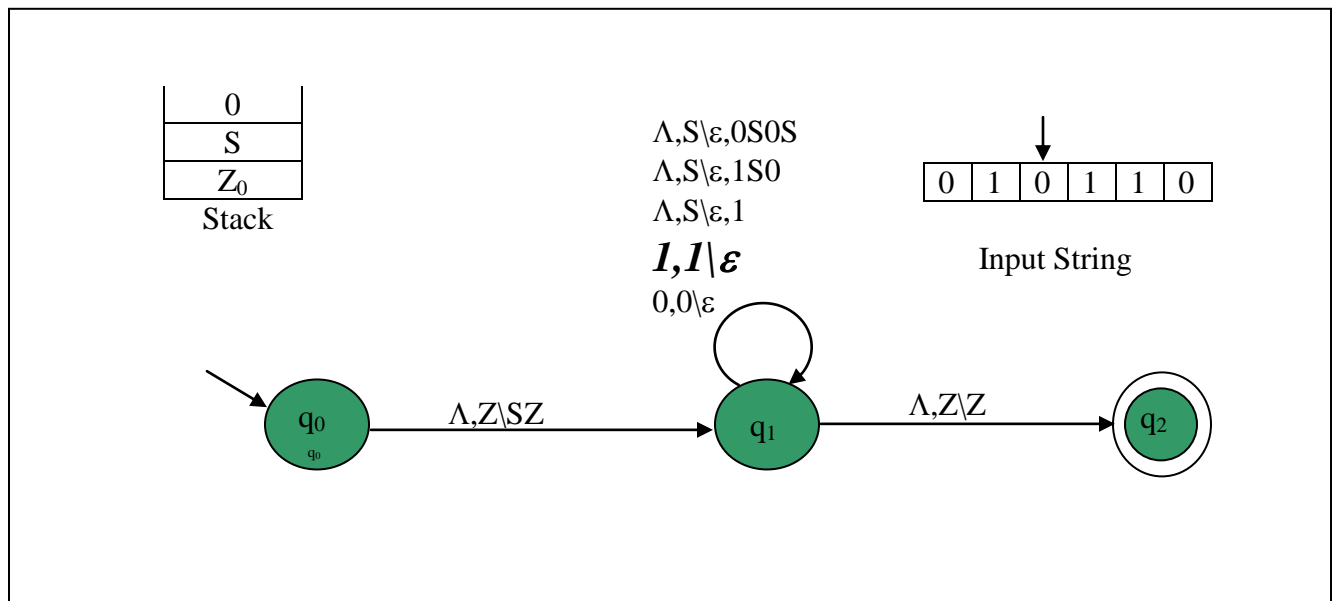
3- read 0, pop 0, push nothing, and return to q1



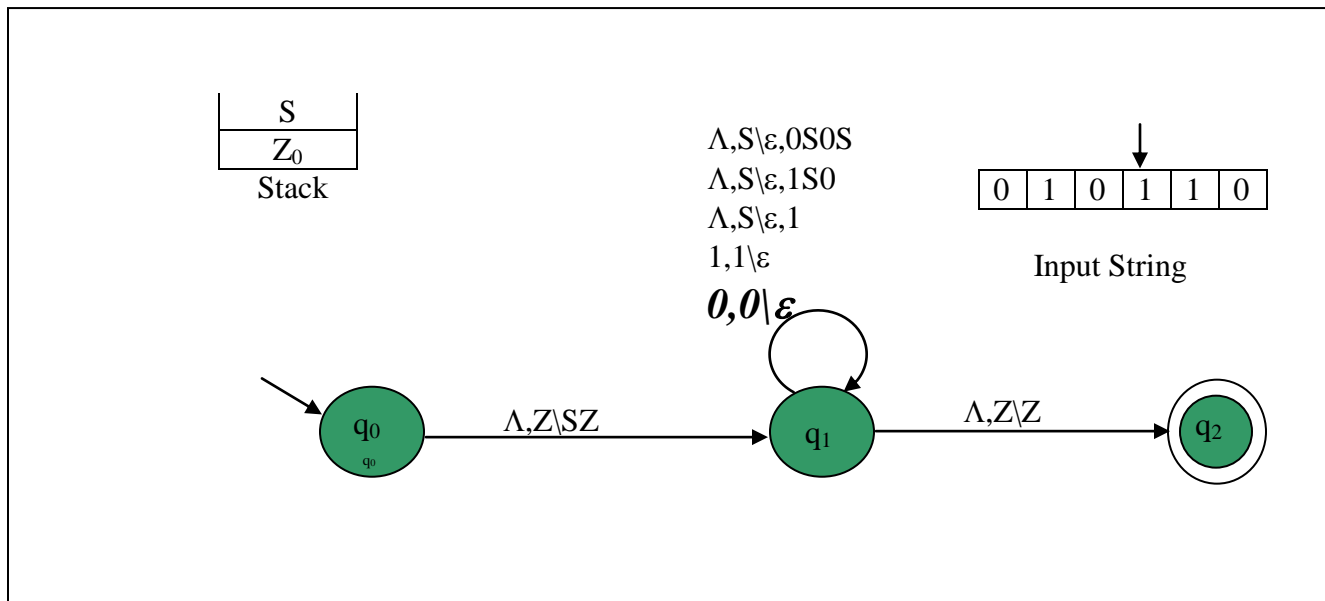
4- read nothing, pop S, push 1, and return to q1



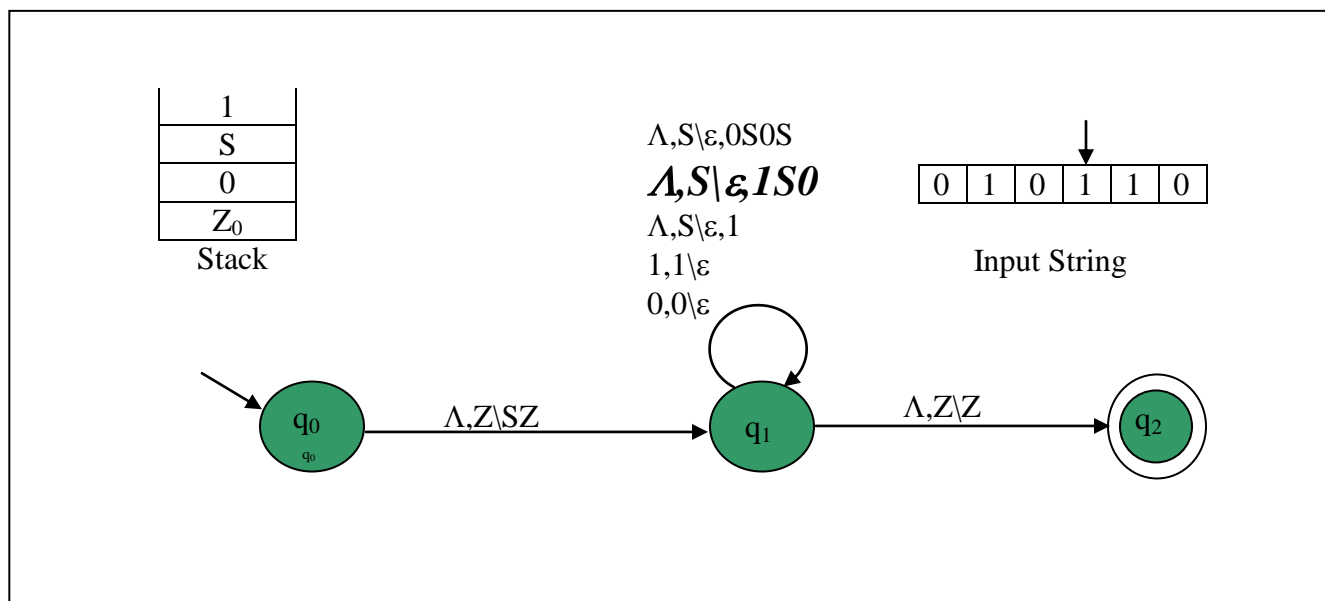
5- read 1, pop 1, push nothing and return to q1



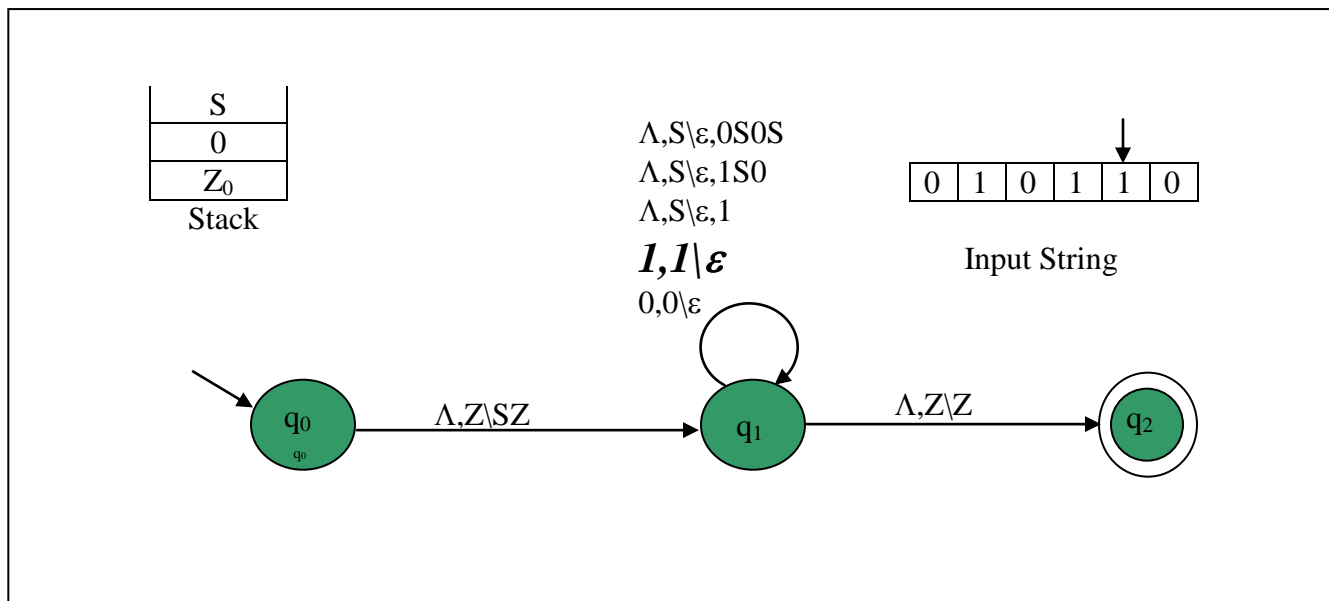
6- read 0, pop 0, push nothing, and return to q1



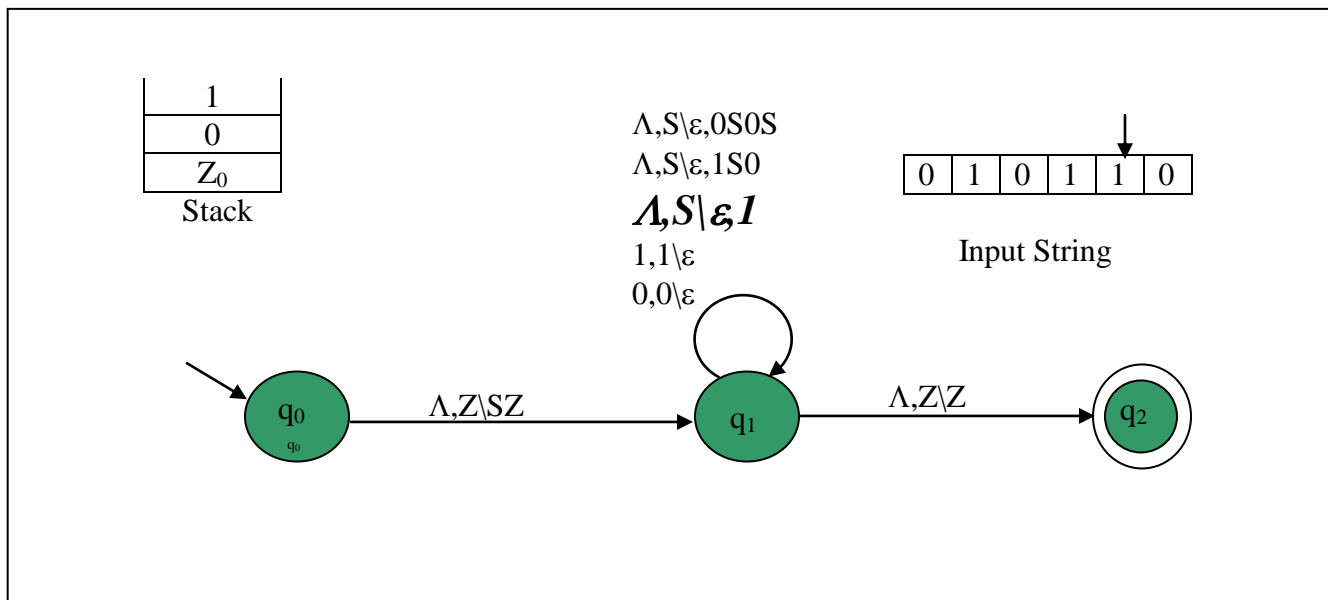
7- read nothing, pop S, push 1S0, and return to q1



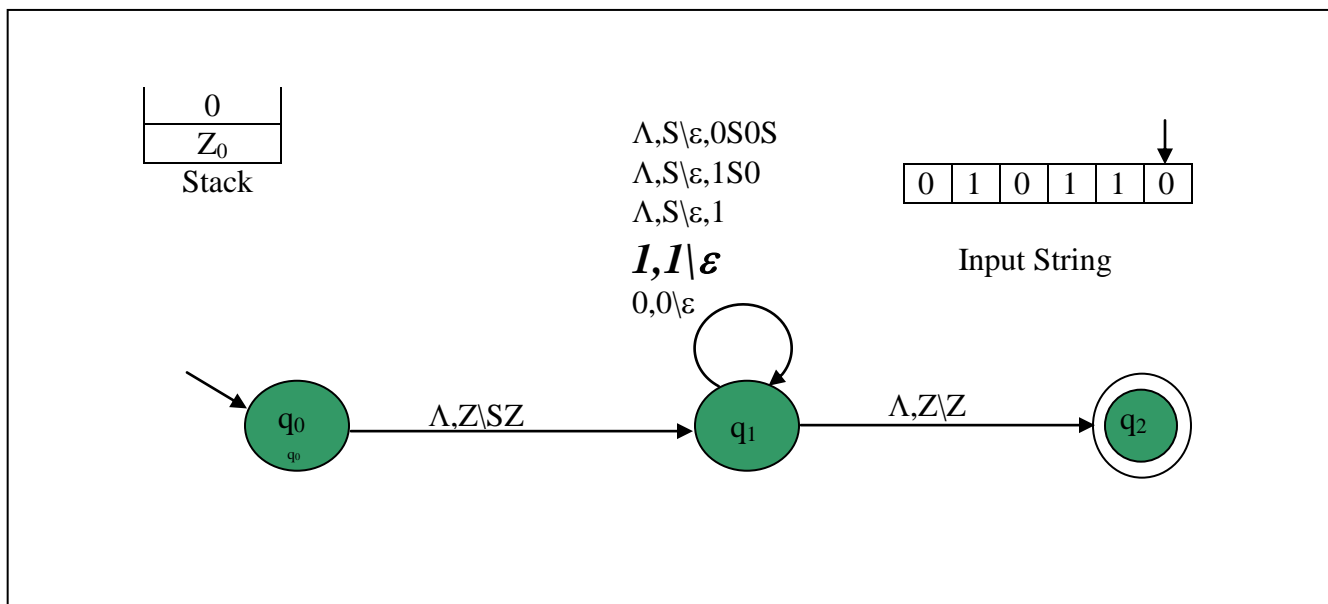
8- read 1, pop1, push nothing, and return to q1



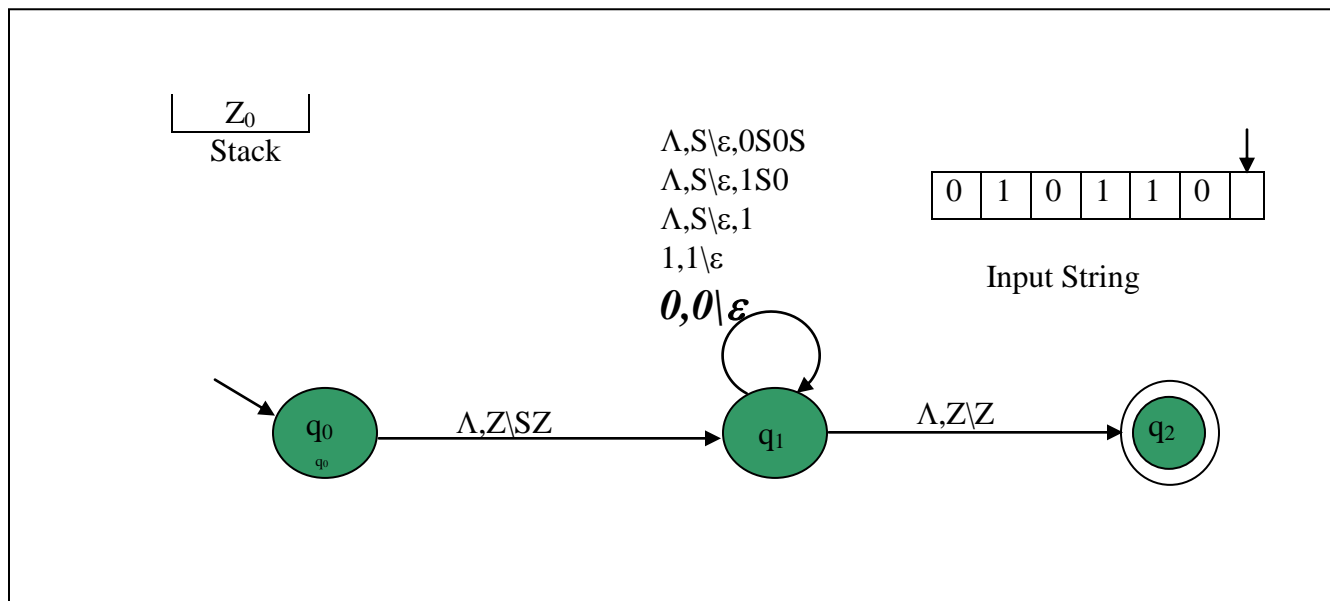
9- read nothing, pop S, push 1, and return to q1



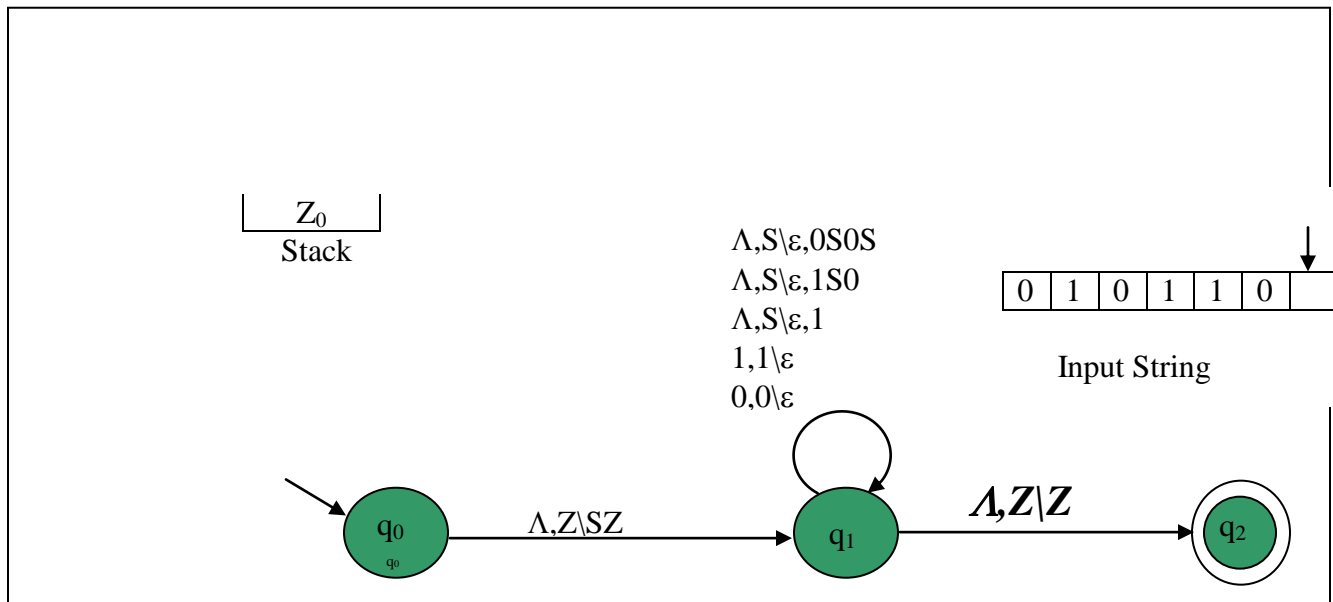
10- read 1, pop1, push nothing, and return to q1



11- read 0, pop0, push nothing, and return to q1



12- read nothing, pop nothing, push nothing move to q2 and accept



H.W

Convert CFG below to PDA

$S \rightarrow XSX / aY$

$X \rightarrow Y / S$

$Y \rightarrow b$

My Best Wishes
Mohamed U. Mahdi