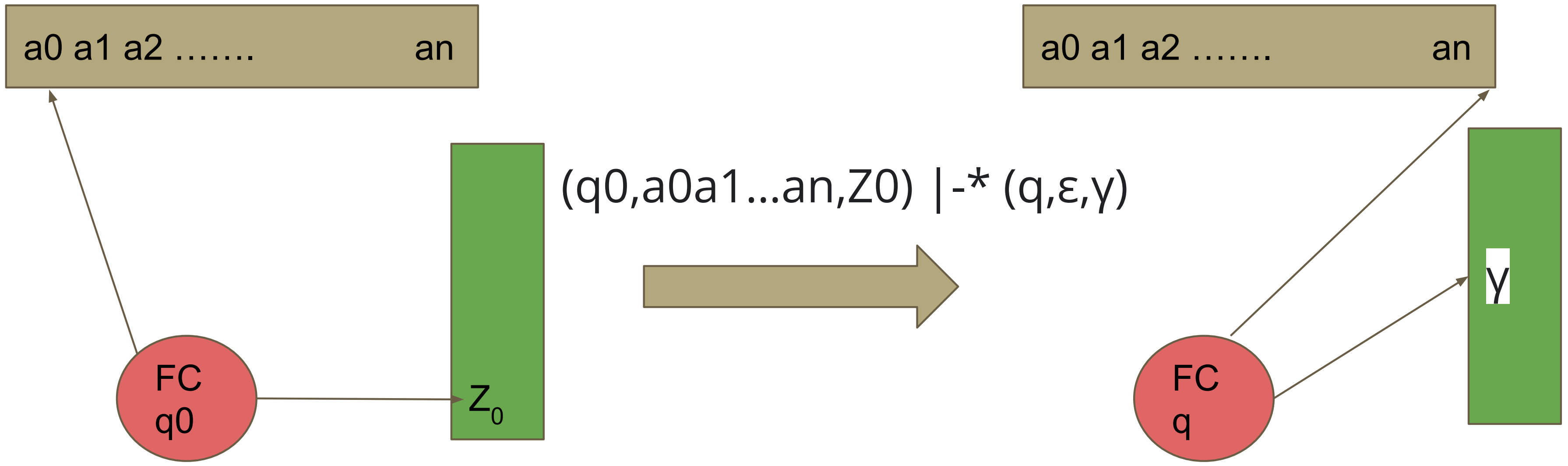
Acceptance of a string by PDA

PDA starts processing with initial ID. It makes state transition from one state to another state and its ID changes from one ID to another. It can accept input in two ways

1. Acceptance by final state
2. Acceptance by empty store



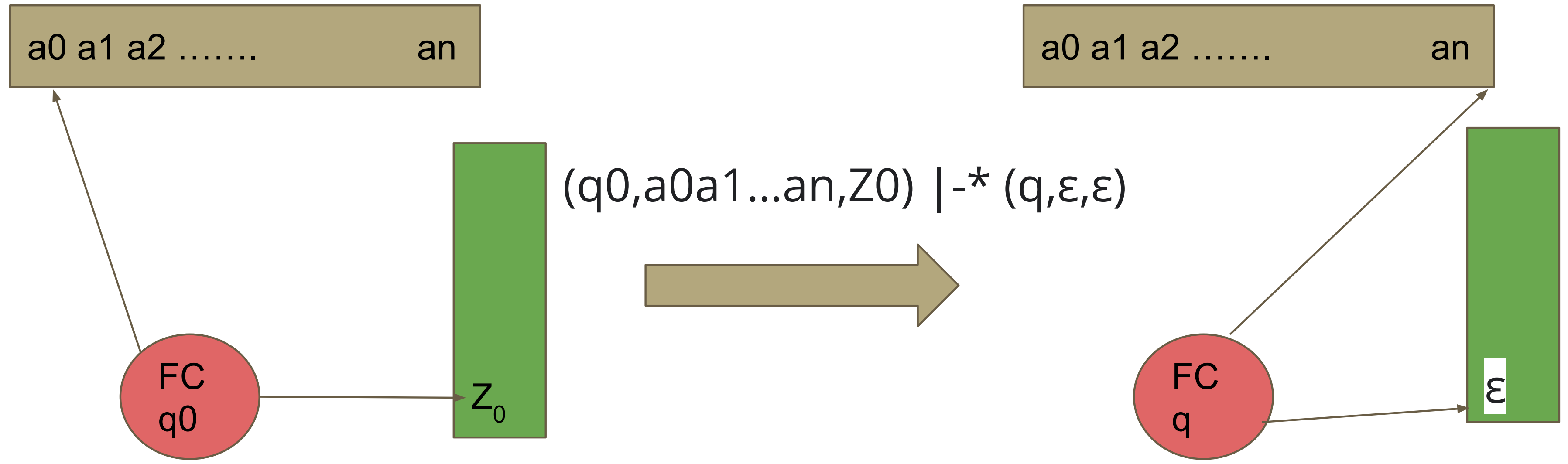
(q0,a0a1…an,Z0) |-* (q,ε,γ)

Acceptance by Final state - PDA starts with Initial ID (q0,a0a1…an,Z0) and keeps on making state transitions reading all the input one by one and finally reaches some ID (q,ε,γ). If q belongs to final state F, input w is accepted otherwise not.

Acceptance of a string by PDA

PDA starts processing with initial ID. It makes state transition from one state to another state and its ID changes from one ID to another. It can accept input in two ways

1. Acceptance by final state
2. Acceptance by empty store



Acceptance by Empty Store - PDA starts with Initial ID and keeps on making state transitions reading inputs one by one. If after reading entire input it reaches some ID such that stack becomes empty then input is accepted otherwise not.

Language accepted by PDA

It is defined as set of all strings w accepted by PDA M. It is defined in two ways

1. Language accepted by PDA (by final state)

   L(M)={w ε Σ*|(q0,w,zo)|-*(q,ε,γ) and q ε F} also denoted as T(M)

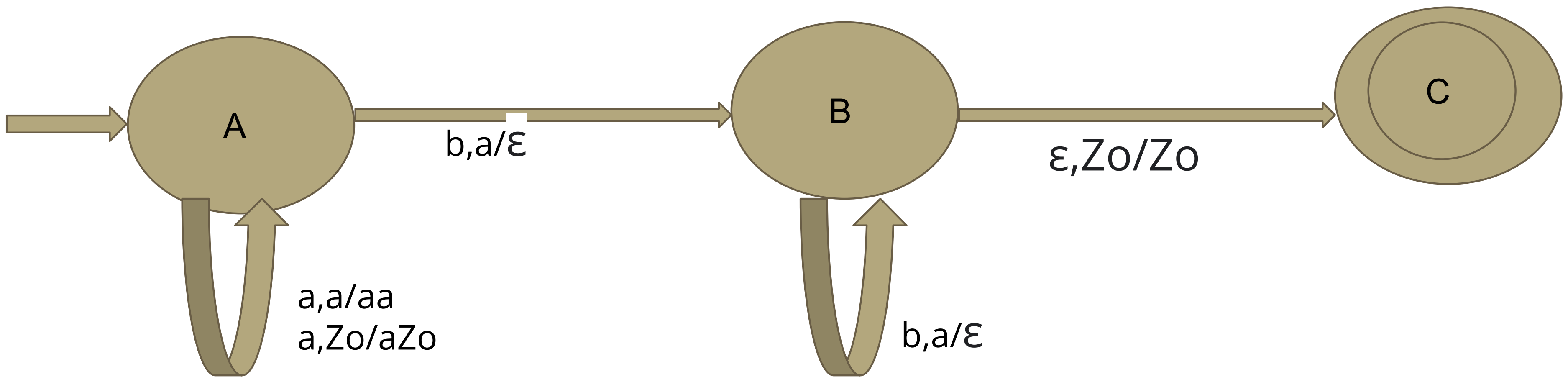
2. Language accepted by PDA (by empty store)

   L(M)={w ε Σ*|(q0,w,zo)|-*(q,ε,ε)} also denoted as N(M)

PDA Example 1 - PDA to accept $L=\{a^nb^n|n>=1\}$

Algorithm

1. Push a’s on to stack
2. Pop for b’s in input


A
b,a/Ɛ
B
ε,Zo/Zo
C
a,a/aa
a,Zo/aZo
b,a/Ɛ