Proof idea of L2

- Our task: given a PDA P that accepts by empty stack, design a CFG that generates $L(P)$.
- Fundamental event in a PDA's history: the popping of a symbol from the stack while consuming input.
- We design a grammar that will have variables named A_{pXq} such that $A_{pXq} \stackrel{*}{\Rightarrow} w$ *iff* $(p, w, X) \stackrel{*}{\vdash} (q, \epsilon, \epsilon)$, $\forall p, q, X$
- But to facilitate this task, we must make sure P has the following features:
  - It accepts by empty stack.
  - Each transition either pushes a symbol onto the stack or pops one off the stack, but not both at the same time.

Describing G's rules.

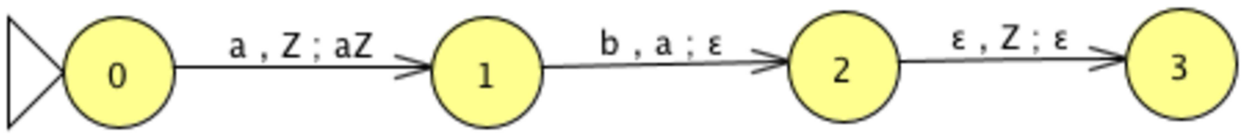
Given $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$, let's construct $G = (V, \Sigma, R, S)$:

1. V consists of the start symbol S and all symbols A_{pXq}, where $p, q \in Q$ and $X \in \Gamma$.
2. The production rules in R are:
   (a) For all states p, G has the production $S \to A_{q_0 Z_0 p}$. Therefore, these productions say: S will generate all strings w that cause P to empty its stack.
   (b) Let $\delta(q, a, X) = \{(r, Y_1 Y_2 \cdots Y_k), \cdots\}$, where a is either a symbol of Σ or ϵ, and $k \geq 0$.
      - if $k = 0$, i.e., $\delta(q, a, X) = \{(r, \epsilon), \cdots\}$, then $A_{qXr} \to a$.
      - if $k > 0$, then for all lists of states $r_1 r_2 \cdots r_k$, G has the production

$$A_{qX\underline{r_k}} \to a \; \underbrace{A_{rY_1\underline{r_1}}}_{\text{pops } Y_1} \; \underbrace{A_{\underline{r_1}Y_2\underline{r_2}}}_{\text{pops } Y_2} \cdots \underbrace{A_{\underline{r_{k-1}}Y_k\underline{r_k}}}_{\text{pops } Y_k}$$

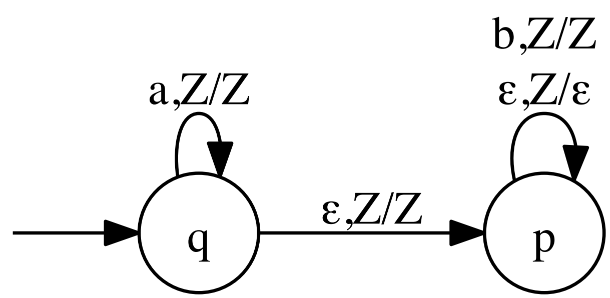
PDA to CFG conversion: example

To understand how the conversion takes place and have an intuition that the CFG defines the same language recognized by the PDA, let's see an example.



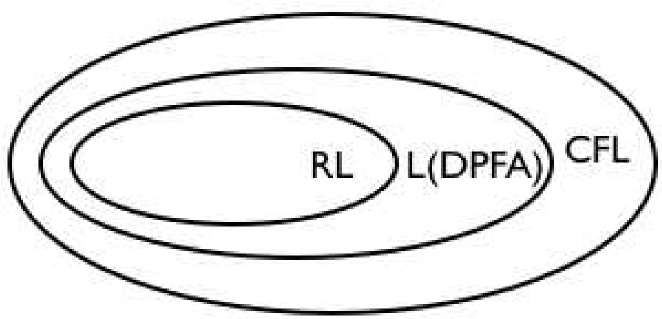
Exercise

Obtain a CFG for the following PDA:



Deterministic PDA

- Recall that DFAs and NFAs are equivalent in language recognition power.
- The same does not happen with PDAs.
- NPDAs are more powerful then DPDAs.
- DPDAs accept a class of languages that is between RLs and CFLs.
- Let's start by defining what is a DPDA. Then show that DPDA languages include all RLs.

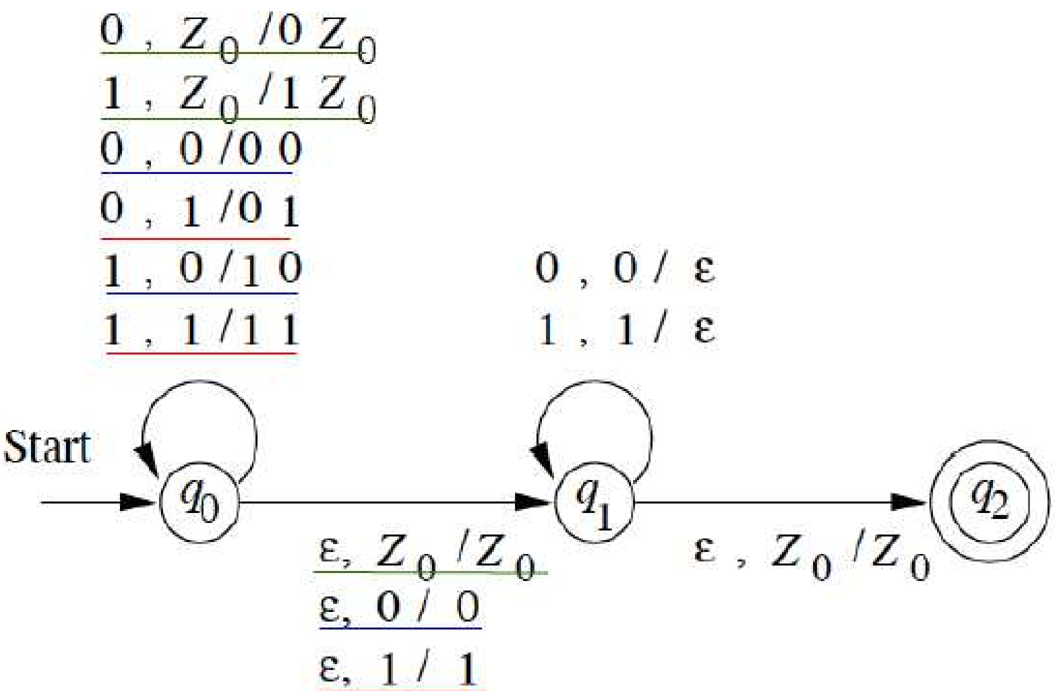


Deterministic PDA (DPDA)

A PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is deterministic iff:

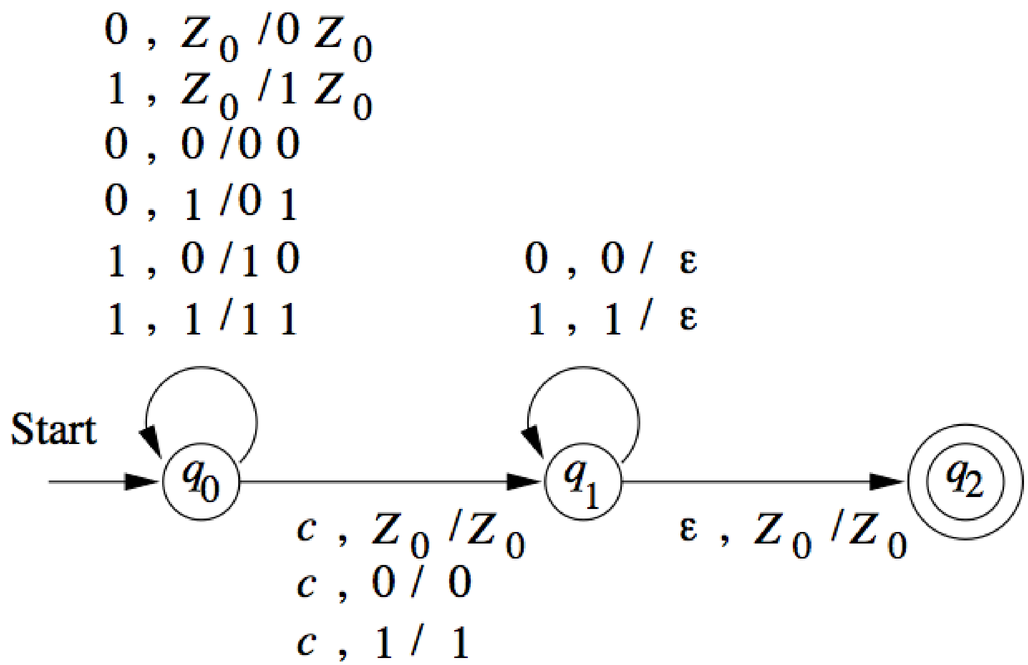
- $\delta(q, a, X)$ is always empty or a singleton.
- If $\delta(q, a, X)$ is nonempty, then $\delta(q, \epsilon, X)$ must be empty.

But before analyzing a DPDA, let's see the source of nondeterminism in the PDA for $\{ww^R : w \in \{0,1\}^*\}$



Example of a DPDA

Let us define the following language
$L_{wcwr} = \{wcw^R : w \in \{0,1\}^*\}$
Then L_{wcwr} is recognized by the following DPDA



Unlike in the previous example, all move choices for this PDA are deterministic.

$pda.1$

Exercise

Provide a DPDA for the following languages:

$$L = \{0^n 1^m : n \leq m\}$$

$$L = \{0^n 1^m : n \geq m\}$$