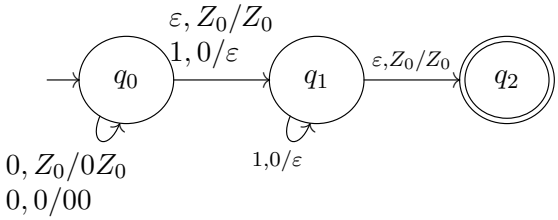
CS 360 - MODULE 6 - ADDITIONAL NOTES

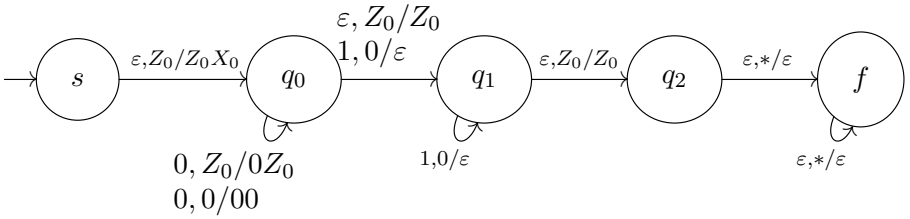
COLLIN ROBERTS

1. Example: From Final State to Empty Stack

Input:



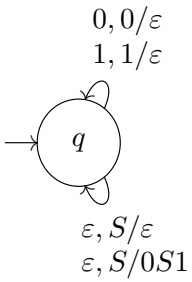
Output:



Exercise: Process $w_1 = 01$ (should be accepted) and $w_2 = 011$ (should be rejected) through both machines.

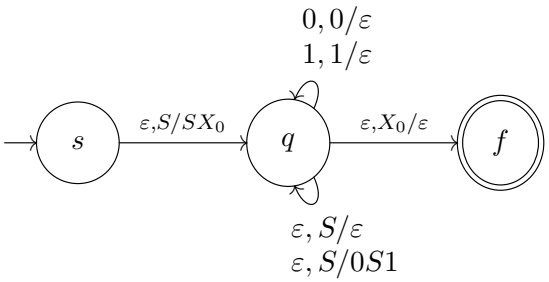
2. Example: From Empty Stack to Final State

Input:



$$\begin{aligned} \Sigma &= \{0, 1\} \\ \Gamma &= \{0, 1, S\}, \text{ with starting stack symbol } S. \end{aligned}$$

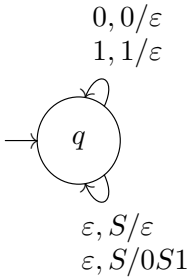
Output:



Exercise: Process $w_1 = 01$ (should be accepted) and $w_2 = 011$ (should be rejected) through both machines.

3. Example: From PDA to CFG

Input:



$$\begin{aligned} \Sigma &= \{0, 1\} \\ \Gamma &= \{0, 1, S\}, \text{ with starting stack symbol } S. \end{aligned}$$

Output: The variables are $[qSq], [q0q]$ and $[q1q]$.
From the given transitions, we obtain these productions:

- The transition $\delta(q, \varepsilon, S) = (q, \varepsilon)$ $(k = 0)$ gives the production $[qSq] \to \varepsilon$.
- The transition $\delta(q, \varepsilon, S) = (q, 0S1)$ $(k = 3)$ gives the production $[qSq] \to \varepsilon[q0q][qSq][q1q]$.
- The transition $\delta(q, 0, 0) = (q, \varepsilon)$ $(k = 0)$ gives the production $[q0q] \to 0$.
- The transition $\delta(q, 1, 1) = (q, \varepsilon)$ $(k = 0)$ gives the production $[q1q] \to 1$.
- Last, we add the production $S \to [qSq]$.

For simplicity of notation, retain S as the start variable and let

$$\begin{aligned} T &= [qSq] \\ U &= [q0q] \\ V &= [q1q]. \end{aligned}$$

Then our productions are

$$\begin{aligned} S &\to T \\ T &\to \varepsilon | \varepsilon UTV \\ U &\to 0 \\ V &\to 1, \end{aligned}$$

or more simply,

$$T \to \varepsilon | 0T1,$$

which is clearly a correct grammar for $\{0^i 1^i \mid i \geq 0\}$.

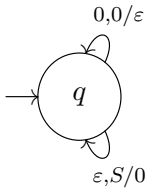
4. Questions From Class: 2015-11-03

4.1. Proving $N(P) \subseteq L(G)$ - Inducting on the Length of a Computation.

- We prove by induction on n for all $n \geq 1$ that

$$\text{For any variable } A, \text{ if } (q, w, A) \overset{n}{\underset{P}{\vdash}} (q, \varepsilon, \varepsilon), \text{ then } A \overset{*}{\underset{G}{\Rightarrow}} w.$$

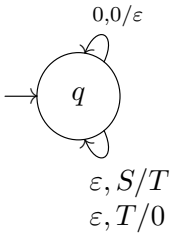
- You should convince yourself that the induction still holds together to give us what we want, even if there do not exist computations of any length n in any particular example.
- This example of two grammars generating $\{0\}$ and the corresponding constructed PDAs recognizing $\{0\}$ may shed some light on why the induction case of the proof works. Note that in both sub-examples, there do **not** exist any accepting computations of length $n = 1$.
  (1) Starting with the grammar $G : S \to 0$, we construct the PDA, P, with $\Sigma = \{0\}, \Gamma = \{0, S\}$ and starting stack symbol S:



  The only accepting computation is then

$$(q, 0, S) \vdash (q, 0, 0) \vdash (q, \varepsilon, \varepsilon).$$

  Because in the PDA, we must resolve and then match, we need two steps to complete the computation, even though the corresponding leftmost derivation only takes one step. Although we go into the induction case here, we do not actually invoke the induction hypothesis. The second configuration, $(q, 0, 0)$, has a terminal, rather than a variable at the top of stack. Hence the 'If Y_1 is a terminal' clause in the induction case of the proof handles the final step.
  (2) Starting with the grammar $G : S \to T; T \to 0$, we construct the PDA, P, with $\Sigma = \{0\}, \Gamma = \{0, S, T\}$ and starting stack symbol S:



  The only accepting computation is then

$$(q, 0, S) \vdash (q, 0, T) \vdash (q, 0, 0) \vdash (q, \varepsilon, \varepsilon).$$

  We go into the induction case here, and this time we do invoke the induction hypothesis. The second configuration, $(q, 0, T)$, has a variable at the top of stack. Note that as in the earlier part the 'If Y_1 is a terminal' clause in the induction case of the proof handles the final step.
- Also note that the question from class about "only recursing when needed" is actually buried in the proof, and is demonstrated in the example.
  – If Y_i is a terminal, then no recursion is required.

- If Y_i is a variable, then recursion is required, and the induction hypothesis kicks in.

5. Can DPDAs Accept by Empty Stack?

Yes, but the two notions of acceptance are not equivalent for DPDAs, as they are for PDAs. The following is excerpted from our textbook.

> If we want the DPDA to accept by empty stack, then we find that our language-recognizing capability is rather limited. Say a language L has the **prefix property** if there are no two different strings x and y in the language such that x is a prefix of y.
> ...there are some very simple languages that do not have the prefix property. Consider $\{0\}^*$, i.e., the set of all strings of 0's. Clearly, there are pairs of (different) strings in this language, one of which is a prefix of the other, so this language does not have the prefix property....
> Note that the language $\{0\}^*$ is a regular language. Thus, it is not even true that every regular language is $N(P)$ for some DPDA P. We leave as an exercise the following relationship:
> **Theorem 6.19:** A language L is $N(P)$ for some DPDA P if and only if L has the prefix property and L is $L(P')$ for some DPDA P'.