Pushdown automata:

Trace through instantaneous descriptions for one run of a PDA. (The last one we talked about Wed)

How does that PDA accept abaaba?

A PDA accepts w if

$$(q_0, w, z) \vdash^* (q', \lambda, u')$$

where $q' \in F$

So to say how our PDA accepts abaaba - we should trace some seq. of instant. descr. that leads to a final state:

instant descr	transition we used
$(q_0, abaaba, z)$	$(q_0, a, z) \to (q_0, Az)$
$\vdash (q_0, baaba, Az)$	$(q_0, b, A) \to (q_0, BA)$
$\vdash (q_0, aaba, BAz)$	$(q_0, a, B) \to (q_0 AB)$
$\vdash (q_0, aba, ABAz)$	

$\vdash (q_1, aba, ABAz)$

$\vdash (q_1, ba, BAz)$

$\vdash (q_1, a, Az)$

$\vdash (q_1, \lambda, z)$

$\vdash (q_2, \lambda, z)$

$(q_0, \lambda, A) \rightarrow (q_1, A)$

$(q_1, a, A) \rightarrow (q_1, \lambda)$

$(q_1, b, B) \rightarrow (q_1, \lambda)$

$(q_1, a, A) \rightarrow (q_1, \lambda)$

$(q_1, \lambda, z) \rightarrow (q_2, z)$

Since $q_2 \in F$, we accept ababa.

Equivalence of PDAs and CFGs

Goal: Prove that ~~a language is~~ Given a language L, there exists a PDA M that accepts L iff there exists a CFG G that generates L.

How to do this?

① Give a method for taking a CFG G and constructing a PDA M from G so that ~~the string~~ a string is accepted by M if and only if it is generated by G.

(2) Give a method for taking a PDA M and constructing a ~~gramma~~ CFG G so that a string can be gen by G if and only if it can be accepted by M.

Let's do (1):

We have a grammar G - step 0 is to find an equiv grammar G' in CNF.

We'll have the PDA mimic a leftmost derivation - in CNF, ~~the~~ anytime in a leftmost derivation, we have

$\underbrace{\quad\quad}_{\text{letters}}\underbrace{\quad\quad\quad\quad}_{\text{variables}}$ - the letters will be the part of the string already processed by the PDA, and the variables will be on the stack.

E.g. Grammar G is given by

$$S \rightarrow AB \mid AA$$
$$A \rightarrow SB \mid a$$
$$B \rightarrow BA \mid b$$

(leftmost) Derivation in the grammar:

$$S \rightarrow AA \rightarrow SBA \rightarrow ABBA$$
$$\rightarrow aBBA \rightarrow abBA \rightarrow abBAA \rightarrow abbAA$$
$$\rightarrow abbaA \rightarrow abbaa.$$

I want my PDA to accept the string abbaa through a process mimicking this derivation:

always have this transition

$(q_0, abbaa, z)$ $(q_0, \lambda, z) \rightarrow (q_1, Sz)$

$\vdash (q_1, abbaa, Sz)$ comes from $S \rightarrow AA$ $(q_1, \lambda, S) \rightarrow (q_1, AA)$

$\vdash (q_1, abbaa, AAz)$

$\vdash (q_1, abbaa, SBAz)$ $A \rightarrow SB$ $(q_1, \lambda, A) \rightarrow (q_1, SB)$

$\vdash (q_1, abbaa, ABBAz)$ $S \rightarrow AB$ $(q_1, \lambda, S) \rightarrow (q_1, AB)$

$\vdash (q_1, \cancel{a}bbaa, BBAz)$

$\vdash (q_1, baa, BAz)$

$\vdash (q_1, baa, BAAz)$

$\vdash (q_1, aa, AAz)$

$\vdash (q_1, a, Az)$

$\vdash (q_1, \lambda, z)$

$\vdash (q_2, \lambda, z)$

$A \to a$ $(q_1, a, A) \to (q_1, \lambda)$

$B \to b$ $(q_1, b, B) \to (q_1, \lambda)$

$B \to BA$ $(q_1, \lambda, B) \to (q_1, BA)$

$(q_1, \cancel{b}, B) \to (q_1, \lambda)$

$(q_1, a, A) \to (q_1, \lambda)$

$(q_1, a, A) \to (q_1, \lambda)$

$\boxed{(q_1, \lambda, z) \to (q_2, z)}$

always have this one

~~Formally~~ Our construction algorithm for getting our PDA from our CFG:

M always has 3 states: $\{q_0, q_1, q_2\}$

$F = \{q_2\}$. $\Sigma = \Sigma$ for G, $\Gamma = V$ for G $\cup \{z\}$

The transitions in M are:

$(q_0, \lambda, z) \to (q_1, Sz)$

$(q_1, \lambda, z) \to (q_2, z)$

For each production

$$A \to BC$$

in G, we have

$$(q_1, \lambda, A) \longrightarrow (q_1, BC)$$

For each production

$$A \to a$$

in G, we have

$$(q_1, a, A) \longrightarrow (q_1, \lambda)$$