Deterministic PDA

10/19

A PDA $M = (Q, \Sigma, \Gamma, q_0, F, z, \delta)$ is <u>deterministic</u> if (1) Given a state q, a letter ℓ, and a stack symbol A, there is at most one transition

$$(q, \ell, A) \longrightarrow (q', B) \in \delta$$

$$(q' \in Q,\ B \in \Gamma^*)$$

&

(2) Given a state q, and a stack symbol A, if there is a transition

$$(q, \lambda, A) \longrightarrow (q', B) \in \delta,$$

then there does not exist any transition

$$(q, \ell, A) \longrightarrow (q'', B') \in \delta$$

for any $\ell \in \Sigma$.

$L = \{ww^R \mid w \in \{a,b\}^*\}$

$(S, a, z) \longrightarrow (S, az)$
$(S, b, z) \longrightarrow (S, bz)$
$(S, a, a) \longrightarrow (S, aa)$
$(S, b, a) \longrightarrow (S, ba)$
$(S, a, b) \longrightarrow (S, ab)$
$(S, b, b) \longrightarrow (S, bb)$

In state S, push whatever we read onto the stack

$(S, \lambda, z) \longrightarrow (T, z)$
$\cancel{(S, \lambda, a) \longrightarrow}$

I had to handle empty string

$(S, a, a) \longrightarrow (T, \lambda)$
$(S, b, b) \longrightarrow (T, \lambda)$
$(T, a, a) \longrightarrow (T, \lambda)$
$(T, b, b) \longrightarrow (T, \lambda)$

pull ~~stuff~~ letter of stack if it matches input

$(T, \lambda, z) \longrightarrow (F, \lambda)$

done when nothing on stack & nothing in string.

F is final state.

It turns out that it is impossible to construct a det. PDA that accepts L.

<u>Note</u>: Being deterministic has nothing to do with being ambiguous:

L does have an unambiguous grammar.

$$S \to aSa \mid bSb \mid \lambda.$$

It's hard to characterize grammars that correspond to deterministic PDAs (there is no short statement)

A ~~language~~ context-free language that I can argue is not accepted by a det PDA:

$$L = \{a^n b^n \mid n \in \mathbb{Z}_{>0}\} \cup \{a^n b^{2n} \mid n \in \mathbb{Z}_{>0}\}$$

① Why is this CF?

$$S \to A \mid B$$
$$A \to aAb \mid ab$$
$$B \to aBbb \mid abb$$

is a CFG for it.

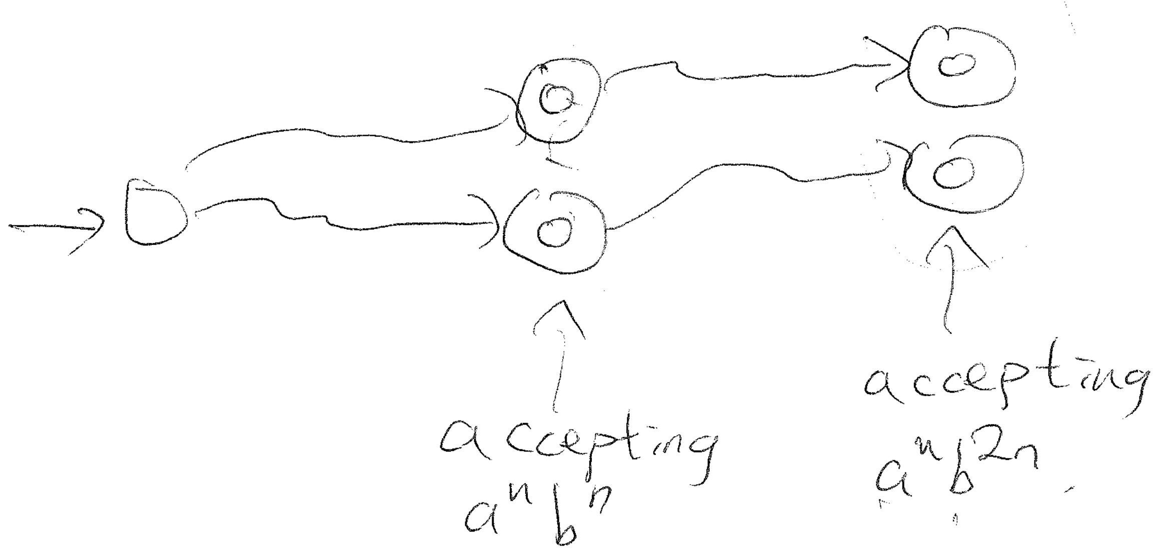
② ~~A~~ Why no det PDA?

Consider $L' = \cancel{\{a^n b^n} \; L \cup \{a^n b^n c^n \mid n \in \mathbb{Z}_{>0}\}$

If there is det PDA for L, then there must be a nondet PDA for L'. But we will see L' is not CF.

Think about a det PDA: M for L:

~~a~~ paths through it look like



Construct M' by:

Make 2 copies of M.

In the extra copy (Copy 2), change ~~b~~ b-transitions to c-transitions.

Connect the $a^n b^n$ accepting states to their analogous ~~a~~ states in Copy 2 by λ-transitions.

Since M accepts $a^n b^{2n}$, it goes through these $a^n b^n$-accepting states in processing $a^n b^{2n}$, so M' goes through these states for $a^n b^n c^n$, and then uses λ-transition to Copy 2 to accept $a^n b^n c^n$.

It takes a little thought to make sure

1) This doesn't accept anything else

2) I can ~~tell~~ distinguish the $a^n b^n$-accepting states from $a^n b^{2n}$-accepting states.