Deterministic Pushdown Automata (DPDA)

PDA = NPDA $((p, a, A), (q, \beta))$

DPDA → PDA with two restrictions

(1) DPDA at every situation has a unique transition to follow

(2) A DPDA cannot get stuck.

1. The end-of-input is marked by a special symbol $\dashv$

2. The bottom marker $\perp$ always stays in the stack. $((p, a, \perp), (q, \beta\perp))$, $\beta \in \Gamma^*$

3. $\forall\, p, a, A$ there is a unique transition $((p, a, A), (q, \beta))$.
   If not, there must be a unique transition $((p, \epsilon, A), (q, \beta))$

4. A DPDA accepts by final state.
   [No explicit concept of reject state] → DPDA must consume the entire input including $\dashv$ and be in a final state.

x is accepted $\Longleftrightarrow$ $(s, x\dashv, \perp) \xrightarrow{*} (q, \epsilon, \beta\perp)$ for some $q \in F$ and some $\beta \in \Gamma^*$.

L is called a deterministic context-free language (DCFL) if $L = \mathcal{L}(D)$ for some DPDA D.

L is DCFL $\Rightarrow$ L is CFL

Converse is not necessarily true.

$\exists$ languages that are CFL but not DCFL.

- DCFL are closed under complement ✓
- CFL are <u>not</u> closed under complement.

$$\mathrm{DCFL} \subsetneq \mathrm{CFL}$$

Non-determinism implies better language recognition capabilities.

$$L = \left\{ a^n b^n \mid n \geq 0 \right\}$$

$$\Sigma = \{a, b\}$$
$$\Gamma = \{A, R, \perp\}$$

reject

$\dashv$ must be handled as input.

$a, A/RA$
$a, \perp/R\perp$ } a came after a b (aaba, aba, ba)

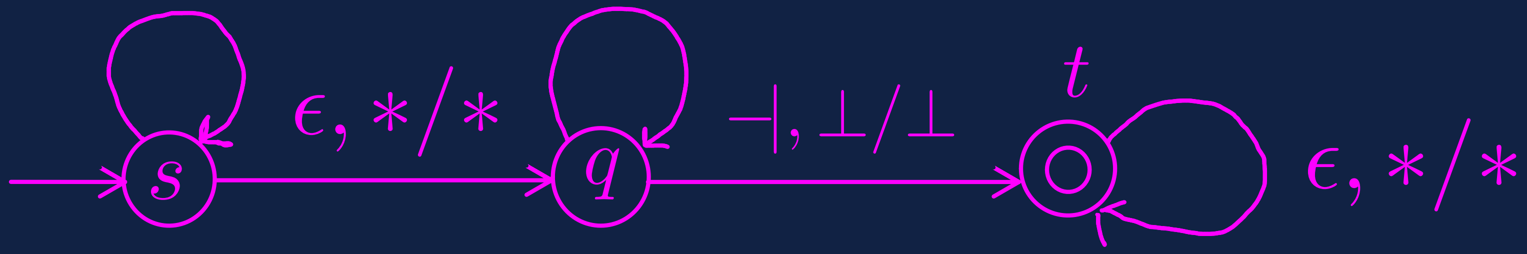
$b, A/\epsilon \longrightarrow$ earlier

$b, \perp/R\perp \longrightarrow$ more b's than a's (abb)

$\dashv, A/RA \longrightarrow$ more a's than b's (aab)

$\epsilon, R/R$

$a, \perp/A\perp$
$a, A/AA$



Switch final and non-final states.

This construction will not work.

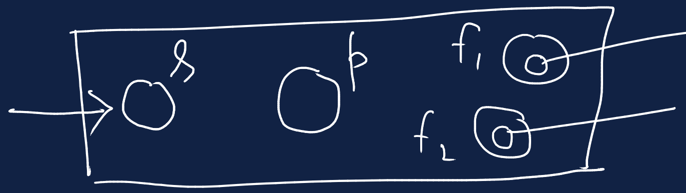
- If M accepts, M consumes $x\dashv$ and is in a final state.
- If M rejects, it may loop without consuming the entire $x\dashv$
  - To detect and remove looping behavior which may happen after/before reading $\dashv$.
  - Note whether $\dashv$ is read or not.
- Add a reject state r. The augmented machine loops in the state r <u>after</u> consuming $x\dashv$.

1. Handle seeing the marker $\dashv$

2. Handle all loops except at the explicit accept and reject states.

__Step 1__ :

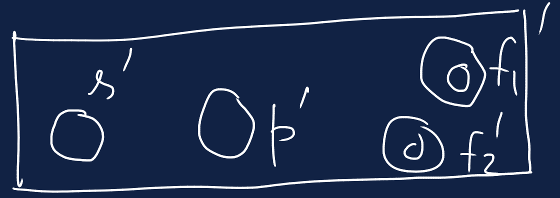
M



no longer final

Q

M'



Q'

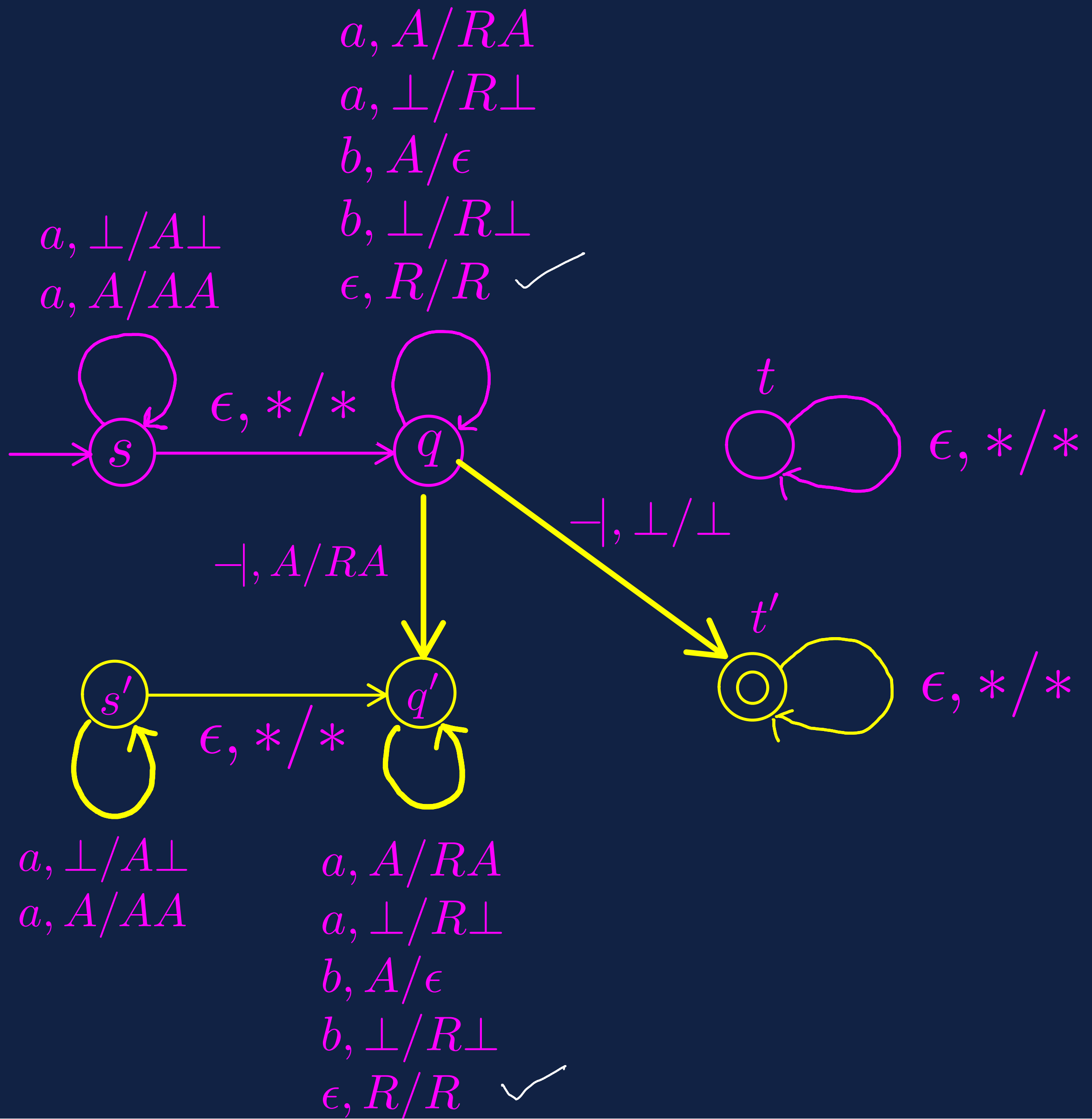
$F' = \{ f' \mid f \in F \}$

All transitions in M stay in M' except

replace $((p, \dashv, A), (q, \beta))$

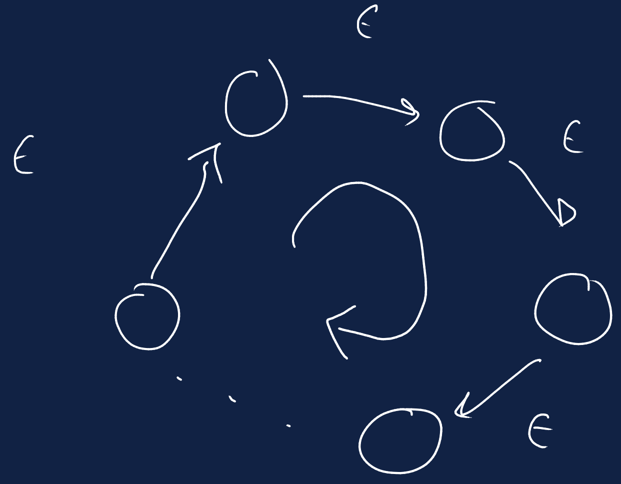
by $((p, \dashv, A), (q', \beta))$

Handling the end of input



Handling loops

(p, ϵ, A)

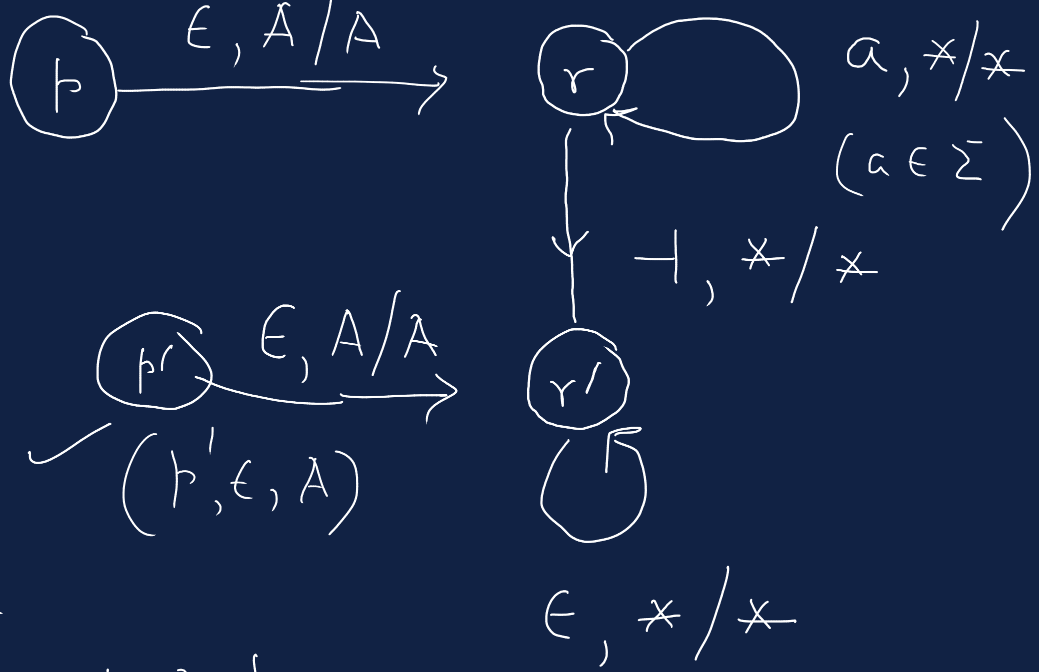


infinite loop

$\Rightarrow$ All the transitions must be ϵ

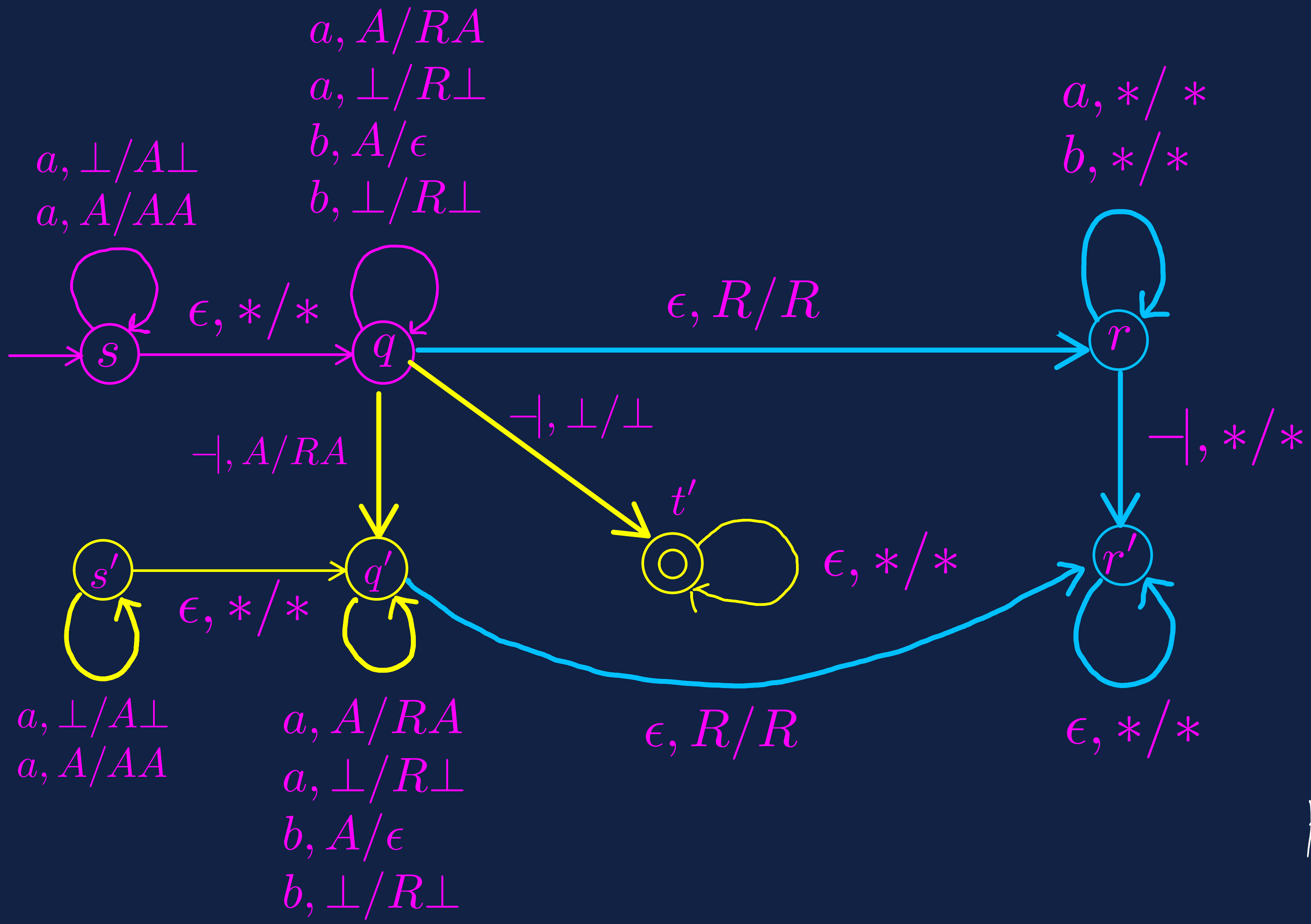
$\Rightarrow$ stack cannot shrink

(p, ϵ, A) may lead to looping (p', ϵ, A)



r' is the only state for looping

Handling infinite loops



Switch the roles of r' and t'

Infinite looping is possible in only these two states.

DCFL are <u>closed under complement</u> ✓

$L = \{ ww \mid w \in \{a, b\}^* \}$ is not CF

$\sim L = \{a, b\}^* \setminus L$ is CF

$\sim L$ is CFL but not DCFL.

DCFL are not closed under intersection ✗, union ✗, reversal ✗

$$\{a^m b^m c^n\} \cap \{a^m b^n c^n\} = \{a^n b^n c^n\}$$

$$L = \{a^m b^n c^k \mid m \neq n\} \cup \{a^m b^n c^k \mid n \neq k\}$$

$(\sim L) \cap \mathcal{L}(a^* b^* c^*) = \{a^n b^n c^n\}$ is not even CF.

$\{b a^m b^n c^k \mid m \neq n\} \cup \{c a^m b^n c^k \mid n \neq k\}$ is not DCFL upon reversal.