Lecture 12

- HW 2 due tonight

- HW 1 - graded (see gradescope)

  * Submissions for regrading of HW1 due by Oct 25 11:59 pm

  Note: requesting a remark could make your mark go up or down or stay unchanged.

Min:	18 / 60
Median:	49 / 60
Max:	60 / 60
Mean:	47.21 / 60

Solutions to HW1 are posted

Lecture 12

- HW 2 due tonight

- HW 1 - graded (see gradescope)

  * Submissions for regrading of HW1 due by Oct 26

  Note: requesting a remark could make your mark go up or down or stay unchanged.

- Review of CFL's and Solns to extra problems: posted

- Review session for Test 1: Saturday Oct 21, 1-2pm CSB 451

  Review Problems will be posted Tuesday

- Today: Equiv. between CFG + PDA

Context-Free Languages + PushDown Automata

Now we will define a larger class of languages that includes all regular languages plus new ones.

✓ 1 We will first define CFL's to be those languages accepted by PUSHDOWN AUTOMATA (PDA)

✓ 2 Then we give an alternative characterization of CFLs Language/generation Model: Context Free grammars (CFGs)

TODAY! → 2a We will prove these 2 characterizations are equivalent:

$$\underbrace{\text{PUSHDOWN AUTOMATA (PDA)}}_{\text{Machine Model}} \equiv \underbrace{\text{Context Free grammars (CFGs)}}_{\text{Language/generation Model}}$$

✓ 3 Pumping Lemma (for CFL's): used to prove that some languages are Not Context Free Languages

2a Equivalence of PDA and CFG's

Theorem 1 If L has a CFG then there exists a PDA M accepting L.

Theorem 2 If L is accepted by a PDA, then L has a CFG.

CFG ⇒ PDA

<u>Theorem 1</u> If L has a CFG then there exists a PDA M accepting L.

<u>Proof sketch</u>: Let $\mathcal{G}$ be a CFG. We show how to convert $\mathcal{G}$ into a PDA, $M_{\mathcal{G}}$. such that the language $L(\mathcal{G})$ generated by $\mathcal{G}$ $= L(M_{\mathcal{G}})$

Informal description of $M_{\mathcal{G}}$ on input w:

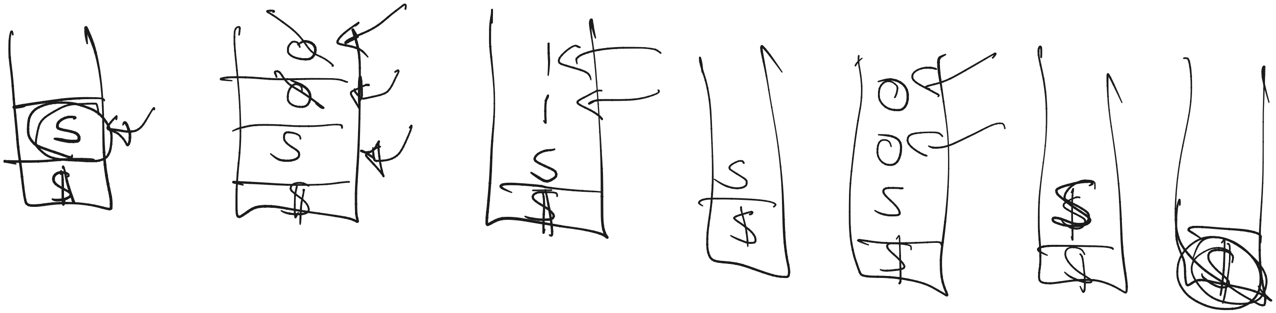
- Put '$' on stack
- Put start symbol, S, of $\mathcal{G}$ on stack
  - Repeat (until all of w is processed)
  - If top of stack is a terminal symbol $a \in \Sigma$, read Next symbol from w, if they don't match reject
  - If top of stack is a variable symbol $A \in \Gamma$ nondeterministically guess a rule for A, say $A \to u$ and substitute A on the stack by u (in reverse order)
  - If top of stack is '$' enter <u>accept</u> state.

Say w is generated by $G = (V, \Sigma, R, S)$

$S \to 00S \mid 11S \mid \varepsilon$

$W = 001100$

$S \to 00S \longrightarrow 0011S \longrightarrow 001100S \longrightarrow 001100$



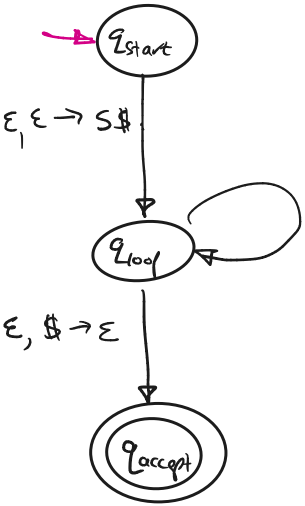
$\mathcal{G} \Rightarrow$ PDA

Let $\mathcal{G} = (\Sigma, \Gamma, S, R)$

$M_{\mathcal{G}}$: States are $Q = \{q_{start}, q_{loop}, q_{accept}\} \cup E$

start state

accept state


q_{start}
$\varepsilon, \varepsilon \to S\$$
q_{loop}
$\varepsilon, \$ \to \varepsilon$
q_{accept}

$\varepsilon, A \to u$ for every rule $A \to u$

$a, a \to \varepsilon$ for terminal $a \in \Sigma$

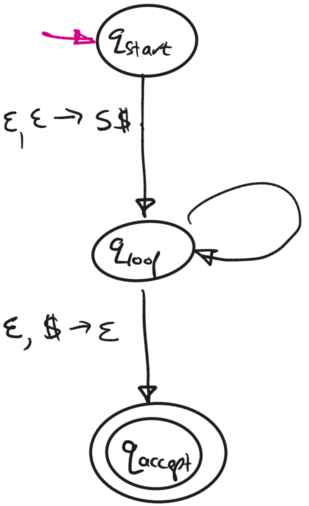
CFG ⇒ PDA

Let $G = (\Sigma, \Gamma, S, R)$

M_G : States are $Q = \{q_{start}, q_{loop}, q_{accept}\} \cup E$

start state ↗ q_{start}

accept state ↑ q_{accept}



q_{start}
$\varepsilon, \varepsilon \to S\$$
q_{loop}
$\varepsilon, \$ \to \varepsilon$
q_{accept}

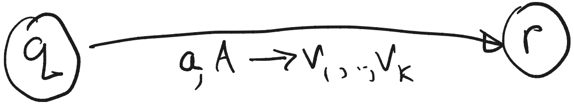
abbreviation (see next slide)

$\varepsilon, A \to w$ for every rule $A \to w$

$a, a \to \varepsilon$ for terminal $a \in \Sigma$

CFG ⇒ PDA

Shorthand Notation



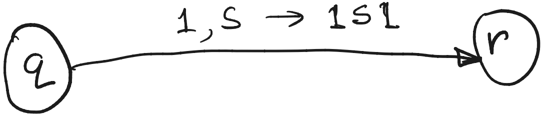
means if in state q, next input symbol read is a, and s is top symbol on stack we pop s, push $v_1 \ldots v_k$ on stack and move to state r

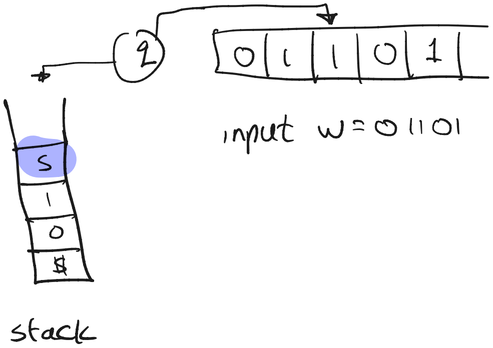
$a \in \Sigma \cup \varepsilon$

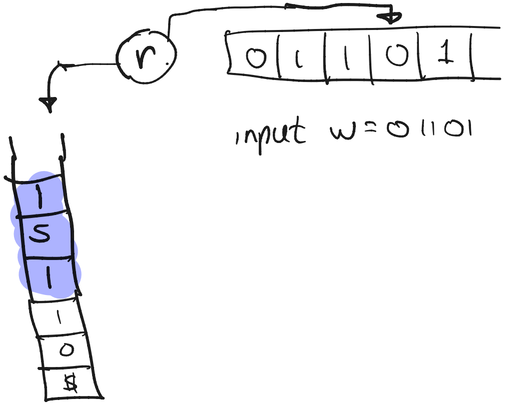
$A \in \Sigma \cup \Gamma \cup \varepsilon$

$v_1, \ldots, v_k \in \Sigma \cup \Gamma \cup \varepsilon$

Example

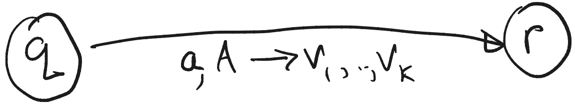






CFG ⇒ PDA

Implementing this transition:

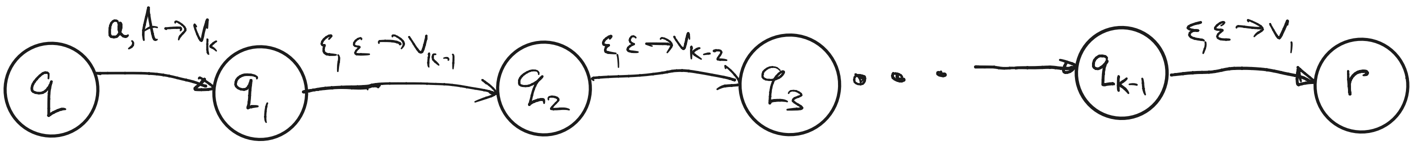


means if in state q,
next input symbol read is a,
and s is top symbol on stack
we pop s, push $v_1 .. v_k$ on stack
and move to state r

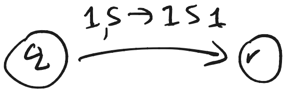
$a \in \Sigma \cup \varepsilon$

$A \in \Sigma \cup \Gamma \cup \varepsilon$

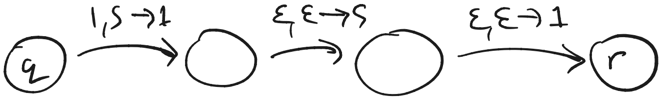
$v_1, \dots, v_k \in \Sigma \cup \Gamma \cup \varepsilon$


$a, A \to v_k$
$\varepsilon, \varepsilon \to v_{k-1}$
$\varepsilon, \varepsilon \to v_{k-2}$
$\varepsilon, \varepsilon \to v_1$
q
q_1
q_2
q_3
q_{k-1}
r

Previous example:



abbreviates:

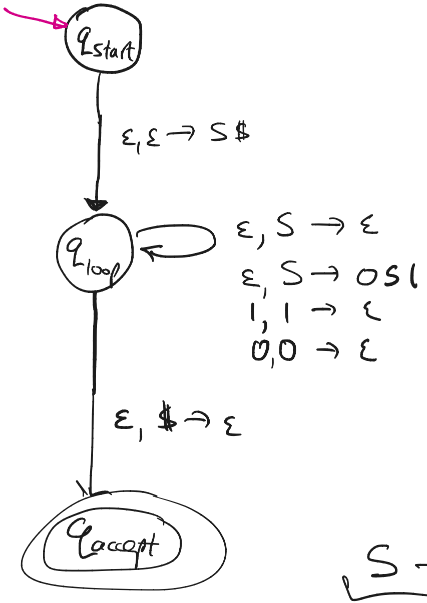

q
$1, S \to 1$
$\varepsilon, \varepsilon \to S$
$\varepsilon, \varepsilon \to 1$
r

Example

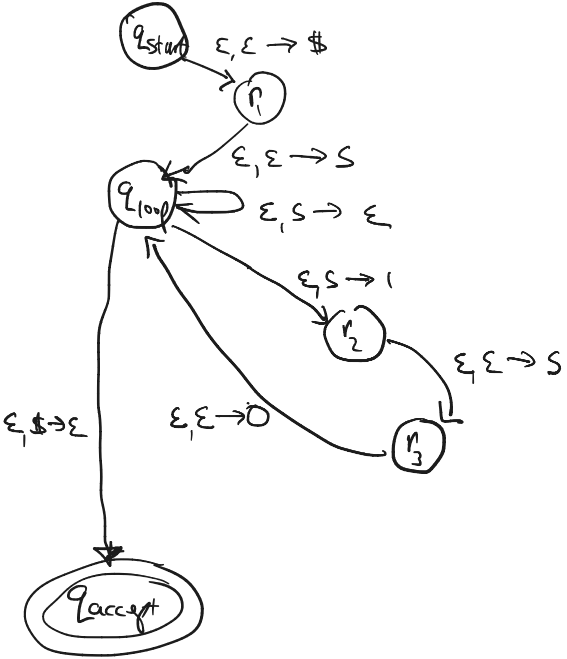
Let $g = (\Sigma, \Gamma, S, R)$ R: $S \to \varepsilon \mid 0S1$

M_g : States are $Q = \{q_{start}, q_{loop}, q_{accept}\} \cup E$ $E = \{r_1, r_2, r_3\}$

start state accept state



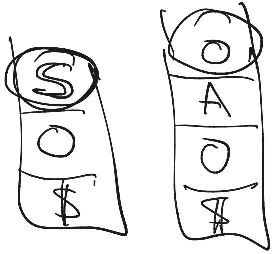
$S \to u$


q_{start}
$\varepsilon, \varepsilon \to \$$
r_1
$\varepsilon, \varepsilon \to S$
q_{loop}
$\varepsilon, S \to \varepsilon$
$\varepsilon, S \to 1$
r_2
$\varepsilon, \varepsilon \to S$
r_3
$\varepsilon, \varepsilon \to 0$
$\varepsilon, \$ \to \varepsilon$
q_{accept}

(See Example 2.14 in Book for another more complicated example)

Example:

$S \rightarrow 0S \mid 11S \mid \varepsilon$



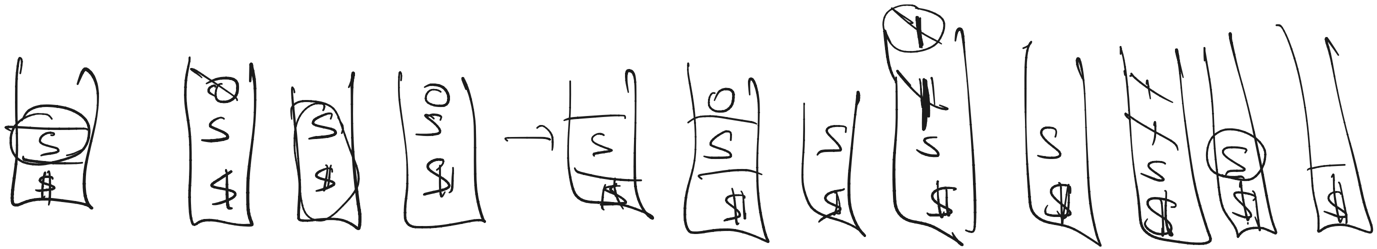
→ $w = \cancel{0}0001111$

$S \rightarrow 0S \rightarrow 00S \rightarrow 000S$

$\rightarrow 00011S$

$\rightarrow 0001111$

PDA



CFG → PDA

Theorem 1 If L has a CFG then there exists a PDA M accepting L.

Theorem (Proof of correctness)

Let $G = (\Sigma, \Gamma, S, R)$ be a CFG generating L,

and Let M_G be the PDA defined in previous slides.

then : (1) $\forall w \in L$, M_G generates w

(2) $\forall w \notin L$, M_G does not generate w

(Proof omitted - see book)

PDA → CFG

Theorem 2 If L is accepted by a PDA, then L has a CFG.

Pf sketch Assume $M = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{accept}\})$.

Modify M slightly so that it has these properties & still accepts L:

- M has a single accept state, q_{accept}
- M empties its stack before accepting
- Every transition either pushes one symbol onto stack or pops one symbol but Not both

Constructing grammar G_M from M:

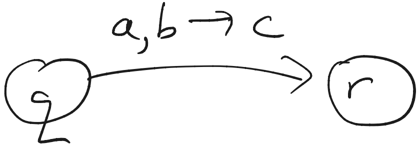
For every pair of states $p, q \in Q$ we have a variable A_{pq}

Our rules will guarantee that A_{pq} will generate exactly the set of all strings $w \in \Sigma^*$ that can take M from state p (on empty stack) to state q (on empty stack)

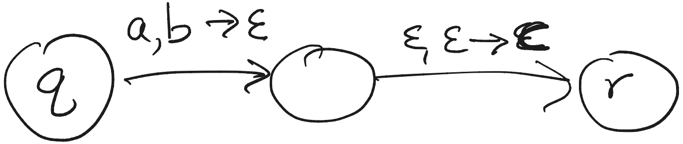
} same as starting & ending with same stack contents

Converting M to M' that only pushes or pops one symbol to/from stack at each step.

Say initially had transition



becomes



PDA → CFG

Theorem 2 If L is accepted by a PDA, then L has a CFG.

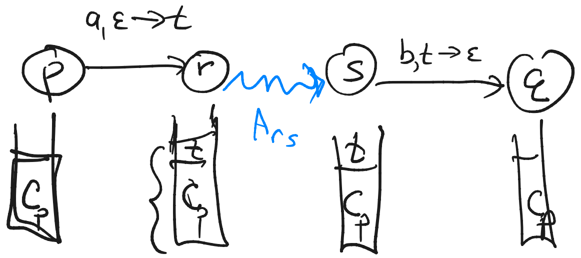
Pf sketch Assume $M = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{accept}\})$.

Variables of G : $\{A_{pq} \mid p,q \in Q\}$ Start Variable : $A_{q_0 q_{accept}}$

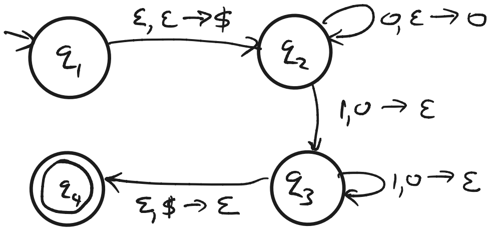
1. For each $p \in Q$ add the rule $A_{pp} \to \varepsilon$
2. For each $p,q,r \in Q$ add the rule $A_{pq} \to A_{pr} A_{rq}$
3. For each $p,q,r,s \in Q$, add rule $A_{pq} \to a A_{rs} b$ if

read a, push u → $(r,t) \in \delta(p,a,\varepsilon)$

read b, pop u → $(q,\varepsilon) \in \delta(s,b,t)$



Example $L = \{0^n 1^n \mid n \geq 1\}$



CFG: Start variable: A_{14}

$A_{11} \to \varepsilon$; $A_{22} \to \varepsilon$; $A_{33} \to \varepsilon$; $A_{44} \to \varepsilon$

$A_{11} \to A_{11}A_{11} \mid A_{12}A_{21} \mid A_{13}A_{31} \mid A_{14}A_{41}$

$A_{12} \to A_{11}A_{12} \mid A_{12}A_{22} \mid A_{13}A_{32} \mid A_{14}A_{42}$

$A_{13} \to A_{11}A_{13} \mid A_{12}A_{23} \mid A_{13}A_{33} \mid A_{14}A_{43}$

$\vdots$

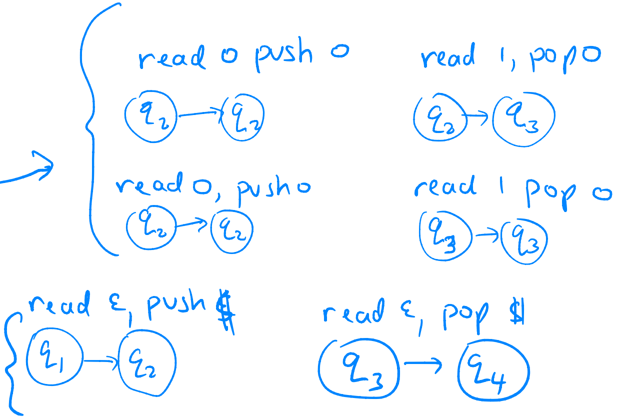
$A_{42} \to A_{41}A_{12} \mid A_{42}A_{22} \mid A_{43}A_{32} \mid A_{44}A_{42}$

$A_{43} \to A_{41}A_{13} \mid A_{42}A_{23} \mid A_{43}A_{33} \mid A_{44}A_{43}$

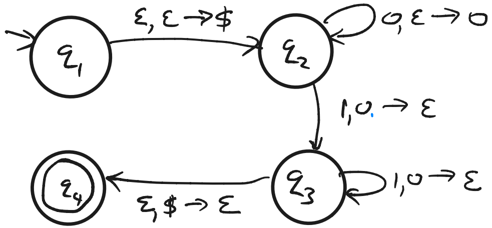
$A_{44} \to A_{41}A_{14} \mid A_{42}A_{24} \mid A_{43}A_{34} \mid A_{44}A_{44}$

$A_{23} \to 0A_{22}1 \mid 0A_{23}1$

$A_{14} \to \varepsilon A_{23} \varepsilon$



Example $L = \{0^n 1^n \mid n \geq 1\}$



Note: Any state A_{pq} such that q is not reachable at all from p can be removed. Removing these useless vars and associated rules we are left with:

Start var: A_{14}

$A_{11} \to \varepsilon$; $A_{22} \to \varepsilon$; $A_{33} \to \varepsilon$; $A_{44} \to \varepsilon$

$A_{11} \to A_{11} A_{11}$ $A_{22} \to A_{22} A_{22}$ $A_{33} \to A_{33} A_{33}$ $A_{44} \to A_{44} A_{44}$

$A_{12} \to A_{11} A_{12} \mid A_{12} A_{22}$

$A_{13} \to A_{11} A_{13} \mid A_{12} A_{23} \mid A_{13} A_{33}$

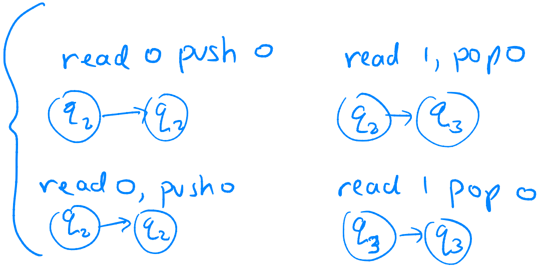
$A_{23} \to A_{24} A_{34}$

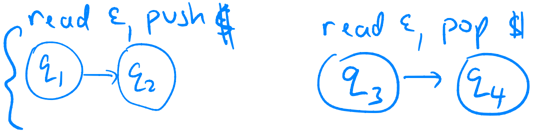
$A_{24} \to A_{23} A_{34}$

$A_{34} \to A_{33} A_{34} \mid A_{34} A_{44}$

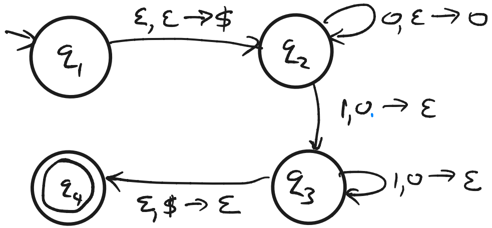
$A_{23} \to 0 A_{22} 1 \mid 0 A_{23} 1$

$A_{14} \to \varepsilon A_{23} \varepsilon$





Example $L = \{0^n 1^n \mid n \geq 1\}$



Further simplifying (e.g., $A_{11} \to \varepsilon \mid A_{11}A_{11}$ is same as $A_{11} \to \varepsilon$)

Start var: A_{14}

$A_{11} \to \varepsilon$; $A_{22} \to \varepsilon$; $A_{33} \to \varepsilon$; $A_{44} \to \varepsilon$

$A_{13} \to A_{12}A_{23}$

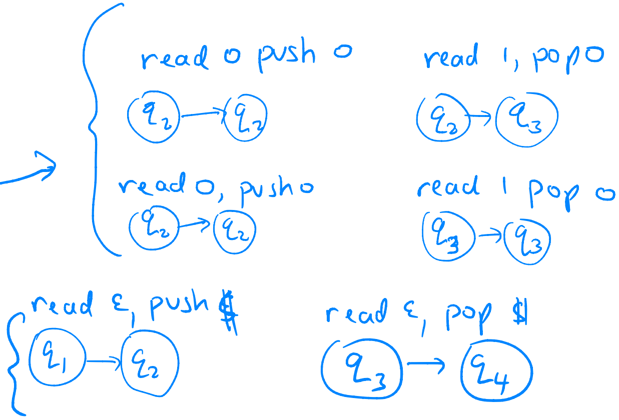
$A_{23} \to A_{24}A_{34}$

$A_{24} \to A_{23}A_{34}$

$A_{23} \to 0A_{22}1 \mid 0A_{23}1$

$A_{14} \to \varepsilon A_{23} \varepsilon$

left with just these


read 0 push 0
q2
q2
read 1, pop 0
q2
q3
read 0, push 0
q2
q2
read 1 pop 0
q3
q3
read ε, push $
q1
q2
read ε, pop $
q3
q4