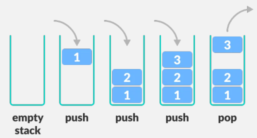
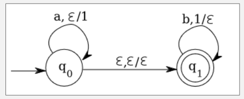
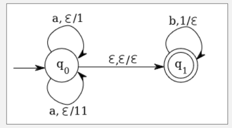
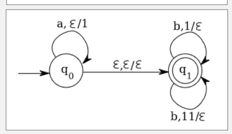
Pushdown Automata

- a *pushdown automaton* is a finite-state automaton with a stack which acts as a memory









  - each transition σ,*x*/*y* represents consuming symbol σ from the input string, popping *x* from the stack, and pushing *y* onto the stack
  - a string is accepted if the machine finishes in an accepting state with an empty stack

https://www.programiz.com/dsa/stack

Pushdown Automata

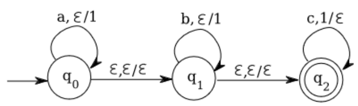
Definition 4.4. A pushdown automaton M is specified by six components $M = (Q, \Sigma, \Lambda, q_0, \partial, F)$ where

- Q is a finite set of states.
- Σ is an alphabet. Σ is the ***input alphabet*** for M.
- Λ is an alphabet. Λ is the ***stack alphabet*** for M.
- $q_0 \in Q$ is the ***start state*** of M.
- $F \subseteq Q$ is the set of ***final*** or ***accepting*** states in M.
- ∂ is the set of transitions in M. ∂ can be taken to be a finite subset of the set $(Q \times (\Sigma \cup \{\varepsilon\}) \times \Lambda^*) \times (Q \times \Lambda^*)$. An element $((q_1, \sigma, x), (q_2, y))$ of ∂ represents a transition from state q_1 to state q_2 in which M reads σ from its input string, pops x from the stack, and pushes y onto the stack.

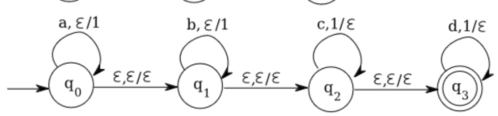
the symbols used in the stack do not have to be the same as those in the language accepted

1. Identify the context-free language that is accepted by each of the following pushdown automata. Explain your answers.

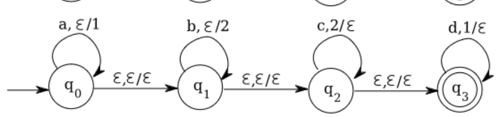
a)



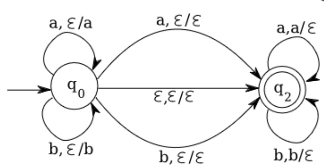
b)



c)



d)



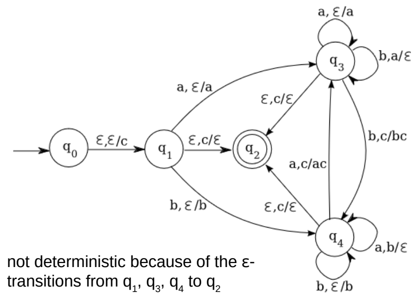
2. Let B be the language over the alphabet $\{\,(\,,\,)\,\}$ that consists of strings of parentheses that are balanced in the sense that every left parenthesis has a matching right parenthesis. Examples include (), (())(), ((())())()(()), and the empty string. Find a pushdown automaton with a single state that accepts the language B. Explain how your automaton works, and explain the circumstances in which it will *fail* to accept a given string of parentheses.

Deterministic Pushdown Automata

- a pushdown automaton is *deterministic* if there is no circumstance under which two different transition rules can apply
  - ε-transitions are allowed if they are the only transitions from a state

Definition 4.5. Let L be a language over an alphabet Σ, and let $\$$ be a symbol that is not in Σ. We say that L is a ***deterministic context-free language*** if there is a deterministic pushdown automaton that accepts the language $L\$$ (which is equal to $\{w\$ \mid w \in L\}$).

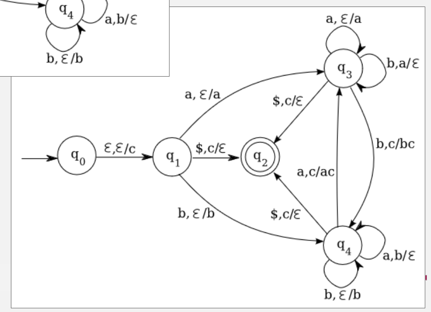
  - a deterministic context-free language can be parsed efficiently
  - there are context-free languages that are not deterministic context-free
  - why the $?



not deterministic because of the ε-transitions from q_1, q_3, q_4 to q_2

in this case, those ε-transitions are only there to allow a last step to clean up the stack before accepting – when the end of the string is reached, pop the *c* off the stack to leave an empty stack

since ε-transitions are always applicable, add a special end-of-string symbol to limit the stack cleanup to the last step



4. Find a deterministic pushdown automaton that accepts the language $\{wcw^R \mid w \in \{a,b\}^*\}$.

5. Show that the language $\{a^n b^m \mid n \neq m\}$ is deterministic context-free.

6. Show that the language $L = \{w \in \{a,b\}^* \mid n_a(w) > n_b(w)\}$ is deterministic context-free.

3. Suppose that L is language over an alphabet Σ. Suppose that there is a deterministic pushdown automaton that accepts L. Show that L is deterministic context-free. That is, show how to construct a deterministic pushdown automaton that accepts the language $L\$$. (Assume that the symbol $\$$ is not in Σ.)