PUSHDOWN AUTOMATA

Recall: A context-free language is one which is generated by a CFG.

Examples: Strings with an equal number of 'a's and 'b's

Strings with balanced parentheses

Strings which are palindromes over $\Sigma = \{a, b\}$ etc.

Exercise: Construct an unambiguous CFG for this language

Today: A machine model for context-free languages

We said that regular expressions code up the class of languages recognized by NFAs/DFAs.

What is the equivalent machine model for context-free languages?

Consider an NFA with access to a global stack. One can

- push a symbol onto the stack
- pop the top symbol off the (non-empty) stack

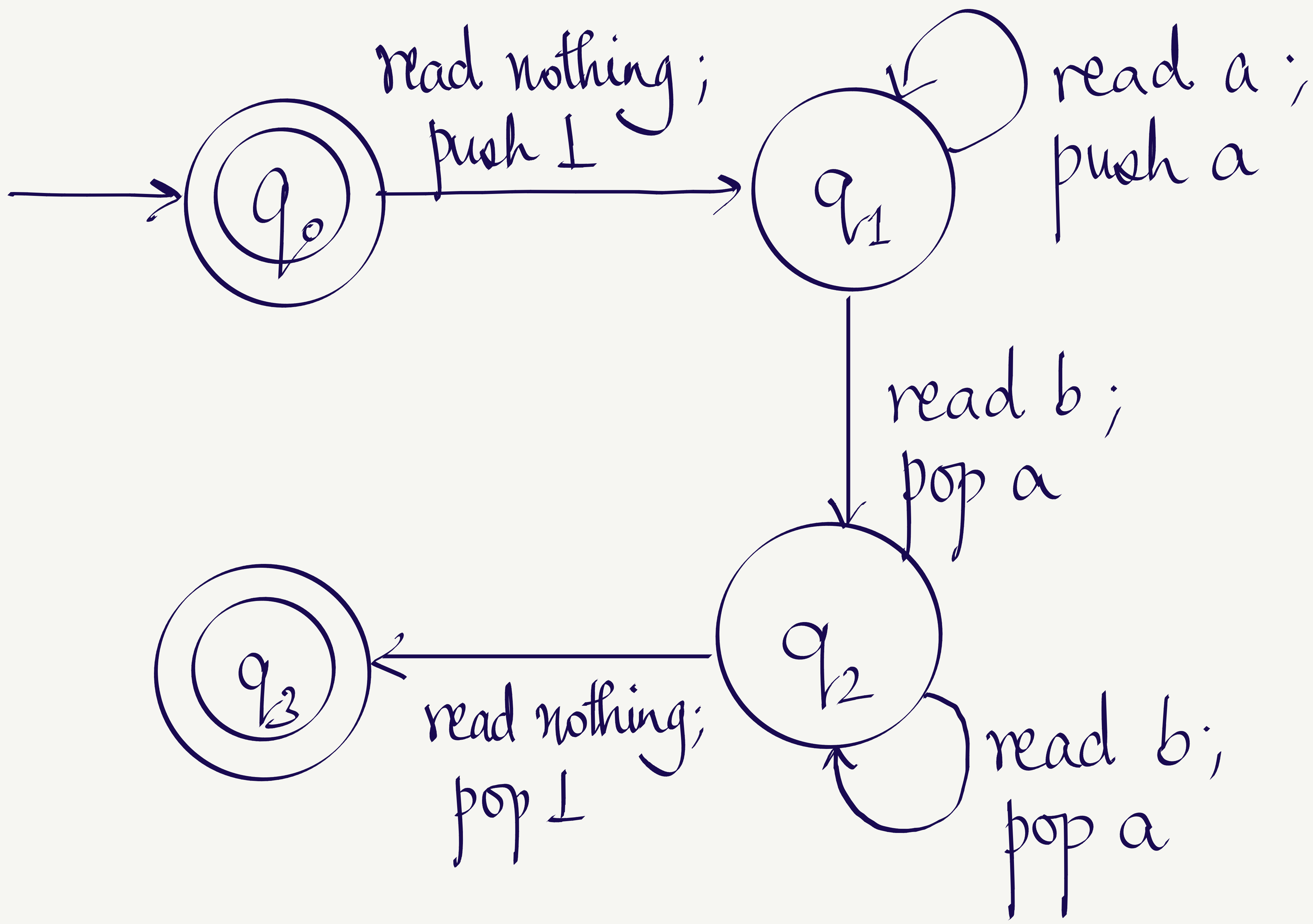
So how do we recognize $\mathcal{L} = \{a^n b^n \mid n \geq 0\}$?

Start in the initial state; stack contains an end marker $\perp$.

Consume input letters one at a time:

- if you see an 'a', push it onto the stack
- if you see a 'b', pop off the top symbol, as long as it is not $\perp$

When do we accept?



* pop operations get stuck if the top letter in the stack is NOT the letter required to be popped.

Pushdown automata (PDA): A 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$

Q: set of states Σ: input alphabet Γ: stack alphabet, $\perp \in \Gamma$.

$\Delta \subseteq (Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\})) \times (Q \times \Gamma^*)$: transition relation

$q_0 \in Q$: start state $F \subseteq Q$: set of accepting states

How do we interpret $((q, a, C), (q', D_1 D_2 \cdots D_k)) \in \Delta$?

Whenever the machine is in state q reading letter $a \in \Sigma$,
and the symbol $C \in \Gamma$ is on the top of the stack,
it can
- pop C off the stack (if $C = \varepsilon$, no need to pop anything)
- push D_k, then D_{k-1}, ..., then D_1 onto the stack.
- move to state q', and read the next input letter.

If a is ε, do the same thing, but without reading anything!

Quiz