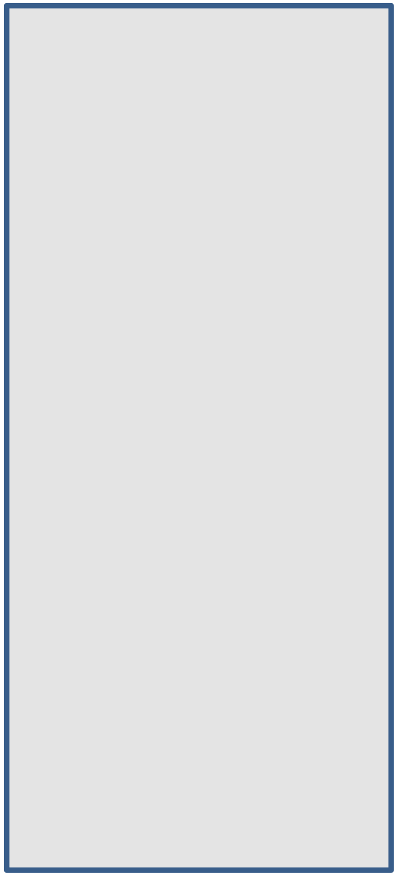
PushDown Automata

By Dr. Fathy

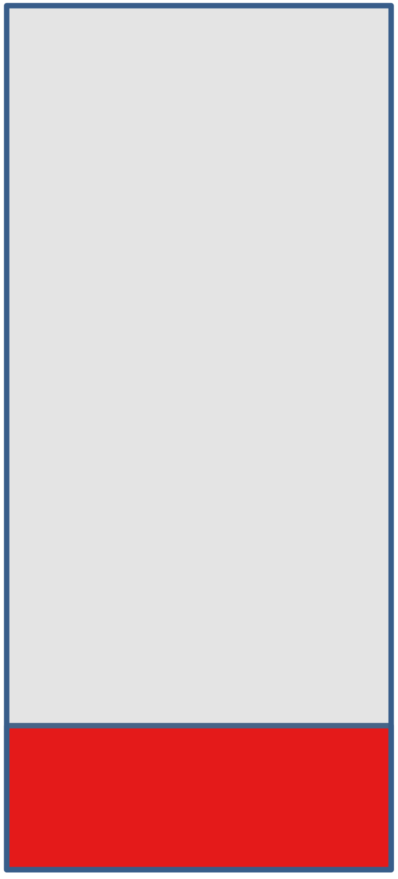
What is a stack?

- A stack is a Last In First Out data structure where I only have access to the last element inserted in the stack.
- In order to access other elements I have to remove those that are on top one by one.

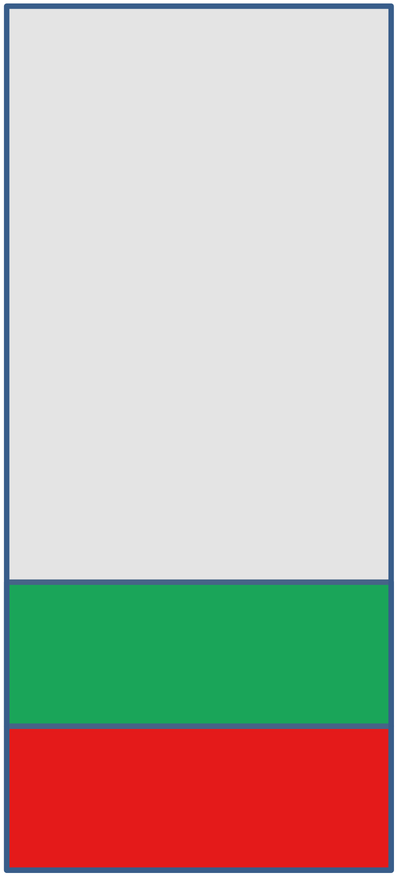
Stack



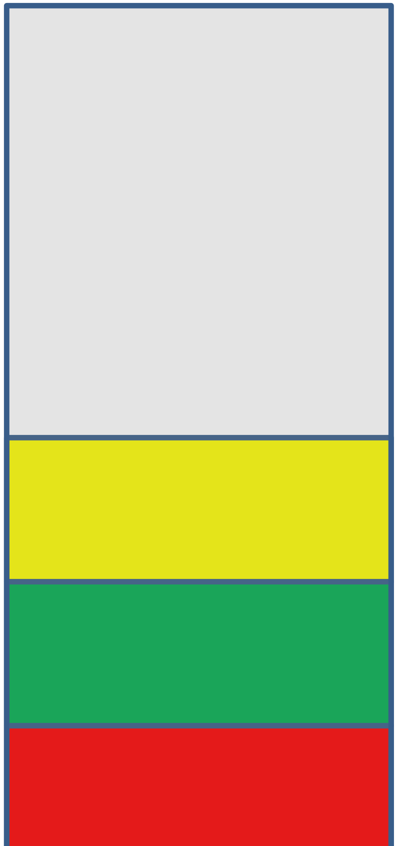
Stack



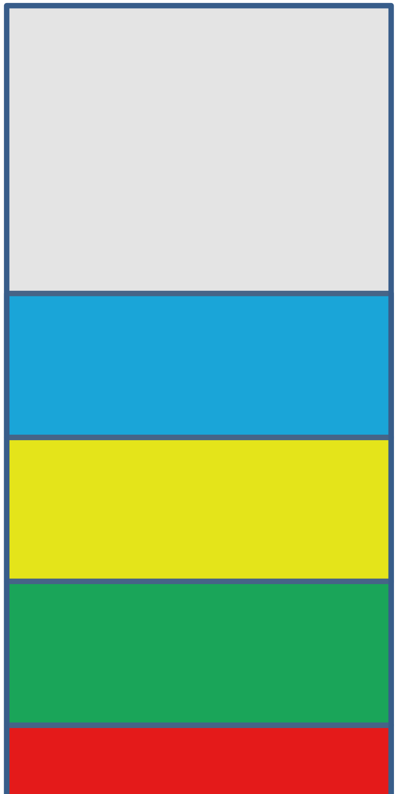
Stack



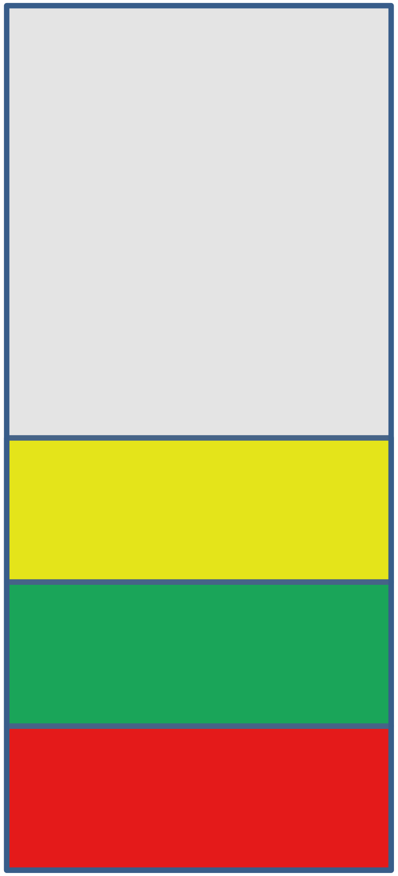
Stack



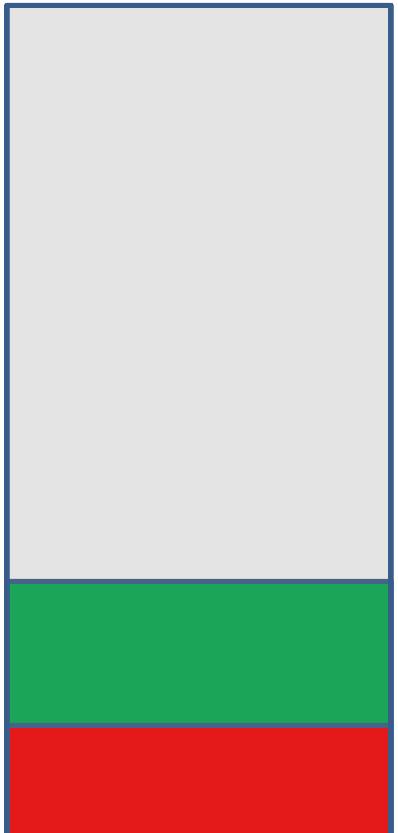
Stack



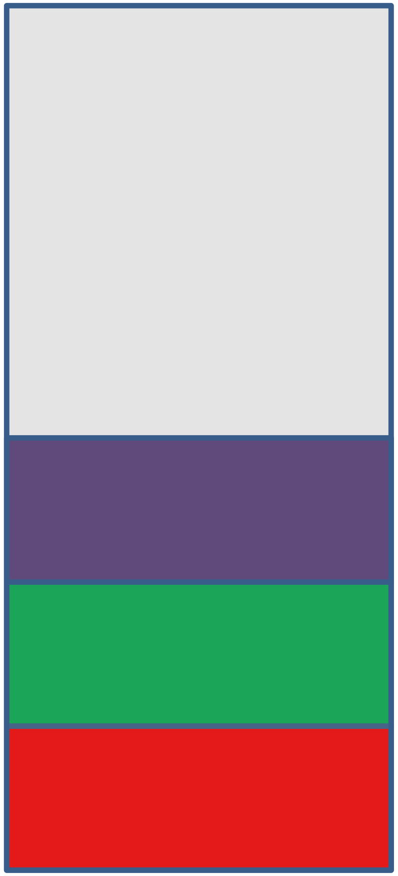
Stack



Stack



Stack



What is a PDA?

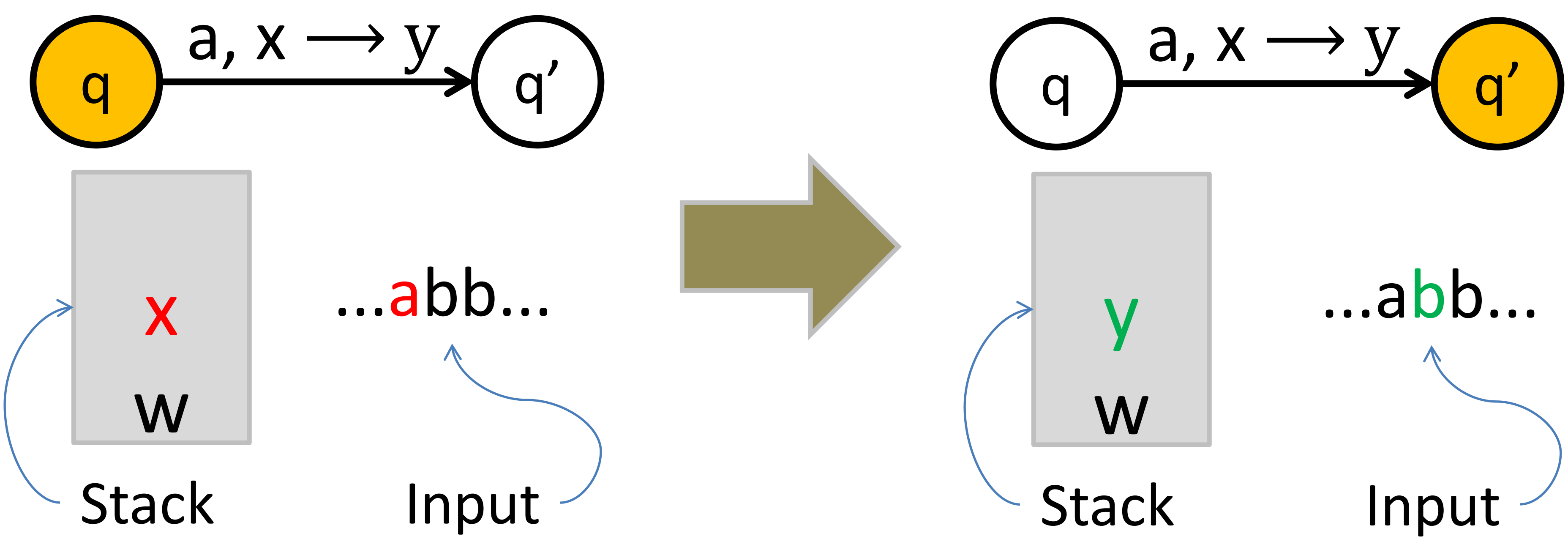
- A PDA is an enhanced finite automaton that also contains an **infinite** stack.
- The transitions in a PDA are of the form

  $$a, x \longrightarrow y$$

  meaning that if you see an a in the input string and the stack contains the symbol x on top then you remove the x and add a y.
- The stack gives us extra power to recognize non-regular languages.

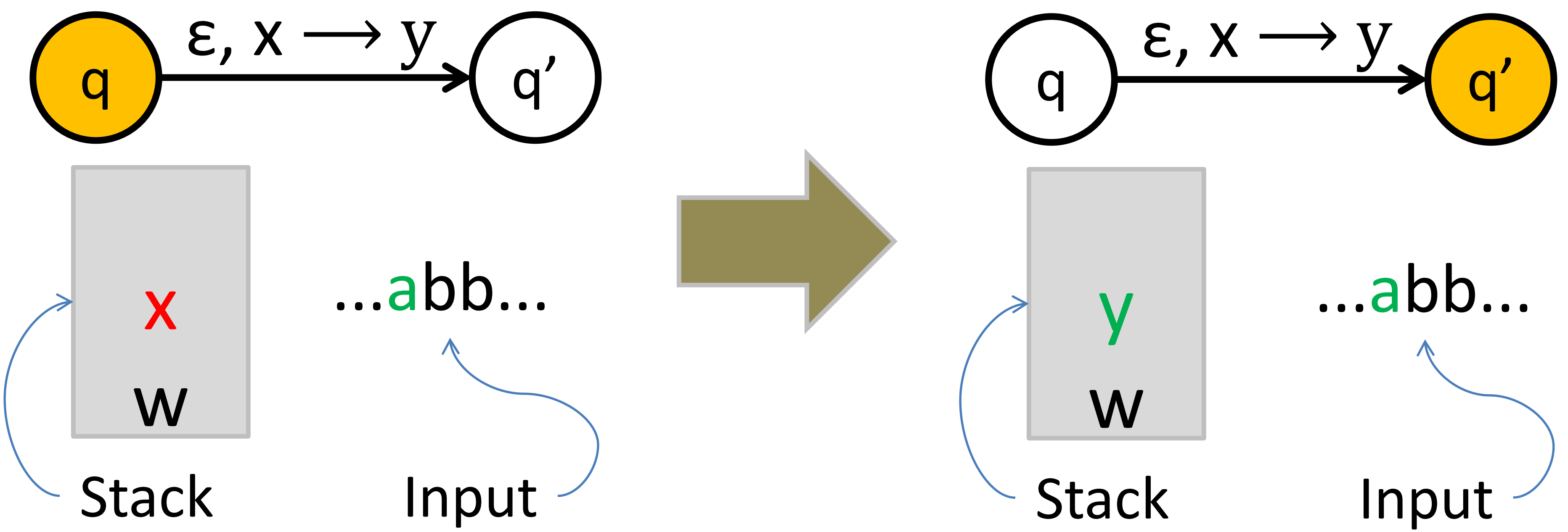
Transitions

- Transitions of the form *a, x* $\longrightarrow$ *y* require that the next input symbol should be *a* and the top stack symbol should be *x.*



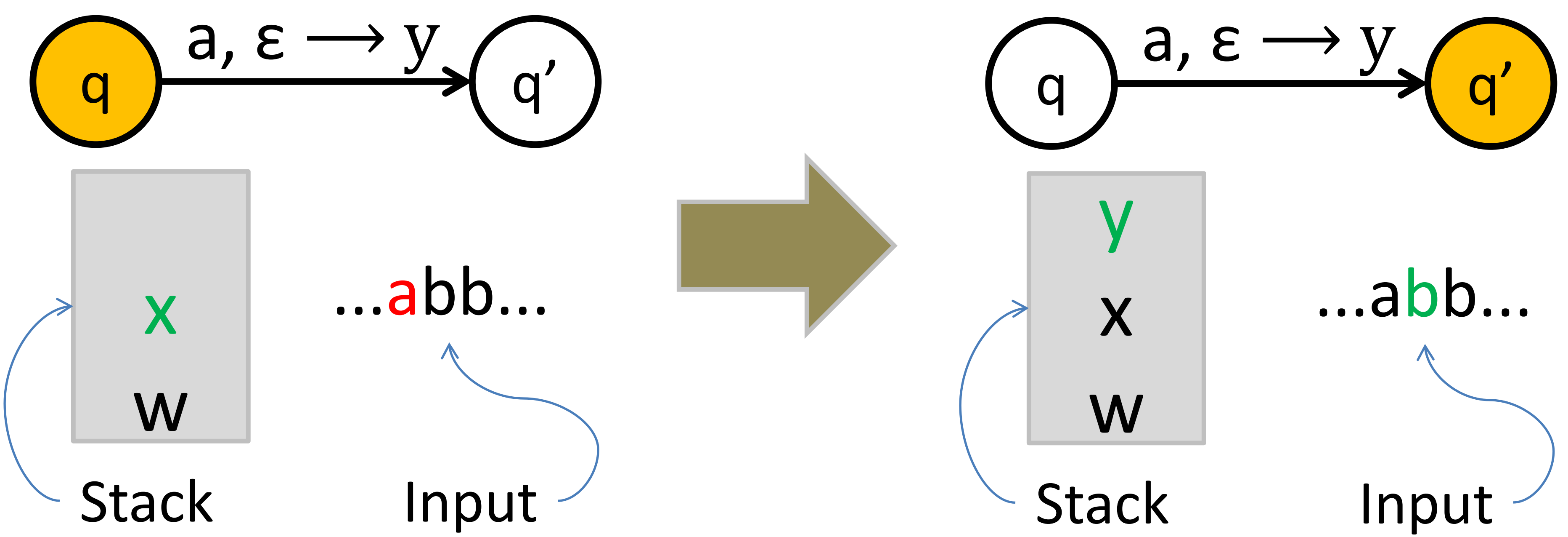
Transitions

- Transitions of the form *ε, x* $\longrightarrow$ *y* require that the top stack symbol is *x.*



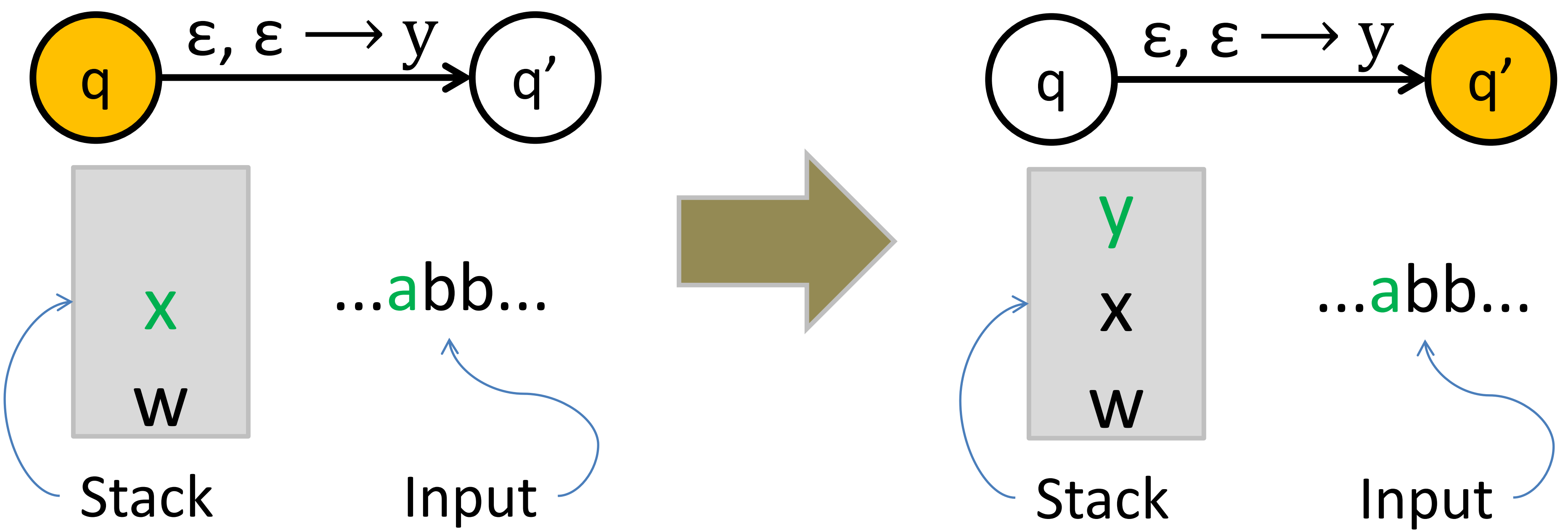
Transitions

- Transitions of the form *a, ε ⟶ y* require that the next input symbol is *a.*



Transitions

- Transitions of the form $\varepsilon, \varepsilon \longrightarrow y$ can be followed without restrictions.



PDA Accept – Reject Status

- The PDA accepts when there exists a computation path such that:
  - The computation path ends in an accept state
  - All the input is consumed
  - (no requirement for the stack)
- The PDA rejects when all the paths:
  - Either end in a non-accepting state
  - Or are incomplete (meaning that at some point there is no possible transition under the current input and stack symbols)

A PDA for $\{a^n b^n : n \geq 0\}$

- We usually use the stack for counting.
- For this language for example, you first insert all the as in the stack until you start seeing bs .
- When you see the first b start removing as from the stack.
- When you have consumed the whole string you check the stack: if it's empty then this means that the number of as equals the number of bs.

Is the stack empty?

How can you check if the stack is empty?

- What we usually do is to place a special symbol (for example a $) at the bottom of the stack.
- Whenever we find the $ again we know that we reached the end of the stack.
- **In order to accept a string there is no need for the stack to be empty.**

Stack push and pop in PDA

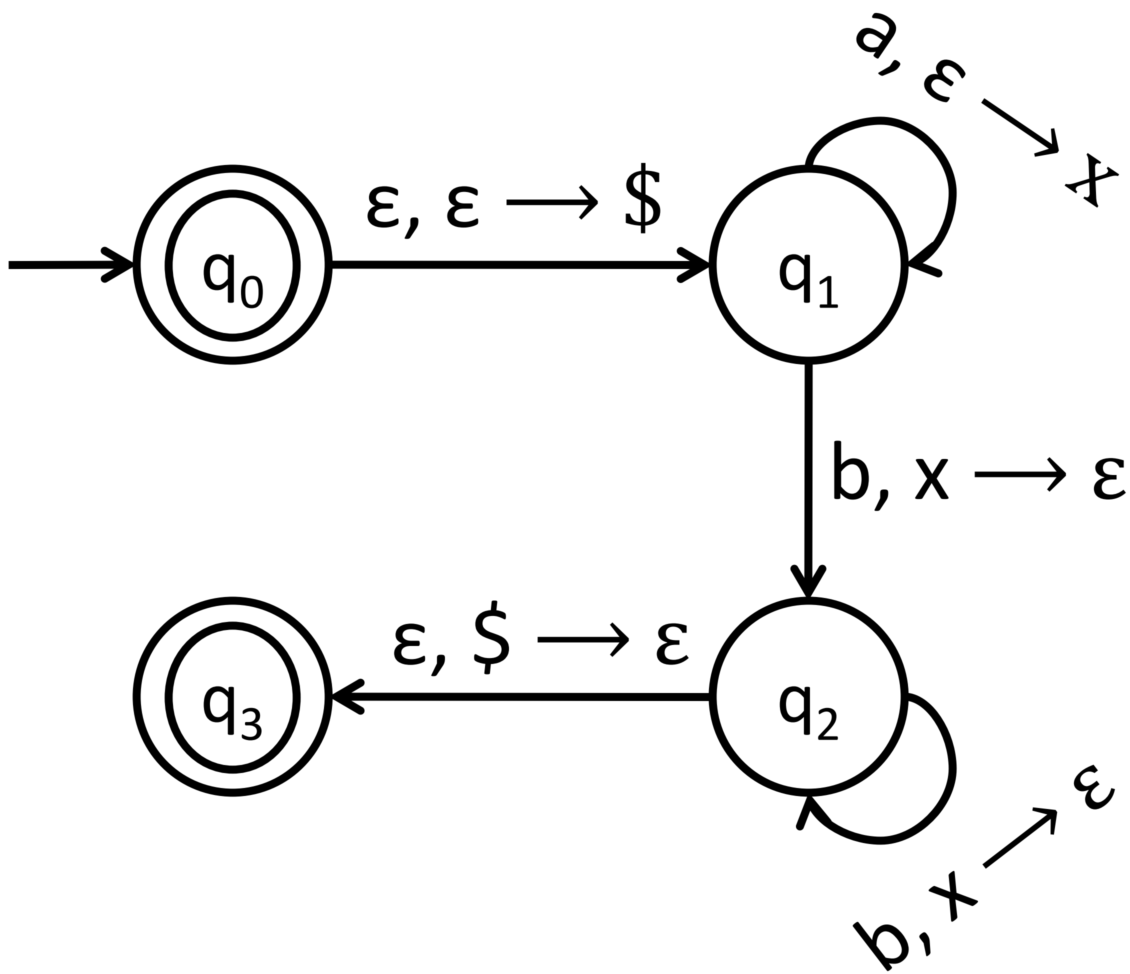
- a, ε ⟶ t

when you see an a in the input push t on the stack

- a, b ⟶ ε

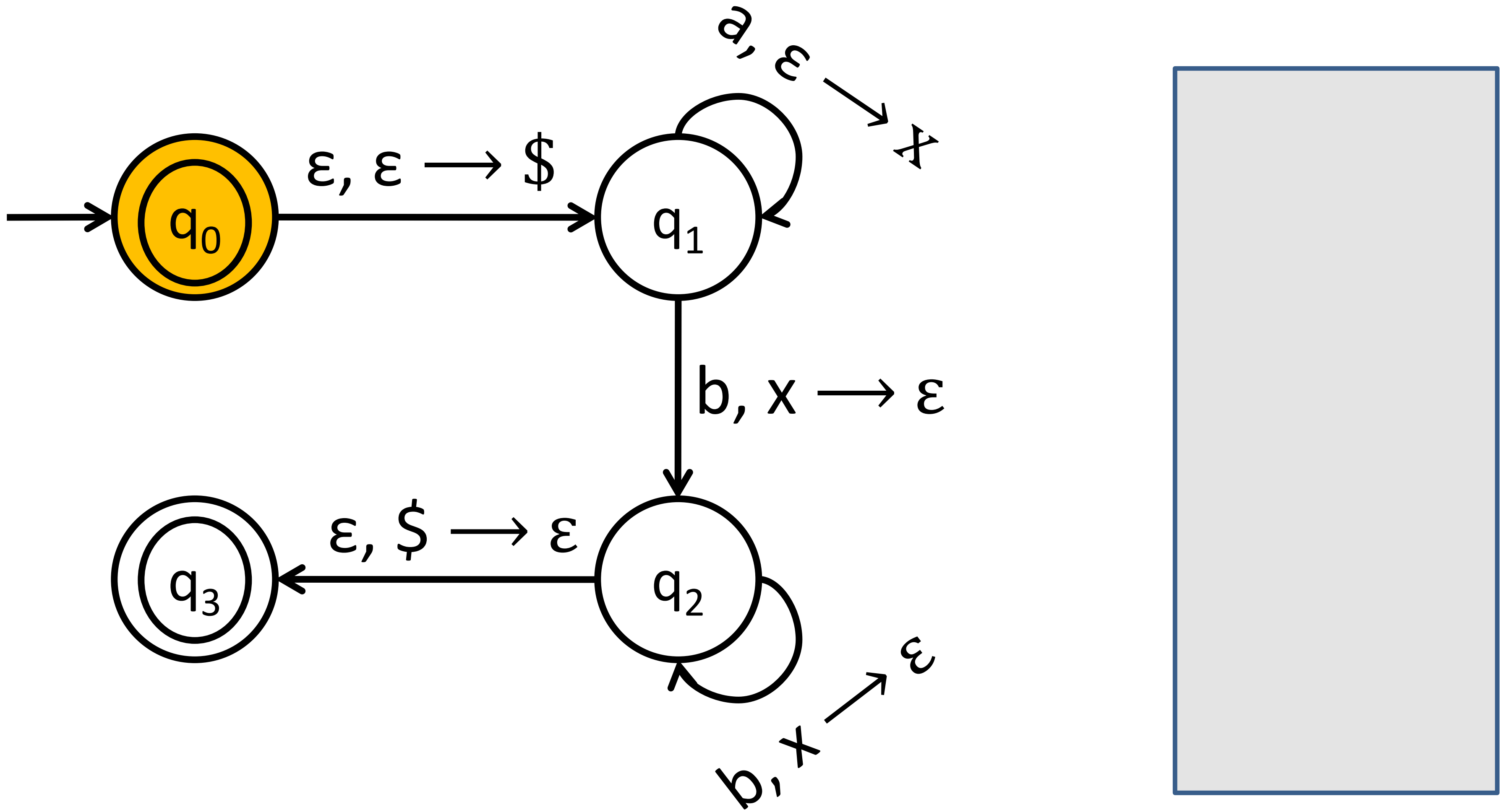
when you see an a in the input and b is on the top of the stack, pop b out.

A PDA for $\{a^n b^n : n \geq 0\}$



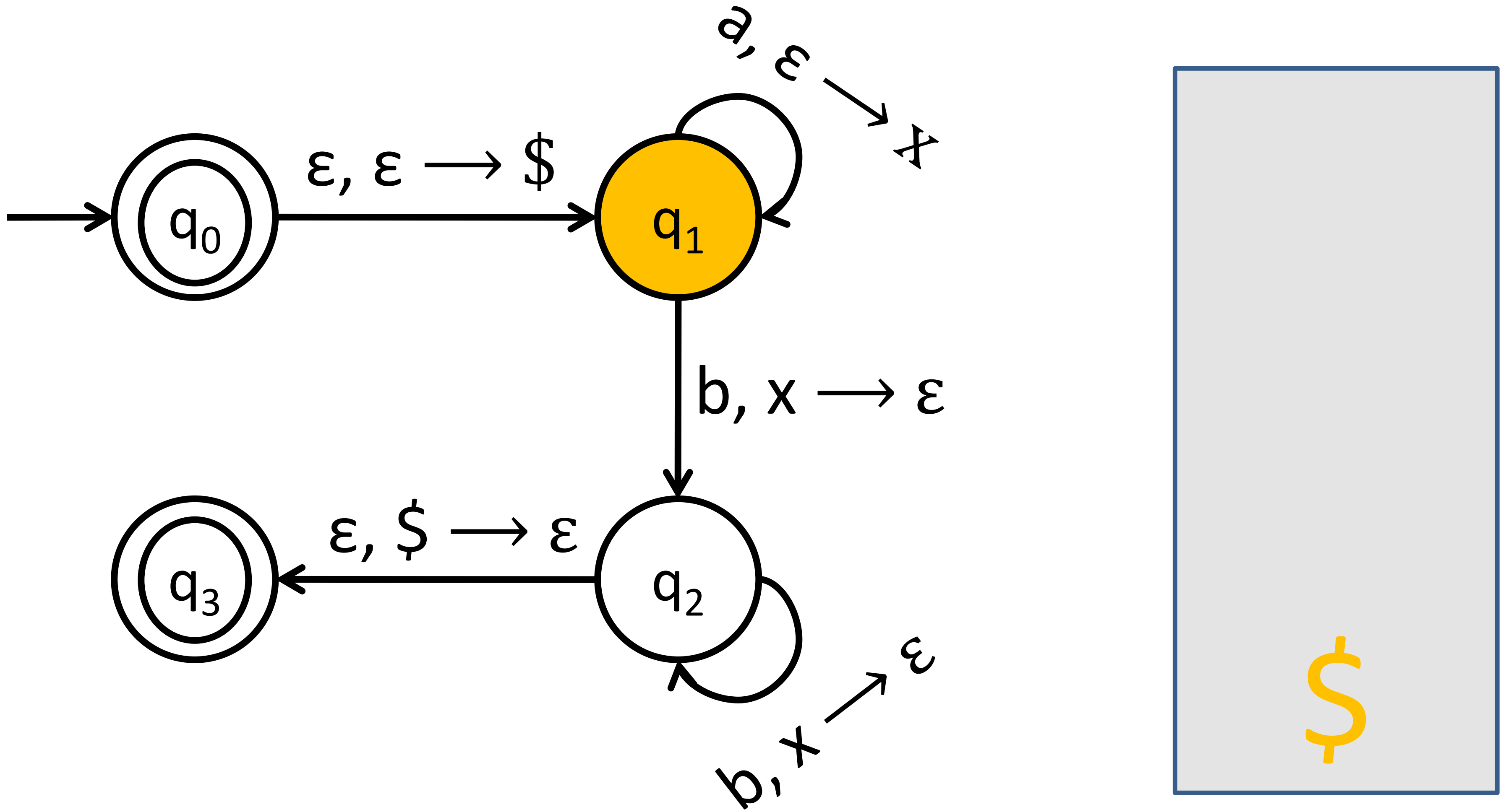
Visualization of $\{a^n b^n : n \geq 0\}$

aaabbb



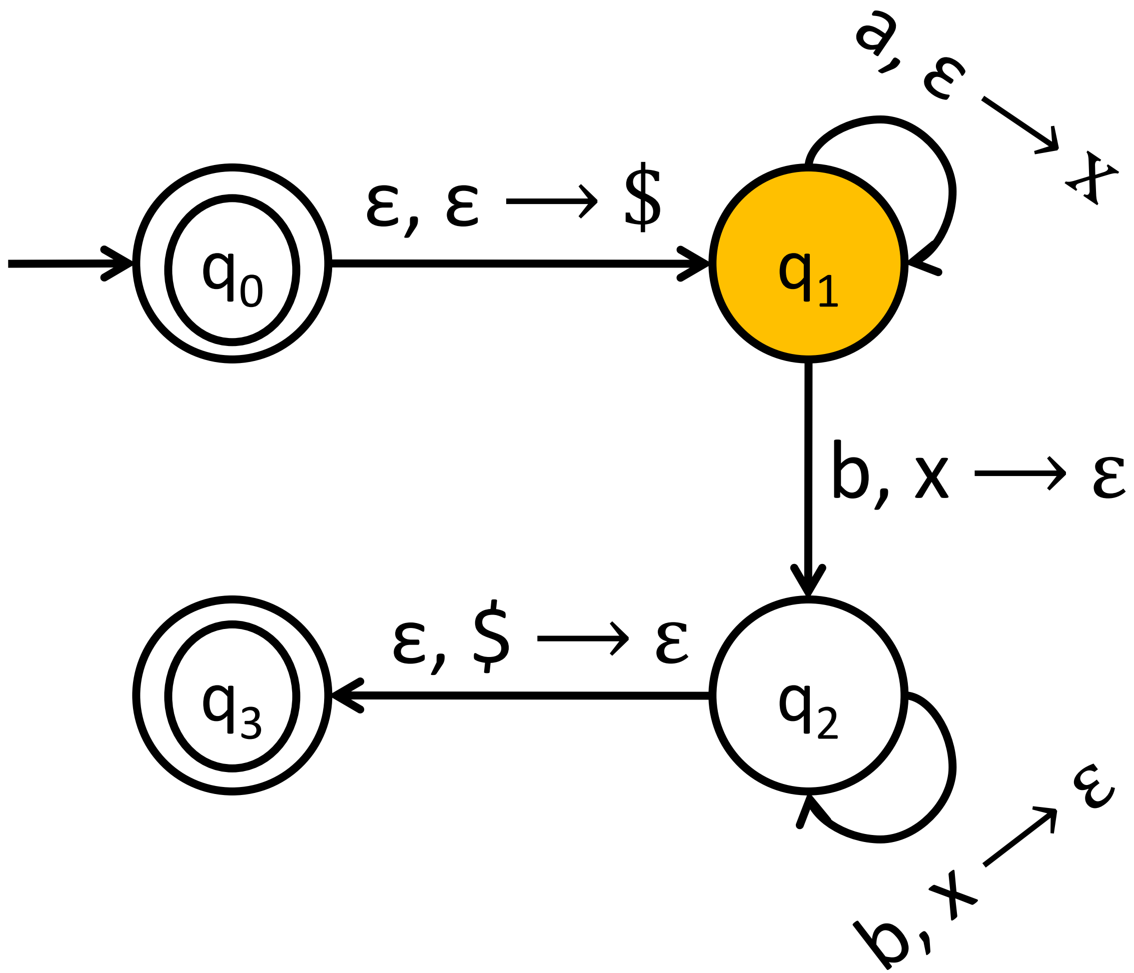
Visualization of $\{a^n b^n : n \geq 0\}$

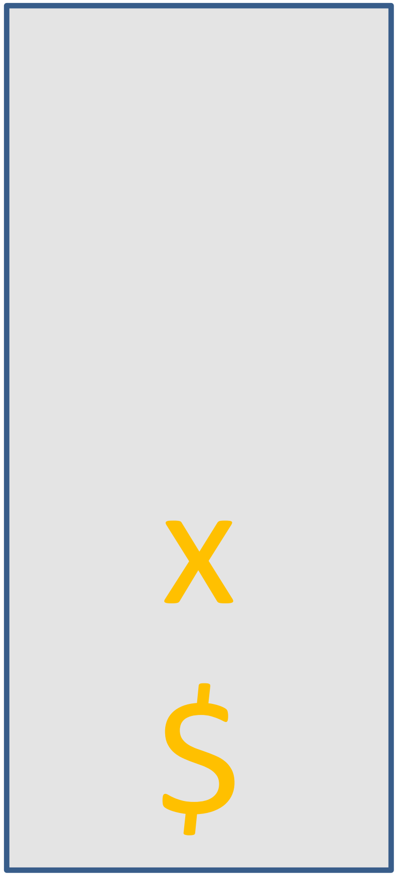
aaabbb



Visualization of $\{a^n b^n : n \geq 0\}$

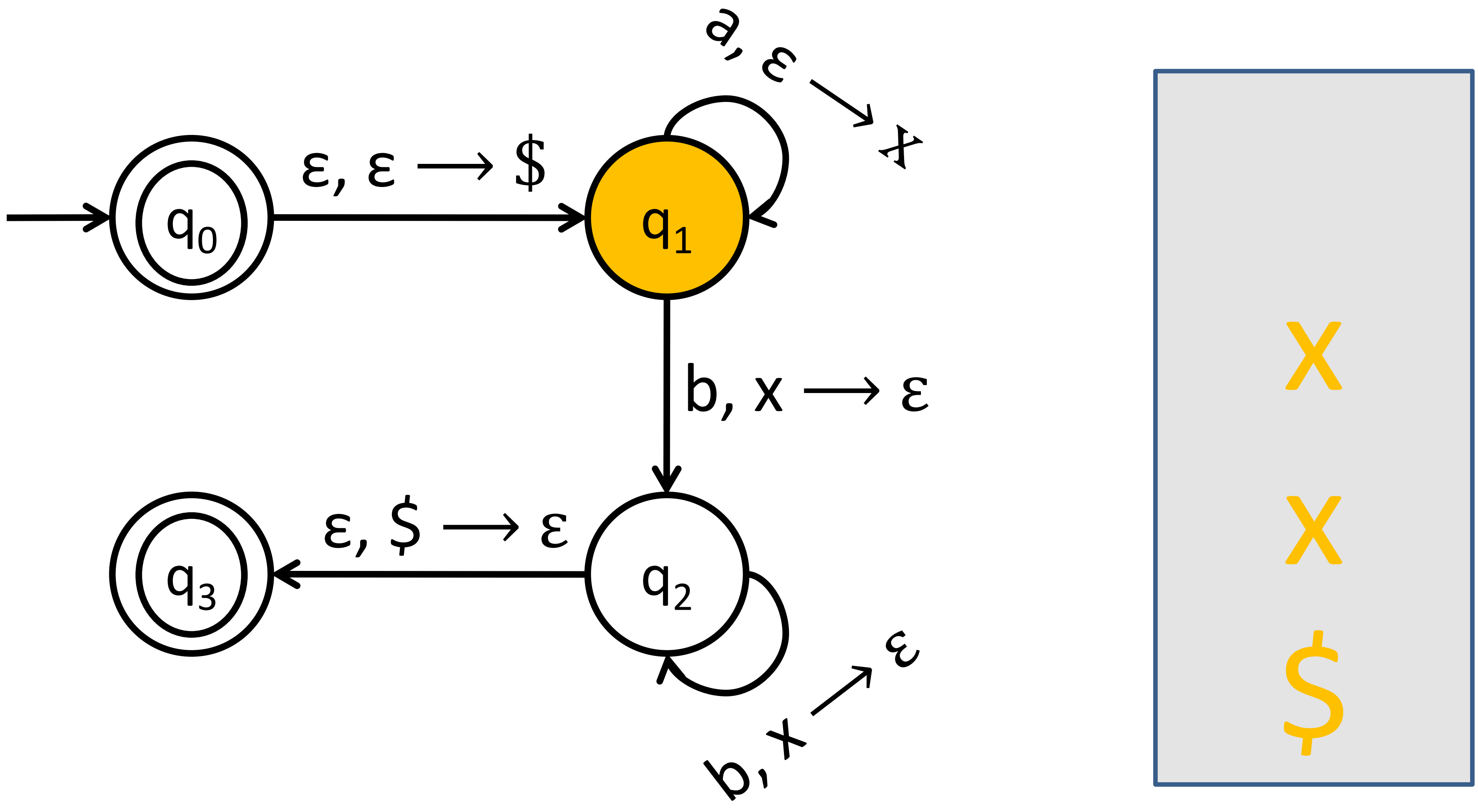
aaabbb





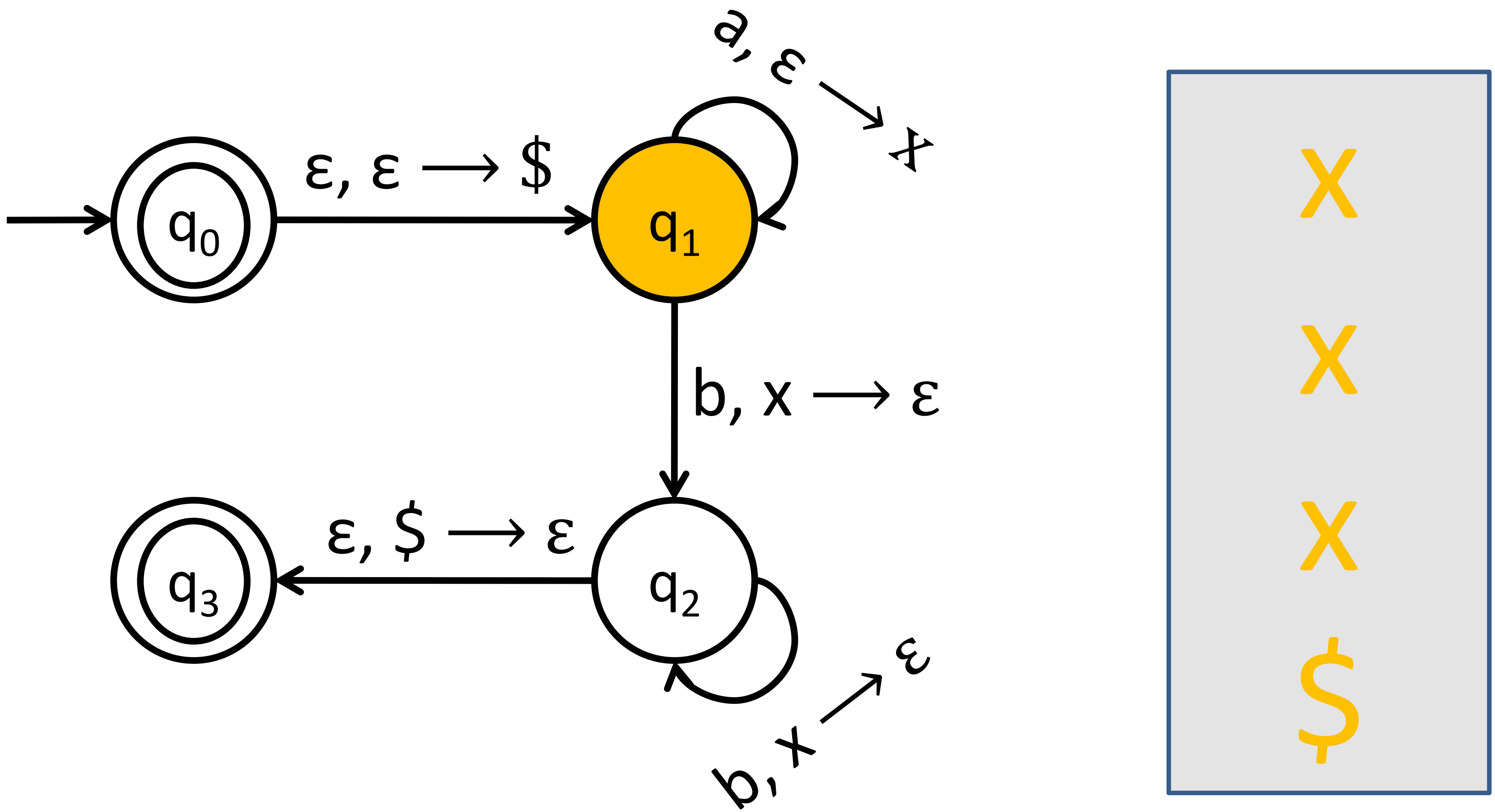
Visualization of $\{a^n b^n : n \geq 0\}$

aaabbb



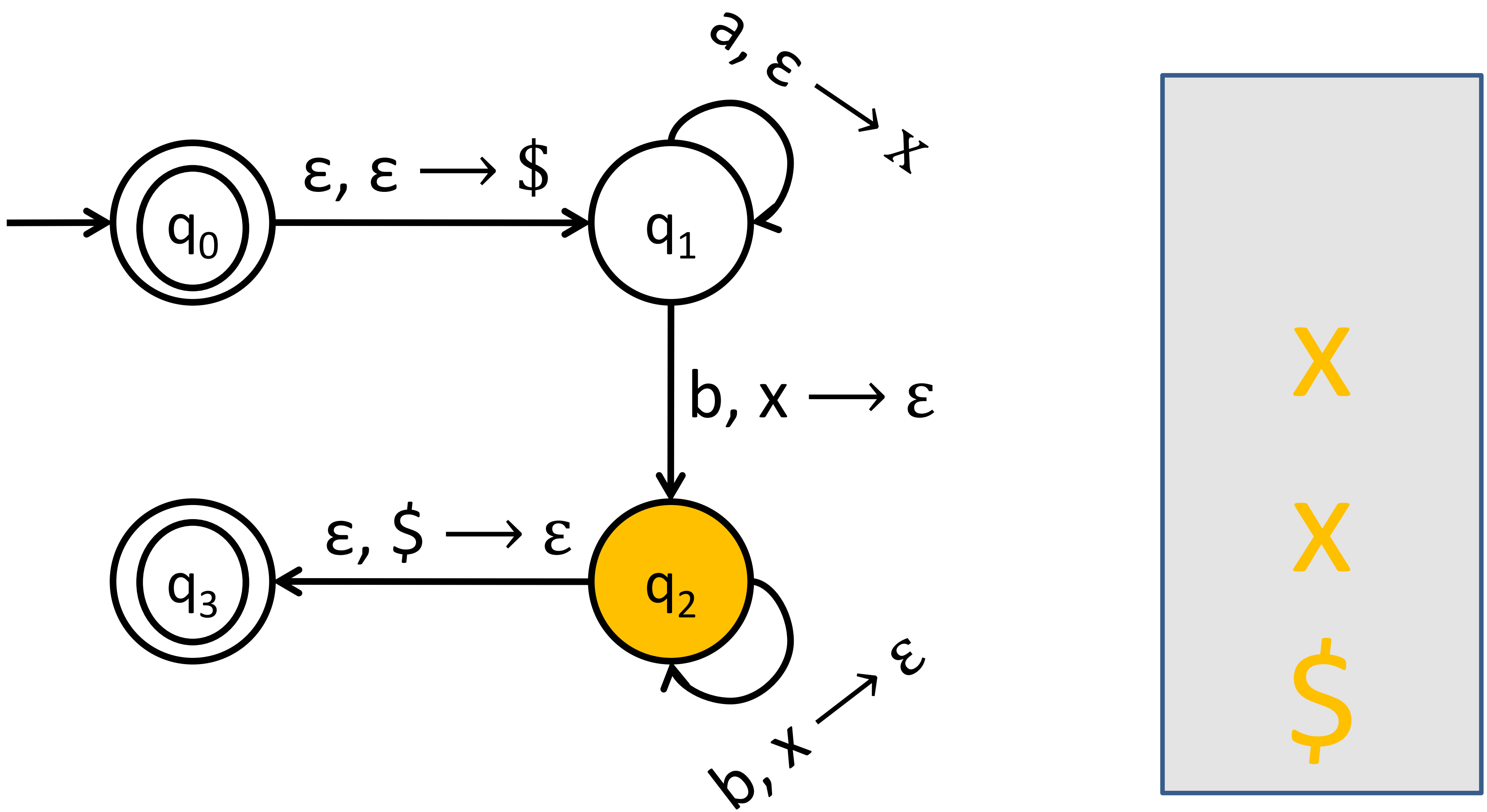
Visualization of $\{a^n b^n : n \geq 0\}$

aaabbb



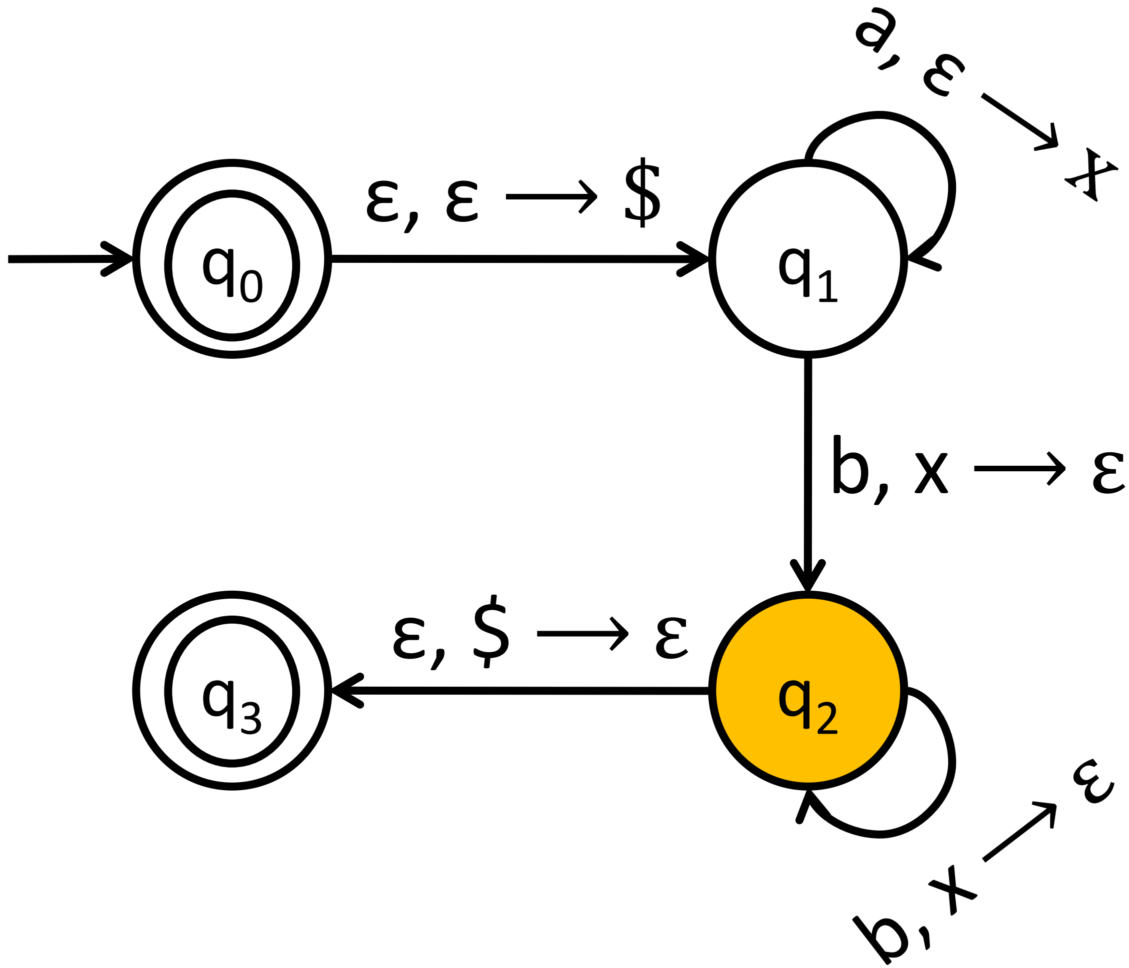
Visualization of $\{a^n b^n : n \geq 0\}$

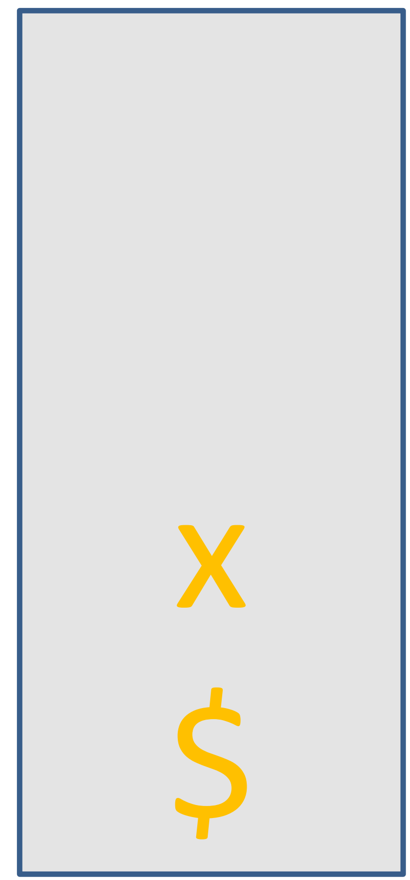
aaabbb



Visualization of $\{a^n b^n : n \geq 0\}$

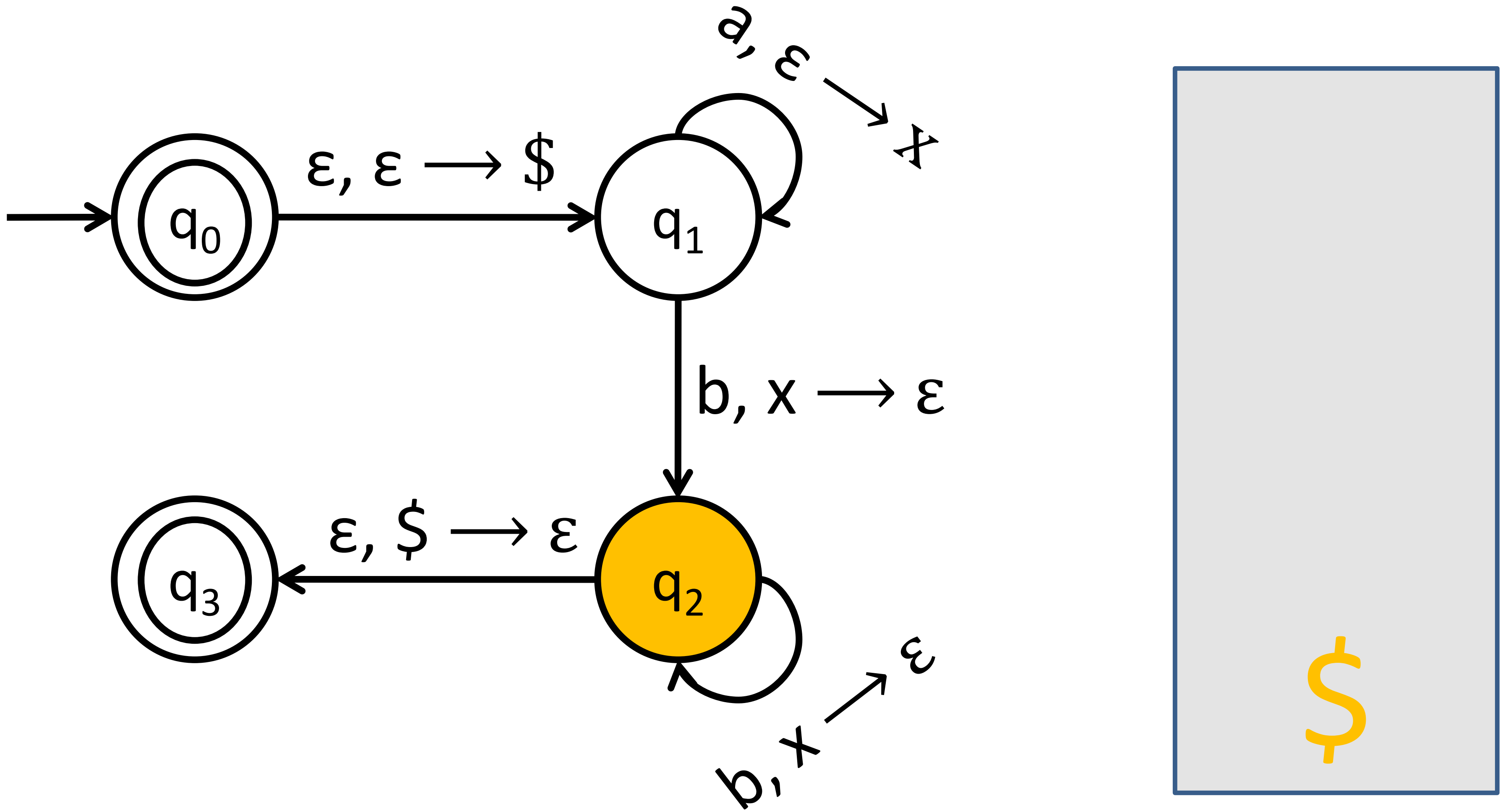
aaabbb





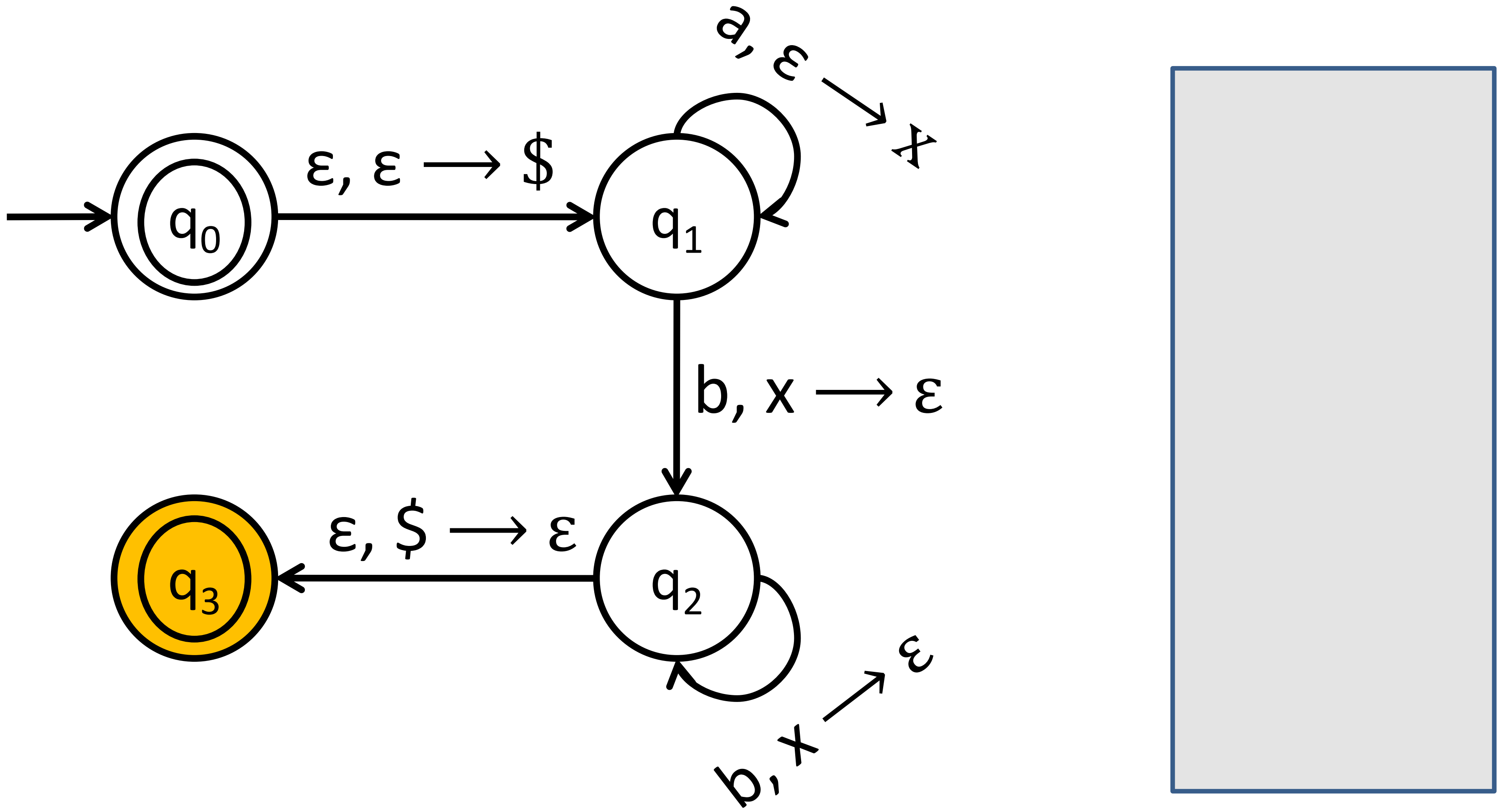
Visualization of $\{a^n b^n : n \geq 0\}$

aaabbb



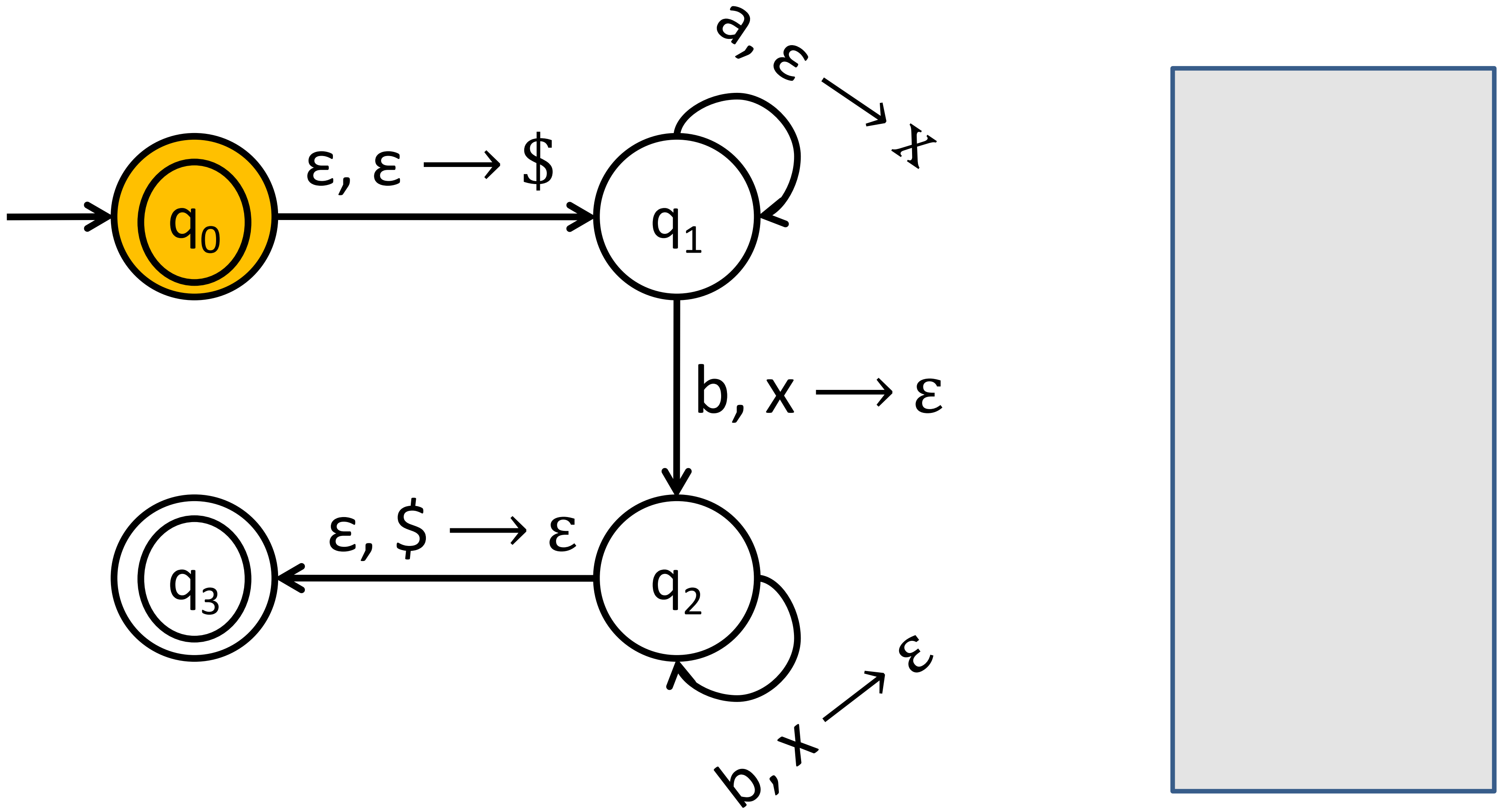
Visualization of $\{a^n b^n : n \geq 0\}$

aaabbb ✓



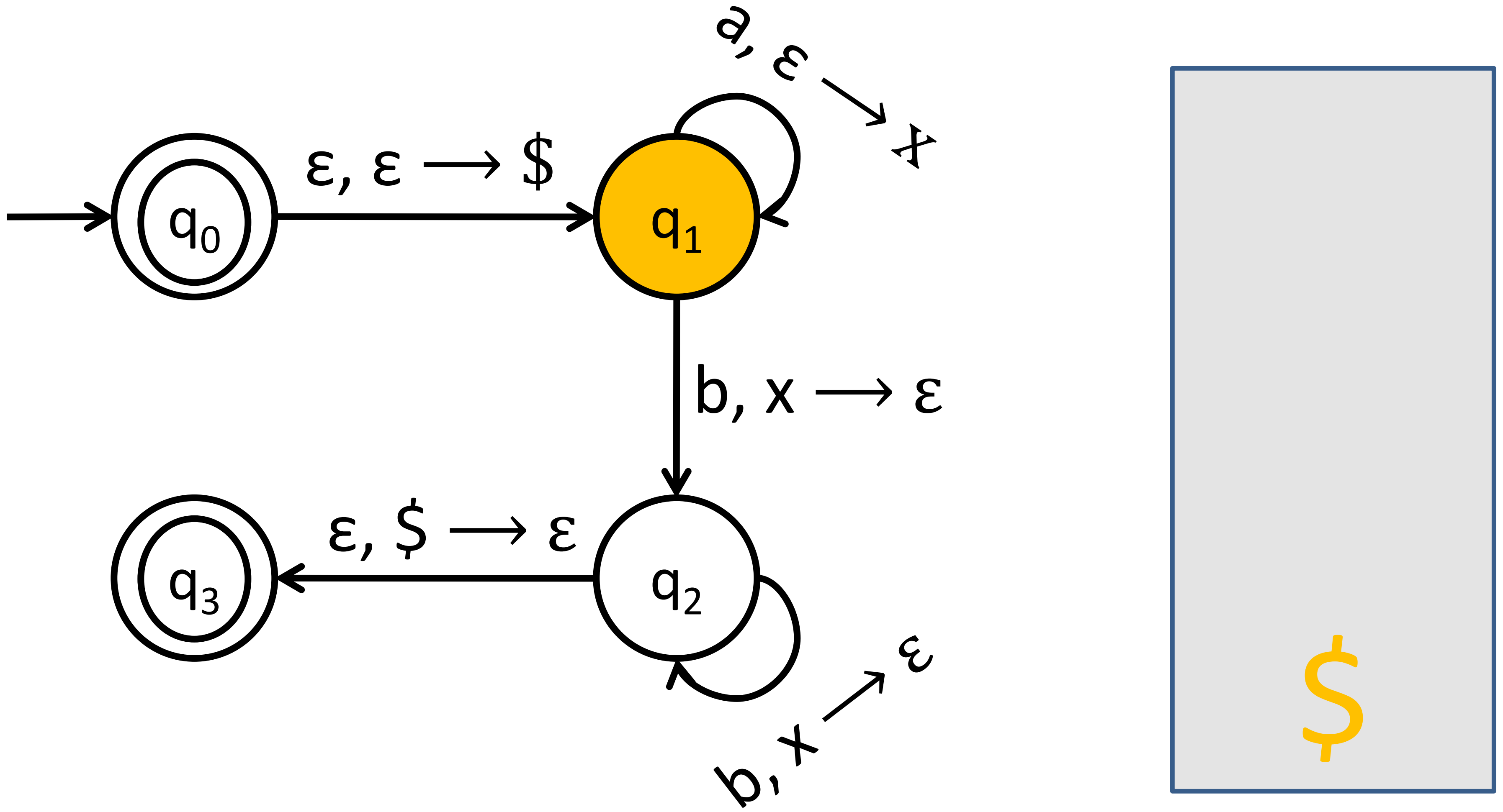
Visualization of $\{a^n b^n : n \geq 0\}$

aab



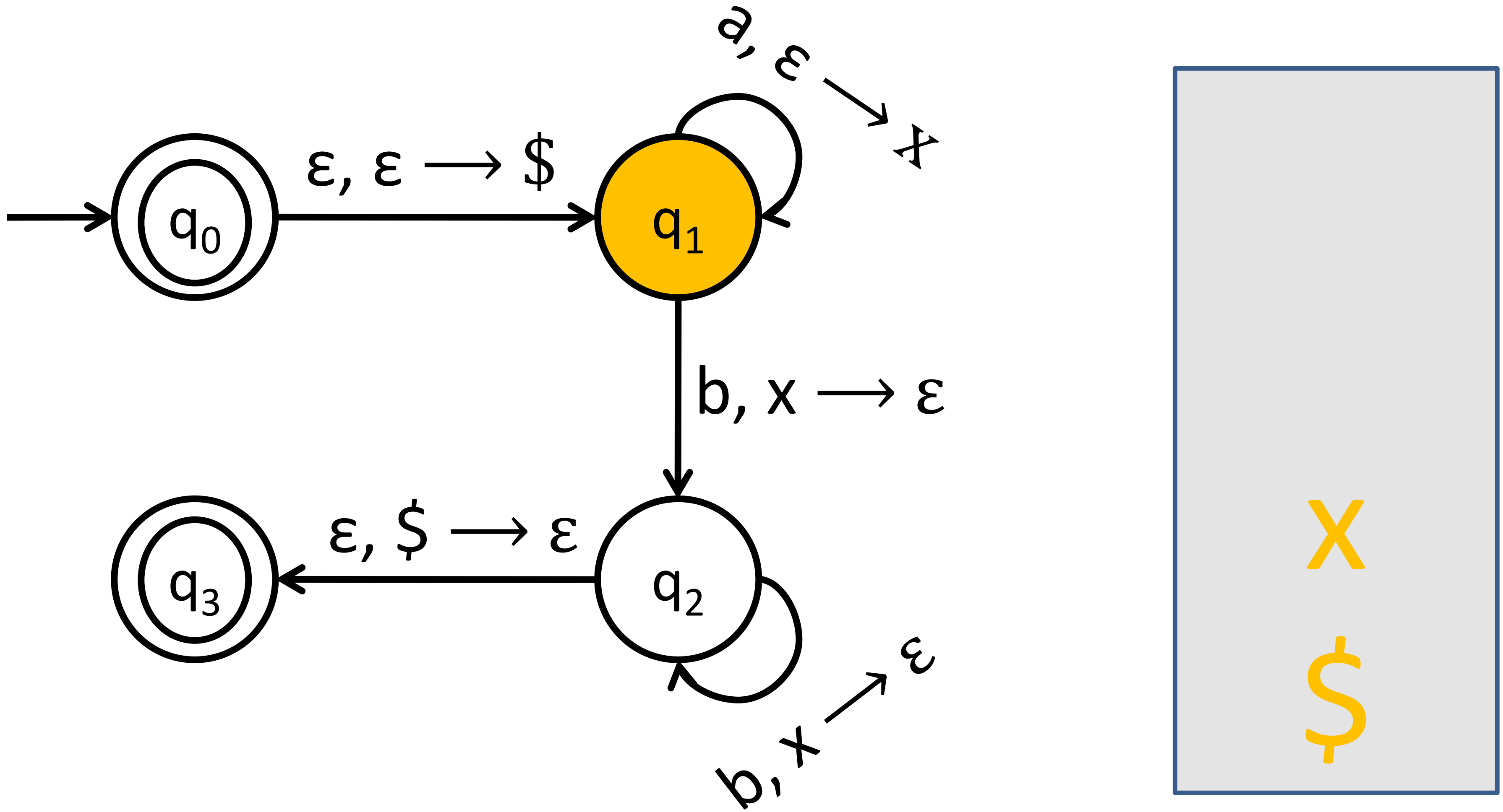
Visualization of $\{a^n b^n : n \geq 0\}$

aab



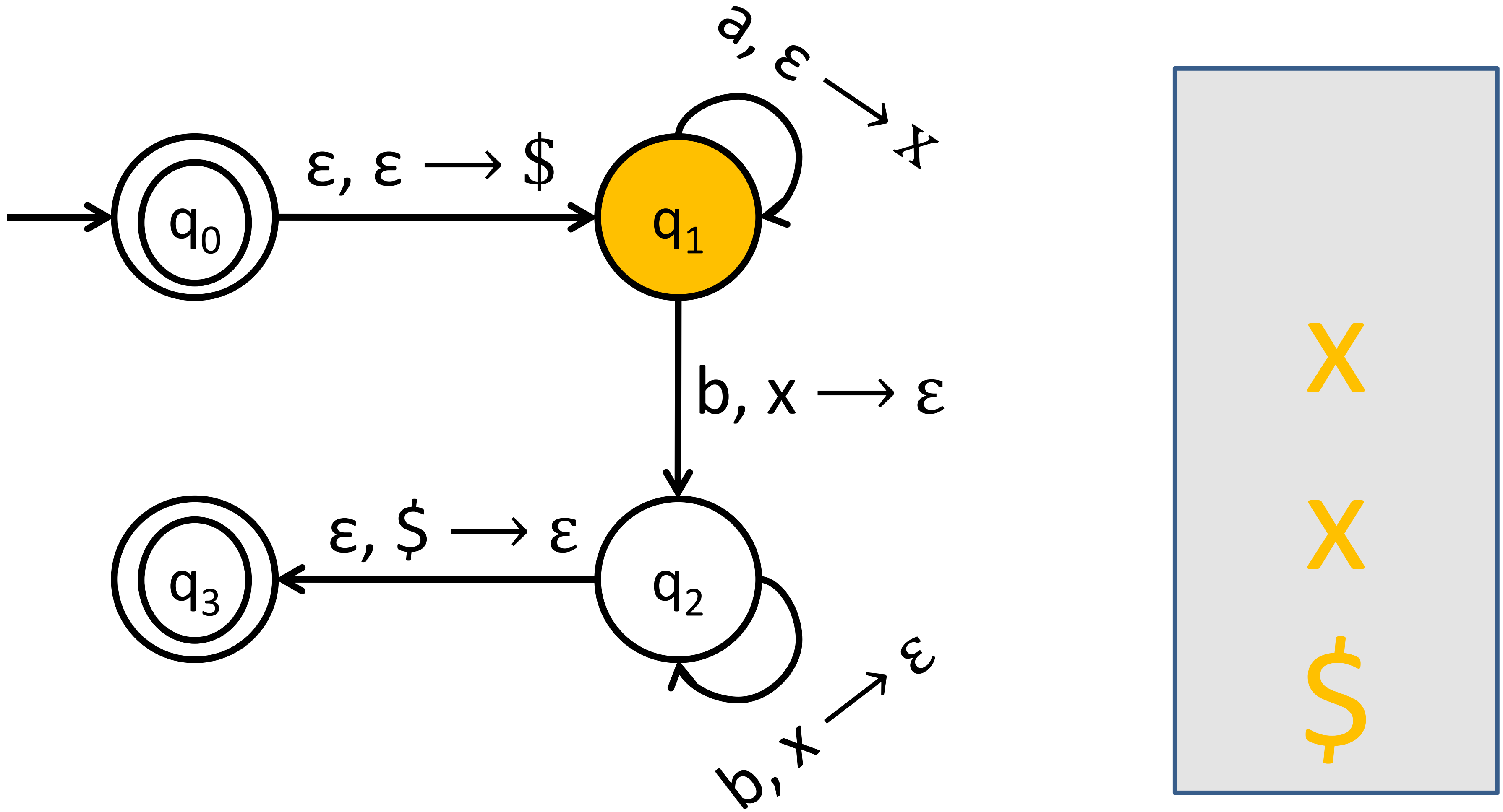
Visualization of $\{a^n b^n : n \geq 0\}$

aab



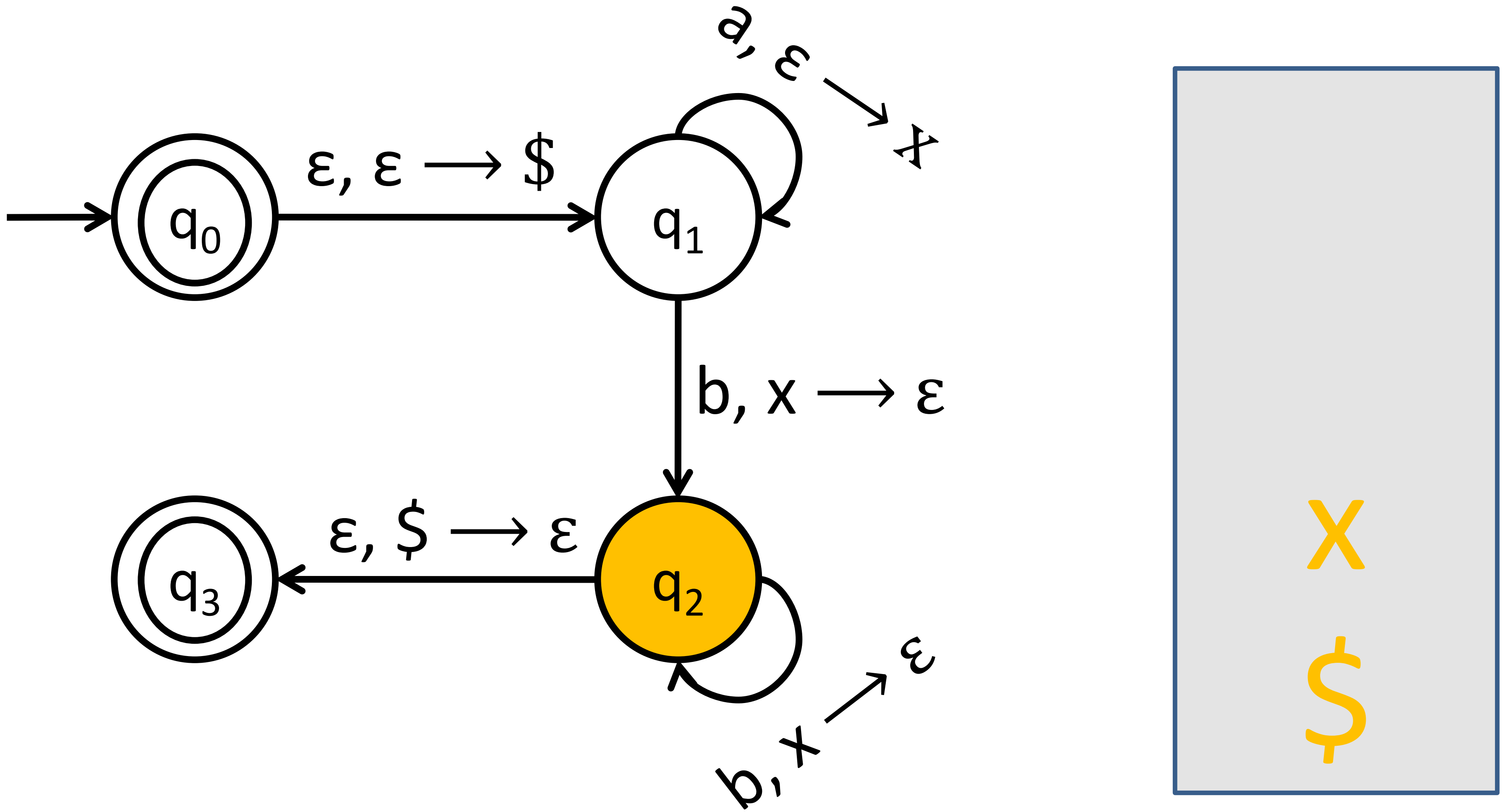
Visualization of $\{a^n b^n : n \geq 0\}$

aab



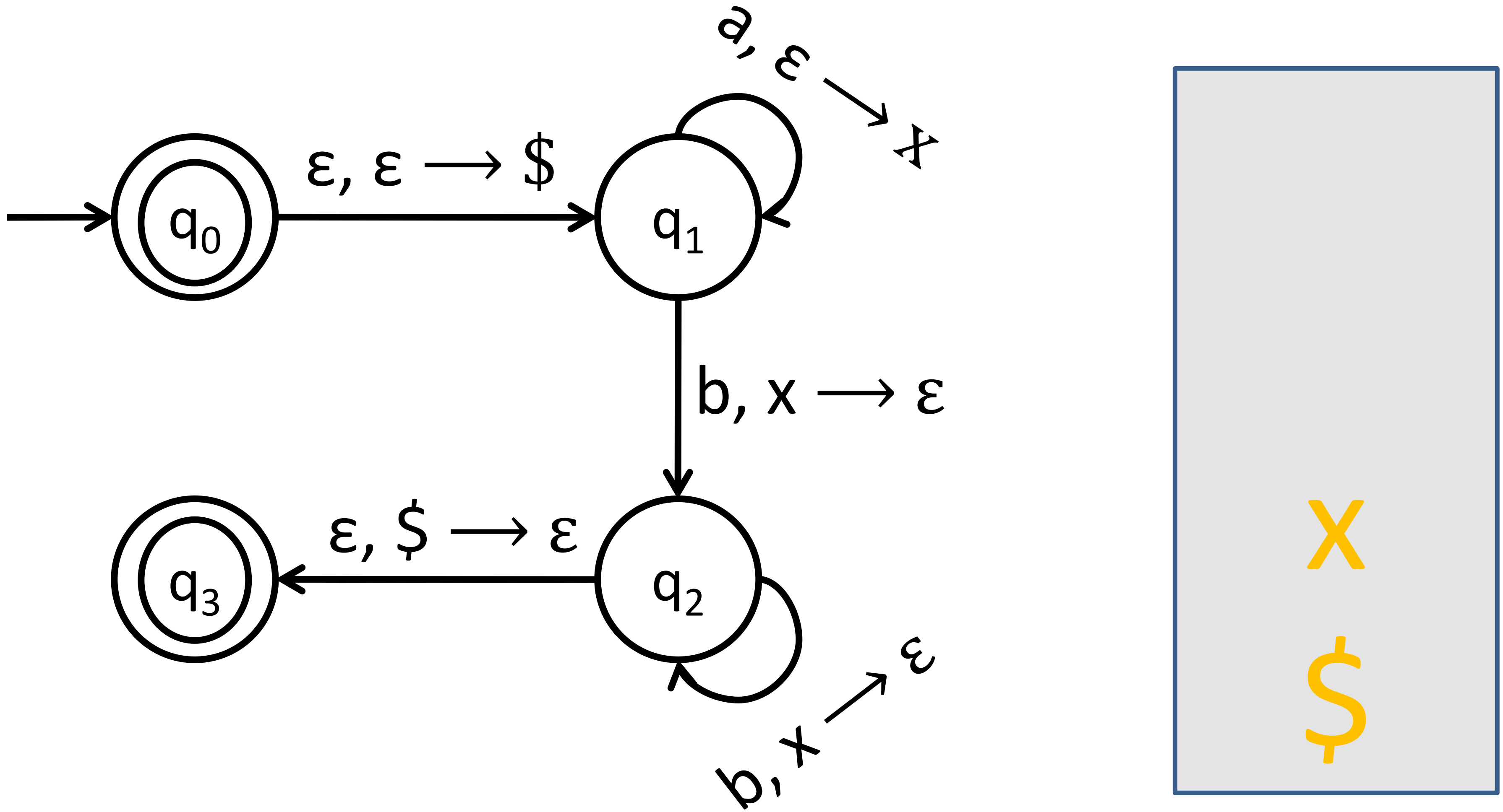
Visualization of $\{a^n b^n : n \geq 0\}$

aab



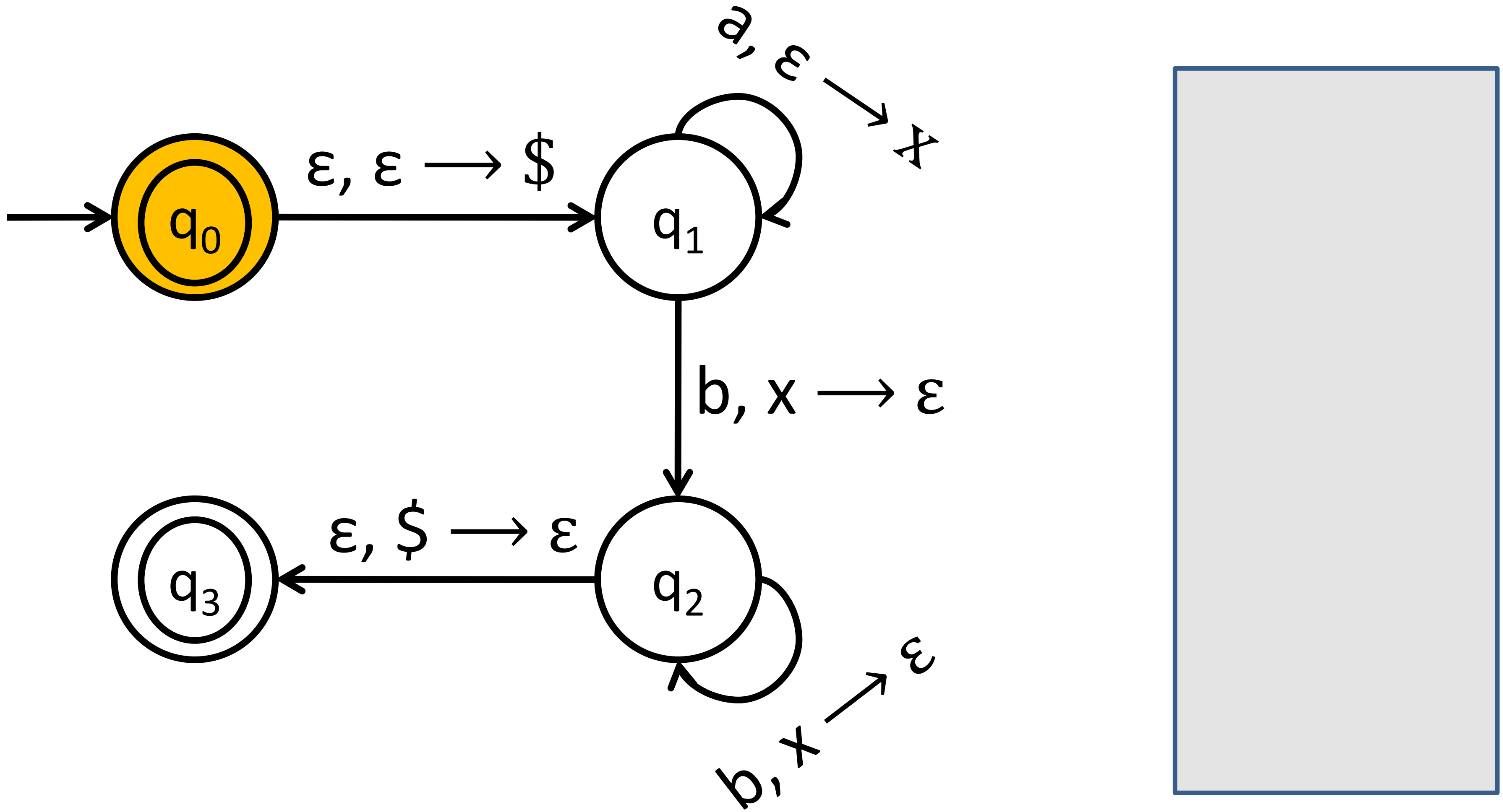
Visualization of $\{a^n b^n : n \geq 0\}$

aab ✕



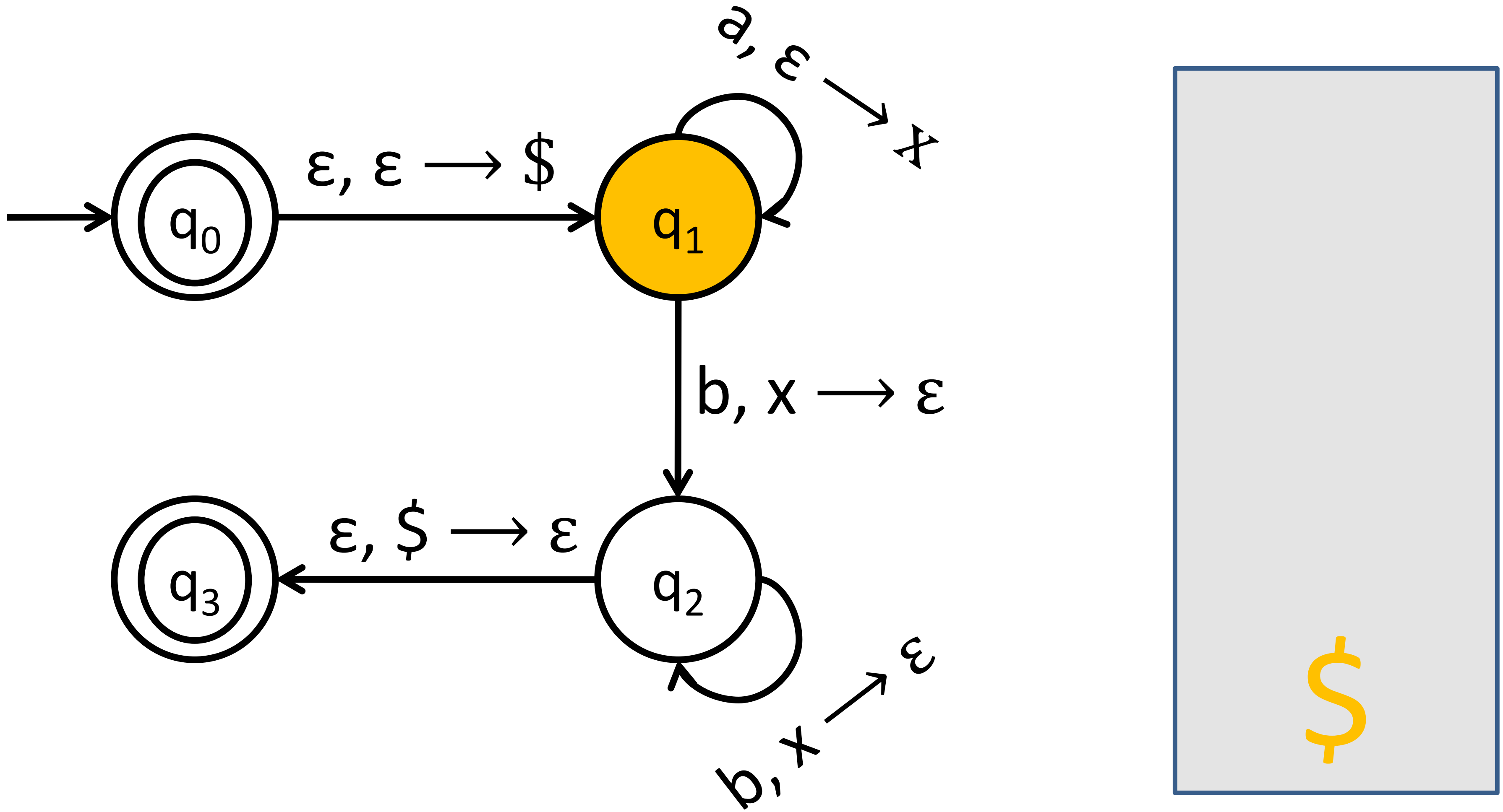
Visualization of $\{a^n b^n : n \geq 0\}$

abb



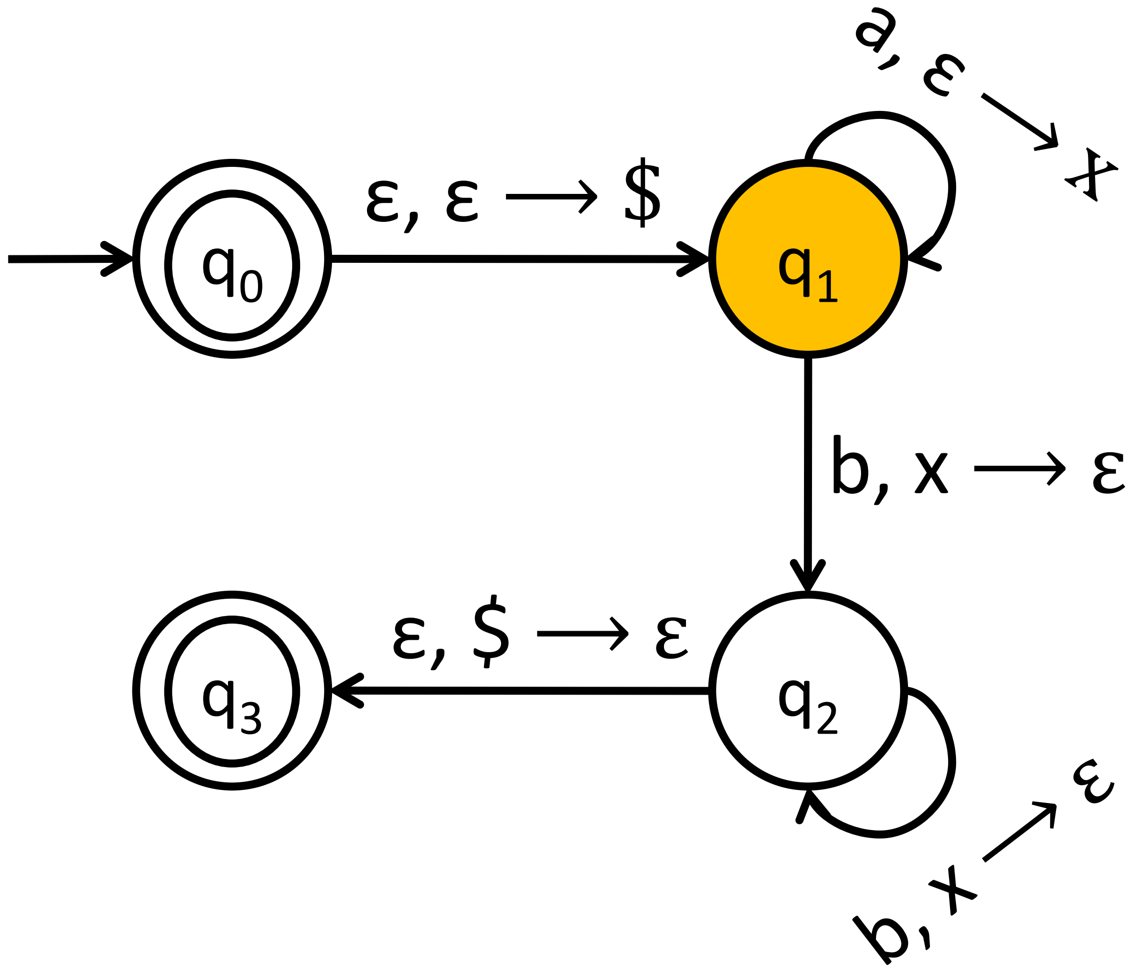
Visualization of $\{a^n b^n : n \geq 0\}$

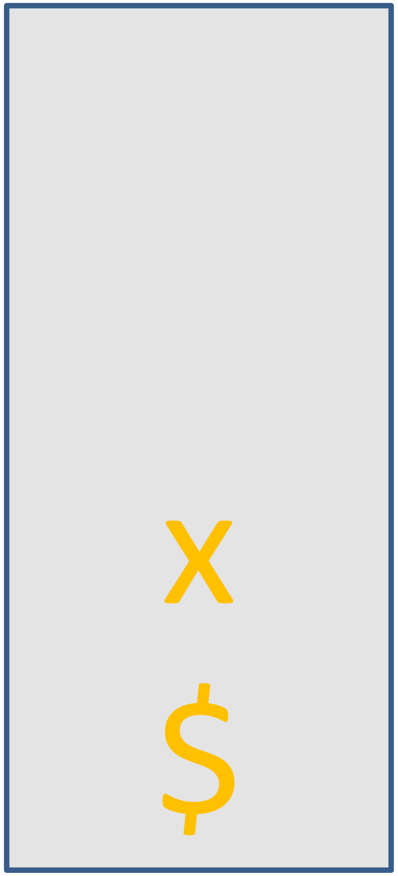
abb



Visualization of $\{a^n b^n : n \geq 0\}$

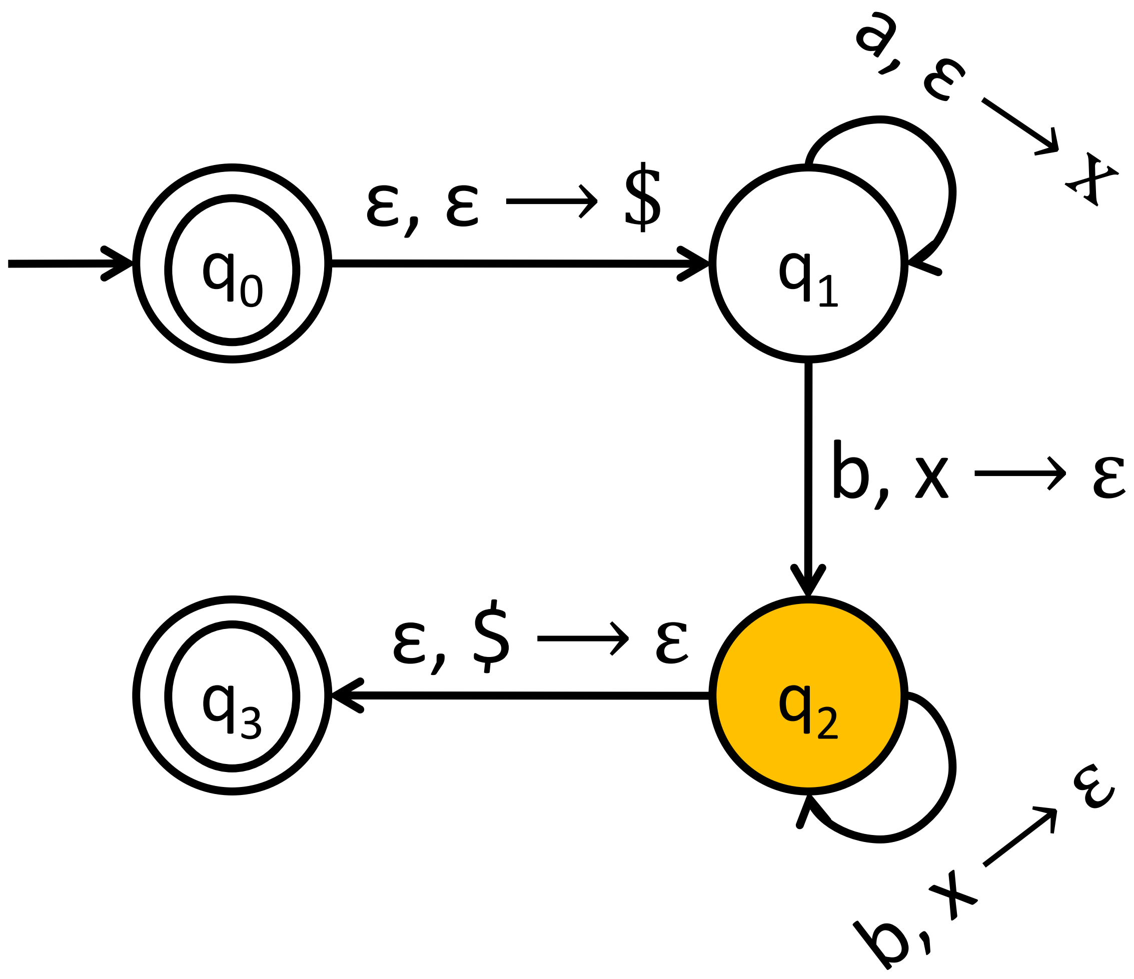
abb

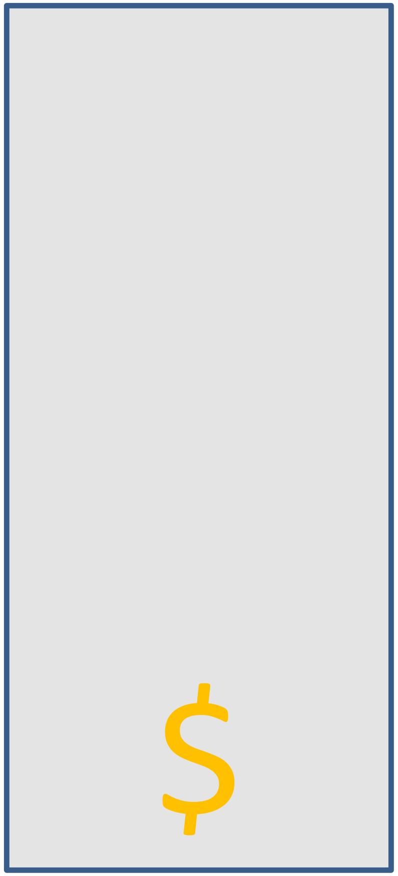




Visualization of $\{a^n b^n : n \geq 0\}$

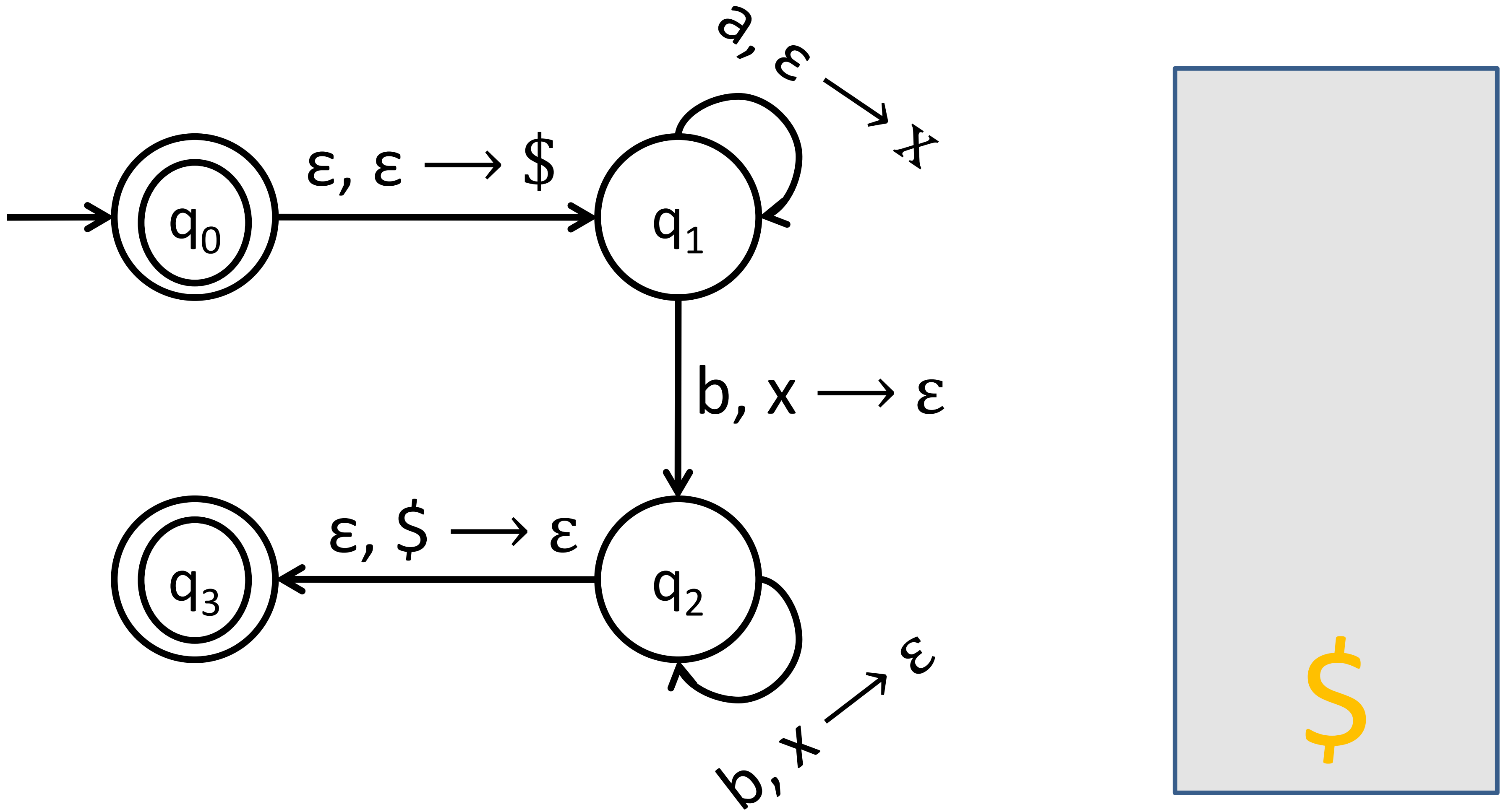
abb





Visualization of $\{a^n b^n : n \geq 0\}$

abb ✗



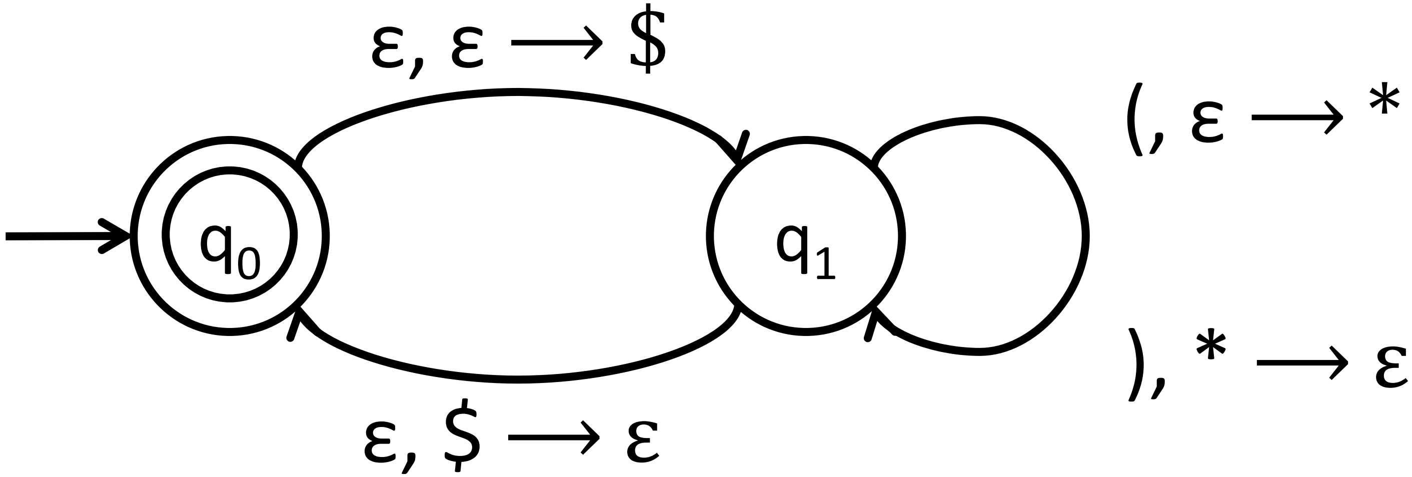
PDA formally

- A PDA is a sextuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$, where:
  - Q is the set of states
  - Σ is the input alphabet
  - Γ is the alphabet for the stack
  - δ is the transition function
  - q_0 is the start state
  - F is the set of accepting states

About Γ: The stack alphabet can contain any symbol you want. It can be completely disjoint from Σ.

$L_{()}$: proper opening and closing parenthesis

$L_{()}$: proper opening and closing parenthesis

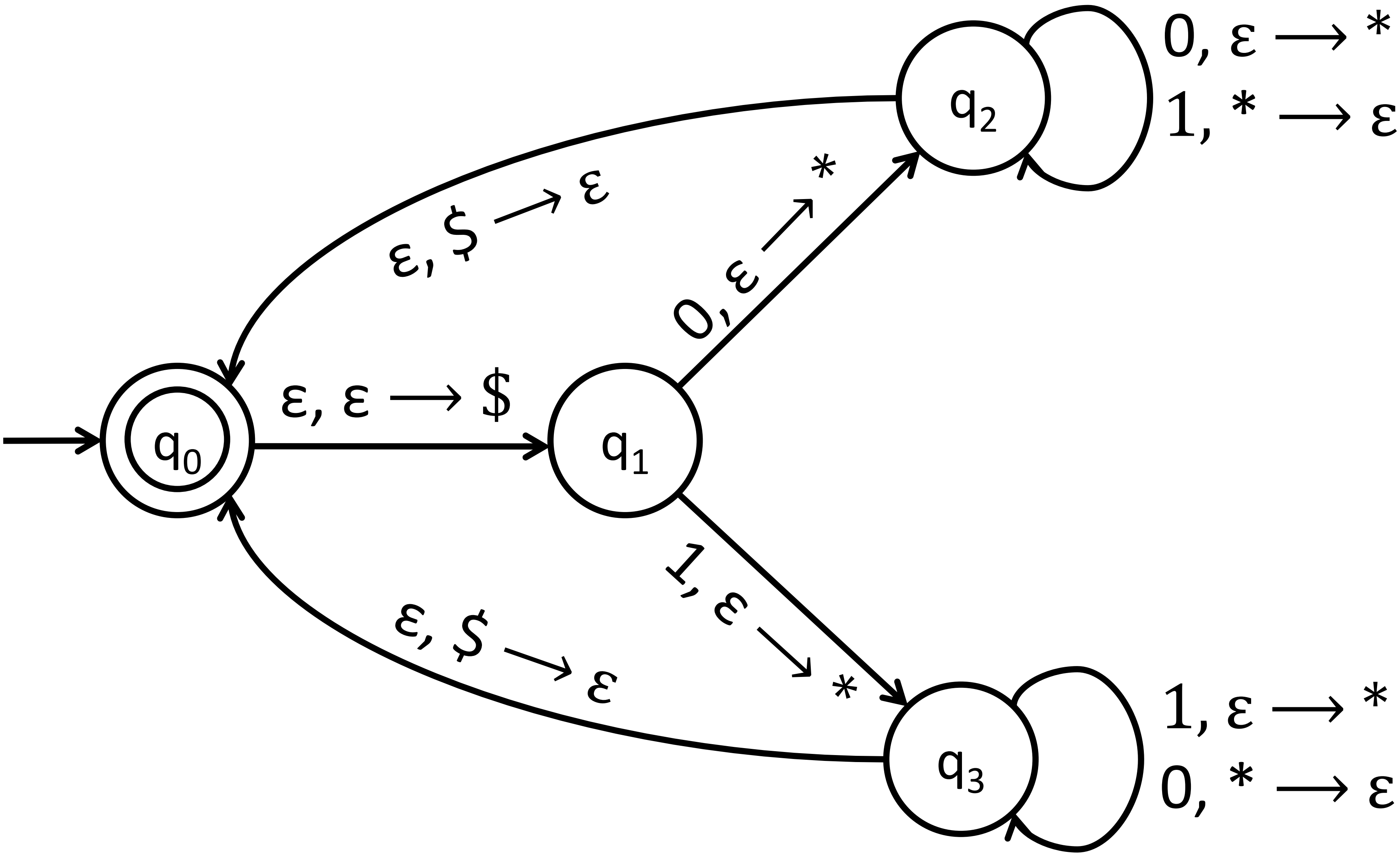


Try it yourself

- Create a PDA for the language:

  $L_= = \{w : w \text{ contains an equal number of 0s and 1s}\}$

$L_=$: equal number of 0s and 1s



$L_=$: equal number of 0s and 1s

NPDA for this language