CS103
Winter 2025

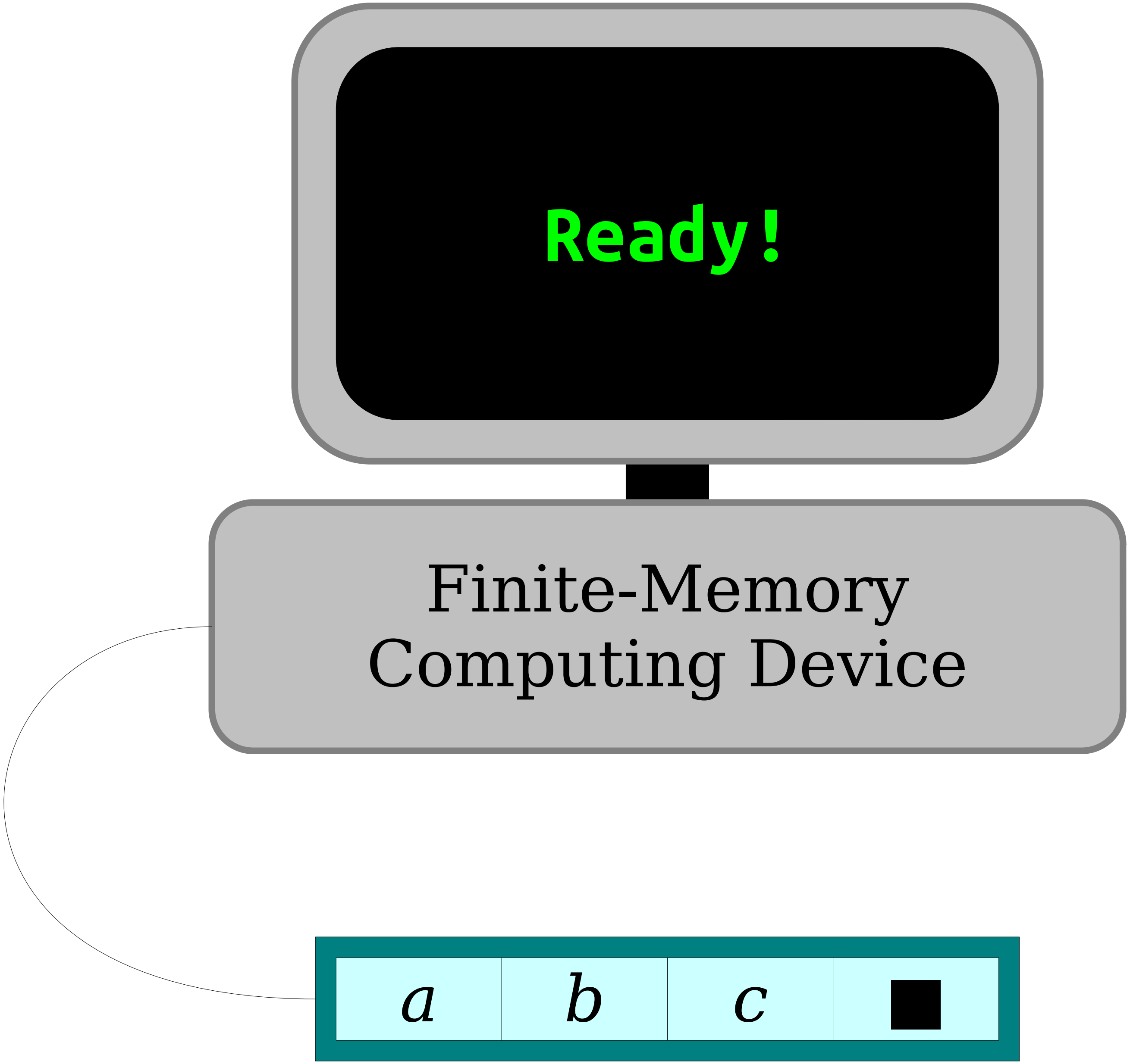
Lecture 18:

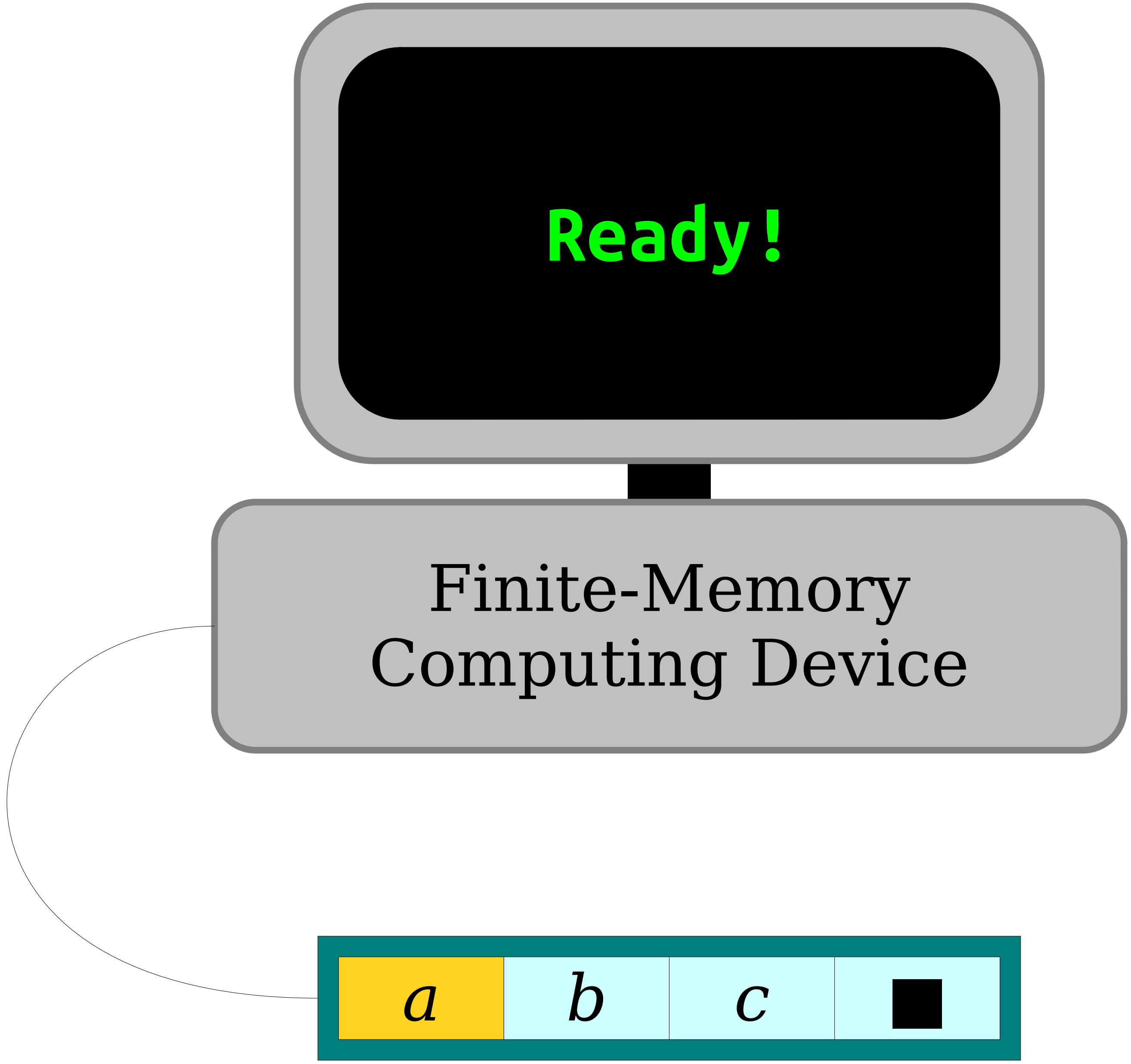
Nonregular Languages

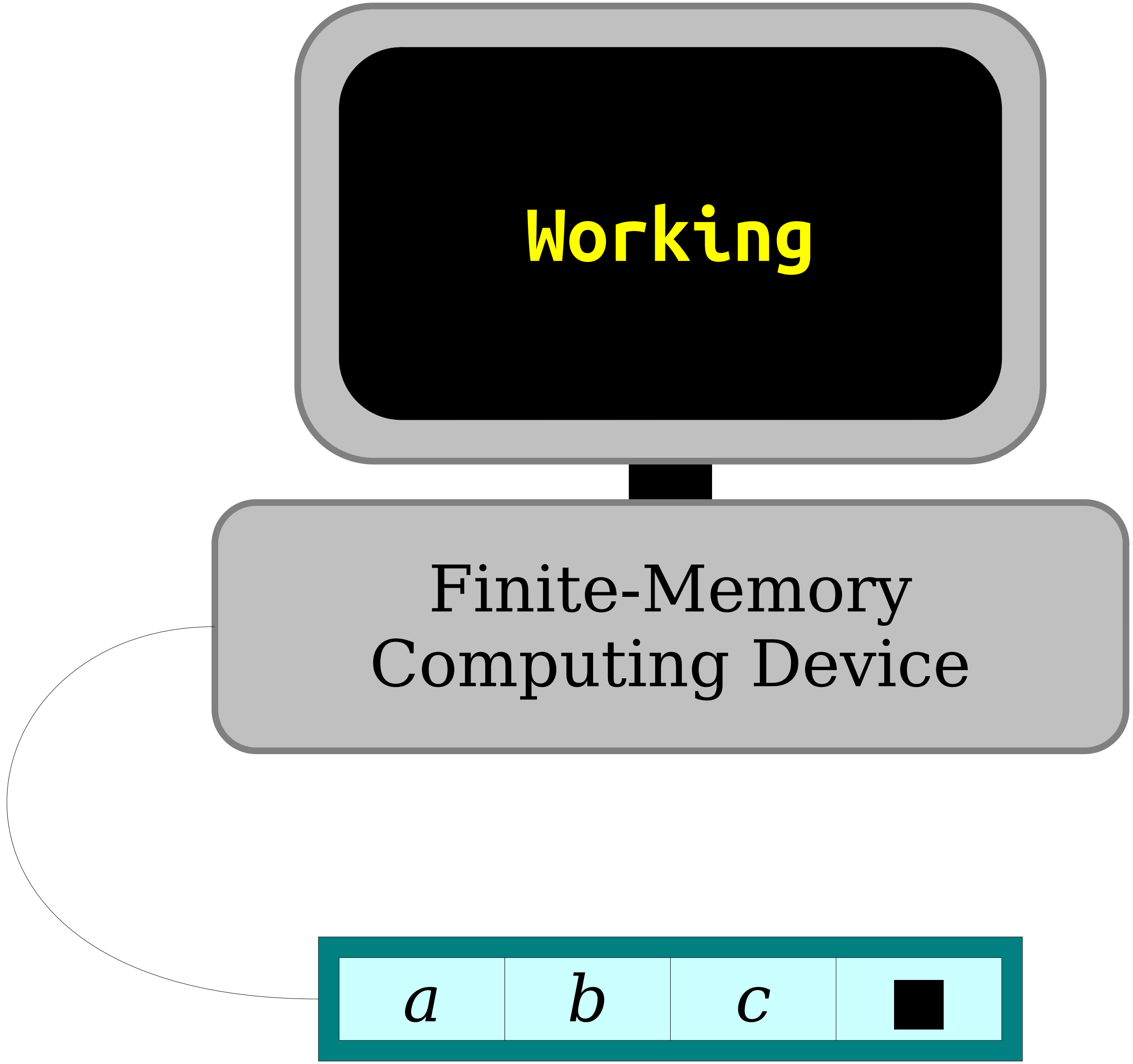
Recap from Last Time

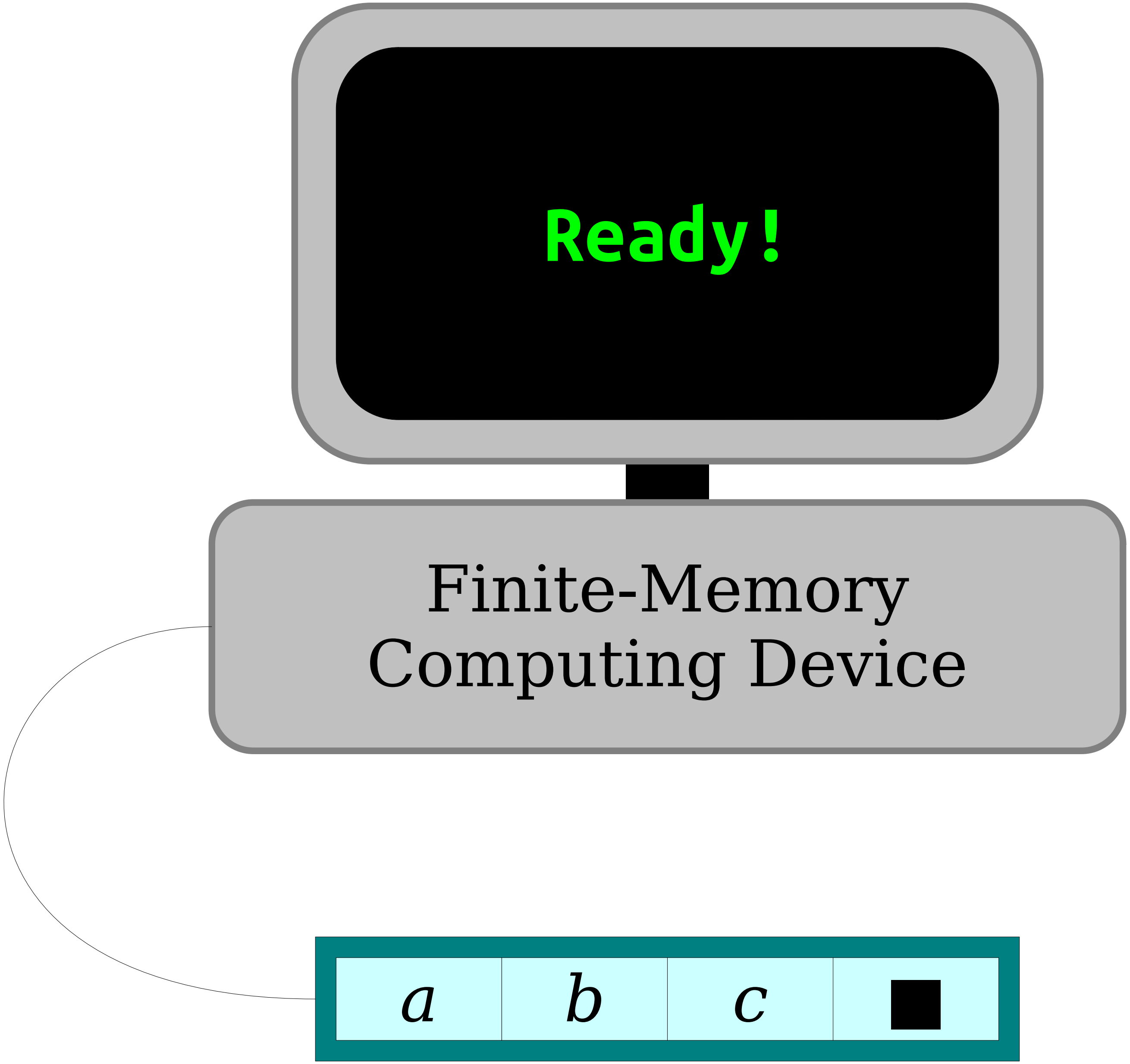
Theorem: The following are all equivalent:

- L is a regular language.
- There is a DFA D such that $\mathscr{L}(D) = L$.
- There is an NFA N such that $\mathscr{L}(N) = L$.
- There is a regular expression R such that $\mathscr{L}(R) = L$.

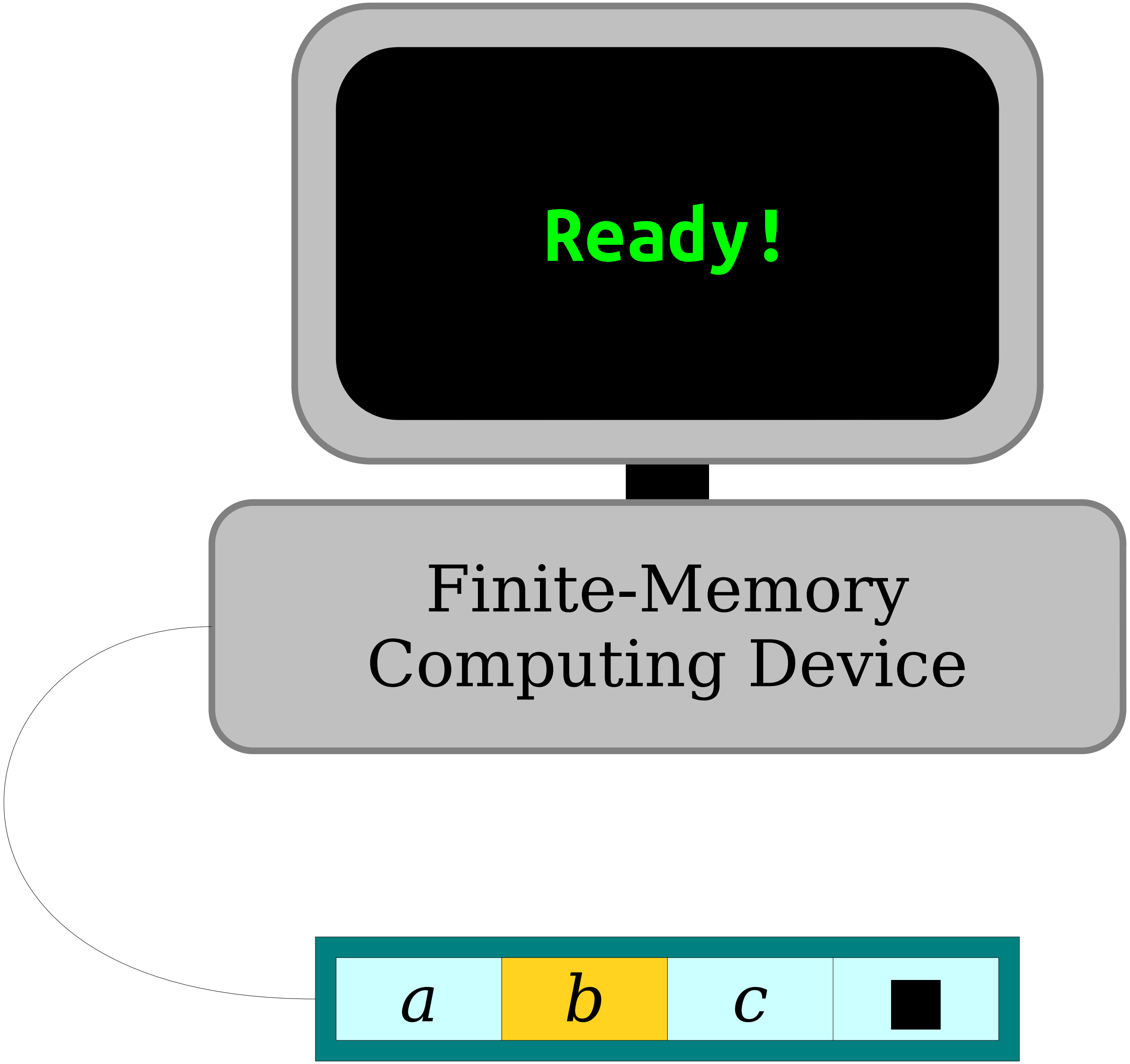

Ready!
Finite-Memory
Computing Device
a
b
c

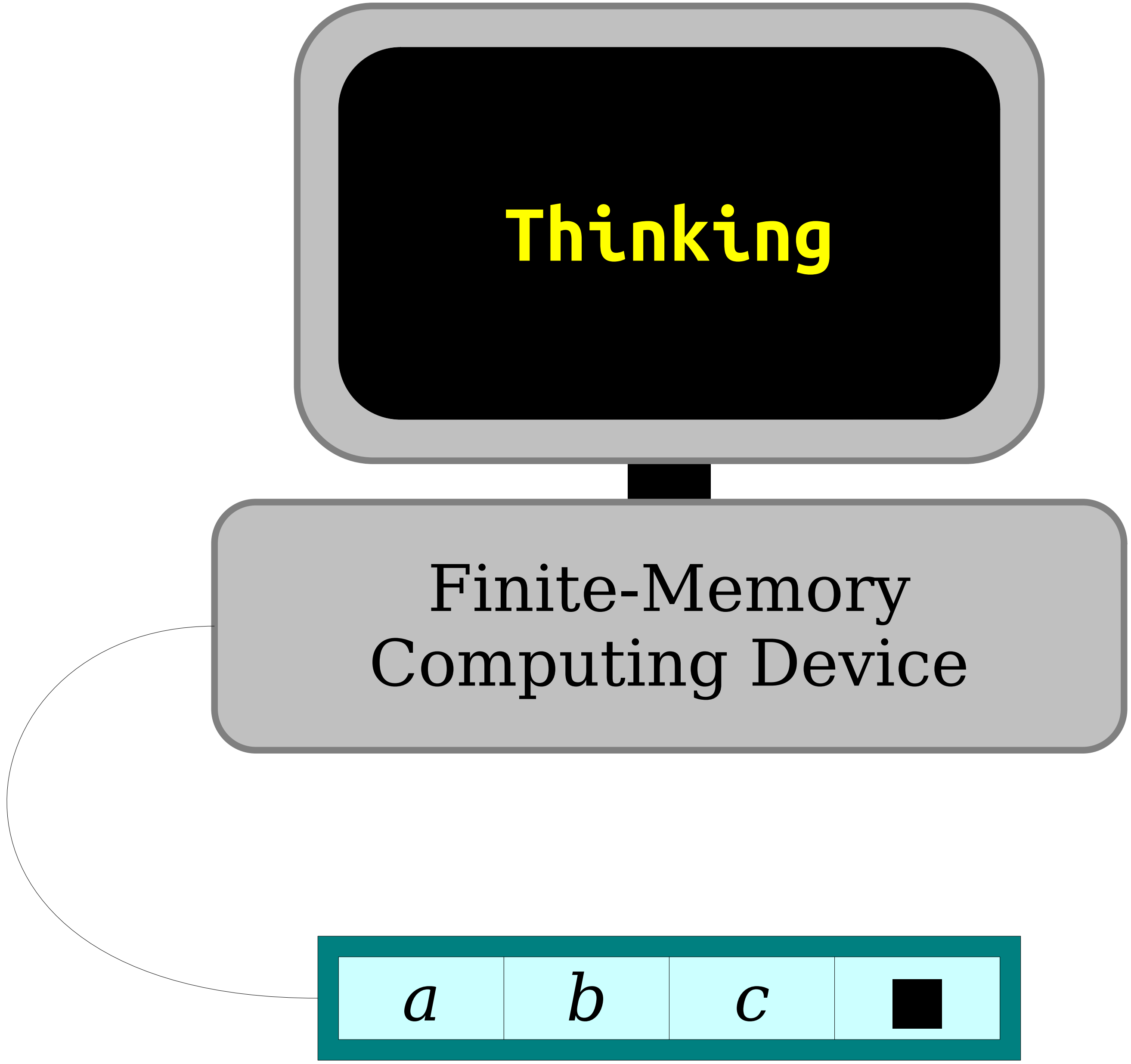

Ready!
Finite-Memory
Computing Device
a
b
c

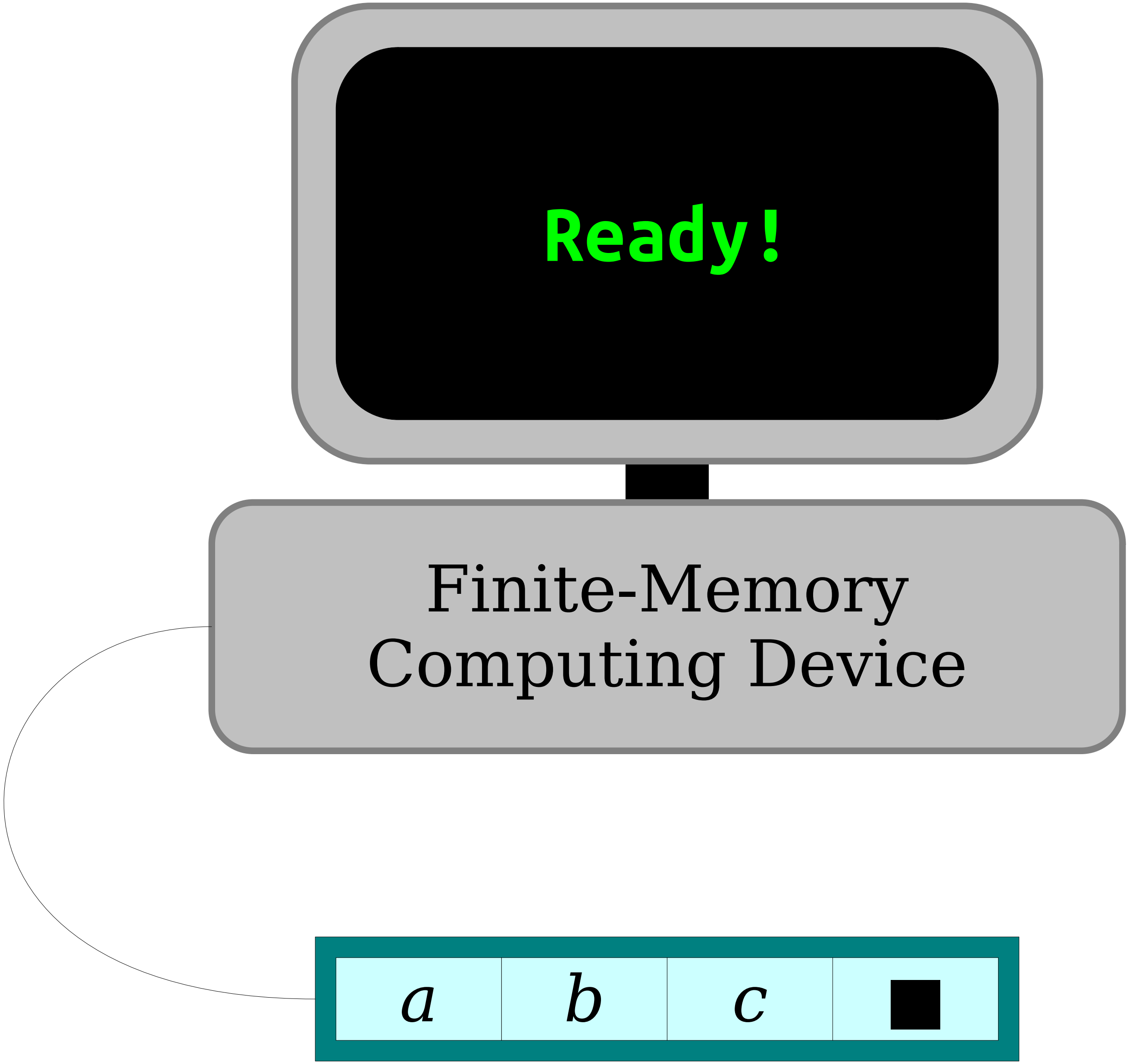

Working
Finite-Memory
Computing Device
a
b
c

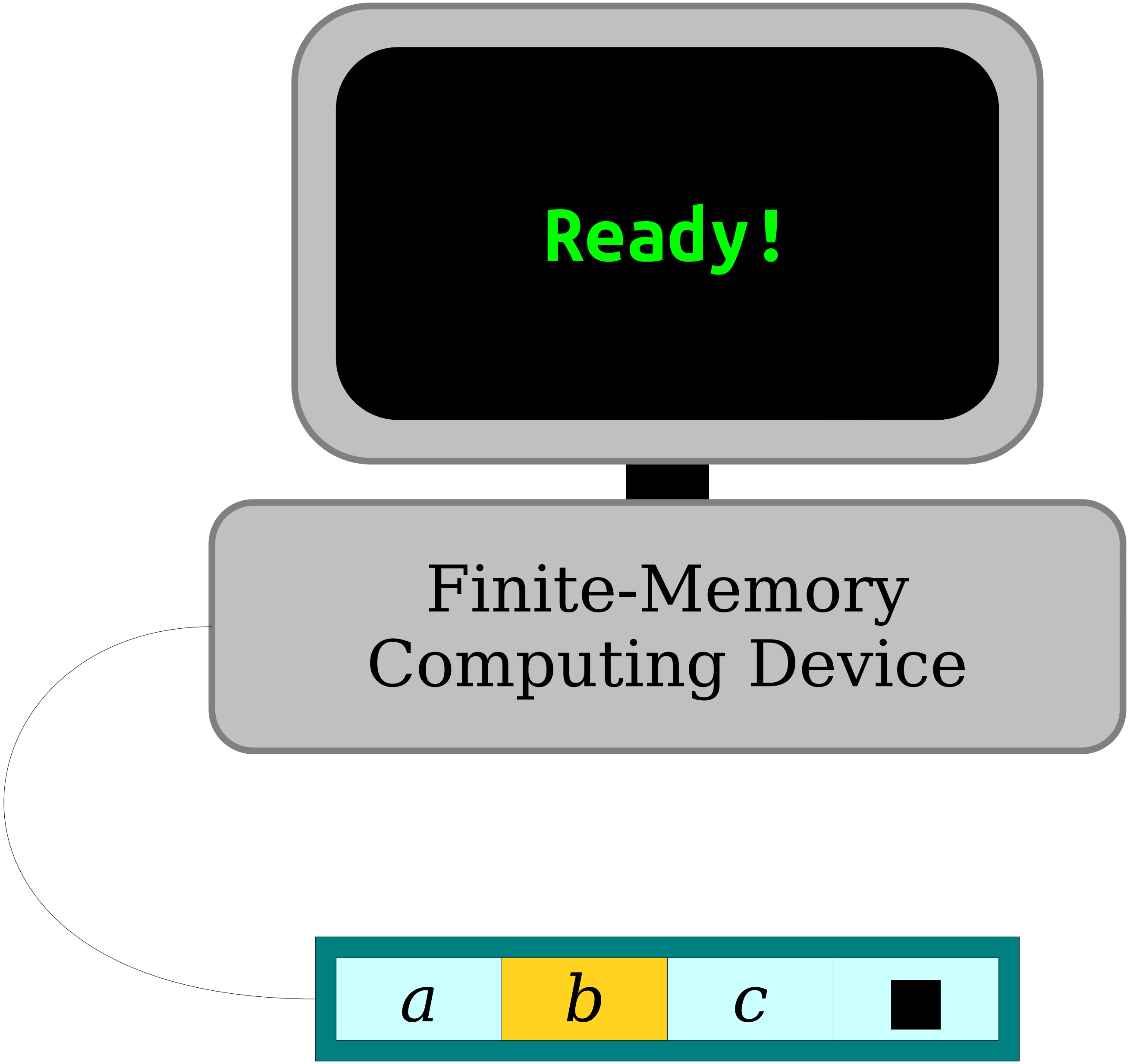


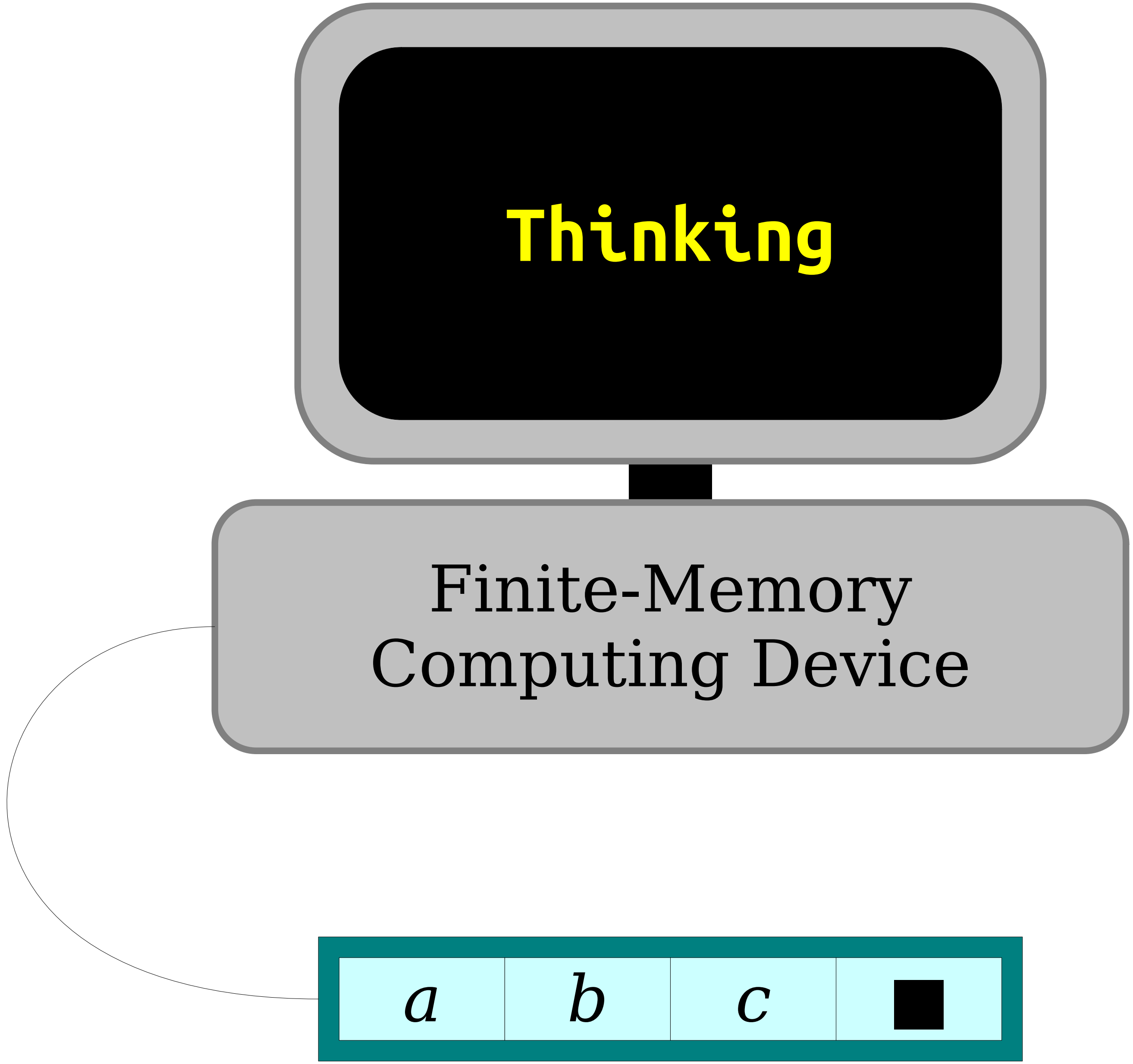
Ready!
Finite-Memory
Computing Device
a
b
c

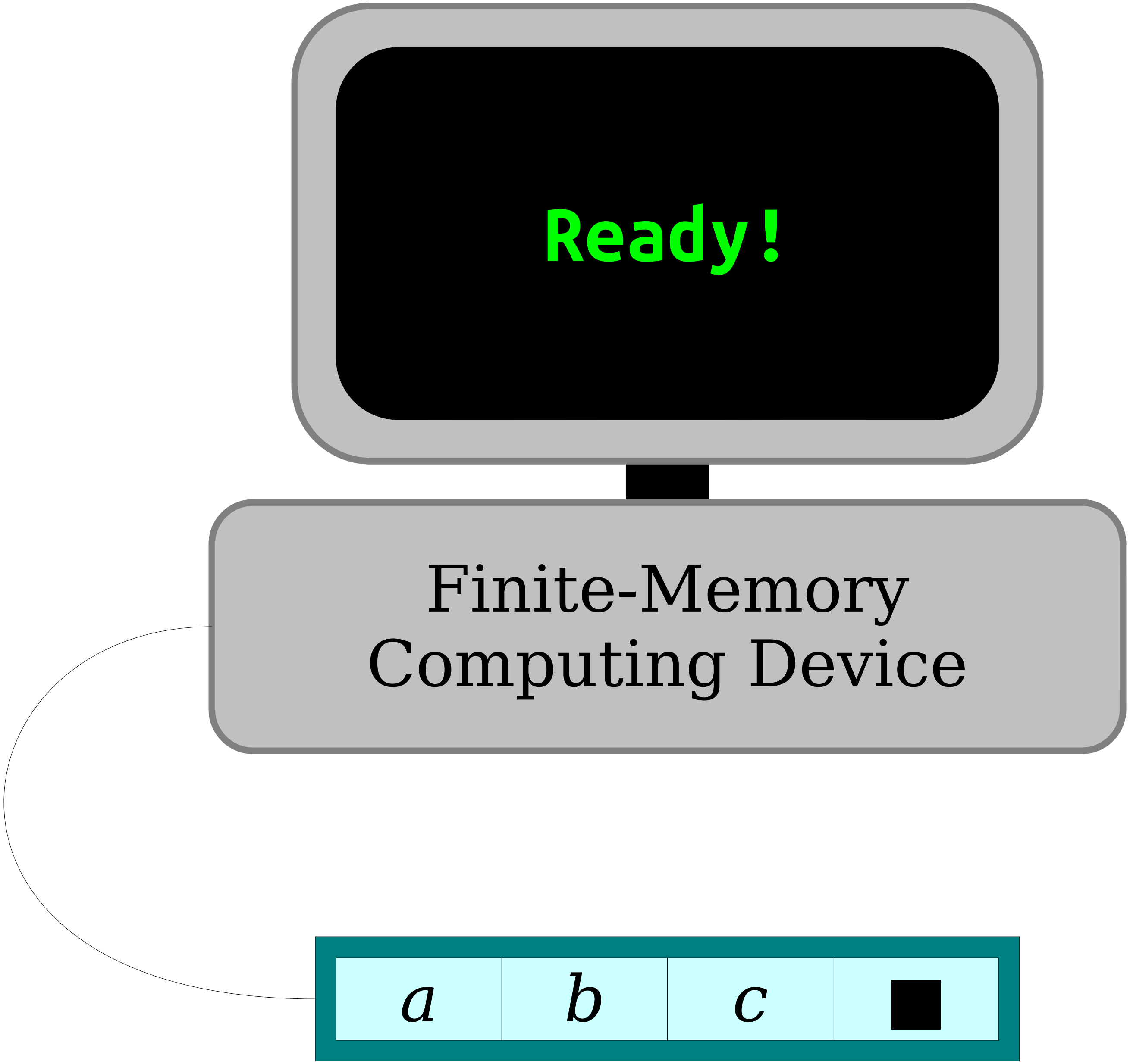

Ready!
Finite-Memory
Computing Device
a
b
c

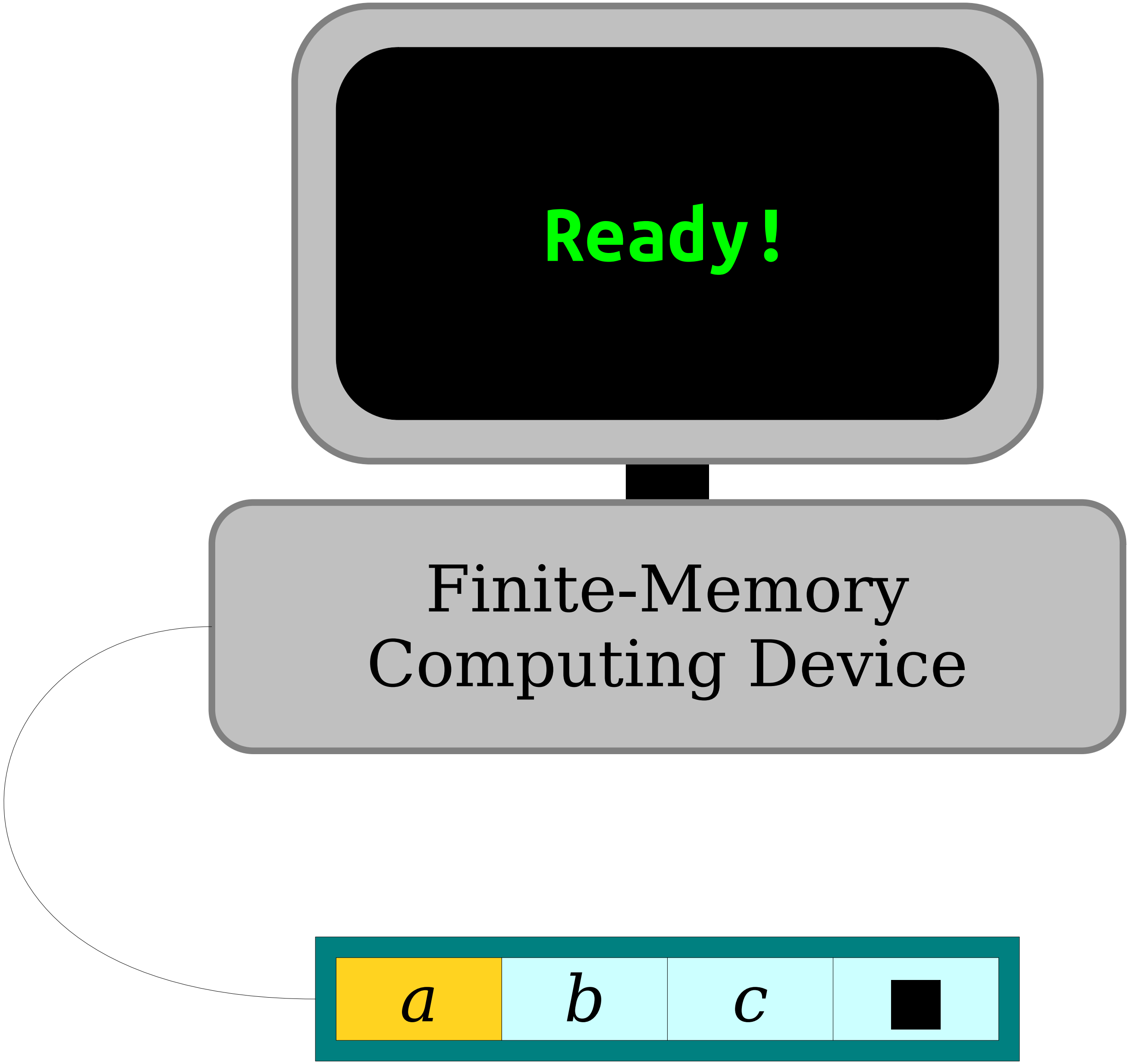

Thinking
Finite-Memory
Computing Device
a
b
c

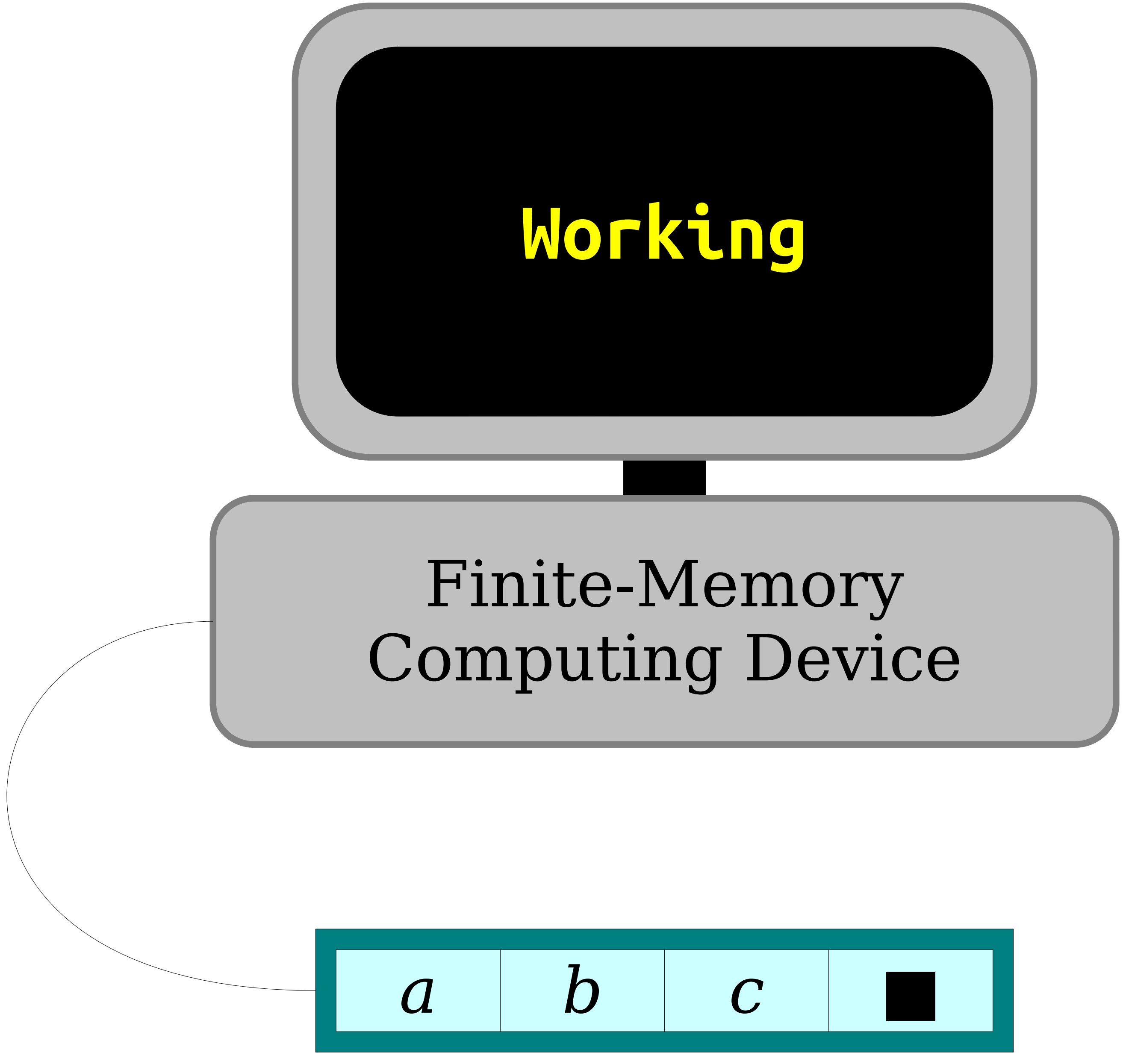

Ready!
Finite-Memory
Computing Device
a
b
c

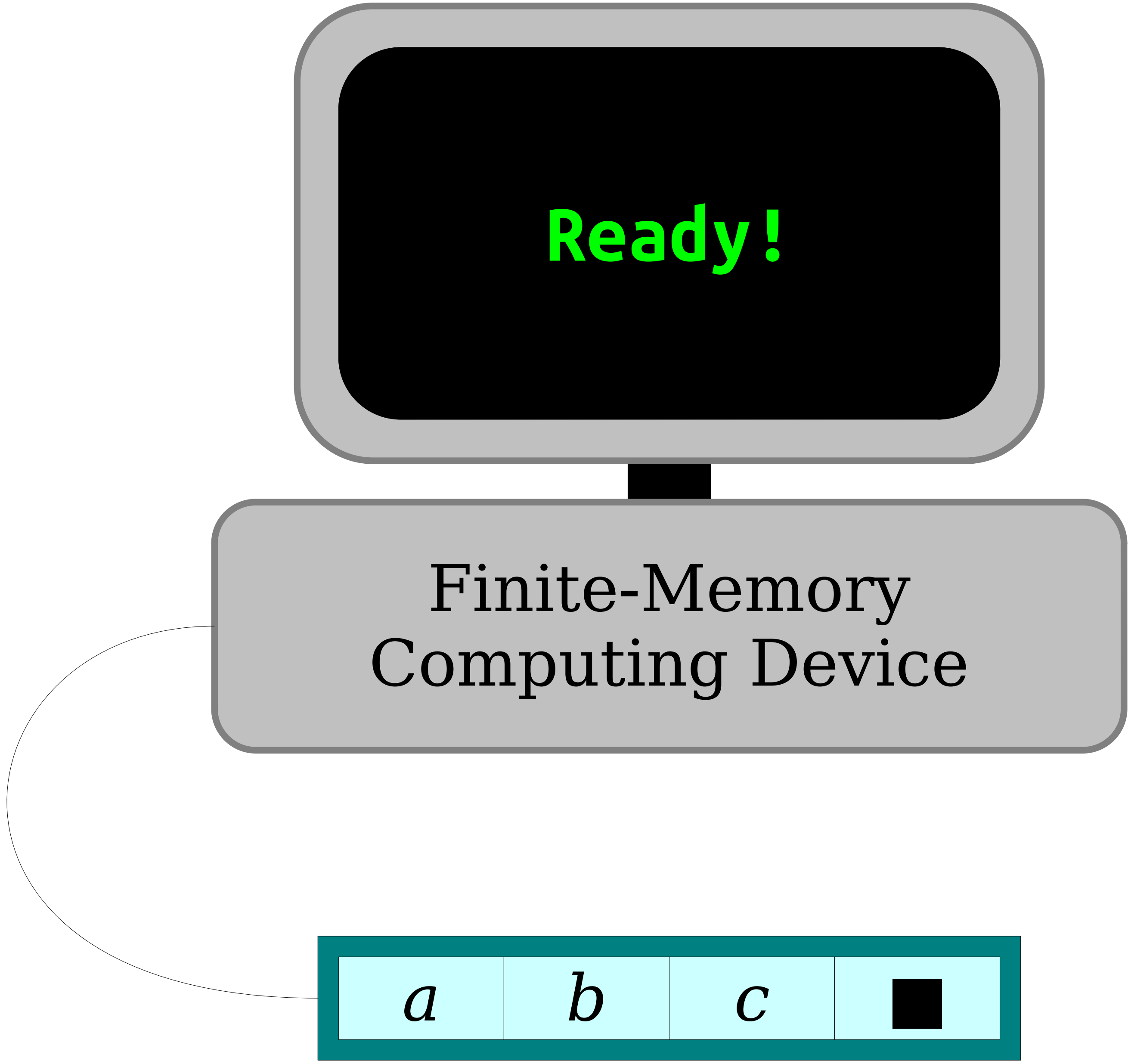

Ready!
Finite-Memory
Computing Device
a
b
c

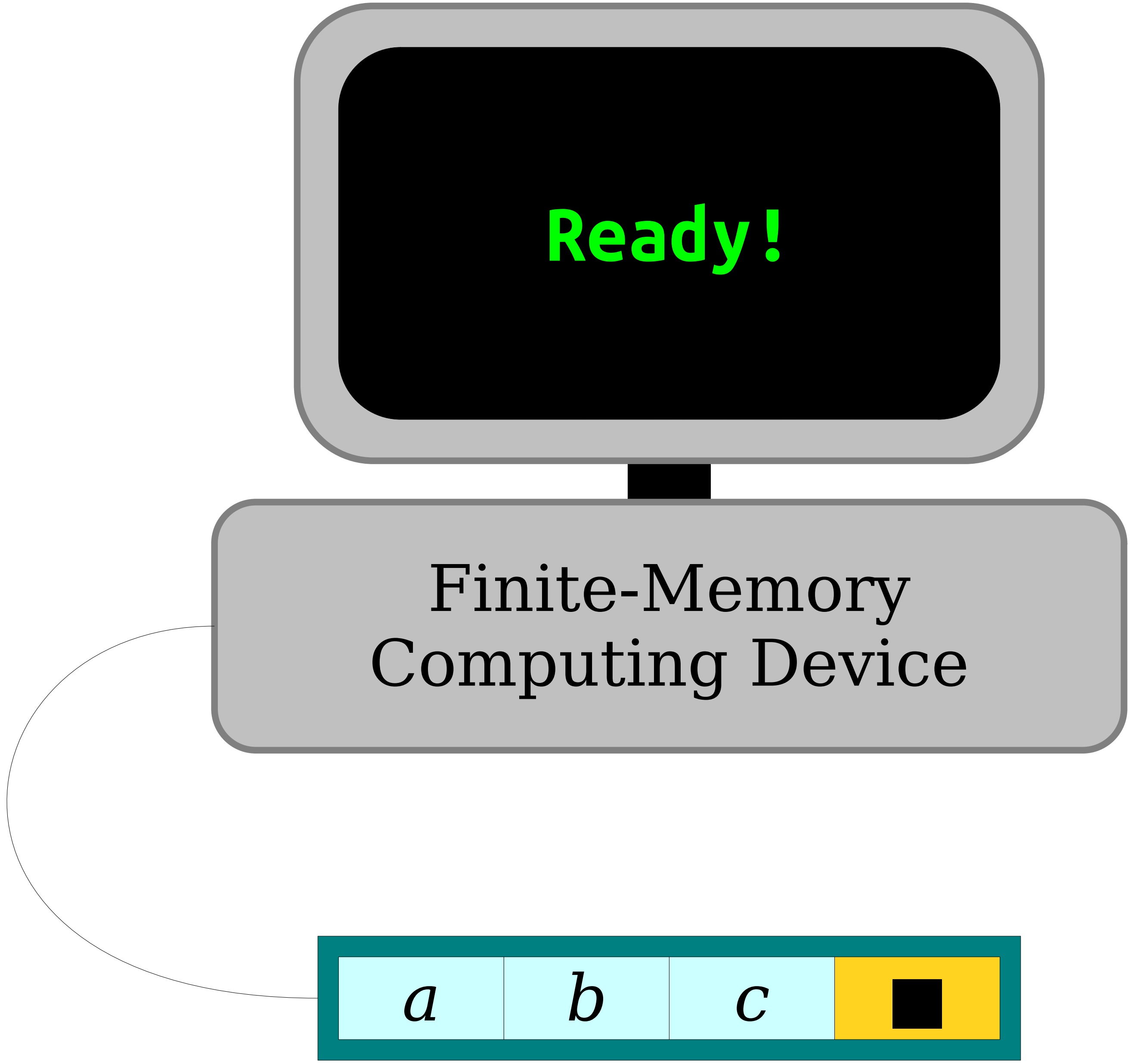

Thinking
Finite-Memory
Computing Device
a
b
c

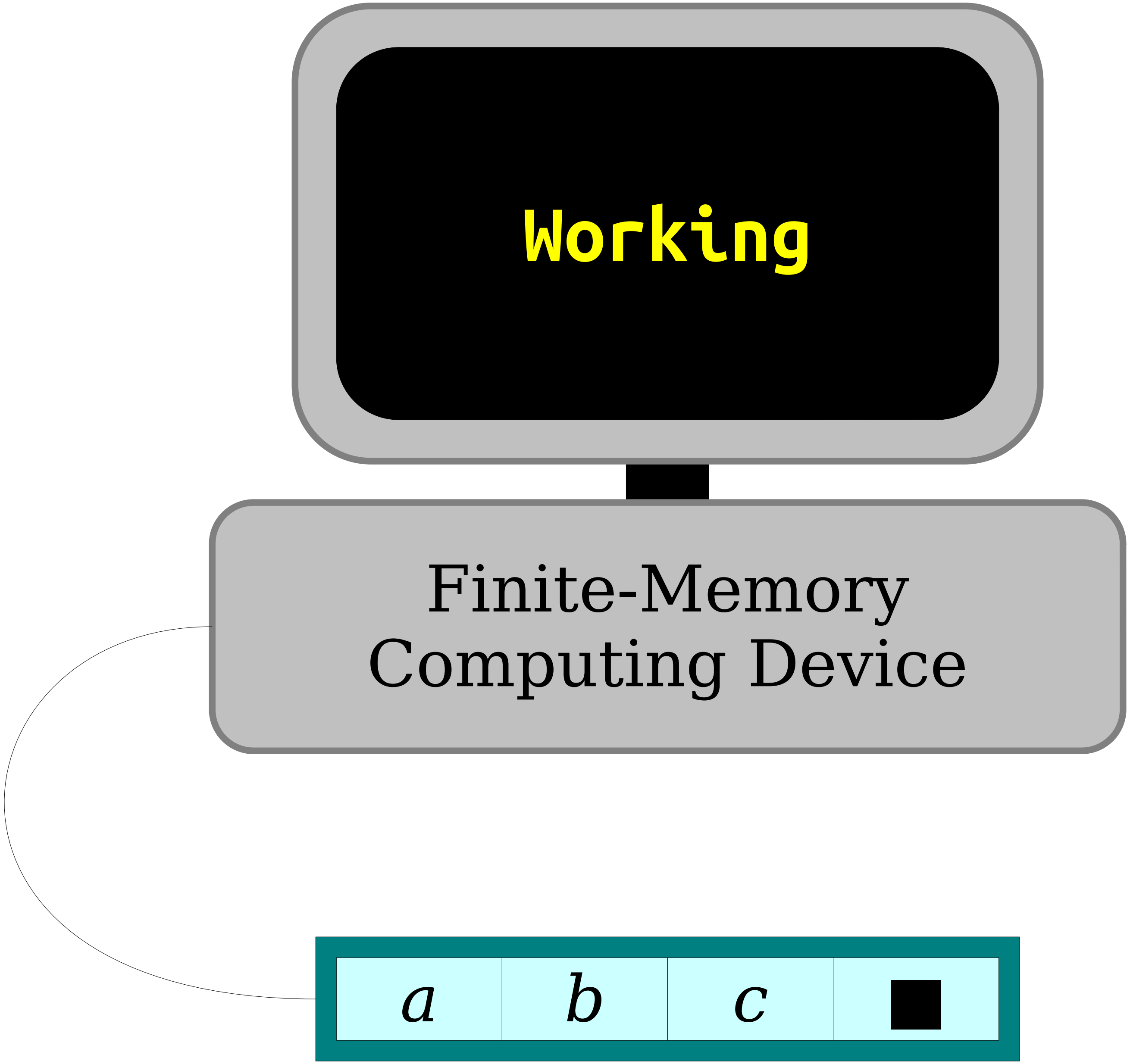

Ready!
Finite-Memory
Computing Device
a
b
c

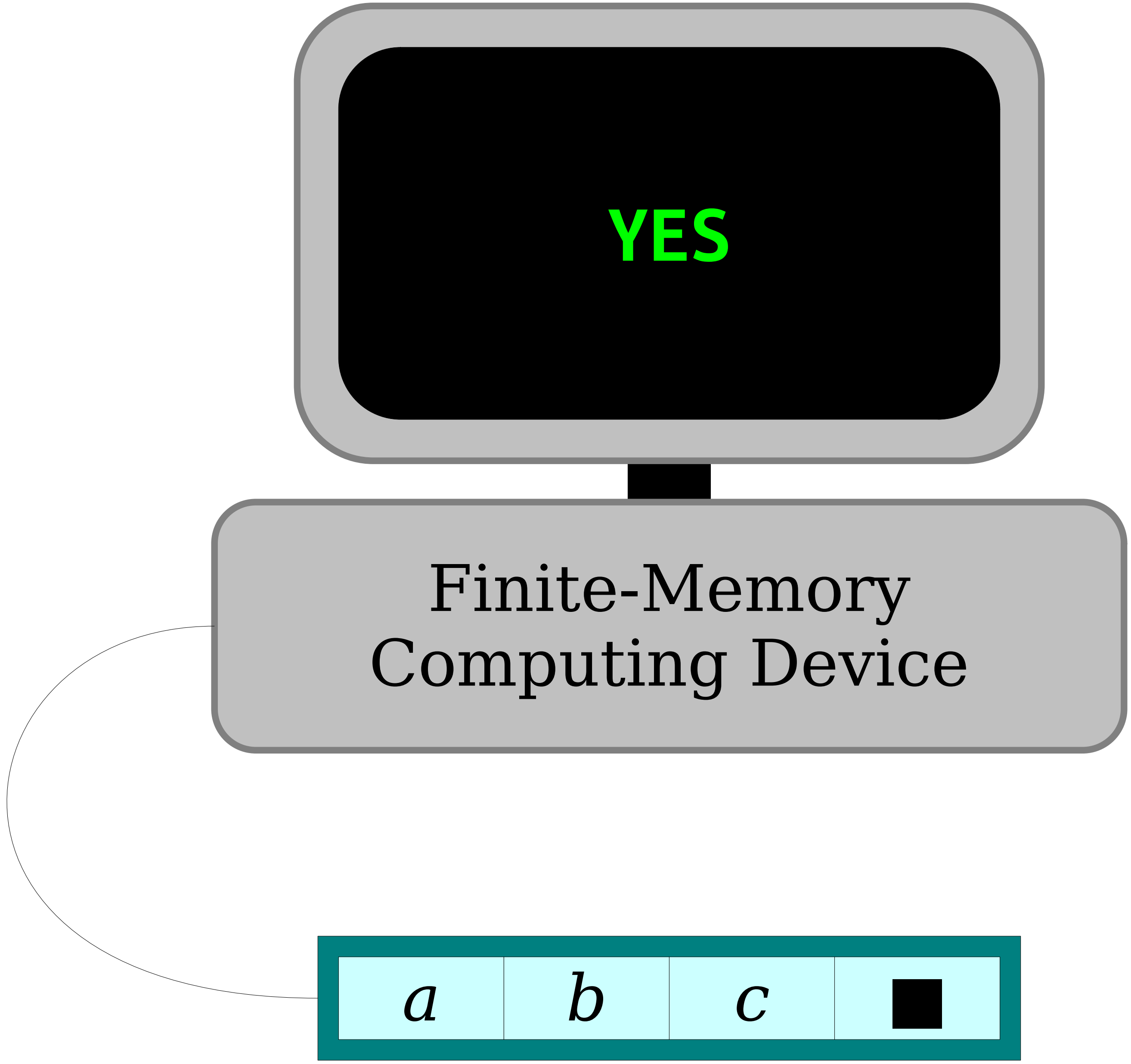

Ready!
Finite-Memory
Computing Device
a
b
c


Working
Finite-Memory
Computing Device
a
b
c


Ready!
Finite-Memory Computing Device
a
b
c


Ready!
Finite-Memory
Computing Device
a
b
c


Working
Finite-Memory
Computing Device
a
b
c


YES
Finite-Memory
Computing Device
a
b
c

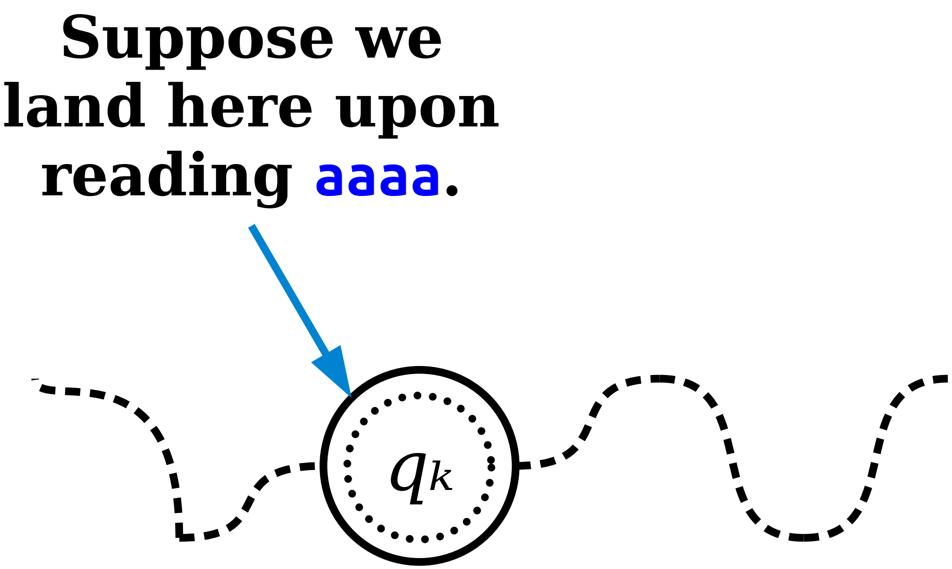
The Model

- The computing device has internal workings that can be in one of finitely many possible configurations.
  - Each ***state*** in a DFA corresponds to some possible configuration of the internal workings.
- After each button press, the computing device does some amount of processing, then gets to a configuration where it's ready to receive more input.
  - Each ***transition*** abstracts away how the computation is done and just indicates what the ultimate configuration looks like.
- After the user presses the “done” button, the computer outputs either YES or NO.
  - The ***accepting*** and ***rejecting*** states of the machine model what happens when that button is pressed.

New Stuff!

First, a Preliminary (and Crucial) Exercise

Suppose we have a DFA for ℒ(**a*∪b***).



Note: We have not indicated whether q_k accepts or rejects.

Not knowing what the rest of the DFA looks like, which of the following can we say are true?

(a) **aaa** **<u>must</u>** also land us in this state

(b) **aaa** **<u>might</u>** also land us in this state

(c) **aaa** **<u>could not</u>** land us in this state

(d) **bbb** **<u>must</u>** also land us in this state

(e) **bbb** **<u>might</u>** also land us in this state

(f) **bbb** **<u>could not</u>** land us in this state

Answer at
https://cs103.stanford.edu/pollev

Nonregular Languages

A Powerful Intuition

- ***Regular languages correspond to problems that can be solved with finite memory.***
  - At each point in time, we only need to store one of finitely many pieces of information.
- Nonregular languages, in a sense, correspond to problems that cannot be solved with finite memory.
- Since every computer ever built has finite memory, in a sense, nonregular languages correspond to problems that cannot be solved by physical computers!

Finding Nonregular Languages

Finding Nonregular Languages

- To prove that a language is regular, we can just find a DFA, NFA, or regex for it.
- To prove that a language is not regular, we need to prove that there are no possible DFAs, NFAs, or regexes for it.
  - ***Claim:*** We can actually just prove that there's no DFA for it. Why is this?
- ***This sort of argument will be challenging.*** Our arguments will be somewhat technical in nature, since we need to rigorously establish that no amount of creativity could produce a DFA for a given language.
- Let's see an example of how to do this.

A Simple Language

- Let $\Sigma = \{\texttt{a}, \texttt{b}\}$ and consider the following language:

$$E = \{\texttt{a}^n\texttt{b}^n \mid n \in \mathbb{N}\ \}$$

- E is the language of all strings of n **a**'s followed by n **b**'s:

$$\{\ \varepsilon, \texttt{ab}, \texttt{aabb}, \texttt{aaabbb}, \texttt{aaaabbbb}, \dots\ \}$$

A Simple Language

$$E = \{ \mathbf{a}^n \mathbf{b}^n \mid n \in \mathbb{N} \}$$

None of these regular expressions are regexes for the language *E*. Explain why not.

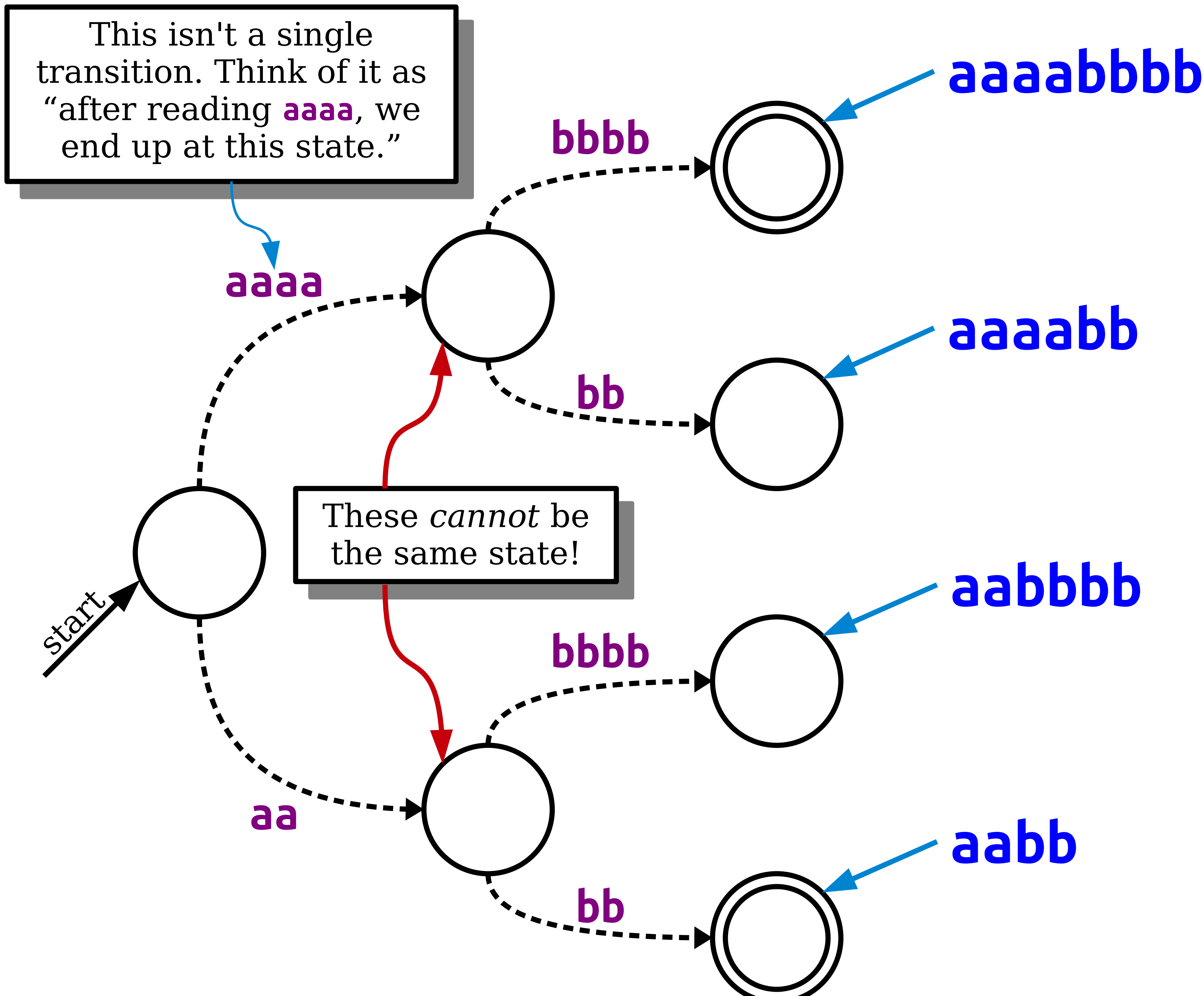
a*b*
(ab)*
$\varepsilon \cup ab \cup a^2b^2 \cup a^3b^3$

Answer at ***https://cs103.stanford.edu/pollev***

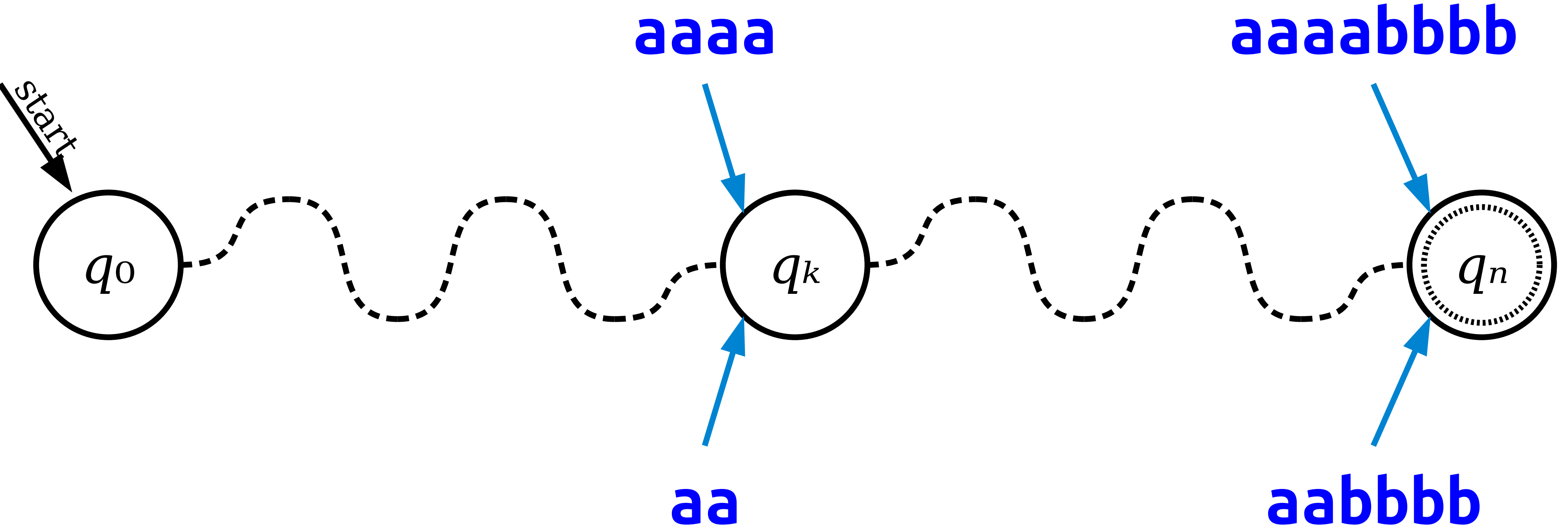
We seem to be running into some trouble.
Why is that?

Let's imagine what a DFA for the language $\{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}\}$ would have to look like.

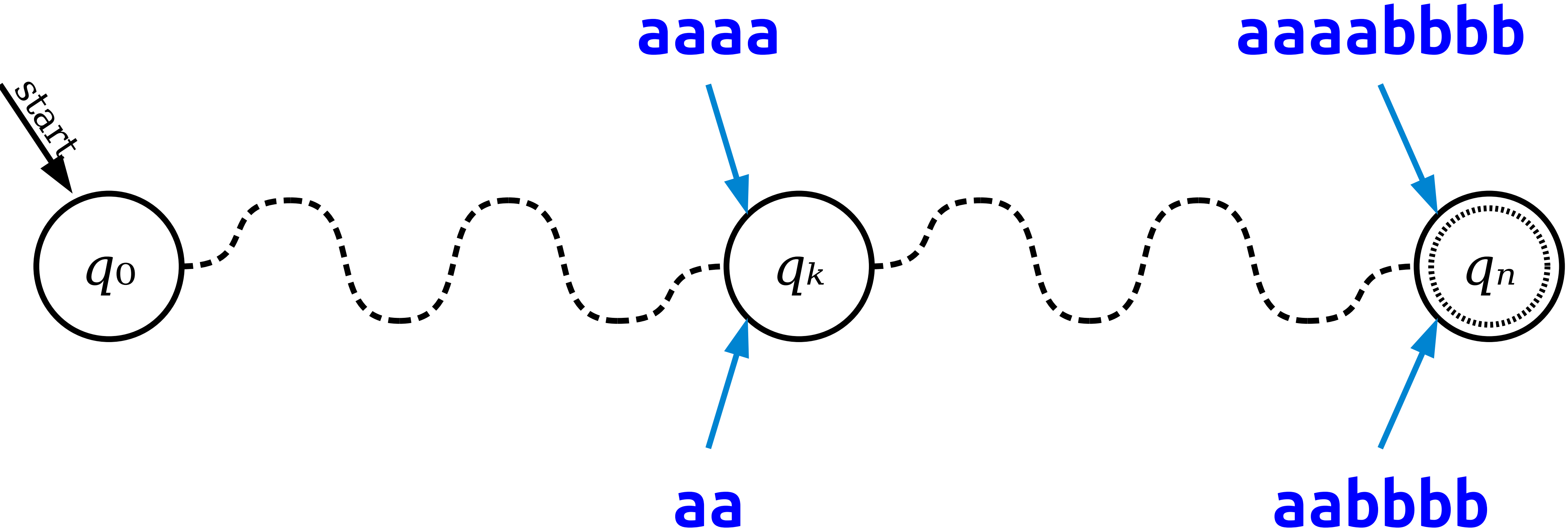
Can we say anything about it?


This isn't a single transition. Think of it as "after reading aaaa, we end up at this state."
aaaa
bbbb
bb
aaaabbbb
aaaabb
These *cannot* be the same state!
start
aa
bbbb
bb
aabbbb
aabb

A Different Perspective



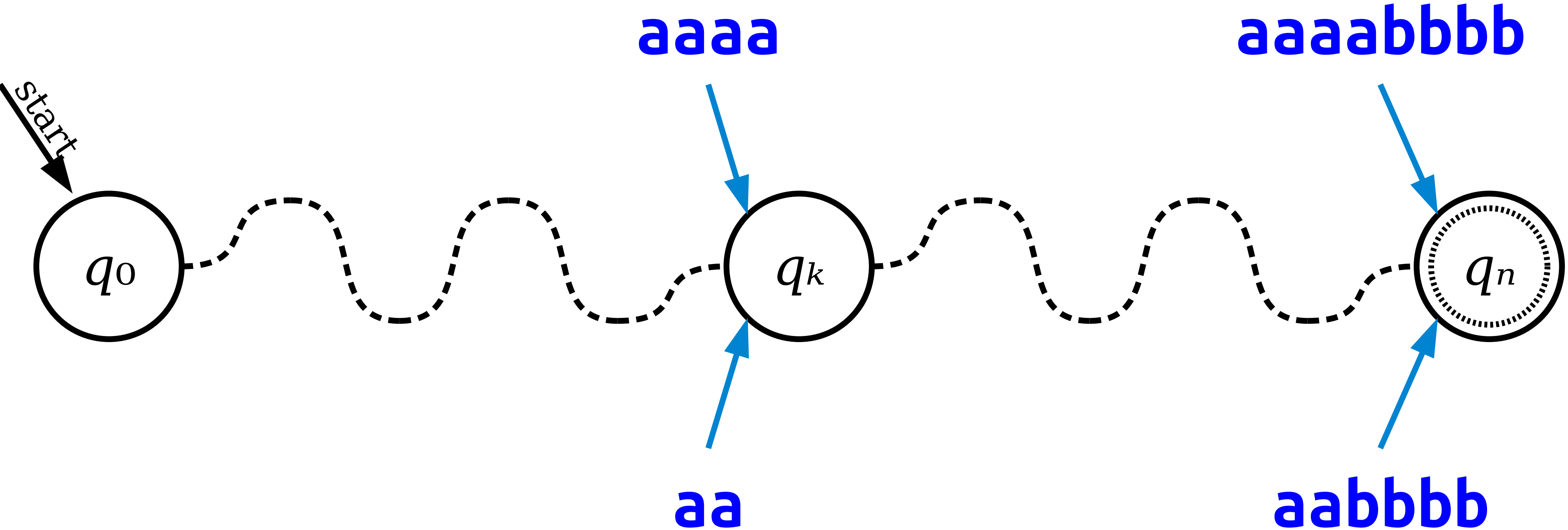
A Different Perspective



What happens if q_n is…

…an accepting state?
…a rejecting state?

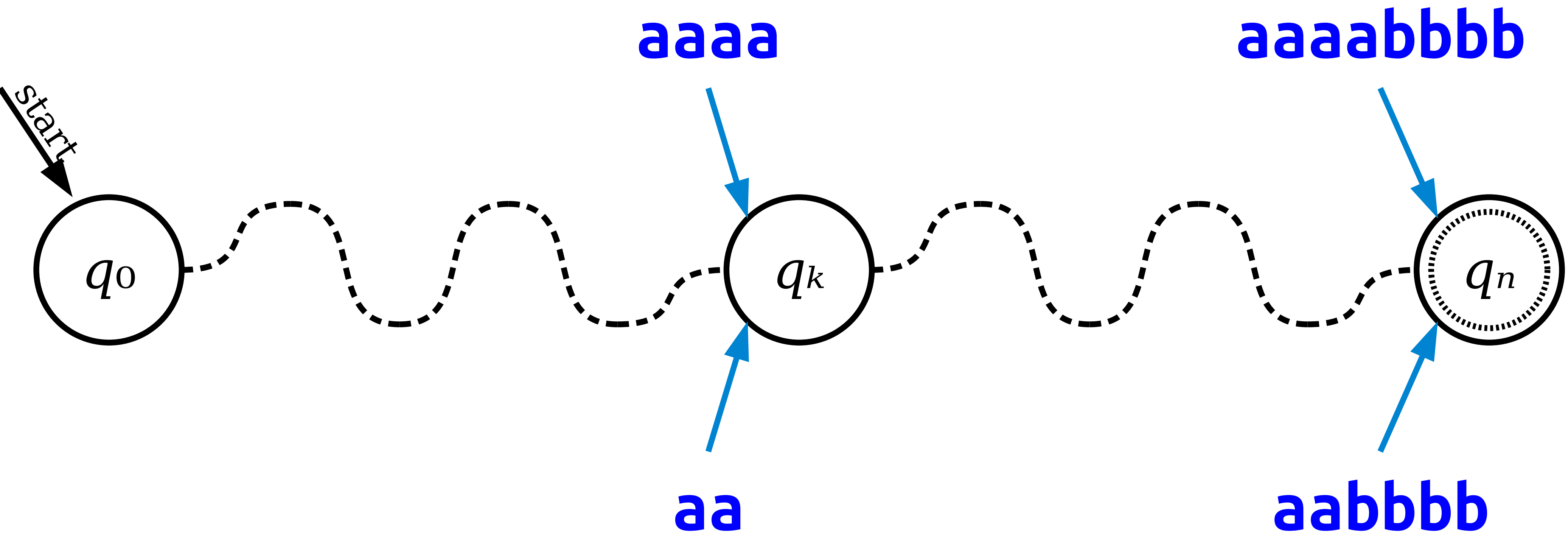
A Different Perspective



What happens if q_n is…

…an accepting state? We accept **aabbbb** $\notin E$!
…a rejecting state?

A Different Perspective

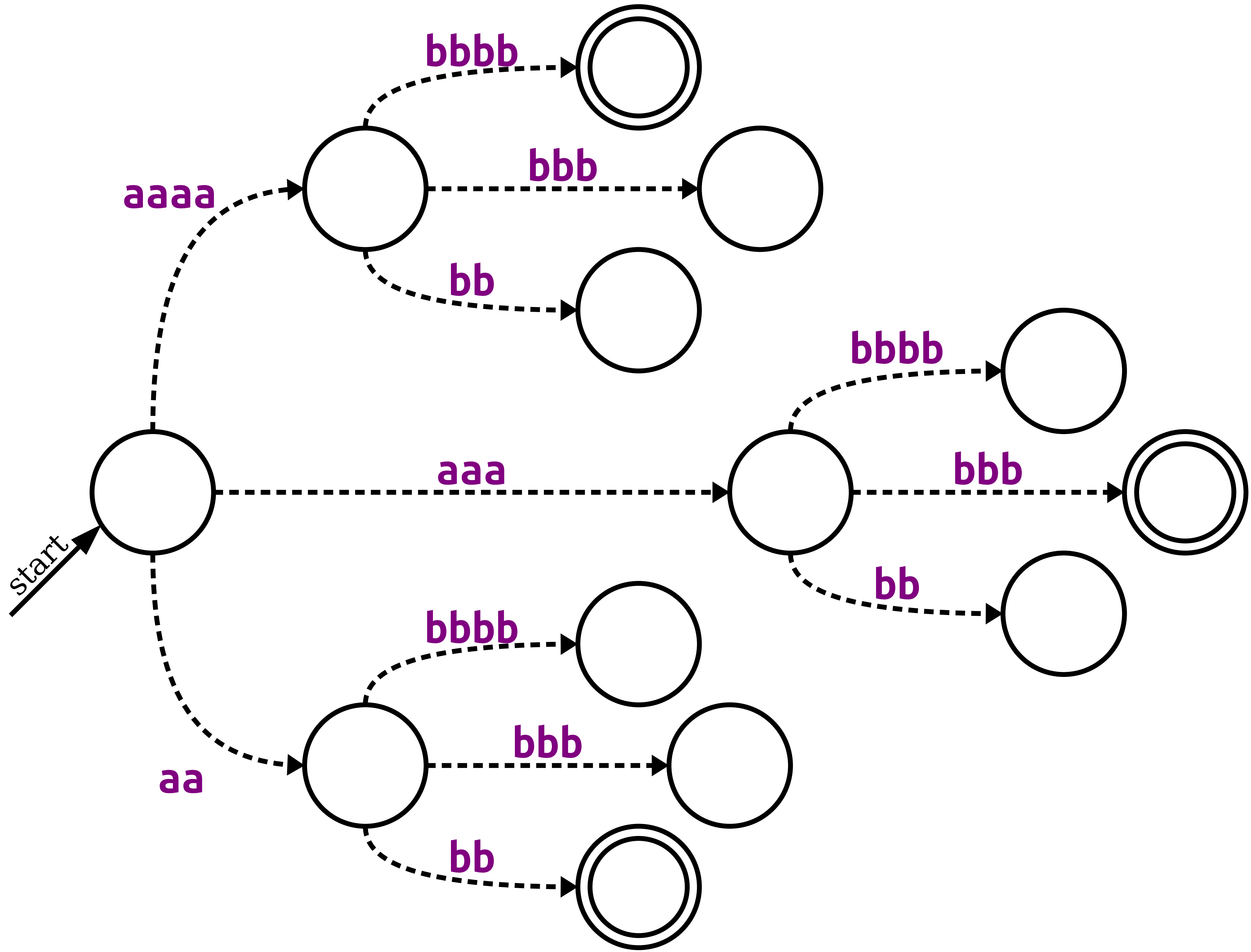


What happens if q_n is…

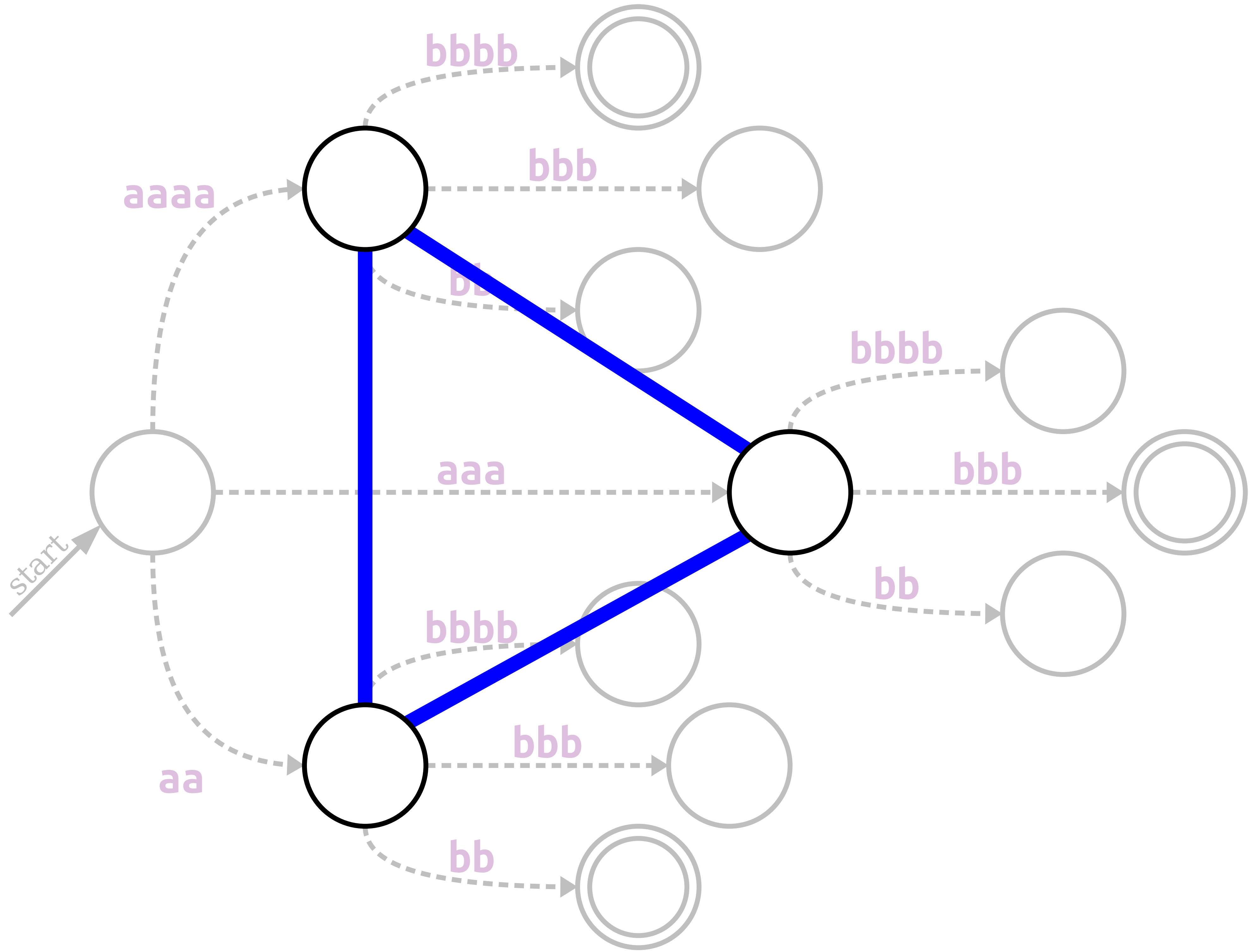
…an accepting state? We accept **aabbbb** $\notin E$!
…a rejecting state? We reject **aaaabbbb** $\in E$!

What's Going On?

- ***Lemma:*** If D is a DFA for $E = \{\mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}\}$ and we run D on both $\mathtt{a}^2$ and $\mathtt{a}^4$, then those strings do not end in the same state.
- ***Two Proof Ideas:***
  - *Direct:* The states you reach for $\mathtt{a}^4$ and $\mathtt{a}^2$ have to behave differently when reading $\mathtt{b}^4$ – in one case it should lead to an accepting state, in the other it should lead to a rejecting state. Therefore, they must be different states.
  - *Contradiction:* Suppose you do end up in the same state. Then $\mathtt{a}^4\mathtt{b}^4$ and $\mathtt{a}^2\mathtt{b}^4$ end up in the same state, so we either reject $\mathtt{a}^4\mathtt{b}^4$ (oops) or accept $\mathtt{a}^2\mathtt{b}^4$ (oops).
- ***Powerful intuition:*** Any DFA for E must keep $\mathtt{a}^2$ and $\mathtt{a}^4$ separated. It needs to remember something fundamentally different after reading those strings.



start
aaaa
aaa
aa
bbbb
bbb
bb
bbbb
bbb
bb
bbbb
bbb
bb



bbbb
bbb
aaaa
bb
bbbb
aaa
bbb
start
bb
bbbb
bbb
aa
bb

A More General Result

- ***Lemma:*** Let D be a DFA for $E = \{\mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}\}$. For any distinct strings $\mathbf{a}^m$ and $\mathbf{a}^n$, if we run D on both $\mathbf{a}^m$ and $\mathbf{a}^n$, then those strings do not end in the same state.
- ***Two Proof Ideas:***
  - *Direct:* The states you reach for $\mathbf{a}^m$ and $\mathbf{a}^n$ have to behave differently when reading $\mathbf{b}^m$ – in one case it should lead to an accepting state, in the other it should lead to a rejecting state. Therefore, they must be different states.
  - *Contradiction:* Suppose you do end up in the same state. Then $\mathbf{a}^m\mathbf{b}^m$ and $\mathbf{a}^m\mathbf{b}^n$ end up in the same state, so we either reject $\mathbf{a}^m\mathbf{b}^m$ (oops) or accept $\mathbf{a}^m\mathbf{b}^n$ (oops).
- ***Powerful intuition:*** Any DFA for E must keep $\mathbf{a}^m$ and $\mathbf{a}^n$ separated. It needs to remember something fundamentally different after reading those strings.

A Bad Combination

- Suppose there is a DFA D for the language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$.
- We know the following:
  - Any two strings of the form $\mathtt{a}^m$ and $\mathtt{a}^n$, where $m \neq n$, cannot end in the same state when run through D.
  - There are infinitely many strings of the form $\mathtt{a}^m$.
  - However, there are only *finitely many* states they can end up in, since D is a deterministic ***finite*** automaton!
- What happens if we put these pieces together?

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof:

Theorem: The language $E = \{ \texttt{a}^n\texttt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Suppose for the sake of contradiction that E is regular.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Suppose for the sake of contradiction that E is regular. Let D be a DFA for E, and let k be the number of states in D.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Suppose for the sake of contradiction that E is regular. Let D be a DFA for E, and let k be the number of states in D. Consider the strings $\mathtt{a}^0$, $\mathtt{a}^1$, $\mathtt{a}^2$, …, $\mathtt{a}^k$.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Suppose for the sake of contradiction that E is regular. Let D be a DFA for E, and let k be the number of states in D. Consider the strings $\mathtt{a}^0$, $\mathtt{a}^1$, $\mathtt{a}^2$, …, $\mathtt{a}^k$. This is a collection of $k+1$ strings and there are only k states in D.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Suppose for the sake of contradiction that E is regular. Let D be a DFA for E, and let k be the number of states in D. Consider the strings $\mathtt{a}^0$, $\mathtt{a}^1$, $\mathtt{a}^2$, …, $\mathtt{a}^k$. This is a collection of $k+1$ strings and there are only k states in D. There must be two distinct strings $\mathtt{a}^m$ and $\mathtt{a}^n$ that end in the same state when run through D. (***How do we know??***)

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Suppose for the sake of contradiction that E is regular. Let D be a DFA for E, and let k be the number of states in D. Consider the strings $\mathtt{a}^0$, $\mathtt{a}^1$, $\mathtt{a}^2$, …, $\mathtt{a}^k$. This is a collection of $k+1$ strings and there are only k states in D. Therefore, by the pigeonhole principle, there must be two distinct strings $\mathtt{a}^m$ and $\mathtt{a}^n$ that end in the same state when run through D.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Suppose for the sake of contradiction that E is regular. Let D be a DFA for E, and let k be the number of states in D. Consider the strings $\mathtt{a}^0$, $\mathtt{a}^1$, $\mathtt{a}^2$, …, $\mathtt{a}^k$. This is a collection of $k+1$ strings and there are only k states in D. Therefore, by the pigeonhole principle, there must be two distinct strings $\mathtt{a}^m$ and $\mathtt{a}^n$ that end in the same state when run through D. But this is impossible, since we know that $\mathtt{a}^m$ and $\mathtt{a}^n$ cannot end in the same state when run through D.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Suppose for the sake of contradiction that E is regular. Let D be a DFA for E, and let k be the number of states in D. Consider the strings $\mathtt{a}^0$, $\mathtt{a}^1$, $\mathtt{a}^2$, …, $\mathtt{a}^k$. This is a collection of $k+1$ strings and there are only k states in D. Therefore, by the pigeonhole principle, there must be two distinct strings $\mathtt{a}^m$ and $\mathtt{a}^n$ that end in the same state when run through D. But this is impossible, since we know that $\mathtt{a}^m$ and $\mathtt{a}^n$ cannot end in the same state when run through D.

We have reached a contradiction, so our assumption must have been wrong.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Suppose for the sake of contradiction that E is regular. Let D be a DFA for E, and let k be the number of states in D. Consider the strings $\mathtt{a}^0$, $\mathtt{a}^1$, $\mathtt{a}^2$, …, $\mathtt{a}^k$. This is a collection of $k+1$ strings and there are only k states in D. Therefore, by the pigeonhole principle, there must be two distinct strings $\mathtt{a}^m$ and $\mathtt{a}^n$ that end in the same state when run through D. But this is impossible, since we know that $\mathtt{a}^m$ and $\mathtt{a}^n$ cannot end in the same state when run through D.

We have reached a contradiction, so our assumption must have been wrong. Therefore, E is not regular.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Suppose for the sake of contradiction that E is regular. Let D be a DFA for E, and let k be the number of states in D. Consider the strings $\mathtt{a}^0$, $\mathtt{a}^1$, $\mathtt{a}^2$, …, $\mathtt{a}^k$. This is a collection of $k+1$ strings and there are only k states in D. Therefore, by the pigeonhole principle, there must be two distinct strings $\mathtt{a}^m$ and $\mathtt{a}^n$ that end in the same state when run through D. But this is impossible, since we know that $\mathtt{a}^m$ and $\mathtt{a}^n$ cannot end in the same state when run through D.

We have reached a contradiction, so our assumption must have been wrong. Therefore, E is not regular. ■

We're going to see a simpler proof of this result later on once we've built more machinery. If (hypothetically speaking) you want to prove something like this in the future, we'd recommend not using this proof as a template.

What Just Happened?

- ***We've just hit the limit of finite-memory computation.***
- To build a DFA for $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$, we need to have different memory configurations (states) for all possible strings of the form $\mathtt{a}^n$.
- There's no way to do this with finitely many possible states!

What Just Happened?

- ***We've just hit the limit of finite-memory computation.*** 
- To build a DFA for $E = \{ \texttt{a}^n\texttt{b}^n \mid n \in \mathbb{N} \}$, we need to have different memory configurations (states) for all possible strings of the form $\texttt{a}^n$.
- There's no way to do this with finitely many possible states!

What Just Happened?

- ***We've just hit the limit of finite-memory computation.***
- To build a DFA for $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$, we need to have different memory configurations (states) for all possible strings of the form $\mathtt{a}^n$.
- There's no way to do this with finitely many possible states!

What Just Happened?

- ***We've just hit the limit of finite-memory computation.***
- To build a DFA for $E = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N} \}$, we need to have different memory configurations (states) for all possible strings of the form $\mathbf{a}^n$.
- There's no way to do this with finitely many possible states!

What Just Happened?

- ***We've just hit the limit of finite-memory computation.***
- To build a DFA for $E = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N} \}$, we need to have different memory configurations (states) for all possible strings of the form $\mathbf{a}^n$.
- There's no way to do this with finitely many possible states!

Time-Out for Announcements!

Second Midterm Logistics

- Our second midterm exam is next ***Tuesday, February 25th*** from ***7-9 PM***. Locations vary (mostly CEMEX).
- Check seating assignment page! Big shake-up!
- Topic coverage is primarily lectures 06 – 13 (functions through induction) and PS3 – PS5. Finite automata and onward won't be tested here.
  - Because the material is cumulative, topics from PS1 – PS2 and Lectures 00 – 05 are also fair game.
- The exam is closed-book and closed-computer. You can bring one double-sided 8.5" × 11" sheet of notes with you.
- Students with accommodations and alternate arrangements: check seating assignment page. Contact us if anything is amiss.

Review Session

- Anisha and Zach will be holding a review session ***Sunday, February 23rd*** from ***4–6 PM*** in ***CoDa E160***.
  - As with last time, this is not recorded.
  - As with last time, come prepared with questions you want to ask.
- We also have a ton of practice exams up on the course website.
- ***Best of luck – you can do this!***

Don't be afraid
to take a deep
breath & smile!
Cuz you got this :)

ty

Back to CS103!

Generalizing the Proof

What We Did

- Our proof that $E = \{\mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}\}$ is not regular relied on several observations:
  - No two strings of the form $\mathtt{a}^m$ and $\mathtt{a}^n$ can end in the same state in any DFA for E, because there's a string we can append that puts one in the language and keeps the other out.
  - There are infinitely many strings of this form, so we can run as many of them as we'd like through a DFA for E.
  - DFAs only have finitely many states, so by the pigeonhole principle any DFA for E necessarily has to put two of these strings in the same place.
  - So there can't be a DFA for E.
- ***Question:*** Can we generalize this idea?

What We Did

Our proof that $E = \{\mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}\}$ is not regular relied on several observations:

- No two strings of the form $\mathbf{a}^m$ and $\mathbf{a}^n$ can end in the same state in any DFA for E, because there's a string we can append that puts one in the language and keeps the other out.

  There are infinitely many strings of this form, so we can run as many of them as we'd like through a DFA for E.

  DFAs only have finitely many states, so by the pigeonhole principle any DFA for E necessarily has to put two of these strings in the same place.

  So there can't be a DFA for E.

Question: Can we generalize this idea?

Distinguishability

- Let L be an arbitrary language over Σ.
- Two strings $x \in \Sigma^*$ and $y \in \Sigma^*$ are called ***distinguishable relative to L*** if there is a string $w \in \Sigma^*$ such that exactly one of xw and yw is in L.
- We denote this by writing $\boldsymbol{x \not\equiv_L y}$.
- Formally, we say that $x \not\equiv_L y$ if the following is true:

$$\exists w \in \Sigma^*. \ (xw \in L \leftrightarrow yw \notin L)$$

This is how we express exclusive "OR" in propositional logic.

Distinguishability

- Consider the language

$$E = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N} \}.$$

- There's a collection of strings to the right.
- Which pairs of these strings are distinguishable relative to E? What would you append to distinguish them?
- Two strings x and y are distinguishable relative to E if there's a string w where exactly one of xw and yw belongs to E.

aab

abb

aba

aaa

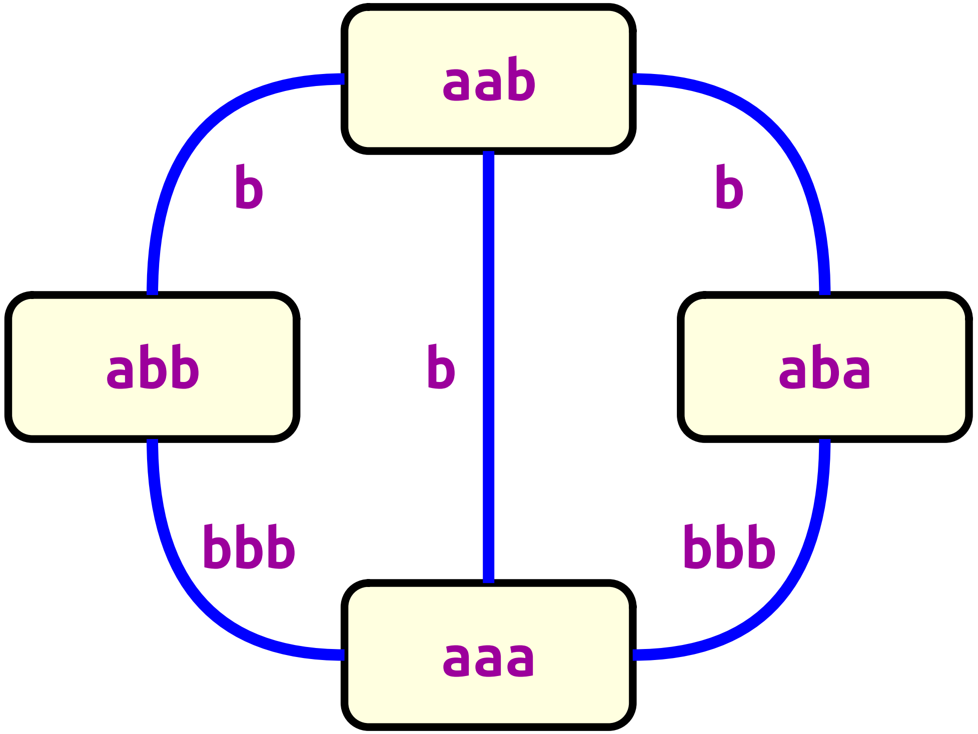
Answer at
https://cs103.stanford.edu/pollev

Distinguishability

- Consider the language

$$E = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N} \}.$$

- There's a collection of strings to the right.
- Which pairs of these strings are distinguishable relative to E? What would you append to distinguish them?
- Two strings x and y are distinguishable relative to E if there's a string w where exactly one of xw and yw belongs to E.



Distinguishability

- A ***palindrome*** is a string that is the same when the characters are read left-to-right and right-to-left.
- Consider the language

$$L = \{\ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w \text{ is a palindrome }\}$$

- Which pairs of the strings below are distinguishable relative to L? What would you append to distinguish them?

aab

abb **aba**

aaa

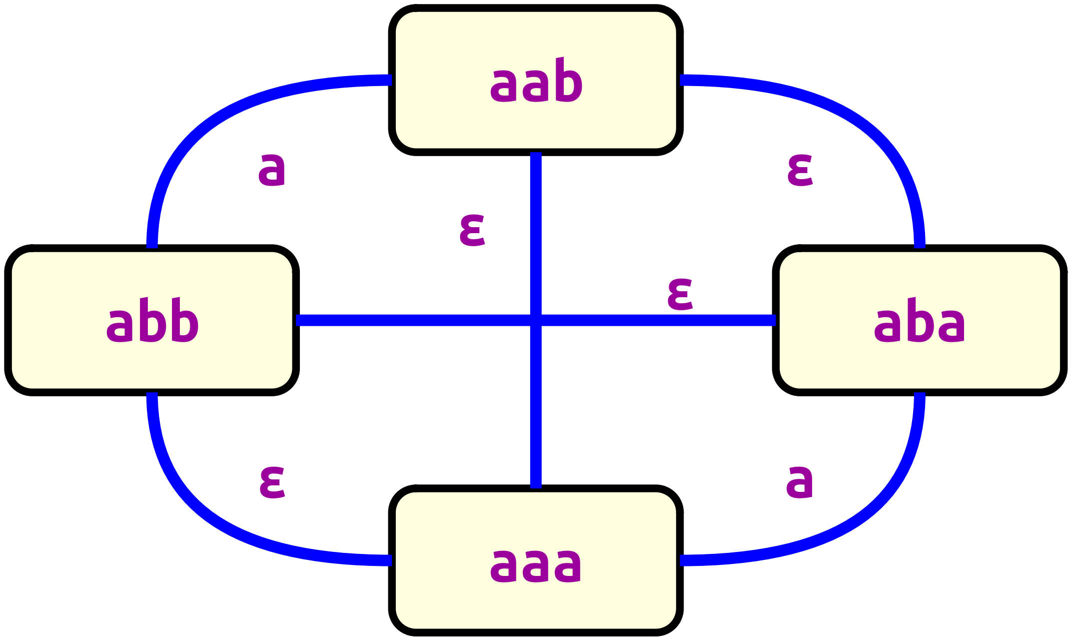
Answer at
https://cs103.stanford.edu/pollev

Distinguishability

- A ***palindrome*** is a string that is the same when the characters are read left-to-right and right-to-left.
- Consider the language

$$L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w \text{ is a palindrome } \}$$

- Which pairs of the strings below are distinguishable relative to L? What would you append to distinguish them?

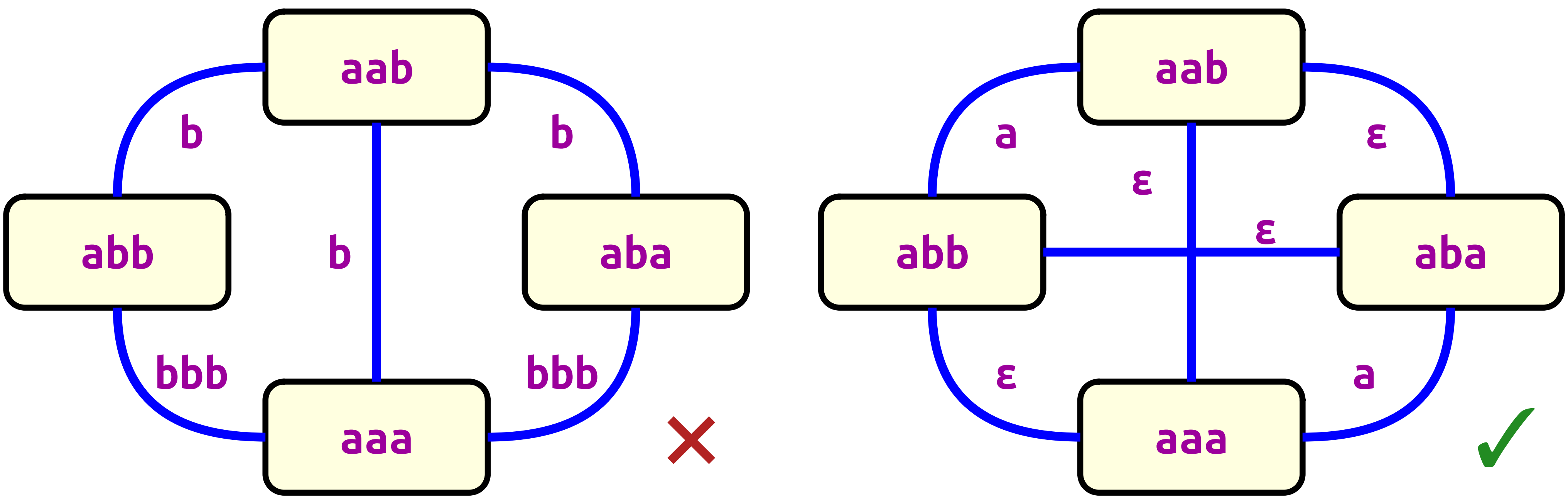


Distinguishing Sets

- Let $L \subseteq \Sigma^*$ be a language. A ***distinguishing set for L*** is set $S \subseteq \Sigma^*$ where the following is true:

$$\forall x \in S.\ \forall y \in S.\ (x \neq y \to x \not\equiv_L y).$$

- In other words, it's a set of strings S where all pairs of distinct strings in S are distinguishable relative to L.



Distinguishing Sets

- Let $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$.
- Which of the following are distinguishing sets for E?

$$\{ \varepsilon, \mathtt{a}, \mathtt{ab} \}$$

$$\mathtt{a}^*$$

$$\{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$$

Answer at
https://cs103.stanford.edu/pollev

Distinguishing Sets

- Let $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w \text{ is a palindrome} \}$.
- Which of the following are distinguishing sets for L?

$$\{ \varepsilon, \mathtt{a}, \mathtt{ab} \}$$

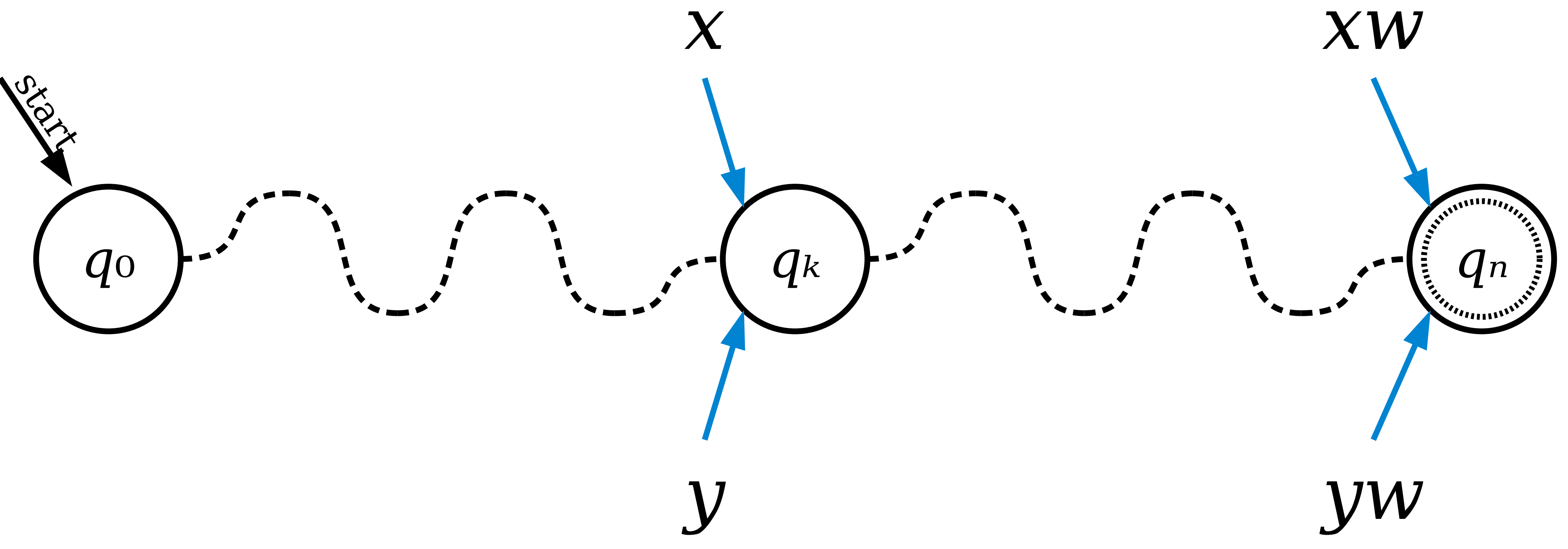
$$\mathtt{a}^*$$

$$\{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$$

Answer at
https://cs103.stanford.edu/pollev

Distinguishability

- ***Theorem:*** Let L be an arbitrary language over Σ. Let $x \in \Sigma^*$ and $y \in \Sigma^*$ be strings where $x \not\equiv_L y$. Then if D is ***any*** DFA for L, then D must end in different states when run on inputs x and y.
- ***Proof sketch:***



Theorem (Myhill-Nerode): Let L be a language. If L has an infinite distinguishing set (a distinguishing set containing infinitely many strings), then L is not regular.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof:

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

We know L is regular, so there is a DFA D for L.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

We know L is regular, so there is a DFA D for L. Let k be the number of states in D.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

We know L is regular, so there is a DFA D for L. Let k be the number of states in D. Since there are infinitely many strings in S, we can pick $k+1$ distinct strings w_1, w_2, …, and w_{k+1} from S.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

We know L is regular, so there is a DFA D for L. Let k be the number of states in D. Since there are infinitely many strings in S, we can pick $k+1$ distinct strings w_1, w_2, …, and w_{k+1} from S.

Consider what happens when we run D on all those strings.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

We know L is regular, so there is a DFA D for L. Let k be the number of states in D. Since there are infinitely many strings in S, we can pick $k+1$ distinct strings w_1, w_2, …, and w_{k+1} from S.

Consider what happens when we run D on all those strings. There are only k states in D and we have $k+1$ strings, so by the pigeonhole principle there are strings $w_i \neq w_j$ in S that end in the same state when run through D.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

We know L is regular, so there is a DFA D for L. Let k be the number of states in D. Since there are infinitely many strings in S, we can pick $k+1$ distinct strings w_1, w_2, …, and w_{k+1} from S.

Consider what happens when we run D on all those strings. There are only k states in D and we have $k+1$ strings, so by the pigeonhole principle there are strings $w_i \neq w_j$ in S that end in the same state when run through D.

Because $w_i \neq w_j$ and S is a distinguishing set for L, we know that $w_i \not\equiv_L w_j$.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

We know L is regular, so there is a DFA D for L. Let k be the number of states in D. Since there are infinitely many strings in S, we can pick $k+1$ distinct strings w_1, w_2, …, and w_{k+1} from S.

Consider what happens when we run D on all those strings. There are only k states in D and we have $k+1$ strings, so by the pigeonhole principle there are strings $w_i \neq w_j$ in S that end in the same state when run through D.

Because $w_i \neq w_j$ and S is a distinguishing set for L, we know that $w_i \not\equiv_L w_j$. As we saw earlier, when we run w_i and w_j through D, they therefore end up in different states.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

We know L is regular, so there is a DFA D for L. Let k be the number of states in D. Since there are infinitely many strings in S, we can pick $k+1$ distinct strings w_1, w_2, …, and w_{k+1} from S.

Consider what happens when we run D on all those strings. There are only k states in D and we have $k+1$ strings, so by the pigeonhole principle there are strings $w_i \neq w_j$ in S that end in the same state when run through D.

Because $w_i \neq w_j$ and S is a distinguishing set for L, we know that $w_i \not\equiv_L w_j$. As we saw earlier, when we run w_i and w_j through D, they therefore end up in different states. But this is impossible: w_i and w_j end in the same state when run through D.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

We know L is regular, so there is a DFA D for L. Let k be the number of states in D. Since there are infinitely many strings in S, we can pick $k+1$ distinct strings w_1, w_2, …, and w_{k+1} from S.

Consider what happens when we run D on all those strings. There are only k states in D and we have $k+1$ strings, so by the pigeonhole principle there are strings $w_i \neq w_j$ in S that end in the same state when run through D.

Because $w_i \neq w_j$ and S is a distinguishing set for L, we know that $w_i \not\equiv_L w_j$. As we saw earlier, when we run w_i and w_j through D, they therefore end up in different states. But this is impossible: w_i and w_j end in the same state when run through D.

We have reached a contradiction, so our assumption must have been wrong.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

We know L is regular, so there is a DFA D for L. Let k be the number of states in D. Since there are infinitely many strings in S, we can pick $k+1$ distinct strings w_1, w_2, …, and w_{k+1} from S.

Consider what happens when we run D on all those strings. There are only k states in D and we have $k+1$ strings, so by the pigeonhole principle there are strings $w_i \neq w_j$ in S that end in the same state when run through D.

Because $w_i \neq w_j$ and S is a distinguishing set for L, we know that $w_i \not\equiv_L w_j$. As we saw earlier, when we run w_i and w_j through D, they therefore end up in different states. But this is impossible: w_i and w_j end in the same state when run through D.

We have reached a contradiction, so our assumption must have been wrong. Thus L is not a regular language.

Theorem: Let L be a language. If L has an infinite distinguishing set, then L is not regular.

Proof: Assume for the sake of contradiction that there is an infinite distinguishing set S for L but that L is regular.

We know L is regular, so there is a DFA D for L. Let k be the number of states in D. Since there are infinitely many strings in S, we can pick $k+1$ distinct strings w_1, w_2, …, and w_{k+1} from S.

Consider what happens when we run D on all those strings. There are only k states in D and we have $k+1$ strings, so by the pigeonhole principle there are strings $w_i \neq w_j$ in S that end in the same state when run through D.

Because $w_i \neq w_j$ and S is a distinguishing set for L, we know that $w_i \not\equiv_L w_j$. As we saw earlier, when we run w_i and w_j through D, they therefore end up in different states. But this is impossible: w_i and w_j end in the same state when run through D.

We have reached a contradiction, so our assumption must have been wrong. Thus L is not a regular language. ■

Using the Myhill-Nerode Theorem

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof:

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for E.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for E.

To see that S is infinite, note that S contains one string for each natural number.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for E.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for E, consider any two strings $\mathtt{a}^m, \mathtt{a}^n \in S$ where $m \neq n$.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for E.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for E, consider any two strings $\mathtt{a}^m, \mathtt{a}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{b}^m \in E$ and that $\mathtt{a}^n\mathtt{b}^m \notin E$.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for E.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for E, consider any two strings $\mathtt{a}^m, \mathtt{a}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{b}^m \in E$ and that $\mathtt{a}^n\mathtt{b}^m \notin E$. Therefore, we see that $\mathtt{a}^m \not\equiv_E \mathtt{a}^n$, as required.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for E.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for E, consider any two strings $\mathtt{a}^m, \mathtt{a}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{b}^m \in E$ and that $\mathtt{a}^n\mathtt{b}^m \notin E$. Therefore, we see that $\mathtt{a}^m \not\equiv_E \mathtt{a}^n$, as required.

Since S is infinite and is a distinguishing set for E, by the Myhill-Nerode theorem we see that E is not regular.

Theorem: The language $E = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for E.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for E, consider any two strings $\mathtt{a}^m, \mathtt{a}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{b}^m \in E$ and that $\mathtt{a}^n\mathtt{b}^m \notin E$. Therefore, we see that $\mathtt{a}^m \not\equiv_E \mathtt{a}^n$, as required.

Since S is infinite and is a distinguishing set for E, by the Myhill-Nerode theorem we see that E is not regular. ■

Theorem: The language $L = \{ w \in \{\texttt{a}, \texttt{b}\}^* \mid w \text{ is a palindrome} \}$ is not regular.

Theorem: The language $L = \{ w \in \{\texttt{a}, \texttt{b}\}^* \mid w \text{ is a palindrome} \}$ is not regular.

Proof:

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w \text{ is a palindrome} \}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for L, consider any two strings $\mathtt{a}^m\mathtt{b}^m$, $\mathtt{a}^n\mathtt{b}^n \in S$ where $m \neq n$.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for L, consider any two strings $\mathtt{a}^m\mathtt{b}^m, \mathtt{a}^n\mathtt{b}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{b}^m\mathtt{b}^m\mathtt{a}^m \in L$ and that $\mathtt{a}^n\mathtt{b}^n\mathtt{b}^m\mathtt{a}^m \notin L$.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for L, consider any two strings $\mathtt{a}^m\mathtt{b}^m, \mathtt{a}^n\mathtt{b}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{b}^m\mathtt{b}^m\mathtt{a}^m \in L$ and that $\mathtt{a}^n\mathtt{b}^n\mathtt{b}^m\mathtt{a}^m \notin L$. Therefore, we see that $\mathtt{a}^m\mathtt{b}^m \not\equiv_L \mathtt{a}^n\mathtt{b}^n$, as required.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for L, consider any two strings $\mathtt{a}^m\mathtt{b}^m, \mathtt{a}^n\mathtt{b}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{b}^m\mathtt{b}^m\mathtt{a}^m \in L$ and that $\mathtt{a}^n\mathtt{b}^n\mathtt{b}^m\mathtt{a}^m \notin L$. Therefore, we see that $\mathtt{a}^m\mathtt{b}^m \not\equiv_L \mathtt{a}^n\mathtt{b}^n$, as required.

Since S is infinite and is a distinguishing set for L, by the Myhill-Nerode theorem we see that L is not regular.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for L, consider any two strings $\mathtt{a}^m\mathtt{b}^m, \mathtt{a}^n\mathtt{b}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{b}^m\mathtt{b}^m\mathtt{a}^m \in L$ and that $\mathtt{a}^n\mathtt{b}^n\mathtt{b}^m\mathtt{a}^m \notin L$. Therefore, we see that $\mathtt{a}^m\mathtt{b}^m \not\equiv_L \mathtt{a}^n\mathtt{b}^n$, as required.

Since S is infinite and is a distinguishing set for L, by the Myhill-Nerode theorem we see that L is not regular. ■

Theorem: The language $L = \{ w \in \{\texttt{a}, \texttt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof:

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w \text{ is a palindrome} \}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

Theorem: The language $L = \{ w \in \{\texttt{a}, \texttt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \texttt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for L, consider any two strings $\texttt{a}^m, \texttt{a}^n \in S$ where $m \neq n$.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for L, consider any two strings $\mathtt{a}^m, \mathtt{a}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{ba}^m \in L$ and that $\mathtt{a}^n\mathtt{ba}^m \notin L$.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for L, consider any two strings $\mathtt{a}^m, \mathtt{a}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{ba}^m \in L$ and that $\mathtt{a}^n\mathtt{ba}^m \notin L$. Therefore, we see that $\mathtt{a}^m \not\equiv_L \mathtt{a}^n$, as required.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for L, consider any two strings $\mathtt{a}^m, \mathtt{a}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{ba}^m \in L$ and that $\mathtt{a}^n\mathtt{ba}^m \notin L$. Therefore, we see that $\mathtt{a}^m \not\equiv_L \mathtt{a}^n$, as required.

Since S is infinite and is a distinguishing set for L, by the Myhill-Nerode theorem we see that L is not regular.

Theorem: The language $L = \{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ is a palindrome $\}$ is not regular.

Proof: Let $S = \{ \mathtt{a}^n \mid n \in \mathbb{N} \}$. We will prove that S is infinite and that S is a distinguishing set for L.

To see that S is infinite, note that S contains one string for each natural number.

To see that S is a distinguishing set for L, consider any two strings $\mathtt{a}^m, \mathtt{a}^n \in S$ where $m \neq n$. Note that $\mathtt{a}^m\mathtt{ba}^m \in L$ and that $\mathtt{a}^n\mathtt{ba}^m \notin L$. Therefore, we see that $\mathtt{a}^m \not\equiv_L \mathtt{a}^n$, as required.

Since S is infinite and is a distinguishing set for L, by the Myhill-Nerode theorem we see that L is not regular. ■

Approaching Myhill-Nerode

- The challenge in using the Myhill-Nerode theorem is finding the right set of strings.
- ***General intuition:***
  - Start by thinking about what information a computer "must" remember in order to answer correctly.
  - Choose a group of strings that all require different information.
  - Prove that you have infinitely many strings an that the group of strings is a distinguishing set.

Tying Everything Together

- One of the intuitions we hope you develop for DFAs is to have each state in a DFA represent some key piece of information the automaton has to remember.
- If you only need to remember one of finitely many pieces of information, that gives you a DFA.
  - This can be made rigorous! Take CS154 for details.
- If you need to remember one of infinitely many pieces of information, you can use the Myhill-Nerode theorem to prove that the language has no DFA.

Where We Stand

Where We Stand

- We've ended up where we are now by trying to answer the question “what problems can you solve with a computer?”
- We defined a computer to be DFA, which means that the problems we can solve are precisely the regular languages.
- We've discovered several equivalent ways to think about regular languages (DFAs, NFAs, and regular expressions) and used that to reason about the regular languages.
- We now have a powerful intuition for where we ended up: DFAs are finite-memory computers, and regular languages correspond to problems solvable with finite memory.
- Putting all of this together, we have a much deeper sense for what finite memory computation looks like – *and what it doesn't look like!*

Where We're Going

- What does computation look like with unbounded memory?
- What problems can you solve with unbounded-memory computers?
- What does it even mean to “solve” such a problem?
- And how do we know the answers to any of these questions?

Your Action Items

- ***Read "Guide to the Myhill-Nerode Theorem"***
  - It's a useful refresher and deep-dive into all the topics we covered today.
  - And it has worked exercises to give you a sense of how the theorem works!

Next Time

- ***Context-Free Languages***
  - Context-Free Grammars
  - Generating Languages from Scratch