Nonregular Languages

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 $\mathscr{L}(R) = L$.

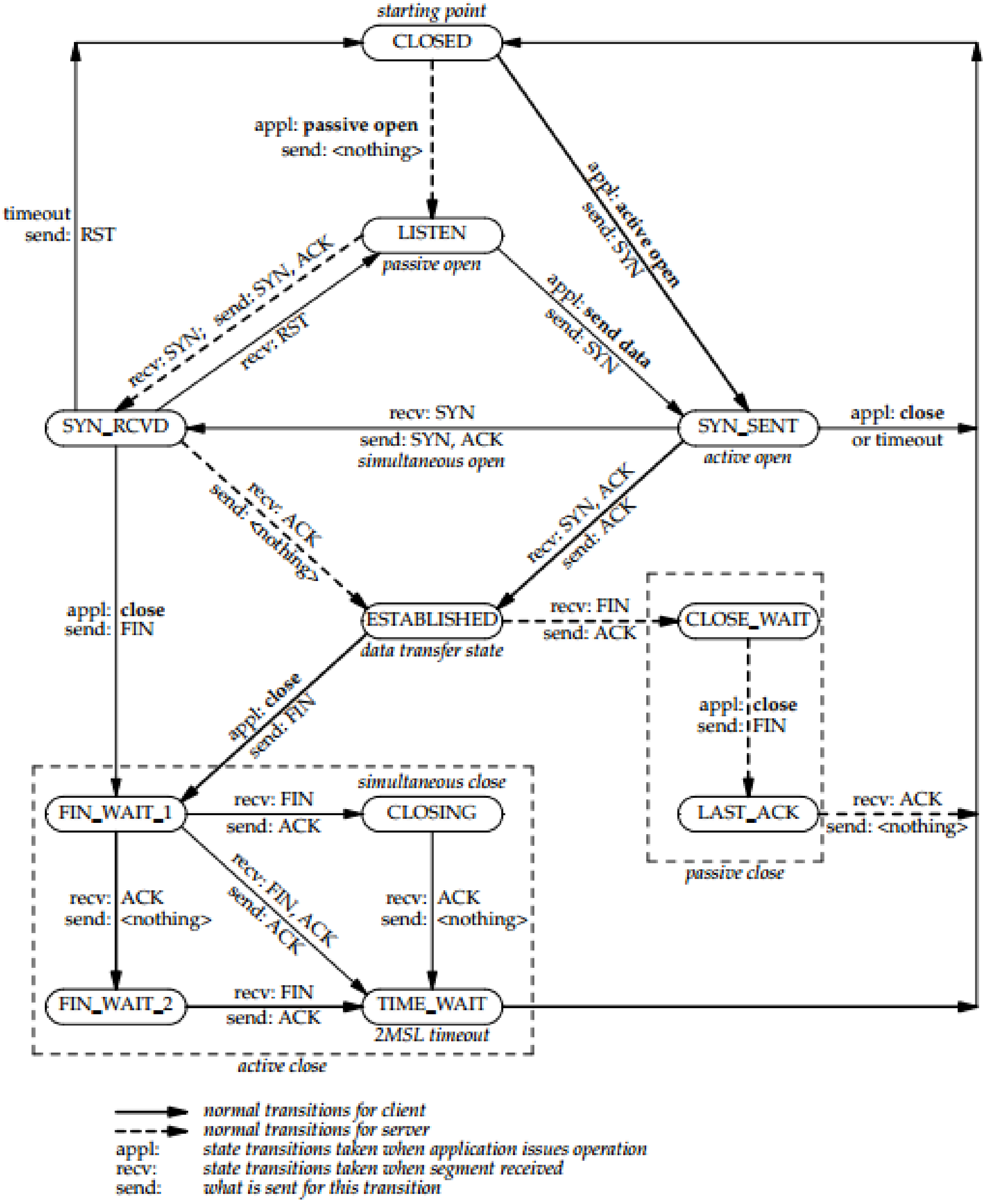
New Stuff!

Why does this matter?

Buttons as Finite-State Machines:

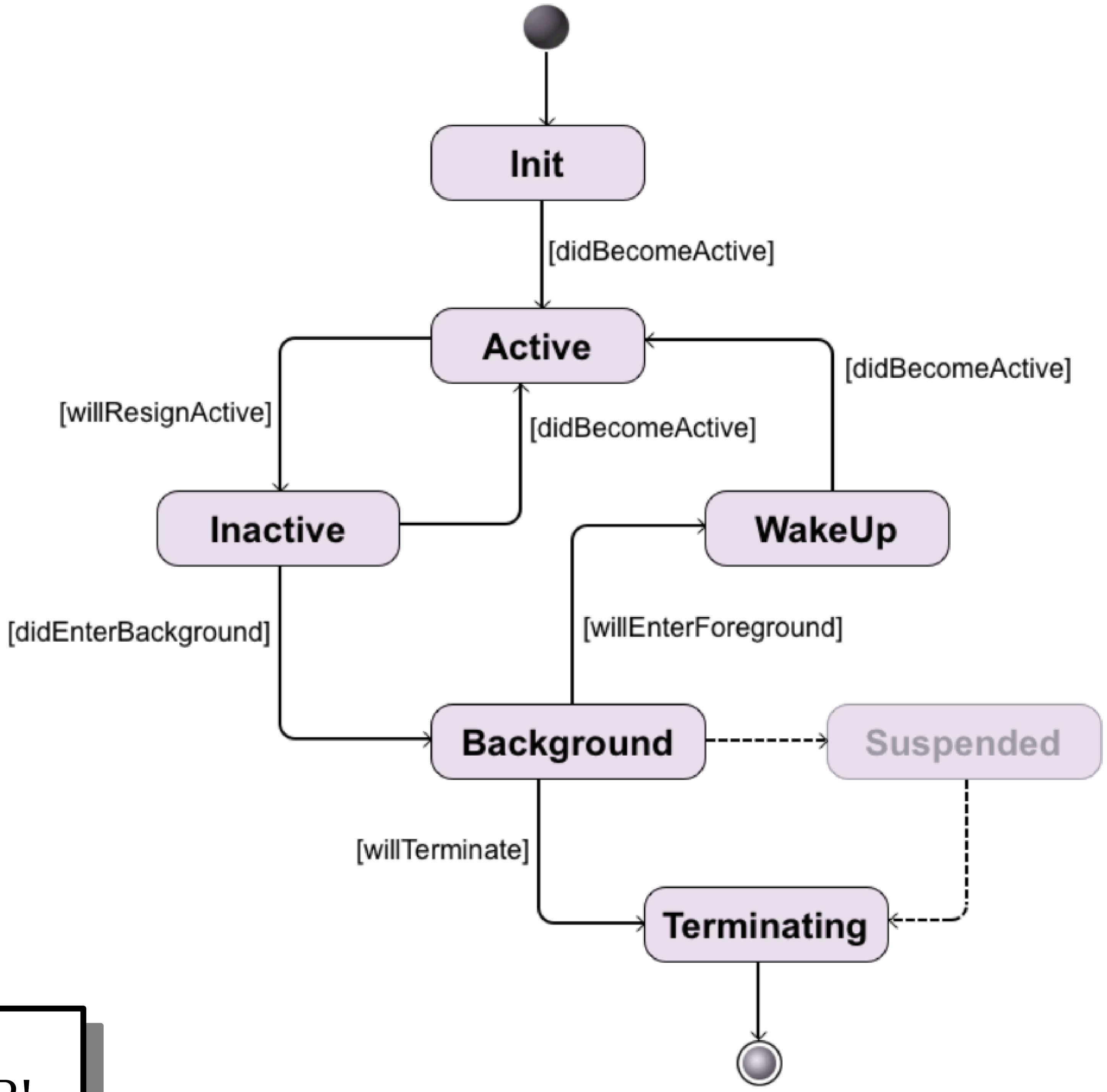
http://cs103.stanford.edu/tools/button-fsm/

Take CS148!



Take CS144!

http://www.tti.unipa.it/~gneglia/ip_networks06/slides/TCPIP_State_Transition_Diagram.pdf



Take
CS193P!

https://blog.codecentric.de/en/2016/07/handling-ios-app-states-state-machine/

Computers as Finite Automata

- My computer has 12GB of RAM and about 150GB of hard disk space.
- That's a total of 162GB of memory, which is 1,391,569,403,904 bits.
- There are “only” $2^{1,391,569,403,904}$ possible configurations of the memory in my computer.
- You could in principle build a DFA representing my computer, where there's one symbol per type of input the computer can receive.

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 is **no possible** DFA for it.
  - *(or no possible NFA, or regex---but since these are equivalent we only need to show one is impossible)*
- ***This sort of argument will be challenging!***

Finding Nonregular Languages

- What kind of characteristics make a language too hard for any of these to handle?
  - Deterministic Finite Automata (DFA)
  - Nondeterministic Finite Automata (NFA)
  - Regular Expression

PollEv.com/cs103spr25:
T/F: You can't make a DFA for a language (set of strings) with infinite cardinality, so a language with infinite cardinality cannot be a regular langauge.

A Simple Language

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

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

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

{ ε, **ab**, **aabb**, **aaabbb**, **aaaabbbb**, … }

A Simple Language

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

PollEv.com/cs103spr25: Which of the following are correct regular expressions for the language E defined above?

a*b*

(ab)*

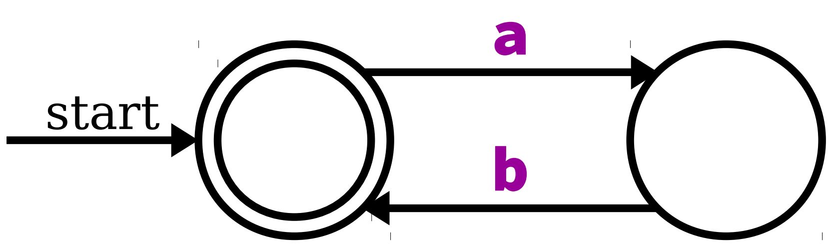
$\boldsymbol{\varepsilon} \cup \mathbf{ab} \cup \mathbf{a^2b^2} \cup \mathbf{a^3b^3}$

Another Attempt

- Let's try to design an NFA for

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

- Does this machine work?



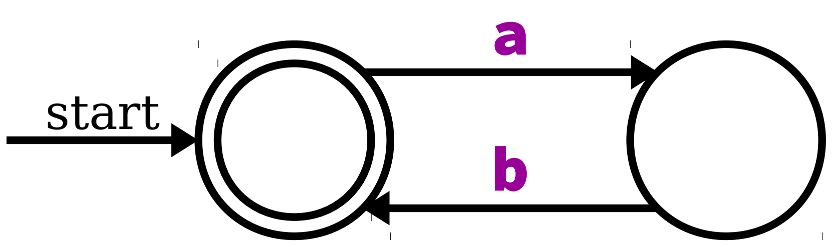
Another Attempt

- Let's try to design an NFA for

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

- Does this machine work?

PollEv.com/cs103spr25:
What is a regex that describes the language of this NFA?

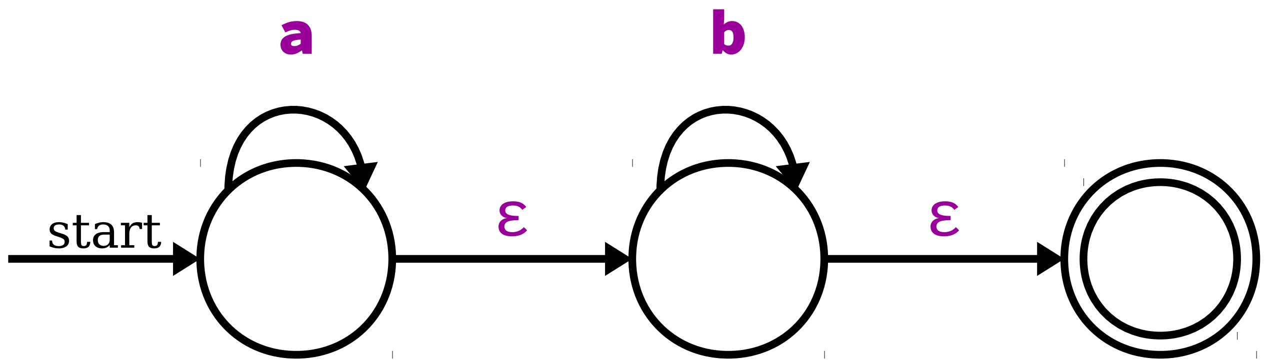


Another Attempt

- Let’s try to design an NFA for

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

- How about this one?



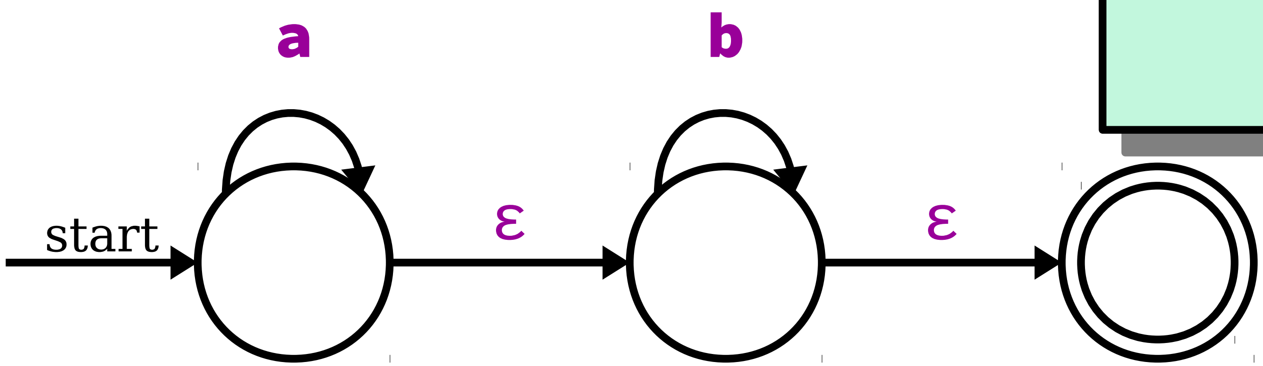
Another Attempt

- Let's try to design an NFA for

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

- How about this one?

PollEv.com/cs103spr25: What is a regex that describes the language of this NFA?

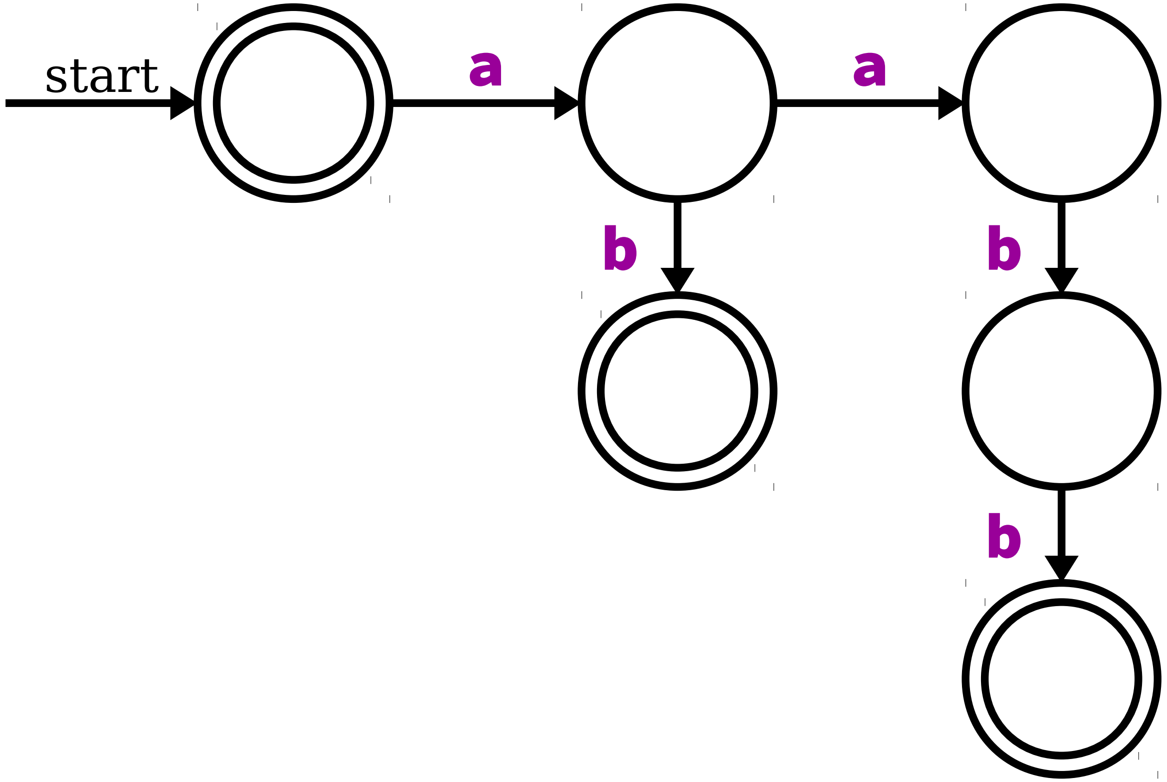


Another Attempt

- Let's try to design an NFA for

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

- What about this?

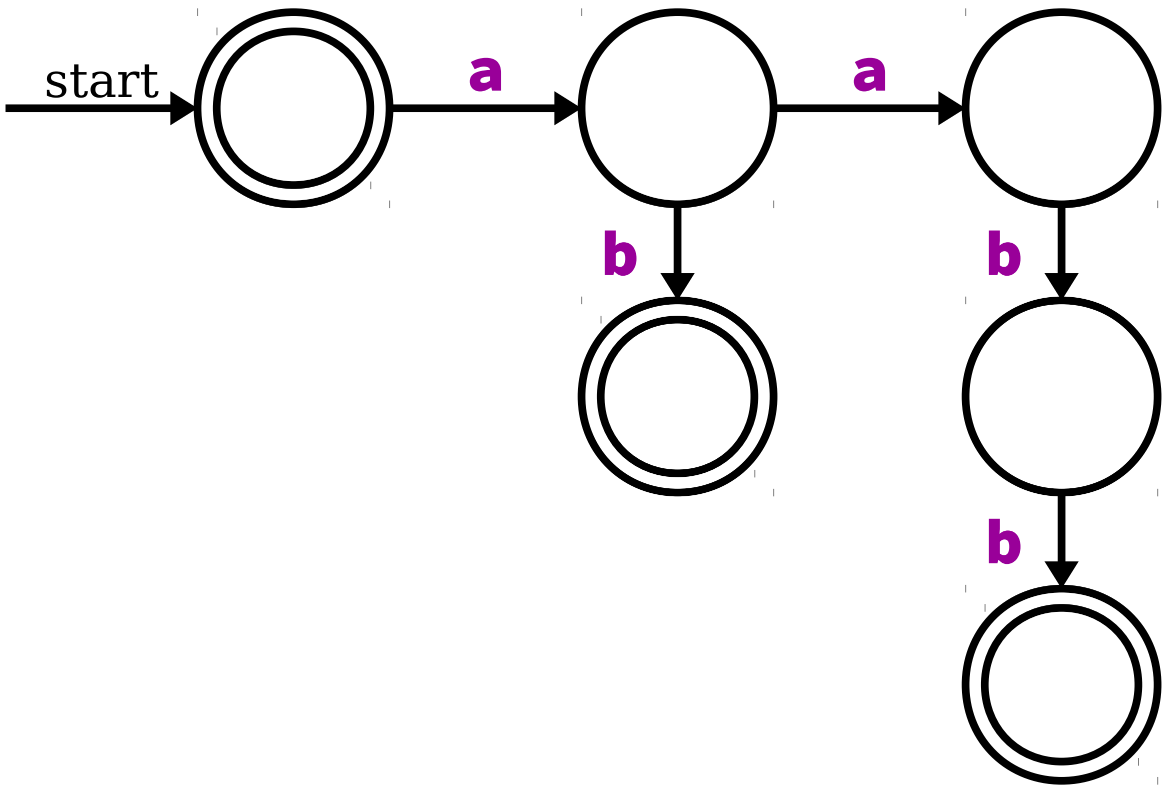


Another Attempt

- Let's try to design an NFA for

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

- What about this?

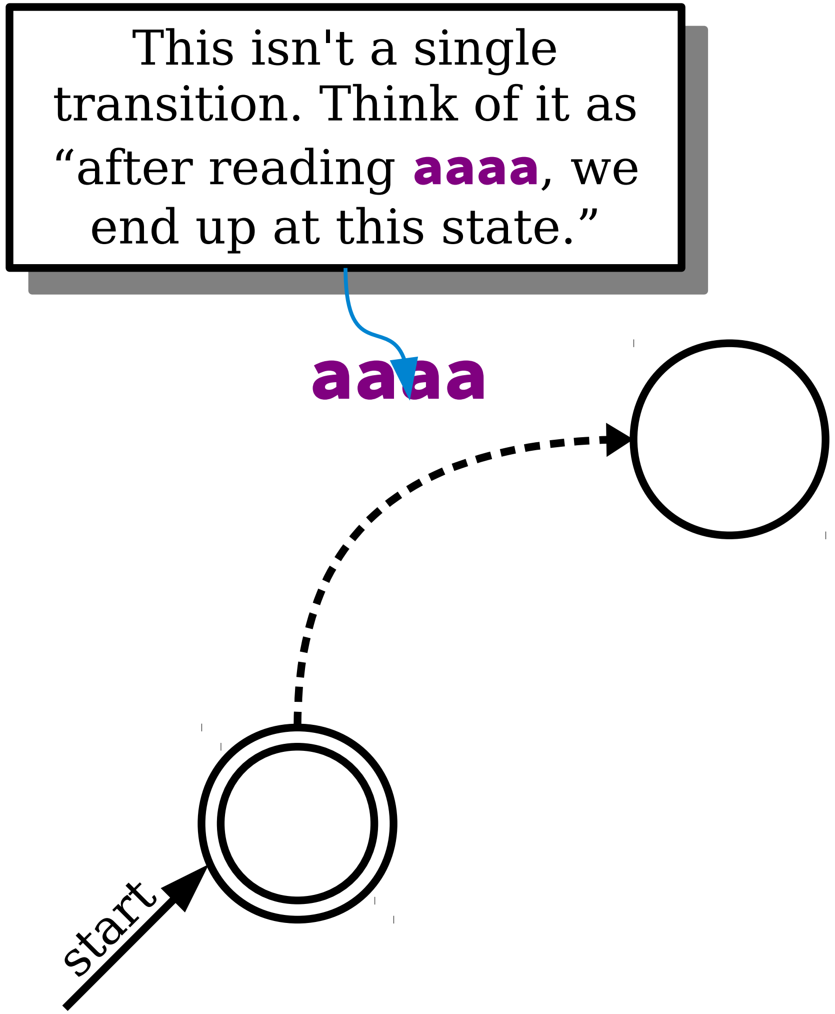


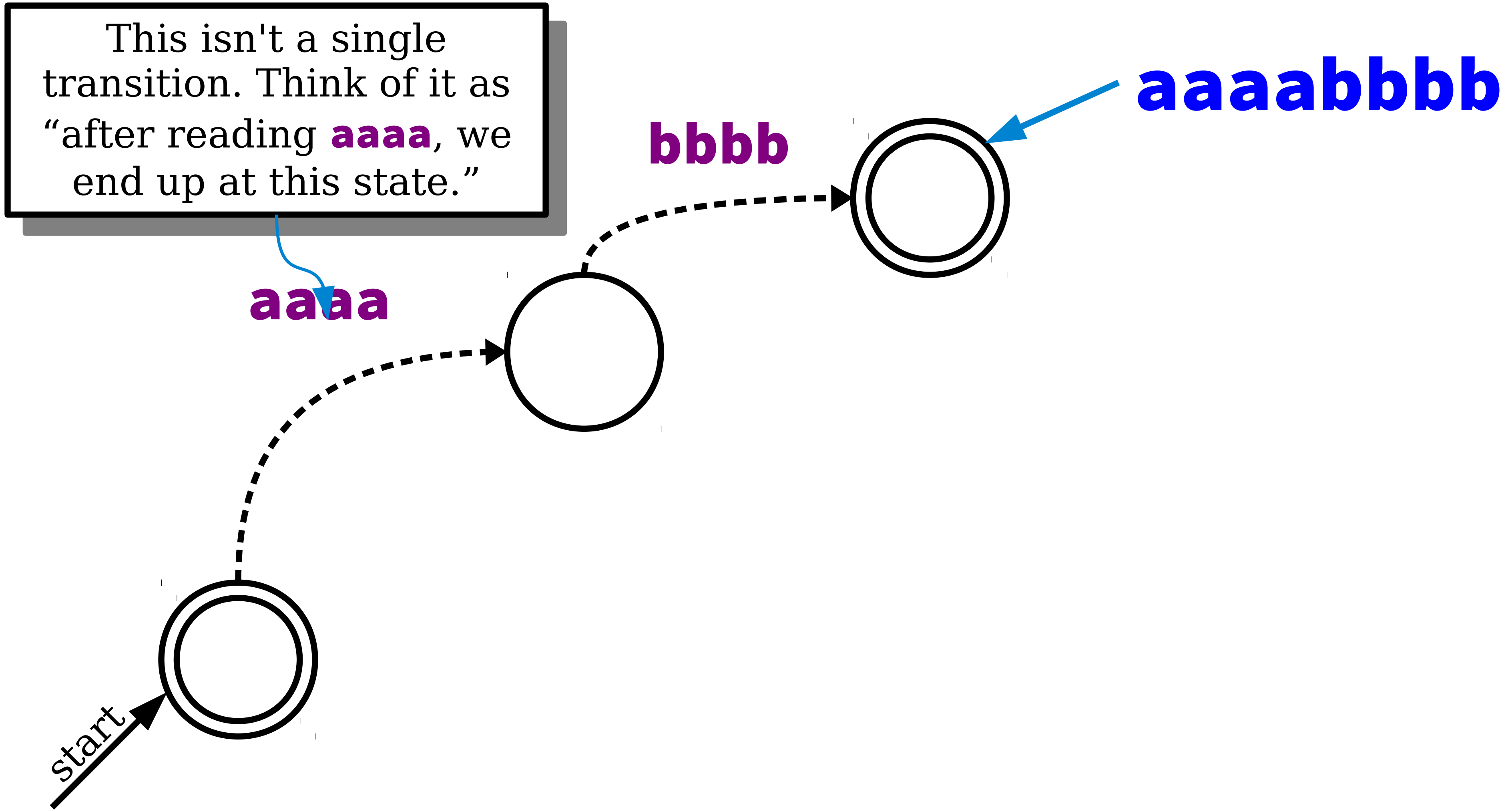
PollEv.com/cs103spr25:
What is a regex that describes the language of this NFA?

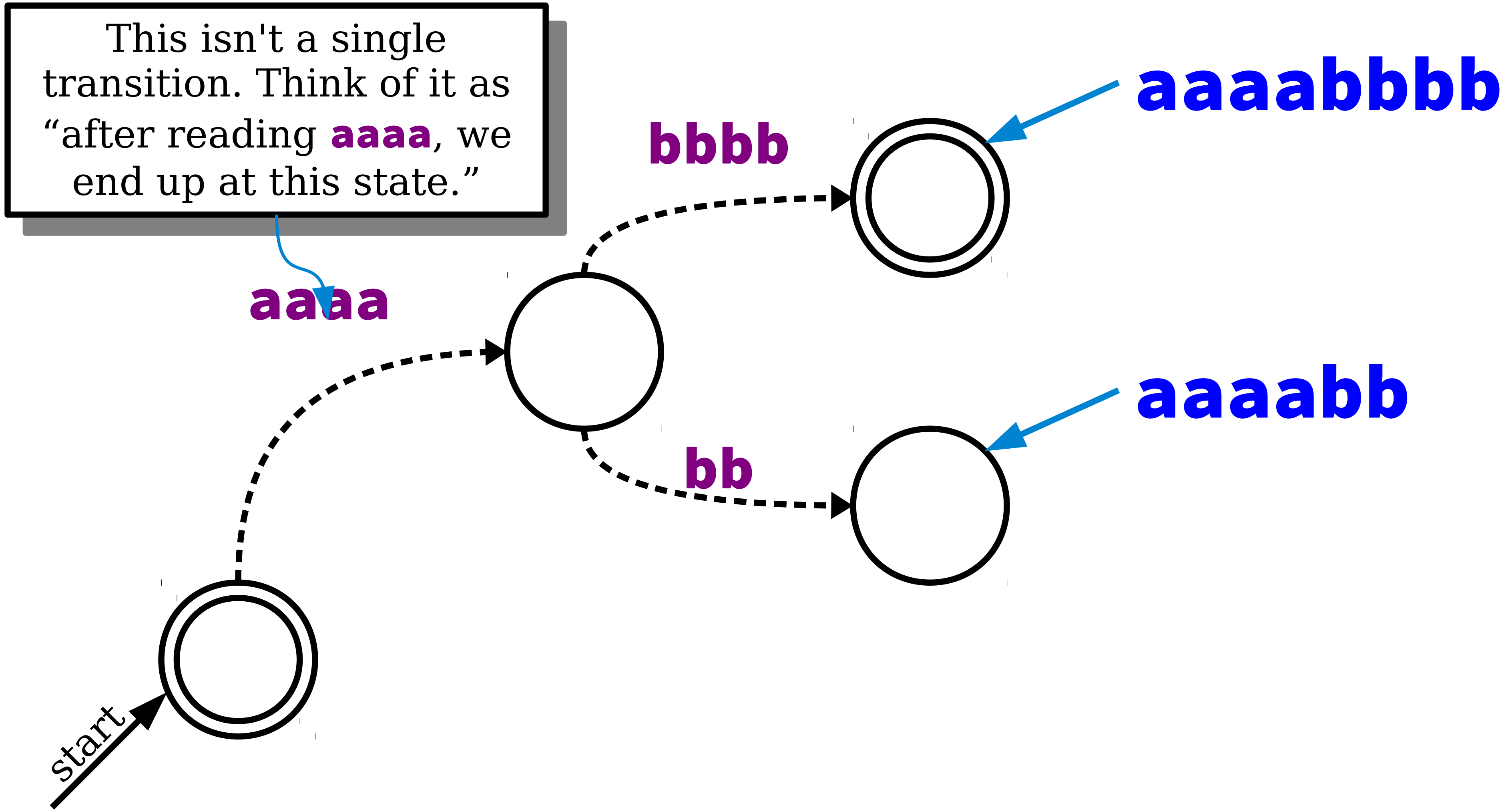
We seem to be running into some trouble.
Why is that?

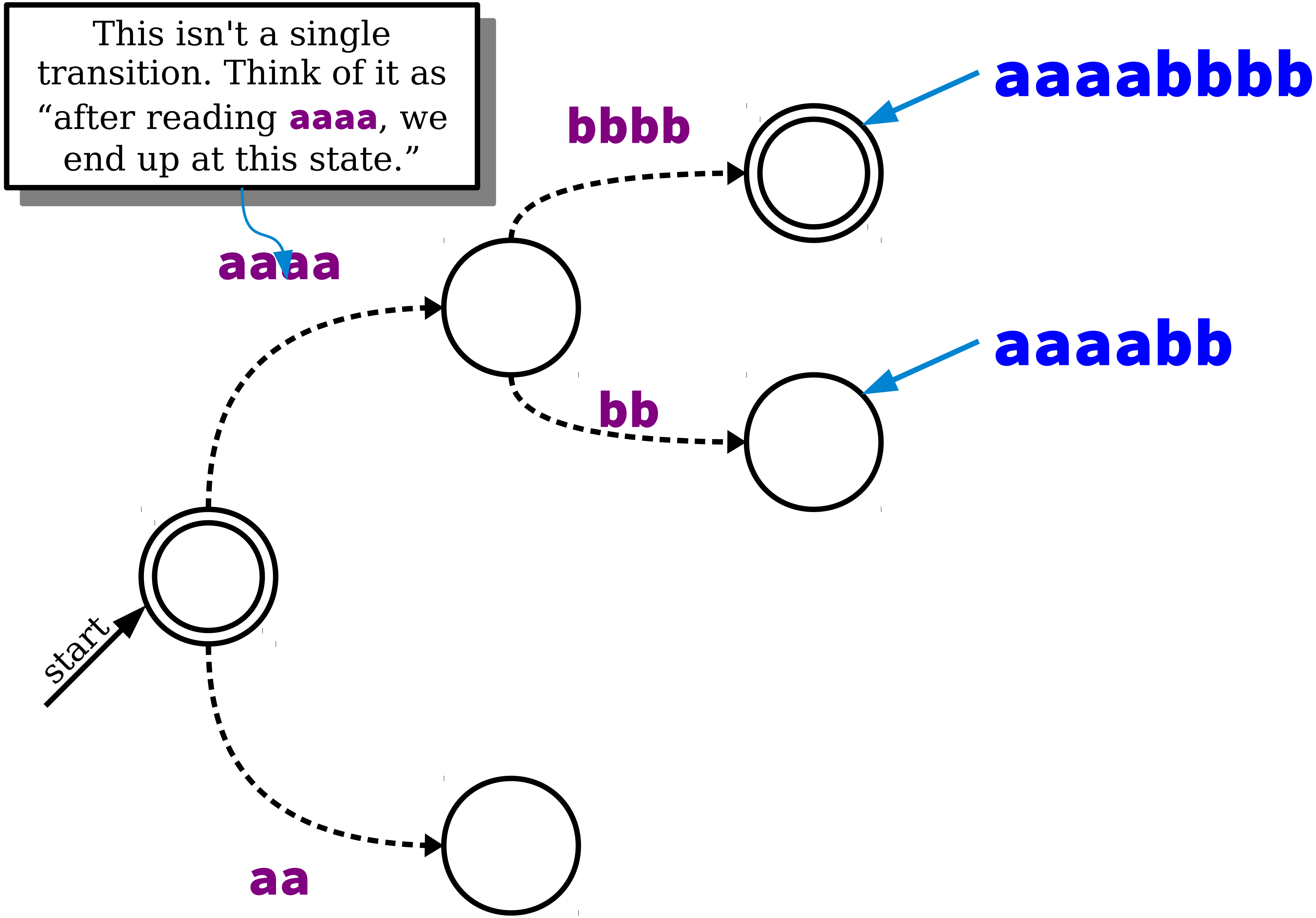
Let's imagine what a DFA for the language $\{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}\}$ would have to look like.

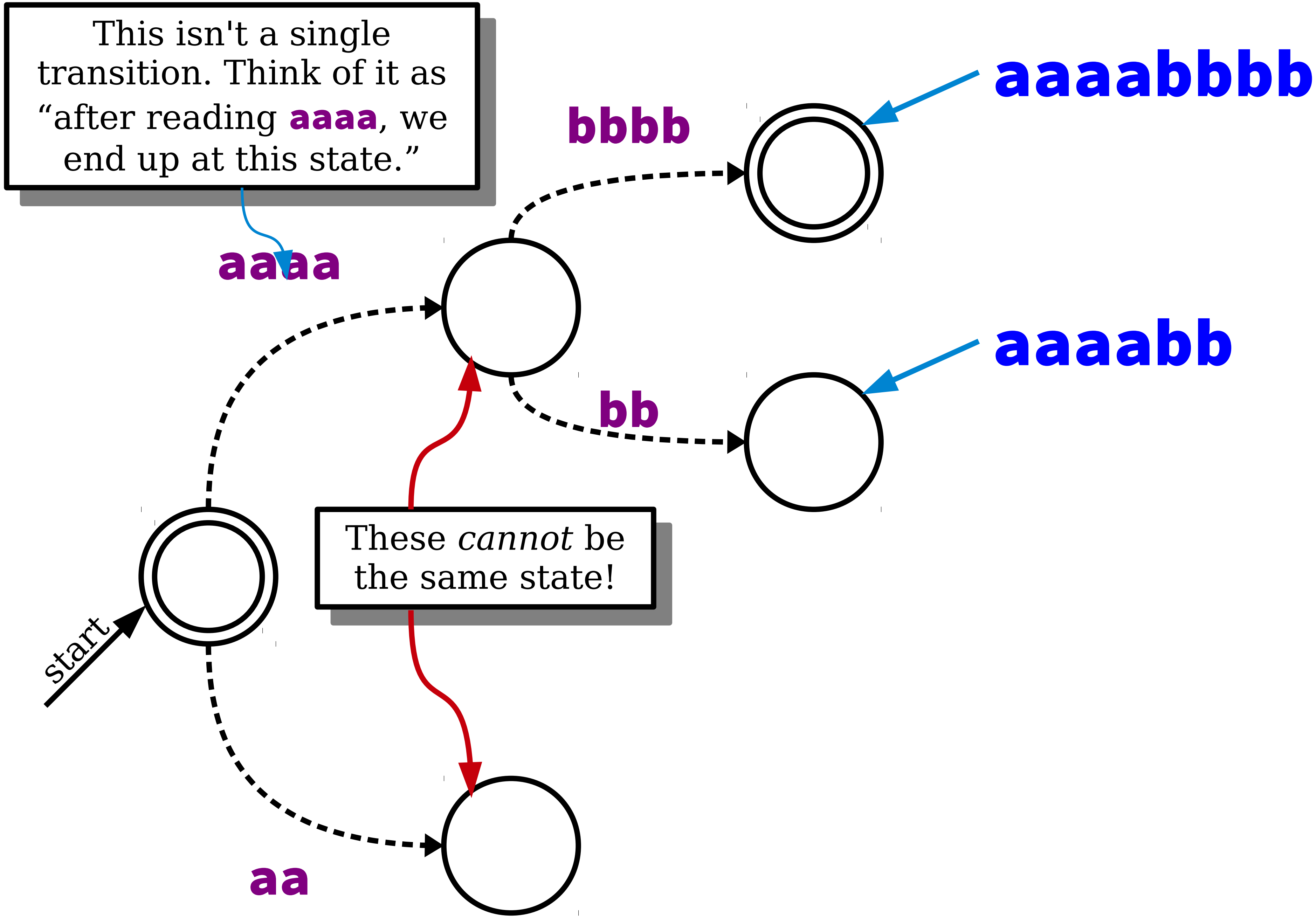
Can we say anything about it?

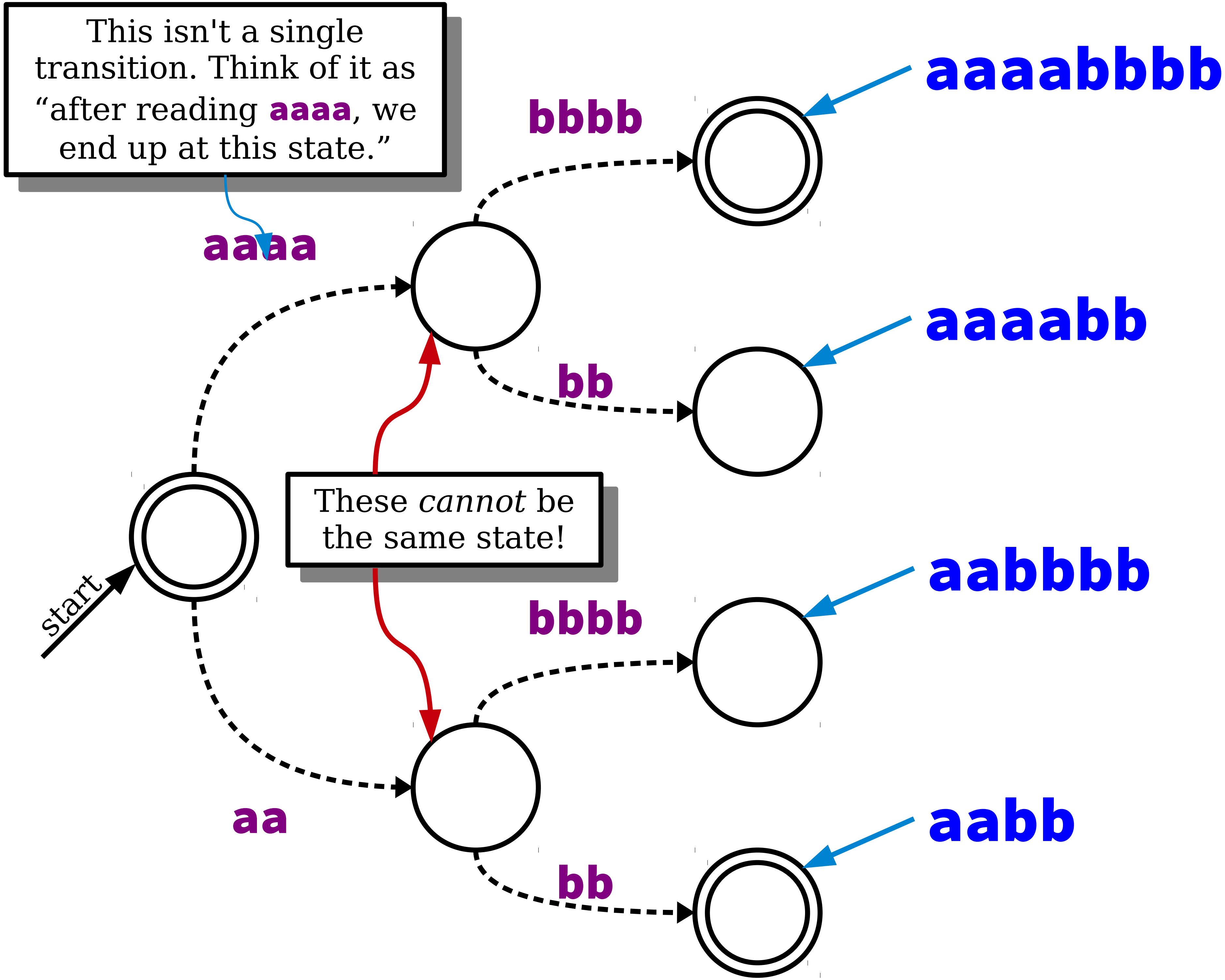

This isn't a single transition. Think of it as "after reading **aaaa**, we end up at this state."
aaaa
start


This isn't a single transition. Think of it as "after reading aaaa, we end up at this state."
aaaabbbb
bbbb
aaaa
start

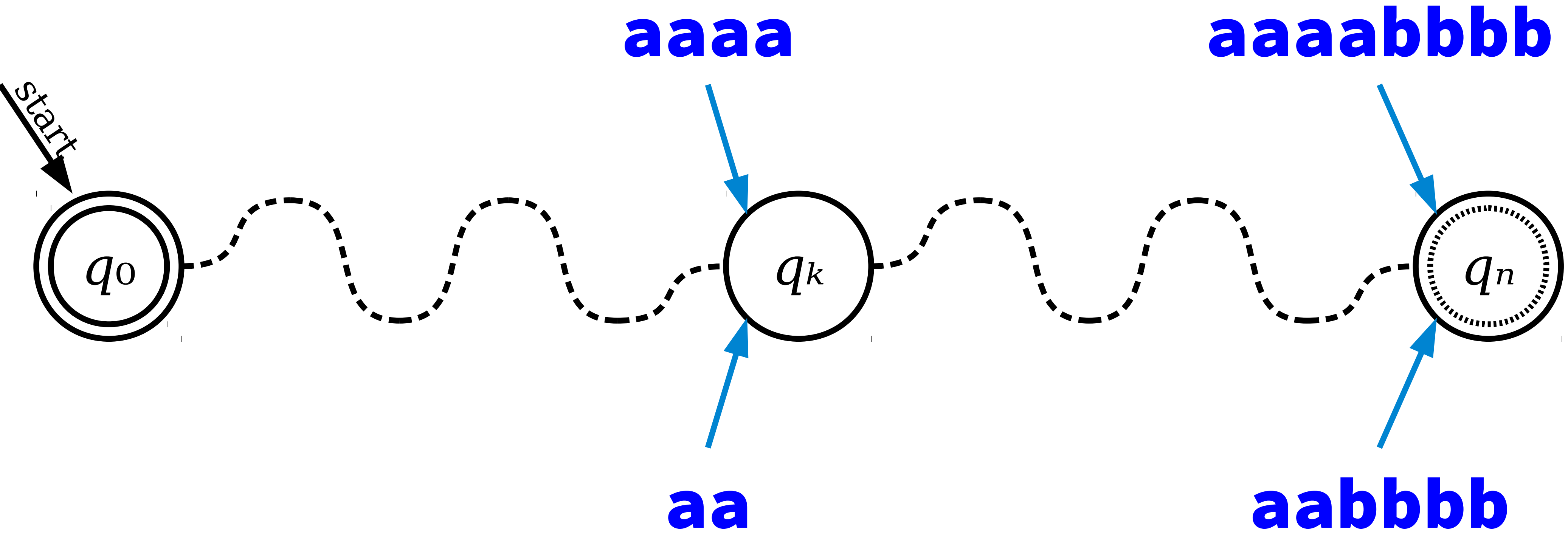

This isn't a single transition. Think of it as "after reading aaaa, we end up at this state."
aaaa
bbbb
bb
aaaabbbb
aaaabb
start


This isn't a single transition. Think of it as “after reading aaaa, we end up at this state.”
aaaabbbb
bbbb
aaaa
aaaabb
bb
start
aa

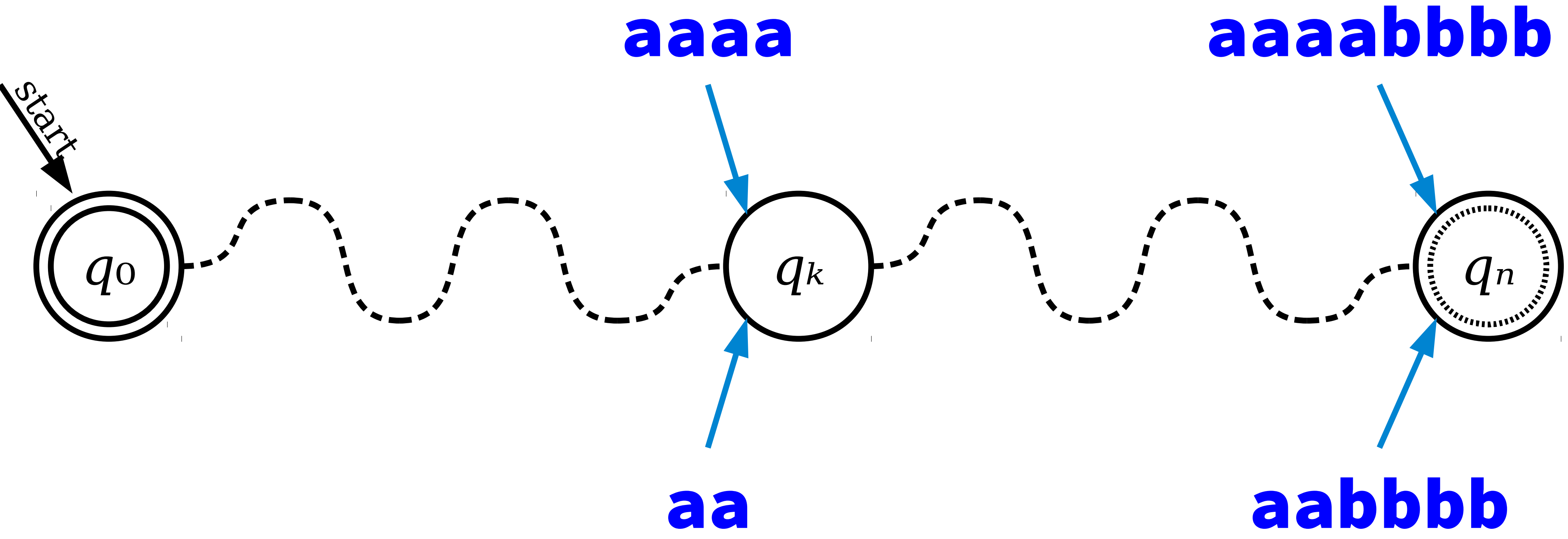

This isn't a single transition. Think of it as "after reading aaaa, we end up at this state."
aaaa
bbbb
aaaabbbb
aaaabb
bb
start
These cannot be the same state!
aa


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



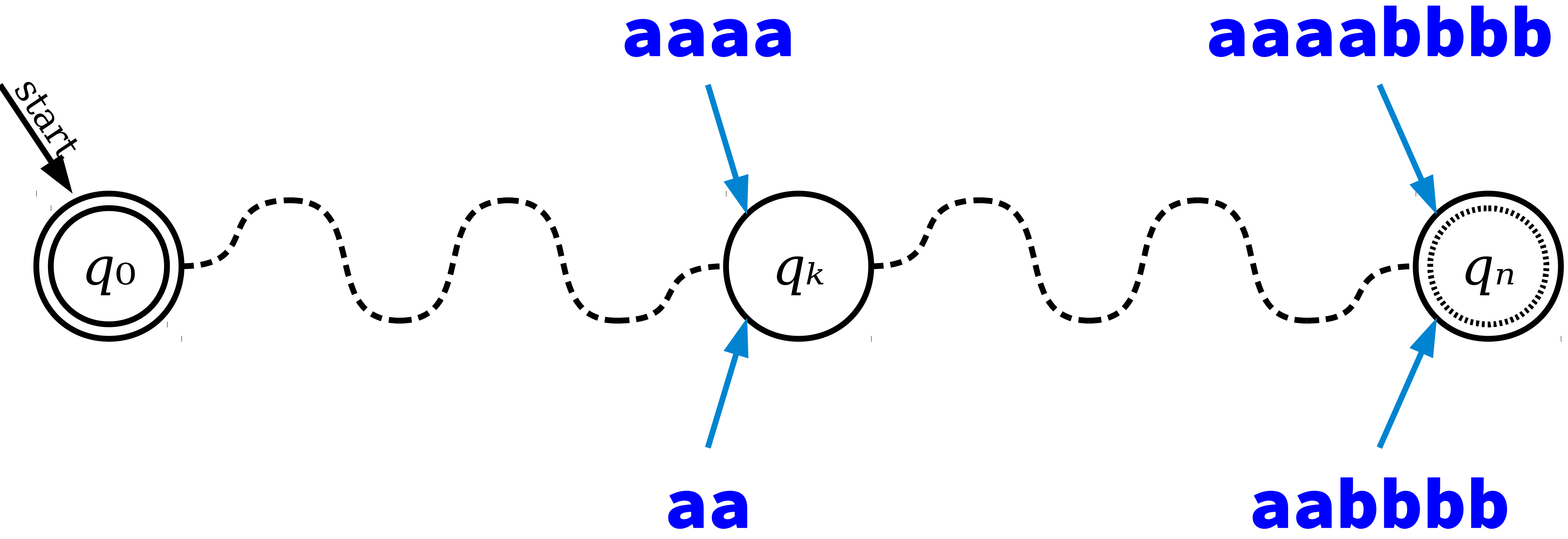
A Different Perspective



What happens if q_n is…

…an accepting state?
…a rejecting state?

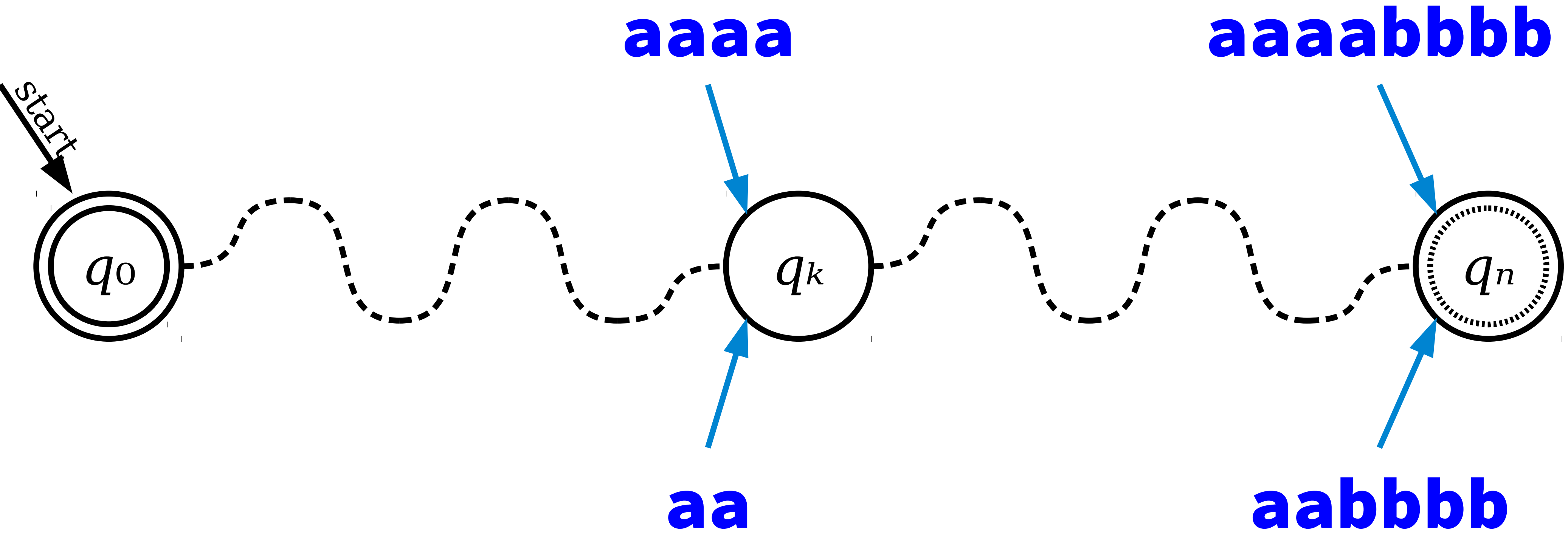
A Different Perspective



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?

- As you just saw, the strings $\mathbf{a}^4$ and $\mathbf{a}^2$ can't end up in the same state in *any* DFA for $E = \{\mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}\}$.
- Two proof routes:
  - *Direct:* The states you reach for $\mathbf{a}^4$ and $\mathbf{a}^2$ have to behave differently when reading $\mathbf{b}^4$ – in one case it should lead to an accept state, in the other it should lead to a reject state. Therefore, they must be different states.
  - *Contradiction:* Suppose you do end up in the same state. Then $\mathbf{a}^4\mathbf{b}^4$ and $\mathbf{a}^2\mathbf{b}^4$ end up in the same state, so we either reject $\mathbf{a}^4\mathbf{b}^4$ (oops) or accept $\mathbf{a}^2\mathbf{b}^4$ (oops).
- ***Powerful intuition:*** Any DFA for E must keep $\mathbf{a}^4$ and $\mathbf{a}^2$ separated. It needs to remember something fundamentally different after reading those strings.

This idea – that two strings shouldn't end up in the same DFA state – is fundamental to discovering nonregular languages.

Let's go formalize this!

Distinguishability

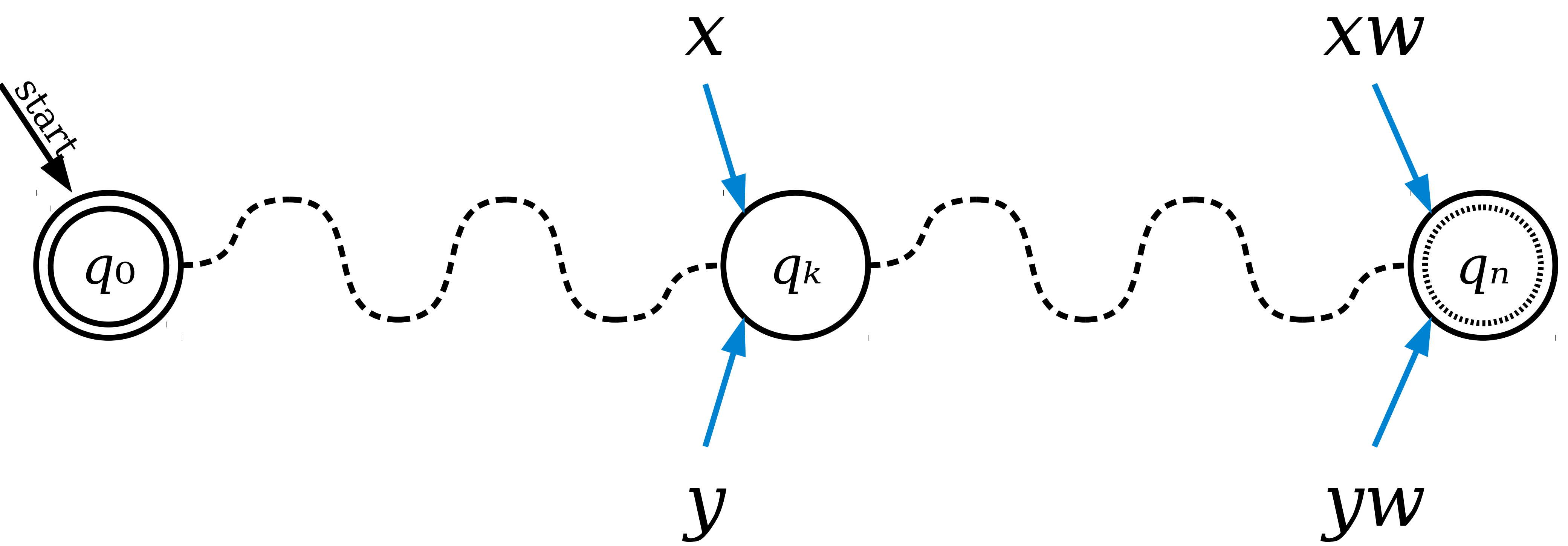
- Let L be an arbitrary language over Σ.
- Two strings $x \in \Sigma^*$ and $y \in \Sigma^*$ are called ***distinguishable relative to L*** if there is a string $w \in \Sigma^*$ such that exactly one of xw and yw is in L.
- We denote this by writing $x \not\equiv_L y$.
- In our previous example, we saw that $\mathbf{a}^2 \not\equiv_E \mathbf{a}^4$.
  - Try appending $\mathbf{b}^4$ to both of them.
- Formally, we say that $x \not\equiv_L y$ if the following is true:

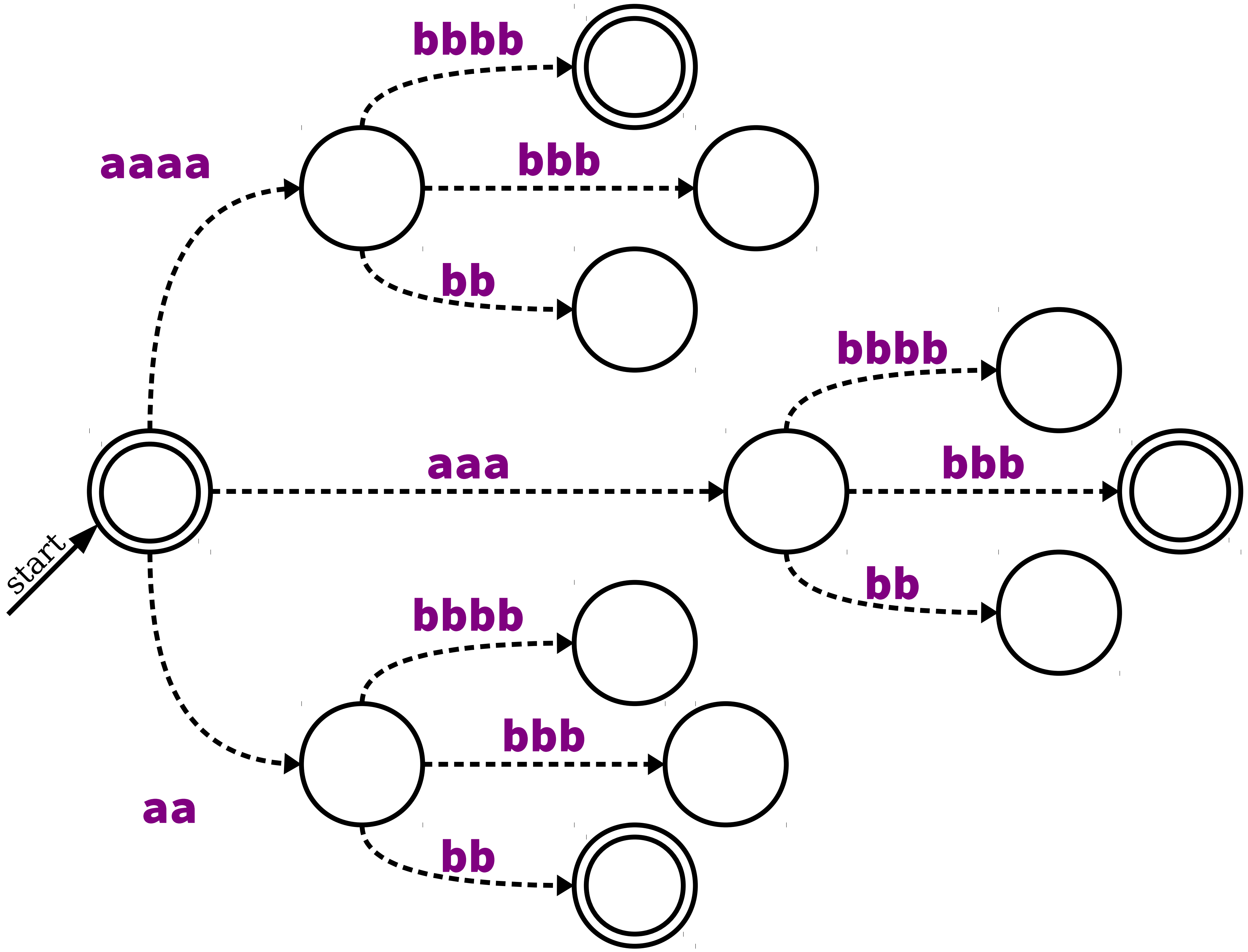
$$\exists w \in \Sigma^*. (xw \in L \leftrightarrow yw \notin L)$$

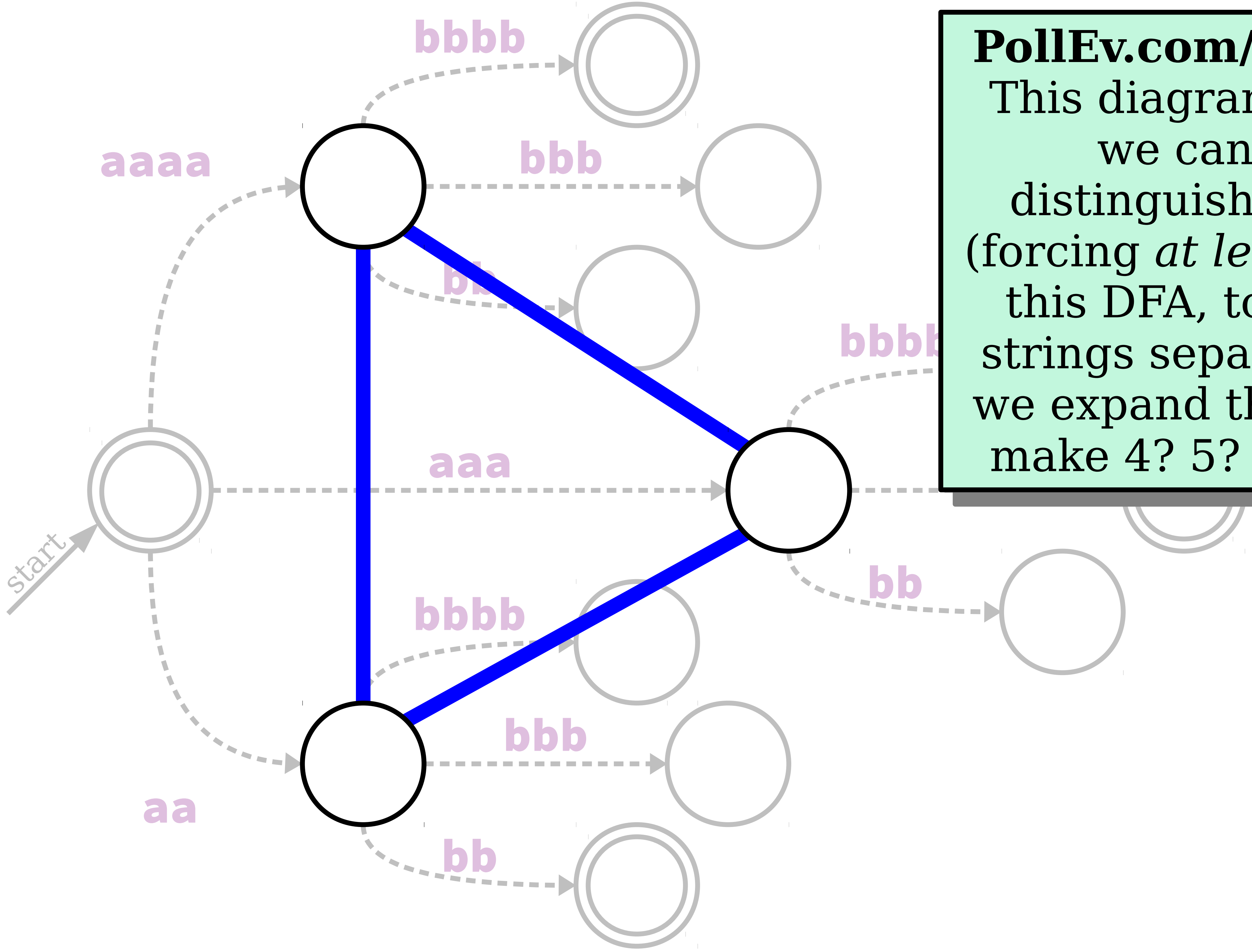
Write this down!

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:***




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



PollEv.com/cs103spr25: This diagram shows that we can make 3 distinguishable strings (forcing *at least* 3 states in this DFA, to keep these strings separated). Could we expand the diagram to make 4? 5? **How many?**

A Bad Combination

- Suppose there is a DFA D for the language $E = \{\mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N}\}$.
- We know the following:
  - Any two strings of the form $\mathbf{a}^m$ and $\mathbf{a}^n$, where $m \neq n$, cannot end in the same state when run through D.
  - There are **infinitely many** strings of the form $\mathbf{a}^m$.
  - 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?

Distinguishing Sets

We say that ***x* $\not\equiv_L$ *y*** if the following is true:
$\exists w \in \Sigma^*. (xw \in L \leftrightarrow yw \notin L)$

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

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

Distinguishing Sets

We say that $\boldsymbol{x \not\equiv_L y}$ if the following is true:

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

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

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

If you pick any two distinct strings in *S*…

… then they're distinguishable relative to *L*.

Distinguishing Sets

> We say that $x \not\equiv_L y$ if the following is true:
> $\exists w \in \Sigma^*. (xw \in L \leftrightarrow yw \notin L)$

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

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

- As an example, here's a distinguishing set for $E = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N} \}$:

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

> **IMPORTANT:**
> A distinguishing set for the language E is **not a subset** of E. It is a set of **prefixes** (beginning part) of strings in E.

Theorem (Myhill-Nerode): If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

Proof: Let L be an arbitrary language over Σ and let S be a distinguishing set for L that contains infinitely many strings. We will show that L is not regular.

Suppose for the sake of contradiction that L is regular. This means that there must be some DFA D for L. Let k be the number of states in D. Since there are infinitely many strings in S, we can choose $k+1$ distinct strings from S and consider what happens when we run D on all of those strings. Because there are only k states in D and we've chosen $k+1$ strings from S, by the pigeonhole principle we know that at least two strings from S must end in the same state in D. Choose any two such strings and call them x and y.

Because $x \neq y$ and S is a distinguishing set for L, we know that $x \not\equiv_L y$. Our earlier theorem therefore tells us that when we run D on inputs x and y, they must end up in different states. But this is impossible – we chose x and y precisely because they 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 for $E = \{ \mathbf{a}^n\mathbf{b}^n \mid n \in \mathbb{N} \}$

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

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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

Proof:

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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

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

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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

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

> ***Myhill-Nerode Theorem:*** If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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

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

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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

Proof: Let $S = \{ \mathbf{a}^n \mid n \in \mathbb{N} \}$. 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 strings strings $\mathbf{a}^m, \mathbf{a}^n \in S$ where $m \neq n$.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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

Proof: Let $S = \{ \mathbf{a}^n \mid n \in \mathbb{N} \}$. 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 strings strings $\mathbf{a}^m$, $\mathbf{a}^n \in S$ where $m \neq n$. Note that $\mathbf{a}^m\mathbf{b}^m \in E$ and that $\mathbf{a}^n\mathbf{b}^m \notin E$.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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

Proof: Let $S = \{ \mathbf{a}^n \mid n \in \mathbb{N} \}$. 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 strings strings $\mathbf{a}^m, \mathbf{a}^n \in S$ where $m \neq n$. Note that $\mathbf{a}^m\mathbf{b}^m \in E$ and that $\mathbf{a}^n\mathbf{b}^m \notin E$. Therefore, we see that $\mathbf{a}^m \not\equiv_E \mathbf{a}^n$, as required.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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

Proof: Let $S = \{ \mathbf{a}^n \mid n \in \mathbb{N} \}$. 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 strings strings $\mathbf{a}^m, \mathbf{a}^n \in S$ where $m \neq n$. Note that $\mathbf{a}^m\mathbf{b}^m \in E$ and that $\mathbf{a}^n\mathbf{b}^m \notin E$. Therefore, we see that $\mathbf{a}^m \not\equiv_E \mathbf{a}^n$, 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.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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

Proof: Let $S = \{ \mathbf{a}^n \mid n \in \mathbb{N} \}$. 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 strings strings $\mathbf{a}^m$, $\mathbf{a}^n \in S$ where $m \neq n$. Note that $\mathbf{a}^m\mathbf{b}^m \in E$ and that $\mathbf{a}^n\mathbf{b}^m \notin E$. Therefore, we see that $\mathbf{a}^m \not\equiv_E \mathbf{a}^n$, 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. ■

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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!

More Nonregular Languages

Another Language

- Consider the following language EQ over the alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}, \stackrel{?}{=}\}$:

$$EQ = \{\ w \stackrel{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^* \}$$

- EQ is the language all strings consisting of the same string of **a**'s and **b**'s twice, with a $\stackrel{?}{=}$ symbol in-between.
- Examples:

$\mathbf{ab} \stackrel{?}{=} \mathbf{ab} \in EQ$ $\mathbf{bbb} \stackrel{?}{=} \mathbf{bbb} \in EQ$ $\stackrel{?}{=} \in EQ$

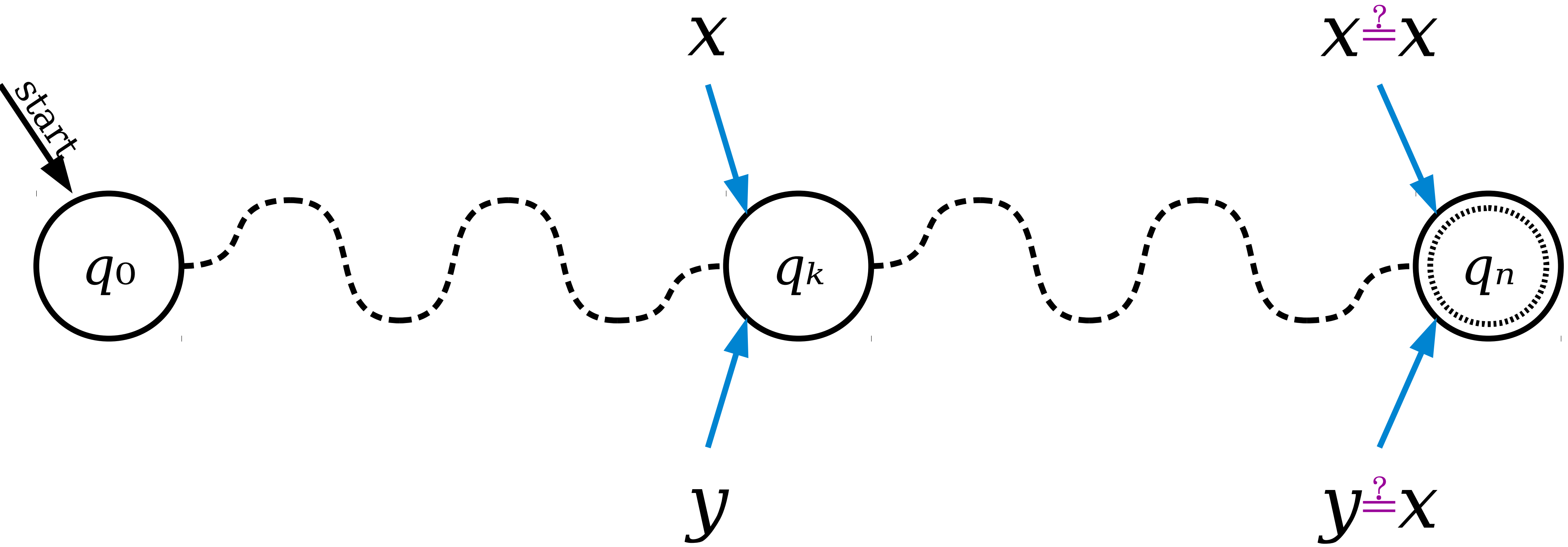
$\mathbf{ab} \stackrel{?}{=} \mathbf{ba} \notin EQ$ $\mathbf{bbb} \stackrel{?}{=} \mathbf{aaa} \notin EQ$ $\mathbf{b} \stackrel{?}{=} \notin EQ$

The Intuition

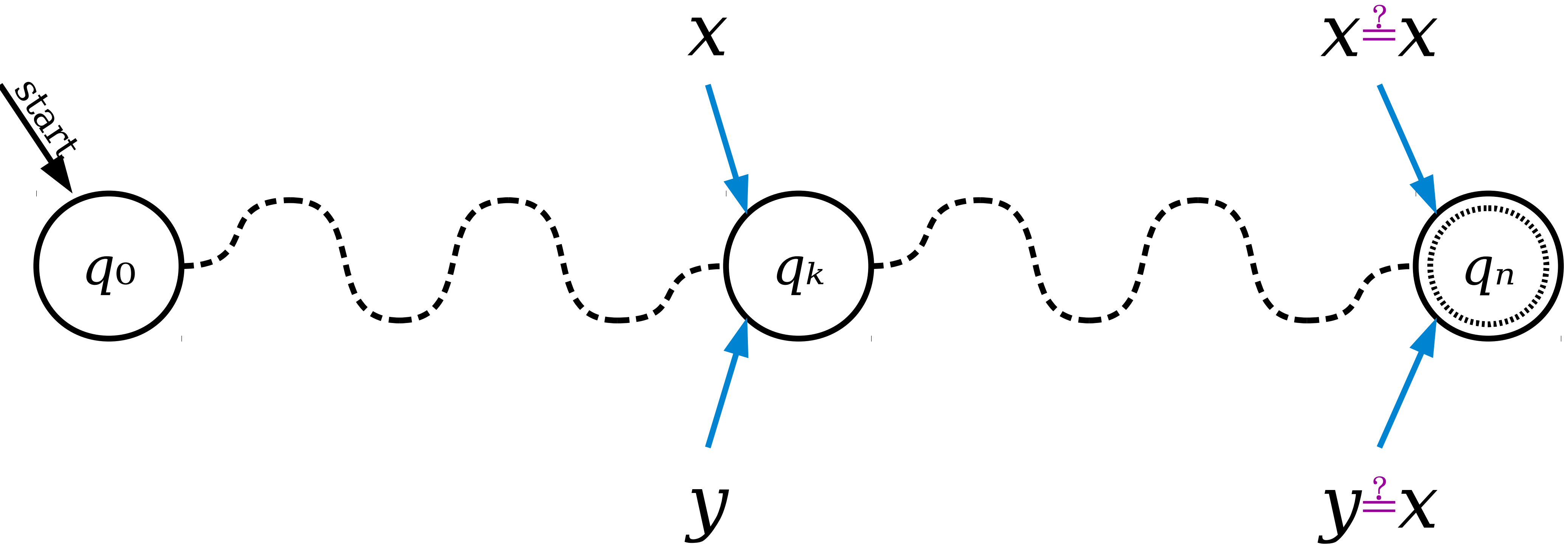
$$EQ = \{\ w \stackrel{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^* \}$$

- Intuitively, any machine for EQ has to be able to remember the contents of everything to the left of the $\stackrel{?}{=}$ so that it can match them against the contents of the string to the right of the $\stackrel{?}{=}$.
- There are infinitely many possible strings we can see, but we only have finite memory to store which string we saw.
- That's a problem... can we formalize this?

The Intuition



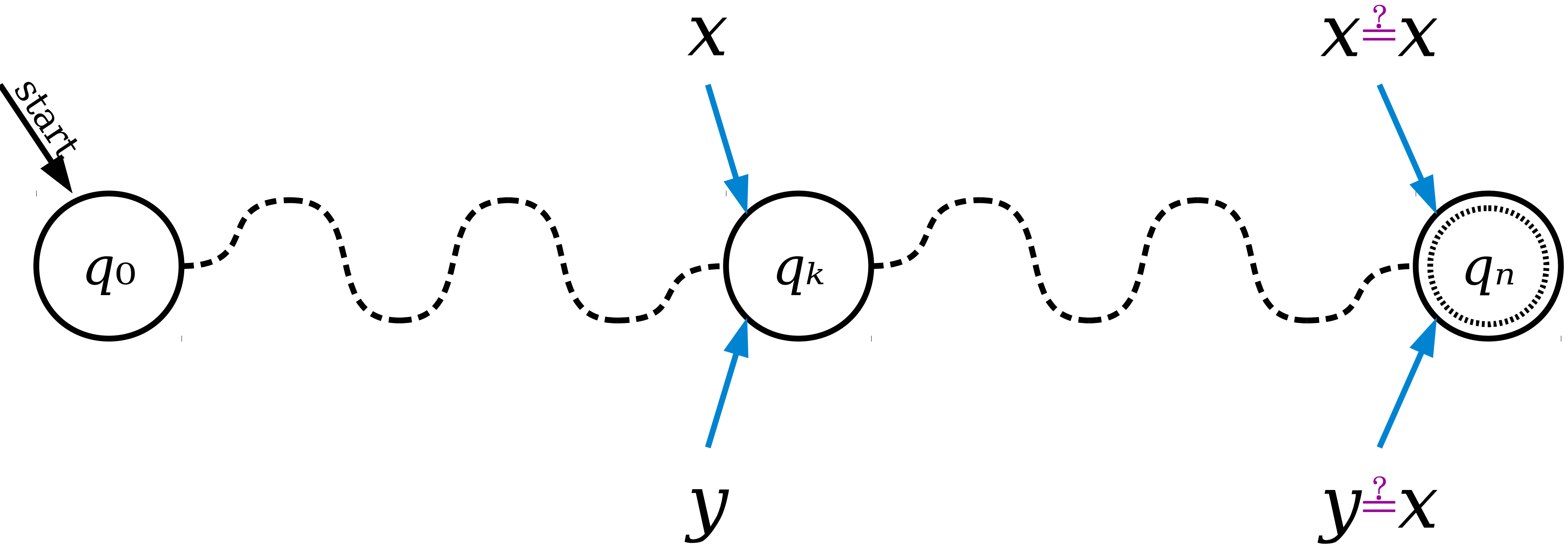
The Intuition



What happens if q_n is…

…an accepting state?
…a rejecting state?

The Intuition

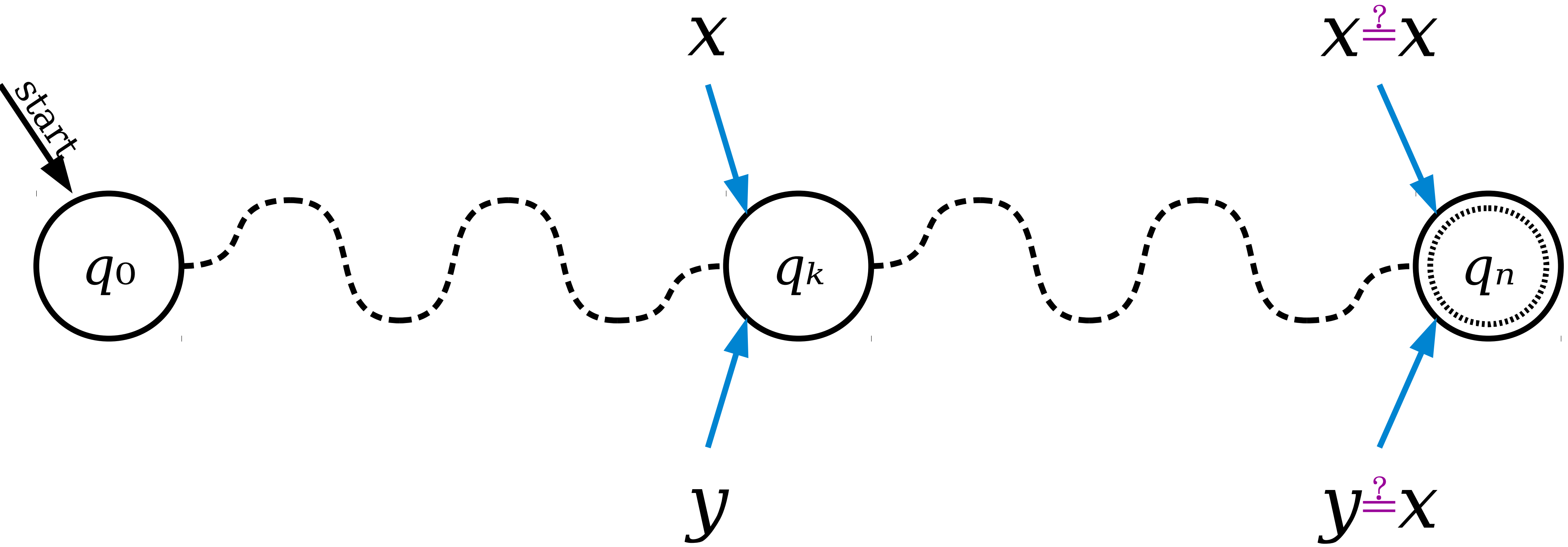


What happens if q_n is…

…an accepting state? We accept $y \stackrel{?}{=} x \notin EQ$!

…a rejecting state?

The Intuition



What happens if q_n is…

…an accepting state? We accept $y \stackrel{?}{=} x \notin EQ$!

…a rejecting state? We reject $x \stackrel{?}{=} x \in EQ$!

Another Language

- Consider the following language E the alphabet $\Sigma = \{$**a**, **b**, $\stackrel{?}{=}\}$:

$$EQ = \{\ w\stackrel{?}{=}w \mid w \in \{\mathbf{a}, \mathbf{b}\}^*\}$$

- EQ is the language all strings cons the same string of **a**'s and **b**'s twice $\stackrel{?}{=}$ symbol in-between.
- Examples:

ab≟ab $\in EQ$ **bbb≟bbb** $\in EQ$ ≟ $\in EQ$

ab≟ba $\notin EQ$ **bbb≟aaa** $\notin EQ$ **b≟** $\notin$

PollEv.com/cs103spr25:
Which of these are good infinite distinguishing sets for EQ that we could use in a Myhill-Nerode proof?

1. $\{\ \mathbf{a}^n \mid n \in \mathbb{N}\ \}$
2. $\{\ \mathbf{b}^n \mid n \in \mathbb{N}\ \}$
3. $\{\mathbf{a}, \mathbf{b}\}^*$
4. $\{\ \mathbf{a}^n\stackrel{?}{=}\mathbf{a}^n \mid n \in \mathbb{N}\ \}$
5. $\{\ \mathbf{a}^n\stackrel{?}{=} \mid n \in \mathbb{N}\ \}$

We say that $x \not\equiv_L y$ if the following is true:

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

Distinguishing Sets

- We've identified a distinguishing set for $EQ = \{ w \stackrel{?}{=} w \mid w \in \{\mathtt{a}, \mathtt{b}\}^* \}$:

$$S = \{\mathtt{a}, \mathtt{b}\}^*$$

We say that $x \not\equiv_L y$ if the following is true:
$\exists w \in \Sigma^*. (xw \in L \leftrightarrow yw \notin L)$

IMPORTANT:
A distinguishing set for the language *EQ* is **not a subset** of *EQ*. It is a set of **prefixes** (beginning part) of strings in *EQ*.

Distinguishing Sets

- We've identified a distinguishing set for $EQ = \{ w \stackrel{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^*\}$:

$$S = \{\mathbf{a}, \mathbf{b}\}^*$$

We say that $x \not\equiv_L y$ if the following is true:
$\exists w \in \Sigma^*. (xw \in L \leftrightarrow yw \notin L)$

IMPORTANT:
A distinguishing set for the language *EQ* is **not a subset** of *EQ*. It is a set of **prefixes** (beginning part) of strings in *EQ*.

Theorem: The language $EQ = \{ w \stackrel{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^* \}$ is not regular.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

Theorem: The language $EQ = \{ w \stackrel{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^* \}$ is not regular.

Proof:

> ***Myhill-Nerode Theorem:*** If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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

Proof: Let $S = \{\mathtt{a}, \mathtt{b}\}^*$.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

Theorem: The language $EQ = \{ w \stackrel{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^* \}$ is not regular.

Proof: Let $S = \{\mathbf{a}, \mathbf{b}\}^*$. We will prove that S is infinite and that S is a distinguishing set for EQ.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

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

Proof: Let $S = \{\texttt{a}, \texttt{b}\}^*$. We will prove that S is infinite and that S is a distinguishing set for EQ.

To see that S is infinite, note that, for any $n \in \mathbb{N}$, we have $\texttt{a}^n \in S$.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

Theorem: The language $EQ = \{ w \stackrel{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^* \}$ is not regular.

Proof: Let $S = \{\mathbf{a}, \mathbf{b}\}^*$. We will prove that S is infinite and that S is a distinguishing set for EQ.

To see that S is infinite, note that, for any $n \in \mathbb{N}$, we have $\mathbf{a}^n \in S$. Therefore, S contains at least one string for each natural number, so S is infinite.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

Theorem: The language $EQ = \{ w \stackrel{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^* \}$ is not regular.

Proof: Let $S = \{\mathbf{a}, \mathbf{b}\}^*$. We will prove that S is infinite and that S is a distinguishing set for EQ.

To see that S is infinite, note that, for any $n \in \mathbb{N}$, we have $\mathbf{a}^n \in S$. Therefore, S contains at least one string for each natural number, so S is infinite.

To see that S is a distinguishing set for EQ, consider any strings $x, y \in S$ where $x \neq y$.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

Theorem: The language $EQ = \{ w \stackrel{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^* \}$ is not regular.

Proof: Let $S = \{\mathbf{a}, \mathbf{b}\}^*$. We will prove that S is infinite and that S is a distinguishing set for EQ.

To see that S is infinite, note that, for any $n \in \mathbb{N}$, we have $\mathbf{a}^n \in S$. Therefore, S contains at least one string for each natural number, so S is infinite.

To see that S is a distinguishing set for EQ, consider any strings $x, y \in S$ where $x \neq y$. Then $x \stackrel{?}{=} x \in EQ$ and $y \stackrel{?}{=} x \notin EQ$.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

Theorem: The language $EQ = \{ w \stackrel{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^* \}$ is not regular.

Proof: Let $S = \{\mathbf{a}, \mathbf{b}\}^*$. We will prove that S is infinite and that S is a distinguishing set for EQ.

To see that S is infinite, note that, for any $n \in \mathbb{N}$, we have $\mathbf{a}^n \in S$. Therefore, S contains at least one string for each natural number, so S is infinite.

To see that S is a distinguishing set for EQ, consider any strings $x, y \in S$ where $x \neq_{EQ} y$. Then $x \stackrel{?}{=} x \in EQ$ and $y \stackrel{?}{=} x \notin EQ$. Therefore, $x \not\equiv \ y$, as required.

Myhill-Nerode Theorem: If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

Theorem: The language $EQ = \{ w \overset{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^* \}$ is not regular.

Proof: Let $S = \{\mathbf{a}, \mathbf{b}\}^*$. We will prove that S is infinite and that S is a distinguishing set for EQ.

To see that S is infinite, note that, for any $n \in \mathbb{N}$, we have $\mathbf{a}^n \in S$. Therefore, S contains at least one string for each natural number, so S is infinite.

To see that S is a distinguishing set for EQ, consider any strings $x, y \in S$ where $x \neq_{EQ} y$. Then $x \overset{?}{=} x \in EQ$ and $y \overset{?}{=} x \notin EQ$. Therefore, $x \not\equiv y$, as required.

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

> ***Myhill-Nerode Theorem:*** If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

Theorem: The language $EQ = \{ w \stackrel{?}{=} w \mid w \in \{\mathbf{a}, \mathbf{b}\}^* \}$ is not regular.

> ***Myhill-Nerode Theorem:*** If L is a language and S is a distinguishing set for L that contains infinitely many strings, then L is not regular.

Proof: Let $S = \{\mathbf{a}, \mathbf{b}\}^*$. We will prove that S is infinite and that S is a distinguishing set for EQ.

To see that S is infinite, note that, for any $n \in \mathbb{N}$, we have $\mathbf{a}^n \in S$. Therefore, S contains at least one string for each natural number, so S is infinite.

To see that S is a distinguishing set for EQ, consider any strings $x, y \in S$ where $x \neq_{EQ} y$. Then $x \stackrel{?}{=} x \in EQ$ and $y \stackrel{?}{=} x \notin EQ$. Therefore, $x \not\equiv y$, as required.

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

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.

Another Language

- Consider the following language P over the alphabet $\Sigma = \{$**a**, **b**$\}$:

$$P = \{\ w \mid w \text{ is a palindrome}\}$$

- P is the language all strings where the second half is a "mirror" (reverse order) copy of the first half.
- Examples:

abba $\in EQ$ **bbb** $\in EQ$ ***a*** $\in EQ$

abaaba $\in EQ$ **abab** $\notin EQ$ **aabb** $\notin EQ$

Another Language

- Consider the following language P over the alphabet $\Sigma = \{$**a**, **b**$\}$:

$$P = \{ w \mid w \text{ is a palindrome} \}$$

- P is the language all strings where the second half is a "mirror" (reverse copy of the first half.
- Examples:

abba $\in EQ$ **bbb** $\in EQ$ ***a*** $\in EQ$

abaaba $\in EQ$ **abab** $\notin EQ$ **aab**

PollEv.com/cs103spr25:
Which of these are good infinite distinguishing sets for P that we could use in a Myhill-Nerode proof?

1. $\{$ **a**$^n \mid n \in \mathbb{N} \}$
2. $\{$ **b**$^n \mid n \in \mathbb{N} \}$
3. $\{$**a**, **b**$\}$*
4. $\{$ **a**^{n}**b** $\mid n \in \mathbb{N} \}$
5. $\{$ **a**^{n}**b**$^n \mid n \in \mathbb{N} \}$

We say that $x \not\equiv_L y$ if the following is true:
$\exists w \in \Sigma^*. (xw \in L \leftrightarrow yw \notin L)$

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?

Next Time

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