Regular Expressions

and

The Limits of Regular Languages

Announcements

- Midterm ***tonight*** in Cubberly Auditorium, 7PM – 10PM.
  - Open-book, open-note, open-computer, closed-network.
  - Covers material up to and including last Monday's lecture.

Regular Expressions

Atomic Regular Expressions

- The regular expressions begin with three simple building blocks.
- The symbol **Ø** is a regular expression that represents the empty language Ø.
- The symbol **ε** is a regular expression that represents the language { ε }
  - This is not the same as Ø!
- For any `a` $\in \Sigma$, the symbol `a` is a regular expression for the language { `a` }

Compound Regular Expressions

- We can combine together existing regular expressions in four ways.
- If R_1 and R_2 are regular expressions, $\boldsymbol{R_1R_2}$ is a regular expression represents the **concatenation** of the languages of R_1 and R_2.
- If R_1 and R_2 are regular expressions, $\boldsymbol{R_1 \mid R_2}$ is a regular expression representing the **union** of R_1 and R_2.
- If R is a regular expression, $\boldsymbol{R^*}$ is a regular expression for the **Kleene closure** of R.
- If R is a regular expression, $\boldsymbol{(R)}$ is a regular expression with the same meaning as R.

Operator Precedence

- Regular expression operator precedence is

$$(R)$$

$$R^*$$

$$R_1R_2$$

$$R_1 \mid R_2$$

- So `ab*c|d` is parsed as `((a(b*))c)|d`

Regular Expressions, Formally

- The **language of a regular expression** is the language described by that regular expression.
- Formally:
  - $\mathscr{L}(\varepsilon) = \{\varepsilon\}$
  - $\mathscr{L}(\emptyset) = \emptyset$
  - $\mathscr{L}(\mathtt{a}) = \{\mathtt{a}\}$
  - $\mathscr{L}(R_1\ R_2) = \mathscr{L}(R_1)\ \mathscr{L}(R_2)$
  - $\mathscr{L}(R_1 \mid R_2) = \mathscr{L}(R_1) \cup \mathscr{L}(R_2)$
  - $\mathscr{L}(R^*) = \mathscr{L}(R)^*$
  - $\mathscr{L}((R)) = \mathscr{L}(R)$

Regular Expressions are Awesome

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$
- Let $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring $\}$

Regular Expressions are Awesome

- Let $\Sigma = \{$`0`, `1`$\}$
- Let $L = \{\ w \in \Sigma^* \mid w$ contains `00` as a substring $\}$

(0 | 1)*00(0 | 1)*

Regular Expressions are Awesome

- Let $\Sigma = \{$`0`, `1`$\}$
- Let $L = \{\, w \in \Sigma^* \mid w$ contains `00` as a substring $\}$

(0 | 1)*00(0 | 1)*

Regular Expressions are Awesome

- Let $\Sigma = \{$`0`, `1`$\}$
- Let $L = \{\ w \in \Sigma^* \mid w$ contains `00` as a substring $\}$

(0 | 1)*00(0 | 1)*

11011100101
0000
11111011110011111

Regular Expressions are Awesome

- Let $\Sigma = \{$`0`, `1`$\}$
- Let $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring $\}$

(0 | 1)*00(0 | 1)*

11011100101
0000
11111011110011111

Regular Expressions are Awesome

- Let $\Sigma = \{$`0`$,$ `1`$\}$
- Let $L = \{\ w \in \Sigma^* \mid |w| = 4\ \}$

Regular Expressions are Awesome

Let $\Sigma = \{\texttt{0}, \texttt{1}\}$

Let $L = \{ w \in \Sigma^* \mid |w| = 4 \}$

Regular Expressions are Awesome

Let $\Sigma = \{0, 1\}$

Let $L = \{ w \in \Sigma^* \mid |w| = 4 \}$

The length of a string w is denoted |w|

Regular Expressions are Awesome

- Let $\Sigma = \{\mathtt{0}, \mathtt{1}\}$
- Let $L = \{\ w \in \Sigma^* \mid |w| = 4\ \}$

Regular Expressions are Awesome

- Let $\Sigma = \{$`0`, `1`$\}$
- Let $L = \{ w \in \Sigma^* \mid |w| = 4 \}$

(0|1)(0|1)(0|1)(0|1)

Regular Expressions are Awesome

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$
- Let $L = \{ w \in \Sigma^* \mid |w| = 4 \}$

(0|1)(0|1)(0|1)(0|1)

Regular Expressions are Awesome

- Let $\Sigma = \{\mathtt{0}, \mathtt{1}\}$
- Let $L = \{\ w \in \Sigma^* \mid |w| = 4\ \}$

(0|1)(0|1)(0|1)(0|1)

0000
1010
1111
1000

Regular Expressions are Awesome

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$
- Let $L = \{\ w \in \Sigma^* \mid |w| = 4\ \}$

(0|1)(0|1)(0|1)(0|1)

0000
1010
1111
1000

Regular Expressions are Awesome

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$
- Let $L = \{ w \in \Sigma^* \mid |w| = 4 \}$

(0|1)^4

0000
1010
1111
1000

Regular Expressions are Awesome

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$
- Let $L = \{\ w \in \Sigma^* \mid |w| = 4\ \}$

$(0|1)^4$

0000
1010
1111
1000

Regular Expressions are Awesome

- Let $\Sigma = \{\mathtt{0}, \mathtt{1}\}$
- Let $L = \{ w \in \Sigma^* \mid w \text{ contains at most one } \mathtt{0} \}$

Regular Expressions are Awesome

- Let $\Sigma = \{0, 1\}$
- Let $L = \{ w \in \Sigma^* \mid w \text{ contains at most one } 0 \}$

1*(0 | ε)1*

Regular Expressions are Awesome

- Let $\Sigma = \{$**0**, **1**$\}$
- Let $L = \{\ w \in \Sigma^* \mid w$ contains at most one **0** $\}$

1*(0 | ε)1*

Regular Expressions are Awesome

- Let $\Sigma = \{0, 1\}$
- Let $L = \{ w \in \Sigma^* \mid w$ contains at most one 0 $\}$

1*(0 | ε)1*

11110111
111111
0111
0

Regular Expressions are Awesome

- Let $\Sigma = \{0, 1\}$
- Let $L = \{ w \in \Sigma^* \mid w \text{ contains at most one } 0 \}$

$$1^*(0 \mid \varepsilon)1^*$$

11110111
111111
0111
0

Regular Expressions are Awesome

- Let $\Sigma = \{$`0`, `1`$\}$
- Let $L = \{\ w \in \Sigma^* \mid w$ contains at most one `0` $\}$

1*0?1*

11110111
111111
0111
0

Regular Expressions are Awesome

- Let $\Sigma = \{$ `a`, `.`, `@` $\}$, where `a` represents "some letter."
- Regular expression for email addresses:

aa* (.aa*)* @ aa*.aa* (.aa*)*

Regular Expressions are Awesome

- Let Σ = { **a**, **.**, **@** }, where **a** represents "some letter."
- Regular expression for email addresses:

aa* (.aa*)* @ aa*.aa* (.aa*)*

cs103@cs.stanford.edu
first.middle.last@mail.site.org
barack.obama@whitehouse.gov

Regular Expressions are Awesome

- Let Σ = { `a`, `.`, `@` }, where `a` represents "some letter."
- Regular expression for email addresses:

aa* (.aa*)* @ aa*.aa* (.aa*)*

cs103@cs.stanford.edu
first.middle.last@mail.site.org
barack.obama@whitehouse.gov

Regular Expressions are Awesome

- Let Σ = { **a**, **.**, **@** }, where **a** represents "some letter."
- Regular expression for email addresses:

aa* (.aa*)* @ aa*.aa* (.aa*)*

cs103@cs.stanford.edu
first.middle.last@mail.site.org
barack.obama@whitehouse.gov

Regular Expressions are Awesome

- Let Σ = { `a`, `.`, `@` }, where `a` represents "some letter."
- Regular expression for email addresses:

aa* (.aa*)* @ aa*.aa* (.aa*)*

cs103@cs.stanford.edu
first.middle.last@mail.site.org
barack.obama@whitehouse.gov

Regular Expressions are Awesome

- Let Σ = { **a**, **.**, **@** }, where **a** represents "some letter."
- Regular expression for email addresses:

$$a^+ \ (.aa^*)^* \ @ \ aa^*.aa^* \ (.aa^*)^*$$

cs103@cs.stanford.edu
first.middle.last@mail.site.org
barack.obama@whitehouse.gov

Regular Expressions are Awesome

- Let Σ = { `a`, `.`, `@` }, where `a` represents "some letter."
- Regular expression for email addresses:

$$a^+ \quad (.a^+)^* \quad @ \quad a^+.a^+ \quad (.a^+)^*$$

cs103@cs.stanford.edu
first.middle.last@mail.site.org
barack.obama@whitehouse.gov

Regular Expressions are Awesome

- Let Σ = { `a`, `.`, `@` }, where `a` represents "some letter."
- Regular expression for email addresses:

$$a^+ \quad (.a^+)^* \quad @ \quad a^+.a^+ \quad (.a^+)^*$$

cs103@cs.stanford.edu
first.middle.last@mail.site.org
barack.obama@whitehouse.gov

Regular Expressions are Awesome

- Let Σ = { `a`, `.`, `@` }, where `a` represents "some letter."
- Regular expression for email addresses:

$$a^+ \quad (.a^+)^* \quad @ \quad a^+ \quad (.a^+)^+$$

cs103@cs.stanford.edu
first.middle.last@mail.site.org
barack.obama@whitehouse.gov

Regular Expressions are Awesome

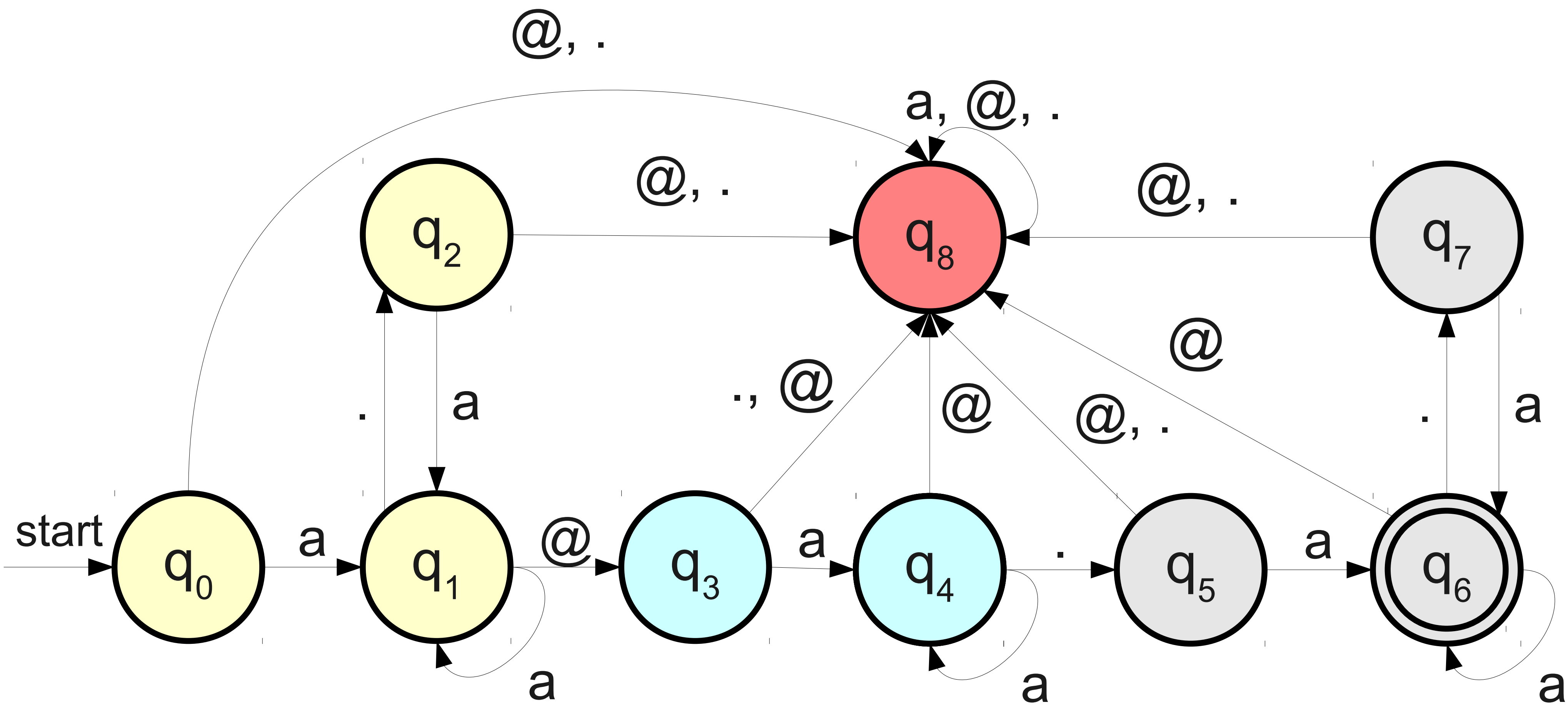
- Let Σ = { **a**, **.**, **@** }, where **a** represents "some letter."
- Regular expression for email addresses:

$$a^+(.a^+)^*@a^+(.a^+)^+$$

cs103@cs.stanford.edu
first.middle.last@mail.site.org
barack.obama@whitehouse.gov

Regular Expressions are Awesome

$$a^{+}(.a^{+})^{*}@a^{+}(.a^{+})^{+}$$



The Power of Regular Expressions

Theorem: If R is a regular expression, then $\mathscr{L}(R)$ is regular.

Proof idea: Induction over the structure of regular expressions. Atomic regular expressions are the base cases, and the inductive step handles each way of combining regular expressions.

Sketch of proof at the appendix of these slides.

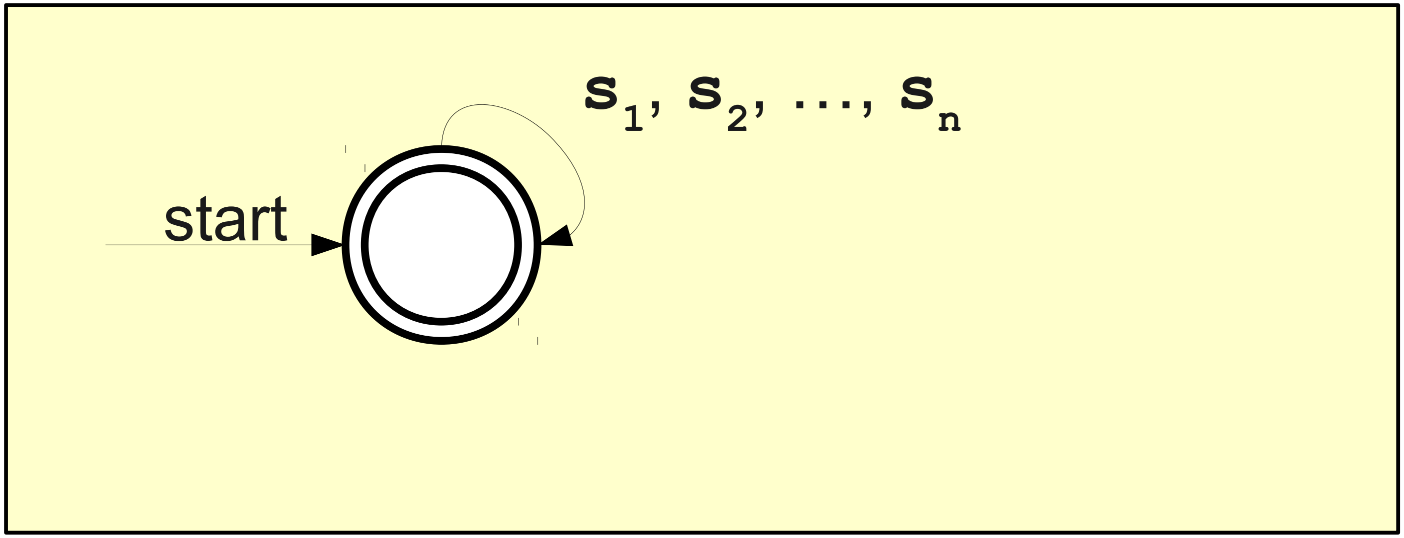
The Power of Regular Expressions

Theorem: If L is a regular language, then there is a regular expression for L.

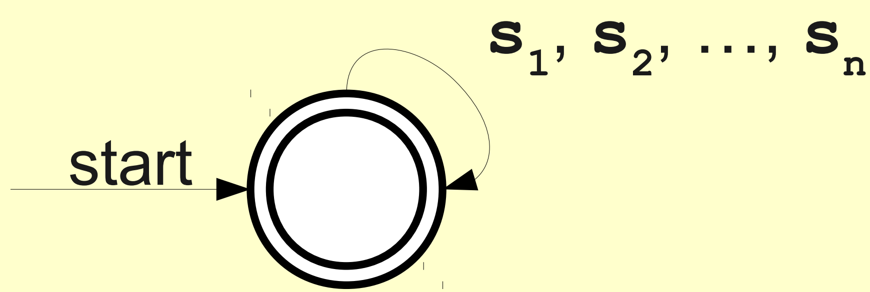
This is not obvious!

Proof idea: Show how to convert an arbitrary NFA into a regular expression.

From NFAs to Regular Expressions

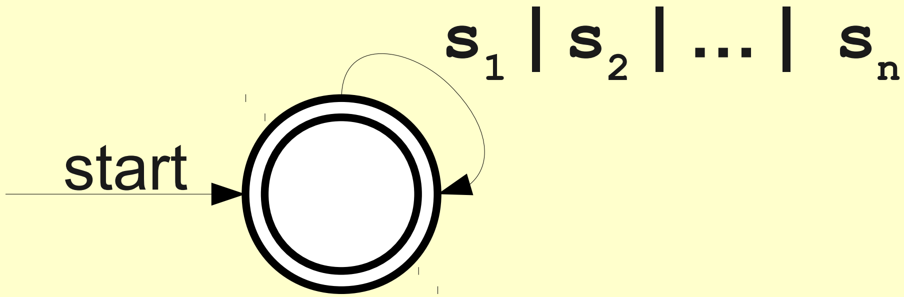


From NFAs to Regular Expressions



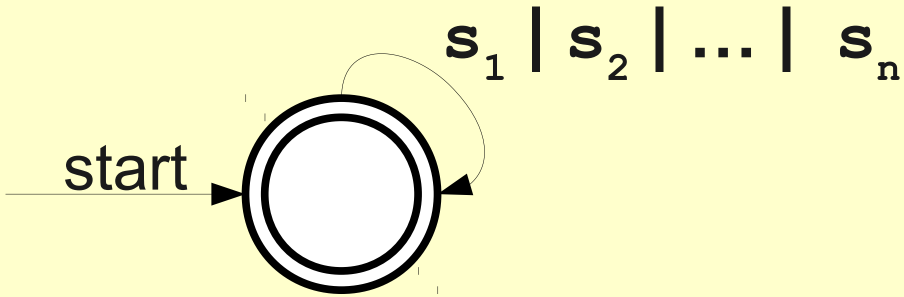
Regular expression: $(s_1 \mid s_2 \mid \ldots \mid s_n)^*$

From NFAs to Regular Expressions



Regular expression: $(s_1 \mid s_2 \mid \ldots \mid s_n)^*$

From NFAs to Regular Expressions



Regular expression: $(s_1 \mid s_2 \mid \ldots \mid s_n)^*$

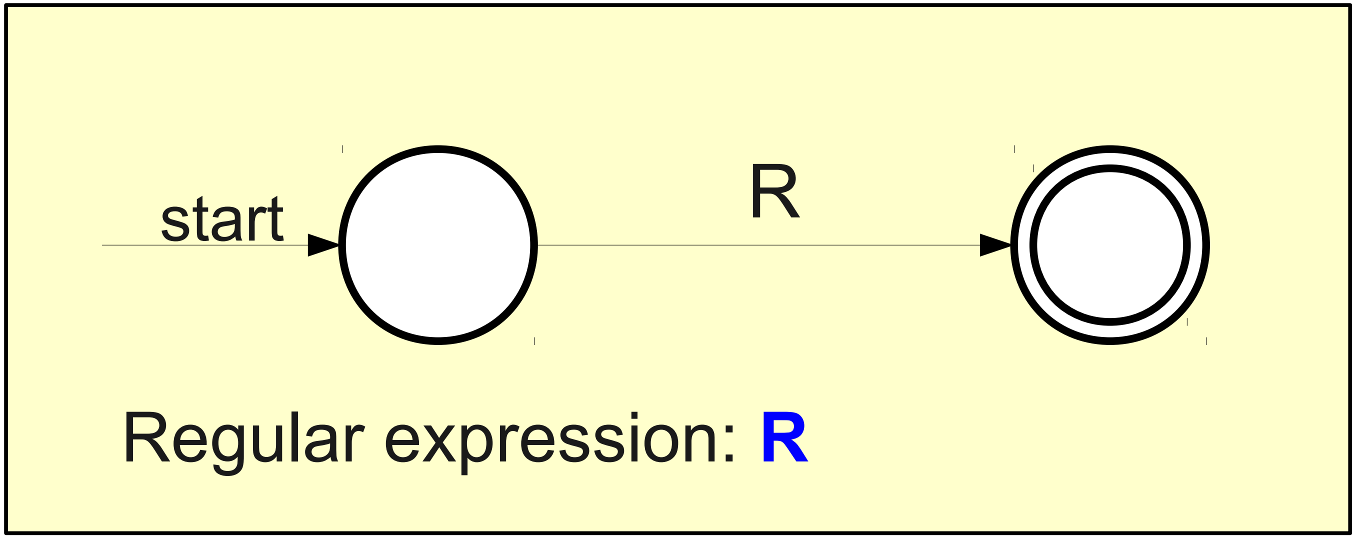
Key idea: Label transitions with arbitrary regular expressions.

From NFAs to Regular Expressions

From NFAs to Regular Expressions

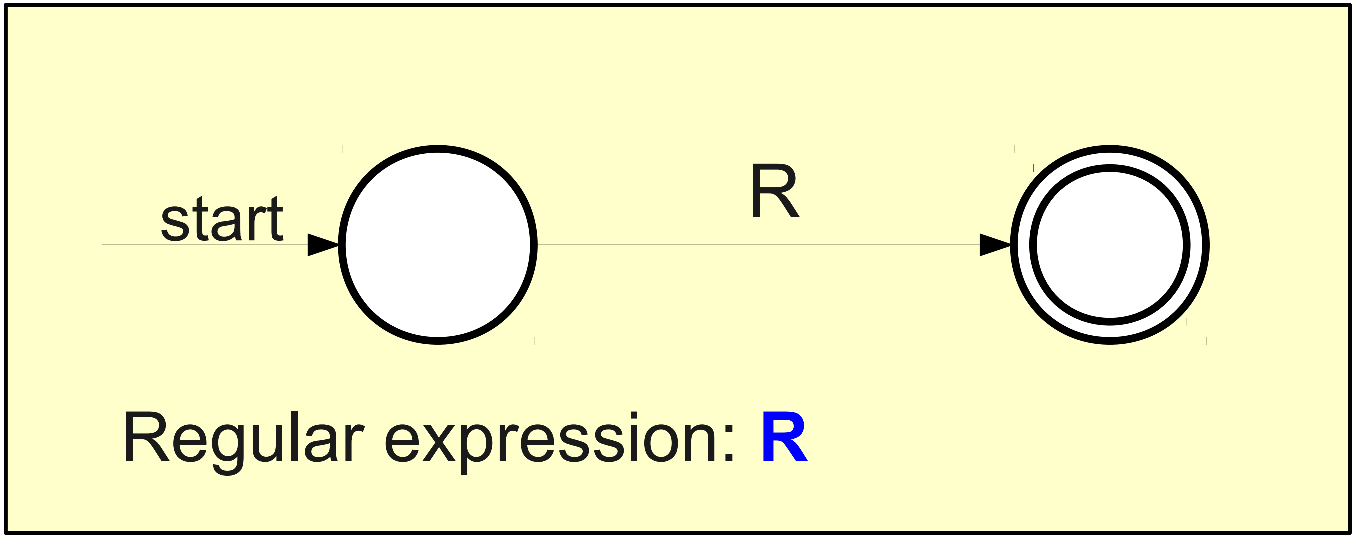
Regular expression: **R**

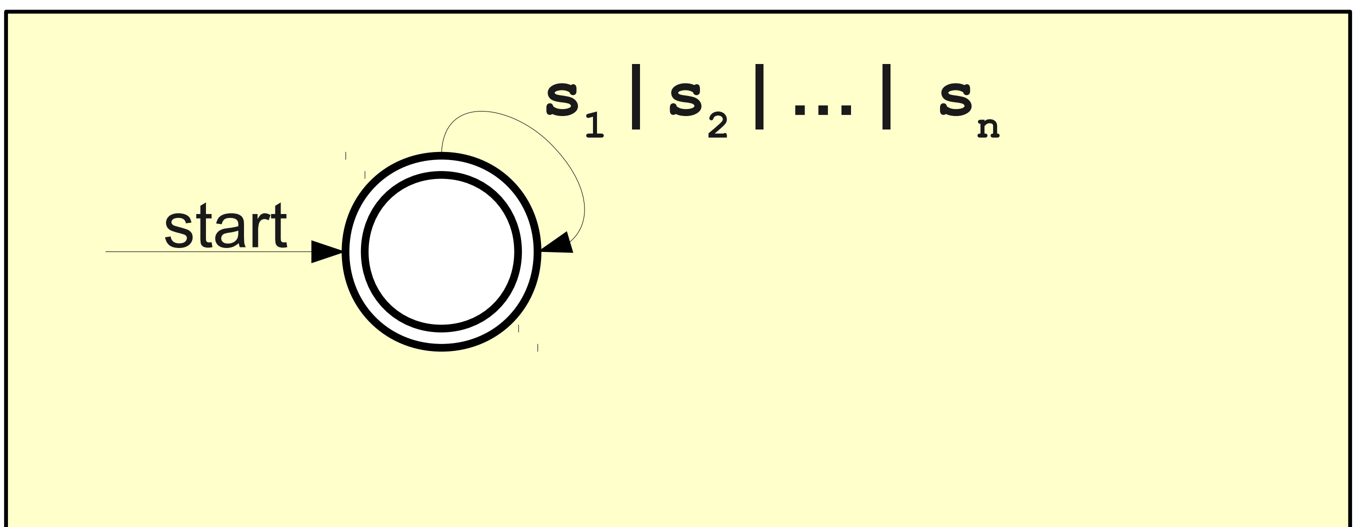
From NFAs to Regular Expressions



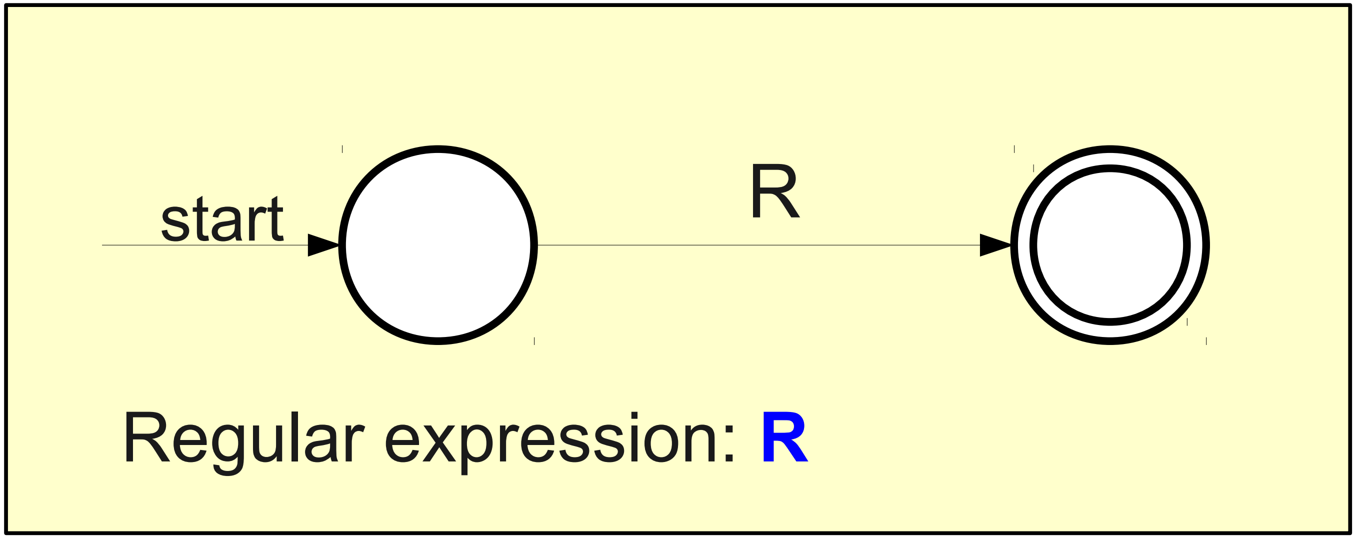
Key idea: If we can convert any NFA into something that looks like this, we can easily read off the regular expression.

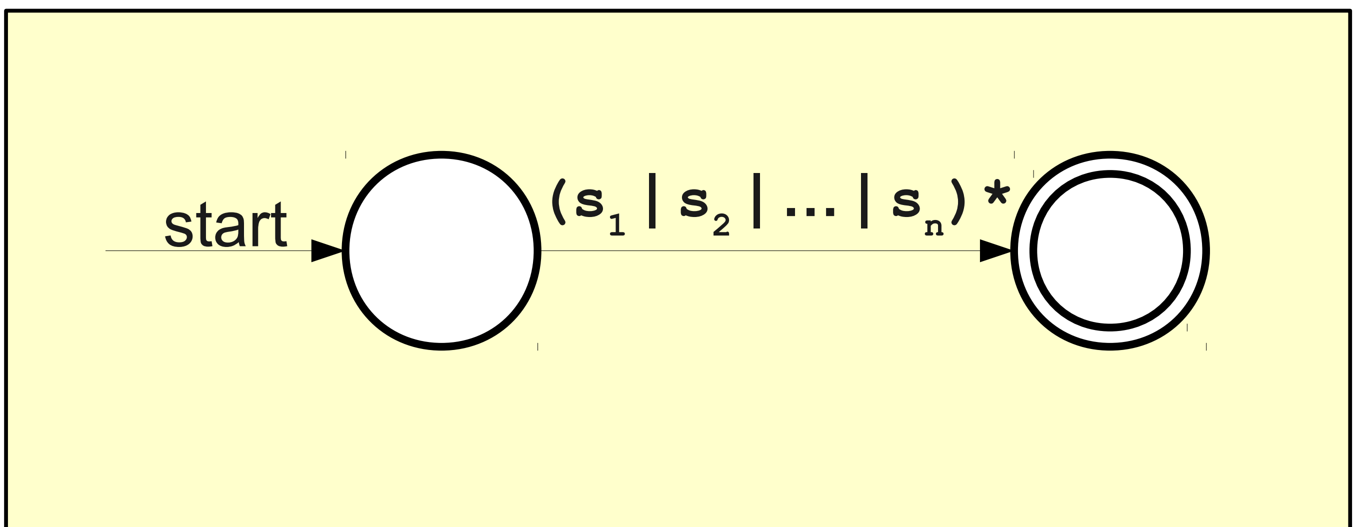
From NFAs to Regular Expressions



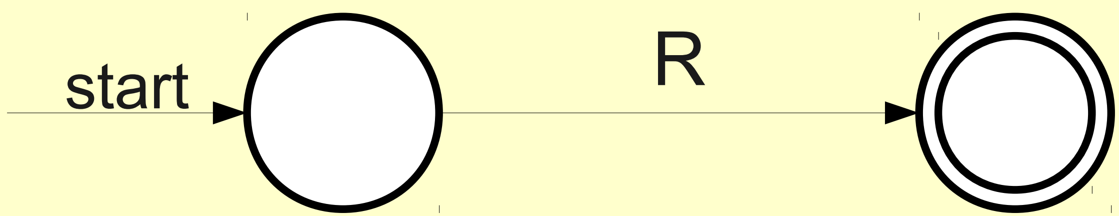


From NFAs to Regular Expressions

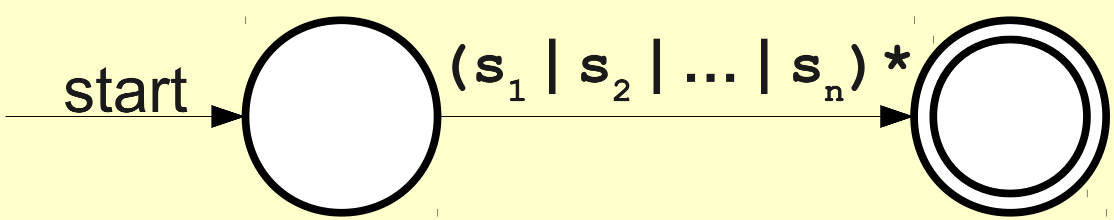




From NFAs to Regular Expressions

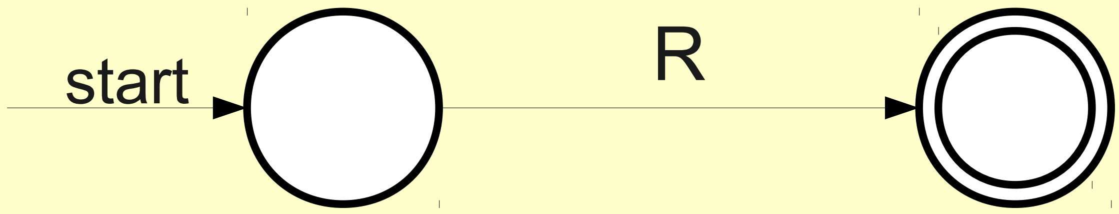


Regular expression: **R**



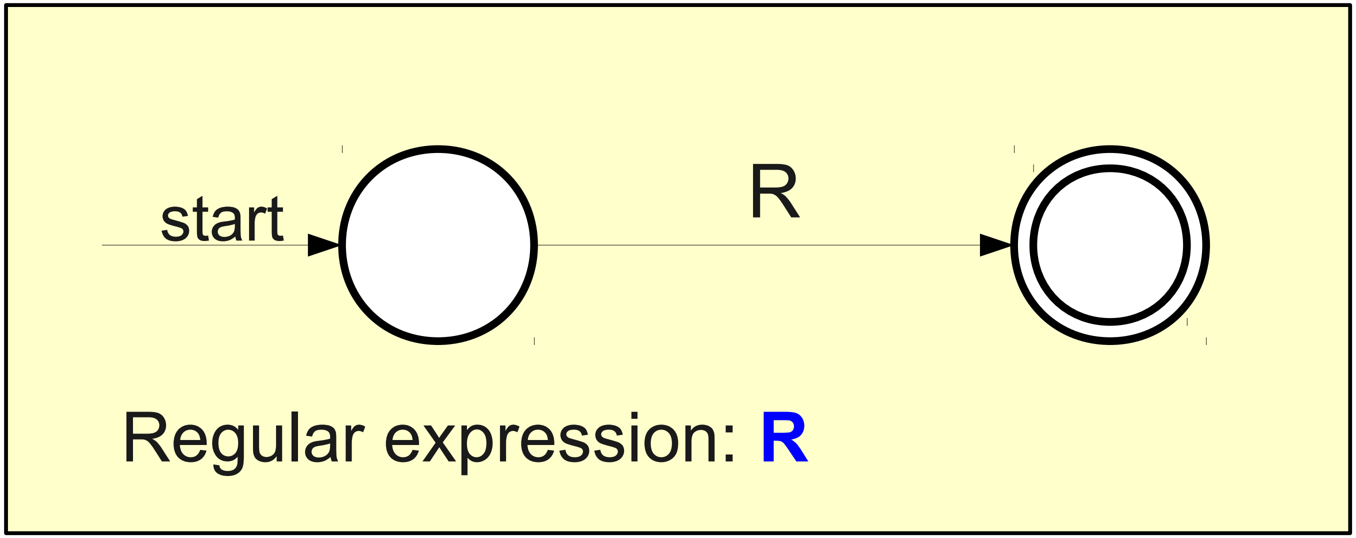
Regular expression: $(s_1 \mid s_2 \mid \ldots \mid s_n)^*$

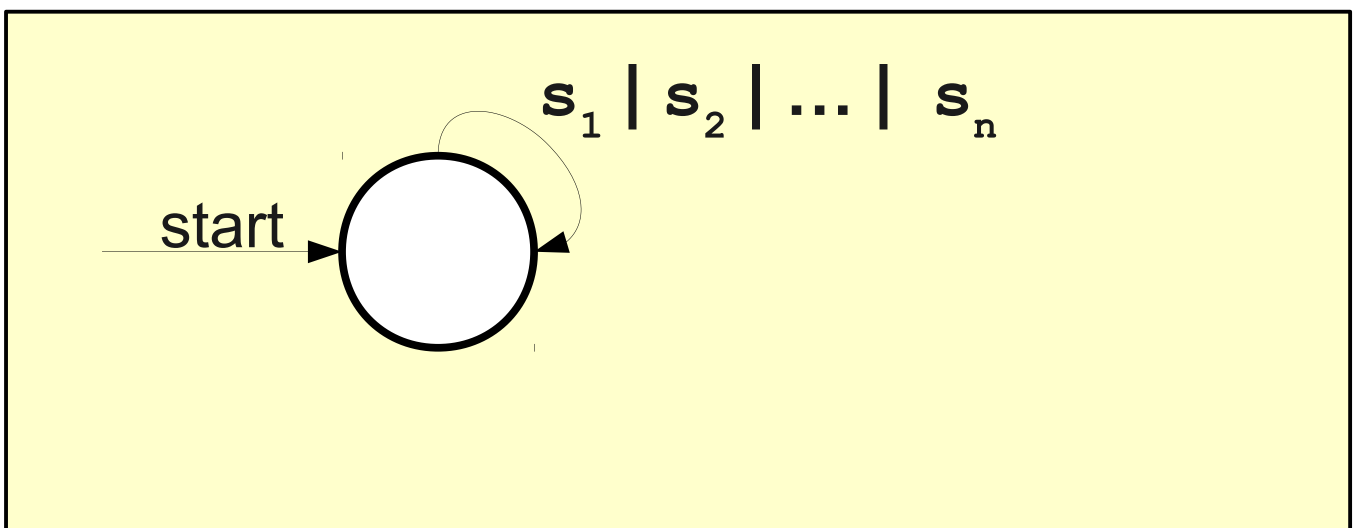
From NFAs to Regular Expressions



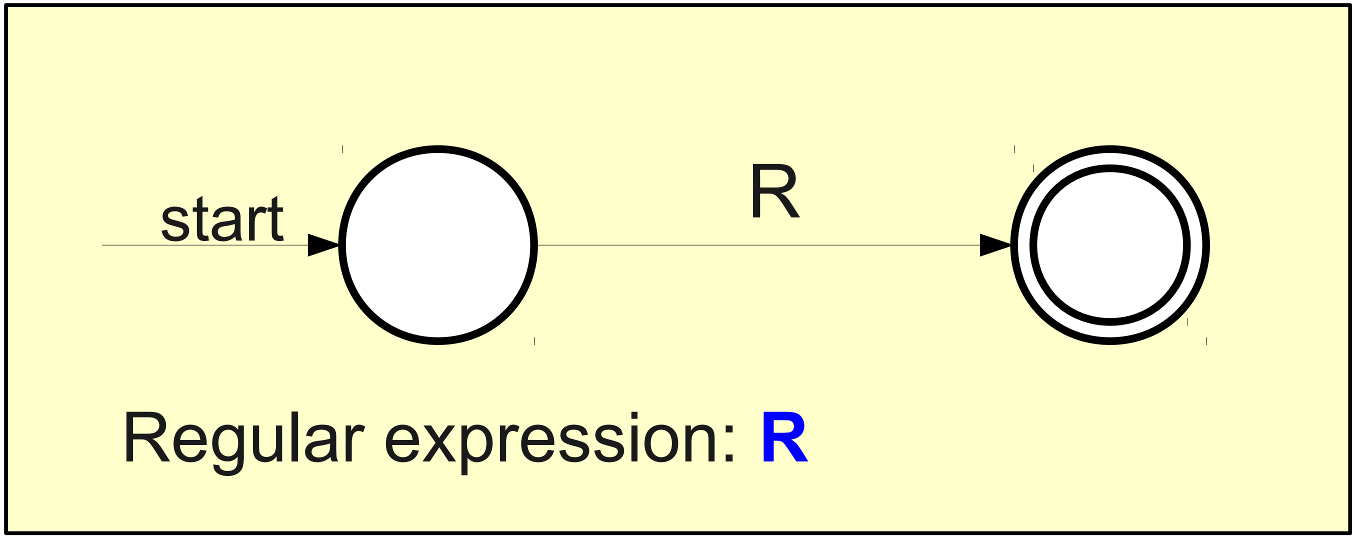
Regular expression: **R**

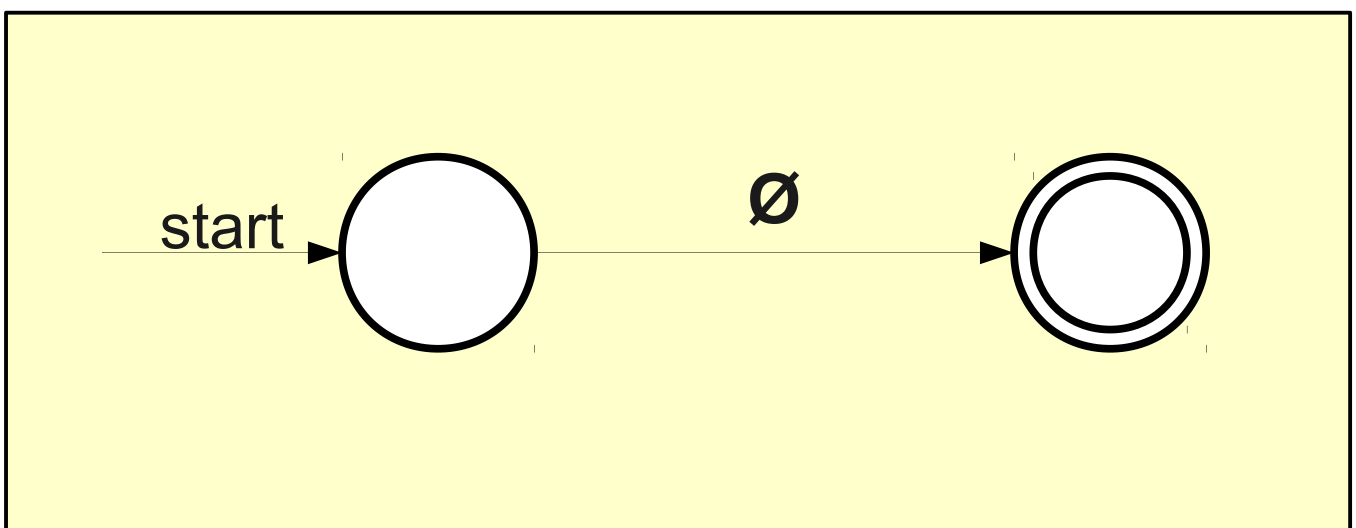
From NFAs to Regular Expressions



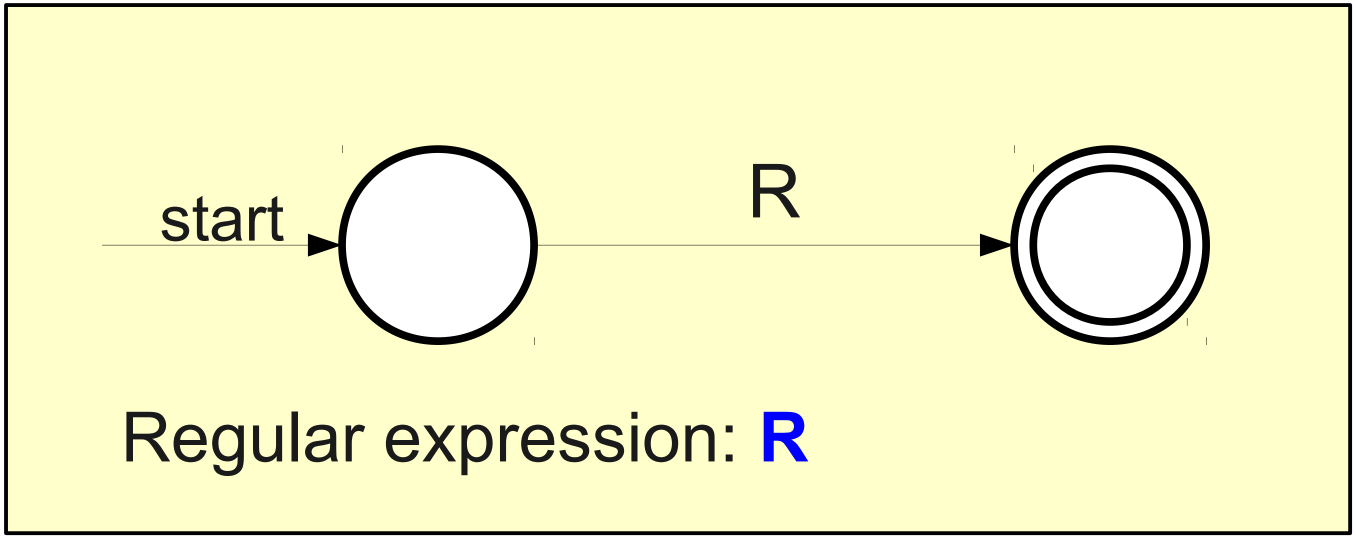


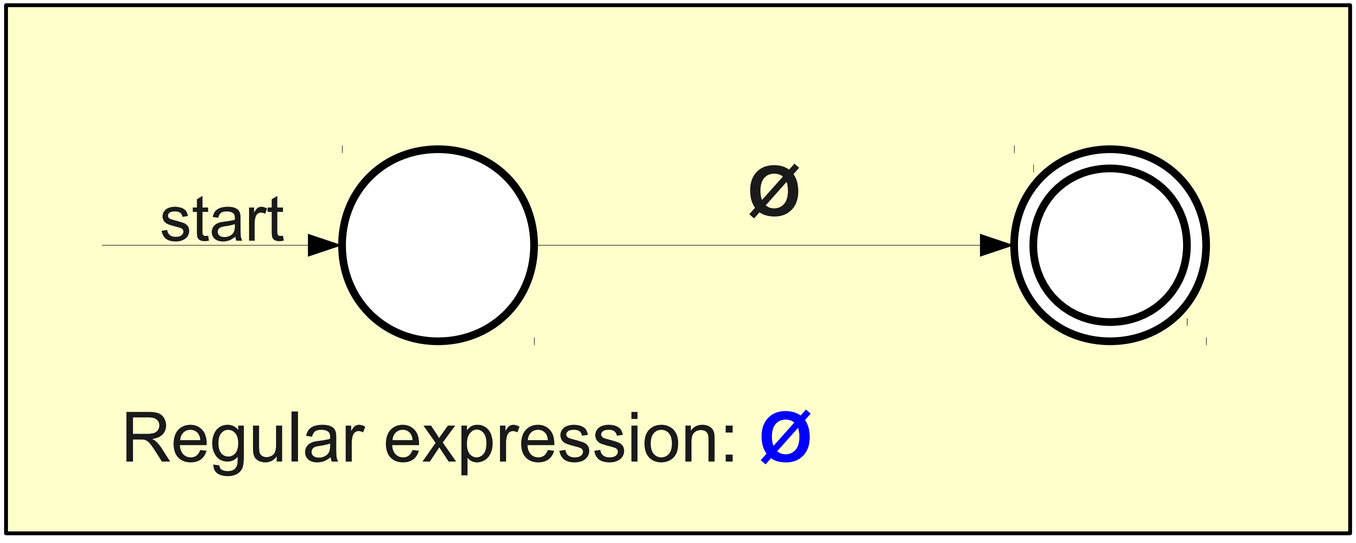
From NFAs to Regular Expressions





From NFAs to Regular Expressions

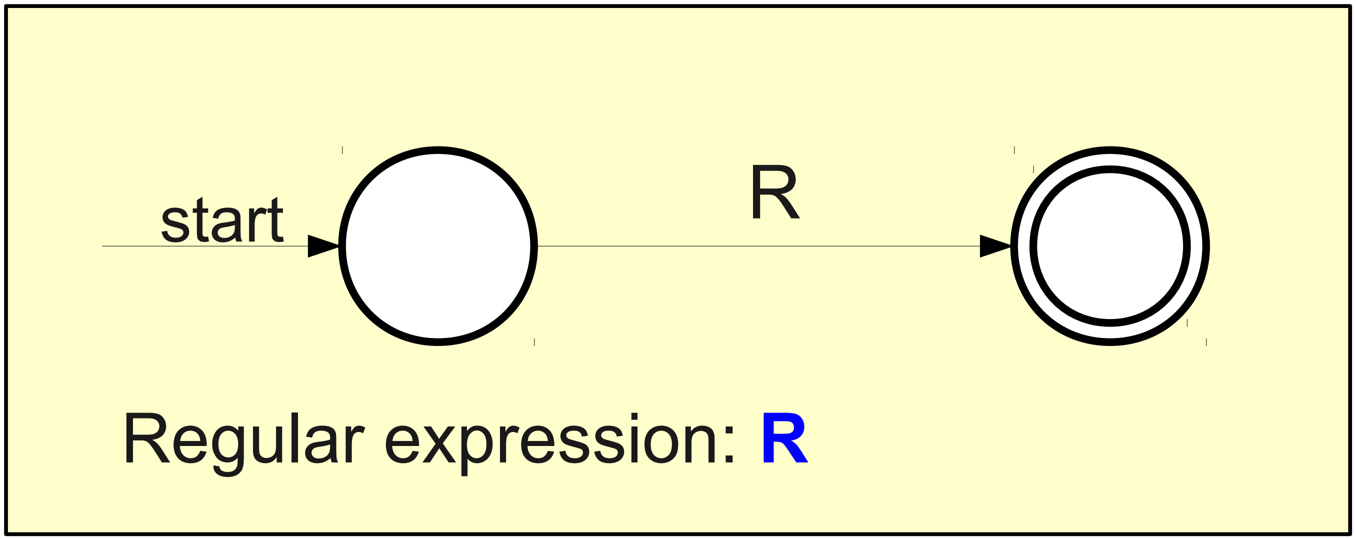


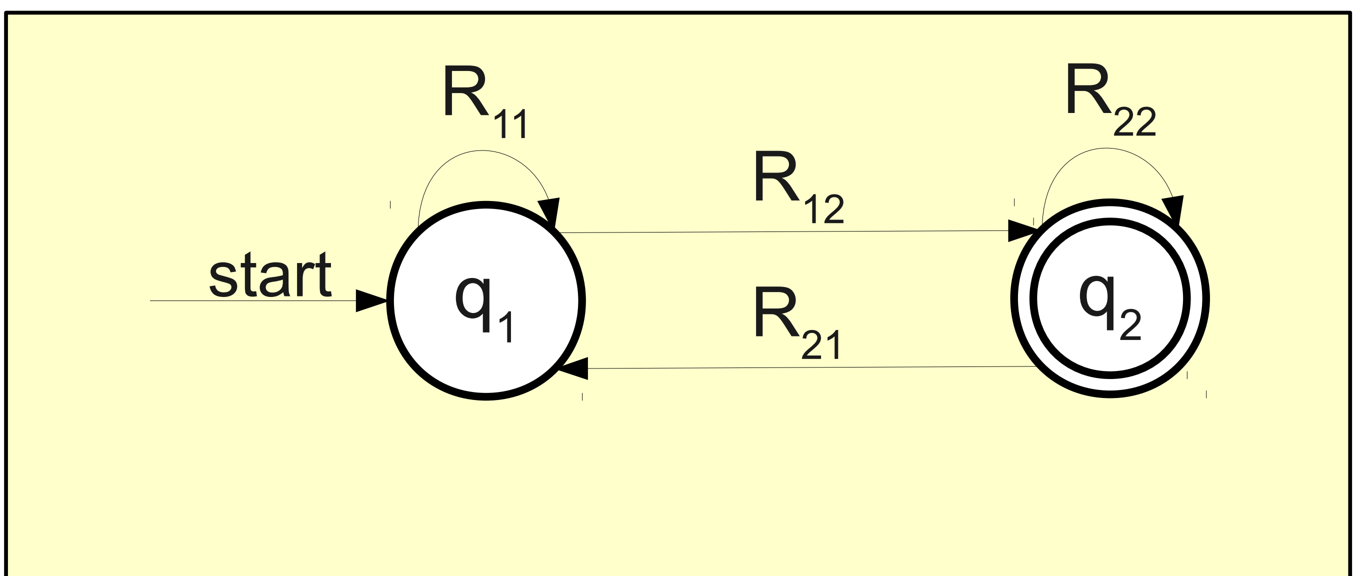


From NFAs to Regular Expressions

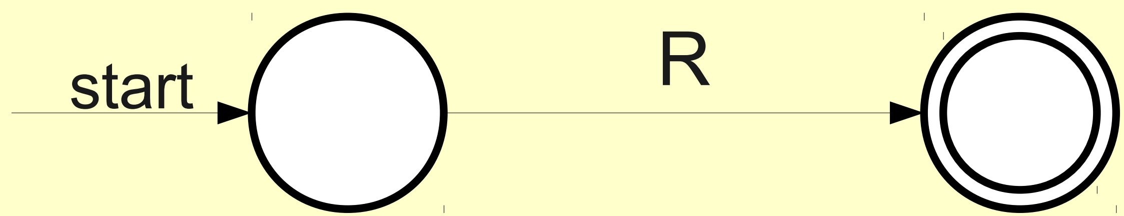
Regular expression: **R**

From NFAs to Regular Expressions



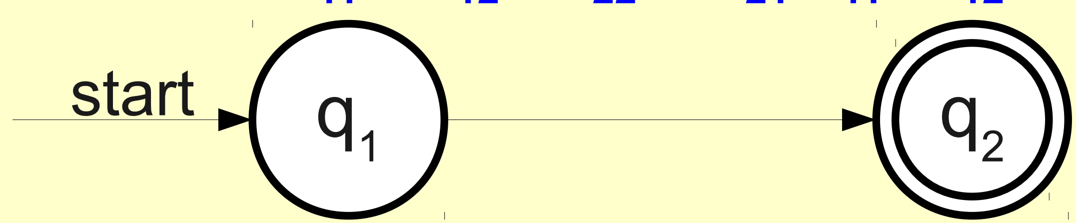


From NFAs to Regular Expressions

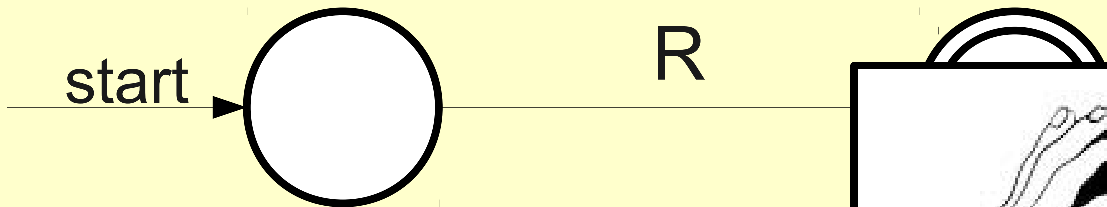


Regular expression: **R**

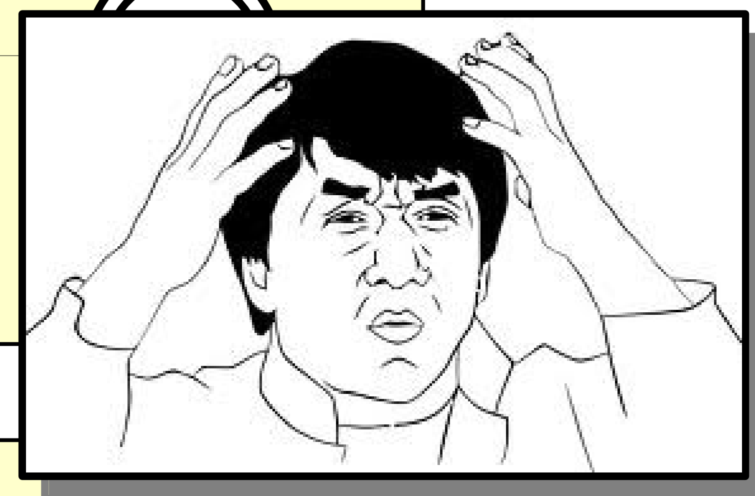
$\mathbf{R_{11}^{*}\ R_{12}\ (R_{22}\ |\ R_{21}R_{11}^{*}R_{12})^{*}}$



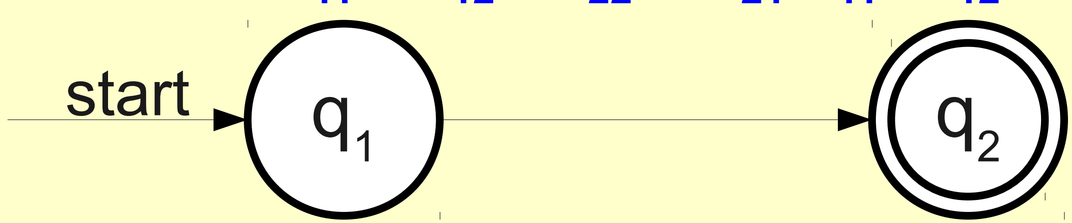
From NFAs to Regular Expressions



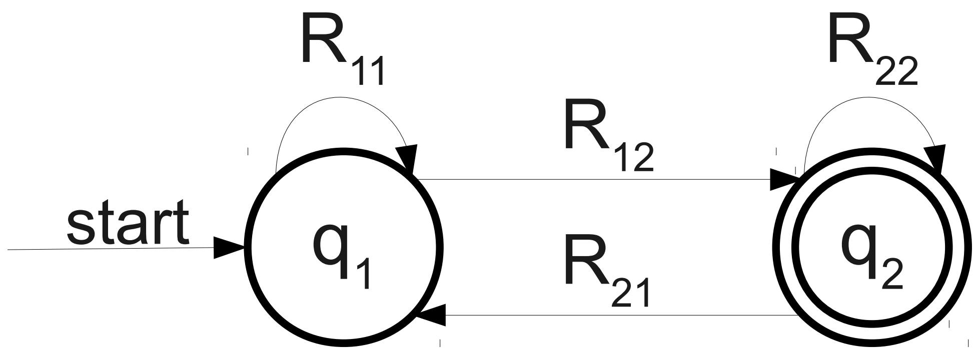
Regular expression: **R**



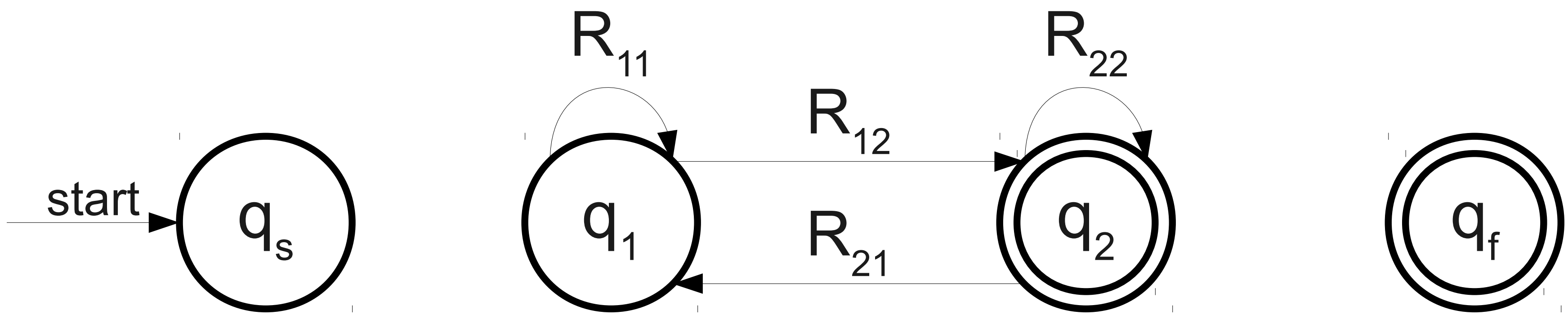
$$R_{11}^* \, R_{12} \, (R_{22} \mid R_{21}R_{11}^*R_{12})^*$$



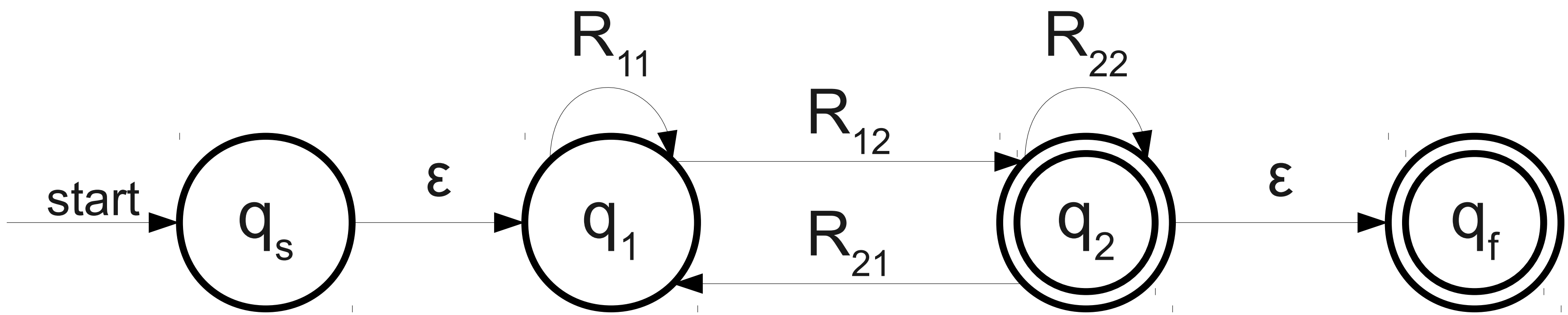
From NFAs to Regular Expressions



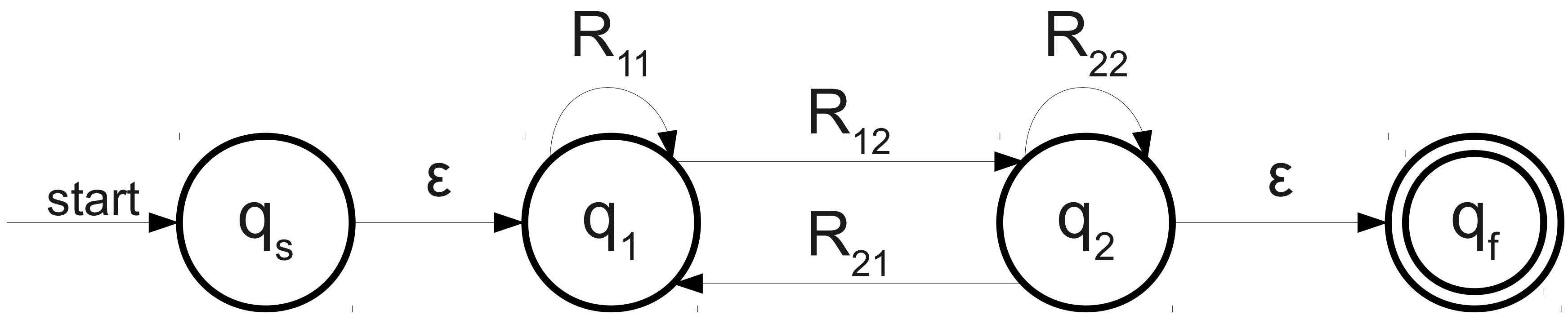
From NFAs to Regular Expressions



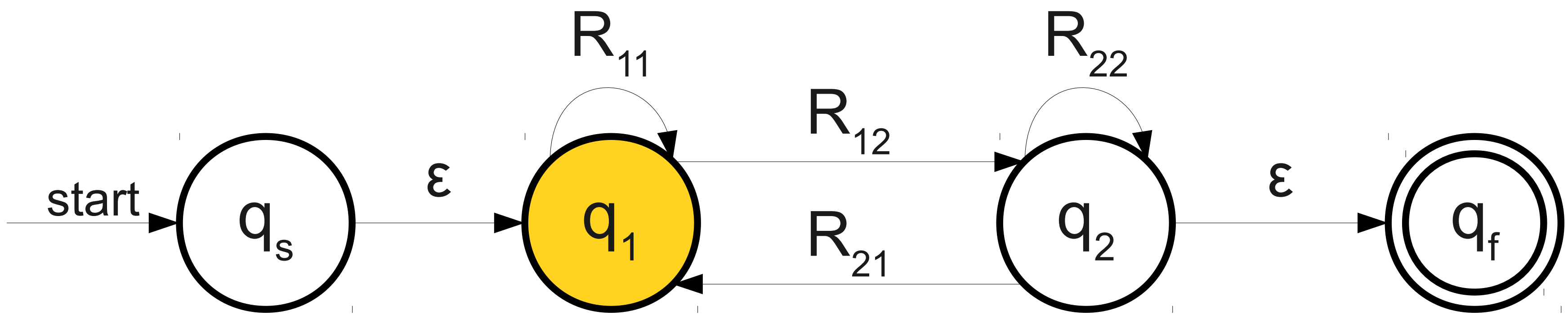
From NFAs to Regular Expressions



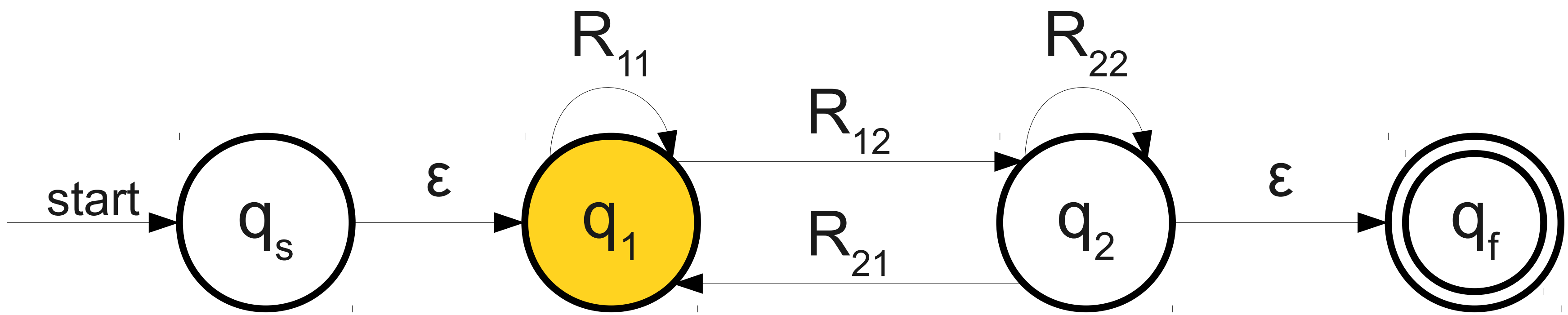
From NFAs to Regular Expressions



From NFAs to Regular Expressions

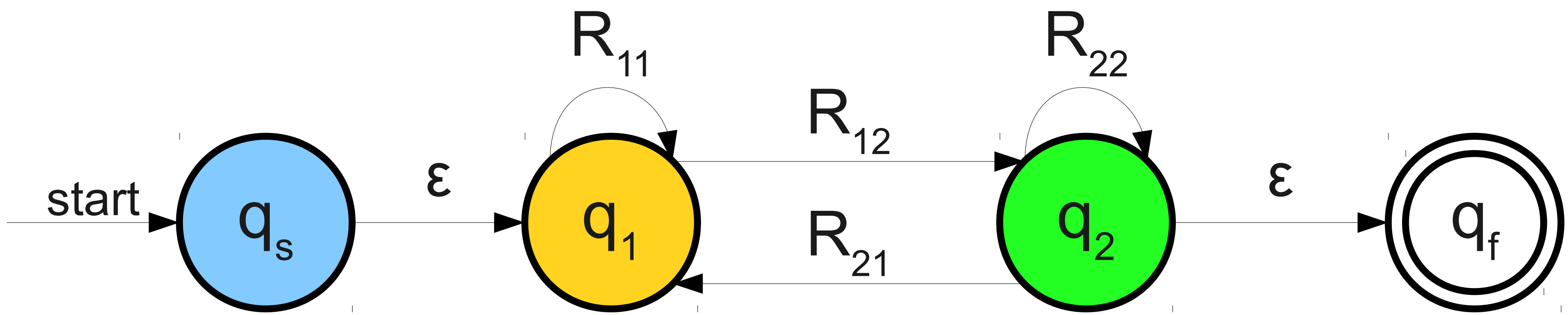


From NFAs to Regular Expressions

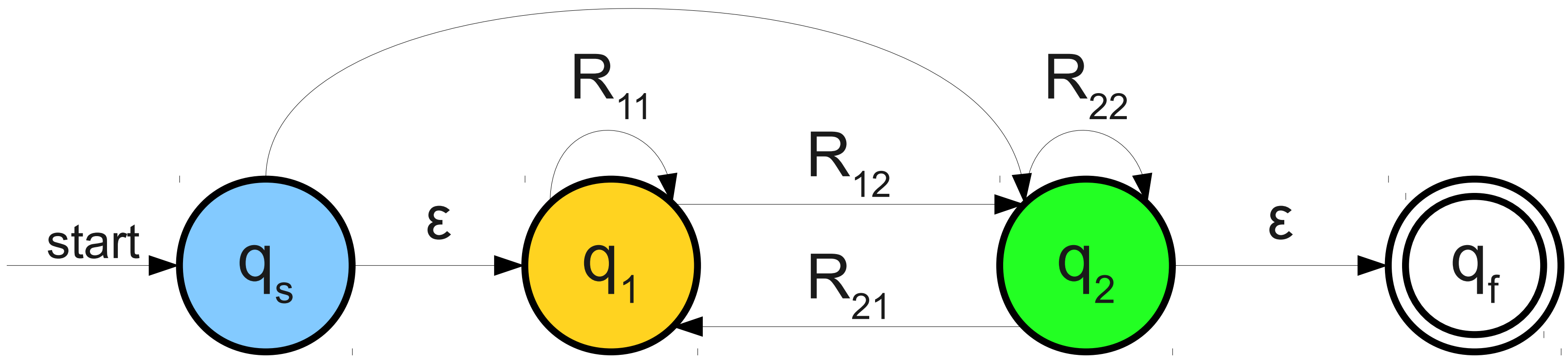


Could we eliminate this state from the NFA?

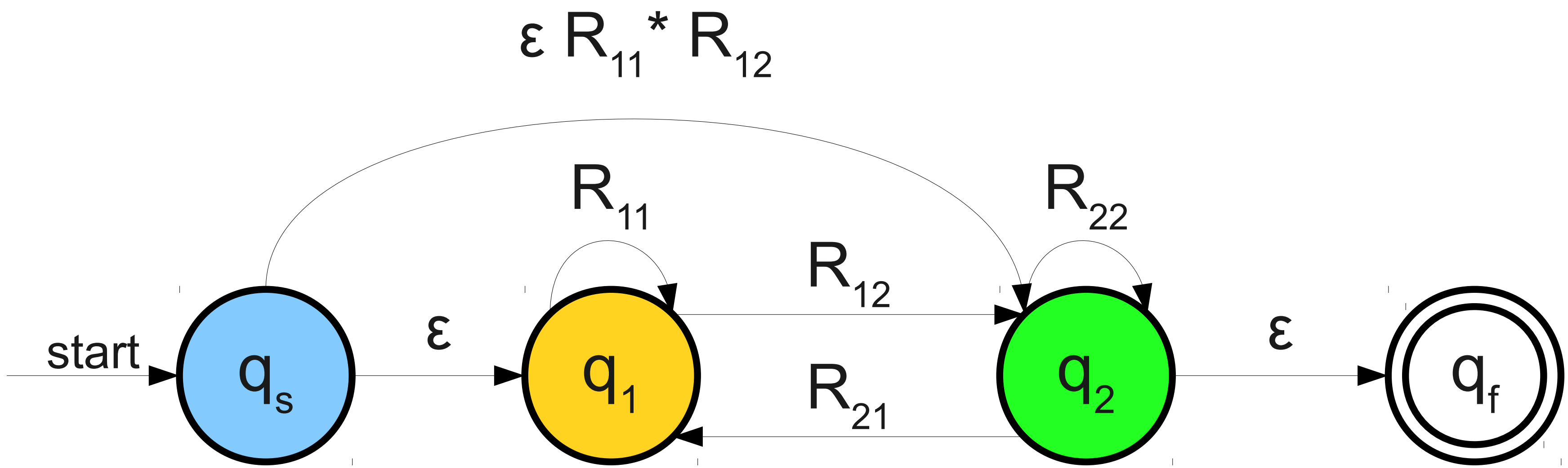
From NFAs to Regular Expressions



From NFAs to Regular Expressions

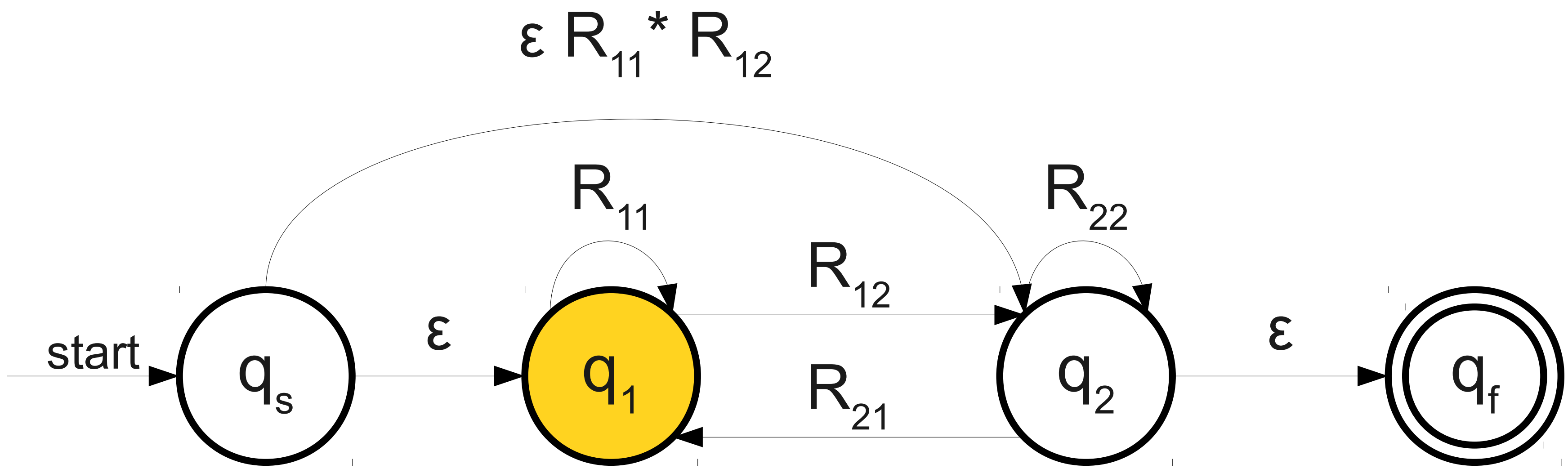


From NFAs to Regular Expressions

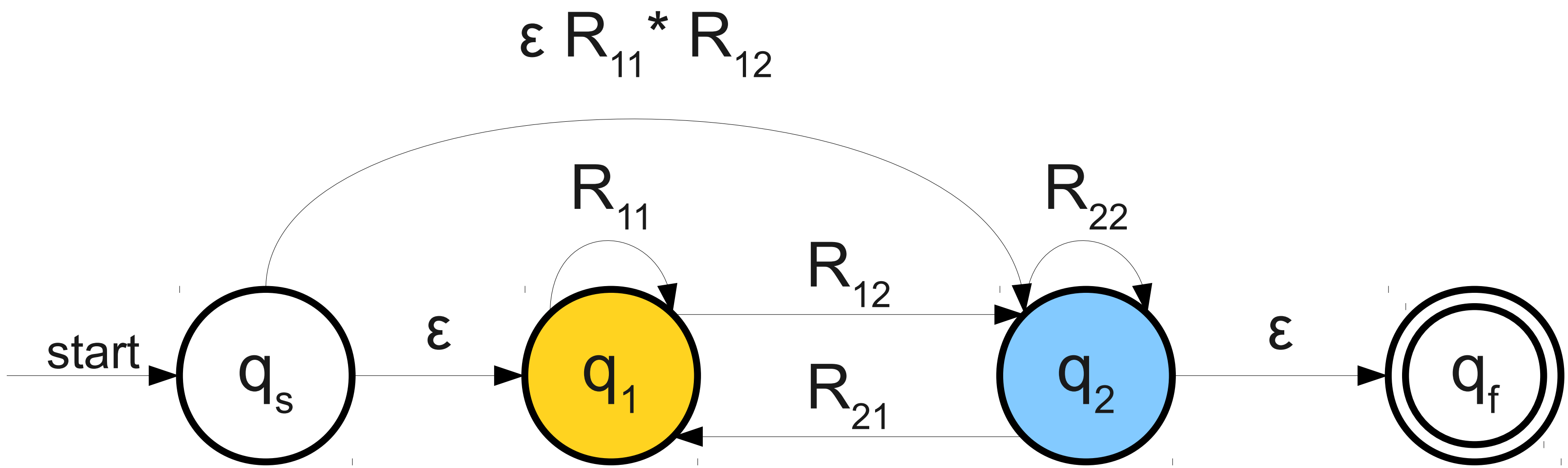


Note: We're using concatenation and Kleene closure in order to skip this state.

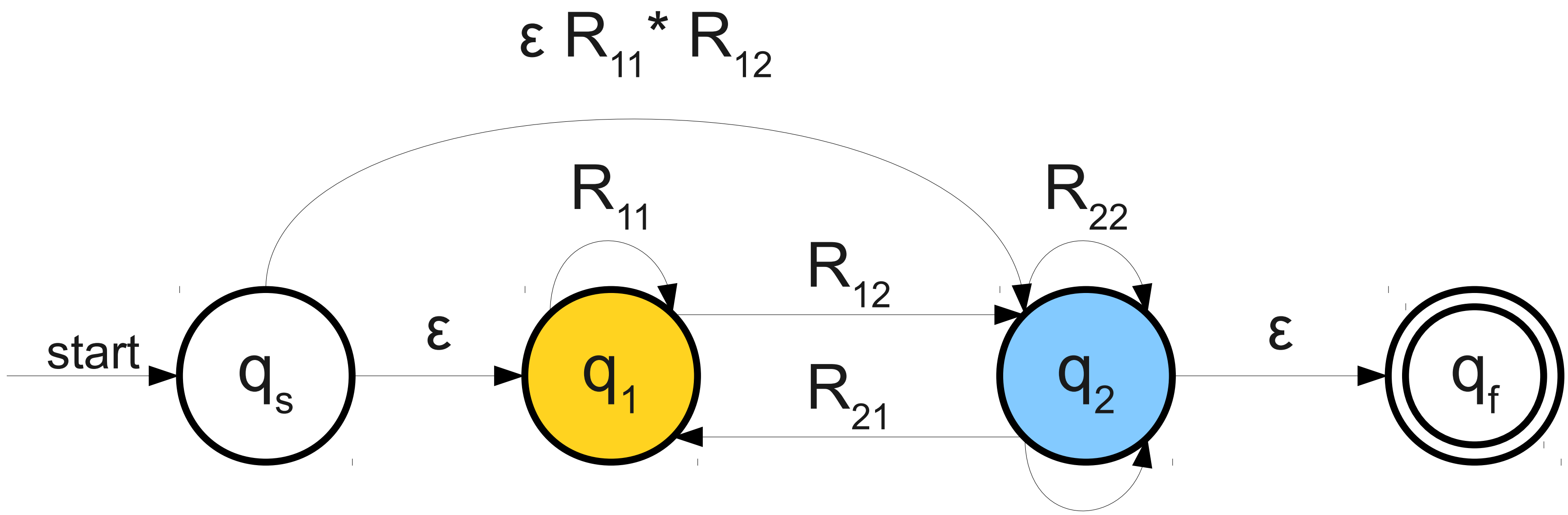
From NFAs to Regular Expressions



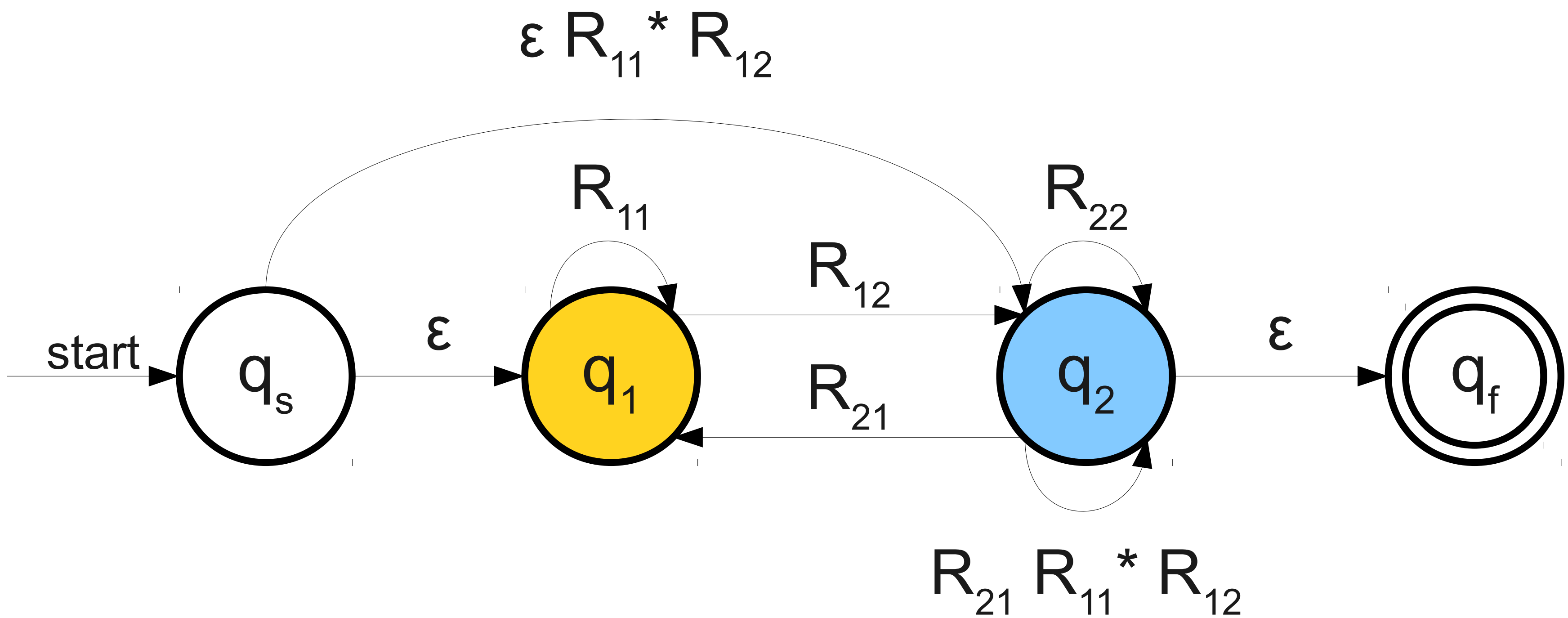
From NFAs to Regular Expressions



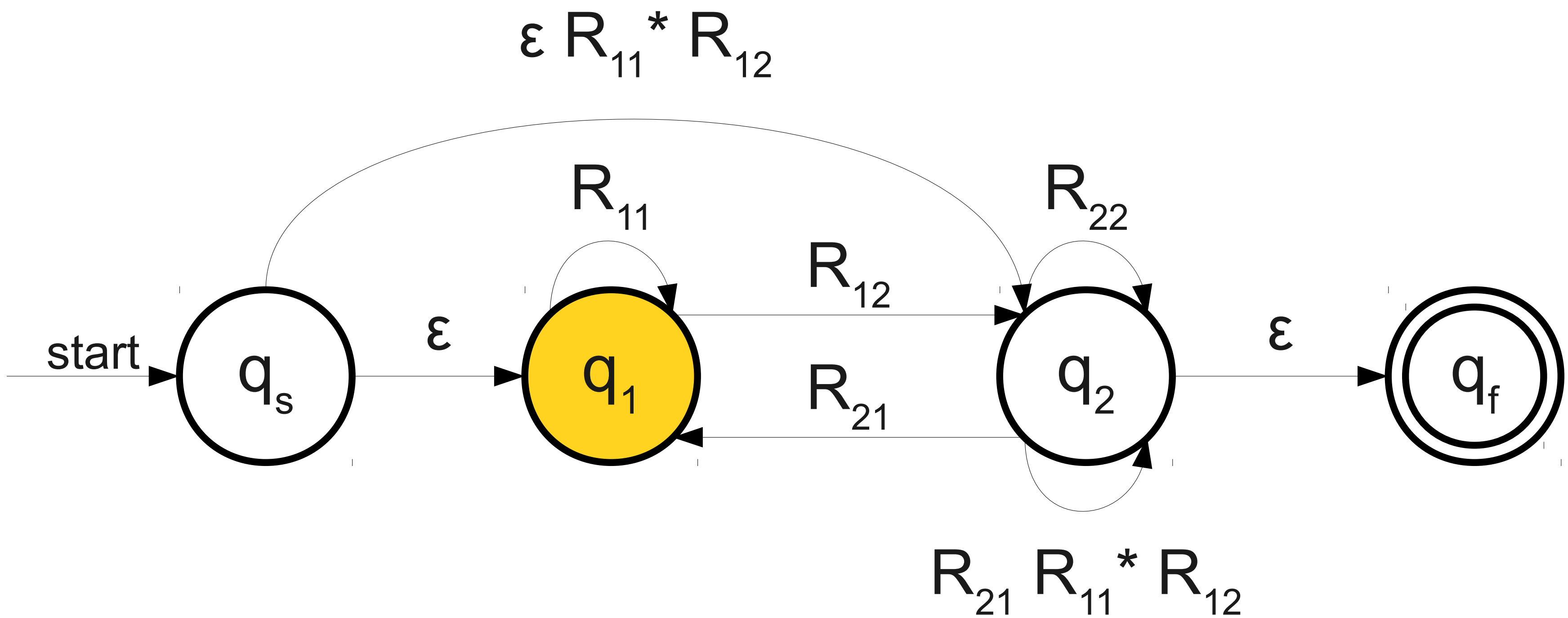
From NFAs to Regular Expressions



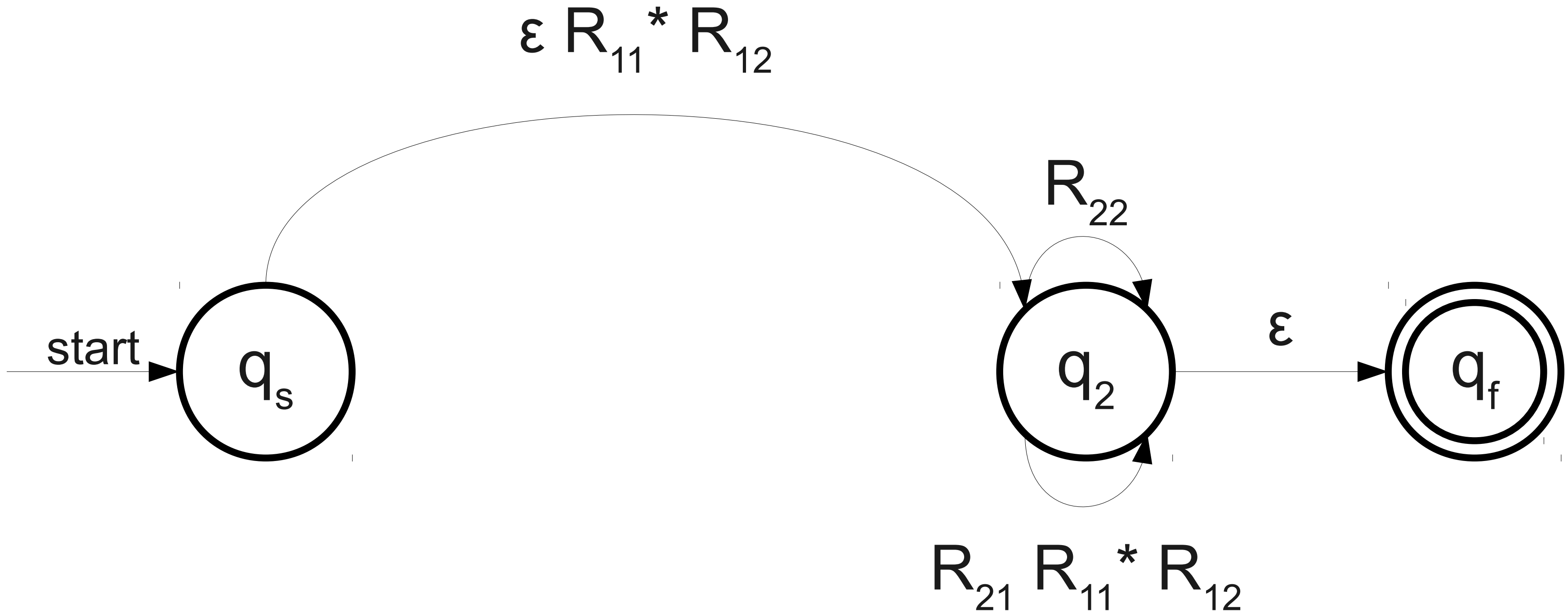
From NFAs to Regular Expressions



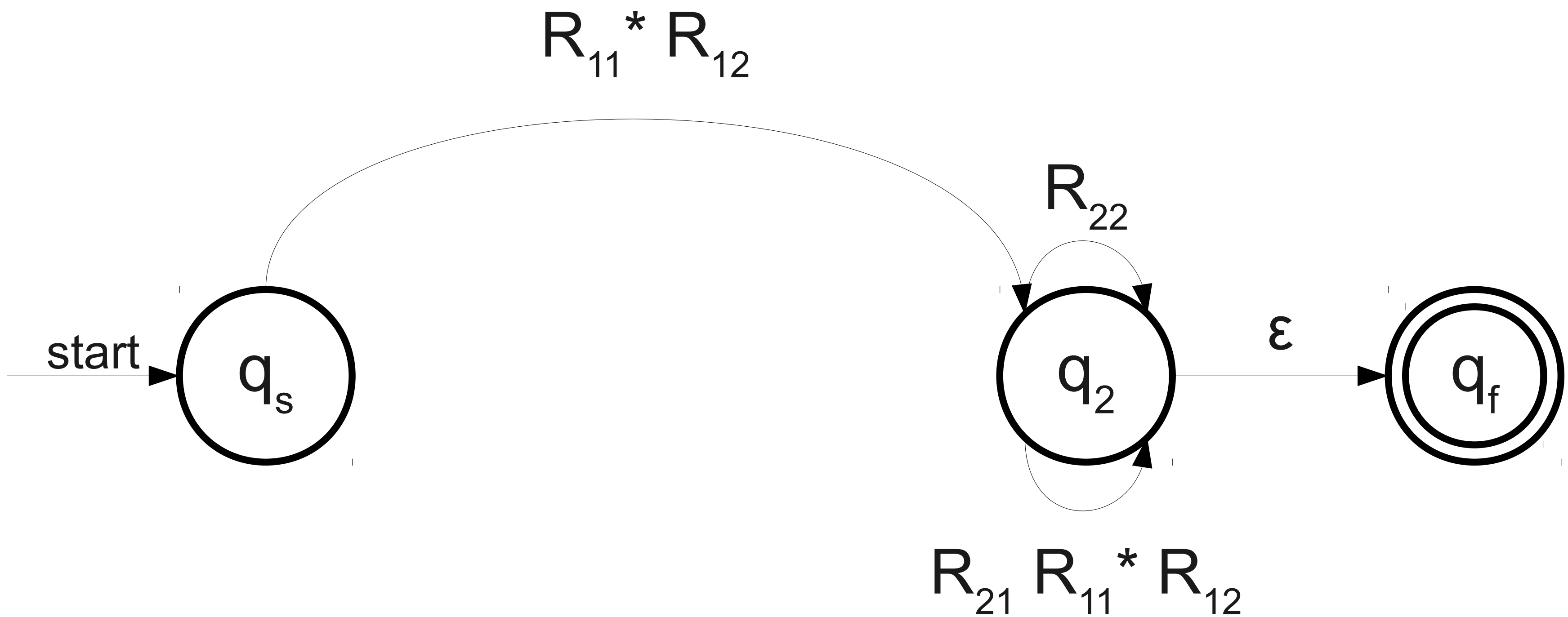
From NFAs to Regular Expressions



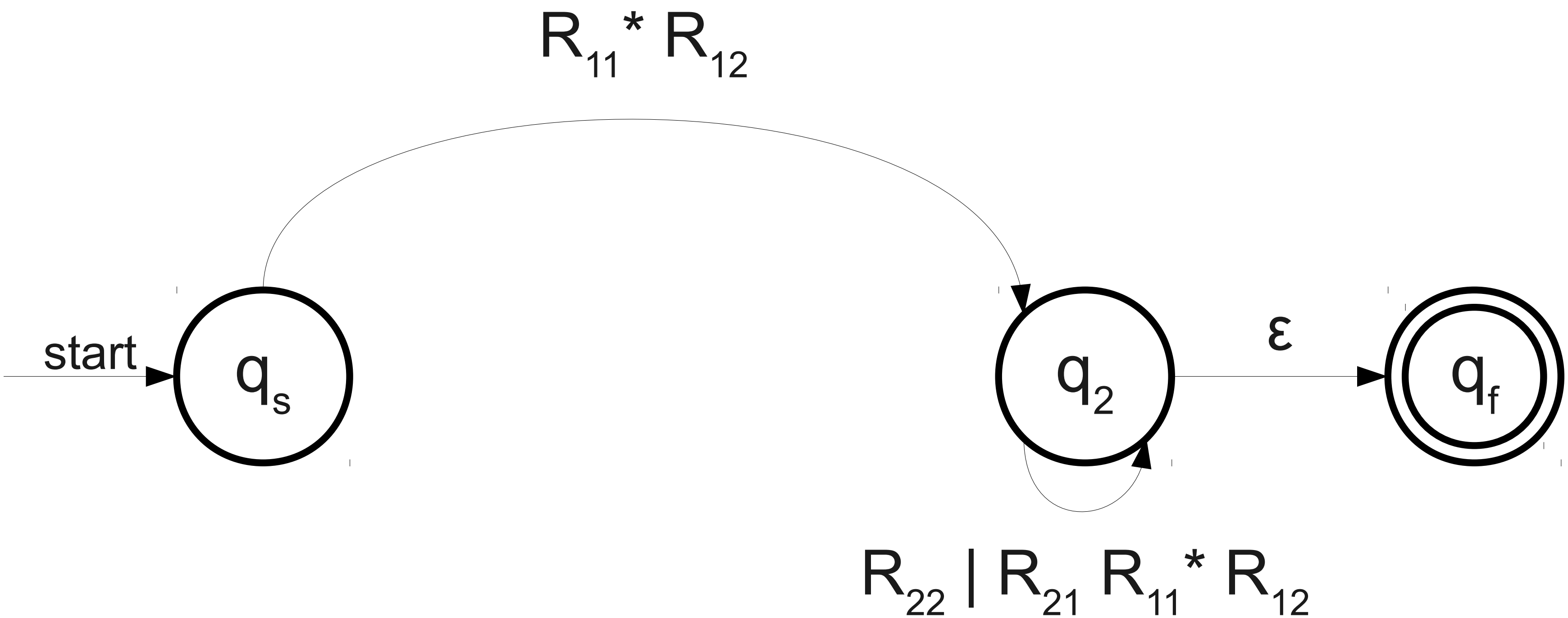
From NFAs to Regular Expressions



From NFAs to Regular Expressions

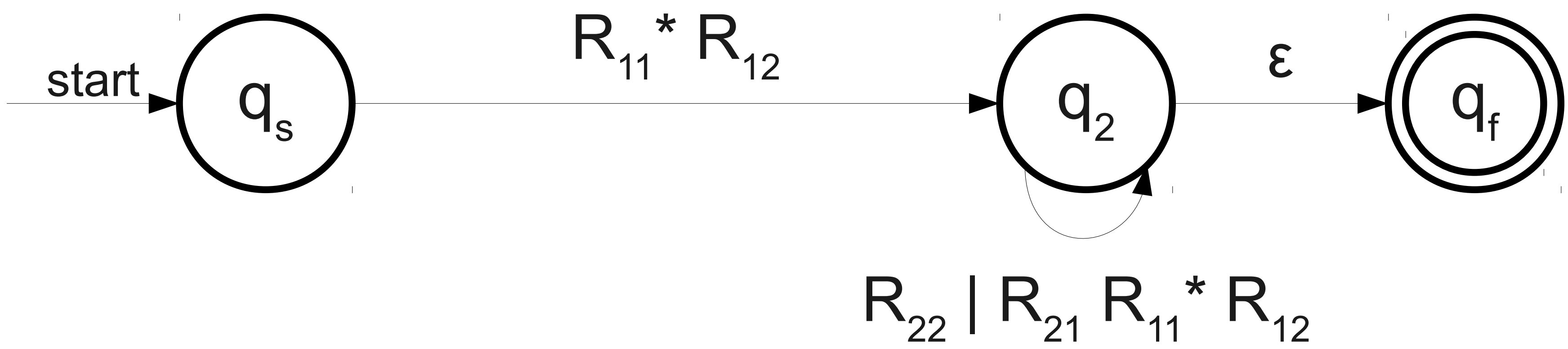


From NFAs to Regular Expressions

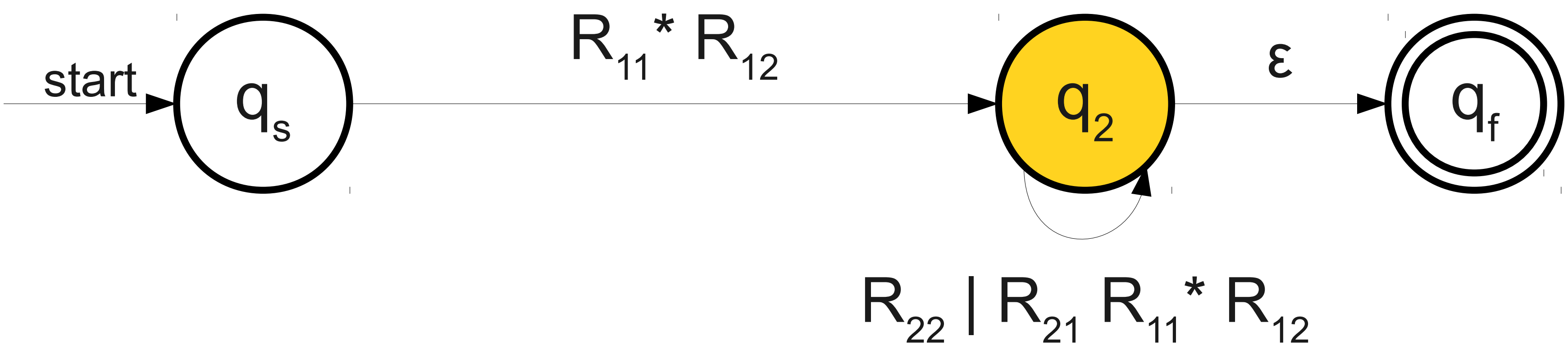


Note: We're using union to combine these transitions together.

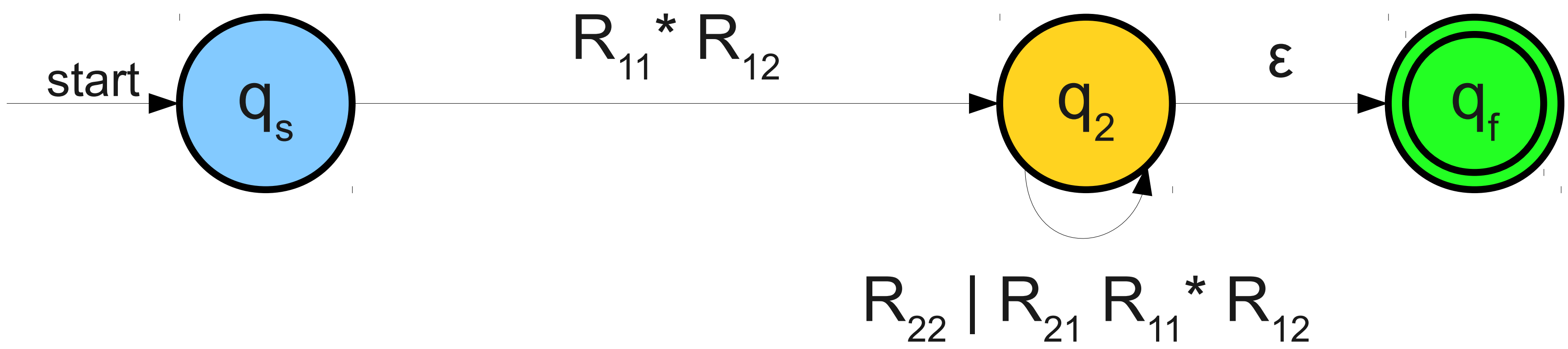
From NFAs to Regular Expressions



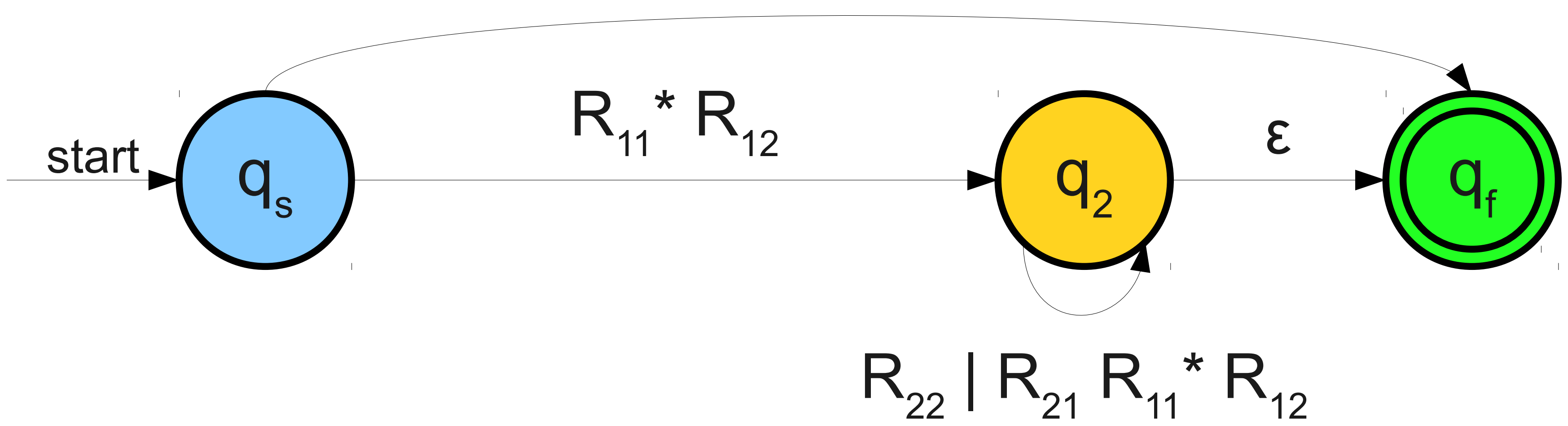
From NFAs to Regular Expressions



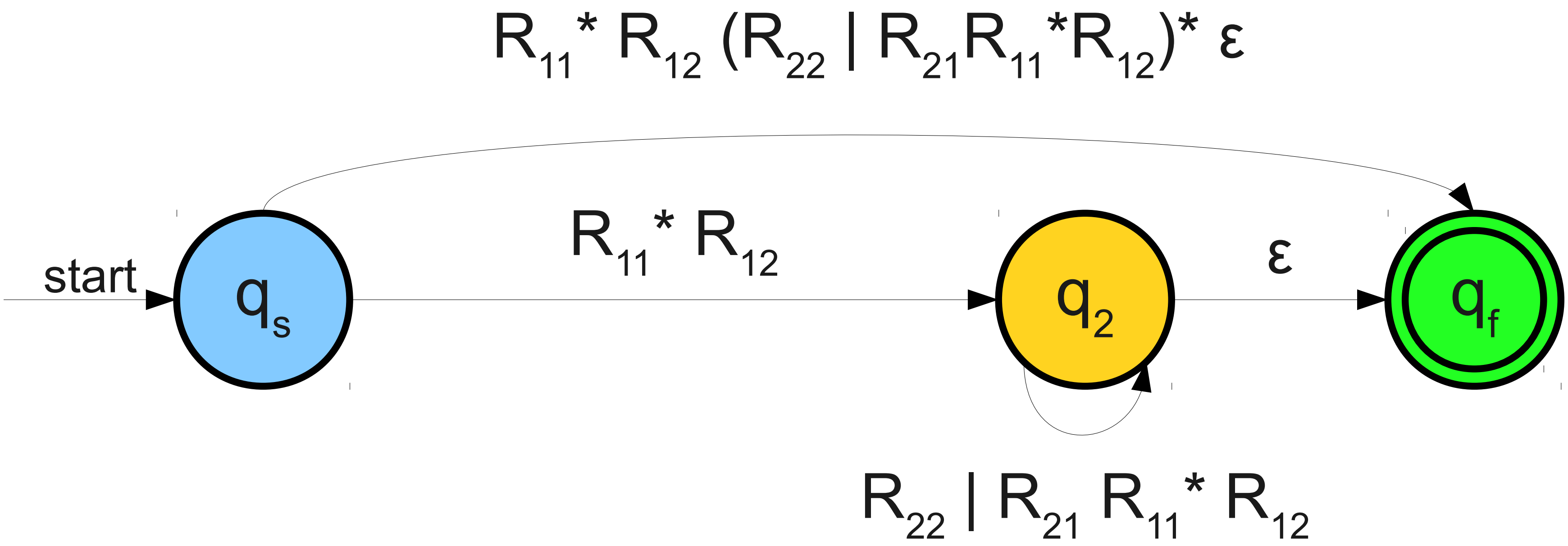
From NFAs to Regular Expressions



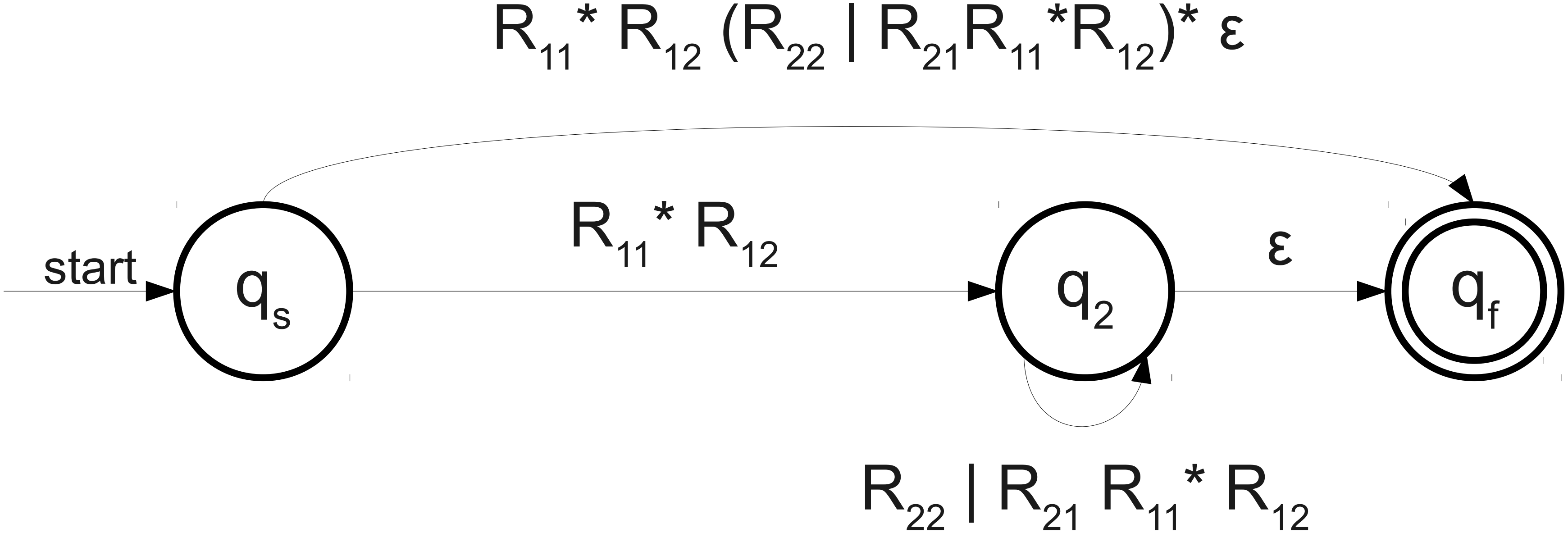
From NFAs to Regular Expressions



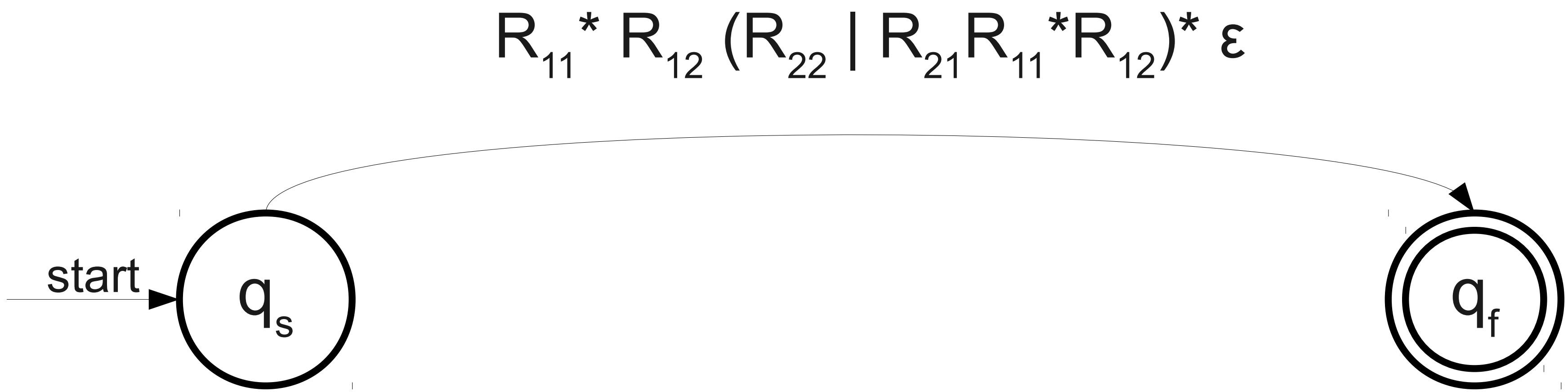
From NFAs to Regular Expressions



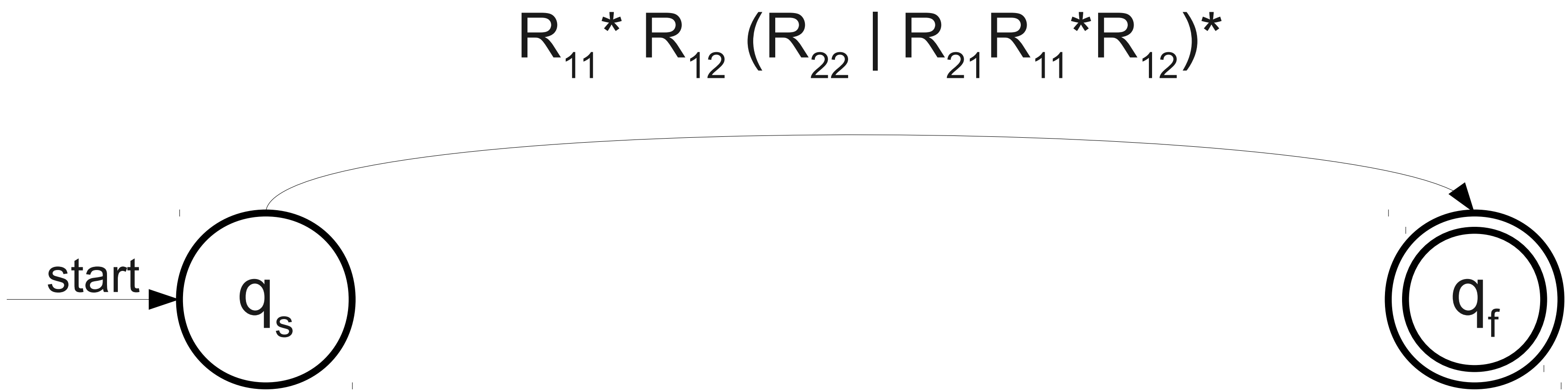
From NFAs to Regular Expressions



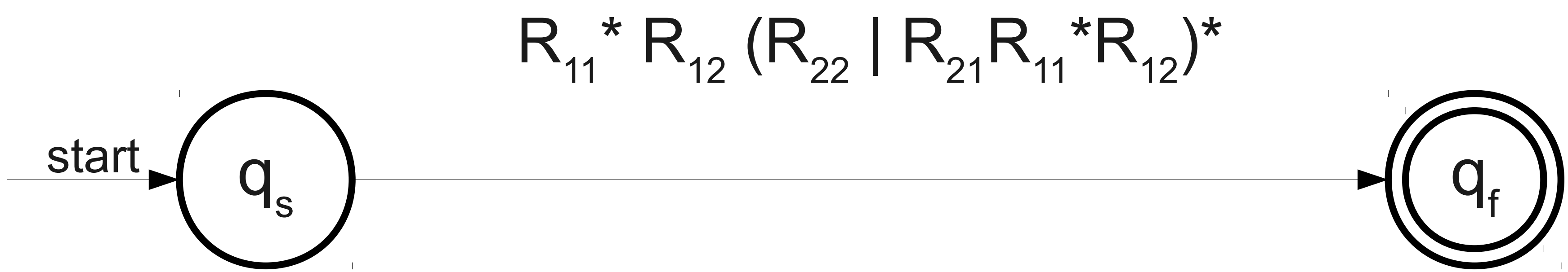
From NFAs to Regular Expressions



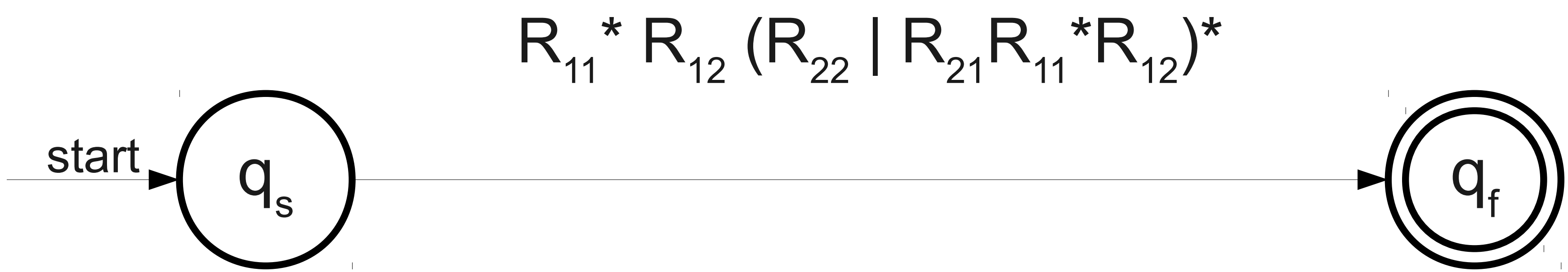
From NFAs to Regular Expressions

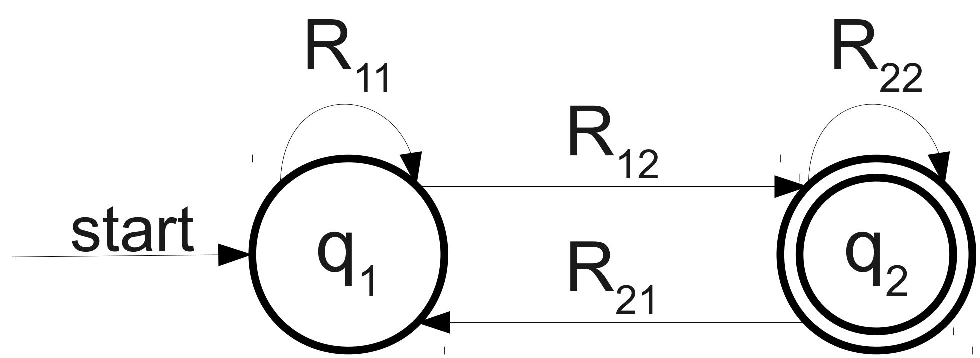


From NFAs to Regular Expressions



From NFAs to Regular Expressions





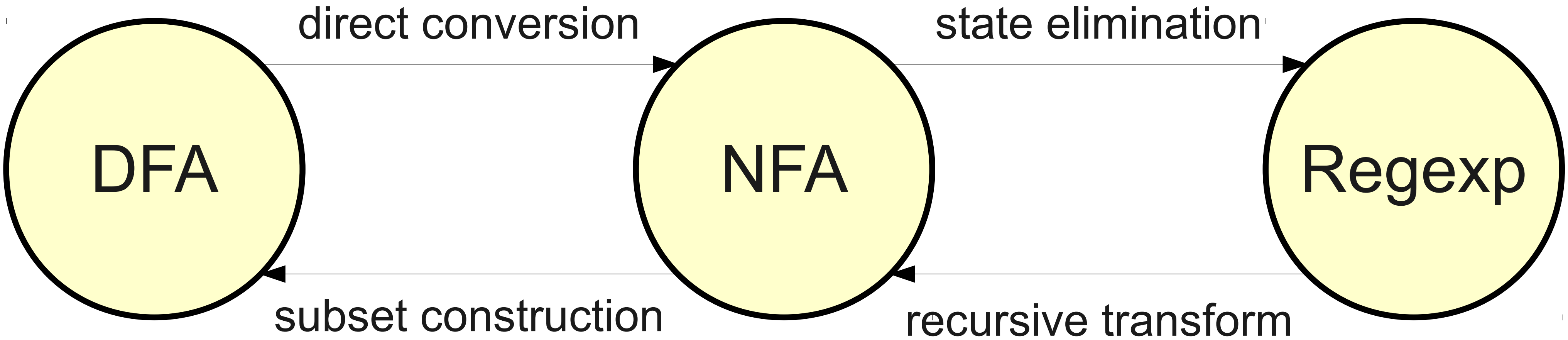
The Construction at a Glance

- Start with an NFA for the language L.
- Add a new start state q_s and accept state q_f to the NFA.
  - Add ε-transitions from each original accepting state to q_f, then mark them as not accepting.
- Repeatedly remove states other than q_s and q_f from the NFA by "shortcutting" them until only two states remain: q_s and q_f.
- The transition from q_s to q_f is then a regular expression for the NFA.

There's another example!

Check the appendix to this slide deck.

Our Transformations



Regular Languages

- A language L is regular iff
  - L is accepted by some DFA.
  - L is accepted by some NFA.
  - L is described by some regular expression.
- What constructions on regular languages can we do with regular expressions?

String Homomorphism

- Let Σ_1 and Σ_2 be alphabets.
- Consider any function $h : \Sigma_1 \rightarrow \Sigma_2^*$ that associates symbols in Σ_1 with strings in Σ_2^*.
- For example:
  - $\Sigma_1 = \{ \mathtt{0}, \mathtt{1} \}$
  - $\Sigma_2 = \{ \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d} \}$
  - $h(\mathtt{0}) = \mathtt{acdb}$
  - $h(\mathtt{1}) = \mathtt{ccc}$

String Homomorphism

- Let Σ_1 and Σ_2 be alphabets.
- Consider any function $h : \Sigma_1 \to \Sigma_2^*$ that associates symbols in Σ_1 with strings in Σ_2^*.
- For example:
  - $\Sigma_1 = \{$ **`a`**, **`b`**, **`c`**, **`d`**, … $\}$
  - $\Sigma_2 = \{$ **`A`**, **`B`**, **`C`**, **`D`**, … $\}$
  - $h($**`a`**$) =$ **`A`**
  - $h($**`b`**$) =$ **`B`**
  - …

String Homomorphism

- Let Σ_1 and Σ_2 be alphabets.
- Consider any function $h : \Sigma_1 \to \Sigma_2^*$ that associates symbols in Σ_1 with strings in Σ_2^*.
- For example:
  - $\Sigma_1 = \{ \texttt{0}, \texttt{1} \}$
  - $\Sigma_2 = \{ \texttt{0}, \texttt{1} \}$
  - $h(\texttt{0}) = \varepsilon$
  - $h(\texttt{1}) = \texttt{1}$

String Homomorphism

- Given a function $h : \Sigma_1 \to \Sigma_2^*$, the function $h^* : \Sigma_1^* \to \Sigma_2^*$ is formed by applying h to each character of a string w.
- This function is called a **string homomorphism**.
  - From Greek "same shape."

String Homomorphism, Intuitively

- Example: Let Σ_1 = { **0**, **1**, **2** } and consider the string **0121**
- If Σ_2 = {**A**, **B**, **C**, …, **Z**, **a**, **b**, …, **z**, **'**, **[**, **]**, **.** }, define $h : \Sigma_1 \to \Sigma_2^*$ as
  - h(**0**) = **That's the way**
  - h(**1**) = **[Uh huh uh huh]**
  - h(**2**) = **I like it**
- Then h^*(**0121**) = **That's the way [Uh huh uh huh] I like it [Uh huh uh huh]**
- Note that h^*(**0121**) has the same structure as **0121**, just expressed differently.

Homomorphisms of Languages

- If $L \subseteq \Sigma_1^*$ is a language and $h^* : \Sigma_1^* \to \Sigma_2^*$ is a homomorphism, the language ***h*(L)*** is defined as

$$h^*(L) = \{ \; h^*(w) \mid w \in L \; \}$$

- The language formed by applying the homomorphism to every string in *L*.

Homomorphisms of Regular Languages

- ***Theorem:*** If L is a regular language over Σ_1 and $h^* : \Sigma_1^* \rightarrow \Sigma_2^*$ is a homomorphism, then $h^*(\mathrm{L})$ is a regular language.
- ***Proof sketch:*** Transform a regular expression for L into a regular expression for $h^*(L)$ by replacing all characters in the regular expression with the value of h applied to that character.
- Examples at the end of these slides.

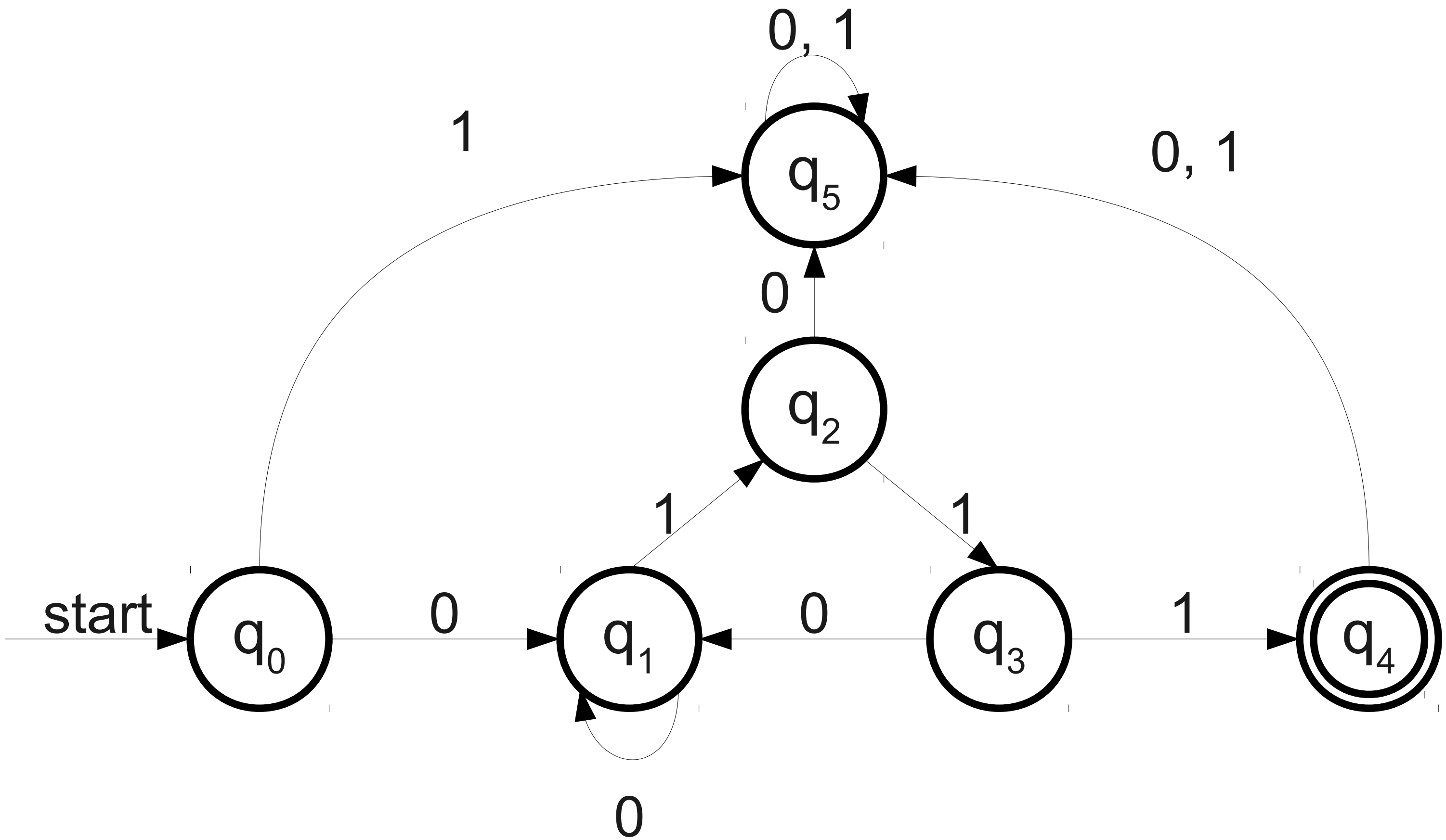
The Big List of Closure Properties

- The regular languages are closed under
  - Union
  - Intersection
  - Complement
  - Concatenation
  - Kleene Closure
  - String Homomorphism
  - **Plus a whole lot more!**

The Limits of Regular Languages

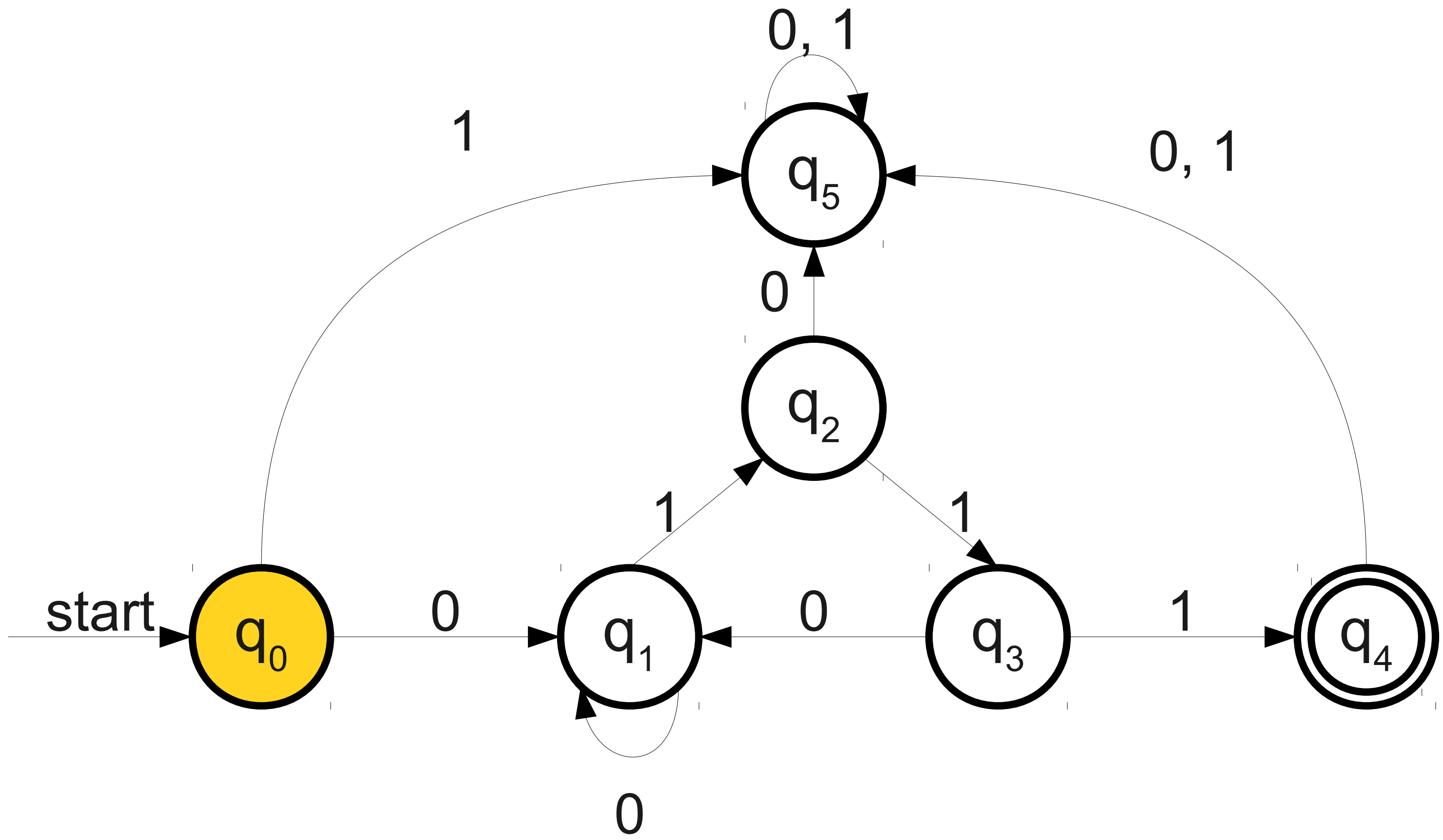
Is every language regular?

An Important Observation



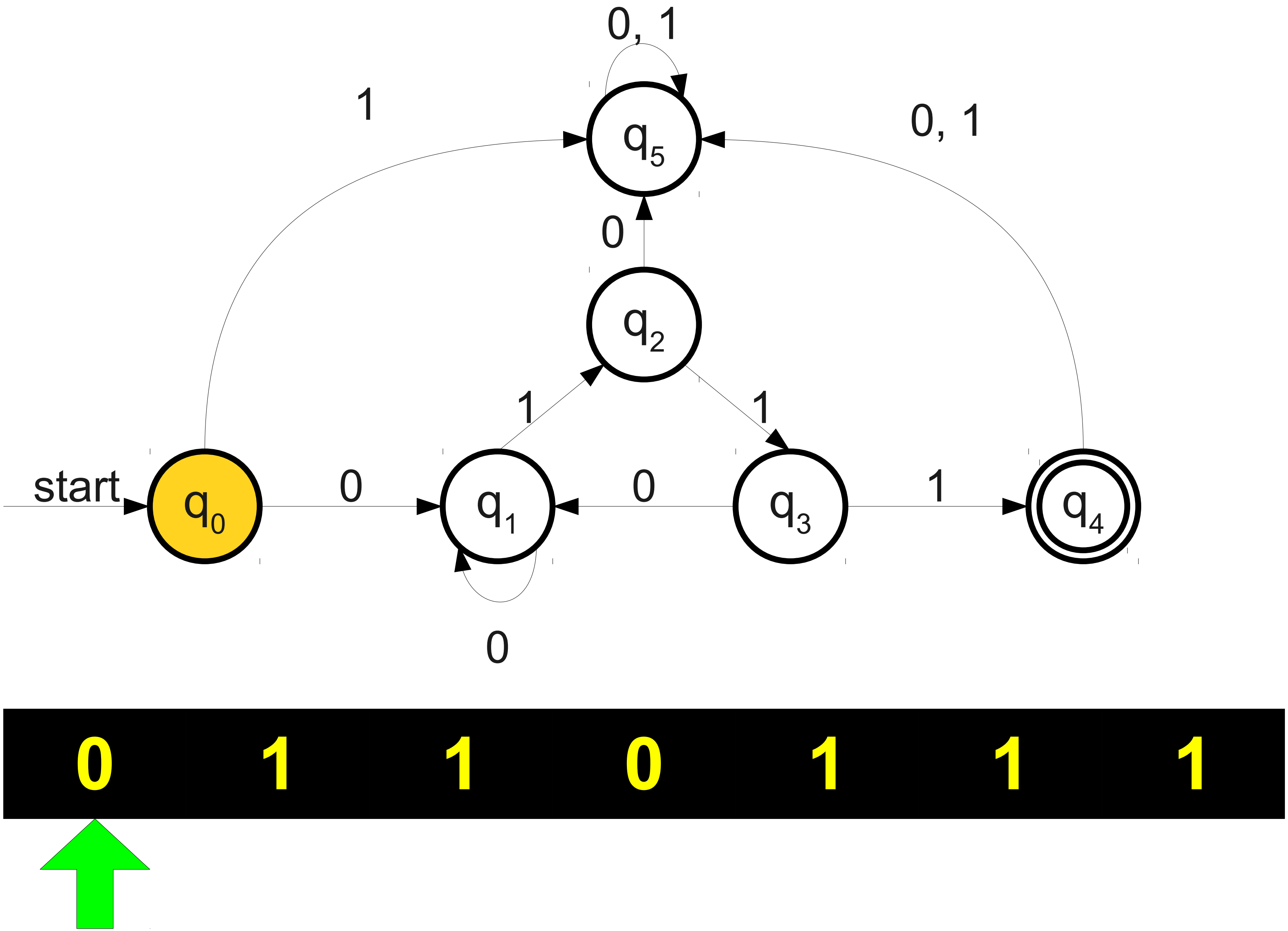
0 1 1 0 1 1 1

An Important Observation

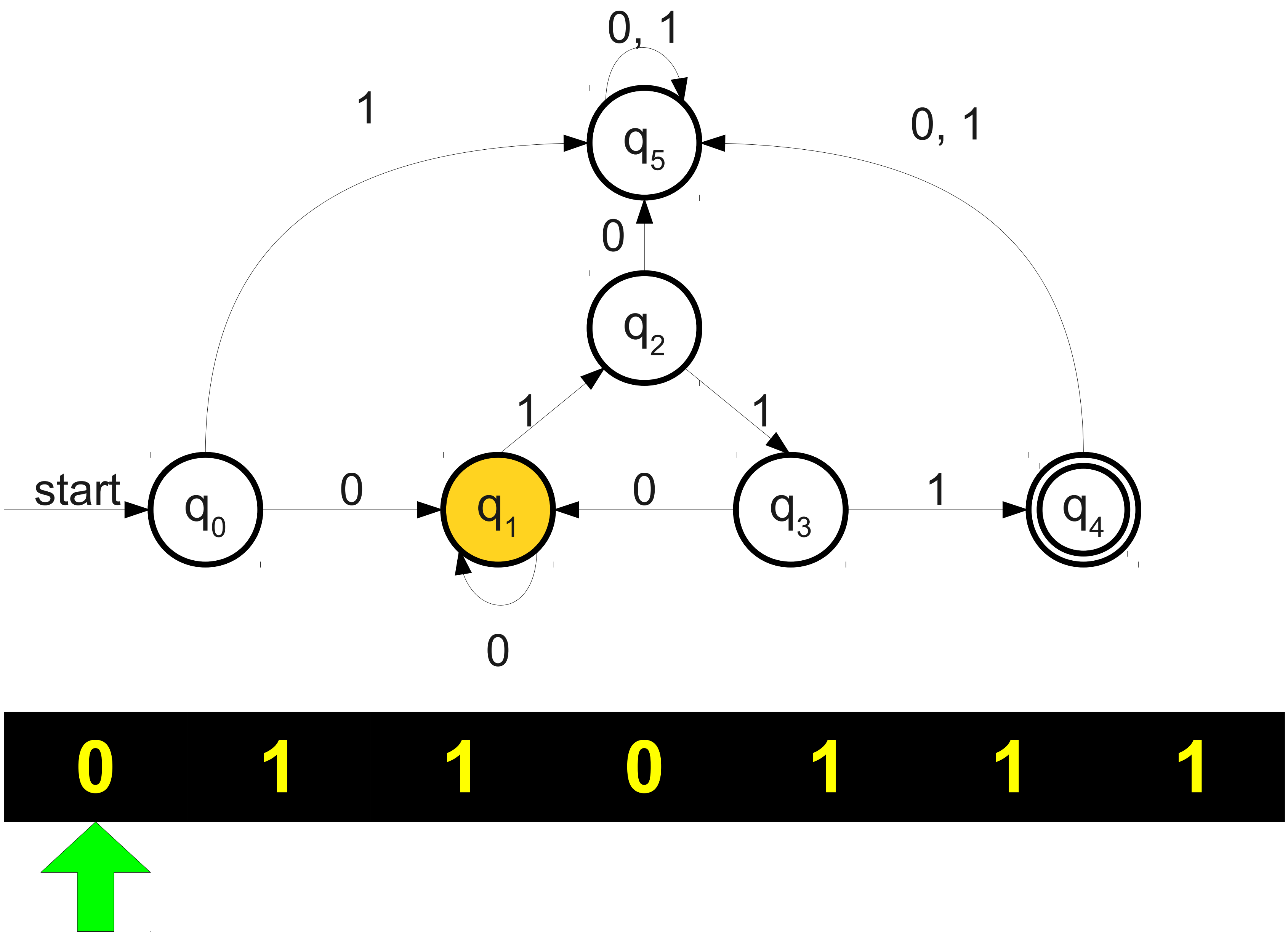


0 1 1 0 1 1 1

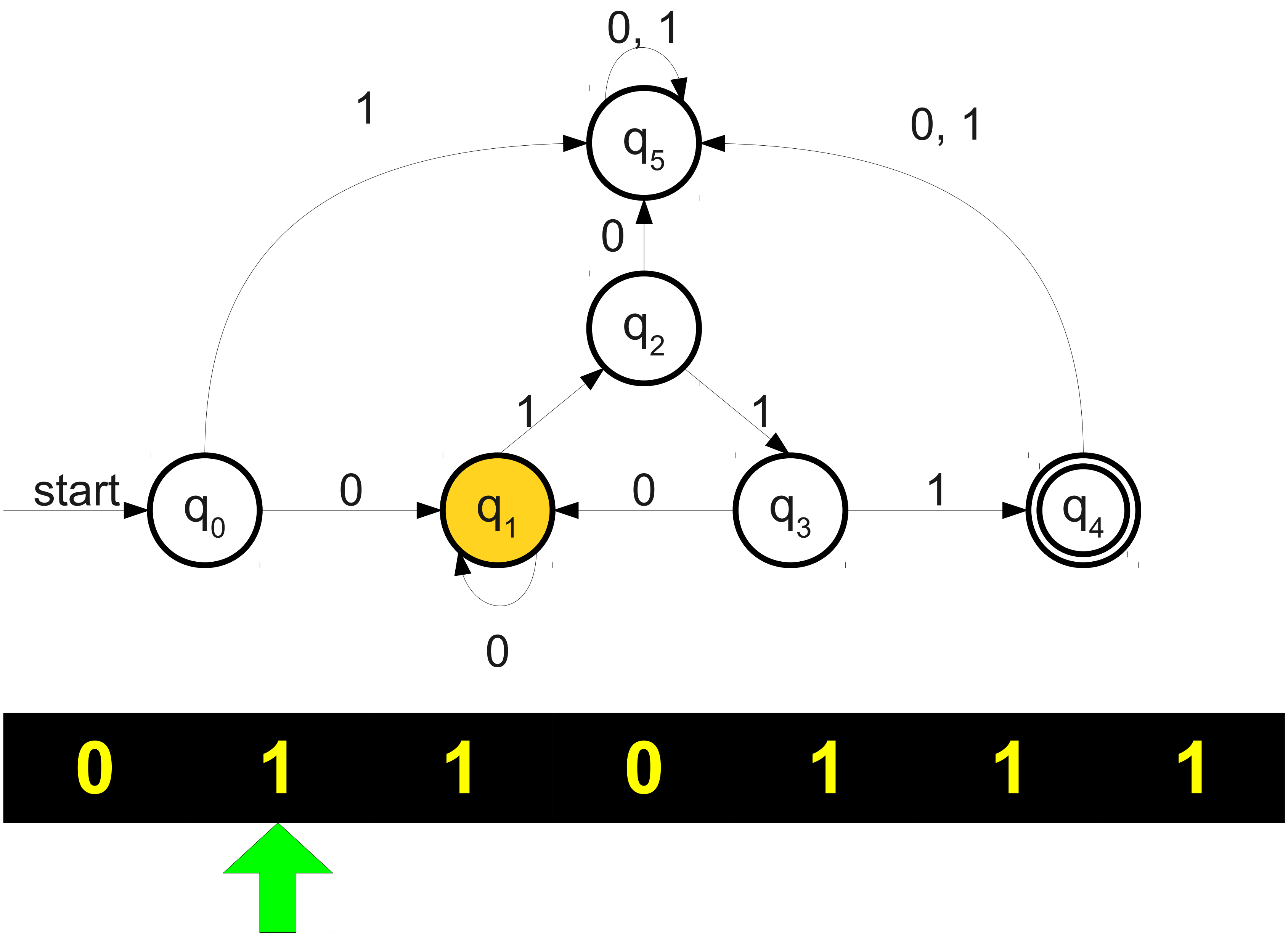
An Important Observation



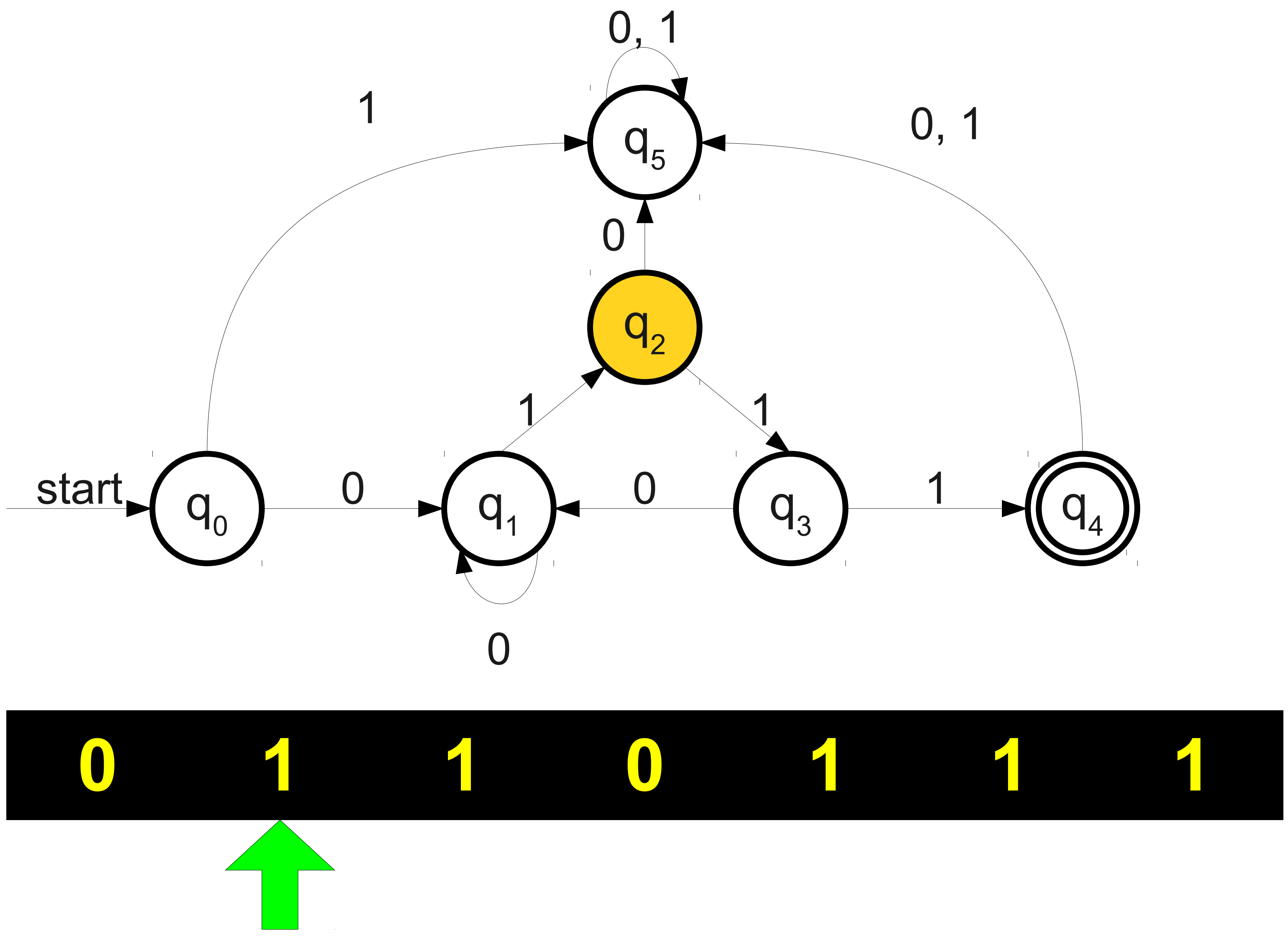
An Important Observation



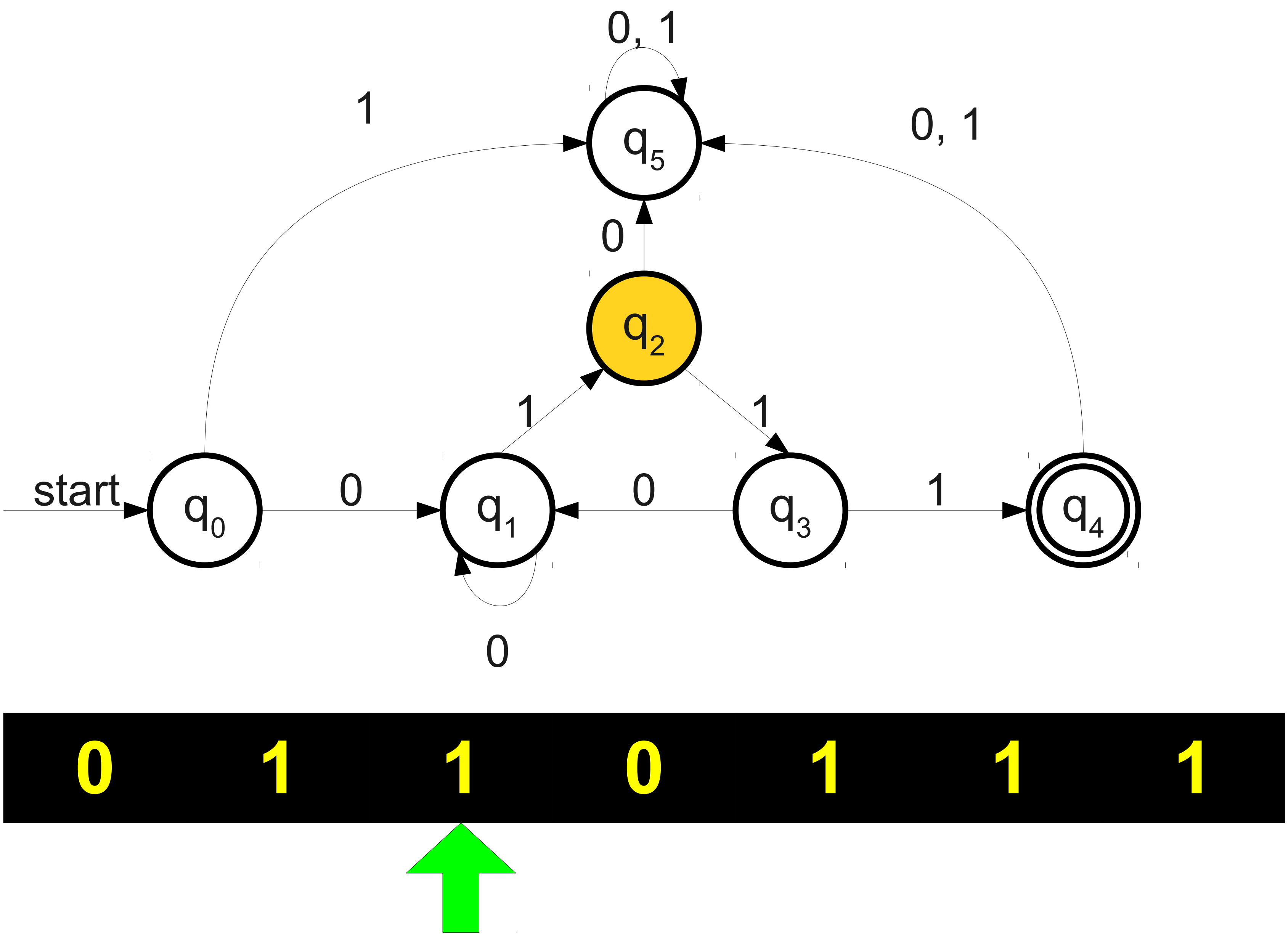
An Important Observation



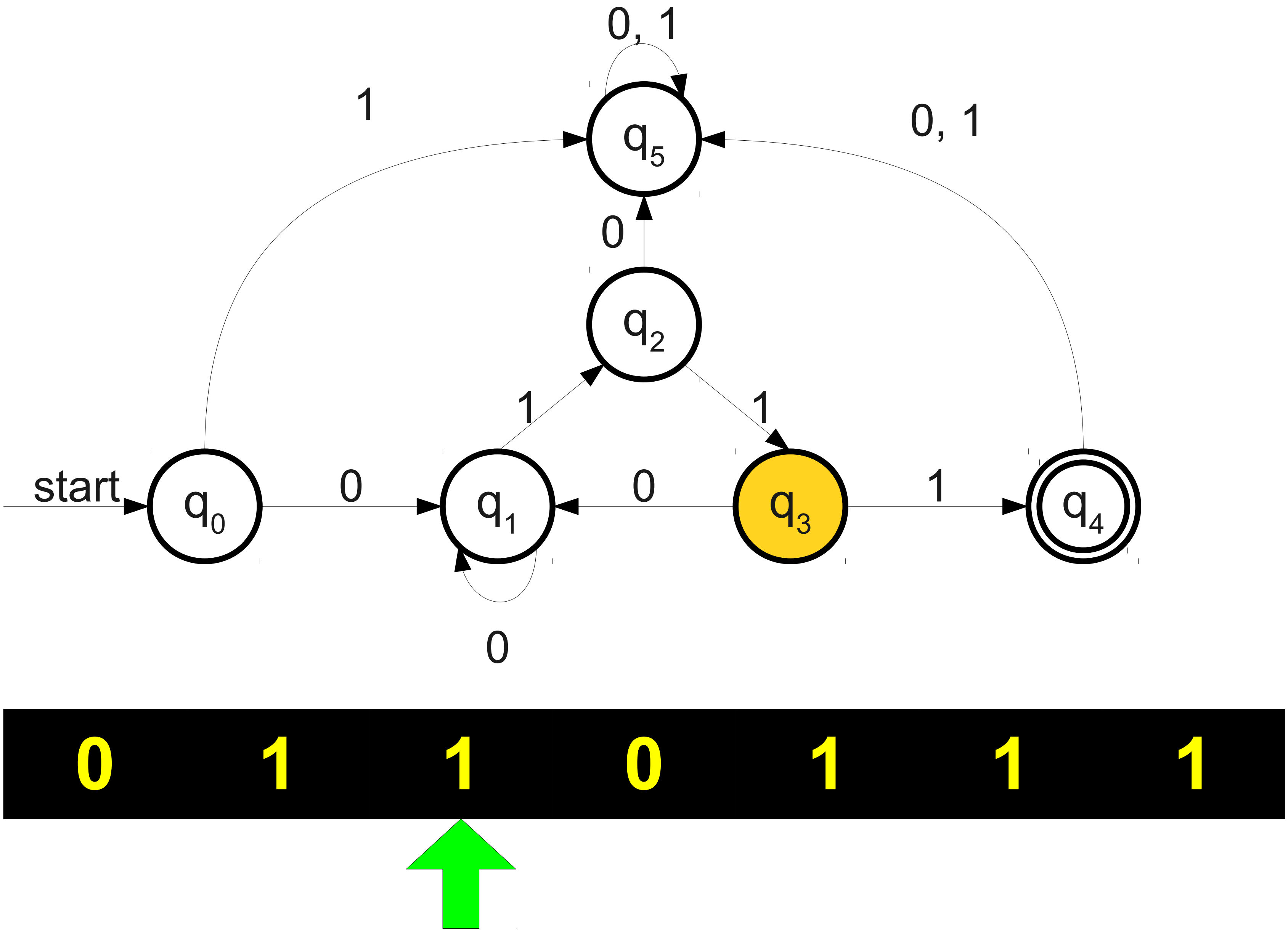
An Important Observation



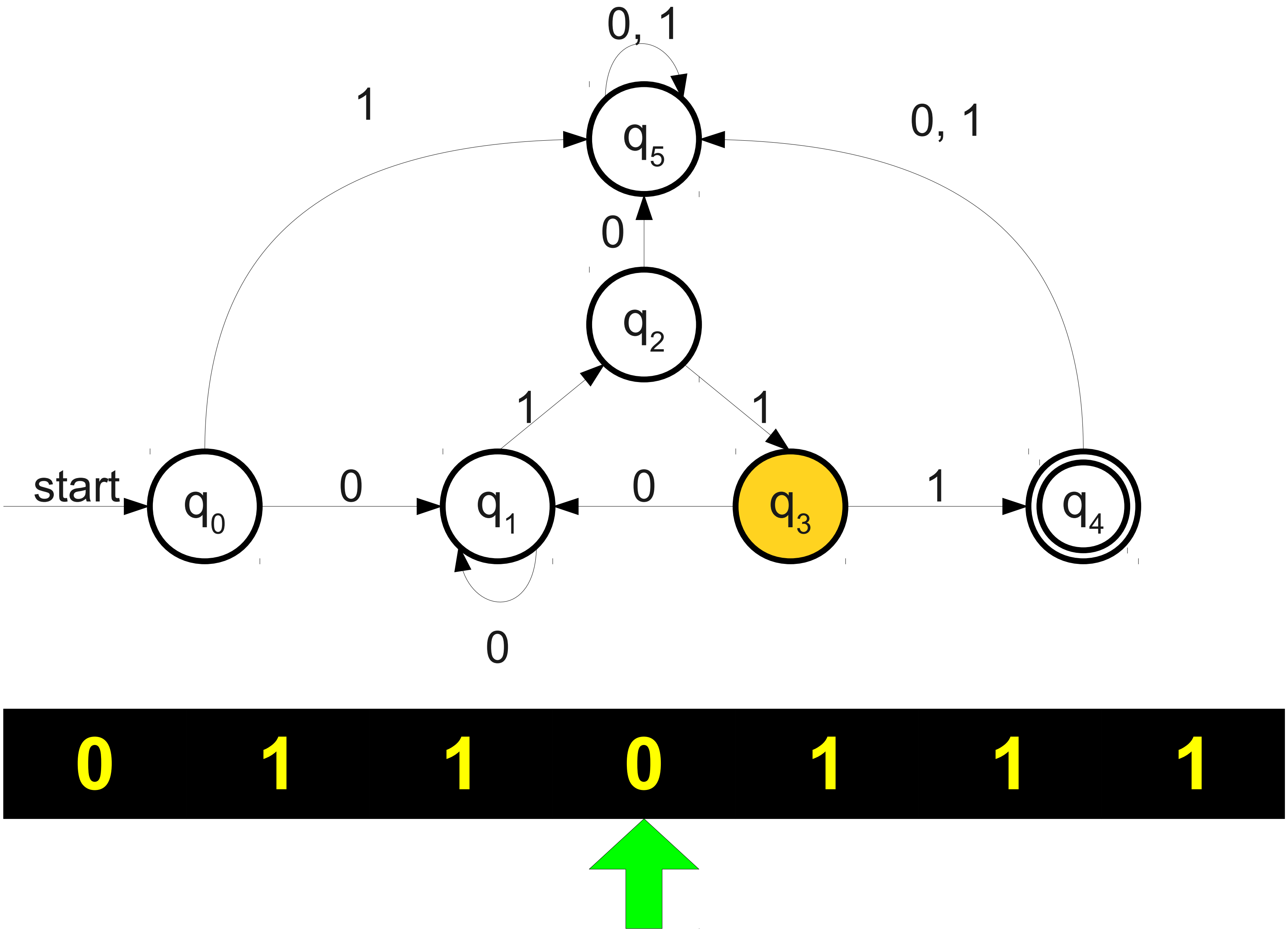
An Important Observation



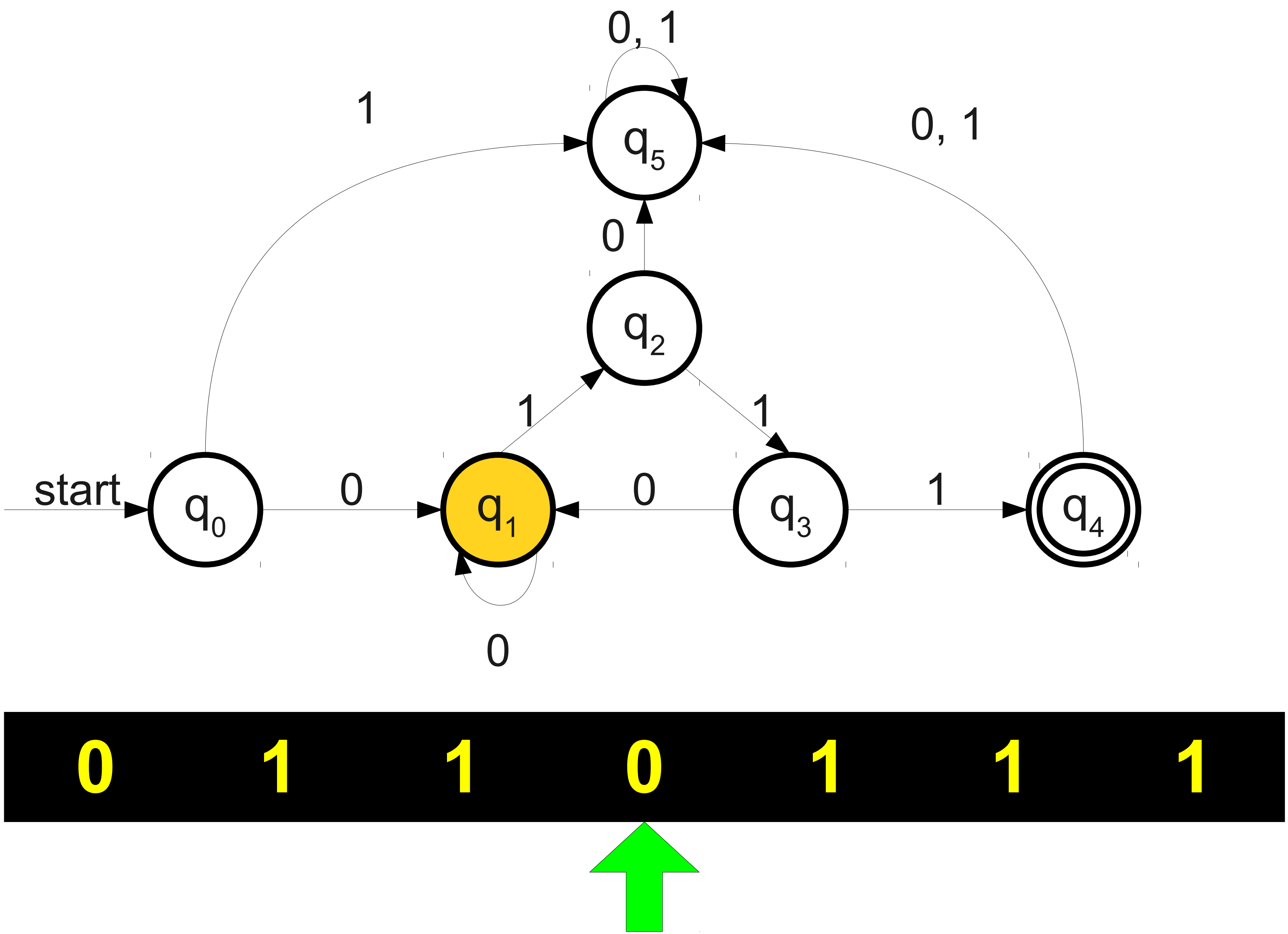
An Important Observation



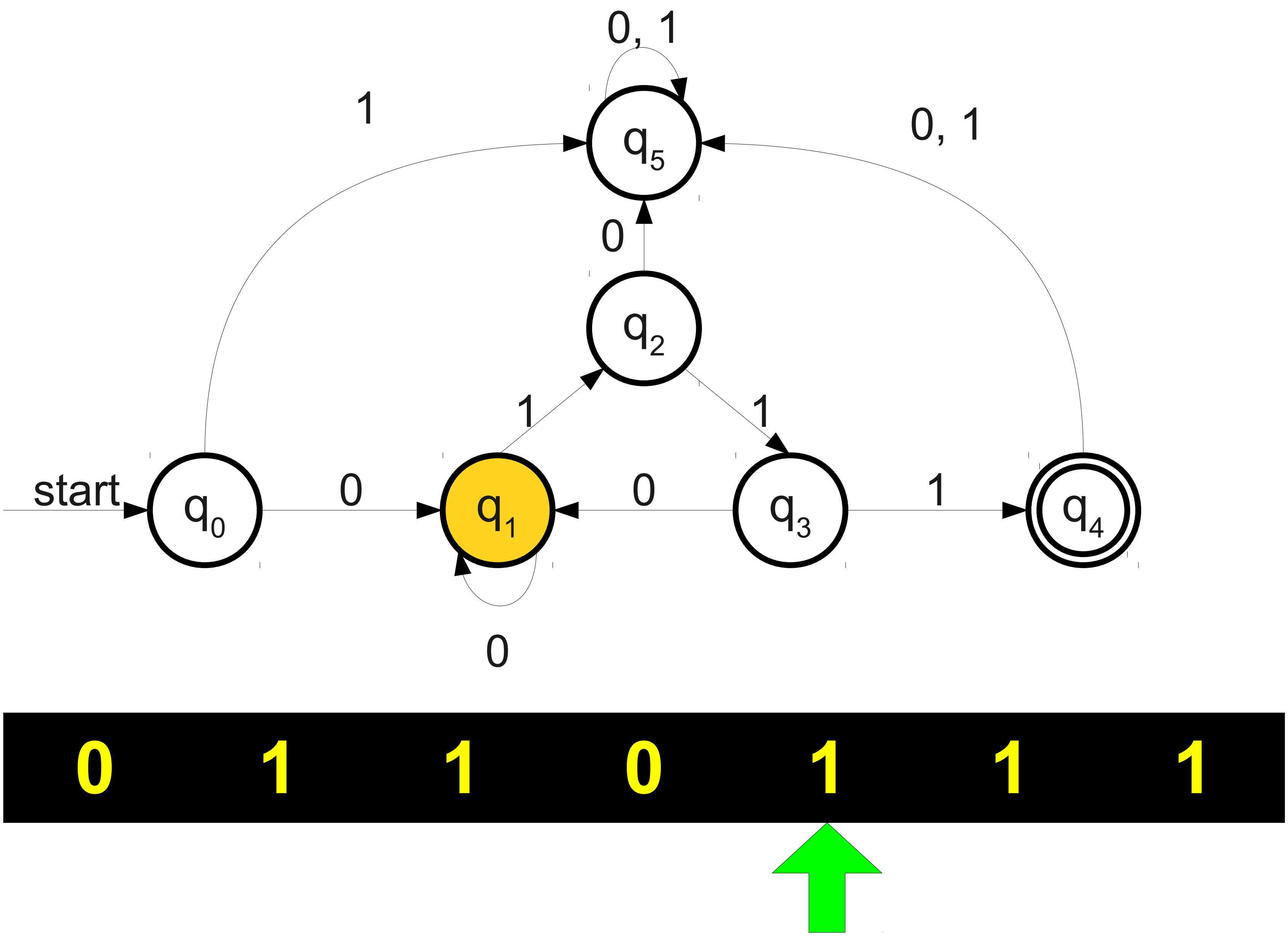
An Important Observation



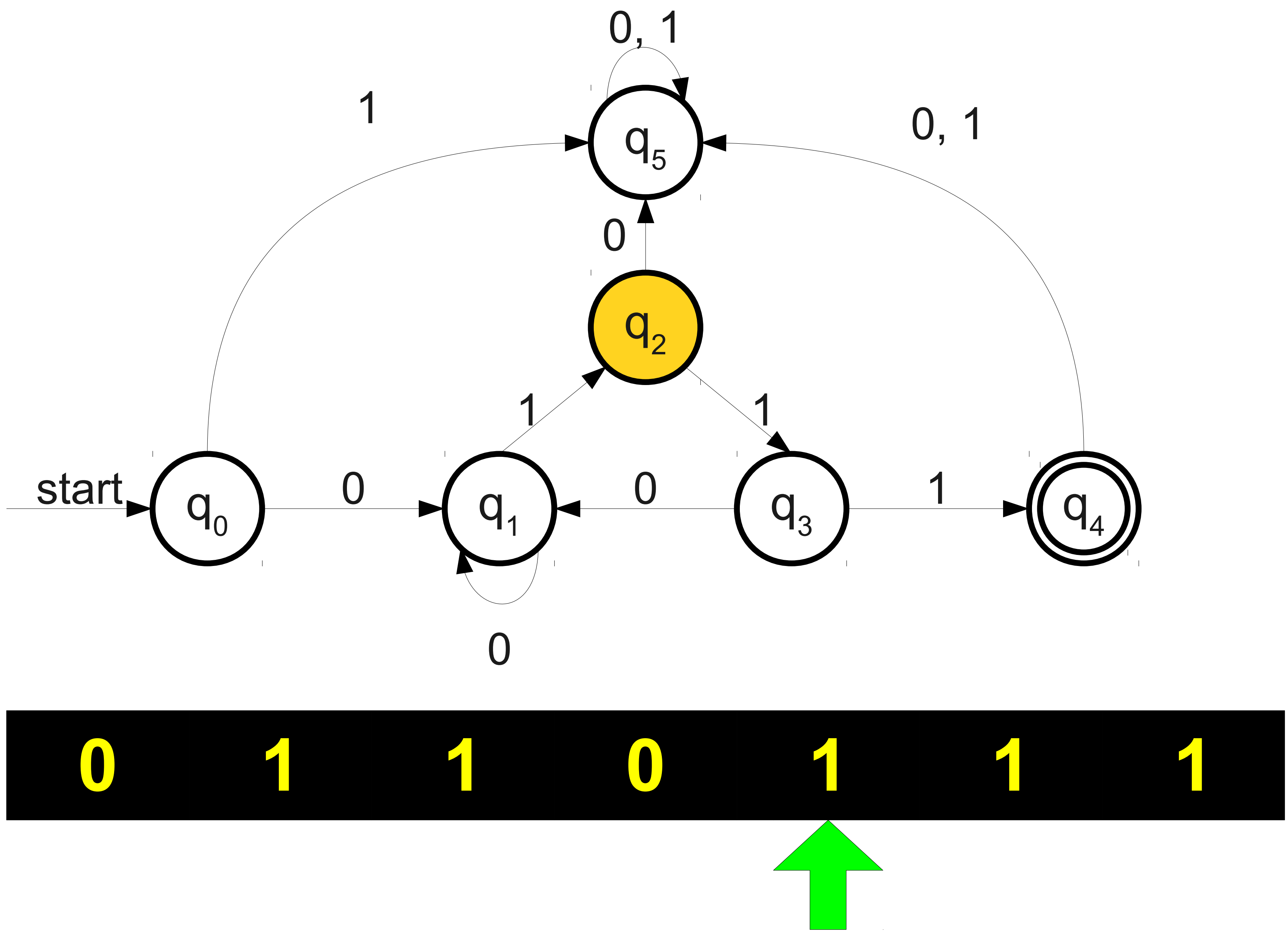
An Important Observation



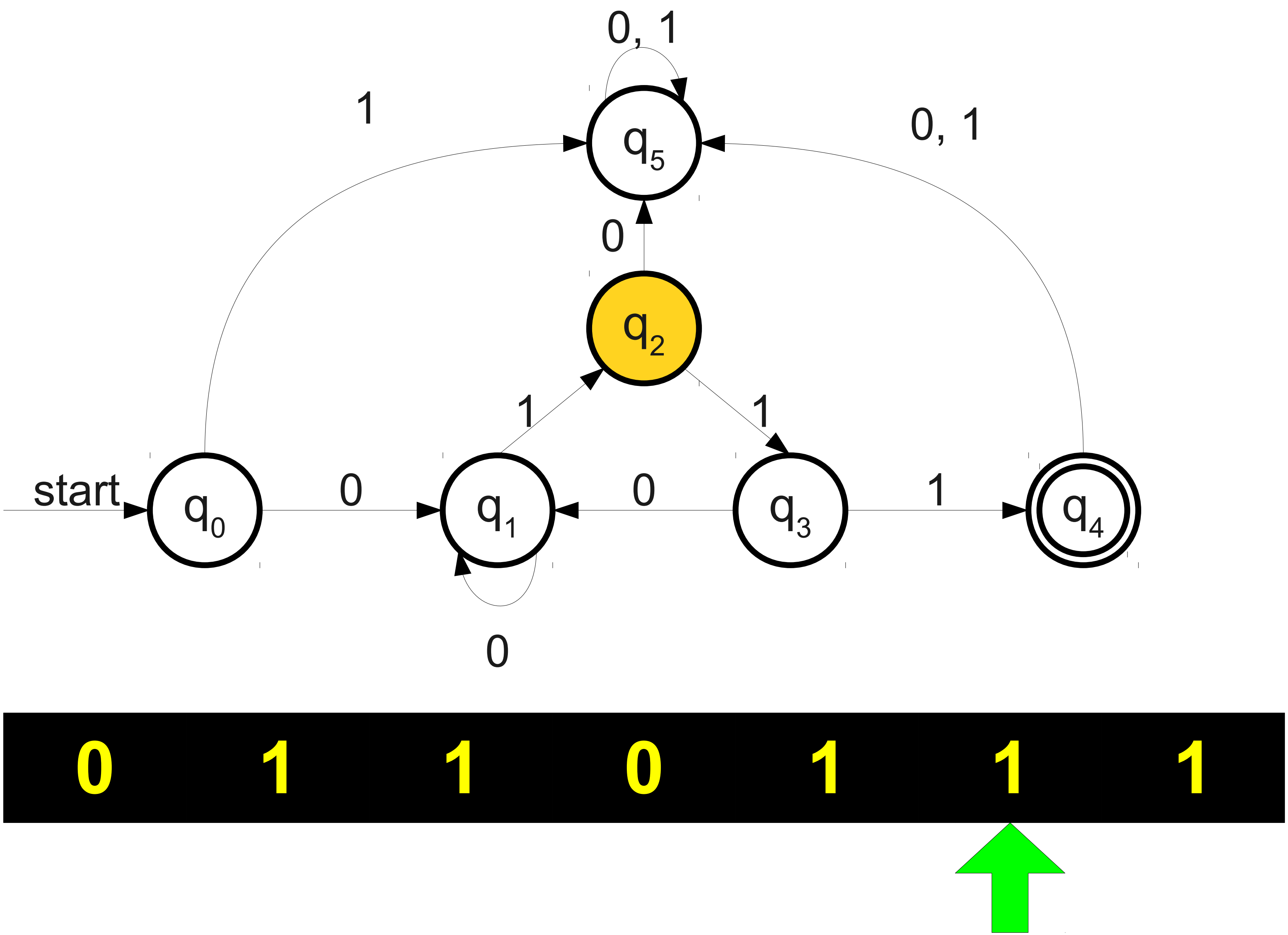
An Important Observation



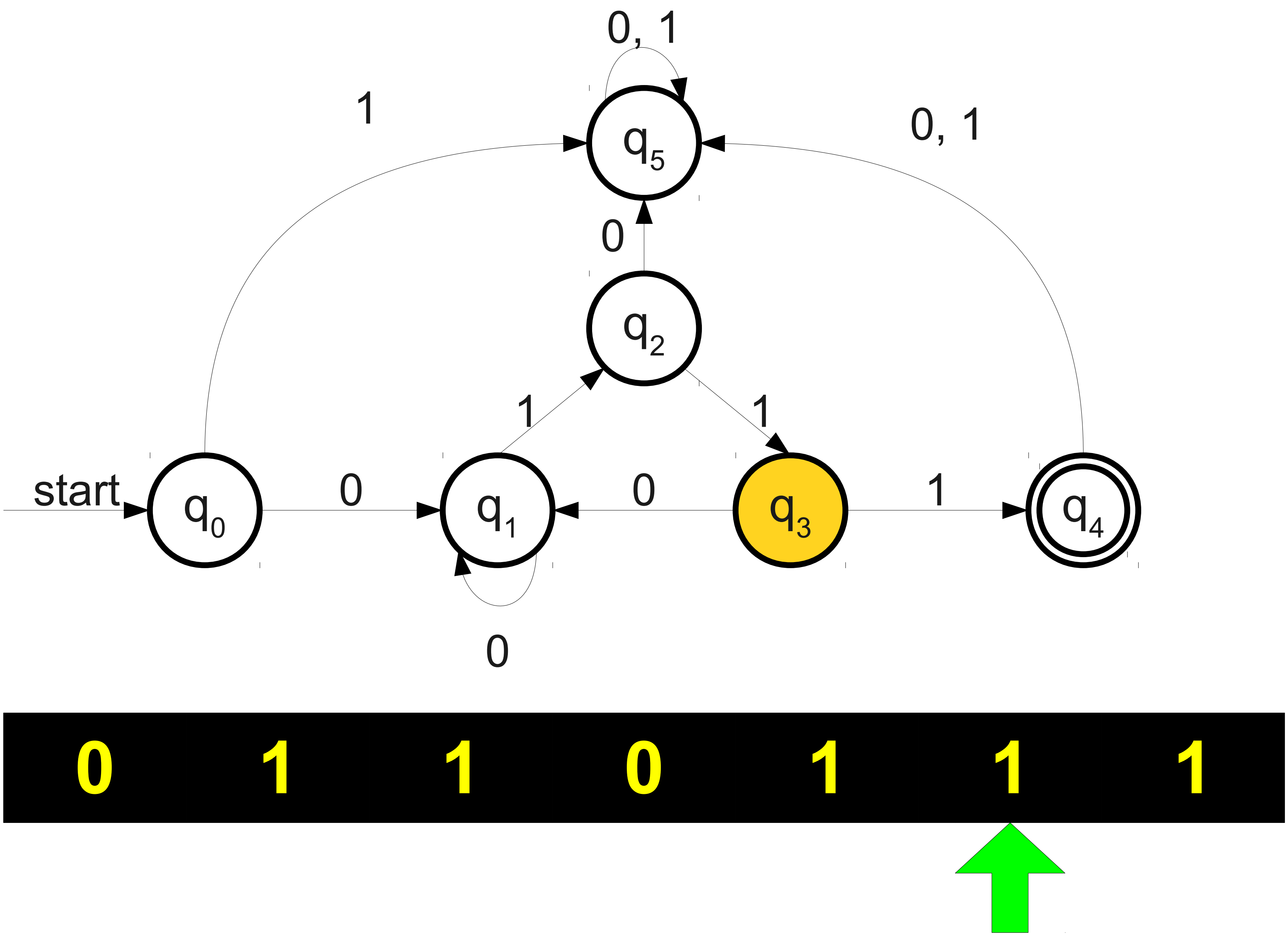
An Important Observation



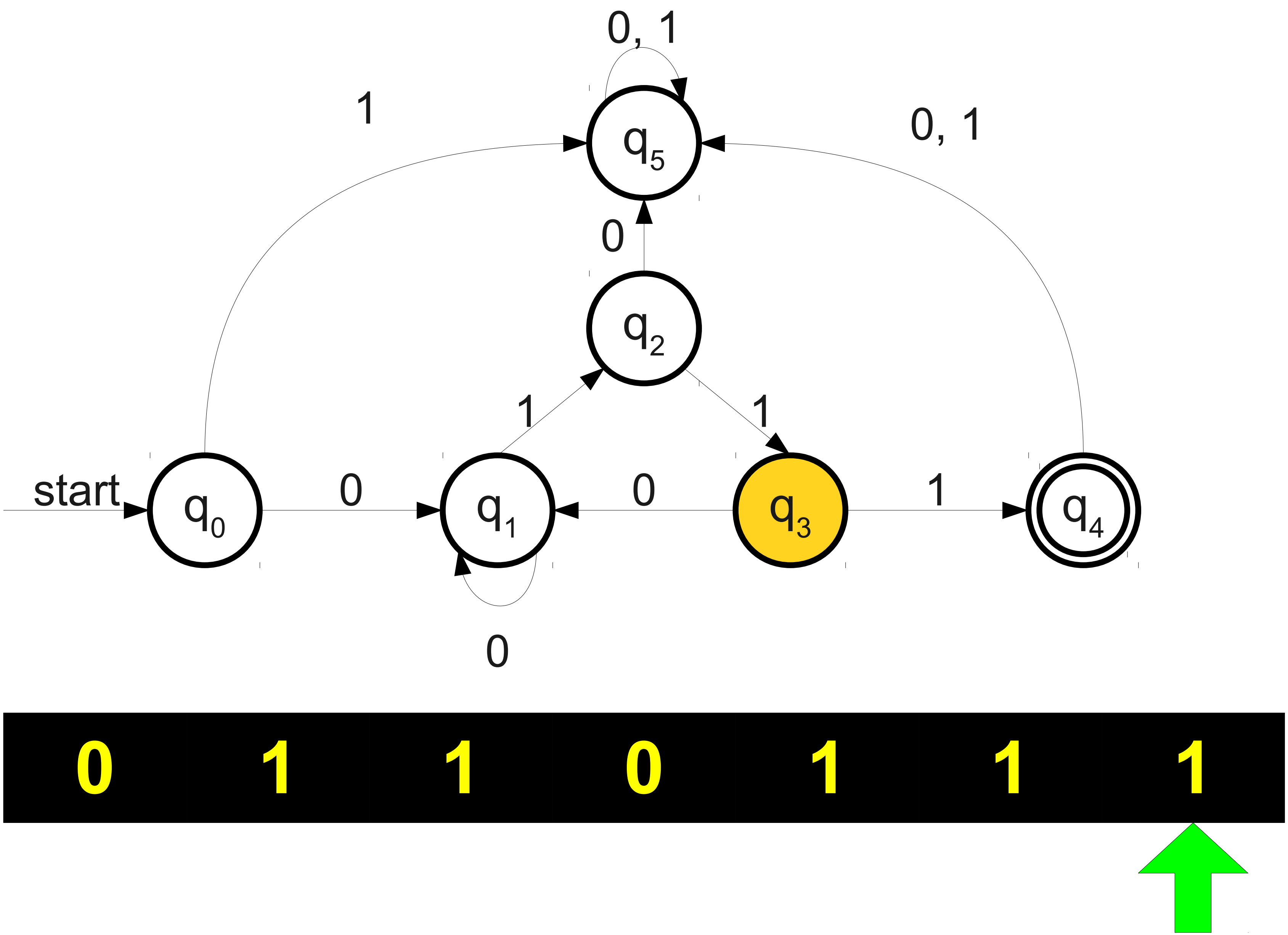
An Important Observation



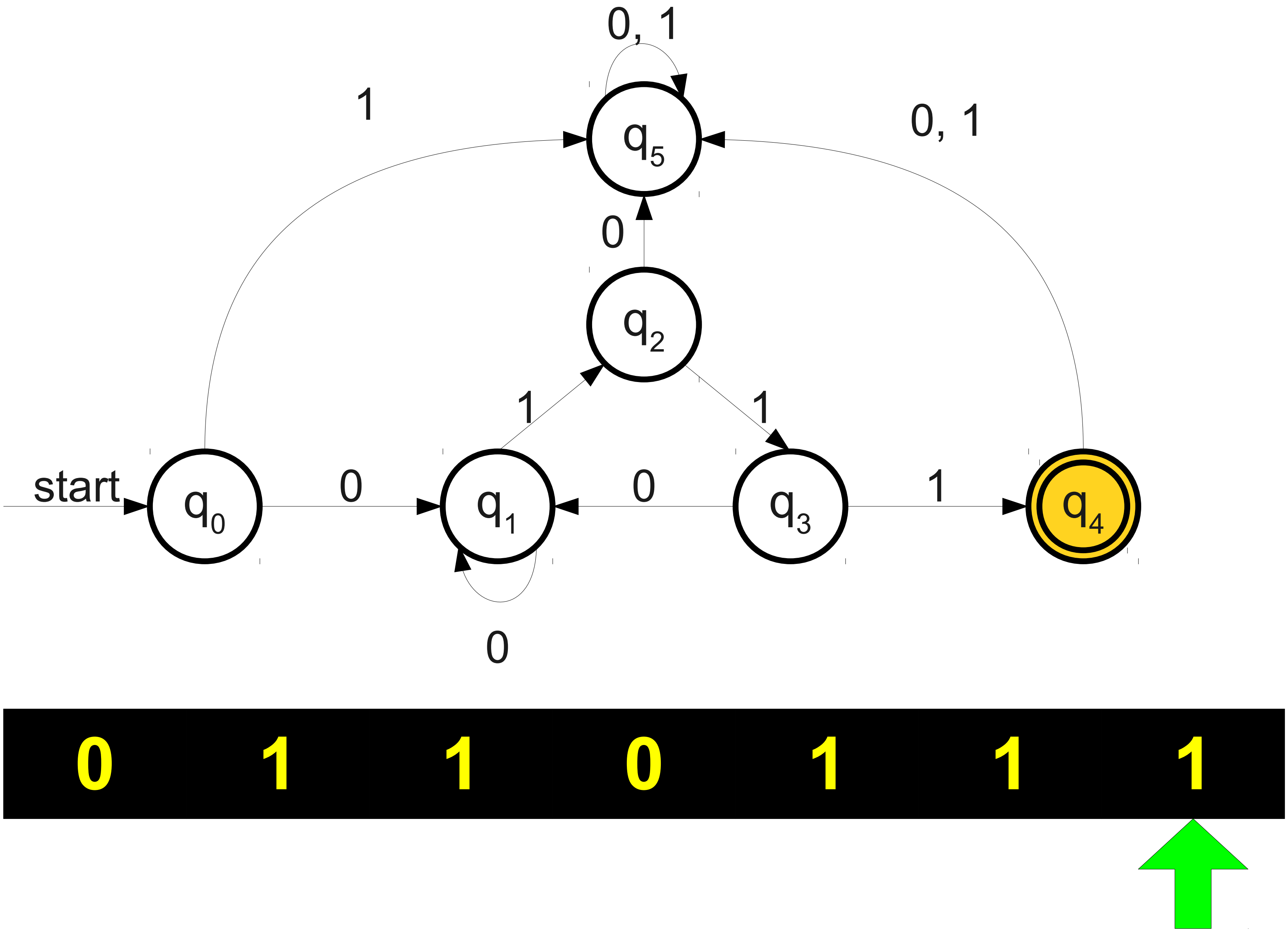
An Important Observation



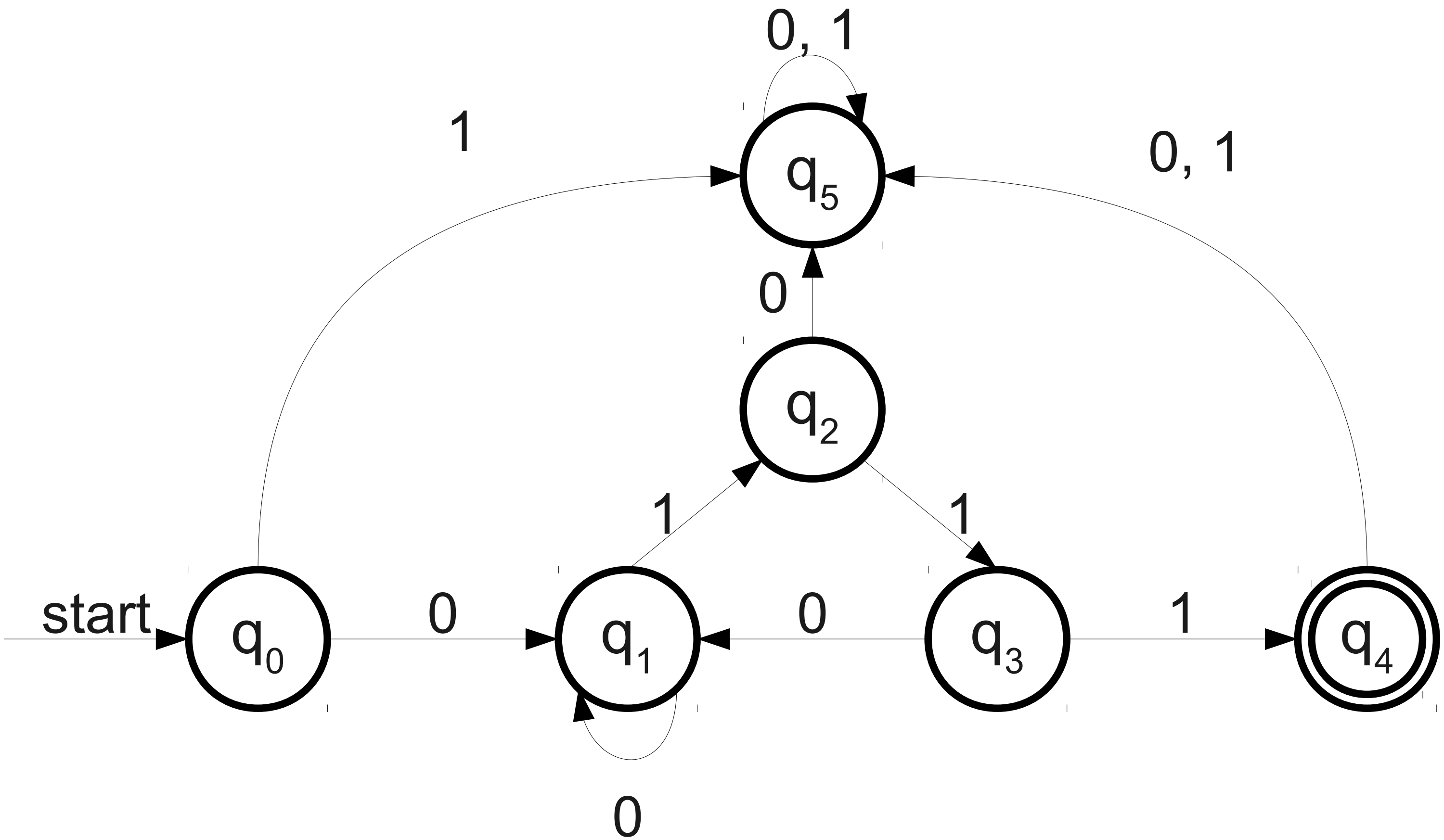
An Important Observation



An Important Observation

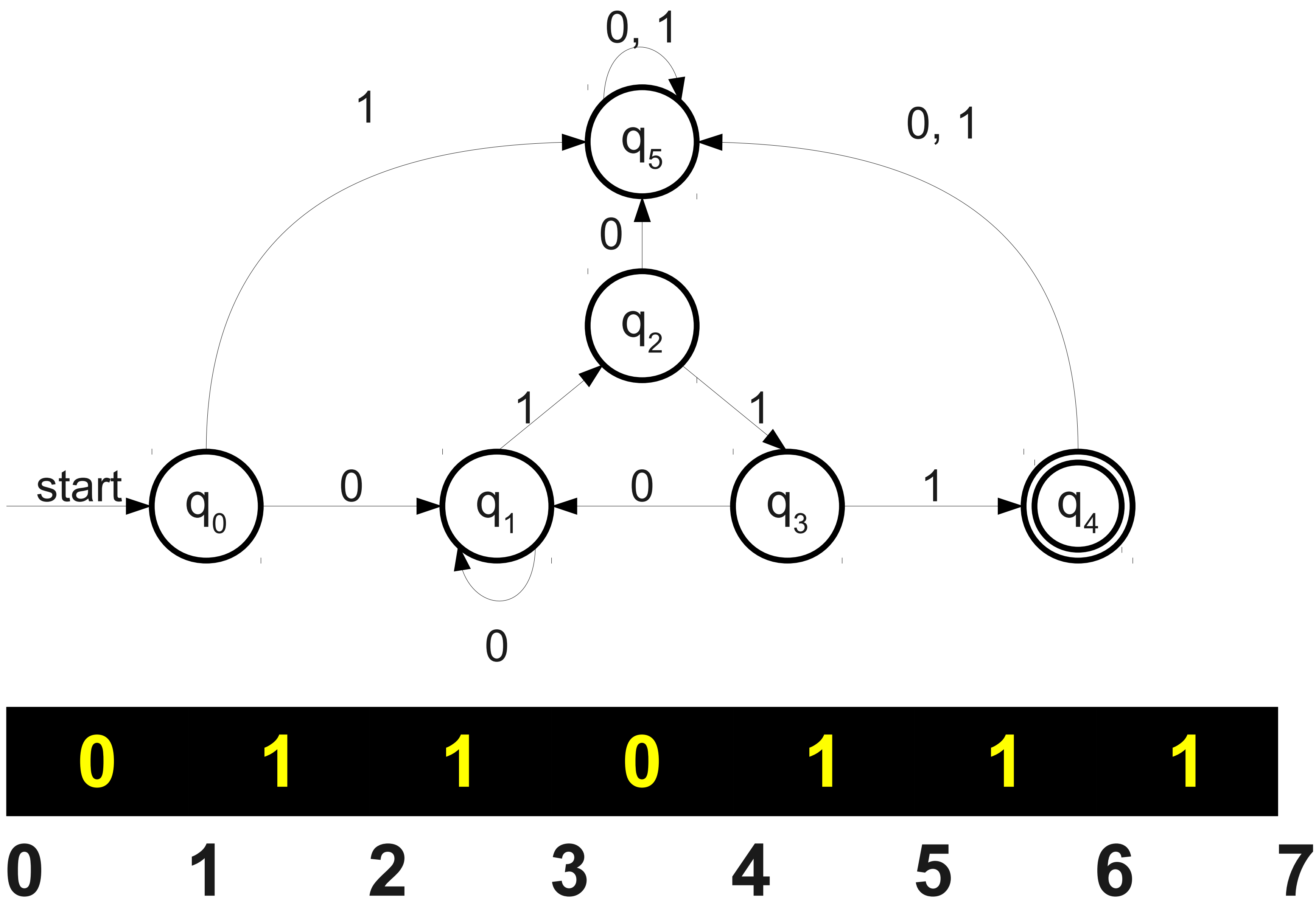


An Important Observation

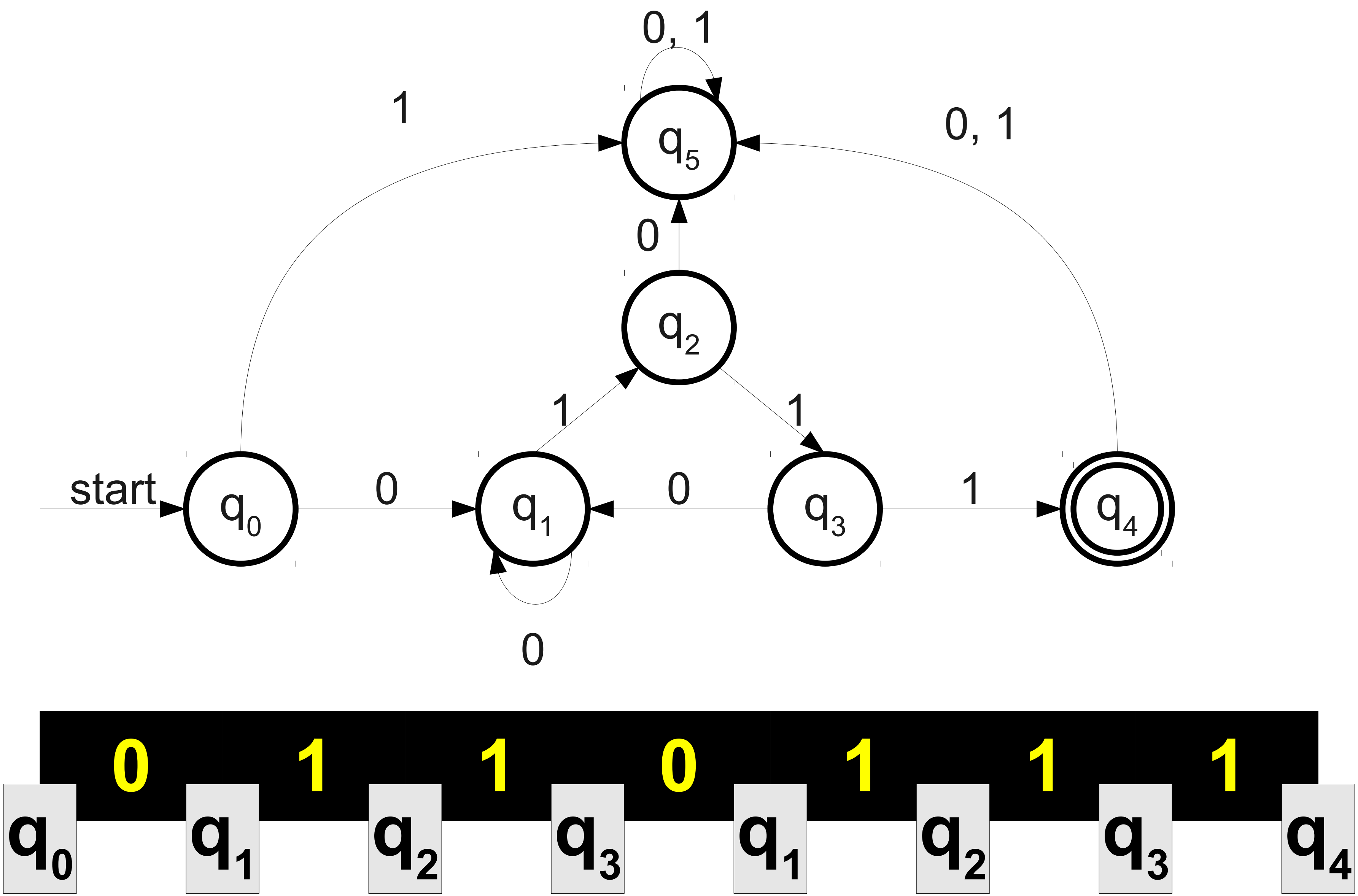


0 1 1 0 1 1 1

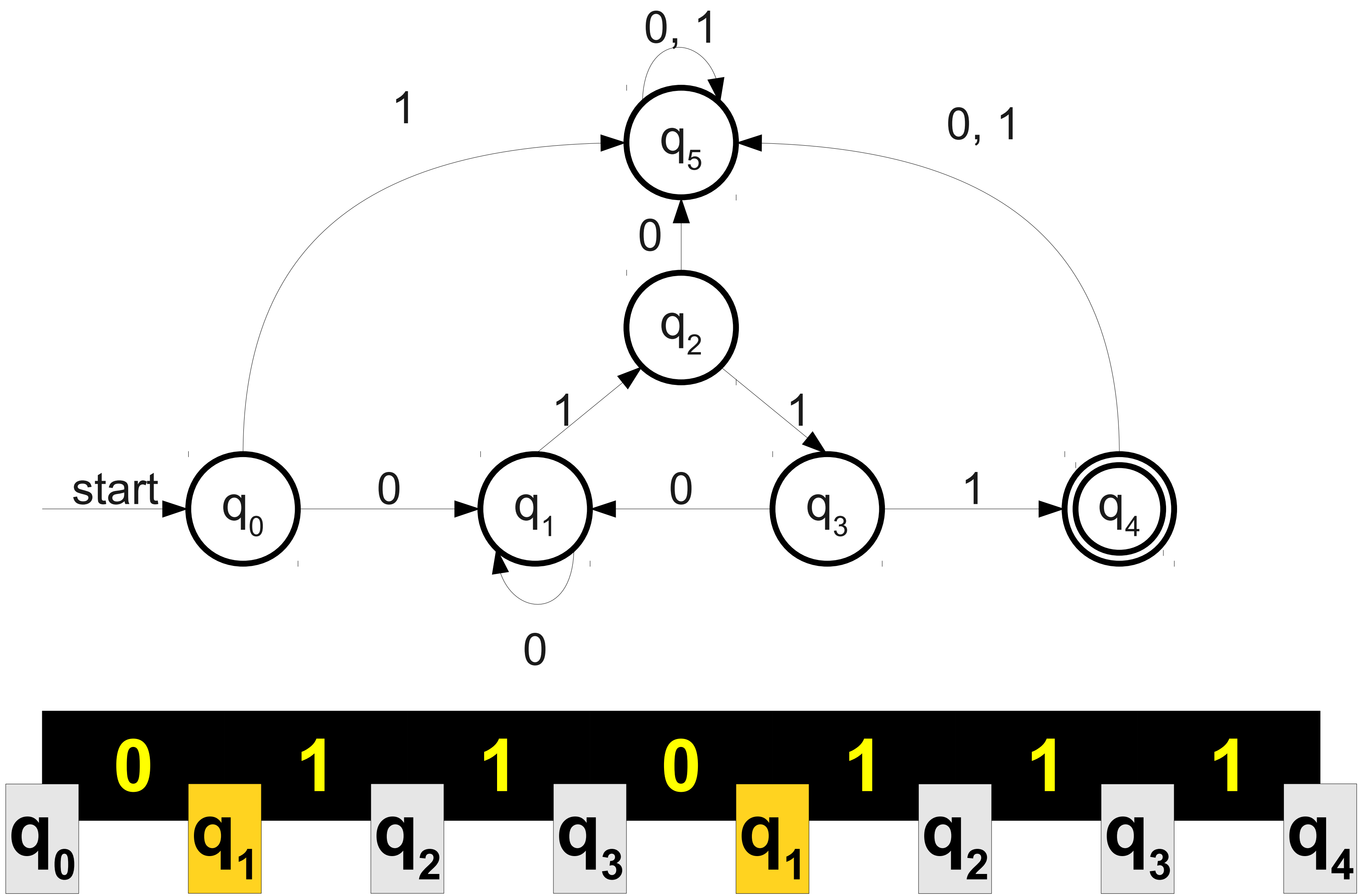
An Important Observation



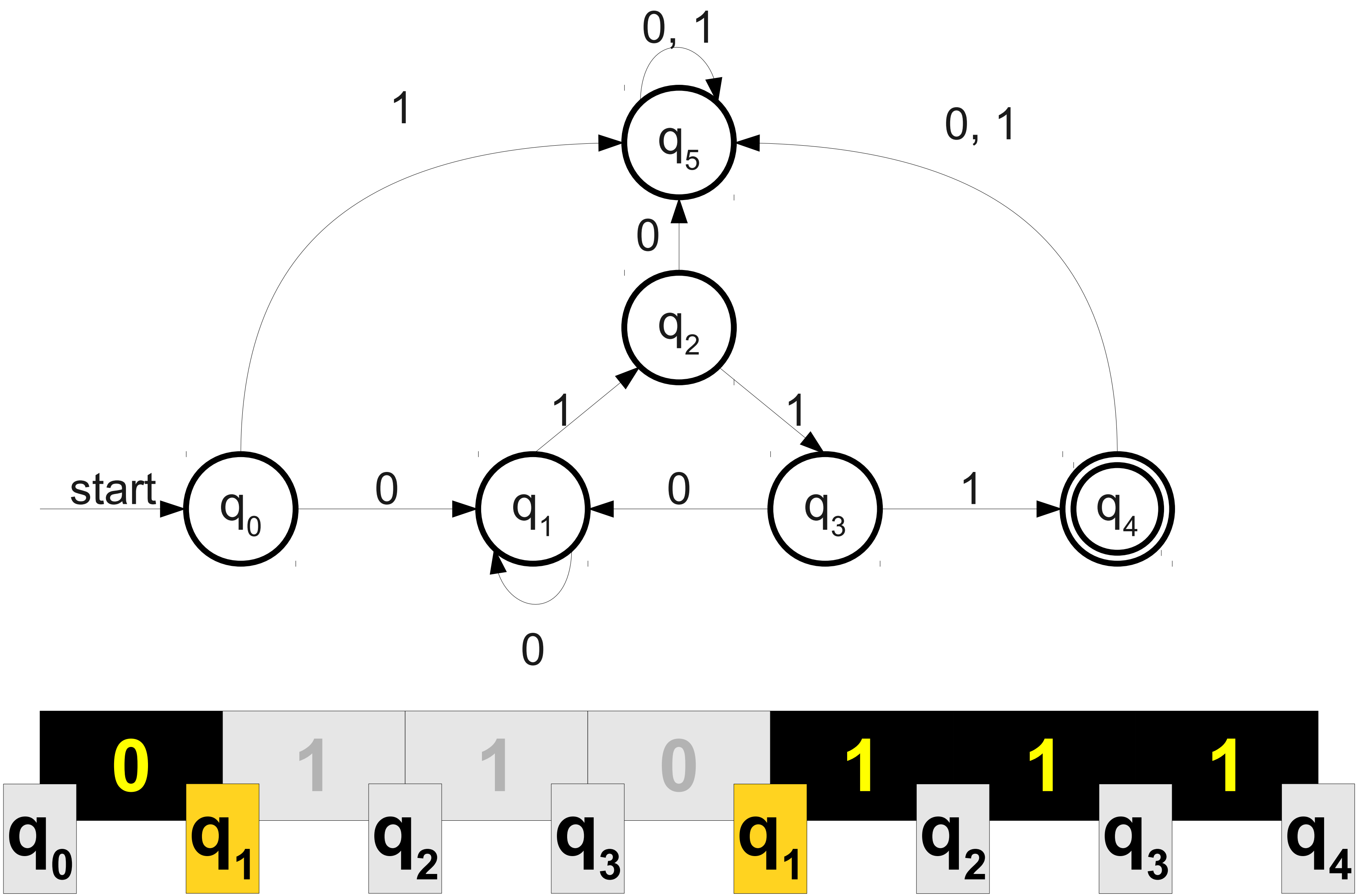
An Important Observation



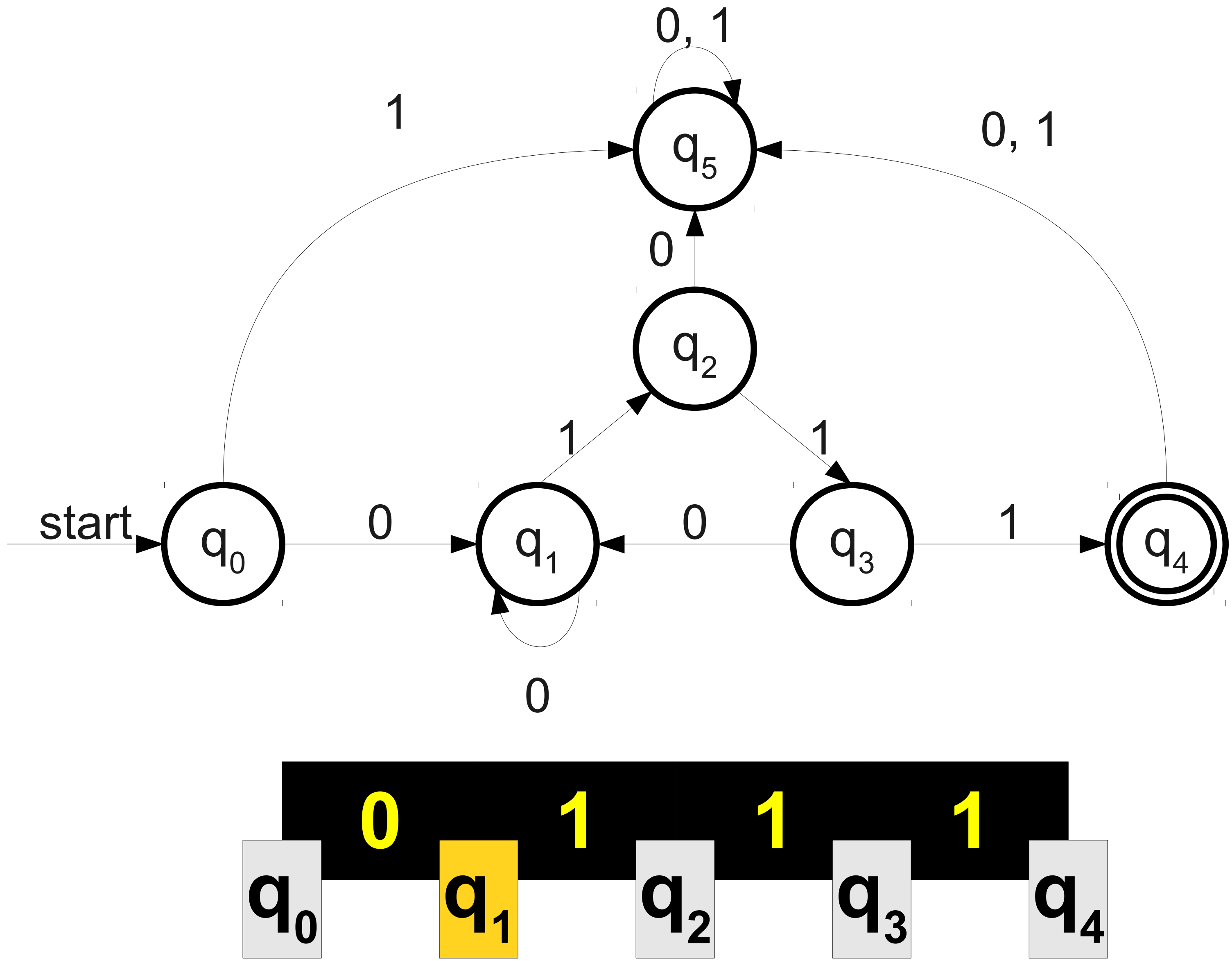
An Important Observation



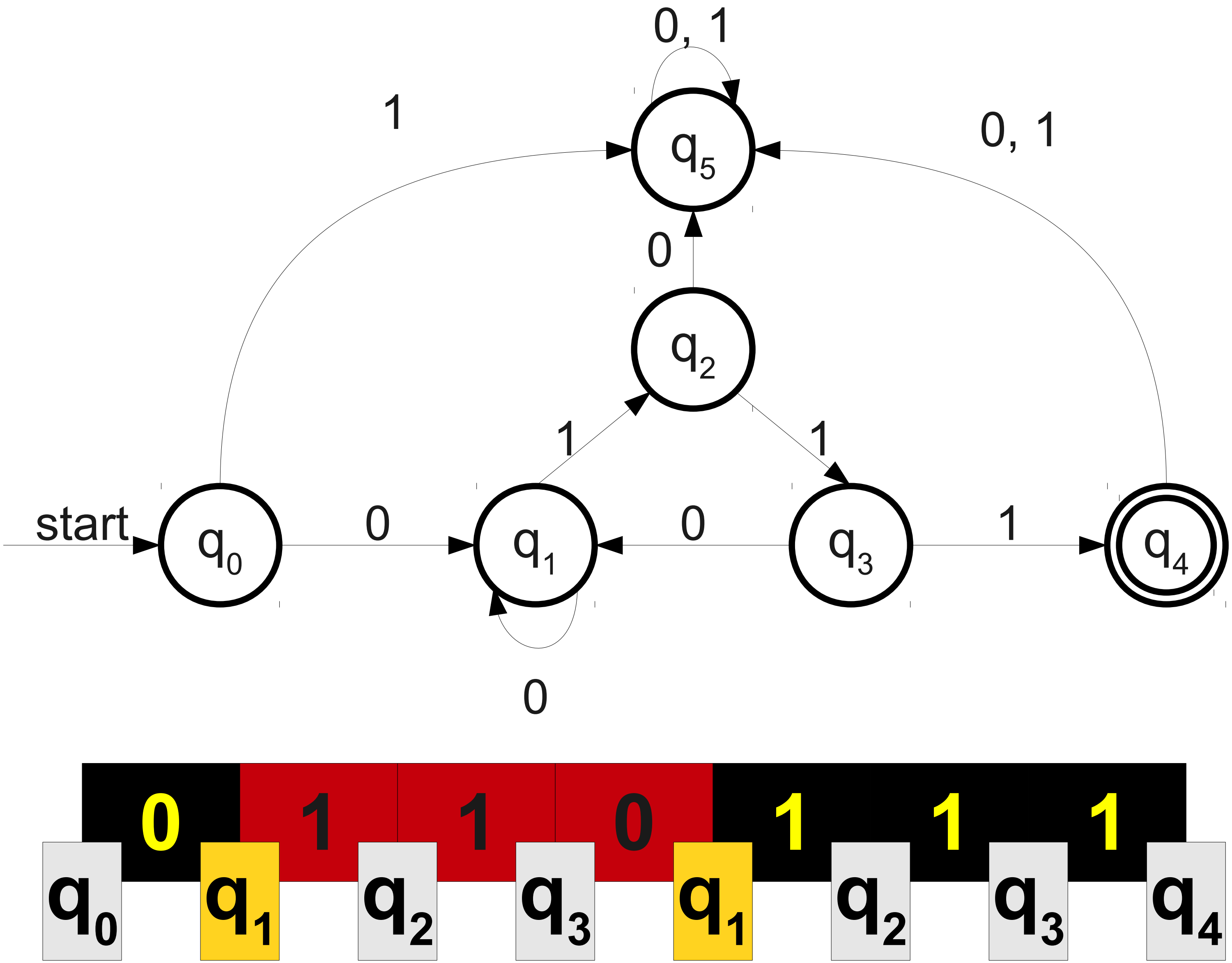
An Important Observation



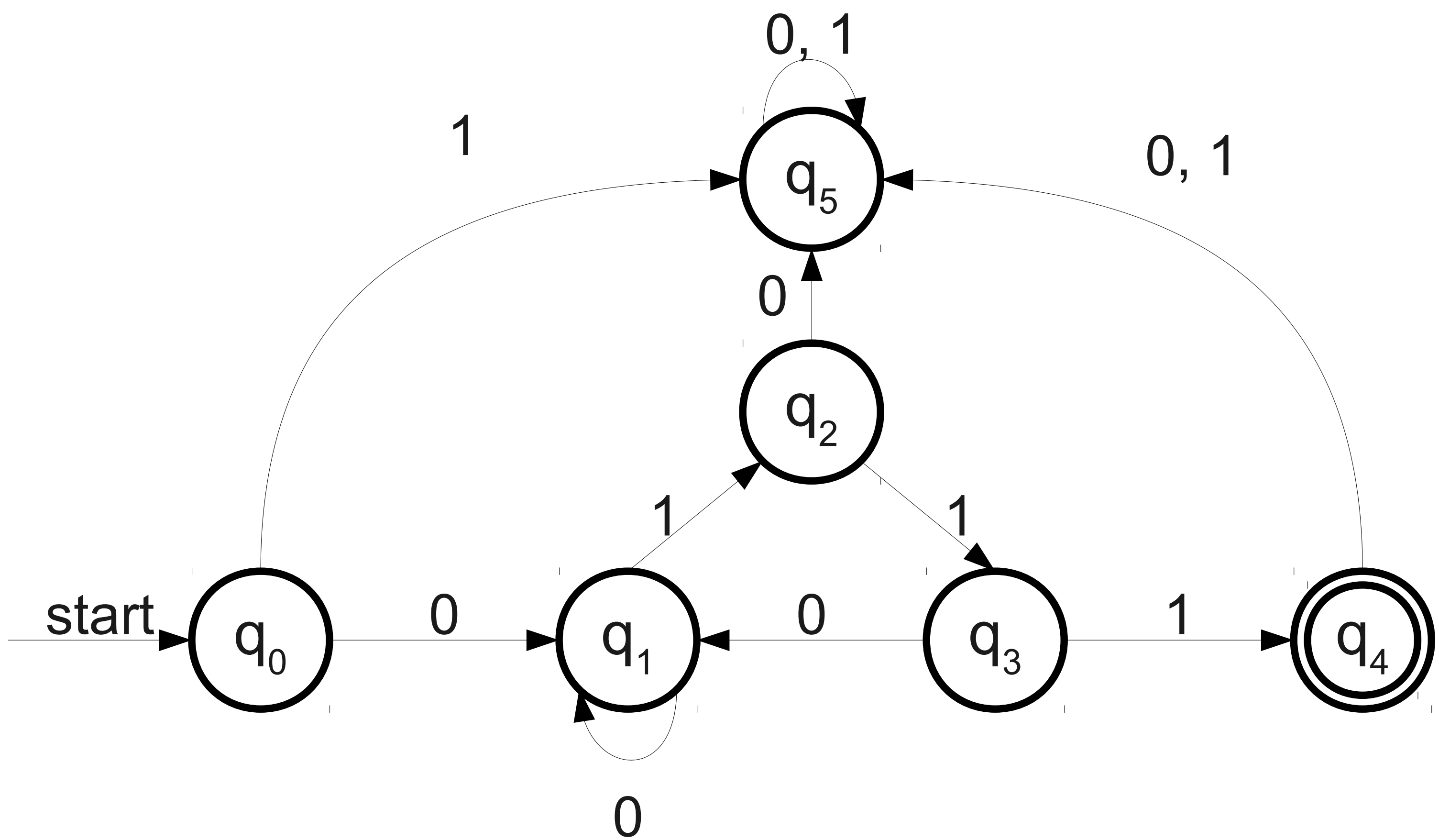
An Important Observation

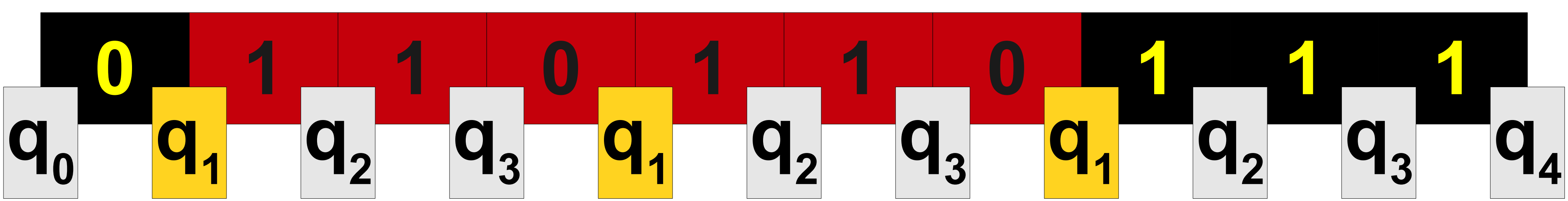


An Important Observation



An Important Observation

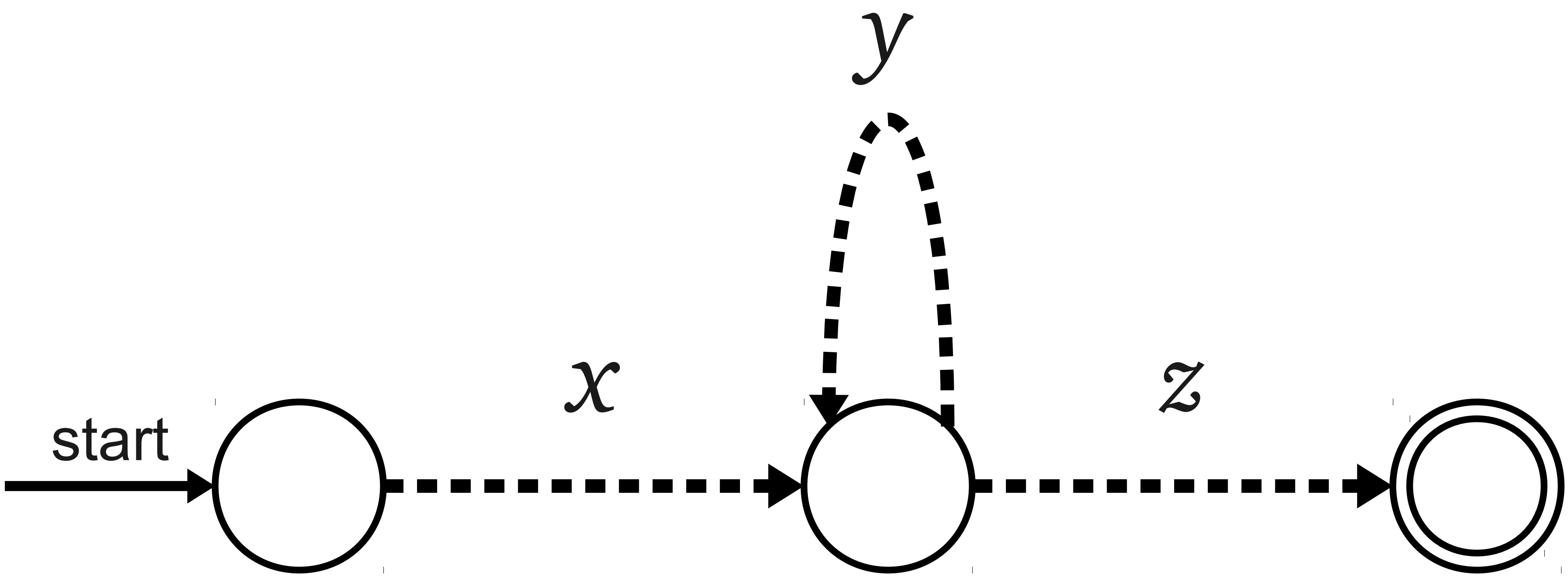




Visiting Multiple States

- Let D be a DFA with n states.
- Any string w accepted by D that has length at least n must visit some state twice.
  - Number of states visited is equal to the length of the string plus one.
  - By the pigeonhole principle, some state is duplicated.
- The substring of w between those revisited states can be removed, duplicated, tripled, etc. without changing the fact that D accepts w.

Intuitively



Informally

- Let L be a regular language.
- If we have a string $w \in L$ that is "sufficiently long," then we can split the string into three pieces and "pump" the middle.
- We can write $w = xyz$ such that xy^0z, xy^1z, xy^2z, …, xy^nz, … are all in L.
  - **Notation**: y^n means "n copies of y."

The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

  For any regular language L,

  There exists a positive natural number n such that

  For any $w \in L$ with $|w| \geq n$,

  There exists strings x, y, z such that

  For any natural number i,

  $w = xyz$,

  $y \neq \varepsilon$

  $xy^iz \in L$

The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

∀ regular language L,

∃ a positive natural number n such that

∀ $w \in L$ with $|w| \geq n$,

∃ strings x, y, z such that

∀ natural number i,

$w = xyz$,

$y \neq \varepsilon$

$xy^iz \in L$

The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

∀ regular language

∃ a positive nat

∀ $w \in L$ with

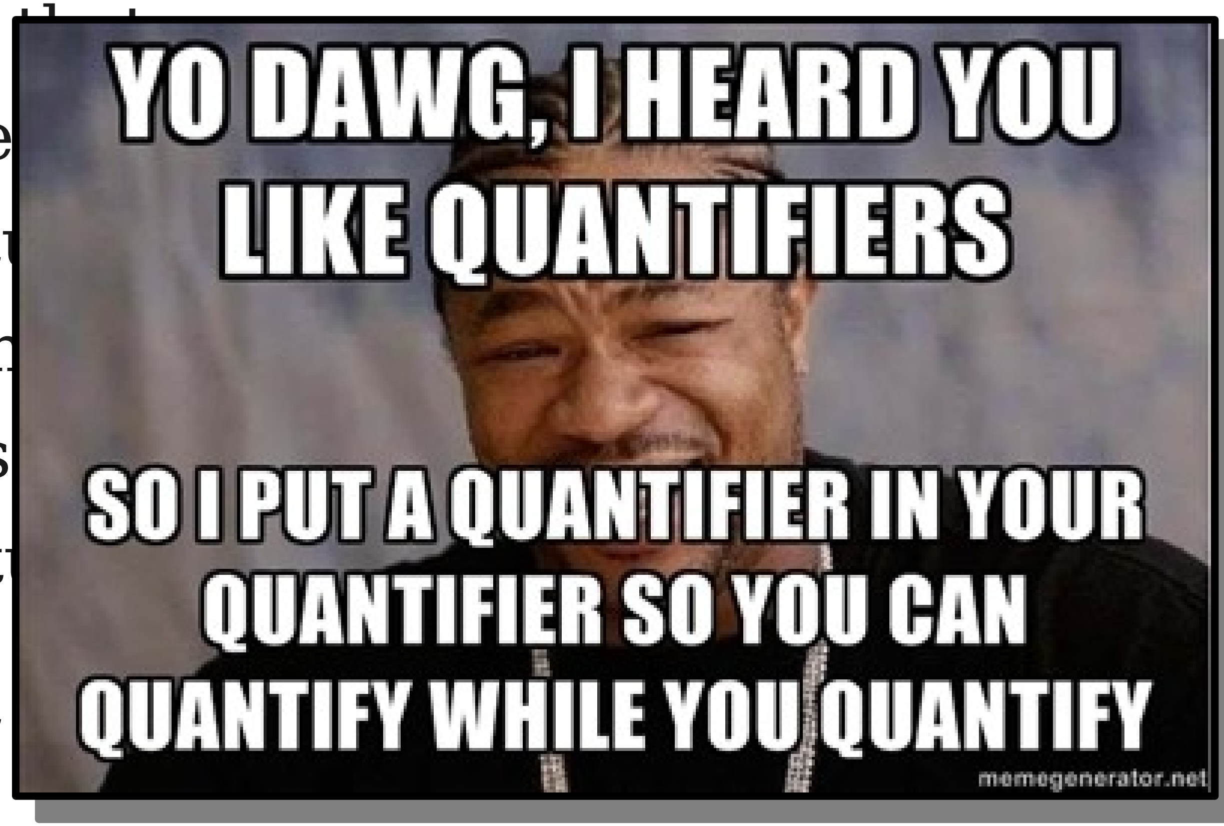
∃ strings

∀ nat

w

$y \neq \varepsilon$

$xy^iz \in L$



The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

For any regular language L,

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz$,

$y \neq \varepsilon$

$xy^iz \in L$

The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

For any regular language L,

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz$,

$y \neq \varepsilon$

$xy^iz \in L$

This number n is sometimes called the pumping length.

The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

For any regular language L,

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz,$

$y \neq \varepsilon$

$xy^iz \in L$

Strings longer than the pumping length must have a special property.

The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

For any regular language L,

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz$, w can be broken into three pieces,

$y \neq \varepsilon$

$xy^iz \in L$

The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

For any regular language L,

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz$, w can be broken into three pieces,

$y \neq \varepsilon$ where the middle piece isn't empty,

$xy^iz \in L$

The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

For any regular language L,

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz$, w can be broken into three pieces,

$y \neq \varepsilon$ where the middle piece isn't empty,

$xy^iz \in L$ where the middle piece can be replicated zero or more times.

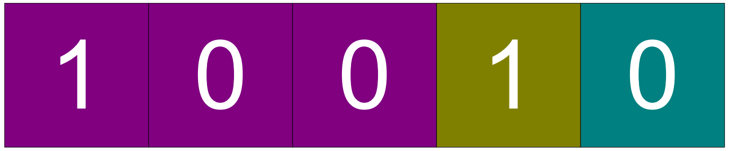
The Weak Pumping Lemma

- Let $\Sigma = \{$`0`$,$ `1`$\}$ and $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	0	0	1	0

The Weak Pumping Lemma

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$ and $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."



The Weak Pumping Lemma

- Let Σ = {`0`, `1`} and L = { w ∈ Σ* | w contains `00` as a substring. }
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	0	0	0

The Weak Pumping Lemma

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$ and $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	0	0	1	0

The Weak Pumping Lemma

- Let Σ = {**0**, **1**} and $L = \{ w \in \Sigma^* \mid w$ contains **00** as a substring. }
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

The Weak Pumping Lemma

- Let Σ = {0, 1} and *L* = { *w* ∈ Σ* | *w* contains 00 as a substring. }
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	0	0	1	1	1	0

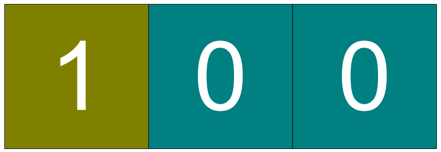
The Weak Pumping Lemma

- Let $\Sigma = \{0, 1\}$ and $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring. }
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	0	0

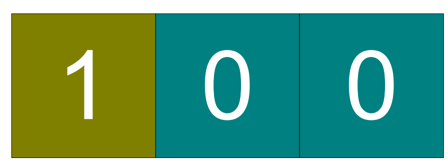
The Weak Pumping Lemma

- Let $\Sigma = \{0, 1\}$ and $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."



The Weak Pumping Lemma

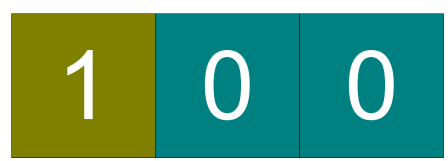
- Let $\Sigma = \{$0, 1$\}$ and $L = \{\ w \in \Sigma^* \mid w$ contains 00 as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be “pumped.”



The first piece is just the empty string! This is perfectly fine.

The Weak Pumping Lemma

- Let Σ = {`0`, `1`} and L = { $w \in \Sigma^*$ | w contains `00` as a substring. }
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."



The Weak Pumping Lemma

- Let $\Sigma = \{$`0`$,$ `1`$\}$ and $L = \{\ w \in \Sigma^* \mid w$ contains `00` as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

0	0

The Weak Pumping Lemma

- Let $\Sigma = \{$**0**, **1**$\}$ and $L = \{ w \in \Sigma^* \mid w$ contains **00** as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	0	0

The Weak Pumping Lemma

- Let $\Sigma = \{$`0`, `1`$\}$ and $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	1	0	0

The Weak Pumping Lemma

- Let $\Sigma = \{$`0`, `1`$\}$ and $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	1	1	0	0

The Weak Pumping Lemma

- Let $\Sigma = \{$`0`, `1`$\}$ and $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring. }
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

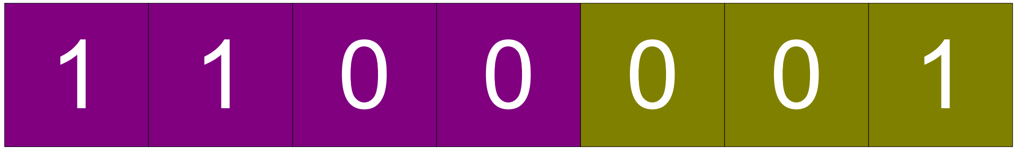
The Weak Pumping Lemma

- Let $\Sigma = \{0, 1\}$ and $L = \{\ w \in \Sigma^* \mid w$ contains `00` as a substring. }
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	1	0	0	0	0	1

The Weak Pumping Lemma

- Let Σ = {0, 1} and L = { w ∈ Σ* | w contains 00 as a substring. }
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."



The Weak Pumping Lemma

- Let $\Sigma = \{$`0`, `1`$\}$ and $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	1	0	0

The Weak Pumping Lemma

- Let Σ = {0, 1} and L = { w ∈ Σ* | w contains 00 as a substring. }
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	1	0	0	0	0	1

The Weak Pumping Lemma

- Let $\Sigma = \{$0, 1$\}$ and $L = \{ w \in \Sigma^* \mid w$ contains 00 as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	1	0	0	0	0	1	0	0	1

The Weak Pumping Lemma

- Let $\Sigma = \{\texttt{0}, \texttt{1}\}$ and $L = \{ w \in \Sigma^* \mid w$ contains `00` as a substring. $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

1	1	0	0	0	0	1	0	0	1	0	0	1

The Weak Pumping Lemma

The Weak Pumping Lemma

- Let Σ = {`0`, `1`} and
  L = { ε, `0`, `1`, `00`, `01`, `10`, `11` }

The Weak Pumping Lemma

- Let $\Sigma = \{$`0`, `1`$\}$ and
  $L = \{ \varepsilon,$ `0`, `1`, `00`, `01`, `10`, `11` $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

The Weak Pumping Lemma

- Let $\Sigma = \{$`0`, `1`$\}$ and
  $L = \{ \varepsilon,$ `0`, `1`, `00`, `01`, `10`, `11` $\}$
- Any string of length 3 or greater can be split into three pieces, the second of which can be "pumped."

The weak pumping lemma holds for finite languages because the pumping length can be longer than the longest string!

Testing Equality

- The **equality problem** is defined as follows:

  Given two strings x and y, decide if x = y.

- Let $\Sigma = \{$`0`, `1`, `?`$\}$. We can encode the equality problem as a string of the form x`?`y.
  - "Is `001` equal to `110` ?" would be `001?110`
  - "Is `11` equal to `11` ?" would be `11?11`
  - "Is `110` equal to `110` ?" would be `110?110`
- Let $EQUAL = \{\ w\texttt{?}w \mid w \in \{\texttt{0}, \texttt{1}\}^*\ \}$
- **Question**: Is $EQUAL$ a regular language?

The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

For any regular language L,

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz$, w can be broken into three pieces,

$y \neq \varepsilon$ where the middle piece isn't empty,

$xy^iz \in L$ where the middle piece can be replicated zero or more times.

Using the Weak Pumping Lemma

$$EQUAL = \{\ w\texttt{?}w \mid w \in \{\texttt{0}, \texttt{1}\}^*\ \}$$

0	0	0	?	0	0	0

Using the Weak Pumping Lemma

$$EQUAL = \{\ w\texttt{?}w \mid w \in \{\texttt{0}, \texttt{1}\}^*\ \}$$

0	0	0	?	0	0	0

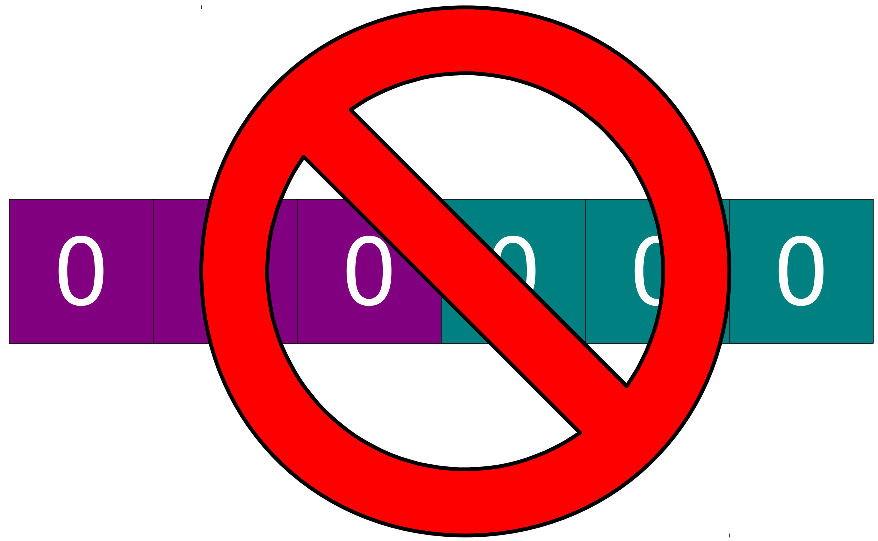
Using the Weak Pumping Lemma

$$EQUAL = \{\ w\mathtt{?}w \mid w \in \{\mathtt{0}, \mathtt{1}\}^*\ \}$$

0	0	0	0	0	0

Using the Weak Pumping Lemma

$$EQUAL = \{\ w\texttt{?}w \mid w \in \{\texttt{0}, \texttt{1}\}^*\ \}$$



Using the Weak Pumping Lemma

$$EQUAL = \{\, w\mathbf{?}w \mid w \in \{\mathbf{0}, \mathbf{1}\}^* \,\}$$

0	0	0	?	0	0	0

Using the Weak Pumping Lemma

$$EQUAL = \{\ w\texttt{?}w \mid w \in \{\texttt{0}, \texttt{1}\}^*\ \}$$

0	0	0	?	0	0	0

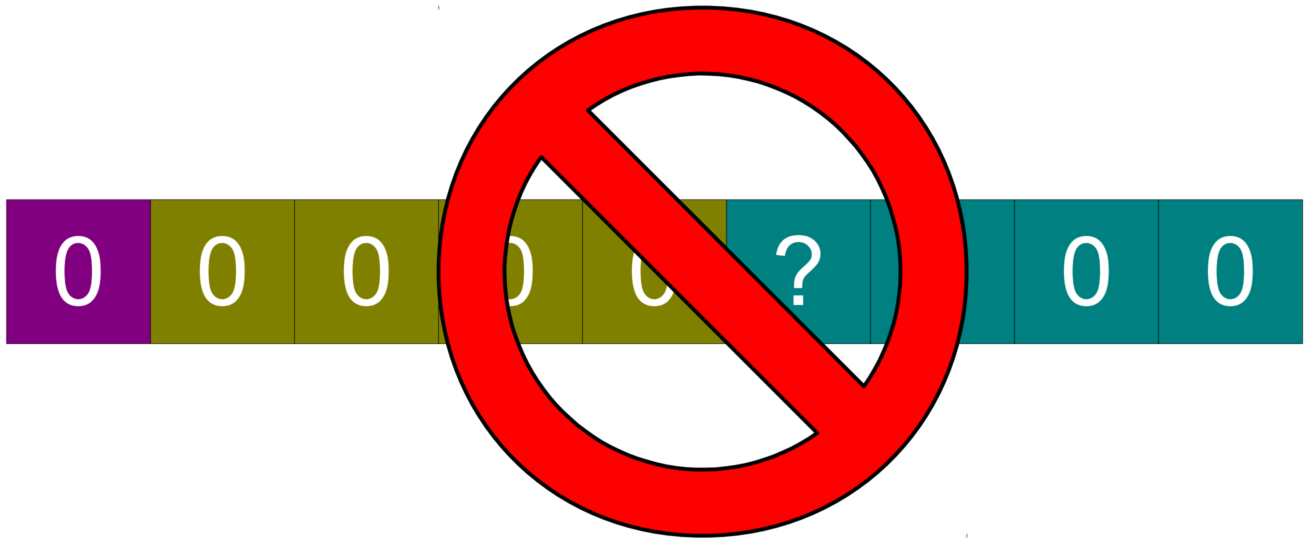
Using the Weak Pumping Lemma

$$EQUAL = \{\ w\texttt{?}w \mid w \in \{\texttt{0}, \texttt{1}\}^*\ \}$$

0	0	0	0	0	?	0	0	0

Using the Weak Pumping Lemma

$$EQUAL = \{\ w\texttt{?}w \mid w \in \{\texttt{0}, \texttt{1}\}^*\ \}$$



Using the Weak Pumping Lemma

$$EQUAL = \{\ w\mathtt{?}w \mid w \in \{\mathtt{0}, \mathtt{1}\}^*\ \}$$

0	0	0	?	0	0	0

Using the Weak Pumping Lemma

$$EQUAL = \{\ w\texttt{?}w \mid w \in \{\texttt{0}, \texttt{1}\}^*\ \}$$

0	0	0	?	0	0	0

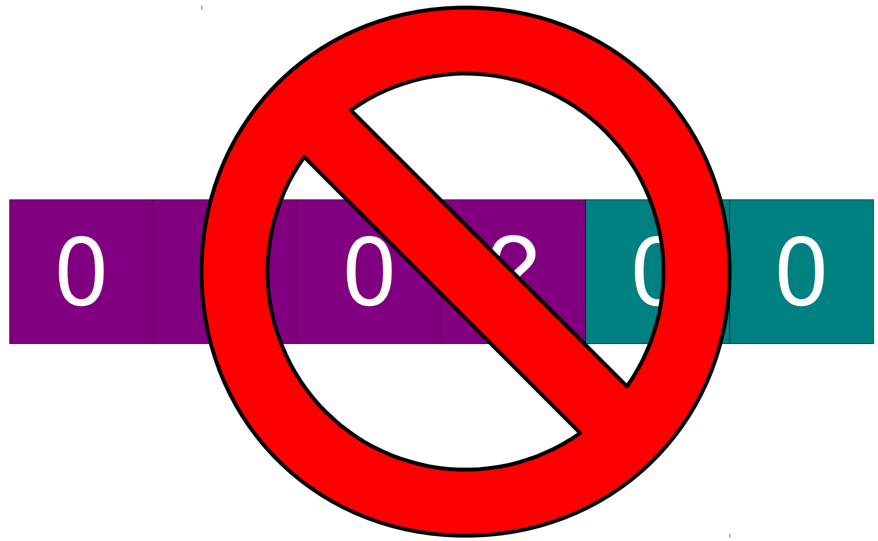
Using the Weak Pumping Lemma

$$EQUAL = \{\ w\texttt{?}w \mid w \in \{\texttt{0}, \texttt{1}\}^*\ \}$$

0	0	0	?	0	0

Using the Weak Pumping Lemma

$$EQUAL = \{ w\texttt{?}w \mid w \in \{\texttt{0}, \texttt{1}\}^* \}$$



What's Going On?

- The weak pumping lemma says that for "sufficiently long" strings, we should be able to pump some part of the string.
- We can't pump any part containing the `?`, because we can't duplicate or remove it.
- We can't pump just one part of the string, because then the strings on opposite sides of the `?` wouldn't match.
- **Can we formally show that *EQUAL* is not regular?**

For any regular language L,
There exists a positive natural number n such that
For any $w \in L$ with $|w| \geq n$,
There exists strings x, y, z such that
For any natural number i,
$w = xyz$,
$y \neq \varepsilon$
$xy^iz \in L$

Theorem: EQUAL is not regular.

For any regular language L,
There exists a positive natural number n such that
For any $w \in L$ with $|w| \geq n$,
There exists strings x, y, z such that
For any natural number i,
$w = xyz$,
$y \neq \varepsilon$
$xy^iz \in L$

Theorem: EQUAL is not regular.
Proof: By contradiction; assume that *EQUAL* is regular.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
xy^iz ∈ L

Theorem: EQUAL is not regular.
Proof: By contradiction; assume that *EQUAL* is regular.

For any regular language L,

  There exists a positive natural number n such that

    For any $w \in L$ with $|w| \geq n$,

      There exists strings x, y, z such that

        For any natural number i,

          $w = xyz$,

          $y \neq \varepsilon$

          $xy^iz \in L$

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let *n* be the pumping length guaranteed by the weak pumping lemma.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
$xy^iz \in L$

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let *n* be the pumping length guaranteed by the weak pumping lemma.

For any regular language L,
There exists a positive natural number n such that
For any $w \in L$ with $|w| \geq n$,
There exists strings x, y, z such that
For any natural number i,
$w = xyz$,
$y \neq \varepsilon$
$xy^iz \in L$

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let *n* be the pumping length guaranteed by the weak pumping lemma.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
$xy^iz \in L$

Theorem: EQUAL is not regular.
Proof: By contradiction; assume that *EQUAL* is regular. Let *n* be the pumping length guaranteed by the weak pumping lemma.

The hardest part of most proofs with the pumping lemma is choosing some string that we should be able to pump but cannot.

For any regular language L,

There exists a positive natural number n such that

For any w ∈ L with |w| ≥ n,

There exists strings x, y, z such that

For any natural number i,

w = xyz,

y ≠ ε

$xy^iz \in L$

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let *n* be the pumping length guaranteed by the weak pumping lemma. Let $w = 0^n?0^n$.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
xy^iz ∈ L

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let *n* be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$.

For any regular language L,

There exists a positive natural number n such that

For any w ∈ L with |w| ≥ n,

There exists strings x, y, z such that

For any natural number i,

w = xyz,

y ≠ ε

xy^iz ∈ L

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let *n* be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$.

For any regular language L,
There exists a positive natural number n such that
For any $w \in L$ with $|w| \geq n$,
There exists strings x, y, z such that
For any natural number i,
$w = xyz$,
$y \neq \varepsilon$
$xy^iz \in L$

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let $w = \mathtt{0}^n\mathtt{?}\mathtt{0}^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
xy^iz ∈ L

Theorem: EQUAL is not regular.
Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
xy^iz ∈ L

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$.

For any regular language L,

There exists a positive natural number n such that

For any w ∈ L with |w| ≥ n,

There exists strings x, y, z such that

For any natural number i,

w = xyz,

y ≠ ε

xy^iz ∈ L

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$.

For any regular language L,

There exists a positive natural number n such that

For any w ∈ L with |w| ≥ n,

There exists strings x, y, z such that

For any natural number i,

w = xyz,

y ≠ ε

$xy^iz \in L$

Theorem: *EQUAL* is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$.

At this point, we have some string that we should be able to split into pieces and pump. The rest of the proof shows that no matter what choice we made, the middle can't be pumped.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
$xy^iz \in L$

Theorem: EQUAL is not regular.
Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$.

For any regular language L,

There exists a positive natural number n such that

For any w ∈ L with |w| ≥ n,

There exists strings x, y, z such that

For any natural number i,

$w = xyz$,

$y \neq \varepsilon$

$xy^iz \in L$

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. Then y cannot contain ?, since otherwise if we let $i = 0$, then $xy^iz = xz$ does not contain ? and would not be in *EQUAL*.

For any regular language L,
There exists a positive natural number n such that
For any $w \in L$ with $|w| \geq n$,
There exists strings x, y, z such that
For any natural number i,
$w = xyz$,
$y \neq \varepsilon$
$xy^iz \in L$

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. Then y cannot contain ?, since otherwise if we let $i = 0$, then $xy^iz = xz$ does not contain ? and would not be in *EQUAL*. So y is either completely to the left of the ? or completely to the right of the ?.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
xy^iz ∈ L

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let *n* be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. Then y cannot contain ?, since otherwise if we let $i = 0$, then $xy^iz = xz$ does not contain ? and would not be in *EQUAL*. So y is either completely to the left of the ? or completely to the right of the ?. Let $|y| = k$, so $k > 0$.

For any regular language L,

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz$,

$y \neq \varepsilon$

$xy^iz \in L$

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. Then y cannot contain ?, since otherwise if we let $i = 0$, then $xy^iz = xz$ does not contain ? and would not be in *EQUAL*. So y is either completely to the left of the ? or completely to the right of the ?. Let $|y| = k$, so $k > 0$. Since y is completely to the left or right of the ?, then $y = 0^k$.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
$xy^iz \in L$

Theorem: EQUAL is not regular.
Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. Then y cannot contain ?, since otherwise if we let $i = 0$, then $xy^iz = xz$ does not contain ? and would not be in *EQUAL*. So y is either completely to the left of the ? or completely to the right of the ?. Let $|y| = k$, so $k > 0$. Since y is completely to the left or right of the ?, then y = 0^k. Now, we consider two cases:

Case 1: y is to the left of the ?.

Case 2: y is to the right of the ?.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
$xy^iz \in L$

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. Then y cannot contain **?**, since otherwise if we let $i = 0$, then $xy^iz = xz$ does not contain **?** and would not be in *EQUAL*. So y is either completely to the left of the **?** or completely to the right of the **?**. Let $|y| = k$, so $k > 0$. Since y is completely to the left or right of the **?**, then $y = 0^k$. Now, we consider two cases:

Case 1: y is to the left of the **?**. Then $xy^2z = 0^{n+k}?0^n \notin EQUAL$, contradicting the weak pumping lemma.

Case 2: y is to the right of the **?**.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
$xy^iz \in L$

Theorem: *EQUAL* is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in$ *EQUAL* and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in$ *EQUAL*. Then y cannot contain ?, since otherwise if we let $i = 0$, then $xy^iz = xz$ does not contain ? and would not be in *EQUAL*. So y is either completely to the left of the ? or completely to the right of the ?. Let $|y| = k$, so $k > 0$. Since y is completely to the left or right of the ?, then $y = 0^k$. Now, we consider two cases:

Case 1: y is to the left of the ?. Then $xy^2z = 0^{n+k}?0^n \notin$ *EQUAL*, contradicting the weak pumping lemma.

Case 2: y is to the right of the ?. Then $xy^2z = 0^n?0^{n+k} \notin$ *EQUAL*, contradicting the weak pumping lemma.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
$xy^iz \in L$

Theorem: EQUAL is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let $w = 0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. Then y cannot contain ?, since otherwise if we let $i = 0$, then $xy^iz = xz$ does not contain ? and would not be in *EQUAL*. So y is either completely to the left of the ? or completely to the right of the ?. Let $|y| = k$, so $k > 0$. Since y is completely to the left or right of the ?, then $y = 0^k$. Now, we consider two cases:

Case 1: y is to the left of the ?. Then $xy^2z = 0^{n+k}?0^n \notin EQUAL$, contradicting the weak pumping lemma.

Case 2: y is to the right of the ?. Then $xy^2z = 0^n?0^{n+k} \notin EQUAL$, contradicting the weak pumping lemma.

In either case we reach a contradiction, so our assumption was wrong.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
$xy^iz \in L$

Theorem: *EQUAL* is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let w = $0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. Then y cannot contain ?, since otherwise if we let $i = 0$, then $xy^iz = xz$ does not contain ? and would not be in *EQUAL*. So y is either completely to the left of the ? or completely to the right of the ?. Let $|y| = k$, so $k > 0$. Since y is completely to the left or right of the ?, then $y = 0^k$. Now, we consider two cases:

Case 1: y is to the left of the ?. Then $xy^2z = 0^{n+k}?0^n \notin EQUAL$, contradicting the weak pumping lemma.

Case 2: y is to the right of the ?. Then $xy^2z = 0^n?0^{n+k} \notin EQUAL$, contradicting the weak pumping lemma.

In either case we reach a contradiction, so our assumption was wrong. Thus *EQUAL* is not regular.

For any regular language L,
There exists a positive natural number n such that
For any w ∈ L with |w| ≥ n,
There exists strings x, y, z such that
For any natural number i,
w = xyz,
y ≠ ε
$xy^iz \in L$

Theorem: *EQUAL* is not regular.

Proof: By contradiction; assume that *EQUAL* is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Let $w = 0^n?0^n$. Then $w \in EQUAL$ and $|w| = 2n + 1 \geq n$. Thus by the weak pumping lemma, we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, $xy^iz \in EQUAL$. Then y cannot contain ?, since otherwise if we let $i = 0$, then $xy^iz = xz$ does not contain ? and would not be in *EQUAL*. So y is either completely to the left of the ? or completely to the right of the ?. Let $|y| = k$, so $k > 0$. Since y is completely to the left or right of the ?, then $y = 0^k$. Now, we consider two cases:

Case 1: y is to the left of the ?. Then $xy^2z = 0^{n+k}?0^n \notin EQUAL$, contradicting the weak pumping lemma.

Case 2: y is to the right of the ?. Then $xy^2z = 0^n?0^{n+k} \notin EQUAL$, contradicting the weak pumping lemma.

In either case we reach a contradiction, so our assumption was wrong. Thus *EQUAL* is not regular. ■

Nonregular Languages

- The weak pumping lemma describes a property common to all regular languages.
- Any language L which does not have this property *cannot be regular*.
- What other languages can we find that are not regular?

A Canonical Nonregular Language

- Consider the language L = $\{ \texttt{0}^n\texttt{1}^n \mid n \in \mathbb{N} \}$.

  $$L = \{ \varepsilon, \texttt{01}, \texttt{0011}, \texttt{000111}, \texttt{00001111}, \dots \}$$

- L is a classic example of a nonregular language.
- Intuitively: If you have only finitely many states in a DFA, you can't "remember" an arbitrary number of `0`s.
- How would we prove that L is nonregular?

The Pumping Lemma as a Game

- The weak pumping lemma can be thought of as a game between **you** and an **adversary**.
- **You win** if you can prove that the pumping lemma fails.
- **The adversary wins** if the adversary can make a choice for which the pumping lemma succeeds.
- The game goes as follows:
  - **The adversary** chooses a pumping length n.
  - **You** choose a string w with $|w| \geq n$ and $w \in L$.
  - **The adversary** breaks it into x, y, and z.
  - **You** choose an i such that $xy^iz \notin L$ (if you can't, you lose!)

The Pumping Lemma Game

ADVERSARY	YOU

The Pumping Lemma Game

ADVERSARY	YOU
Maliciously choose pumping length n.	

The Pumping Lemma Game

ADVERSARY	**YOU**
Maliciously choose pumping length n.	Cleverly choose a string $w \in L$, $\|w\| \geq n$

The Pumping Lemma Game

ADVERSARY	YOU
Maliciously choose pumping length n.	
	Cleverly choose a string $w \in L$, $\|w\| \geq n$
Maliciously split $w = xyz$, $y \neq \varepsilon$	

The Pumping Lemma Game

ADVERSARY	YOU
Maliciously choose pumping length n.	
	Cleverly choose a string $w \in L$, $\lvert w \rvert \geq n$
Maliciously split $w = xyz$, $y \neq \varepsilon$	
	Cleverly choose i such that $xy^iz \notin L$

The Pumping Lemma Game

ADVERSARY	YOU
Maliciously choose pumping length n.	
	Cleverly choose a string $w \in L$, $\lvert w \rvert \geq n$
Maliciously split $w = xyz$, $y \neq \varepsilon$	
	Cleverly choose i such that $xy^iz \notin L$
Grrr! Aaaargh!	

The Pumping Lemma Game

$$L = \{\ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N}\ \}$$

ADVERSARY	YOU
Maliciously choose pumping length n.	
	Cleverly choose a string $w \in L$, $\lvert w\rvert \geq n$
Maliciously split $w = xyz$, $y \neq \varepsilon$	
	Cleverly choose i such that $xy^iz \notin L$
Grrr! Aaaargh!	

The Pumping Lemma Game

$L = \{ 0^n1^n \mid n \in \mathbb{N} \}$

ADVERSARY	YOU
Maliciously choose pumping length n.	
	Cleverly choose a string $w \in L$, $\lvert w \rvert \geq n$
Maliciously split $w = xyz$, $y \neq \varepsilon$	
	Cleverly choose i such that $xy^iz \notin L$
Grrr! Aaaargh!	

Try your best!

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.
Proof: By contradiction; assume L is regular.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of $\mathtt{0}$s.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of $\mathtt{0}$s. Then $xy^0z = xz = \mathtt{0}^{n - |y|}\mathtt{1}^n$, and since $|y| > 0$, $xz \notin L$.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of $\mathtt{0}$s. Then $xy^0z = xz = \mathtt{0}^{n - |y|}\mathtt{1}^n$, and since $|y| > 0$, $xz \notin L$.

Case 2: y consists solely of $\mathtt{1}$s.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of $\mathtt{0}$s. Then $xy^0z = xz = \mathtt{0}^{n - |y|}\mathtt{1}^n$, and since $|y| > 0$, $xz \notin L$.

Case 2: y consists solely of $\mathtt{1}$s. Then $xy^0z = xz = \mathtt{0}^n\mathtt{1}^{n - |y|}$, and since $|y| > 0$, $xz \notin L$.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of $\mathtt{0}$s. Then $xy^0z = xz = \mathtt{0}^{n - |y|}\mathtt{1}^n$, and since $|y| > 0$, $xz \notin L$.

Case 2: y consists solely of $\mathtt{1}$s. Then $xy^0z = xz = \mathtt{0}^n\mathtt{1}^{n - |y|}$, and since $|y| > 0$, $xz \notin L$.

Case 3: y consists of $k > 0$ $\mathtt{0}$s followed by $m > 0$ $\mathtt{1}$s.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of $\mathtt{0}$s. Then $xy^0z = xz = \mathtt{0}^{n - |y|}\mathtt{1}^n$, and since $|y| > 0$, $xz \notin L$.

Case 2: y consists solely of $\mathtt{1}$s. Then $xy^0z = xz = \mathtt{0}^n\mathtt{1}^{n - |y|}$, and since $|y| > 0$, $xz \notin L$.

Case 3: y consists of $k > 0$ $\mathtt{0}$s followed by $m > 0$ $\mathtt{1}$s.

0	0	0	0	0	0	1	1	1	1	1	1

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of $\mathtt{0}$s. Then $xy^0z = xz = \mathtt{0}^{n - |y|}\mathtt{1}^n$, and since $|y| > 0$, $xz \notin L$.

Case 2: y consists solely of $\mathtt{1}$s. Then $xy^0z = xz = \mathtt{0}^n\mathtt{1}^{n - |y|}$, and since $|y| > 0$, $xz \notin L$.

Case 3: y consists of $k > 0$ $\mathtt{0}$s followed by $m > 0$ $\mathtt{1}$s.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of $\mathtt{0}$s. Then $xy^0z = xz = \mathtt{0}^{n - |y|}\mathtt{1}^n$, and since $|y| > 0$, $xz \notin L$.

Case 2: y consists solely of $\mathtt{1}$s. Then $xy^0z = xz = \mathtt{0}^n\mathtt{1}^{n - |y|}$, and since $|y| > 0$, $xz \notin L$.

Case 3: y consists of $k > 0$ $\mathtt{0}$s followed by $m > 0$ $\mathtt{1}$s.

0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1

Theorem: $L = \{ \texttt{0}^n\texttt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \texttt{0}^n\texttt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of 0s. Then $xy^0z = xz = \texttt{0}^{n - |y|}\texttt{1}^n$, and since $|y| > 0$, $xz \notin L$.

Case 2: y consists solely of 1s. Then $xy^0z = xz = \texttt{0}^n\texttt{1}^{n - |y|}$, and since $|y| > 0$, $xz \notin L$.

Case 3: y consists of $k > 0$ 0s followed by $m > 0$ 1s.

0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of $\mathtt{0}$s. Then $xy^0z = xz = \mathtt{0}^{n - |y|}\mathtt{1}^n$, and since $|y| > 0$, $xz \notin L$.

Case 2: y consists solely of $\mathtt{1}$s. Then $xy^0z = xz = \mathtt{0}^n\mathtt{1}^{n - |y|}$, and since $|y| > 0$, $xz \notin L$.

Case 3: y consists of $k > 0$ $\mathtt{0}$s followed by $m > 0$ $\mathtt{1}$s. Then xy^2z has the form $\mathtt{0}^n\mathtt{1}^m\mathtt{0}^k\mathtt{1}^n$, so $xy^2z \notin L$.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of $\mathtt{0}$s. Then $xy^0z = xz = \mathtt{0}^{n - |y|}\mathtt{1}^n$, and since $|y| > 0$, $xz \notin L$.

Case 2: y consists solely of $\mathtt{1}$s. Then $xy^0z = xz = \mathtt{0}^n\mathtt{1}^{n - |y|}$, and since $|y| > 0$, $xz \notin L$.

Case 3: y consists of $k > 0$ $\mathtt{0}$s followed by $m > 0$ $\mathtt{1}$s. Then xy^2z has the form $\mathtt{0}^n\mathtt{1}^m\mathtt{0}^k\mathtt{1}^n$, so $xy^2z \notin L$.

In all three cases we reach a contradiction, so our assumption was wrong and L is not regular.

Theorem: $L = \{ \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \}$ is not regular.

Proof: By contradiction; assume L is regular. Let n be the pumping length guaranteed by the weak pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in L$, so we can write $w = xyz$ such that $y \neq \varepsilon$ and for any $i \in \mathbb{N}$, we have $xy^iz \in L$. We consider three cases:

Case 1: y consists solely of $\mathtt{0}$s. Then $xy^0z = xz = \mathtt{0}^{n - |y|}\mathtt{1}^n$, and since $|y| > 0$, $xz \notin L$.

Case 2: y consists solely of $\mathtt{1}$s. Then $xy^0z = xz = \mathtt{0}^n\mathtt{1}^{n - |y|}$, and since $|y| > 0$, $xz \notin L$.

Case 3: y consists of $k > 0$ $\mathtt{0}$s followed by $m > 0$ $\mathtt{1}$s. Then xy^2z has the form $\mathtt{0}^n\mathtt{1}^m\mathtt{0}^k\mathtt{1}^n$, so $xy^2z \notin L$.

In all three cases we reach a contradiction, so our assumption was wrong and L is not regular. ■

Counting Symbols

- Consider the alphabet $\Sigma = \{$ `0`, `1` $\}$ and the language

$$BALANCE = \{ w \in \Sigma^* \mid w \text{ contains an equal number of } \texttt{0}\text{s and } \texttt{1}\text{s.} \}$$

- For example:
  - `01` $\in BALANCE$
  - `110010` $\in BALANCE$
  - `11011` $\notin BALANCE$
- **Question:** Is *BALANCE* a regular language?

BALANCE and the Weak Pumping Lemma

BALANCE = { *w* ∈ {`0`, `1`}*| *w* contains an equal number of `0`s and `1`s. }

1	1	0	0	0	1

BALANCE and the Weak Pumping Lemma

BALANCE = { w ∈ {0, 1}*| w contains an equal number of 0s and 1s. }

1	1	0	0	0	1

BALANCE and the Weak Pumping Lemma

BALANCE = { w ∈ {`0`, `1`}*| w contains an equal number of `0`s and `1`s. }

1	0	0	1

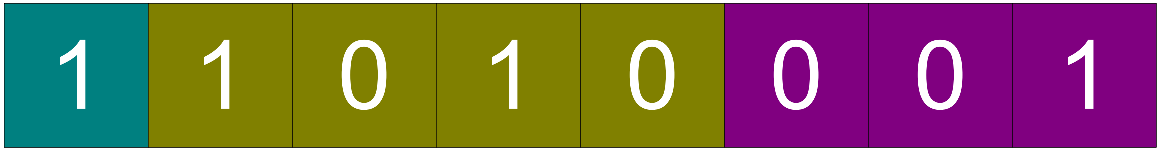
BALANCE and the Weak Pumping Lemma

BALANCE = { $w \in \{0, 1\}^*$| w contains an equal number of 0s and 1s. }

1	1	0	0	0	1

BALANCE and the Weak Pumping Lemma

BALANCE = { w ∈ {0, 1}*| w contains an equal number of 0s and 1s. }



BALANCE and the Weak Pumping Lemma

BALANCE = { $w \in \{\texttt{0}, \texttt{1}\}^*$ | w contains an equal number of `0`s and `1`s. }

1	1	0	1	0	1	0	0	0	1

BALANCE and the Weak Pumping Lemma

BALANCE = { $w \in \{0, 1\}^*$| w contains an equal number of 0s and 1s. }

BALANCE and the Weak Pumping Lemma

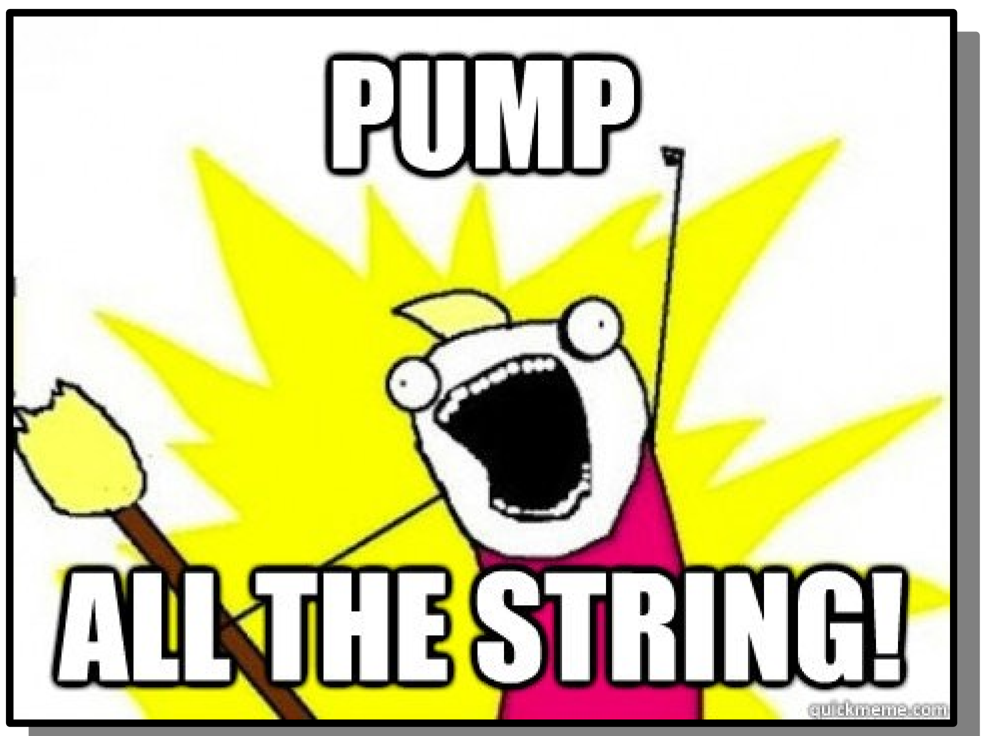
BALANCE = { $w \in \{0, 1\}^*$ | w contains an equal number of 0s and 1s. }

1	0	0	1

BALANCE and the Weak Pumping Lemma

BALANCE = { w ∈ {0, 1}*| w contains an equal number of 0s and 1s. }

1	0	0	1



BALANCE and the Weak Pumping Lemma

BALANCE = { w ∈ {0, 1}*| w contains an equal number of 0s and 1s. }

1	0	0	1

BALANCE and the Weak Pumping Lemma

BALANCE = { $w \in \{\mathtt{0}, \mathtt{1}\}^*$| w contains an equal number of `0`s and `1`s. }

BALANCE and the Weak Pumping Lemma

BALANCE = { $w \in \{0, 1\}^*$ | w contains an equal number of 0s and 1s. }

1	0	0	1

BALANCE and the Weak Pumping Lemma

BALANCE = { $w \in \{0, 1\}^*$| w contains an equal number of 0s and 1s. }

1	0	0	1	1	0	0	1

BALANCE and the Weak Pumping Lemma

BALANCE = { *w* ∈ {`0`, `1`}*| *w* contains an equal number of `0`s and `1`s. }

1	0	0	1	1	0	0	1	1	0	0	1

An Incorrect Proof

Theorem: BALANCE is regular.

Proof: We show that *BALANCE* satisfies the condition of the pumping lemma. Let $n = 2$ and consider any string $w \in BALANCE$ such that $|w| \geq 2$. Then we can write $w = xyz$ such that $x = z = \varepsilon$ and $y = w$, so $y \neq \varepsilon$. Then for any natural number i, $xy^iz = w^i$, which has the same number of `0`s and `1`s. Since *BALANCE* passes the conditions of the weak pumping lemma, *BALANCE* is regular. ■

An Incorrect Proof

Theorem: BAL

Proof: We sho sfies the condition o Let $n = 2$ and conside *ANCE* such that $|w| \geq 2$ $w = xyz$ such that $x = z =$. Then for any natural which has the same numb *BALANCE* passes the pumping lemma, *BA*

The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

For any regular language L,

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz$, w can be broken into three pieces,

$y \neq \varepsilon$ where the middle piece isn't empty,

$xy^iz \in L$ where the middle piece can be replicated zero or more times.

The Weak Pumping Lemma

- The **Weak Pumping Lemma for Regular Languages** states that

For any regular language L,

This says nothing about languages that aren't regular!

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz$, w can be broken into three pieces,

$y \neq \varepsilon$ where the middle piece isn't empty,

$xy^iz \in L$ where the middle piece can be replicated zero or more times.

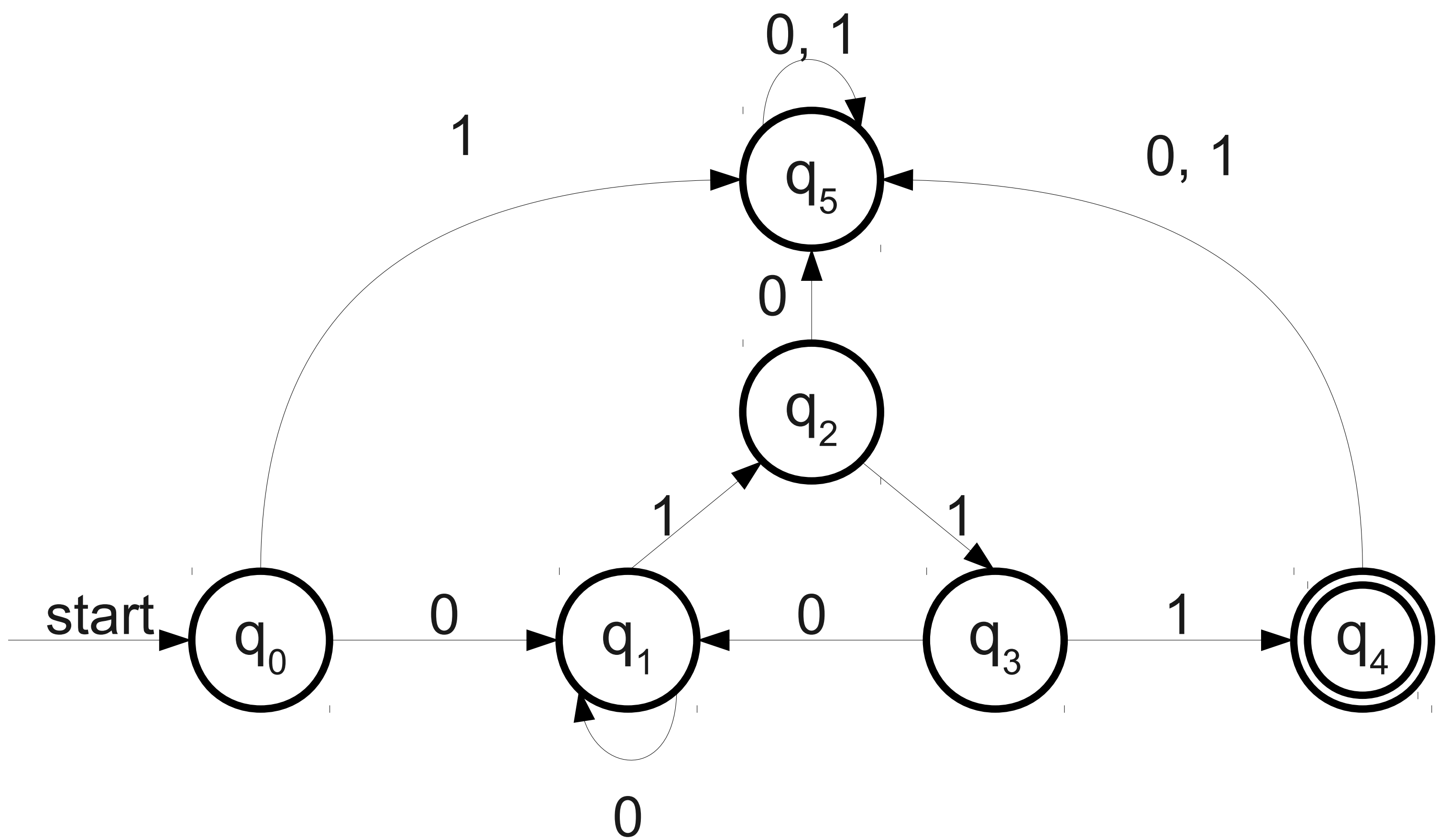
Caution with the Pumping Lemma

- The weak and full pumping lemmas describe a **necessary** condition of regular languages.
  - If L is regular, L passes the conditions of the pumping lemma.
- The weak and full pumping lemmas are not a **sufficient** condition of regular languages.
  - If L is *not* regular, it still might pass the conditions of the pumping lemma!
- If a language fails the pumping lemma, it is definitely not regular.
- If a language passes the pumping lemma, we learn nothing about whether it is regular or not.

BALANCE is Not Regular

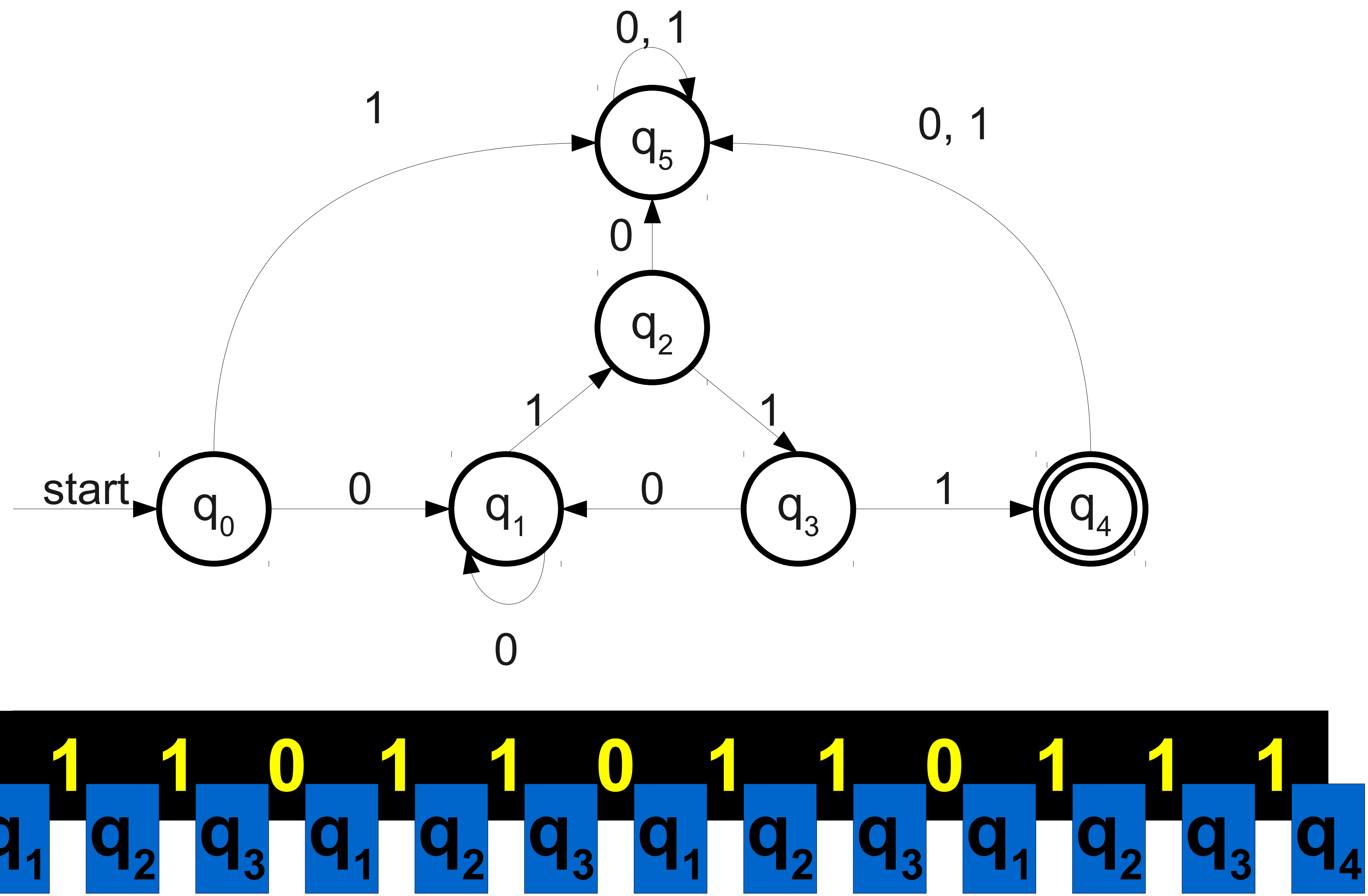
- The language *BALANCE* can be proven not to be regular using a stronger version of the pumping lemma.
- To see the full pumping lemma, we need to revisit our original insight.

An Important Observation

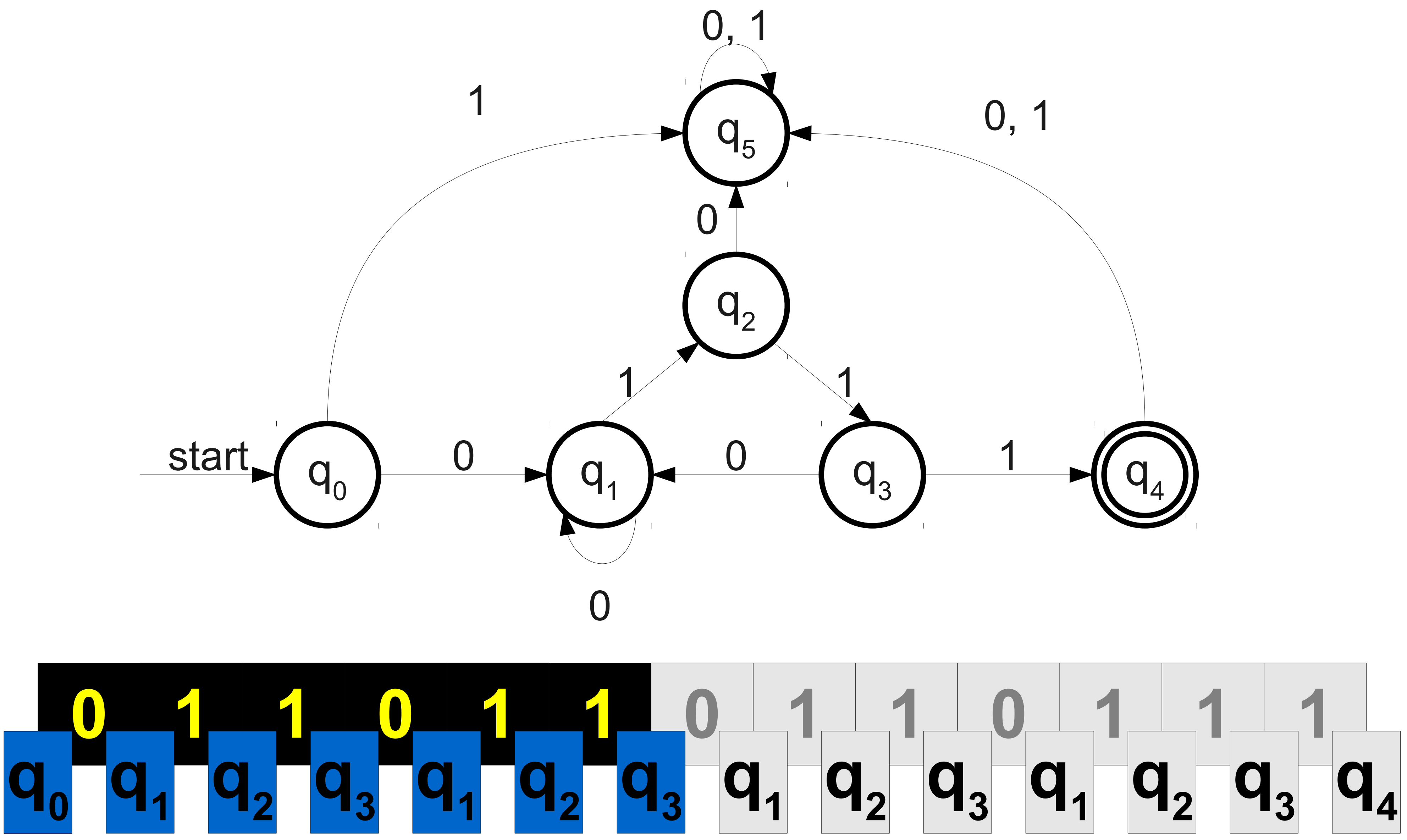


0 1 1 0 1 1 0 1 1 0 1 1 1

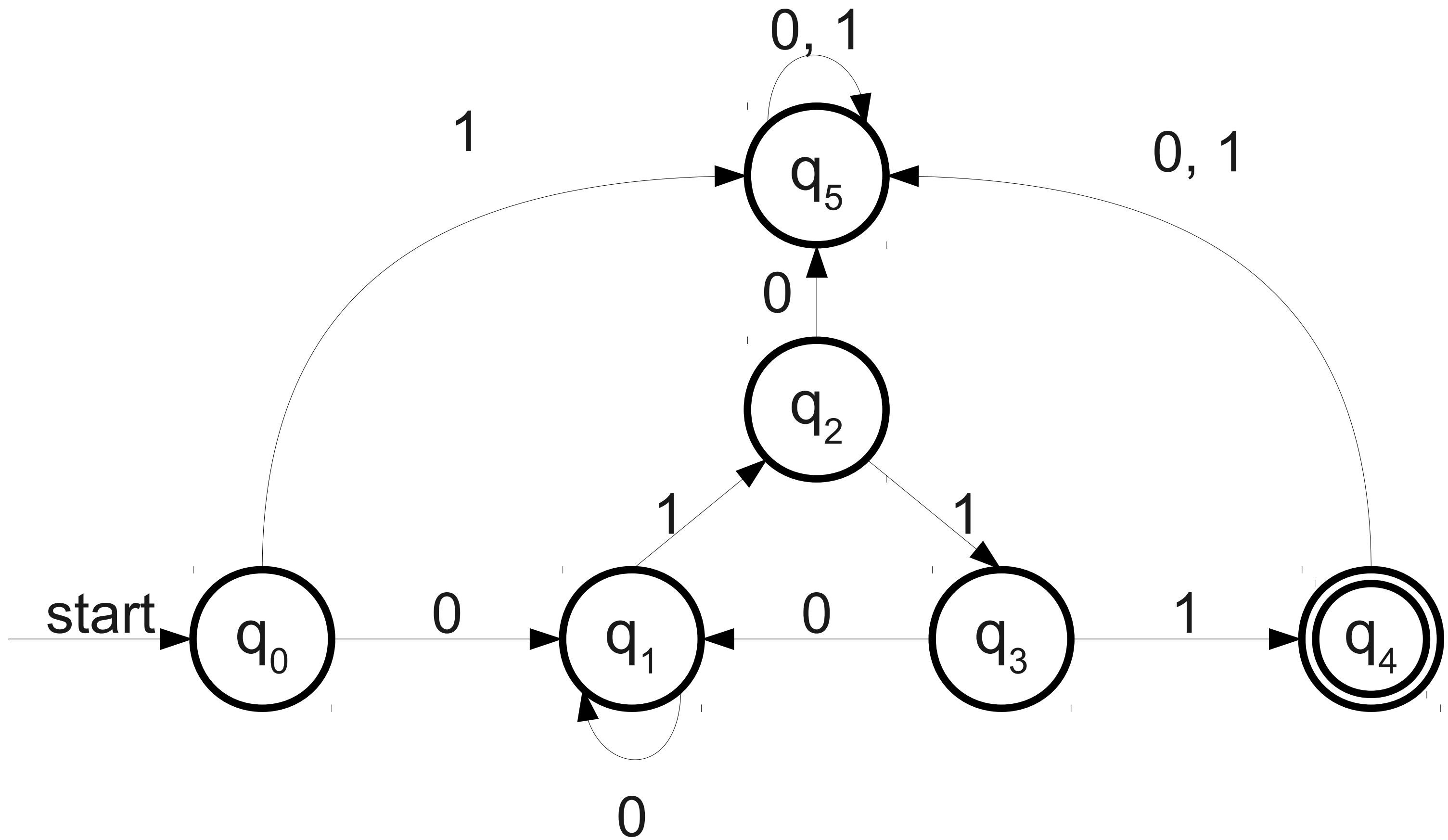
An Important Observation

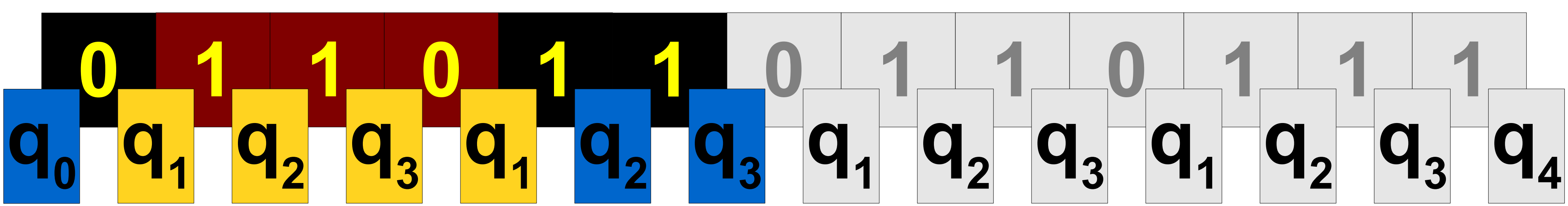


An Important Observation



An Important Observation





Weak Pumping Lemma Intuition

- Let D be a DFA with n states.
- Any string w accepted by D that has length at least n must visit some state twice.
  - Number of states visited is equal to $|w| + 1$.
  - By the pigeonhole principle, some state is duplicated.
- The substring of w in-between those revisited states can be removed, duplicated, tripled, quadrupled, etc. without changing the fact that w is accepted by D.

Pumping Lemma Intuition

- Let *D* be a DFA with n states.
- Any string *w* accepted by *D* that has length at least *n* must visit some state twice **within its first *n* characters**.
  - Number of states visited is equal ***n* + 1**.
  - By the pigeonhole principle, some state is duplicated.
- The substring of *w* in-between those revisited states can be removed, duplicated, tripled, quadrupled, etc. without changing the fact that *w* is accepted by *D*.

The Weak Pumping Lemma

For any regular language L,

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz$, w can be broken into three pieces,

$y \neq \varepsilon$ where the middle piece isn't empty,

$xy^iz \in L$ where the middle piece can be replicated zero or more times.

The Pumping Lemma

For any regular language L,

There exists a positive natural number n such that

For any $w \in L$ with $|w| \geq n$,

There exists strings x, y, z such that

For any natural number i,

$w = xyz,$	w can be broken into three pieces,
$\|xy\| \leq n,$	where the first two pieces occur at the start of the string,
$y \neq \varepsilon$	where the middle piece isn't empty,
$xy^iz \in L$	where the middle piece can be replicated zero or more times.

Why This Change Matters

- The restriction $|xy| \leq n$ means that we can limit where the string to pump must be.
- If we specifically craft the first n characters of the string to pump, we can force y to have a specific property.
- We can then show that y cannot be pumped arbitrarily many times.

BALANCE and the Pumping Lemma

$BALANCE = \{ w \in \{\texttt{0}, \texttt{1}\}^* \mid w$ contains an equal number of `0`s and `1`s. $\}$

BALANCE and the Pumping Lemma

$BALANCE = \{ w \in \{0, 1\}^* \mid w$ contains an equal number of 0s and 1s. $\}$

Suppose the pumping length is 4.

BALANCE and the Pumping Lemma

BALANCE = $\{ w \in \{0, 1\}^* \mid w$ contains an equal number of 0s and 1s. $\}$

Suppose the pumping length is 4.

0	0	0	0	1	1	1	1

BALANCE and the Pumping Lemma

BALANCE = { $w \in \{0, 1\}^*$ | w contains an equal number of 0s and 1s. }

Suppose the pumping length is 4.

$$\underbrace{0 \quad 0 \quad 0 \quad 0}_{\text{Since } |xy| \le 4\text{, the string to pump must be somewhere in here.}} \quad 1 \quad 1 \quad 1 \quad 1$$

BALANCE and the Pumping Lemma

BALANCE = { w ∈ {`0`, `1`}*| w contains an equal number of `0`s and `1`s. }

Suppose the pumping length is 4.

0	0	0	0	1	1	1	1

BALANCE and the Pumping Lemma

BALANCE = { w ∈ {0, 1}*| w contains an equal number of 0s and 1s. }

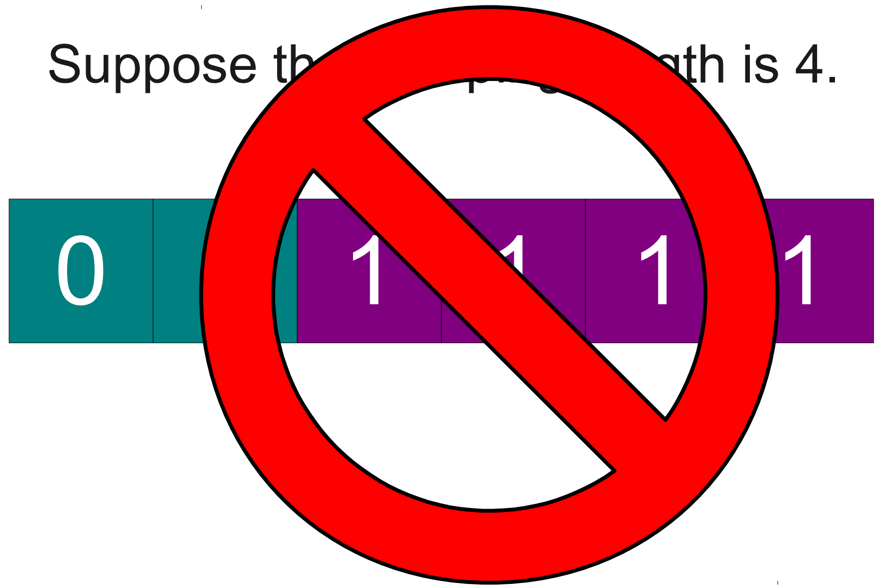
Suppose the pumping length is 4.

0	0	1	1	1	1

BALANCE and the Pumping Lemma

BALANCE = { *w* ∈ {`0`, `1`}*| *w* contains an equal number of `0`s and `1`s. }

Suppose th gth is 4.



BALANCE and the Pumping Lemma

BALANCE = { $w \in \{0, 1\}^*$ | w contains an equal number of 0s and 1s. }

Suppose the pumping length is 4.

0	0	0	0	1	1	1	1

BALANCE and the Pumping Lemma

$BALANCE = \{ w \in \{\mathtt{0}, \mathtt{1}\}^* \mid w$ contains an equal number of $\mathtt{0}$s and $\mathtt{1}$s. $\}$

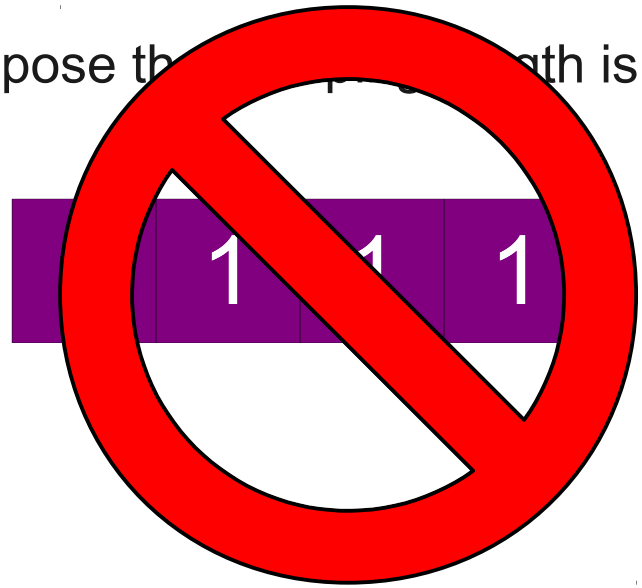
Suppose the pumping length is 4.

1	1	1	1

BALANCE and the Pumping Lemma

BALANCE = { w ∈ {0, 1}*| w contains an equal number of 0s and 1s. }

Suppose th… gth is 4.



BALANCE and the Pumping Lemma

BALANCE = { $w \in \{$`0`, `1`$\}^*$| w contains an equal number of `0`s and `1`s. }

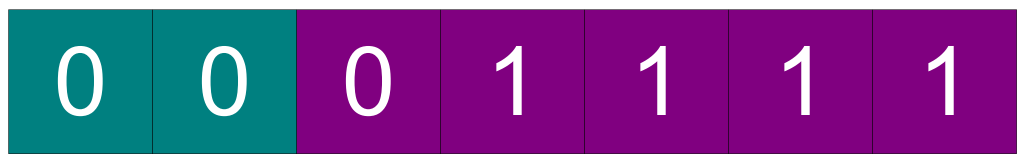
Suppose the pumping length is 4.

0	0	0	0	1	1	1	1

BALANCE and the Pumping Lemma

BALANCE = { *w* ∈ {0, 1}*| *w* contains an equal number of 0s and 1s. }

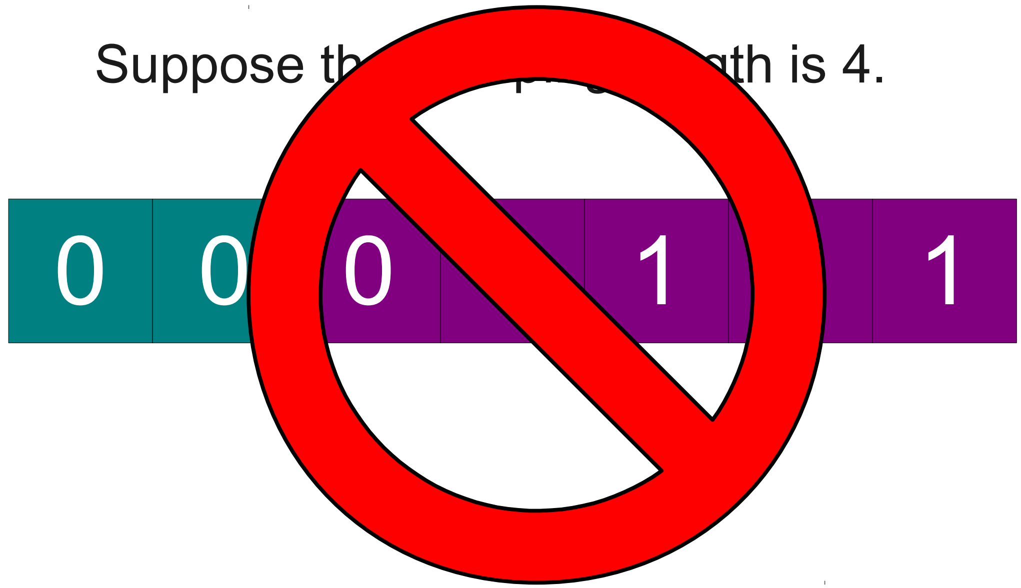
Suppose the pumping length is 4.



BALANCE and the Pumping Lemma

BALANCE = { w ∈ {0, 1}*| w contains an equal number of 0s and 1s. }

Suppose th [illegible] gth is 4.



Theorem: BALANCE is not regular.

Theorem: BALANCE is not regular.

Proof: By contradiction; assume that *BALANCE* is regular.

Theorem: BALANCE is not regular.

Proof: By contradiction; assume that *BALANCE* is regular.
Let n be the length guaranteed by the pumping lemma.

Theorem: *BALANCE* is not regular.

Proof: By contradiction; assume that *BALANCE* is regular.
Let n be the length guaranteed by the pumping lemma.
Consider the string w = $\mathtt{0}^{n}\mathtt{1}^{n}$.

Theorem: BALANCE is not regular.

Proof: By contradiction; assume that *BALANCE* is regular.
Let n be the length guaranteed by the pumping lemma.
Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and
$w \in BALANCE$.

Theorem: BALANCE is not regular.

Proof: By contradiction; assume that *BALANCE* is regular. Let n be the length guaranteed by the pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in BALANCE$. Therefore, there exist strings x, y, and z such that $w = xyz$, $|xy| \leq n$, $y \neq \varepsilon$, and for any natural number i, $xy^iz \in BALANCE$.

Theorem: *BALANCE* is not regular.

Proof: By contradiction; assume that *BALANCE* is regular. Let n be the length guaranteed by the pumping lemma. Consider the string w = 0^n1^n. Then $|w| = 2n \geq n$ and $w \in BALANCE$. Therefore, there exist strings x, y, and z such that $w = xyz$, $|xy| \leq n$, $y \neq \varepsilon$, and for any natural number i, $xy^iz \in BALANCE$. Since $|xy| \leq n$, y must consist solely of 0s.

Theorem: BALANCE is not regular.

Proof: By contradiction; assume that *BALANCE* is regular. Let n be the length guaranteed by the pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in BALANCE$. Therefore, there exist strings x, y, and z such that $w = xyz$, $|xy| \leq n$, $y \neq \varepsilon$, and for any natural number i, $xy^iz \in BALANCE$. Since $|xy| \leq n$, y must consist solely of `0`s.

This is why the pumping lemma is more powerful than the weak pumping lemma. We can force y to be made purely of 0s, rather than some combination of 0s and 1s.

Theorem: BALANCE is not regular.

Proof: By contradiction; assume that *BALANCE* is regular. Let n be the length guaranteed by the pumping lemma. Consider the string w = $\mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in$ *BALANCE*. Therefore, there exist strings x, y, and z such that $w = xyz$, $|xy| \leq n$, $y \neq \varepsilon$, and for any natural number i, $xy^iz \in$ *BALANCE*. Since $|xy| \leq n$, y must consist solely of $\mathtt{0}$s. But then $xy^2z = \mathtt{0}^{n+|y|}\mathtt{1}^n$, and since $|y| > 0$, $xy^2z \notin$ *BALANCE*.

Theorem: *BALANCE* is not regular.

Proof: By contradiction; assume that *BALANCE* is regular. Let n be the length guaranteed by the pumping lemma. Consider the string w = $\mathtt{0}^{n}\mathtt{1}^{n}$. Then $|w| = 2n \geq n$ and $w \in$ *BALANCE*. Therefore, there exist strings x, y, and z such that $w = xyz$, $|xy| \leq n$, $y \neq \varepsilon$, and for any natural number i, $xy^{i}z \in$ *BALANCE*. Since $|xy| \leq n$, y must consist solely of `0`s. But then $xy^{2}z$ = $\mathtt{0}^{n+|y|}\mathtt{1}^{n}$, and since $|y| > 0$, $xy^{2}z \notin$ *BALANCE*.

We have reached a contradiction, so our assumption was wrong and *BALANCE* is not regular.

Theorem: *BALANCE* is not regular.

Proof: By contradiction; assume that *BALANCE* is regular. Let n be the length guaranteed by the pumping lemma. Consider the string $w = \mathtt{0}^n\mathtt{1}^n$. Then $|w| = 2n \geq n$ and $w \in BALANCE$. Therefore, there exist strings x, y, and z such that $w = xyz$, $|xy| \leq n$, $y \neq \varepsilon$, and for any natural number i, $xy^iz \in BALANCE$. Since $|xy| \leq n$, y must consist solely of $\mathtt{0}$s. But then $xy^2z = \mathtt{0}^{n+|y|}\mathtt{1}^n$, and since $|y| > 0$, $xy^2z \notin BALANCE$.

We have reached a contradiction, so our assumption was wrong and *BALANCE* is not regular. ■

Summary of the Pumping Lemma

- Using the pigeonhole principle, we can prove the **weak pumping lemma** and **pumping lemma**.
- These lemmas describe essential properties of the regular languages.
- Any language that fails to have these properties cannot be regular.

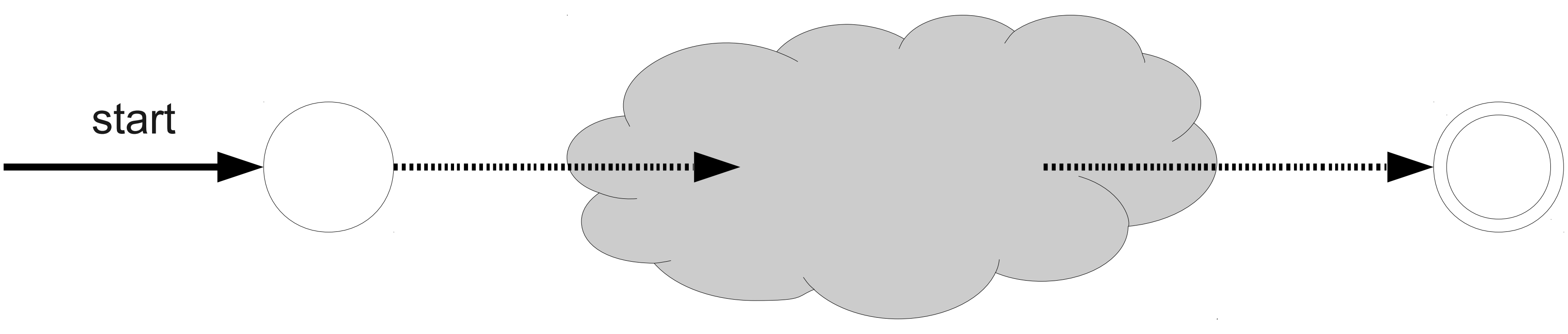
Next Time

- **Beyond Regular Languages**
  - Context-free languages.
  - Context-free grammars.

Appendix: From Regular Expressions to NFAs

A Marvelous Construction

- To show that any language described by a regular expression is regular, we show how to convert a regular expression into an NFA.
- *Theorem:* For any regular expression R, there is an NFA N such that
  - $\mathscr{L}(R) = \mathscr{L}(N)$
  - N has exactly one accepting state.
  - N has no transitions into its start state.
  - N has no transitions out of its accepting state.



A Marvelous Construction

To show that any language described by a regular expression is regular, we show how to convert a regular expression into an NFA.

Theorem: For any regular expression R, there is an NFA N such that

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

- N has exactly one accepting state.
- N has no transitions into its start state.
- N has no transitions out of its accepting state.

A Marvelous Construction

To show that any language c
expression is regular, we sh
regular expression into an N

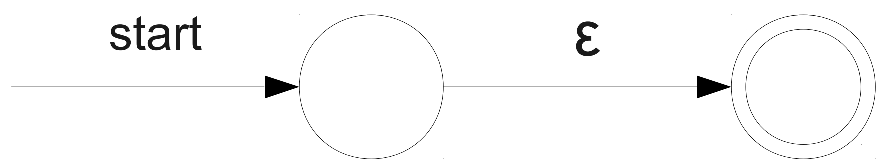
Theorem: For any regular e
NFA N such that

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

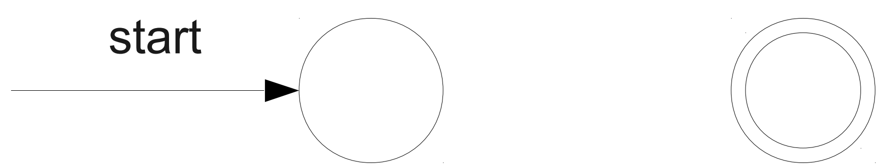
These are stronger requirements than are necessary for a normal NFA. We enforce these rules to simplify the construction.

- N has exactly one accepting state.
- N has no transitions into its start state.
- N has no transitions out of its accepting state.

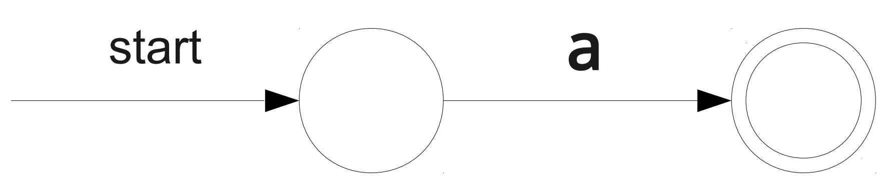
Base Cases



Automaton for ε



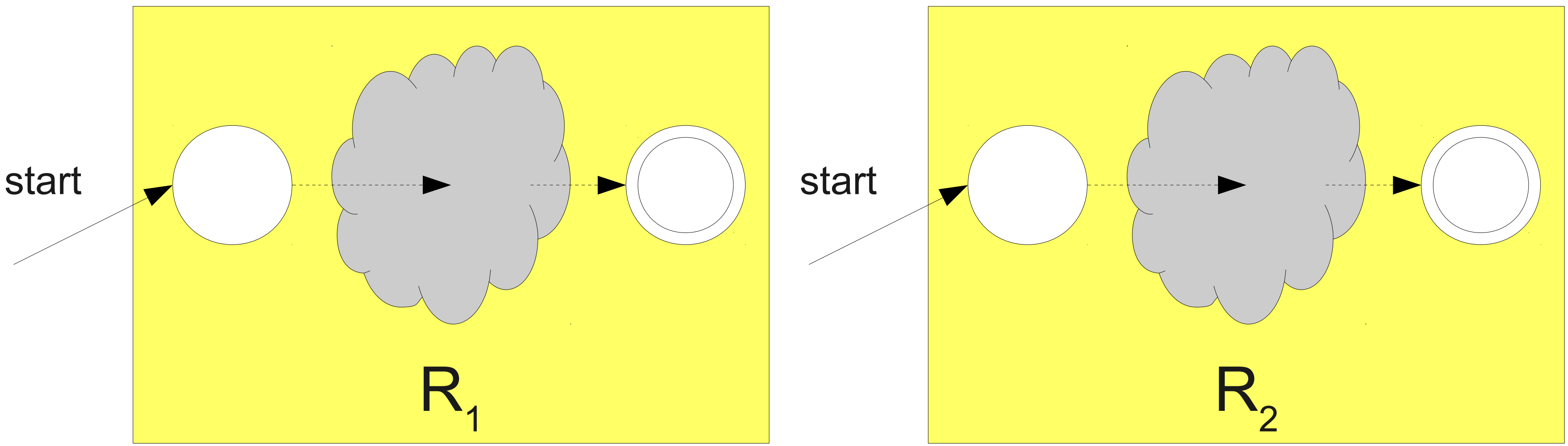
Automaton for Ø



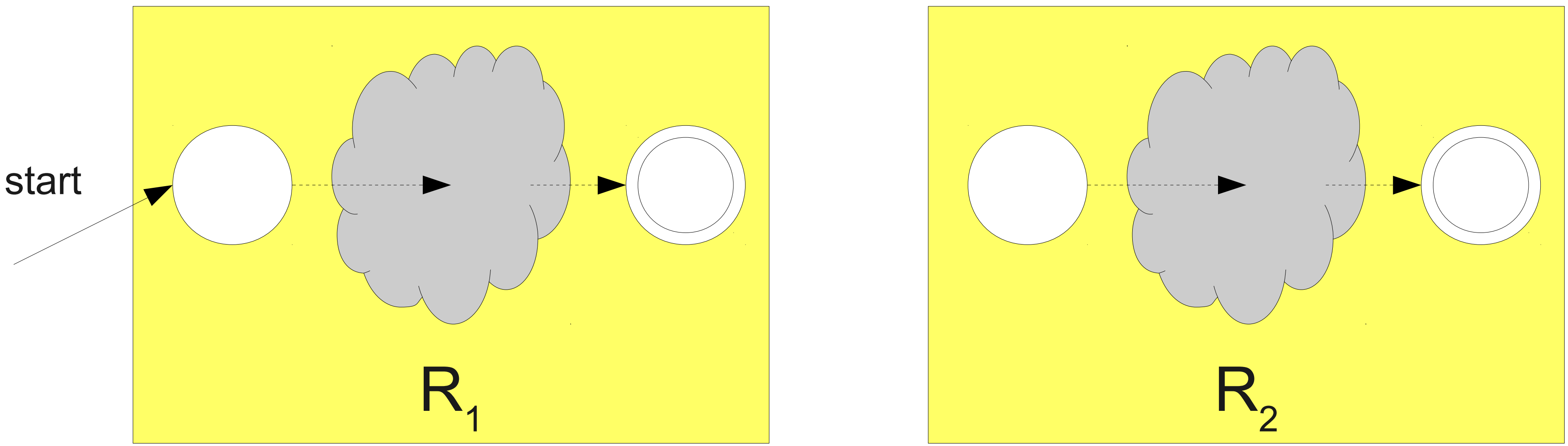
Automaton for single character **a**

Construction for R_1R_2

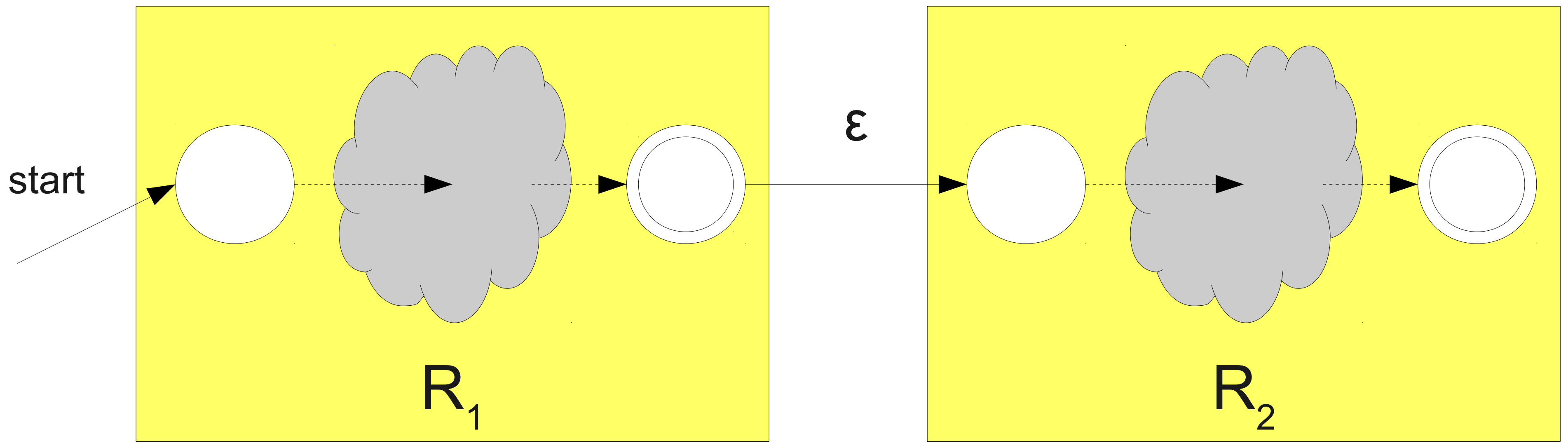
Construction for R_1R_2



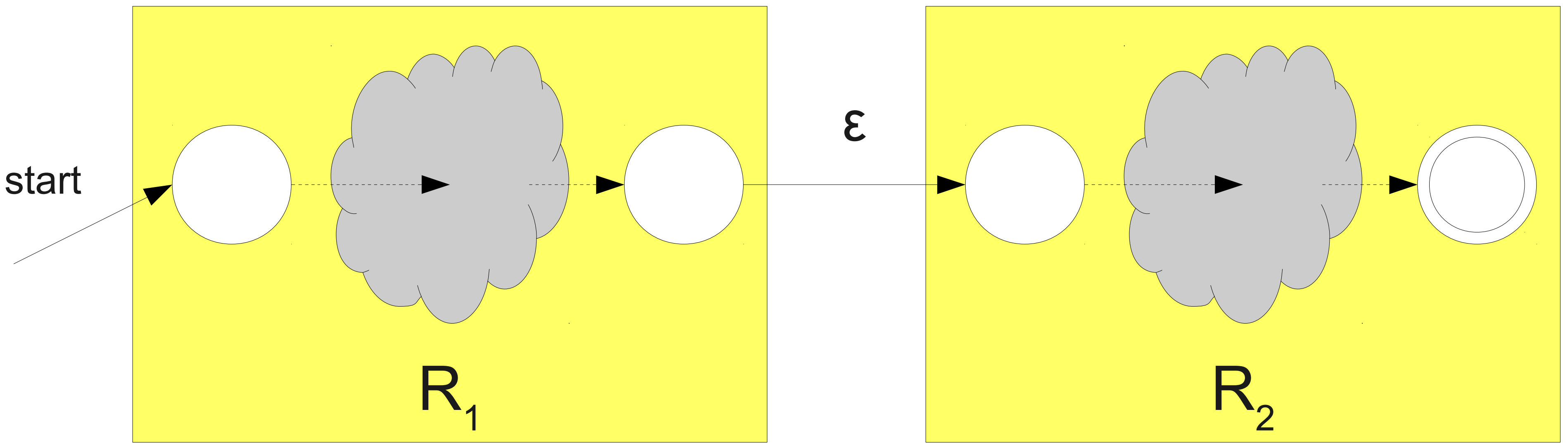
Construction for R_1R_2



Construction for R_1R_2

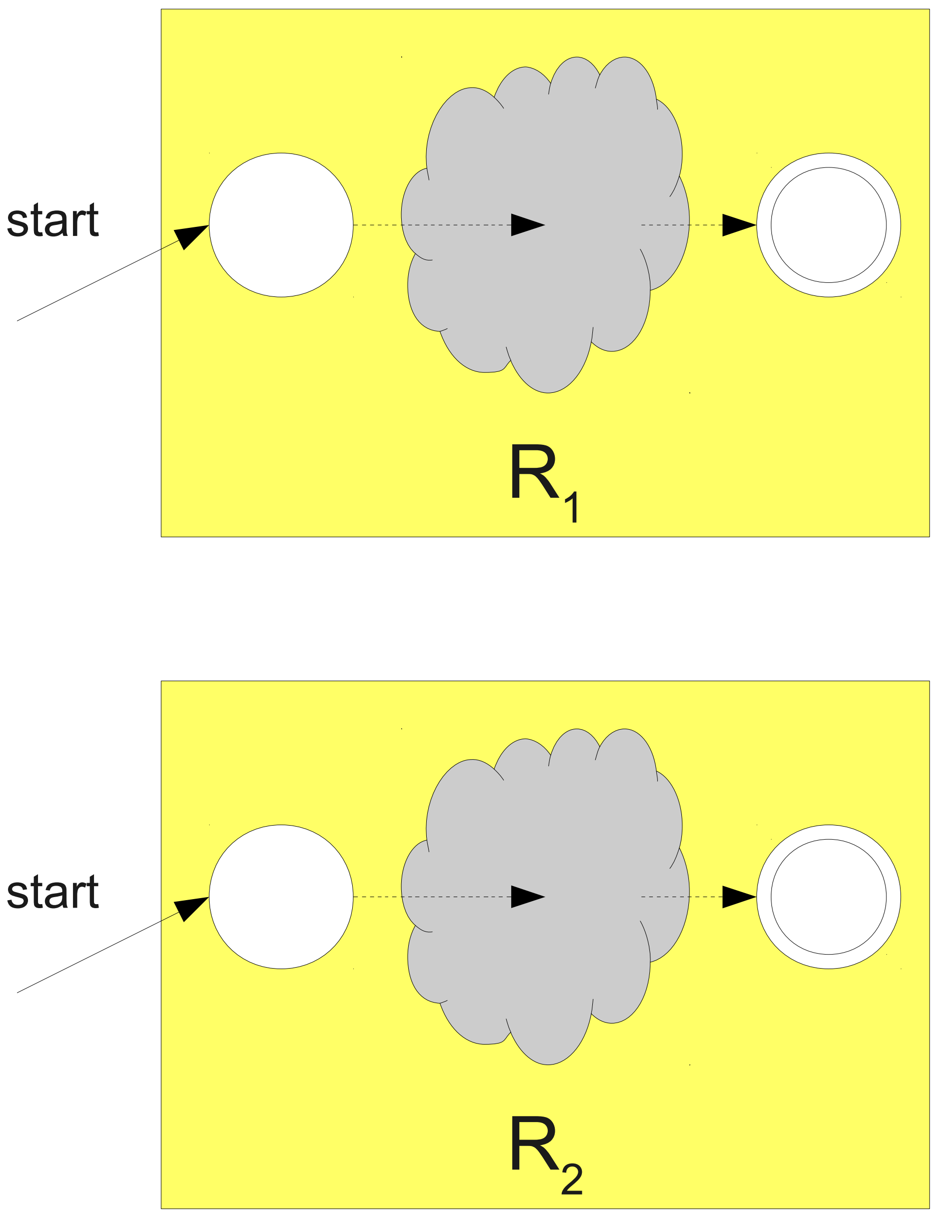


Construction for R_1R_2

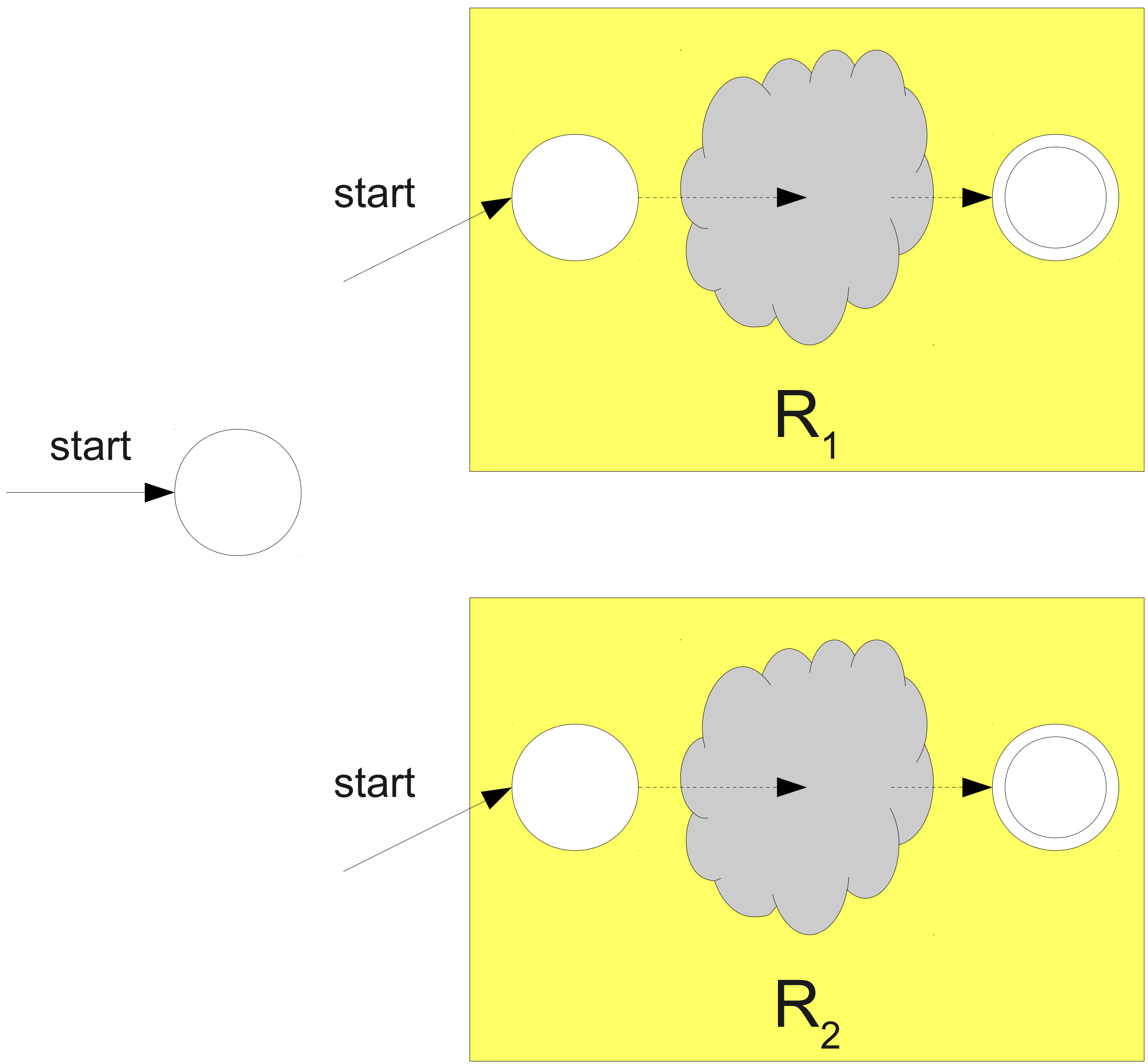


Construction for $R_1 \mid R_2$

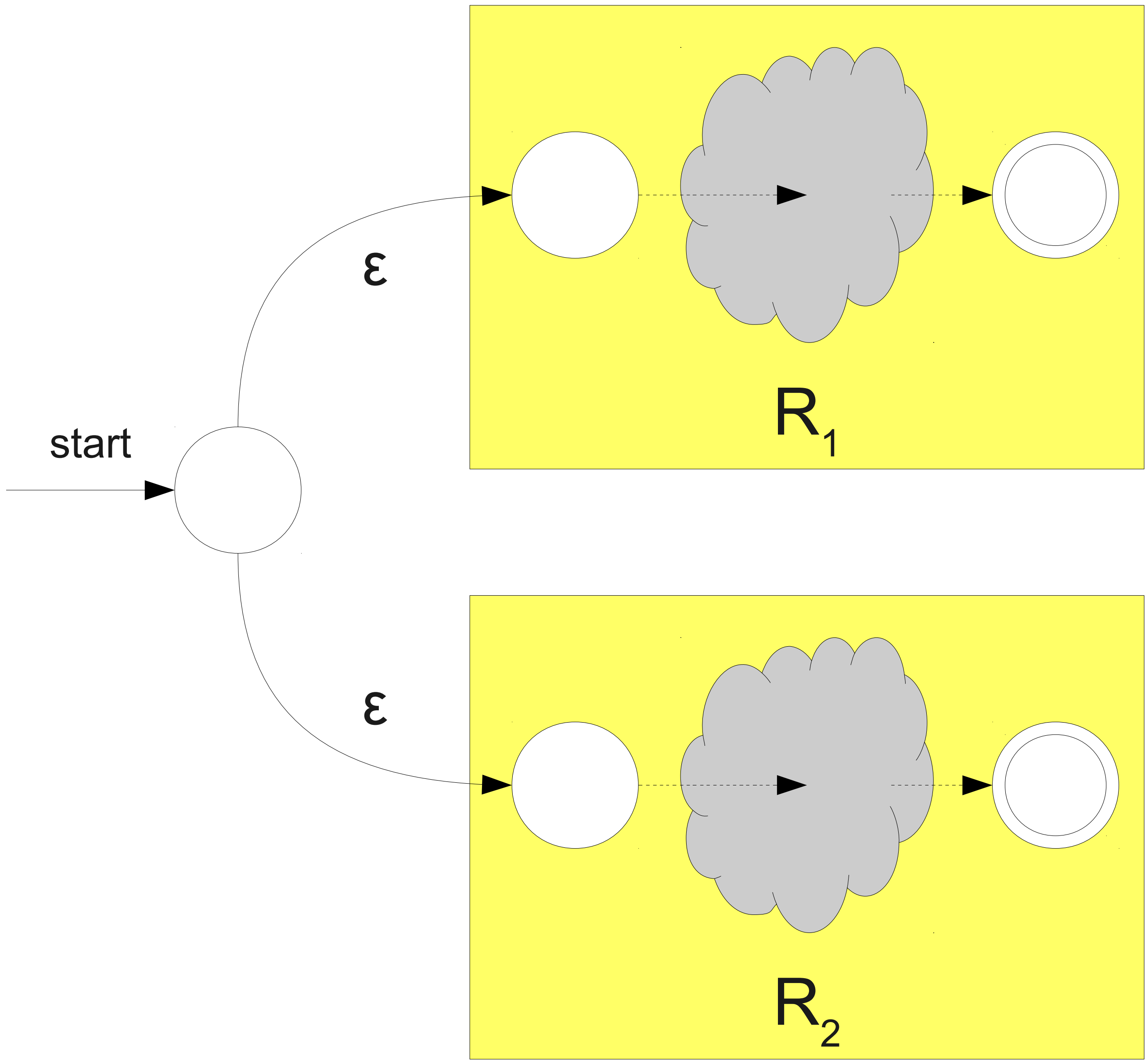
Construction for R_1 | R_2



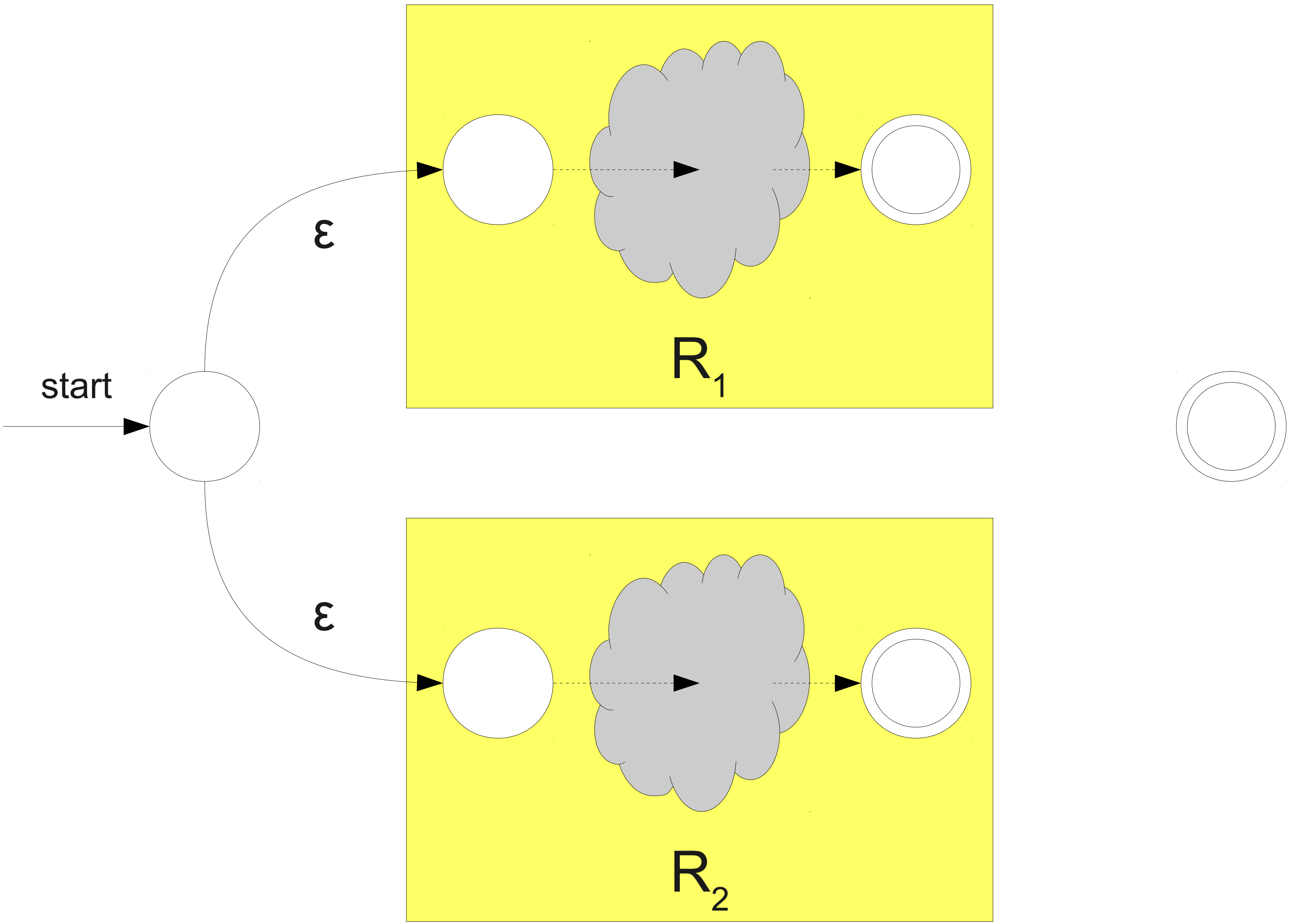
Construction for $R_1 \mid R_2$



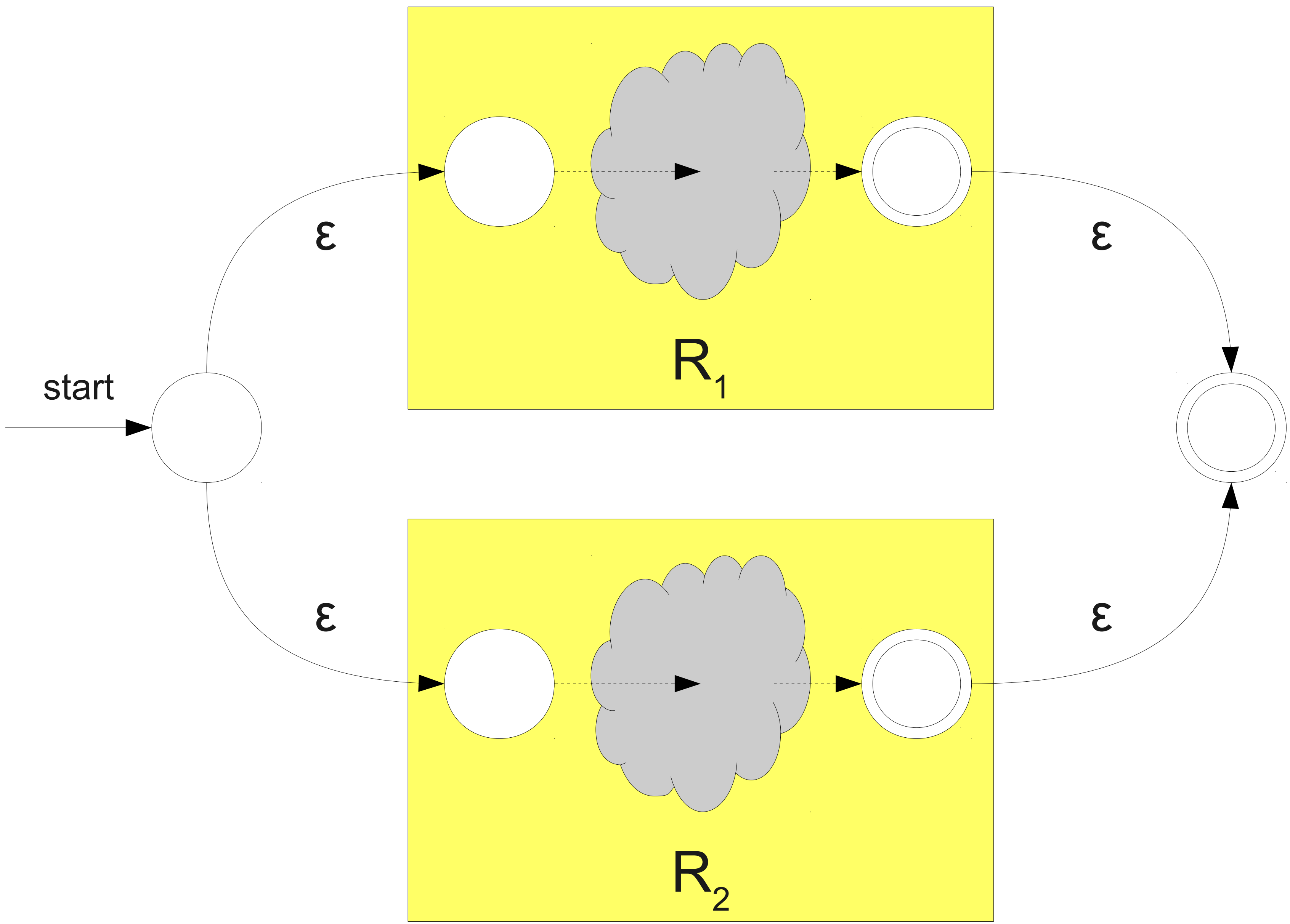
Construction for R_1 | R_2



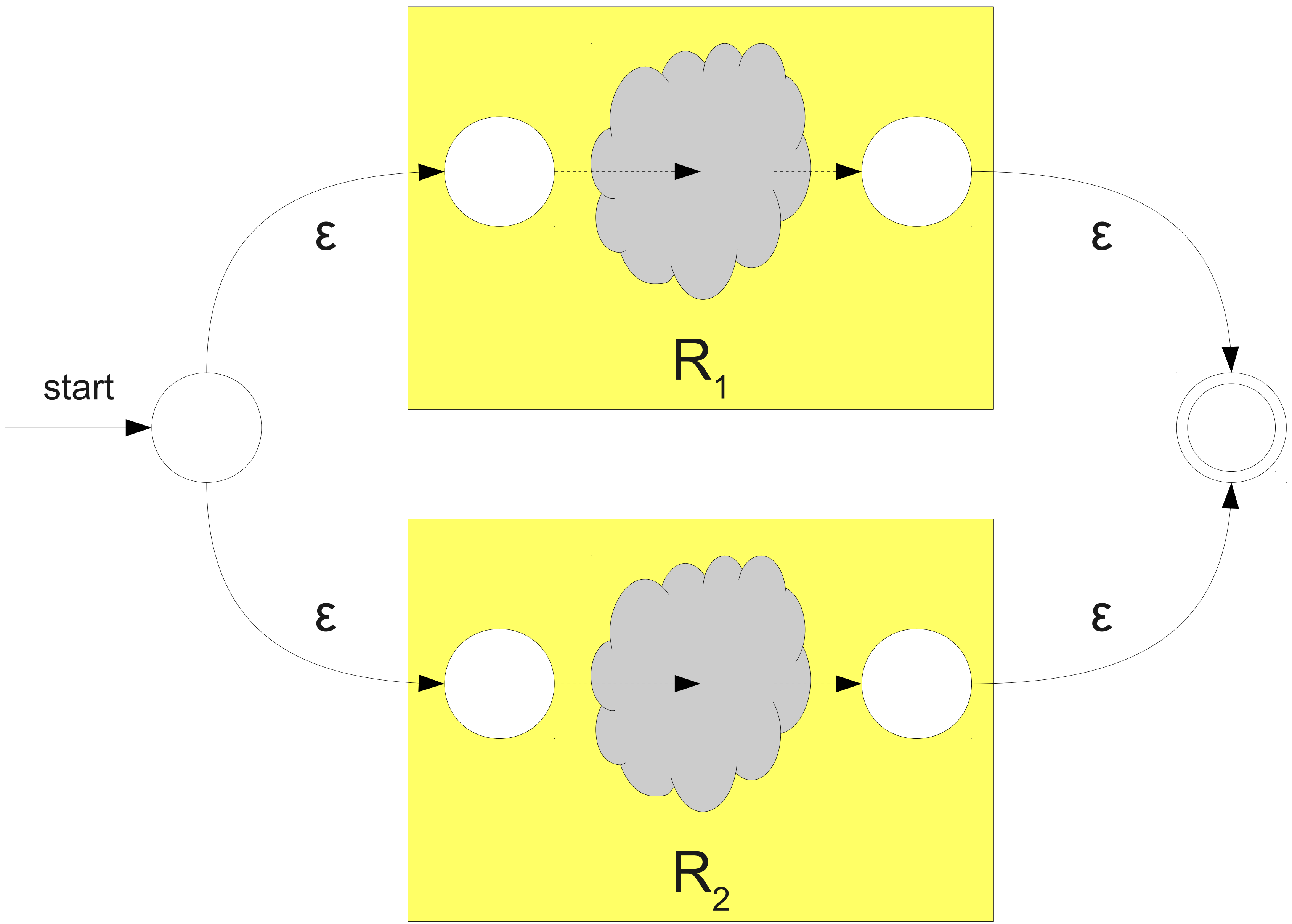
Construction for $R_1 \mid R_2$



Construction for $R_1 \mid R_2$

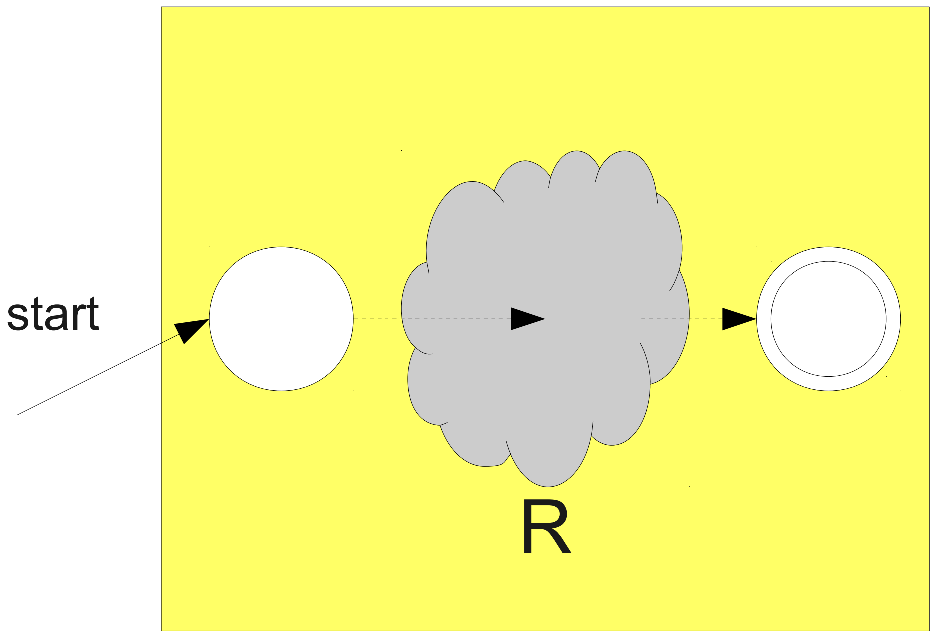


Construction for $R_1 \mid R_2$

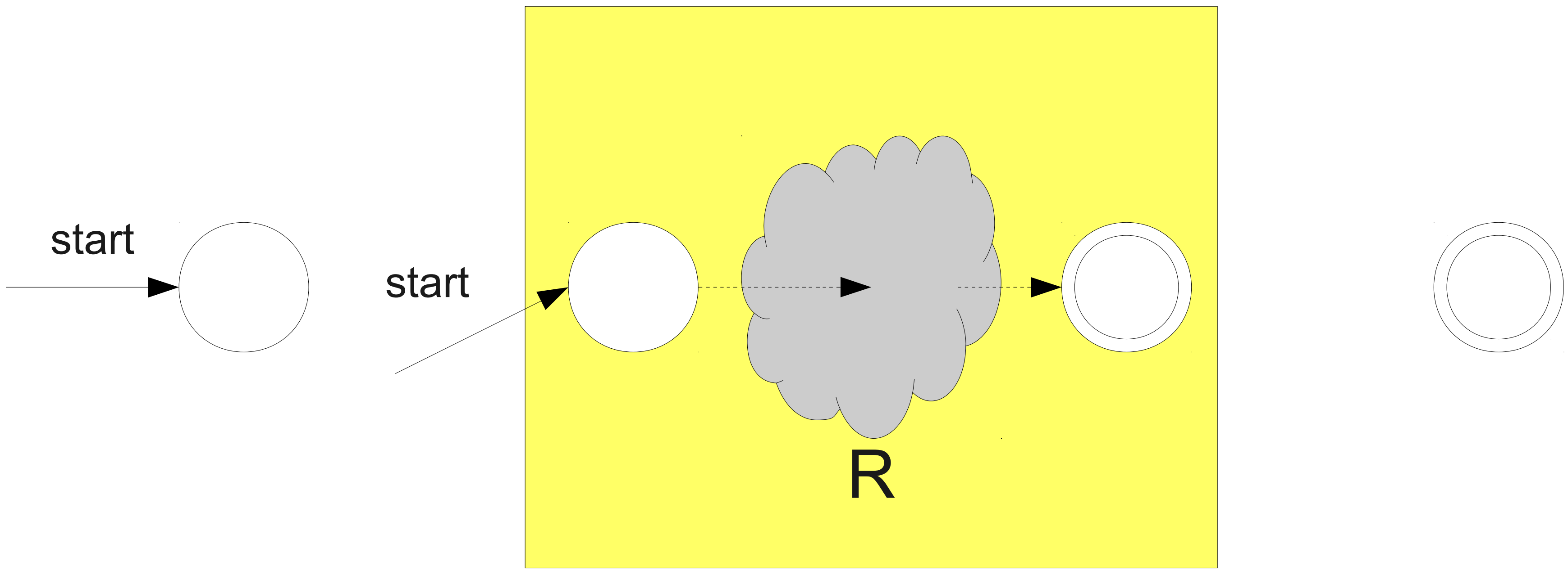


Construction for R*

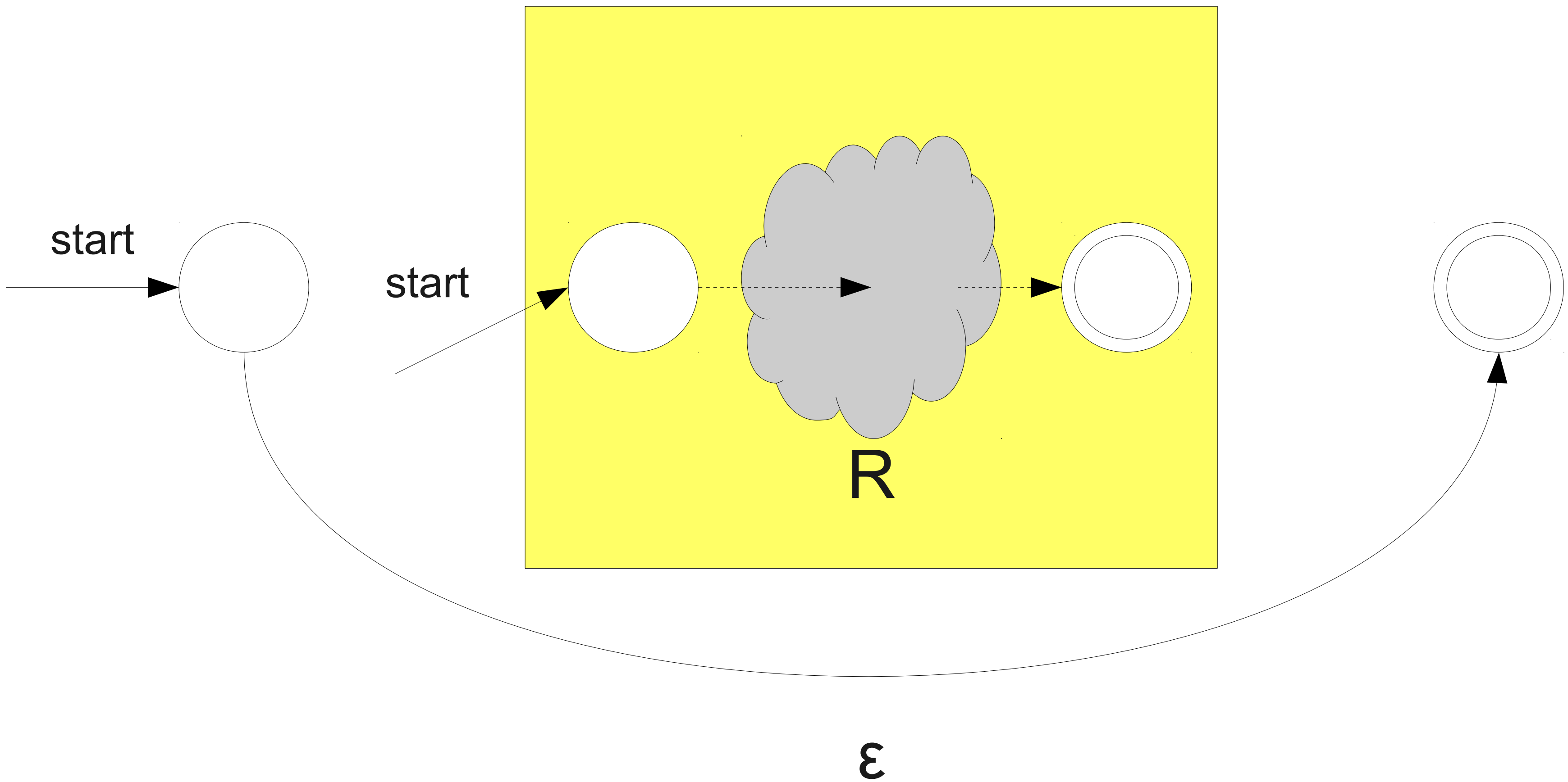
Construction for R*



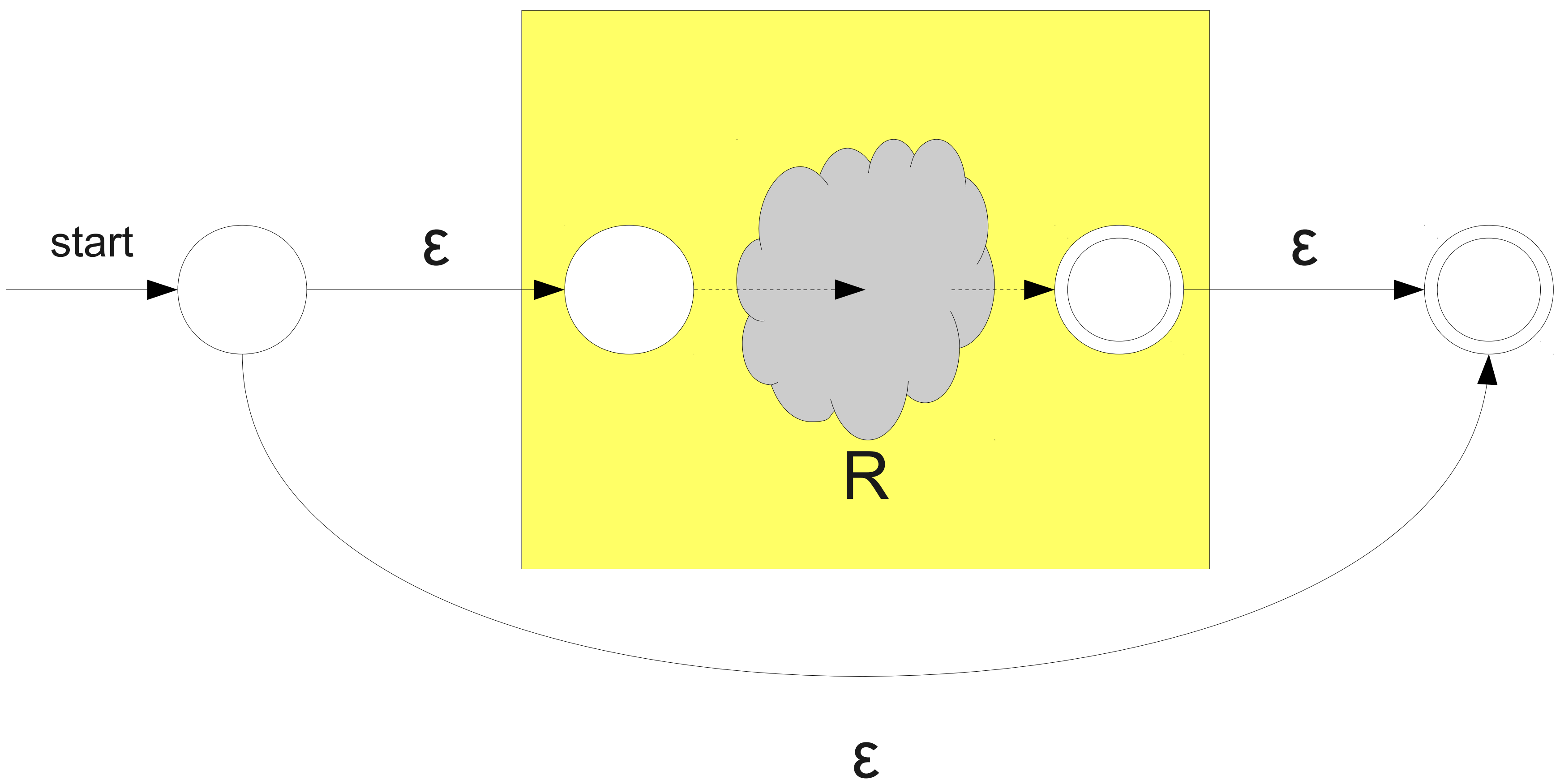
Construction for R*



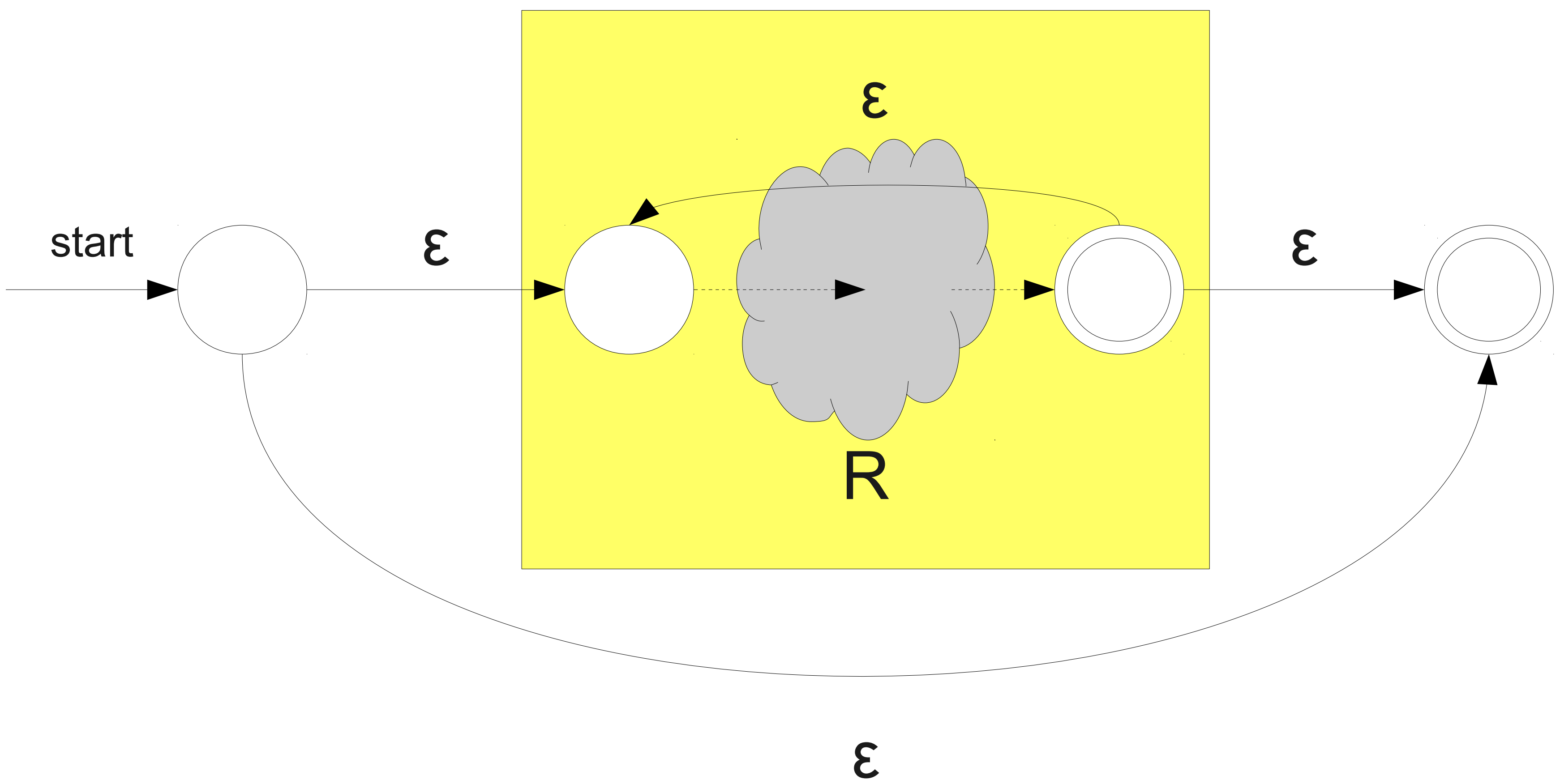
Construction for R*



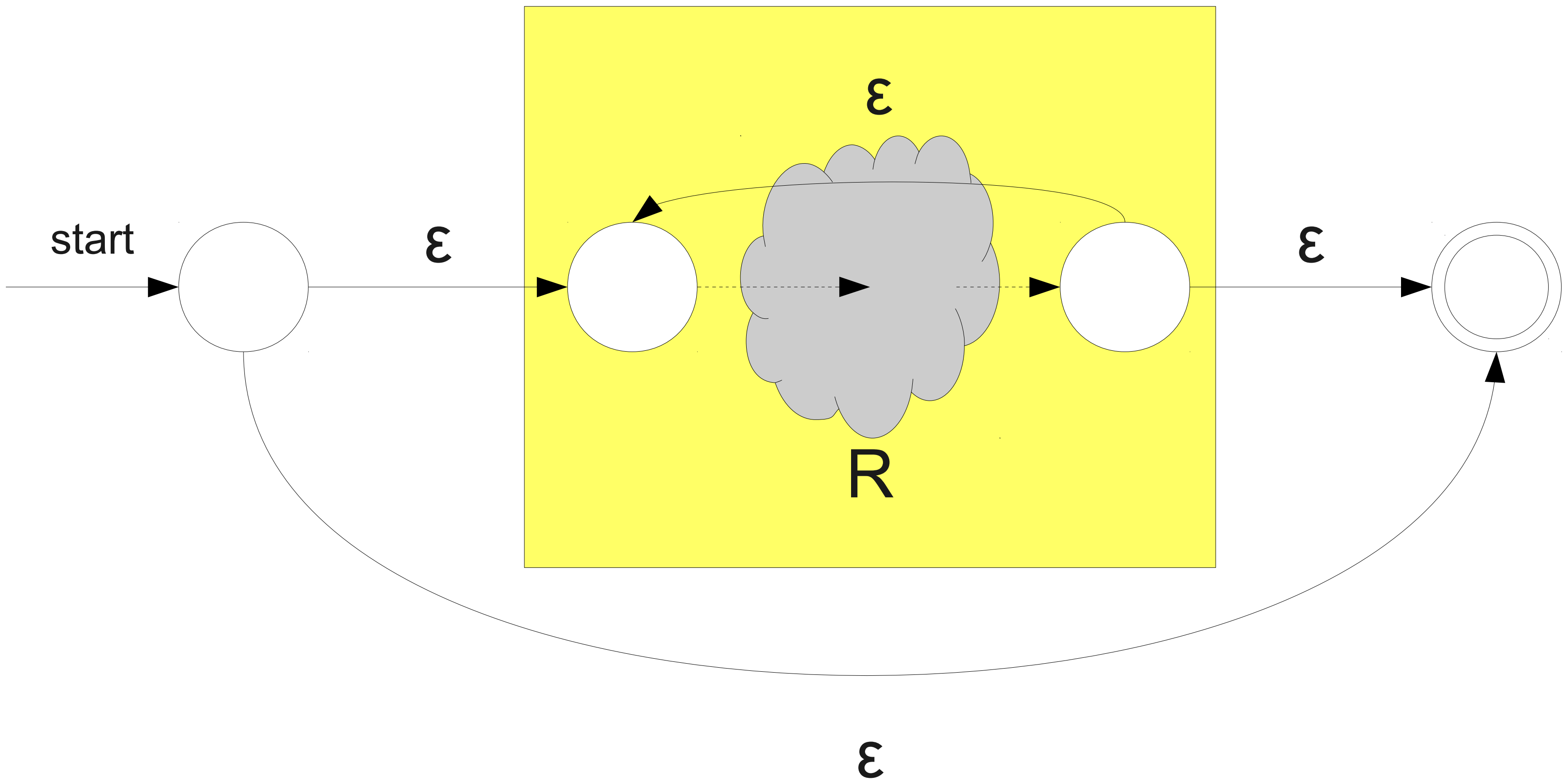
Construction for R*



Construction for R*

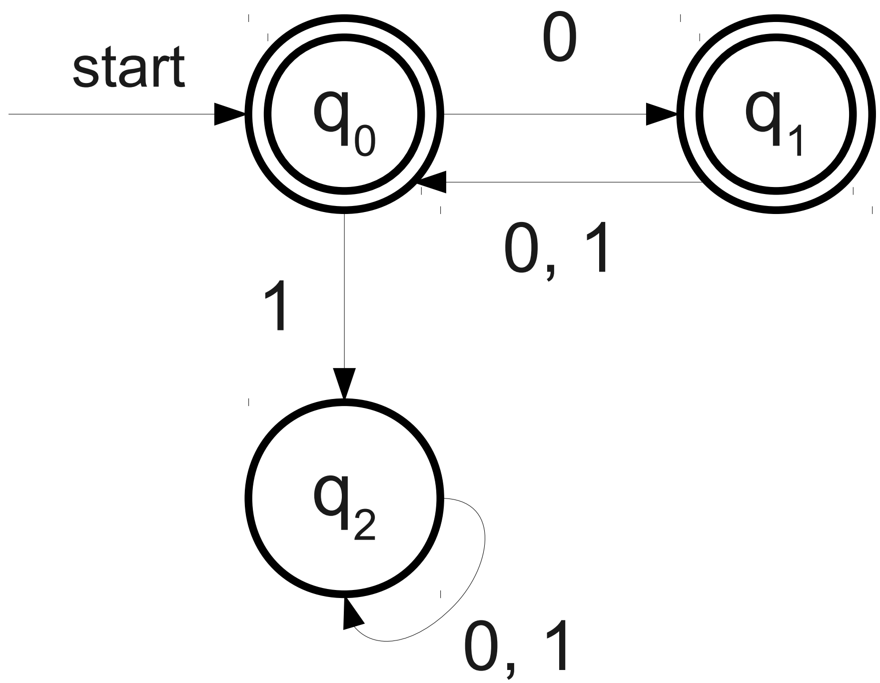


Construction for R*

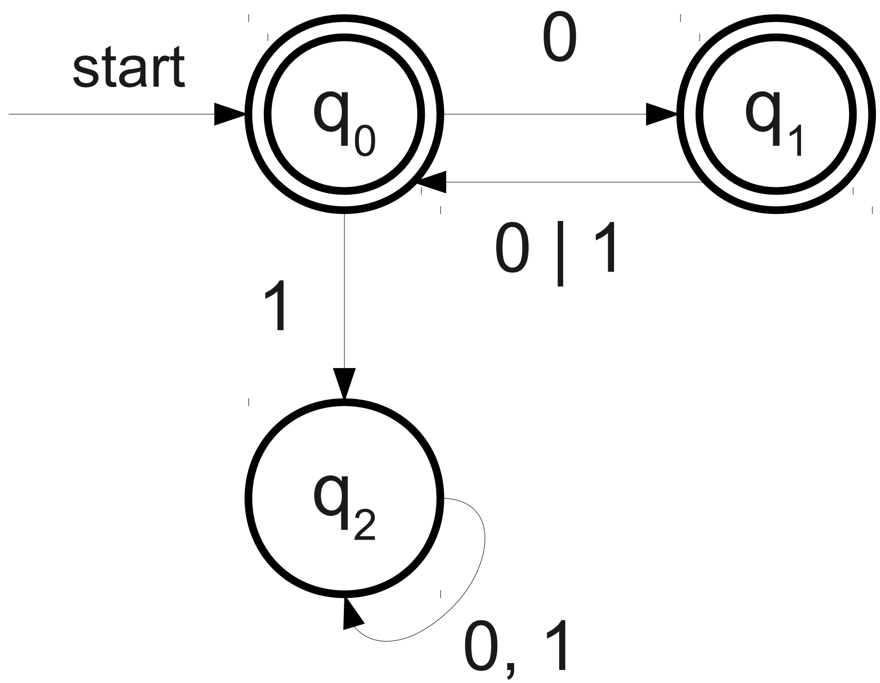


Appendix: From NFAs to Regular Expressions

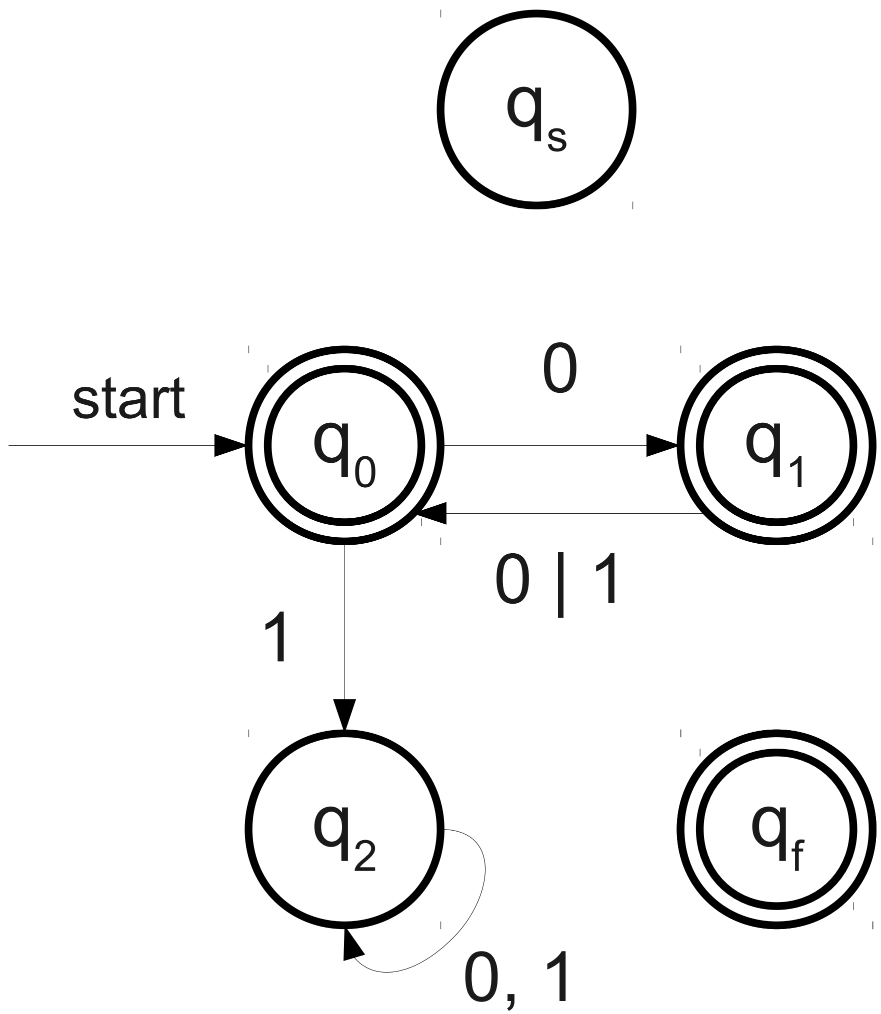
From NFAs to Regular Expressions



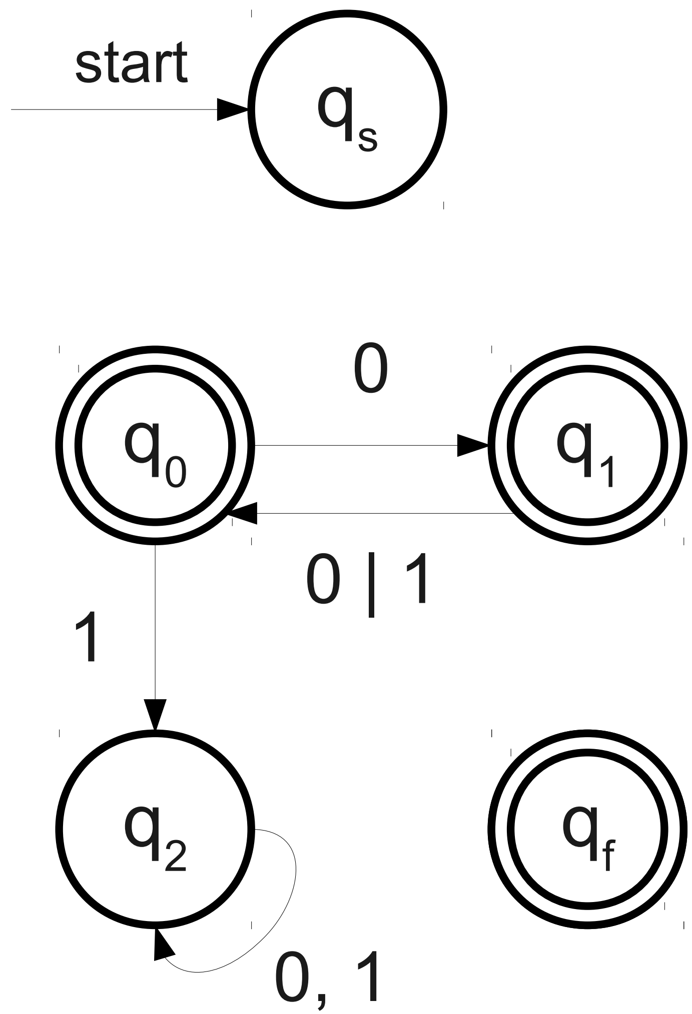
From NFAs to Regular Expressions



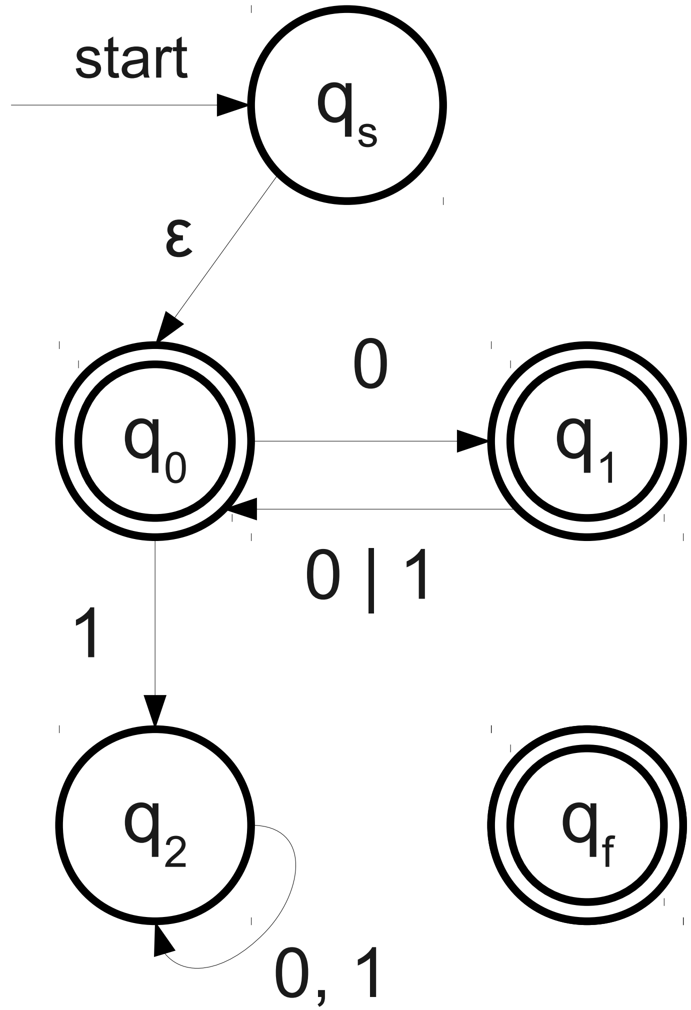
From NFAs to Regular Expressions



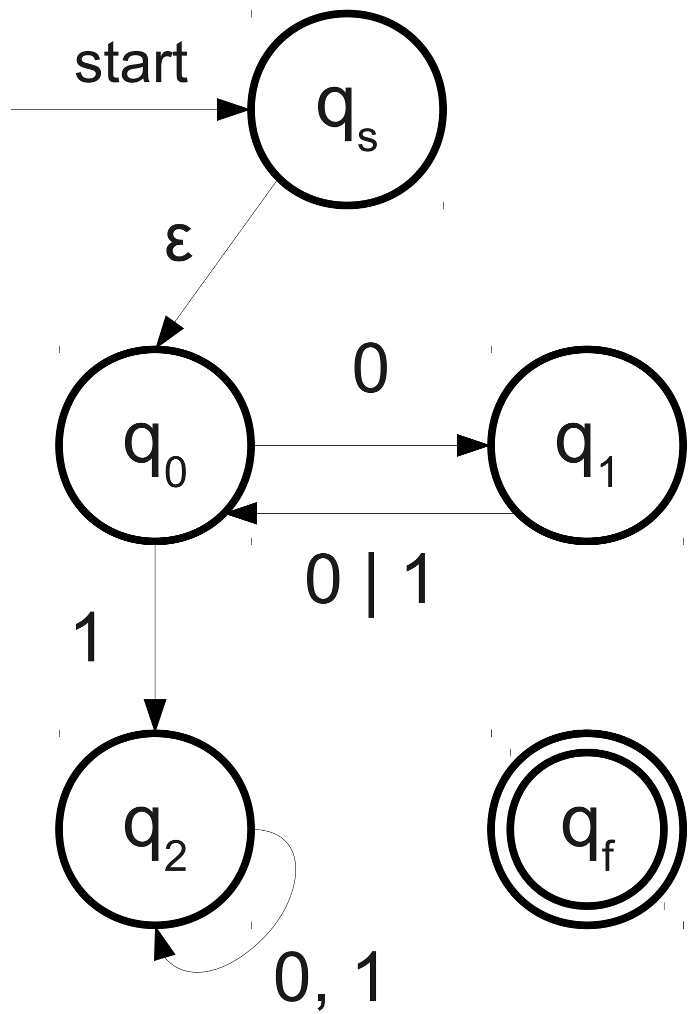
From NFAs to Regular Expressions



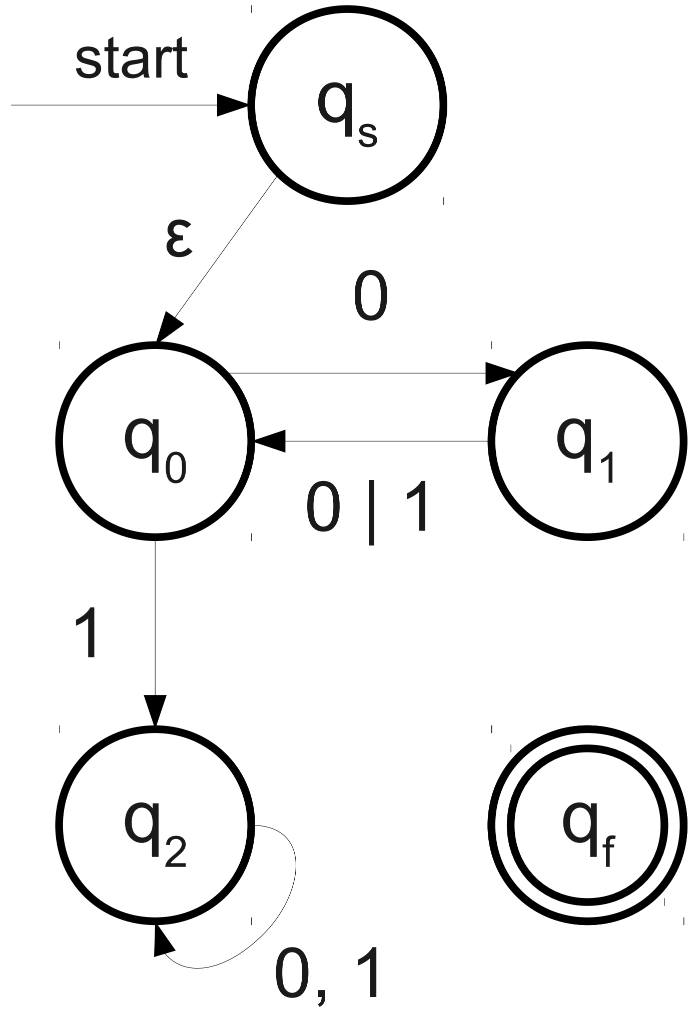
From NFAs to Regular Expressions



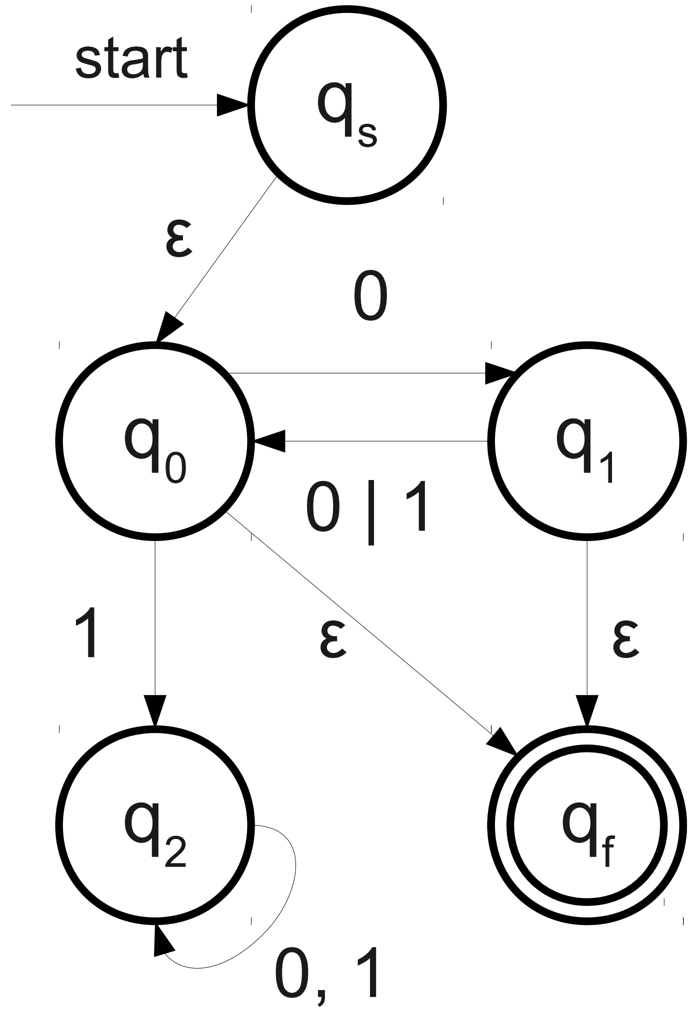
From NFAs to Regular Expressions



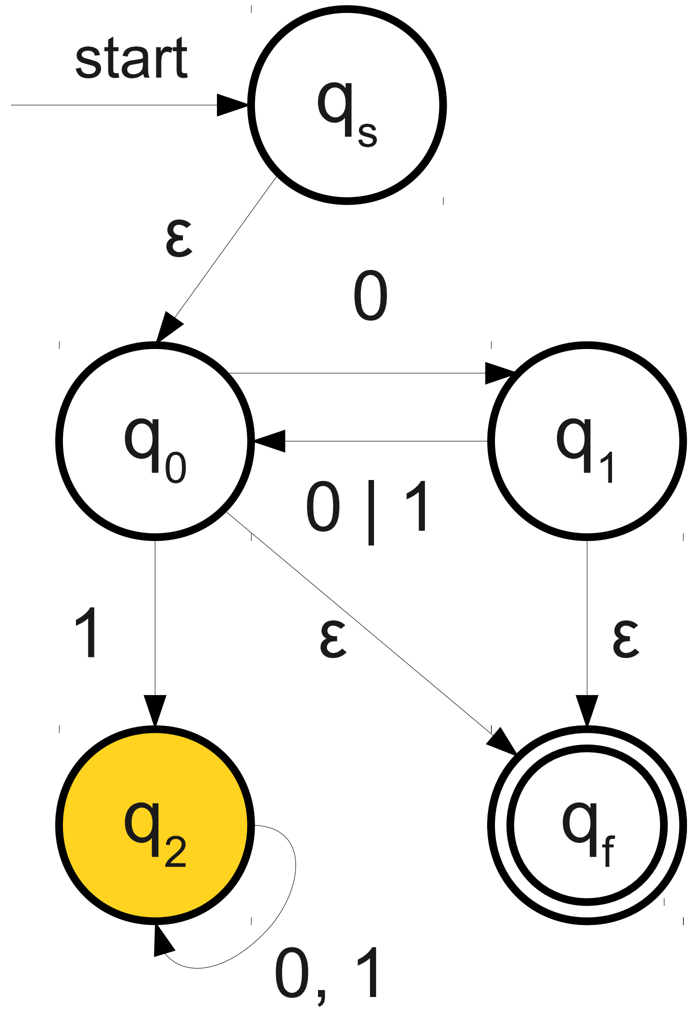
From NFAs to Regular Expressions



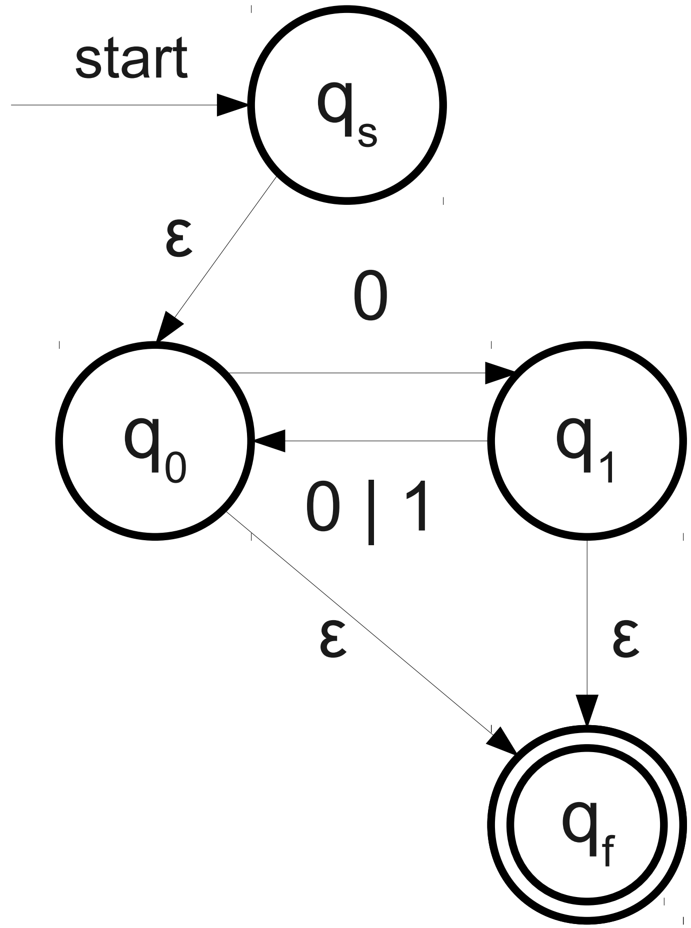
From NFAs to Regular Expressions



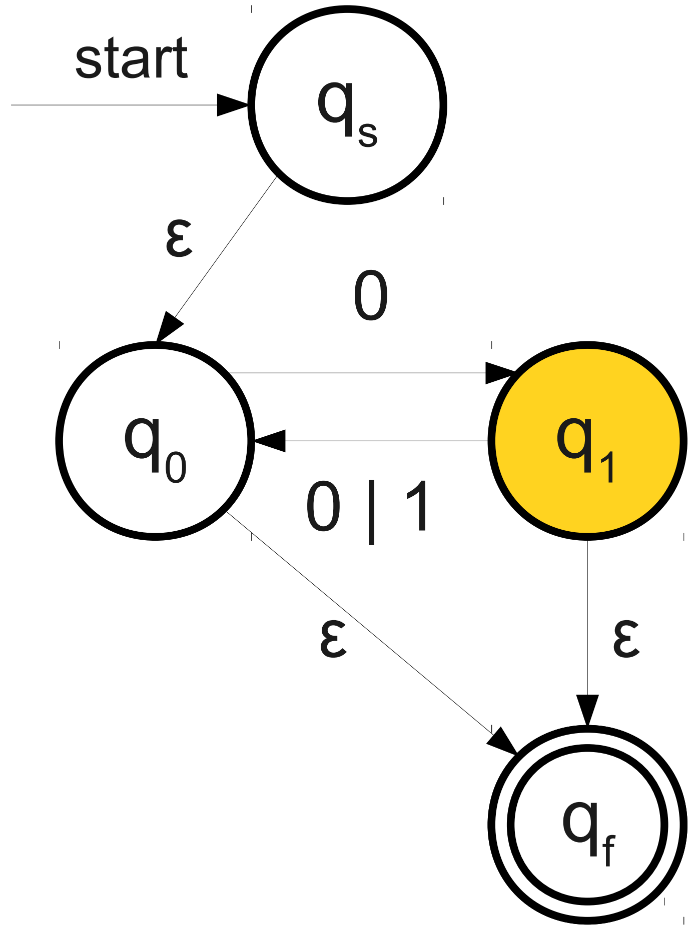
From NFAs to Regular Expressions



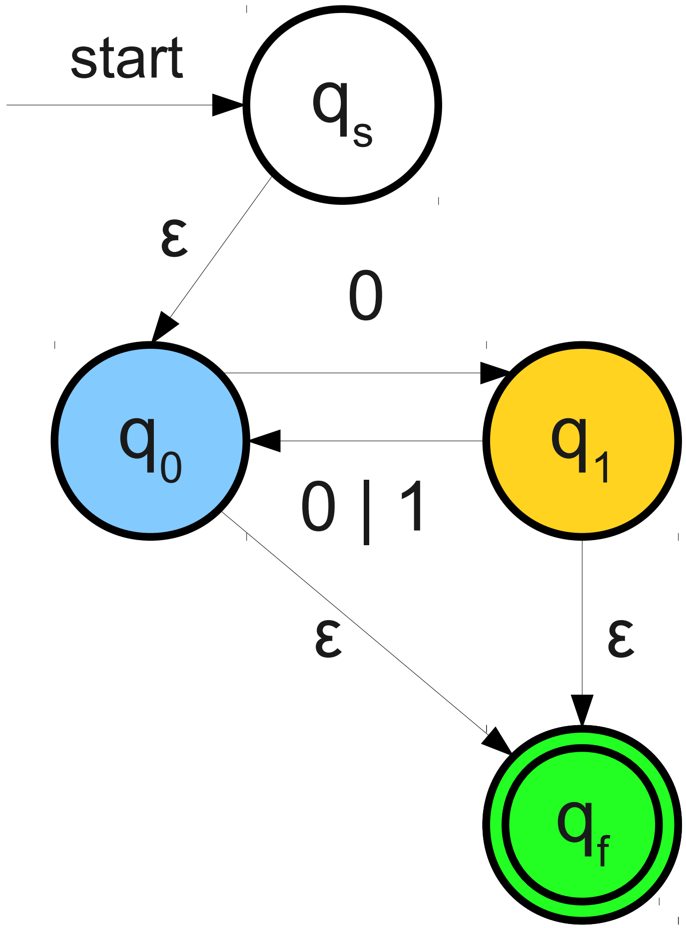
From NFAs to Regular Expressions



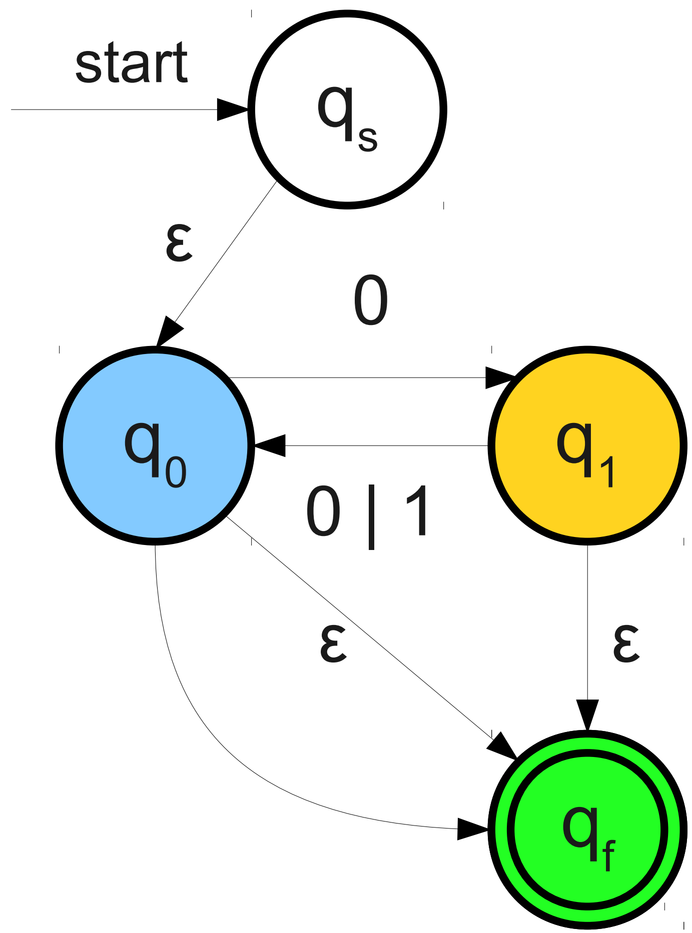
From NFAs to Regular Expressions



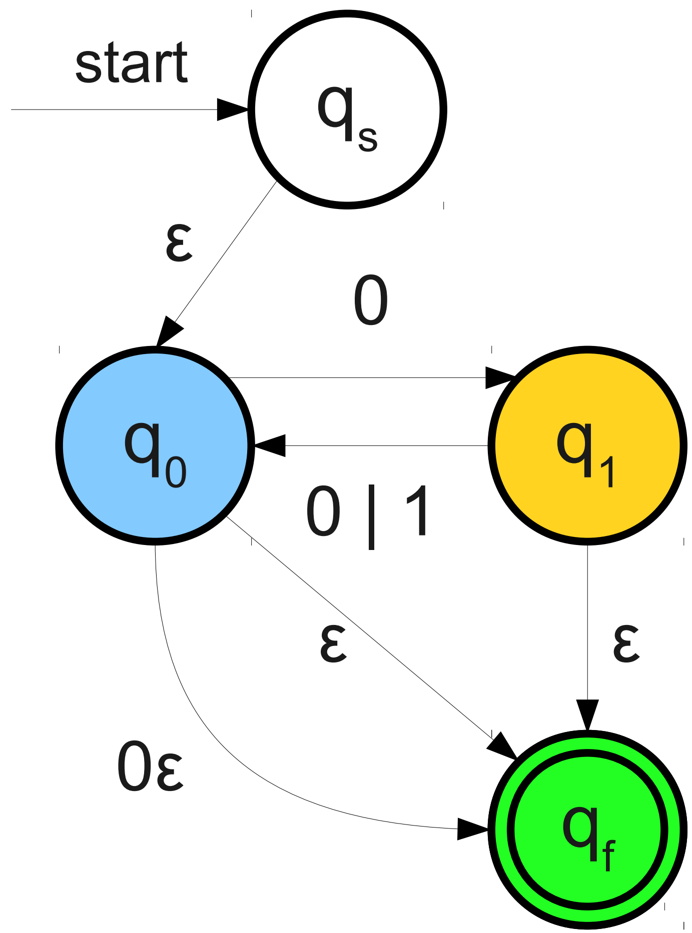
From NFAs to Regular Expressions



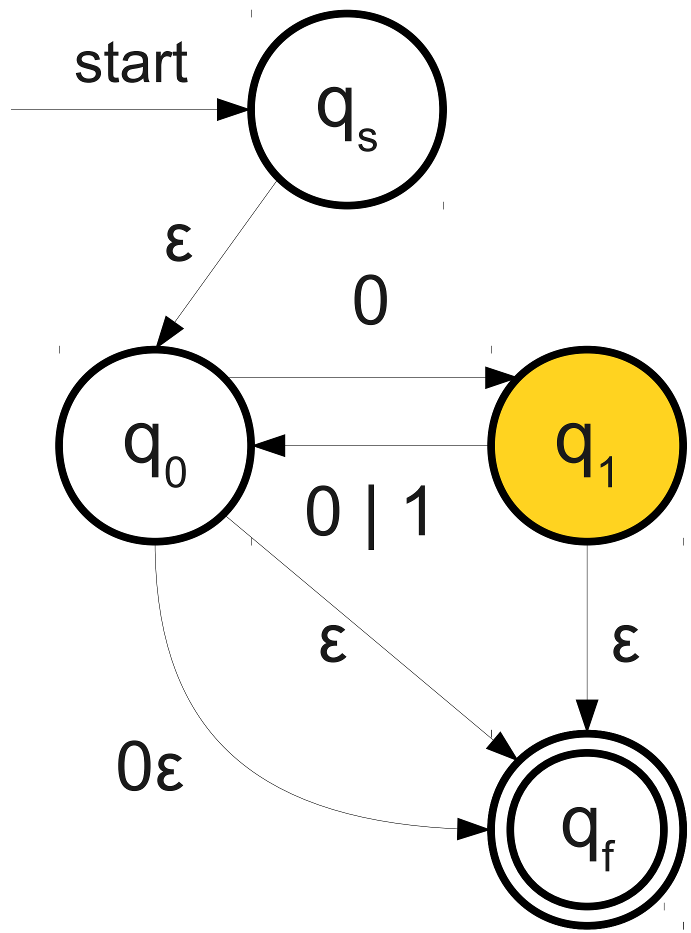
From NFAs to Regular Expressions



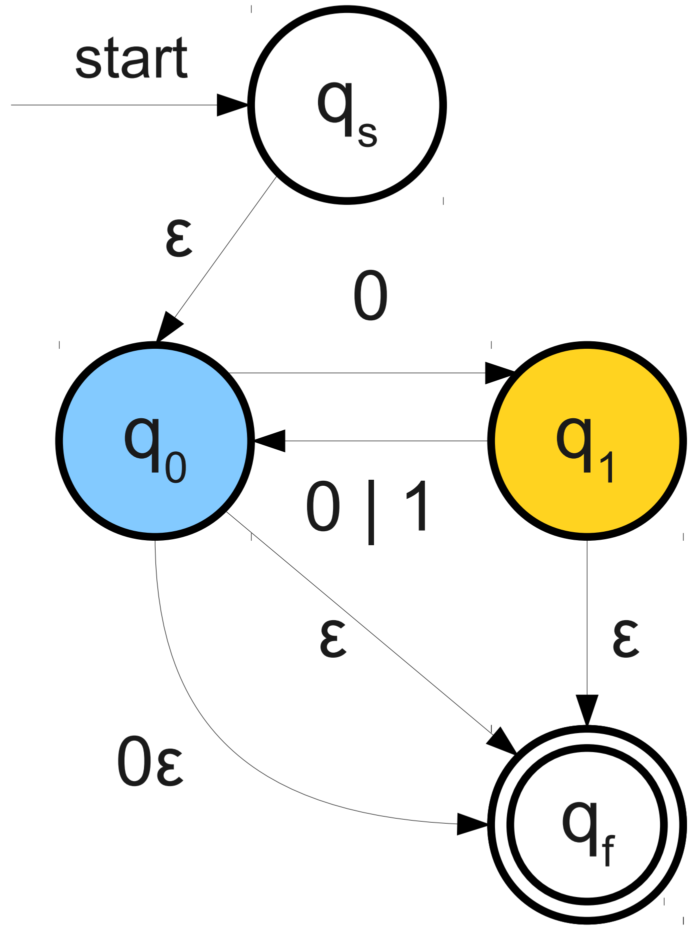
From NFAs to Regular Expressions



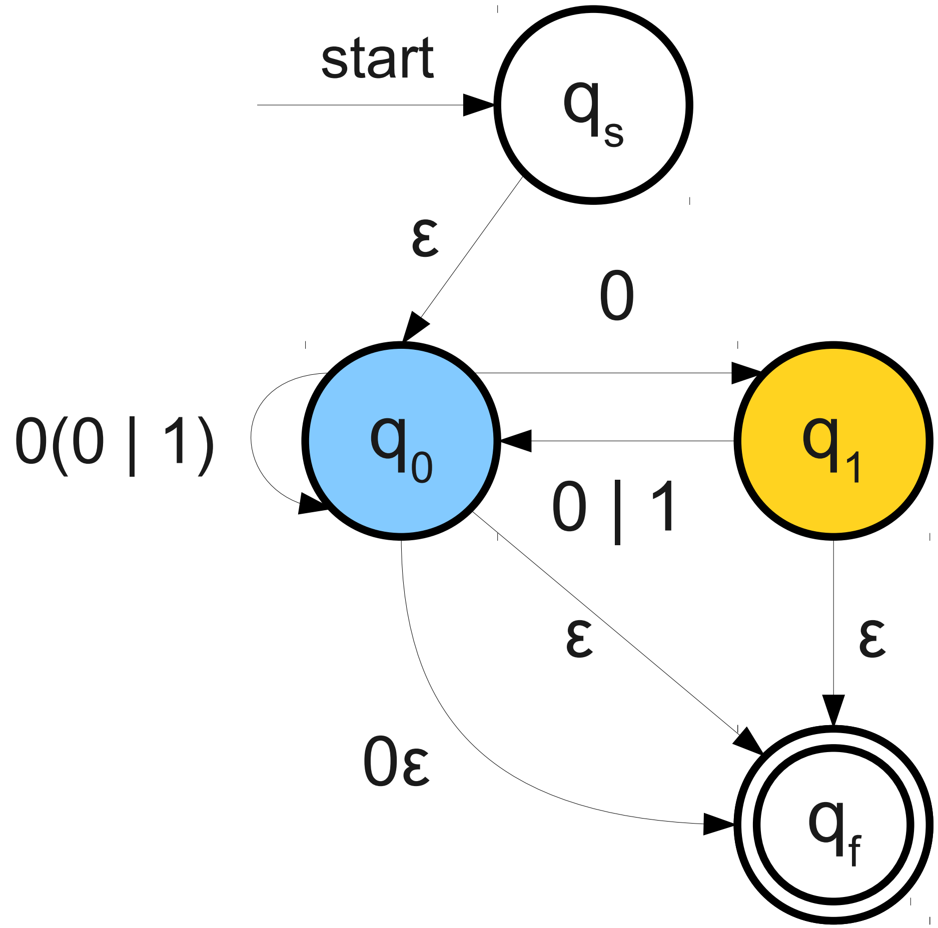
From NFAs to Regular Expressions



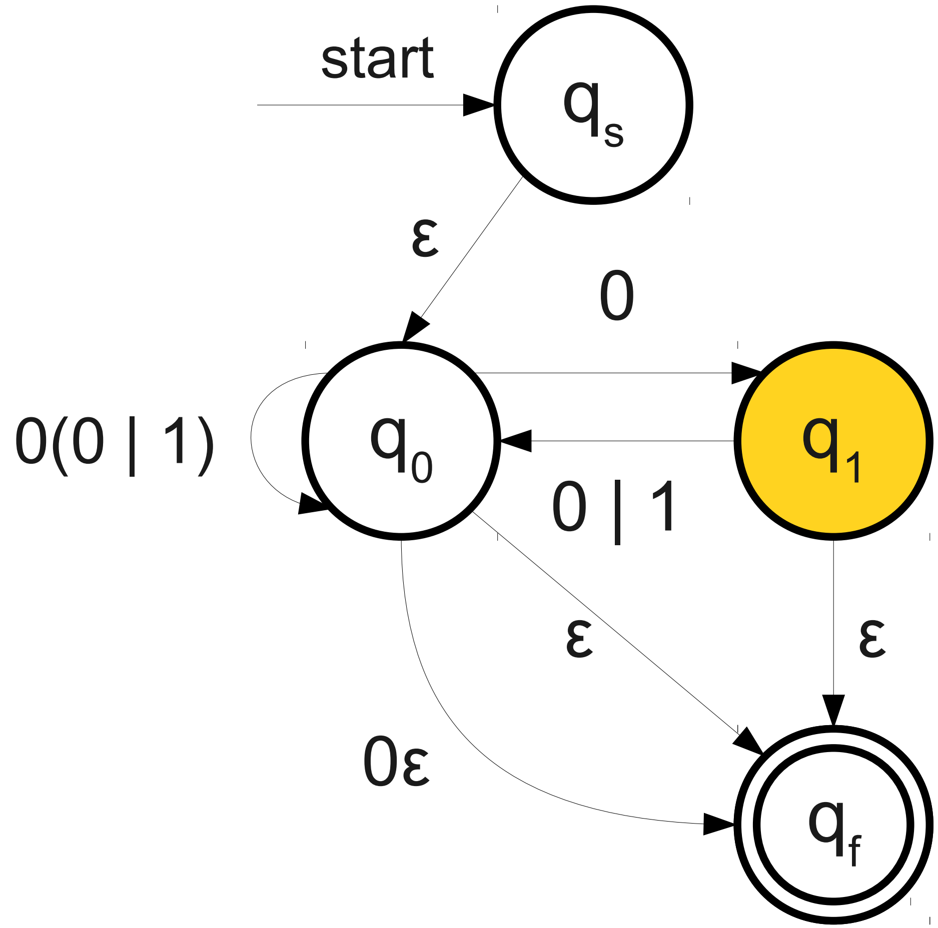
From NFAs to Regular Expressions



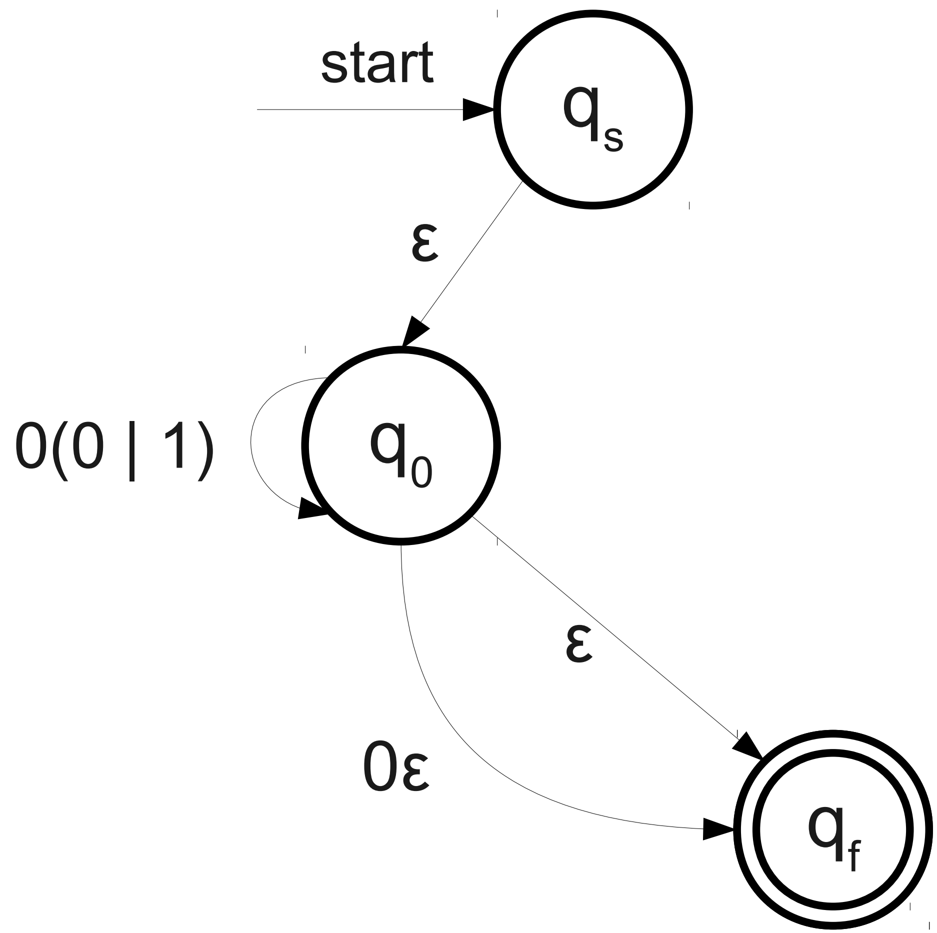
From NFAs to Regular Expressions



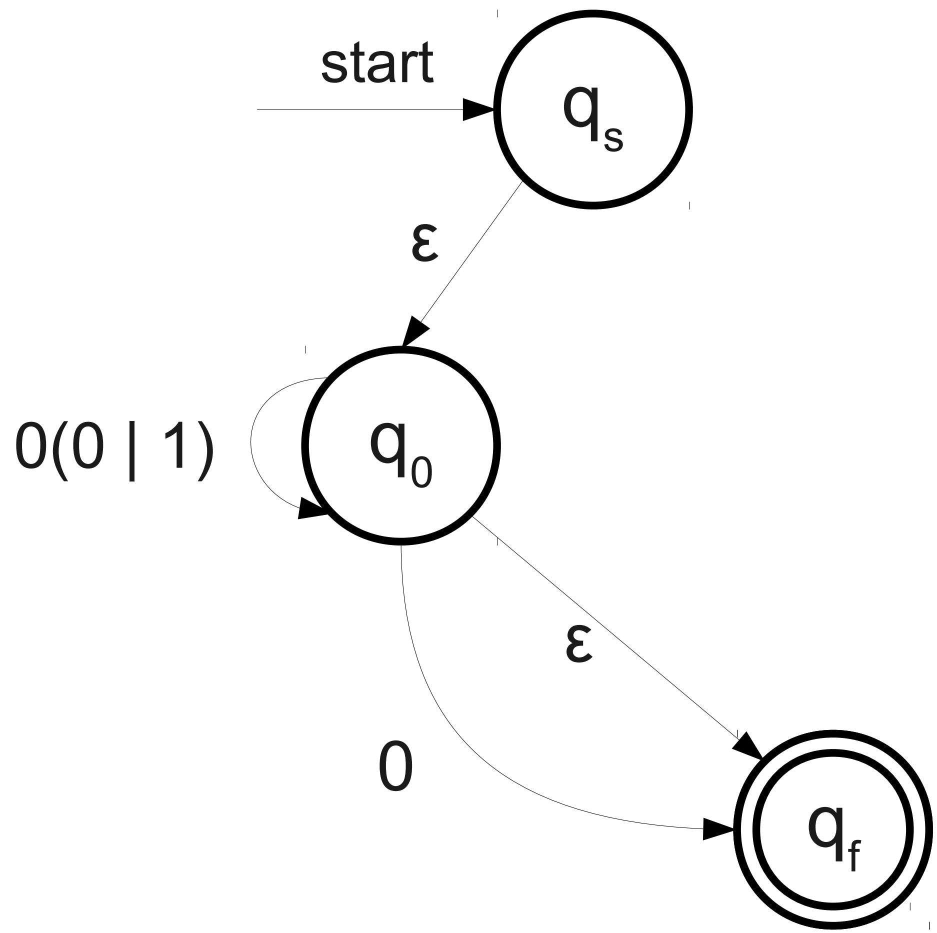
From NFAs to Regular Expressions



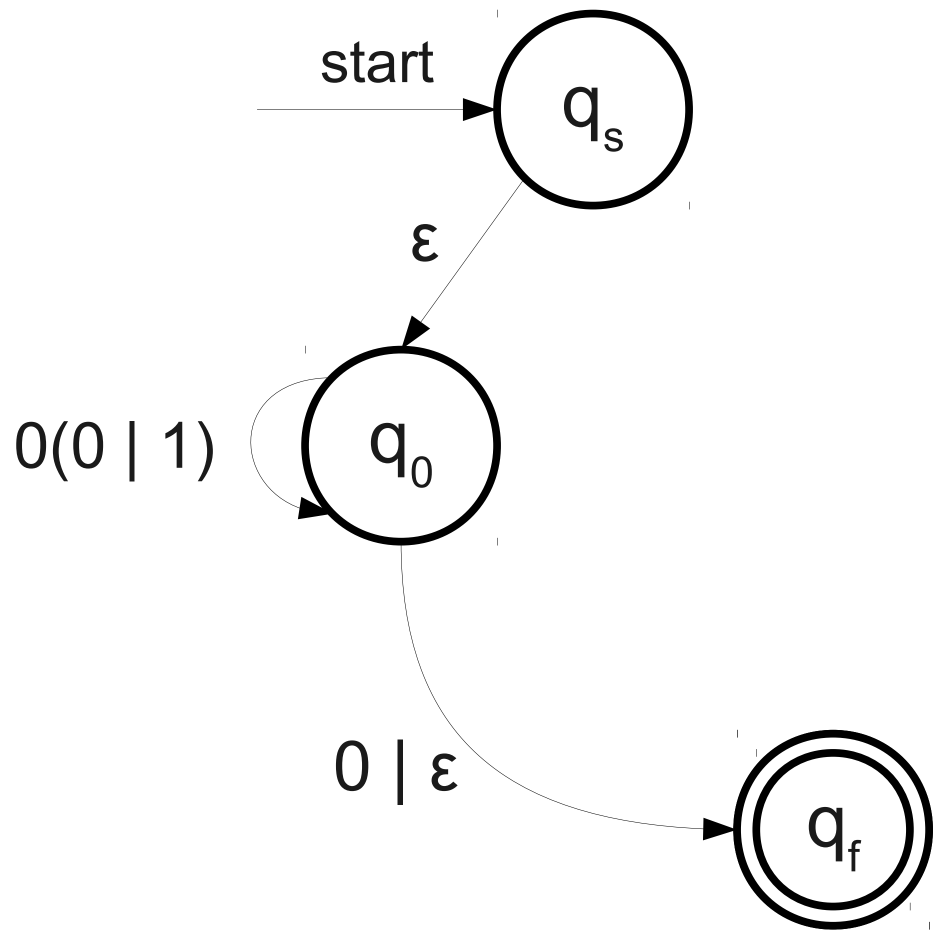
From NFAs to Regular Expressions



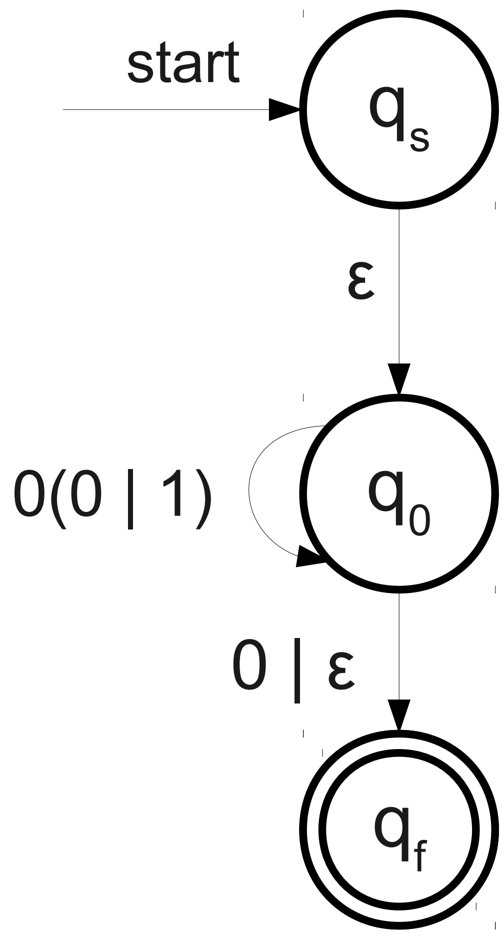
From NFAs to Regular Expressions



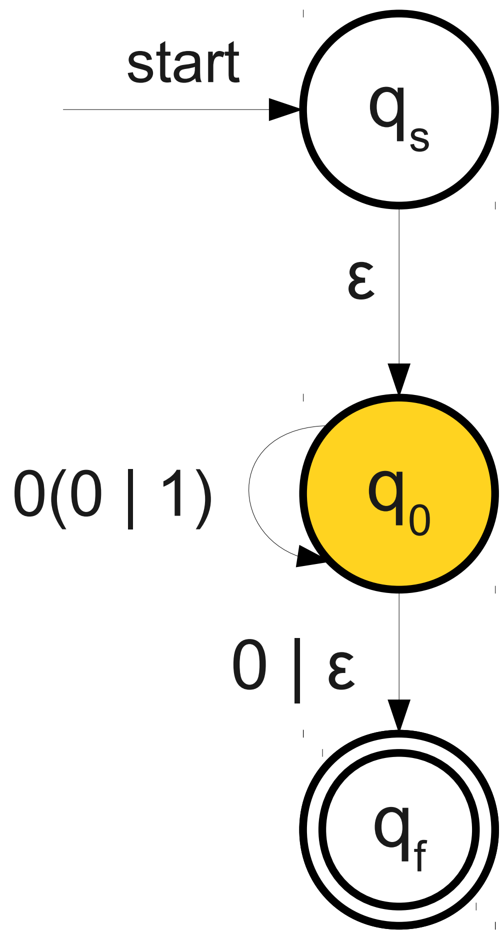
From NFAs to Regular Expressions



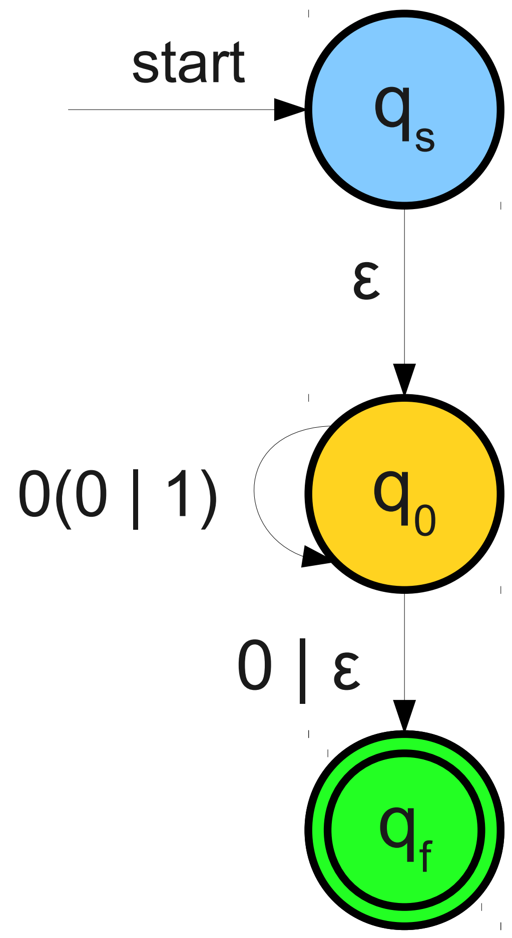
From NFAs to Regular Expressions



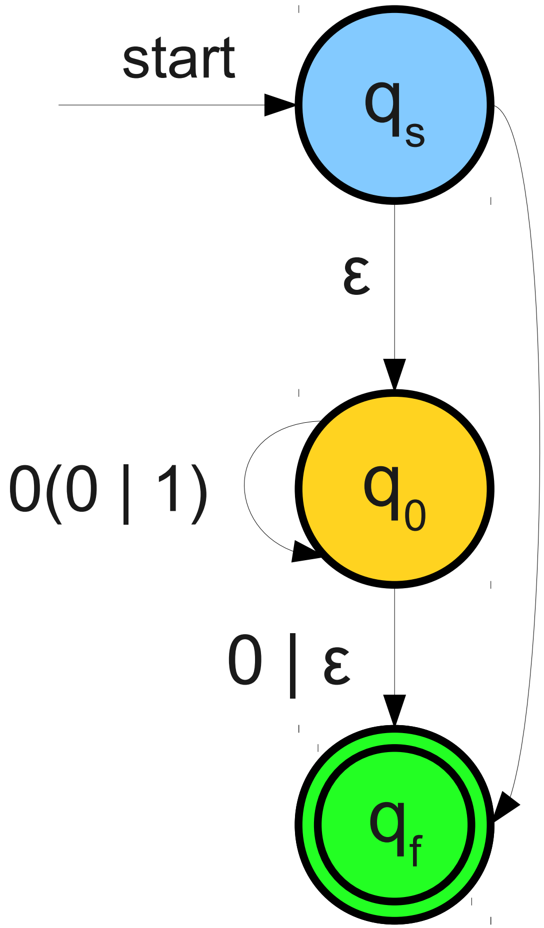
From NFAs to Regular Expressions



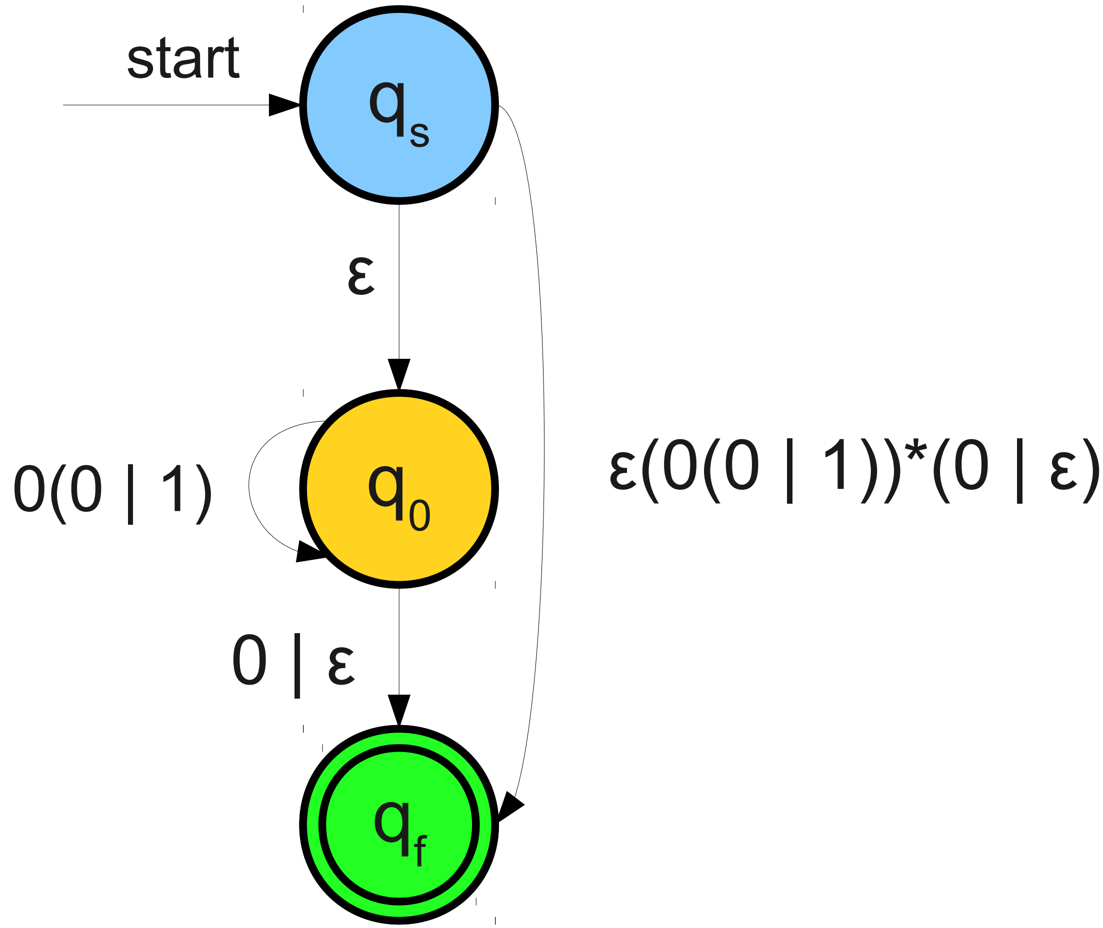
From NFAs to Regular Expressions



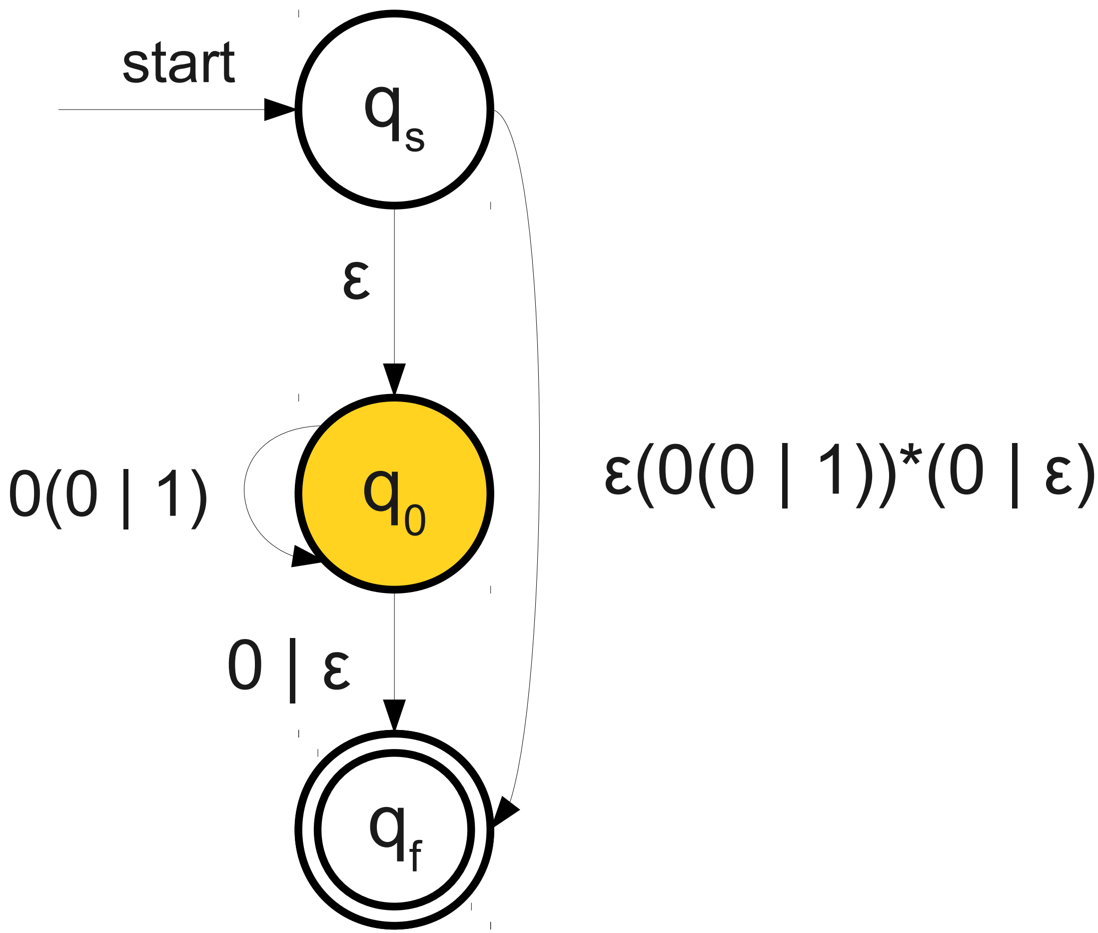
From NFAs to Regular Expressions



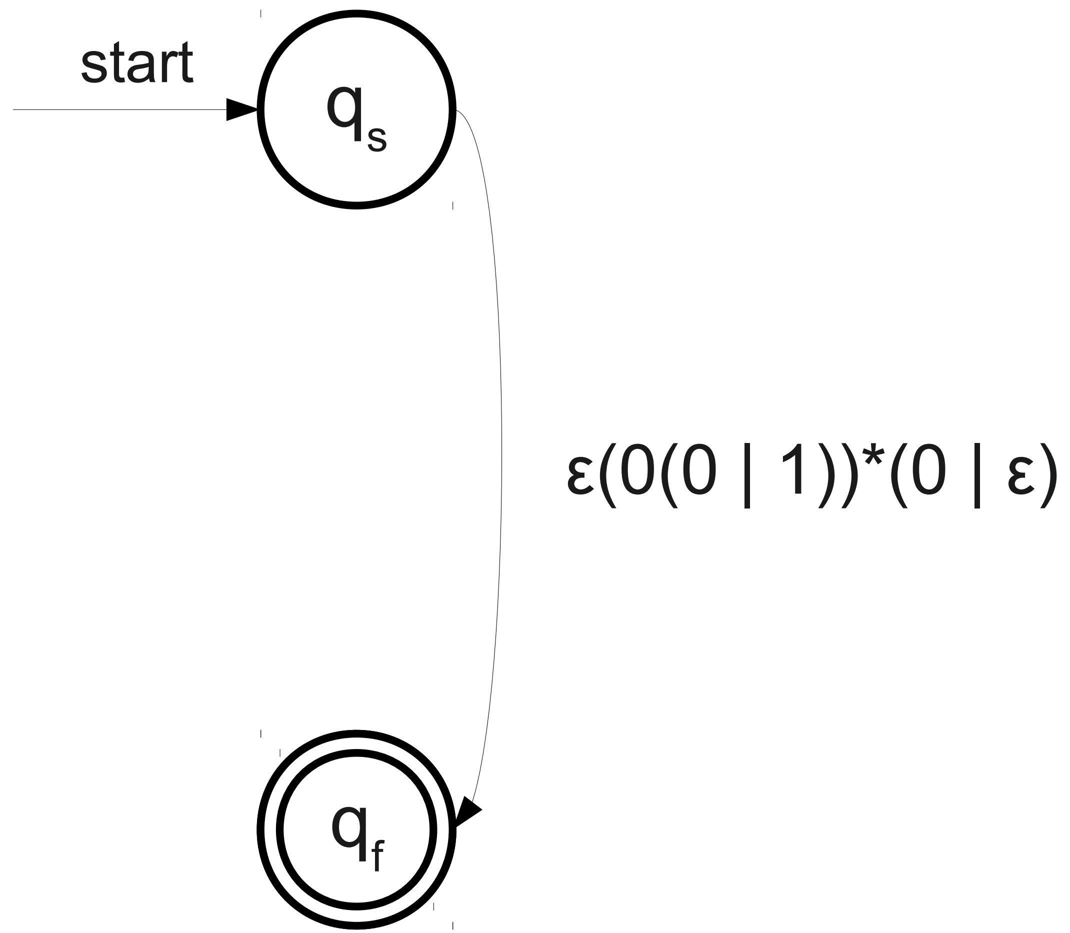
From NFAs to Regular Expressions



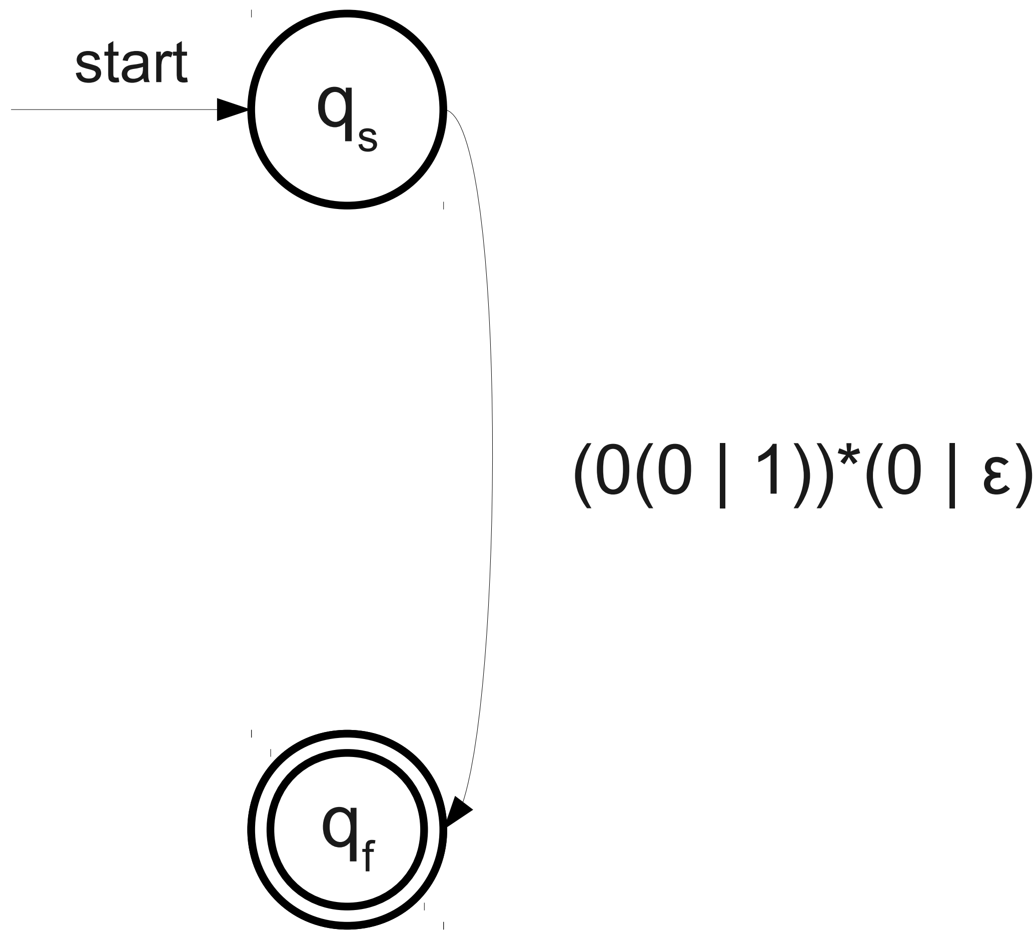
From NFAs to Regular Expressions



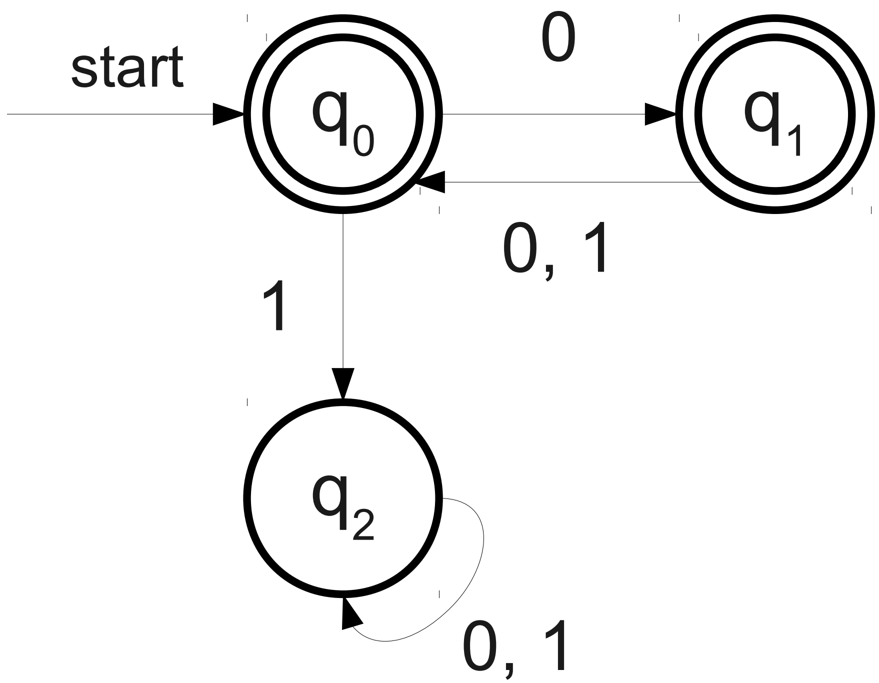
From NFAs to Regular Expressions



From NFAs to Regular Expressions



From NFAs to Regular Expressions



$(0(0 \mid 1))^*(0 \mid \varepsilon)$

Appendix: Homomorphisms of Regular Languages

Homomorphisms of Regular Languages

- Consider the language defined by the regular expression `(0120)*` and the function
  - $h($`0`$) =$ `n`
  - $h($`1`$) =$ `y`
  - $h($`2`$) =$ `a`
- Then $h^*($`(0120)*`$) =$ `((n)(y)(a)(n))*`

Homomorphisms of Regular Languages

- Consider the language `011*` and the function
  - h(`0`) = `Here`
  - h(`1`) = `Kitty`
- Then h^*(`011*`) = `(Here)(Kitty)(Kitty)*`