# Lecture 3

Ambiguity of automata

## Number of accepting runs

$\mathcal{A}$ an automaton (finite or weighted)

# Number of accepting runs

$\mathcal{A}$ an automaton (finite or weighted)

Important: $Q$ states, $I, F \subseteq Q$ initial and final states

(for weighted initial, final are $\{q \mid I(q) \neq \mathbb{0}\}, \quad \{q \mid F(q) \neq \mathbb{0}\}$)

# Number of accepting runs

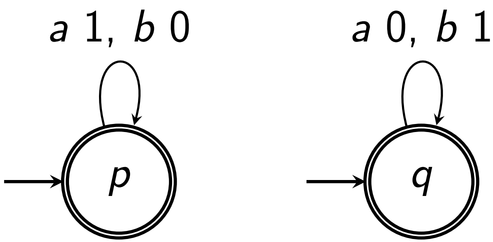
$\mathcal{A}$ an automaton (finite or weighted)

Important: $Q$ states, $I, F \subseteq Q$ initial and final states

(for weighted initial, final are $\{q \mid I(q) \neq \mathbb{0}\}$, $\{q \mid F(q) \neq \mathbb{0}\}$)

How many accepting runs are there for each word?

- Maximum of number of $a$'s and number of $b$'s



# Number of accepting runs

$\mathcal{A}$ an automaton (finite or weighted)

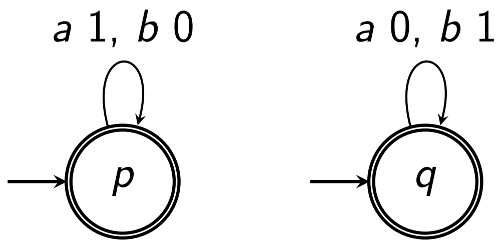
Important: $Q$ states, $I, F \subseteq Q$ initial and final states

(for weighted initial, final are $\{q \mid I(q) \neq \mathbb{0}\}, \quad \{q \mid F(q) \neq \mathbb{0}\}$)

How many accepting runs are there for each word?

- Maximum of number of $a$'s and number of $b$'s

  2 runs



# Number of accepting runs

$\mathcal{A}$ an automaton (finite or weighted)

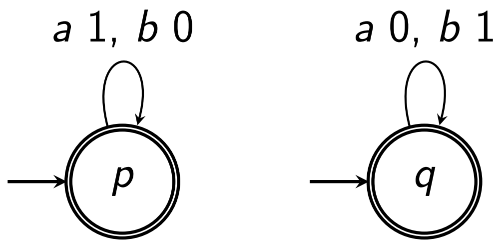
Important: $Q$ states, $I, F \subseteq Q$ initial and final states

(for weighted initial, final are $\{q \mid I(q) \neq \mathbb{0}\}, \quad \{q \mid F(q) \neq \mathbb{0}\}$)

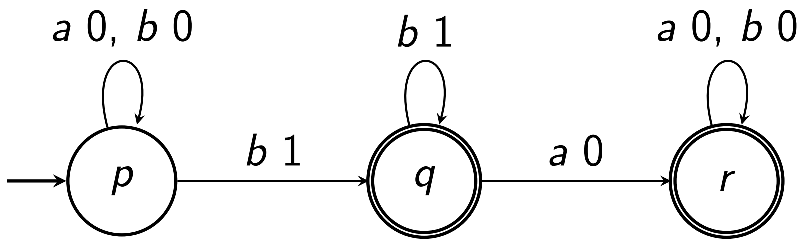
How many accepting runs are there for each word?

- Maximum of number of $a$'s and number of $b$'s

  2 runs




- Longest block of $b$'s



## Number of accepting runs

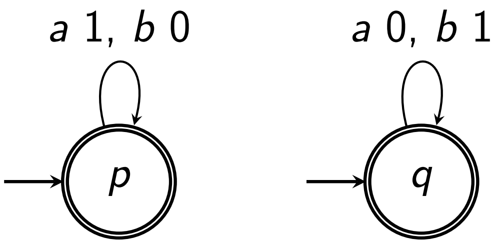
$\mathcal{A}$ an automaton (finite or weighted)

Important: $Q$ states, $I, F \subseteq Q$ initial and final states
(for weighted initial, final are $\{q \mid I(q) \neq \mathbb{0}\}$, $\{q \mid F(q) \neq \mathbb{0}\}$)

How many accepting runs are there for each word?

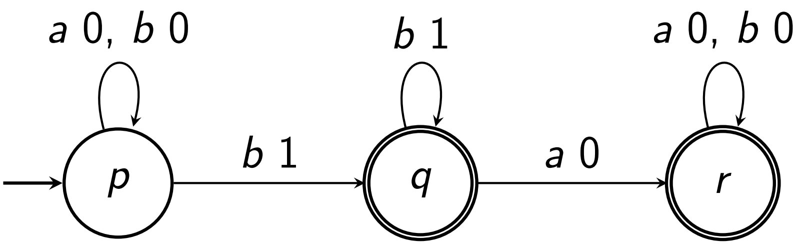
- Maximum of number of $a$'s and number of $b$'s

  2 runs




- Longest block of $b$'s

  $\mathcal{O}(|w|)$ runs



## Ambiguity subclasses

For an automaton $\mathcal{A}$ and a word $w$

we write $Acc(w)$ for the set of accepting runs of $\mathcal{A}$ on $w$

## Ambiguity subclasses

For an automaton $\mathcal{A}$ and a word $w$

we write $Acc(w)$ for the set of accepting runs of $\mathcal{A}$ on $w$

An automaton is:

- Finitely ambiguous if there is a $k$ such that $|Acc(w)| \leqslant k$ for all $w$

  We say it is $k$-ambiguous

  For $k = 1$ we say it is unambiguous

## Ambiguity subclasses

For an automaton $\mathcal{A}$ and a word $w$

we write $Acc(w)$ for the set of accepting runs of $\mathcal{A}$ on $w$

An automaton is:

- Finitely ambiguous if there is a $k$ such that $|Acc(w)| \leqslant k$ for all $w$

  We say it is $k$-ambiguous

  For $k = 1$ we say it is unambiguous

  Maximum of number of $a$'s and number of $b$'s: 2-ambiguous

# Ambiguity subclasses

For an automaton $\mathcal{A}$ and a word $w$

we write $Acc(w)$ for the set of accepting runs of $\mathcal{A}$ on $w$

An automaton is:

- Finitely ambiguous if there is a $k$ such that $|Acc(w)| \leqslant k$ for all $w$

  We say it is $k$-ambiguous

  For $k = 1$ we say it is unambiguous

  Maximum of number of $a$'s and number of $b$'s: 2-ambiguous

- Polynomially ambiguous if there is a polynomial $p$ s.t.

  $|Acc(w)| \leqslant p(|w|)$ for all $w$

  We say it is linearly ambiguous if the degree of $p$ is 1, etc...

# Ambiguity subclasses

For an automaton $\mathcal{A}$ and a word $w$

we write $Acc(w)$ for the set of accepting runs of $\mathcal{A}$ on $w$

An automaton is:

- Finitely ambiguous if there is a $k$ such that $|Acc(w)| \leqslant k$ for all $w$

  We say it is $k$-ambiguous

  For $k = 1$ we say it is unambiguous

  Maximum of number of $a$'s and number of $b$'s: 2-ambiguous

- Polynomially ambiguous if there is a polynomial $p$ s.t.

  $|Acc(w)| \leqslant p(|w|)$ for all $w$

  We say it is linearly ambiguous if the degree of $p$ is 1, etc...

  Longest block of $b$'s: linearly ambiguous

## Hierarchy of classes for weighted automata

Weighted automata (WA)

$\cup\!|$

Polynomially ambiguous WA

$\cup\!|$

Finitely ambiguous WA

$\cup\!|$

Unambiguous WA

$\cup\!|$

Deterministic WA

## Hierarchy of classes for weighted automata

- Are the inclusions strict?

$$
\begin{array}{c}
\text{Weighted automata (WA)} \\
\text{\rotatebox{90}{$\subseteq$}} \\
\text{Polynomially ambiguous WA} \\
\text{\rotatebox{90}{$\subseteq$}} \\
\text{Finitely ambiguous WA} \\
\text{\rotatebox{90}{$\subseteq$}} \\
\text{Unambiguous WA} \\
\text{\rotatebox{90}{$\subseteq$}} \\
\text{Deterministic WA}
\end{array}
$$

# Hierarchy of classes for weighted automata

- Are the inclusions strict?

  Depends on the semiring

$$
\begin{array}{c}
\text{Weighted automata (WA)} \\
\cup\!\!| \\
\text{Polynomially ambiguous WA} \\
\cup\!\!| \\
\text{Finitely ambiguous WA} \\
\cup\!\!| \\
\text{Unambiguous WA} \\
\cup\!\!| \\
\text{Deterministic WA}
\end{array}
$$

## Hierarchy of classes for weighted automata

- Are the inclusions strict?

  Depends on the semiring

- For the Boolean semiring

  It's all equivalent

Weighted automata (WA)

$\cup\!\mid$

Polynomially ambiguous WA

$\cup\!\mid$

Finitely ambiguous WA

$\cup\!\mid$

Unambiguous WA

$\cup\!\mid$

Deterministic WA

## Hierarchy of classes for weighted automata

- Are the inclusions strict?

  Depends on the semiring

- For the Boolean semiring

  It's all equivalent

- Next week we will focus on

  $(\mathbb{Q}, +, \cdot, 0, 1)$

  $(\mathbb{N}_{+\infty}, \min, +, \infty, 0)$

  $(\mathbb{N}_{-\infty}, \max, +, -\infty, 0)$

Weighted automata (WA)

$\cup\!|$

Polynomially ambiguous WA

$\cup\!|$

Finitely ambiguous WA

$\cup\!|$

Unambiguous WA

$\cup\!|$

Deterministic WA

# Hierarchy of classes for weighted automata

- Are the inclusions strict?

  Depends on the semiring

- For the Boolean semiring

  It's all equivalent

- Next week we will focus on

  $(\mathbb{Q}, +, \cdot, 0, 1)$

  $(\mathbb{N}_{+\infty}, \text{min}, +, \infty, 0)$

  $(\mathbb{N}_{-\infty}, \text{max}, +, -\infty, 0)$

- We will focus mostly on two classes

Weighted automata (WA)

$\cup\!|$

Polynomially ambiguous WA

$\cup\!|$

Finitely ambiguous WA

$\cup\!|$

Unambiguous WA

$\cup\!|$

Deterministic WA

# Trimmed automata

## Definition

An automaton $\mathcal{A}$ is trimmed if for every $q \in Q$ there is an initial state $p \in I$ and a final state $r \in F$ s.t. there is a run from $p$ to $q$ and a run from $q$ to $r$.

# Trimmed automata

## Definition

An automaton $\mathcal{A}$ is trimmed if for every $q \in Q$ there is an initial state $p \in I$ and a final state $r \in F$ s.t. there is a run from $p$ to $q$ and a run from $q$ to $r$.

## Remark

By removing states an automaton can be trimmed to an equivalent automaton.

# Trimmed automata

## Definition

An automaton $\mathcal{A}$ is trimmed if for every $q \in Q$ there is an initial state $p \in I$ and a final state $r \in F$ s.t. there is a run from $p$ to $q$ and a run from $q$ to $r$.

## Remark

By removing states an automaton can be trimmed to an equivalent automaton.

**Proof.**

Every run that goes through one of the removed states has value $\mathbb{0}$.

## Trimmed automata

### Definition

An automaton $\mathcal{A}$ is trimmed if for every $q \in Q$ there is an initial state $p \in I$ and a final state $r \in F$ s.t. there is a run from $p$ to $q$ and a run from $q$ to $r$.

### Remark

By removing states an automaton can be trimmed to an equivalent automaton.

**Proof.**

Every run that goes through one of the removed states has value $\mathbb{0}$.

- We will always implicitly assume that automata are trimmed.

# Trimmed automata

## Definition

An automaton $\mathcal{A}$ is trimmed if for every $q \in Q$ there is an initial state $p \in I$ and a final state $r \in F$ s.t. there is a run from $p$ to $q$ and a run from $q$ to $r$.

## Remark

By removing states an automaton can be trimmed to an equivalent automaton.

**Proof.**

Every run that goes through one of the removed states has value $\mathbb{0}$.

- We will always implicitly assume that automata are trimmed.

We write $p \xrightarrow{w} q$ if there is a run from $p$ to $q$ on word $w$.

## Finitely ambiguous class

How to check if $\mathcal{A}$ is finitely ambiguous?

## Finitely ambiguous class

How to check if $\mathcal{A}$ is finitely ambiguous?

**Theorem** (Weber, Seidl 1991)

(1) $\mathcal{A}$ is not finitely ambiguous if and only if

(2) there are two states $p \neq q \in Q$ and a word $w$ s.t.

$$p \xrightarrow{w} p,\ p \xrightarrow{w} q \text{ and } q \xrightarrow{w} q$$

## Finitely ambiguous class

How to check if $\mathcal{A}$ is finitely ambiguous?

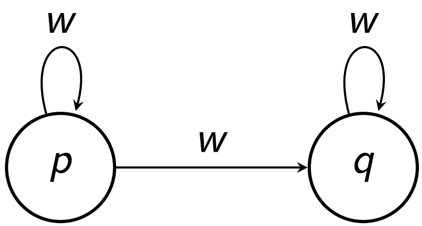
**Theorem** (Weber, Seidl 1991)

(1) $\mathcal{A}$ is not finitely ambiguous if and only if

(2) there are two states $p \neq q \in Q$ and a word $w$ s.t.

$$p \xrightarrow{w} p,\ p \xrightarrow{w} q \text{ and } q \xrightarrow{w} q$$

- We're looking for this pattern



## Finitely ambiguous class

How to check if $\mathcal{A}$ is finitely ambiguous?

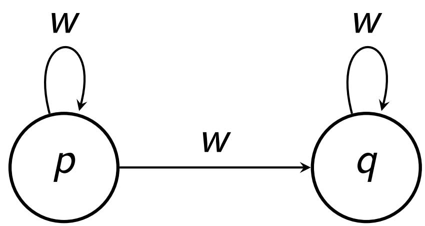
**Theorem** (Weber, Seidl 1991)

(1) $\mathcal{A}$ is not finitely ambiguous if and only if

(2) there are two states $p \neq q \in Q$ and a word $w$ s.t.

$$p \xrightarrow{w} p,\ p \xrightarrow{w} q \text{ and } q \xrightarrow{w} q$$

- We're looking for this pattern




**Proof. (2) $\implies$ (1)**

There is $a \in I$ and $v_a$ s.t. $a \xrightarrow{v_a} p$ and $b \in F$ and $v_b$ s.t. $q \xrightarrow{v_b} b$

# Finitely ambiguous class

How to check if $\mathcal{A}$ is finitely ambiguous?

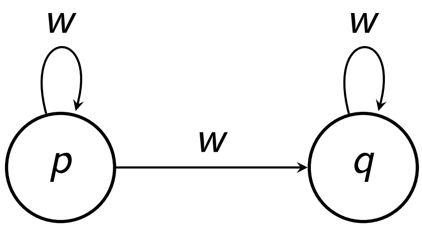
**Theorem** (Weber, Seidl 1991)

(1) $\mathcal{A}$ is not finitely ambiguous if and only if

(2) there are two states $p \neq q \in Q$ and a word $w$ s.t.

$$p \xrightarrow{w} p,\ p \xrightarrow{w} q \text{ and } q \xrightarrow{w} q$$

- We're looking for this pattern




**Proof. (2) $\implies$ (1)**

There is $a \in I$ and $v_a$ s.t. $a \xrightarrow{v_a} p$ and $b \in F$ and $v_b$ s.t. $q \xrightarrow{v_b} b$

Then $|Acc(v_a w^n v_b)| \geqslant n - 1$.

# Not finitely ambiguous $\implies$ pattern (1)

- A strongly connected component is $Q_i \subseteq Q$ s.t. for all $p, q \in Q$ there are $v_1, v_2$: $p \xrightarrow{v_1} q$ and $q \xrightarrow{v_2} p$

# Not finitely ambiguous $\implies$ pattern (1)

- A strongly connected component is $Q_i \subseteq Q$ s.t. for all $p, q \in Q$ there are $v_1, v_2$: $p \xrightarrow{v_1} q$ and $q \xrightarrow{v_2} p$

## Lemma

$Q = Q_1 \cup Q_2 \ldots \cup Q_m$, where $Q_i$ are strongly connected components. Moreover, if $p \in Q_i$ and $q \in Q_j$ and there is a word $w$ s.t. $p \xrightarrow{w} q$ then $i \leqslant j$.

# Not finitely ambiguous $\implies$ pattern (1)

- A strongly connected component is $Q_i \subseteq Q$ s.t. for all $p, q \in Q$ there are $v_1, v_2$: $p \xrightarrow{v_1} q$ and $q \xrightarrow{v_2} p$

## Lemma

$Q = Q_1 \cup Q_2 \ldots \cup Q_m$, where $Q_i$ are strongly connected components. Moreover, if $p \in Q_i$ and $q \in Q_j$ and there is a word $w$ s.t. $p \xrightarrow{w} q$ then $i \leqslant j$.

## Proof.

We put two states $p, q$ in the same $Q_a \subseteq Q$ iff there is $p \xrightarrow{v_1} q$ and $q \xrightarrow{v_2} p$.

# Not finitely ambiguous $\implies$ pattern (1)

- A strongly connected component is $Q_i \subseteq Q$ s.t. for all $p, q \in Q$ there are $v_1, v_2$: $p \xrightarrow{v_1} q$ and $q \xrightarrow{v_2} p$

## Lemma

$Q = Q_1 \cup Q_2 \ldots \cup Q_m$, where $Q_i$ are strongly connected components. Moreover, if $p \in Q_i$ and $q \in Q_j$ and there is a word $w$ s.t. $p \xrightarrow{w} q$ then $i \leqslant j$.

## Proof.

We put two states $p, q$ in the same $Q_a \subseteq Q$ iff there is $p \xrightarrow{v_1} q$ and $q \xrightarrow{v_2} p$.
Let $Q_a$ for $a \in A$ be the set of subsets. Define graph $G$ with edges $Q_a \to Q_b$ if there are $p \in Q_a$ and $q \in Q_b$ s.t. $p \xrightarrow{v} q$.

## Not finitely ambiguous $\implies$ pattern (1)

- A strongly connected component is $Q_i \subseteq Q$ s.t. for all $p, q \in Q$ there are $v_1, v_2$: $p \xrightarrow{v_1} q$ and $q \xrightarrow{v_2} p$

### Lemma

$Q = Q_1 \cup Q_2 \ldots \cup Q_m$, where $Q_i$ are strongly connected components. Moreover, if $p \in Q_i$ and $q \in Q_j$ and there is a word $w$ s.t. $p \xrightarrow{w} q$ then $i \leqslant j$.

**Proof.**

We put two states $p, q$ in the same $Q_a \subseteq Q$ iff there is $p \xrightarrow{v_1} q$ and $q \xrightarrow{v_2} p$.

Let $Q_a$ for $a \in A$ be the set of subsets. Define graph $G$ with edges $Q_a \to Q_b$ if there are $p \in Q_a$ and $q \in Q_b$ s.t. $p \xrightarrow{v} q$.

$G$ is closed under transitivity and antisymmetric

## Not finitely ambiguous $\implies$ pattern (1)

- A strongly connected component is $Q_i \subseteq Q$ s.t. for all $p, q \in Q$ there are $v_1, v_2$: $p \xrightarrow{v_1} q$ and $q \xrightarrow{v_2} p$

### Lemma

$Q = Q_1 \cup Q_2 \ldots \cup Q_m$, where $Q_i$ are strongly connected components. Moreover, if $p \in Q_i$ and $q \in Q_j$ and there is a word $w$ s.t. $p \xrightarrow{w} q$ then $i \leqslant j$.

### Proof.

We put two states $p, q$ in the same $Q_a \subseteq Q$ iff there is $p \xrightarrow{v_1} q$ and $q \xrightarrow{v_2} p$.

Let $Q_a$ for $a \in A$ be the set of subsets. Define graph $G$ with edges $Q_a \to Q_b$ if there are $p \in Q_a$ and $q \in Q_b$ s.t. $p \xrightarrow{v} q$.

$G$ is closed under transitivity and antisymmetric

So sets $G$ is a DAG and $Q_a$ can be topologically sorted

■

# Not finitely ambiguous $\implies$ pattern (2)

## Not finitely ambiguous $\implies$ pattern (2)

- Case one: there is a $Q_i$, states $p', q' \in Q_i$ and a word $v$ s.t.
  there are two different runs $p' \xrightarrow{v} q'$

## Not finitely ambiguous $\implies$ pattern (2)

- Case one: there is a $Q_i$, states $p', q' \in Q_i$ and a word $v$ s.t.

  there are two different runs $p' \xrightarrow{v} q'$

  Let $v = v_1 v_2$ and $p \neq q$

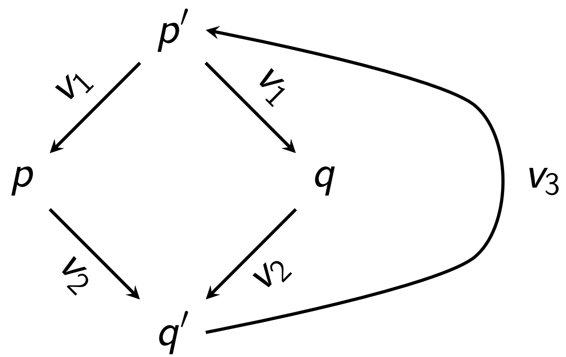
$$\begin{tikzcd}
 & p' \arrow[dl, "v_1"'] \arrow[dr, "v_1"] & \\
p \arrow[dr, "v_2"'] & & q \arrow[dl, "v_2"] \\
 & q' &
\end{tikzcd}$$

# Not finitely ambiguous $\implies$ pattern (2)

- Case one: there is a $Q_i$, states $p', q' \in Q_i$ and a word $v$ s.t.

  there are two different runs $p' \xrightarrow{v} q'$

  Let $v = v_1 v_2$ and $p \neq q$



# Not finitely ambiguous $\implies$ pattern (2)

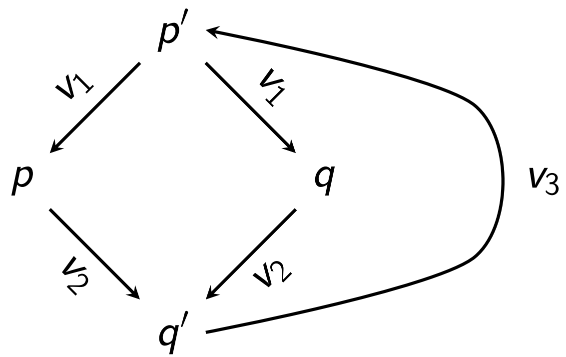
- Case one: there is a $Q_i$, states $p', q' \in Q_i$ and a word $v$ s.t.

  there are two different runs $p' \xrightarrow{v} q'$

  Let $v = v_1 v_2$ and $p \neq q$

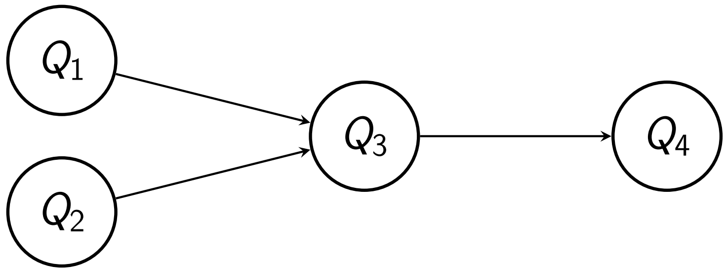
- Let $w = v_2 v_3 v_1$

  $p \xrightarrow{w} p$, $p \xrightarrow{w} q$, $q \xrightarrow{w} q$



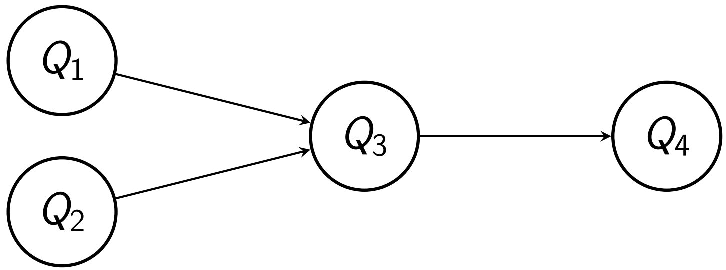
# Not finitely ambiguous $\implies$ pattern (3)

$Q = Q_1 \cup \ldots \cup Q_m$



## Not finitely ambiguous $\implies$ pattern (3)

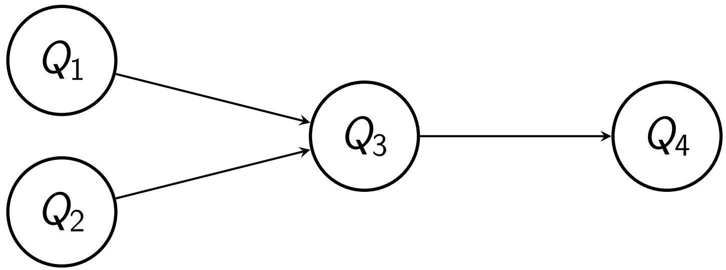
$Q = Q_1 \cup \ldots \cup Q_m$




- Notice that every accepting run starts in some $Q_i$ and ends in some $Q_j$ through some other $Q_{l_1}, \ldots, Q_{l_s}$

# Not finitely ambiguous $\implies$ pattern (3)

$Q = Q_1 \cup \ldots \cup Q_m$

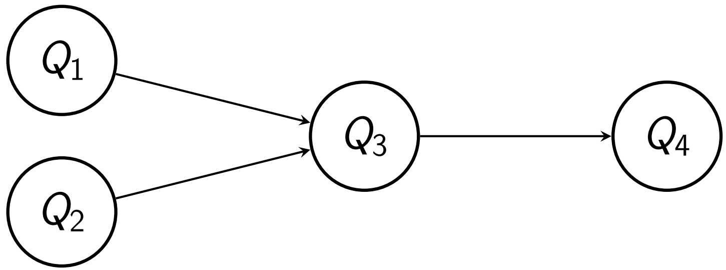



- Notice that every accepting run starts in some $Q_i$ and ends in some $Q_j$ through some other $Q_{l_1}, \ldots, Q_{l_s}$

  Example: From $Q_1$ to $Q_4$ through $Q_3$

## Not finitely ambiguous $\implies$ pattern (3)

$Q = Q_1 \cup \ldots \cup Q_m$



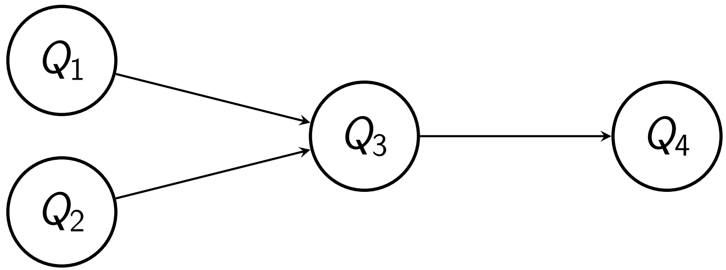

- Notice that every accepting run starts in some $Q_i$ and ends in some $Q_j$ through some other $Q_{l_1}, \ldots, Q_{l_s}$

  Example: From $Q_1$ to $Q_4$ through $Q_3$

- We can decompose the sets of accepting runs into $M \subseteq \{1, \ldots, m\}$ where $M = \{i_1, \ldots, i_s\}$ and $i_1 < i_2 < \ldots < i_s$ means that a run starts in $Q_{i_1}$ goes through $Q_{i_2}, \ldots Q_{i_{s-1}}$ to $Q_{i_s}$

# Not finitely ambiguous $\implies$ pattern (3)

$Q = Q_1 \cup \ldots \cup Q_m$




- Notice that every accepting run starts in some $Q_i$ and ends in some $Q_j$ through some other $Q_{l_1}, \ldots, Q_{l_s}$

  Example: From $Q_1$ to $Q_4$ through $Q_3$

- We can decompose the sets of accepting runs into $M \subseteq \{1, \ldots, m\}$ where $M = \{i_1, \ldots, i_s\}$ and $i_1 < i_2 < \ldots < i_s$ means that a run starts in $Q_{i_1}$ goes through $Q_{i_2}, \ldots Q_{i_{s-1}}$ to $Q_{i_s}$

- Note: the number of such $M$ is bounded by $2^{|Q|}$.

## Not finitely ambiguous $\implies$ pattern (4)

$Q = Q_1 \cup Q_2 \cup \ldots \cup Q_m$

When $M = \{i_1, \ldots, i_k\}$ write $Q_1, \ldots Q_k$ instead of $Q_{i_1}, \ldots Q_{i_k}$

# Not finitely ambiguous $\implies$ pattern (4)

$Q = Q_1 \cup Q_2 \cup \ldots \cup Q_m$

When $M = \{i_1, \ldots, i_k\}$ write $Q_1, \ldots Q_k$ instead of $Q_{i_1}, \ldots Q_{i_k}$

## Definition

Let $x = x_1 \ldots x_s \in \Sigma^*$. The graph $G_M(x) = (V, E)$ is defined as

$V = \{(q, j) \in (Q_1 \cup \ldots \cup Q_k) \times \{0, \ldots, s\} \mid$

$\exists q_I \in I \cap Q_1, q_F \in F \cap Q_k : q_I \xrightarrow{x_1 \ldots x_j} q, q \xrightarrow{x_{j+1} \ldots x_s} q_F\}$

$E = \{(p, j-1) \to (q, j) \mid p \xrightarrow{x_j} q\}$

## Not finitely ambiguous $\implies$ pattern (4)

$Q = Q_1 \cup Q_2 \cup \ldots \cup Q_m$

When $M = \{i_1, \ldots, i_k\}$ write $Q_1, \ldots Q_k$ instead of $Q_{i_1}, \ldots Q_{i_k}$

### Definition

Let $x = x_1 \ldots x_s \in \Sigma^*$. The graph $G_M(x) = (V, E)$ is defined as

$V = \{(q, j) \in (Q_1 \cup \ldots \cup Q_k) \times \{0, \ldots, s\} \mid$

$\exists q_I \in I \cap Q_1, q_F \in F \cap Q_k : q_I \xrightarrow{x_1 \ldots x_j} q, q \xrightarrow{x_{j+1} \ldots x_s} q_F\}$

$E = \{(p, j-1) \to (q, j) \mid p \xrightarrow{x_j} q\}$

- Case 2: for all $i$ if $p', q' \in Q_i$ then for every $v$ at most one run $p' \xrightarrow{v} q'$

# Not finitely ambiguous $\implies$ pattern (4)

$Q = Q_1 \cup Q_2 \cup \ldots \cup Q_m$

When $M = \{i_1, \ldots, i_k\}$ write $Q_1, \ldots Q_k$ instead of $Q_{i_1}, \ldots Q_{i_k}$

## Definition

Let $x = x_1 \ldots x_s \in \Sigma^*$. The graph $G_M(x) = (V, E)$ is defined as

$V = \{(q, j) \in (Q_1 \cup \ldots \cup Q_k) \times \{0, \ldots, s\} \mid$

$\exists q_I \in I \cap Q_1, q_F \in F \cap Q_k : q_I \xrightarrow{x_1 \ldots x_j} q, q \xrightarrow{x_{j+1} \ldots x_s} q_F\}$

$E = \{(p, j-1) \to (q, j) \mid p \xrightarrow{x_j} q\}$

- Case 2: for all $i$ if $p', q' \in Q_i$ then for every $v$ at most one run $p' \xrightarrow{v} q'$

If $\mathcal{A}$ is not finitely ambiguous there is a word $x_1 \ldots x_s$ s.t. $Acc(x) \geqslant N$

For any $N$ (in the end we choose $N$ big enough)

# Not finitely ambiguous $\implies$ pattern (4)

$Q = Q_1 \cup Q_2 \cup \ldots \cup Q_m$

When $M = \{i_1, \ldots, i_k\}$ write $Q_1, \ldots Q_k$ instead of $Q_{i_1}, \ldots Q_{i_k}$

## Definition

Let $x = x_1 \ldots x_s \in \Sigma^*$. The graph $G_M(x) = (V, E)$ is defined as

$V = \{(q, j) \in (Q_1 \cup \ldots \cup Q_k) \times \{0, \ldots, s\} \mid$

$\exists q_I \in I \cap Q_1, q_F \in F \cap Q_k : q_I \xrightarrow{x_1 \ldots x_j} q, q \xrightarrow{x_{j+1} \ldots x_s} q_F\}$

$E = \{(p, j-1) \to (q, j) \mid p \xrightarrow{x_j} q\}$

- Case 2: for all $i$ if $p', q' \in Q_i$ then for every $v$ at most one run $p' \xrightarrow{v} q'$

If $\mathcal{A}$ is not finitely ambiguous there is a word $x_1 \ldots x_s$ s.t. $Acc(x) \geqslant N$

For any $N$ (in the end we choose $N$ big enough)

Then there is an $M$ s.t. the number of accepting runs by $M$ is at least $\frac{N}{2^{|Q|}}$

# Not finitely ambiguous $\Longrightarrow$ pattern (5)

$M = \{i_1, \ldots i_k\}$, $\quad x = x_1 \ldots x_s$, $\quad Acc(x)$ from $Q_1$ to $Q_k$ big

$$Q_1 \longrightarrow Q_2 \longrightarrow \cdots \longrightarrow Q_k$$

## Not finitely ambiguous $\Longrightarrow$ pattern (5)

$M = \{i_1, \dots i_k\}$, $\quad x = x_1 \dots x_s$, $\quad Acc(x)$ from $Q_1$ to $Q_k$ big

$$Q_1 \longrightarrow Q_2 \longrightarrow \cdots \longrightarrow Q_k$$

- for all $l = 1, \dots, k-1$ let $D_l(x) \subseteq E$ s.t.

  $(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$

## Not finitely ambiguous $\implies$ pattern (5)

$M = \{i_1, \ldots i_k\}, \quad x = x_1 \ldots x_s, \quad Acc(x)$ from $Q_1$ to $Q_k$ big

$$\begin{tikzcd} Q_1 \arrow[r] & Q_2 \arrow[r] & \cdots \arrow[r] & Q_k \end{tikzcd}$$

- for all $l = 1, \ldots, k-1$ let $D_l(x) \subseteq E$ s.t.

  $(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$

- Number of accepting runs on $x$ is bounded by

$$|Q_1| \cdot |D_1| \cdot |D_2| \cdot \ldots \cdot |D_{k-1}| \cdot |Q_k|$$

# Not finitely ambiguous $\implies$ pattern (5)

$M = \{i_1, \dots i_k\}$, $\quad x = x_1 \dots x_s$, $\quad Acc(x)$ from $Q_1$ to $Q_k$ big

$$Q_1 \longrightarrow Q_2 \longrightarrow \cdots \longrightarrow Q_k$$

- for all $l = 1, \dots, k-1$ let $D_l(x) \subseteq E$ s.t.

  $(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$

- Number of accepting runs on $x$ is bounded by

$$|Q_1| \cdot |D_1| \cdot |D_2| \cdot \ldots \cdot |D_{k-1}| \cdot |Q_k|$$

- So there is $l$ s.t. $|D_l|$ big

## Not finitely ambiguous $\implies$ pattern (5)

$M = \{i_1, \dots i_k\}, \quad x = x_1 \dots x_s, \quad Acc(x)$ from $Q_1$ to $Q_k$ big

$$Q_1 \longrightarrow Q_2 \longrightarrow \cdots \longrightarrow Q_k$$

- for all $l = 1, \dots, k-1$ let $D_l(x) \subseteq E$ s.t.
  $(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$

- Number of accepting runs on $x$ is bounded by
$$|Q_1| \cdot |D_1| \cdot |D_2| \cdot \ldots \cdot |D_{k-1}| \cdot |Q_k|$$

- So there is $l$ s.t. $|D_l|$ big

- We choose $N$ s.t. $|D_l| > 2^{|Q|}$

## Not finitely ambiguous $\implies$ pattern (6)

$(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$,, $\quad |D_l| > 2^{|Q|}, \quad x = x_1 \ldots x_s$

# Not finitely ambiguous $\implies$ pattern (6)

$(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$,, $\quad |D_l| > 2^{|Q|}, \quad x = x_1 \dots x_s$

- Let $J \subseteq \{1, \dots, s\}$ so $D_l(x) = \{(p, j-1) \to (q, j) \mid j \in J\}$

# Not finitely ambiguous $\implies$ pattern (6)

$(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$,, $\quad |D_l| > 2^{|Q|}, \quad x = x_1 \dots x_s$

- Let $J \subseteq \{1, \dots, s\}$ so $D_l(x) = \{(p, j-1) \to (q, j) \mid j \in J\}$
- For every $j \in J$ let $A_j = \{r \mid (r, j) \in V\}$

## Not finitely ambiguous $\implies$ pattern (6)

$(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$,, $\quad |D_l| > 2^{|Q|}$, $\quad x = x_1 \dots x_s$

- Let $J \subseteq \{1, \dots, s\}$ so $D_l(x) = \{(p, j-1) \to (q, j) \mid j \in J\}$
- For every $j \in J$ let $A_j = \{r \mid (r, j) \in V\}$
- Since $|D_l| > 2^{|Q|}$ there exist $j_1 < j_2$ s.t. $A_{j_1} = A_{j_2}$.

  We write $A = A_{j_1} = A_{j_2}$, $\quad y_1 = x_{j_1+1} \dots x_{j_2-1}$, $\quad a_1 = x_{j_2}$ and $y = y_1 a_1$

## Not finitely ambiguous $\implies$ pattern (6)

$(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$,, $\quad |D_l| > 2^{|Q|}, \quad x = x_1 \dots x_s$

- Let $J \subseteq \{1, \dots, s\}$ so $D_l(x) = \{(p, j-1) \to (q, j) \mid j \in J\}$
- For every $j \in J$ let $A_j = \{r \mid (r, j) \in V\}$
- Since $|D_l| > 2^{|Q|}$ there exist $j_1 < j_2$ s.t. $A_{j_1} = A_{j_2}$.

  We write $A = A_{j_1} = A_{j_2}$, $\quad y_1 = x_{j_1+1} \dots x_{j_2-1}$, $\quad a_1 = x_{j_2}$ and $y = y_1 a_1$
- There are: $q_l \in Q_l, p_{l+1} \in Q_{l+1}$, $\quad$ s.t. $q_l \xrightarrow{a_1} p_{l+1}$ $\quad$ and $r \in A$ s.t. $r \xrightarrow{y_1} q_l$

# Not finitely ambiguous $\implies$ pattern (6)

$(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$,, $\quad |D_l| > 2^{|Q|}, \quad x = x_1 \dots x_s$

- Let $J \subseteq \{1, \dots, s\}$ so $D_l(x) = \{(p, j-1) \to (q, j) \mid j \in J\}$
- For every $j \in J$ let $A_j = \{r \mid (r, j) \in V\}$
- Since $|D_l| > 2^{|Q|}$ there exist $j_1 < j_2$ s.t. $A_{j_1} = A_{j_2}$.
  We write $A = A_{j_1} = A_{j_2}$, $\quad y_1 = x_{j_1+1} \dots x_{j_2-1}$, $\quad a_1 = x_{j_2}$ and $y = y_1 a_1$
- There are: $q_l \in Q_l, p_{l+1} \in Q_{l+1}$, $\quad$ s.t. $q_l \xrightarrow{a_1} p_{l+1}$ $\quad$ and $r \in A$ s.t. $r \xrightarrow{y_1} q_l$

And $\forall s \in A\ \exists r \in A$: $r \xrightarrow{y} s$ $\quad$ and $\forall r \in A\ \exists s \in A$: $r \xrightarrow{y} s$

# Not finitely ambiguous $\implies$ pattern (6)

$(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$,, $\quad |D_l| > 2^{|Q|}$, $\quad x = x_1 \ldots x_s$

- Let $J \subseteq \{1, \ldots, s\}$ so $D_l(x) = \{(p, j-1) \to (q, j) \mid j \in J\}$
- For every $j \in J$ let $A_j = \{r \mid (r, j) \in V\}$
- Since $|D_l| > 2^{|Q|}$ there exist $j_1 < j_2$ s.t. $A_{j_1} = A_{j_2}$.

  We write $A = A_{j_1} = A_{j_2}$, $\quad y_1 = x_{j_1+1} \ldots x_{j_2-1}$, $\quad a_1 = x_{j_2}$ and $y = y_1 a_1$
- There are: $q_l \in Q_l, p_{l+1} \in Q_{l+1}$, $\quad$ s.t. $q_l \xrightarrow{a_1} p_{l+1}$ $\quad$ and $r \in A$ s.t. $r \xrightarrow{y_1} q_l$

  $$\text{And } \forall s \in A\ \exists r \in A\text{: } r \xrightarrow{y} s \quad \text{and } \forall r \in A\ \exists s \in A\text{: } r \xrightarrow{y} s$$
- There is a sequence $r_i$ s.t. $r_1 \xrightarrow{y_1} q_l$ and $r_i \xrightarrow{y} r_{i-1}$

  For some $i_1$, $i_2$ we get $r_{i_1} = r_{i_1+i_2} = p$

## Not finitely ambiguous $\implies$ pattern (6)

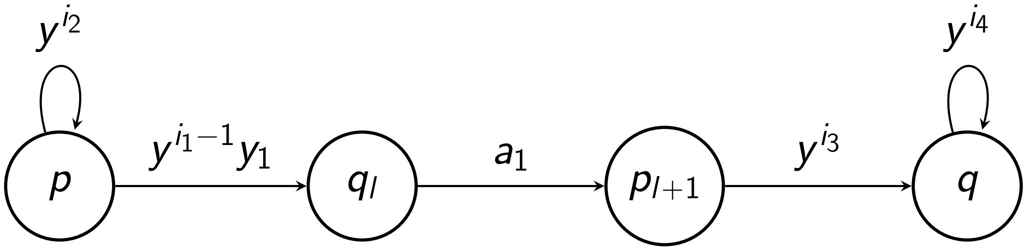
$(p, j-1) \to (q, j) \in D_l(x)$ if $p \in Q_l$ and $q \in Q_{l+1}$,, $\quad |D_l| > 2^{|Q|}$, $\quad x = x_1 \dots x_s$

- Let $J \subseteq \{1, \dots, s\}$ so $D_l(x) = \{(p, j-1) \to (q, j) \mid j \in J\}$
- For every $j \in J$ let $A_j = \{r \mid (r, j) \in V\}$
- Since $|D_l| > 2^{|Q|}$ there exist $j_1 < j_2$ s.t. $A_{j_1} = A_{j_2}$.
  We write $A = A_{j_1} = A_{j_2}$, $\quad y_1 = x_{j_1+1} \dots x_{j_2-1}$, $\quad a_1 = x_{j_2}$ and $y = y_1 a_1$
- There are: $q_l \in Q_l, p_{l+1} \in Q_{l+1}$, $\quad$ s.t. $q_l \xrightarrow{a_1} p_{l+1}$ $\quad$ and $r \in A$ s.t. $r \xrightarrow{y_1} q_l$

  And $\forall s \in A\ \exists r \in A$: $r \xrightarrow{y} s$ $\quad$ and $\forall r \in A\ \exists s \in A$: $r \xrightarrow{y} s$
- There is a sequence $r_i$ s.t. $r_1 \xrightarrow{y_1} q_l$ and $r_i \xrightarrow{y} r_{i-1}$
  For some $i_1$, $i_2$ we get $r_{i_1} = r_{i_1+i_2} = p$
- Similarly a sequence $s_i$: $\quad s_0 = p_{l+1}$ and $s_{i-1} \xrightarrow{y} s_i$, $\quad$ so $s_{i_3} = s_{i_3+i_4} = q$

# Not finitely ambiguous $\implies$ pattern (7)

$y = y_1 a_1$




- $p \xrightarrow{y^{i_2}} p, \quad p \xrightarrow{y^{i_1-1}y_1} q_l, \quad q_l \xrightarrow{a_1} p_{l+1}, \quad p_{l+1} \xrightarrow{y^{i_3}} q, \quad q \xrightarrow{y^{i_4}} q$

# Not finitely ambiguous $\implies$ pattern (7)

$y = y_1 a_1$

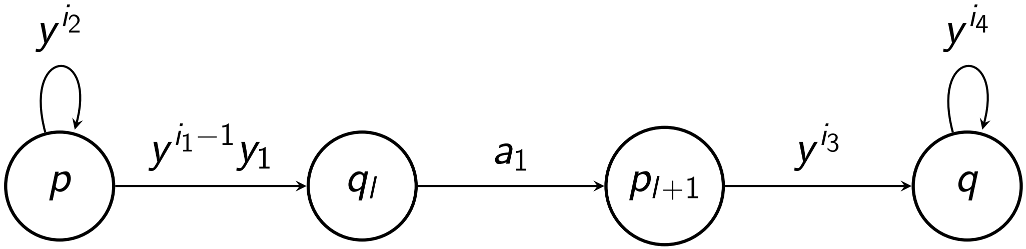
$$\begin{tikzcd}
p \arrow[loop above, "y^{i_2}"] \arrow[r, "y^{i_1-1}y_1"] & q_l \arrow[r, "a_1"] & p_{l+1} \arrow[r, "y^{i_3}"] & q \arrow[loop above, "y^{i_4}"]
\end{tikzcd}$$

- $p \xrightarrow{y^{i_2}} p, \quad p \xrightarrow{y^{i_1-1}y_1} q_l, \quad q_l \xrightarrow{a_1} p_{l+1}, \quad p_{l+1} \xrightarrow{y^{i_3}} q, \quad q \xrightarrow{y^{i_4}} q$

- Given $i_1$, $i_2$ we can choose $i_3$ and $i_4$ s.t. $i_1 + i_3 \equiv 0 \mod i_2 \cdot i_4$
  let $j$ s.t. $i_1 + i_3 = j(i_2 \cdot i_4)$

# Not finitely ambiguous $\implies$ pattern (7)

$y = y_1 a_1$




- $p \xrightarrow{y^{i_2}} p, \quad p \xrightarrow{y^{i_1-1}y_1} q_l, \quad q_l \xrightarrow{a_1} p_{l+1}, \quad p_{l+1} \xrightarrow{y^{i_3}} q, \quad q \xrightarrow{y^{i_4}} q$

- Given $i_1$, $i_2$ we can choose $i_3$ and $i_4$ s.t. $i_1 + i_3 \equiv 0 \mod i_2 \cdot i_4$

  let $j$ s.t. $i_1 + i_3 = j(i_2 \cdot i_4)$

- $p \neq q$, let $w = y^{j \cdot i_2 \cdot i_4}$

  $p \xrightarrow{w} p,\ p \xrightarrow{w} q,\ q \xrightarrow{w} q$

# Polynomially ambiguous

How to check if $\mathcal{A}$ is polynomially ambiguous?

# Polynomially ambiguous

How to check if $\mathcal{A}$ is polynomially ambiguous?

**Theorem** (Weber, Seidl 1991)

(1) $\mathcal{A}$ is not polynomially ambiguous if and only if

(2) there is a state $p \in Q$ and a word $w$ s.t. there are two runs $p \xrightarrow{w} p$

## Polynomially ambiguous

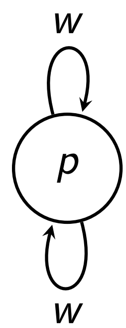
How to check if $\mathcal{A}$ is polynomially ambiguous?

**Theorem** (Weber, Seidl 1991)

(1) $\mathcal{A}$ is not polynomially ambiguous if and only if

(2) there is a state $p \in Q$ and a word $w$ s.t. there are two runs $p \xrightarrow{w} p$

- We're looking for this pattern



# Polynomially ambiguous

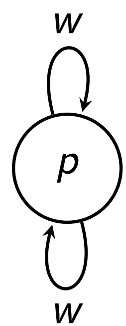
How to check if $\mathcal{A}$ is polynomially ambiguous?

**Theorem** (Weber, Seidl 1991)

(1) $\mathcal{A}$ is not polynomially ambiguous if and only if

(2) there is a state $p \in Q$ and a word $w$ s.t. there are two runs $p \xrightarrow{w} p$

- We're looking for this pattern




**Proof.** **(2) $\implies$ (1)**

There is $a \in I$ and $v_1$ s.t. $a \xrightarrow{v_a} p$ and $b \in F$ and $v_2$ s.t. $p \xrightarrow{v_b} b$

# Polynomially ambiguous

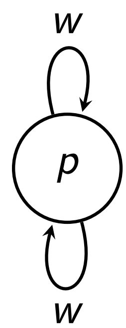
How to check if $\mathcal{A}$ is polynomially ambiguous?

**Theorem** (Weber, Seidl 1991)

(1) $\mathcal{A}$ is not polynomially ambiguous if and only if

(2) there is a state $p \in Q$ and a word $w$ s.t. there are two runs $p \xrightarrow{w} p$

- We're looking for this pattern




**Proof.** **(2) $\implies$ (1)**

There is $a \in I$ and $v_1$ s.t. $a \xrightarrow{v_a} p$ and $b \in F$ and $v_2$ s.t. $p \xrightarrow{v_b} b$

Then $|Acc(v_a w^n v_b)| \geqslant 2^n$.

# No pattern $\implies$ polynomially ambiguous

For every state $p \in Q$ and a word $w$ there is at most one run $p \xrightarrow{w} p$

## No pattern $\implies$ polynomially ambiguous

For every state $p \in Q$ and a word $w$ there is at most one run $p \xrightarrow{w} p$

- Then in every strongly connected component there is at most one $p \xrightarrow{w} q$
  (this was "Case 2" in the previous proof)

# No pattern $\implies$ polynomially ambiguous

For every state $p \in Q$ and a word $w$ there is at most one run $p \xrightarrow{w} p$

- Then in every strongly connected component there is at most one $p \xrightarrow{w} q$

  (this was "Case 2" in the previous proof)

- For every $x = x_1 \dots x_s$ the number of runs was bounded by

$$2^{|Q|} \cdot |Q_1| \cdot |D_1| \cdot |D_2| \cdot \ldots \cdot |D_{k-1}| \cdot |Q_k|$$

  (see slide 10)

## No pattern $\implies$ polynomially ambiguous

For every state $p \in Q$ and a word $w$ there is at most one run $p \xrightarrow{w} p$

- Then in every strongly connected component there is at most one $p \xrightarrow{w} q$

  (this was "Case 2" in the previous proof)

- For every $x = x_1 \ldots x_s$ the number of runs was bounded by

$$2^{|Q|} \cdot |Q_1| \cdot |D_1| \cdot |D_2| \cdot \ldots \cdot |D_{k-1}| \cdot |Q_k|$$

  (see slide 10)

- It remains to observe that $|D_i| \leqslant |Q|^2 \cdot s$

# No pattern $\implies$ polynomially ambiguous

For every state $p \in Q$ and a word $w$ there is at most one run $p \xrightarrow{w} p$

- Then in every strongly connected component there is at most one $p \xrightarrow{w} q$

  (this was "Case 2" in the previous proof)

- For every $x = x_1 \dots x_s$ the number of runs was bounded by

$$2^{|Q|} \cdot |Q_1| \cdot |D_1| \cdot |D_2| \cdot \ldots \cdot |D_{k-1}| \cdot |Q_k|$$

  (see slide 10)

- It remains to observe that $|D_i| \leqslant |Q|^2 \cdot s$

- Then the number of runs is bounded by a polynomial of degree $k-1$

$\blacksquare$

# Finitely ambiguous class more details

**Lemma** (tutorials)

If $\mathcal{A}$ is finitely ambiguous then it is $k$-ambiguous for some $k$ bounded exponentially in $|Q|$.

# Finitely ambiguous class more details

**Lemma** (tutorials)

If $\mathcal{A}$ is finitely ambiguous then it is $k$-ambiguous for some $k$ bounded exponentially in $|Q|$.

**Theorem** (Weber 1994)

If $\mathcal{A}$ is finitely ambiguous then it is equivalent to a finite union of unambiguous automata

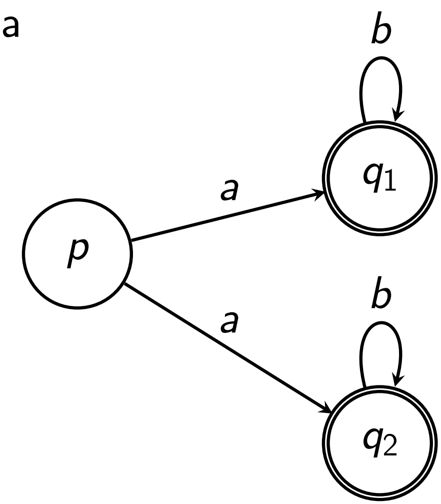
# Finitely ambiguous class more details

**Lemma** (tutorials)

If $\mathcal{A}$ is finitely ambiguous then it is $k$-ambiguous for some $k$ bounded exponentially in $|Q|$.

**Theorem** (Weber 1994)

If $\mathcal{A}$ is finitely ambiguous then it is equivalent to a finite union of unambiguous automata



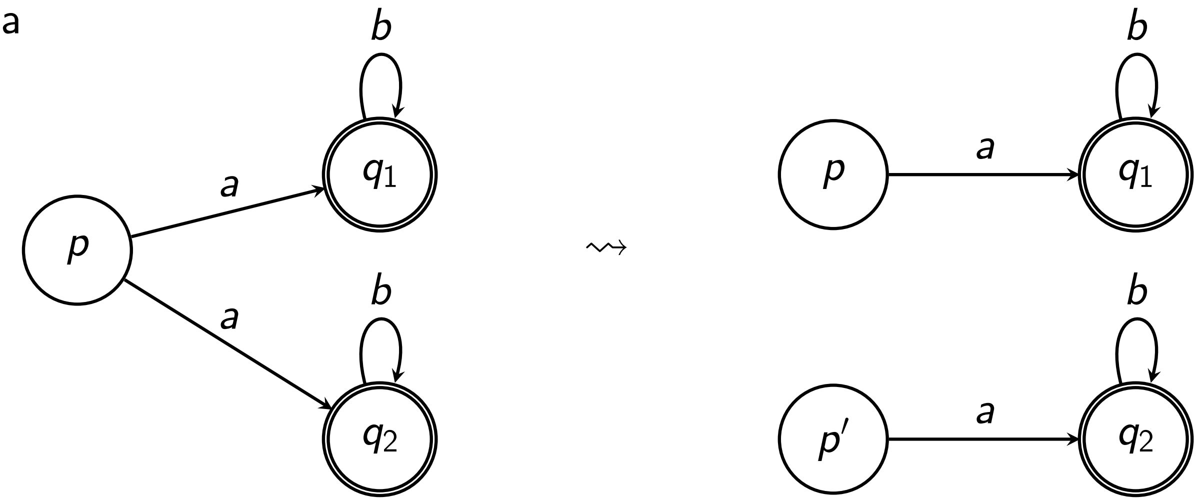
## Finitely ambiguous class more details

**Lemma** (tutorials)

If $\mathcal{A}$ is finitely ambiguous then it is $k$-ambiguous for some $k$ bounded exponentially in $|Q|$.

**Theorem** (Weber 1994)

If $\mathcal{A}$ is finitely ambiguous then it is equivalent to a finite union of unambiguous automata



## Construction of the union of unambiguous automata

**Proof (sketch).**

An *active run* is a run that not necessarily ends in an accepting state

## Construction of the union of unambiguous automata

**Proof (sketch).**

An *active run* is a run that not necessarily ends in an accepting state

- By the previous Lemma there is a bound on active runs for every word
  $kn$: where $k$ is the bound on the ambiguity and $|Q| = n$.

## Construction of the union of unambiguous automata

**Proof (sketch).**

An *active run* is a run that not necessarily ends in an accepting state

- By the previous Lemma there is a bound on active runs for every word $kn$: where $k$ is the bound on the ambiguity and $|Q| = n$.
- Assume that $Q = \{1, 2, \ldots, n\}$ (we will use the order)

## Construction of the union of unambiguous automata

**Proof (sketch).**

An *active run* is a run that not necessarily ends in an accepting state

- By the previous Lemma there is a bound on active runs for every word $kn$: where $k$ is the bound on the ambiguity and $|Q| = n$.
- Assume that $Q = \{1, 2, \ldots, n\}$ (we will use the order)
- Consider the deterministic automaton $\mathcal{B}$ with states that are the set of all partial functions:

$$f : \{1, 2, \ldots, kn\} \to Q$$

# Construction of the union of unambiguous automata

**Proof (sketch).**

An *active run* is a run that not necessarily ends in an accepting state

- By the previous Lemma there is a bound on active runs for every word

  $kn$: where $k$ is the bound on the ambiguity and $|Q| = n$.

- Assume that $Q = \{1, 2, \ldots, n\}$ (we will use the order)

- Consider the deterministic automaton $\mathcal{B}$

  with states that are the set of all partial functions:

$$f : \{1, 2, \ldots, kn\} \to Q$$

- $\mathcal{B}$ keeps track of all active runs in $\mathcal{A}$

## Construction of the union of unambiguous automata (2)

- $\mathcal{B}$ states are partial functions $f : \{1, 2, \ldots, kn\} \to Q$

  The size of the domain is always the number of runs

# Construction of the union of unambiguous automata (2)

- $\mathcal{B}$ states are partial functions $f : \{1, 2, \ldots, kn\} \to Q$

  The size of the domain is always the number of runs

- Initially the domain is $\{1, \ldots, |I|\}$ and the images are $I \subseteq Q$

  We use the order on $Q$ for $\mathcal{B}$ to be deterministic

## Construction of the union of unambiguous automata (2)

- $\mathcal{B}$ states are partial functions $f : \{1, 2, \ldots, kn\} \to Q$

  The size of the domain is always the number of runs

- Initially the domain is $\{1, \ldots, |I|\}$ and the images are $I \subseteq Q$

  We use the order on $Q$ for $\mathcal{B}$ to be deterministic

- Then for every transition in $\mathcal{A}$ we update the states in $\mathcal{B}$

  Given a partial function $f : \{1, 2, \ldots, kn\} \to Q$

  we get $g : \{1, 2, \ldots, kn\} \to Q$

## Construction of the union of unambiguous automata (2)

- $\mathcal{B}$ states are partial functions $f : \{1, 2, \ldots, kn\} \to Q$

  The size of the domain is always the number of runs

- Initially the domain is $\{1, \ldots, |I|\}$ and the images are $I \subseteq Q$

  We use the order on $Q$ for $\mathcal{B}$ to be deterministic

- Then for every transition in $\mathcal{A}$ we update the states in $\mathcal{B}$

  Given a partial function $f : \{1, 2, \ldots, kn\} \to Q$

  we get $g : \{1, 2, \ldots, kn\} \to Q$

- Then $\mathcal{A} \times \mathcal{B}$ has the same accepting runs as $\mathcal{A}$ but with extra information

  Accepting states etc are when they are accepting in the $\mathcal{A}$ component

# Construction of the union of unambiguous automata (3)

- $\mathcal{A} \times \mathcal{B}$ is equivalent to $\mathcal{A}$ but with more information

## Construction of the union of unambiguous automata (3)

- $\mathcal{A} \times \mathcal{B}$ is equivalent to $\mathcal{A}$ but with more information

- In a similar way we add another component of states

  Partial functions $f : Q \to \{1, \ldots, kn\}$

## Construction of the union of unambiguous automata (3)

- $\mathcal{A} \times \mathcal{B}$ is equivalent to $\mathcal{A}$ but with more information

- In a similar way we add another component of states
  Partial functions $f : Q \to \{1, \ldots, kn\}$

- This adds the information, which $\{1, \ldots, kn\}$ has the current run in $\mathcal{A}$
  It can be extracted when updating $\mathcal{B}$

## Construction of the union of unambiguous automata (3)

- $\mathcal{A} \times \mathcal{B}$ is equivalent to $\mathcal{A}$ but with more information

- In a similar way we add another component of states
  Partial functions $f : Q \to \{1, \ldots, kn\}$

- This adds the information, which $\{1, \ldots, kn\}$ has the current run in $\mathcal{A}$
  It can be extracted when updating $\mathcal{B}$

- The final automata are divided into $kn$ unambiguous automata
  Restricting the accepting states to accepting in $\mathcal{A}$
  and $i \in \{1, \ldots, kn\}$ in the final component