A Congruence-based Perspective on Automata Minimization Algorithms

Pierre Ganty, Elena Gutiérrez, Pedro Valero

IMDEA Software Institute, Madrid, Spain

Séminaire MF - LaBRI

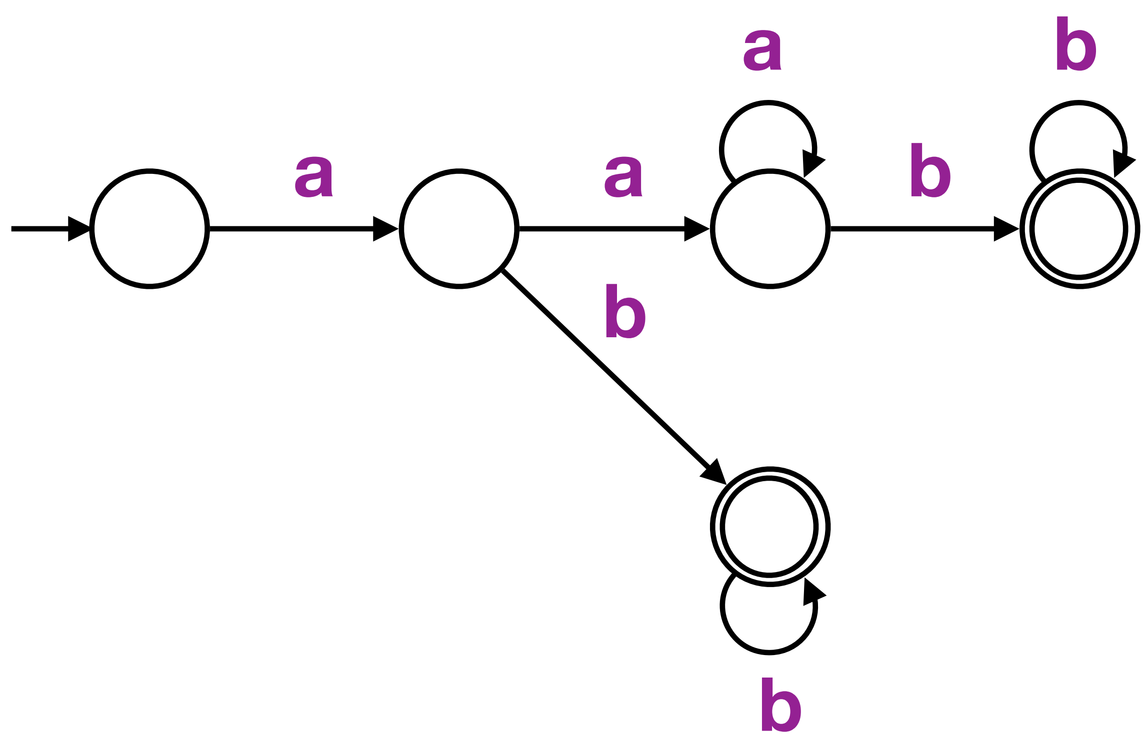
June, 23rd, 2020

institute imdea software

Motivation

Automata Minimization Algorithms

Finite-state automaton



Regular language

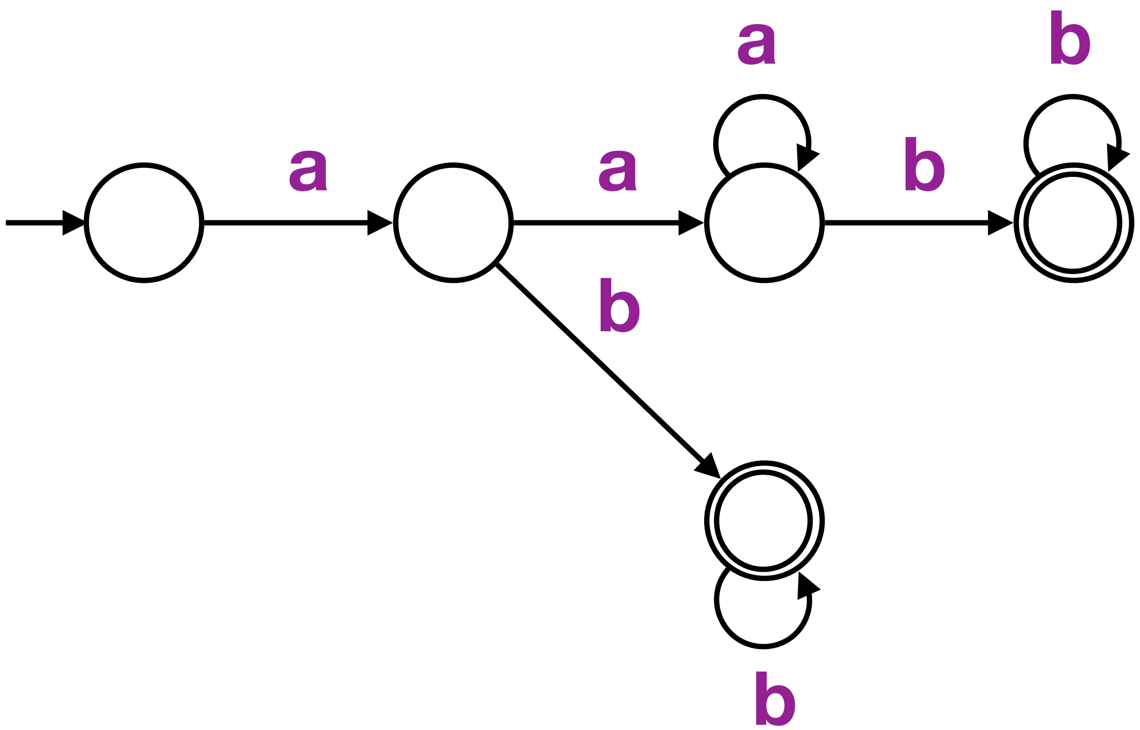
$a^{+} b^{+}$

all words with at least one 'a'
followed by at least one 'b'

Motivation

Automata Minimization Algorithms

Finite-state automaton



Regular language

$a^+ b^+$

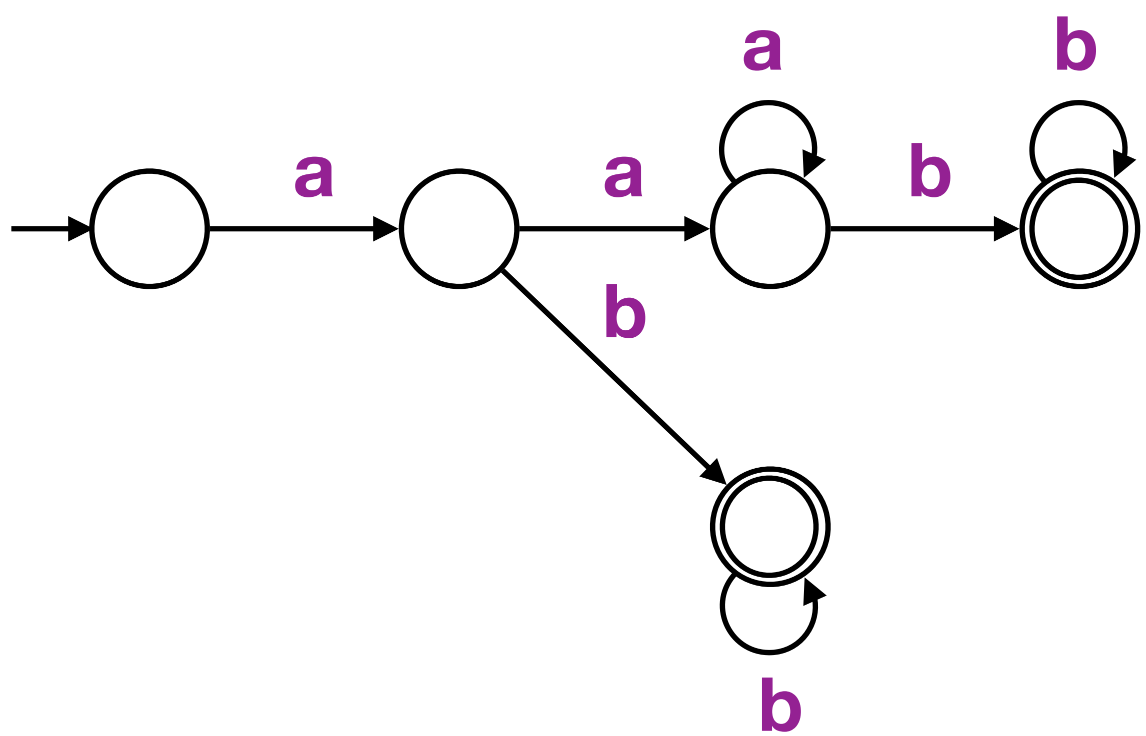
all words with at least one ‘a’
followed by at least one ‘b’

Find the finite-state automaton with the <u>least</u> number of states for the language

Motivation

Automata Minimization Algorithms

Finite-state automaton

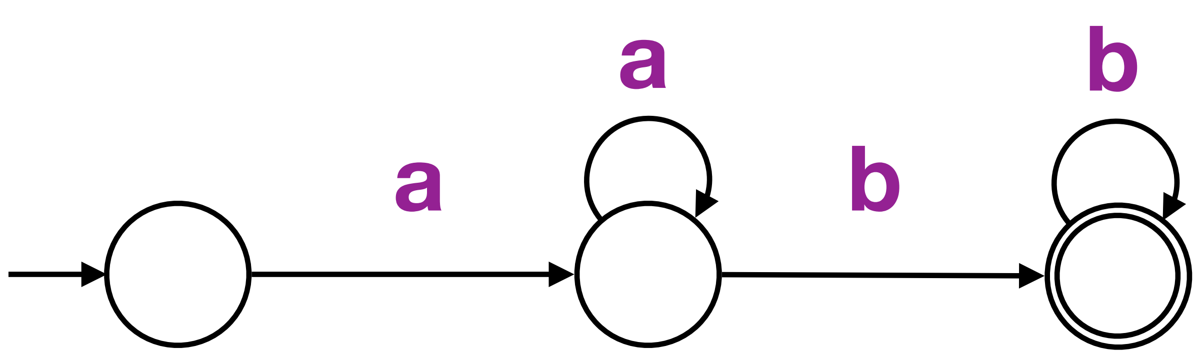


Regular language

$a^+ b^+$

all words with at least one 'a'
followed by at least one 'b'

Minimal (deterministic) finite-state automaton



Find the finite-state automaton with the <u>least</u> number of states for the language

Motivation

Automata Minimization Algorithms

Motivation

Automata Minimization Algorithms

Hopcroft's algorithm

Double-reversal method

Moore's algorithm

Revuz's algorithm

Motivation

Automata Minimization Algorithms

Hopcroft's algorithm

Moore's algorithm

Revuz's algorithm

Double-reversal method

Partition of the set of states

Motivation

Automata Minimization Algorithms

Hopcroft's algorithm

Moore's algorithm

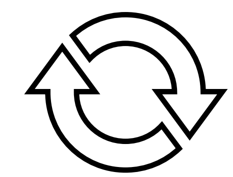
Revuz's algorithm

Partition of the set of states

Double-reversal method

1.Reverse

4.Determinization



2.Determinization

3.Reverse

Combination of automata constructions

Motivation

Automata Minimization Algorithms

Hopcroft's algorithm

Moore's algorithm

Revuz's algorithm

Partition of the set of states

Double-reversal method

1.Reverse

2.Determinization

3.Reverse

4.Determinization

Combination of automata constructions

Goal

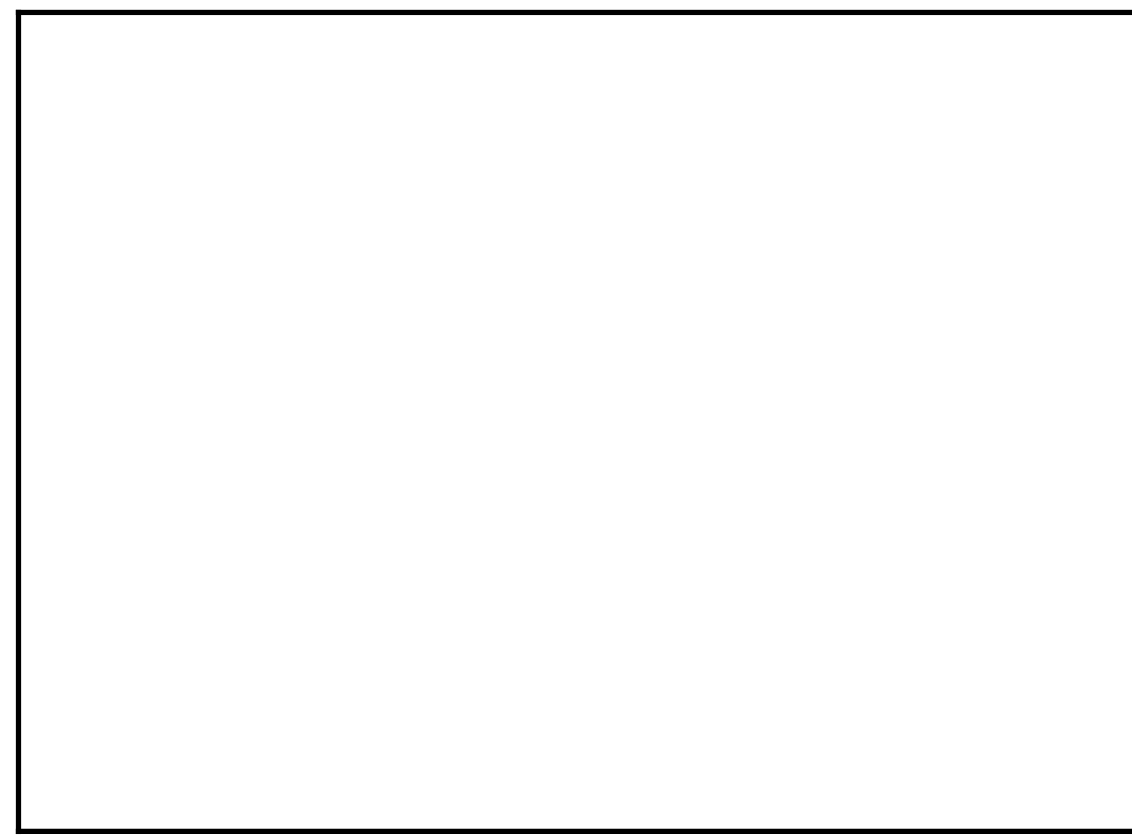
Give new language-theoretical insights on:

- the double-reversal method, and
- its connection with the partition-based methods

Language-theoretical Perspective

Common purpose: Build **Nerode's equivalence relation on words**

$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ

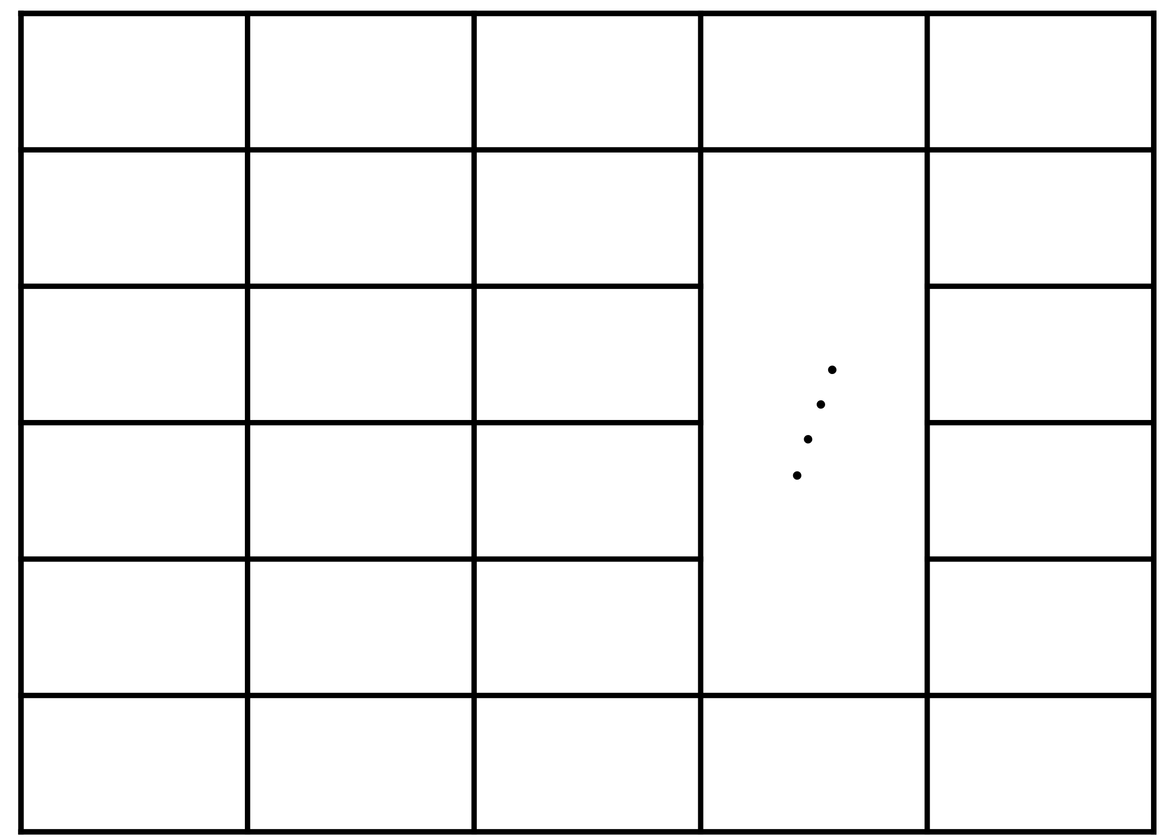


$\sim$

Language-theoretical Perspective

Common purpose: Build **Nerode's equivalence relation on words**

$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ

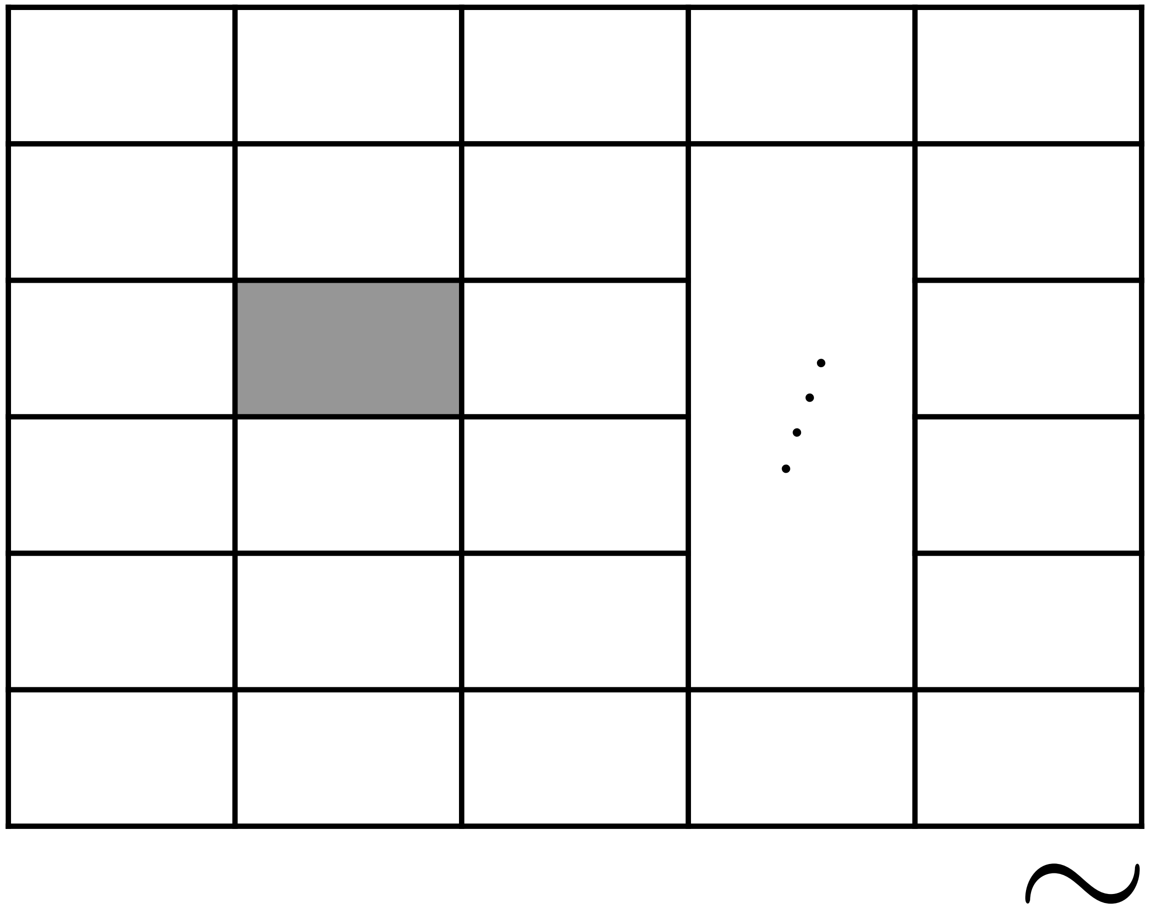


$\sim$

Language-theoretical Perspective

Common purpose: Build **Nerode's equivalence relation on words**

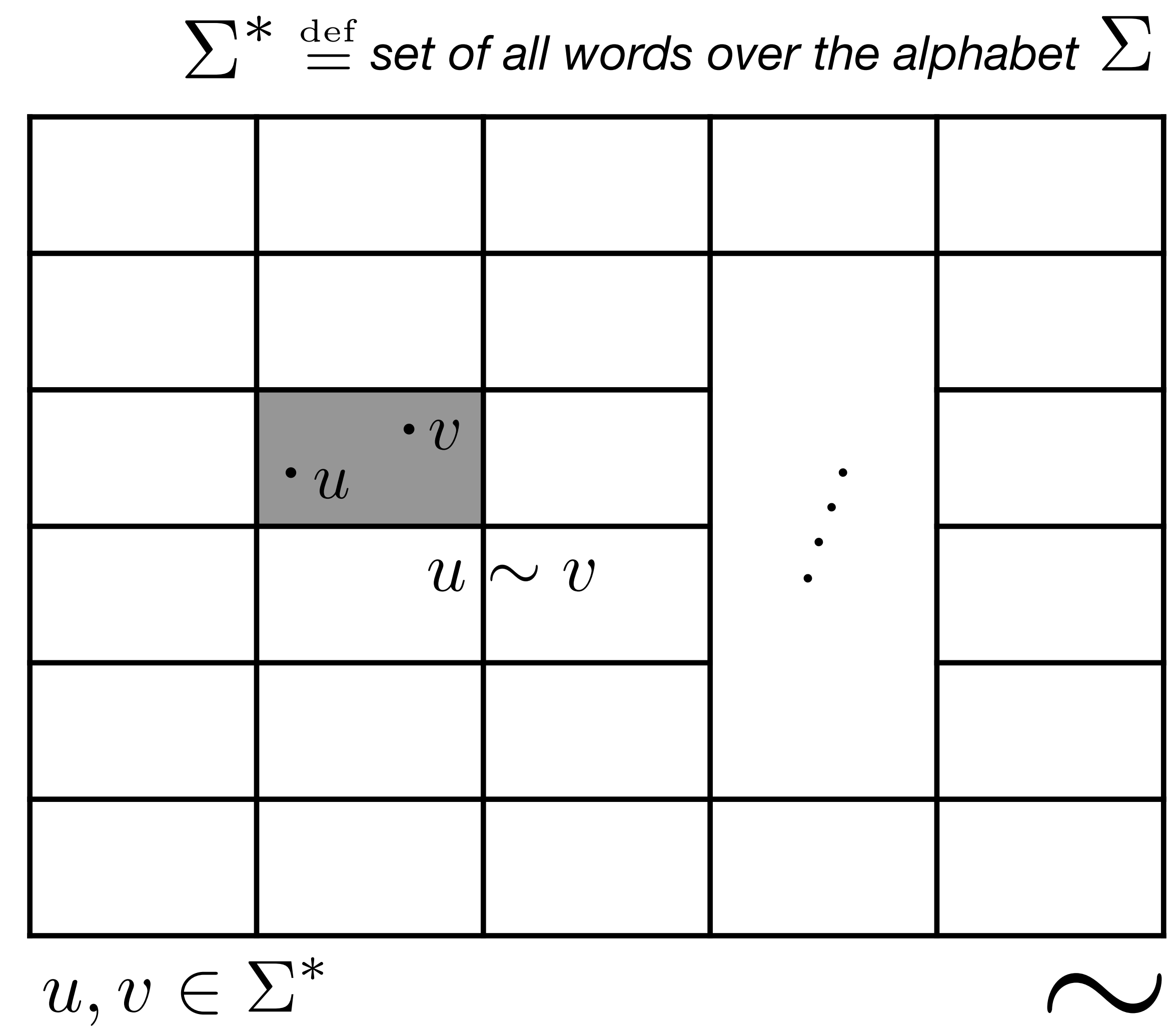
$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ



E. Gutiérrez, IMDEA Software, Madrid

Language-theoretical Perspective

Common purpose: Build **Nerode's equivalence relation on words**



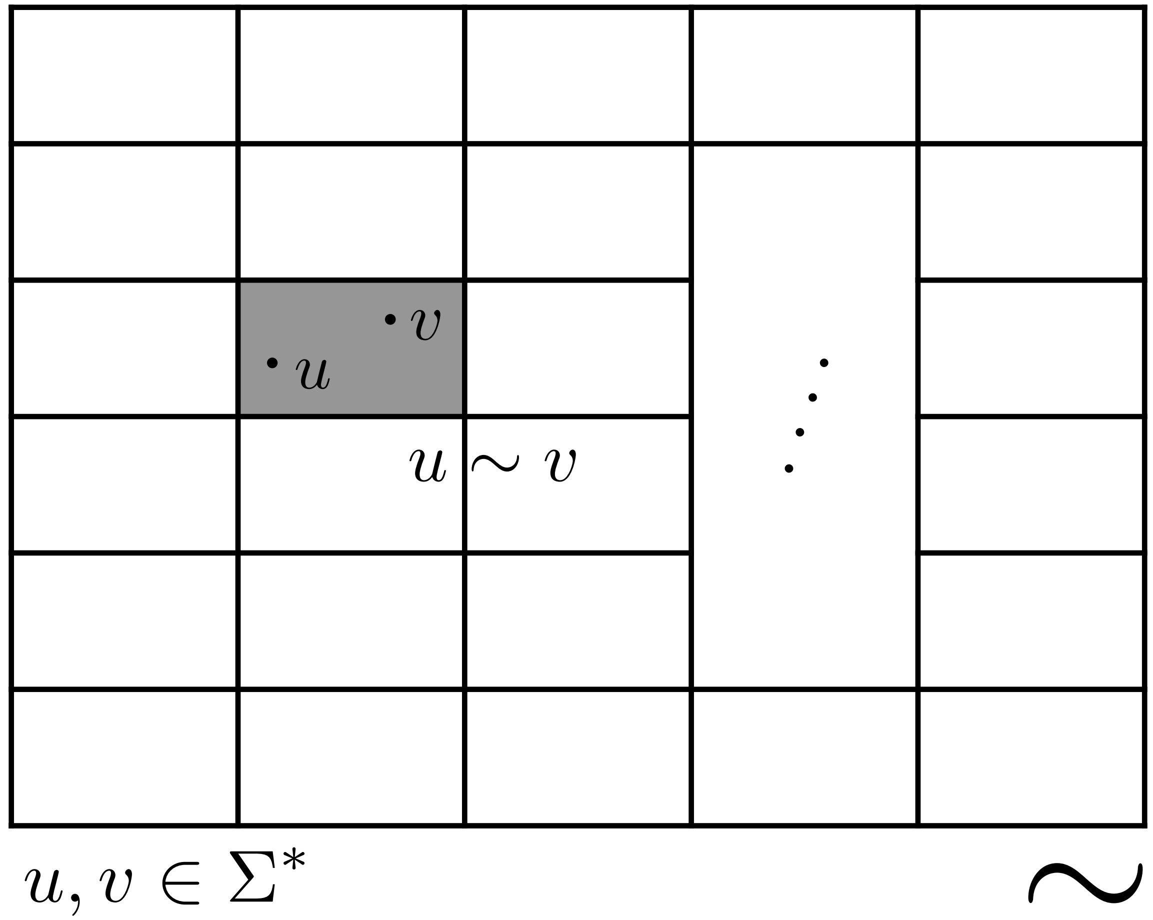
Language-theoretical Perspective

Common purpose: Build **Nerode's equivalence relation on words**

Given an automaton and its language L:

$$u \sim_L v \Leftrightarrow u^{-1}L = v^{-1}L$$

$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ



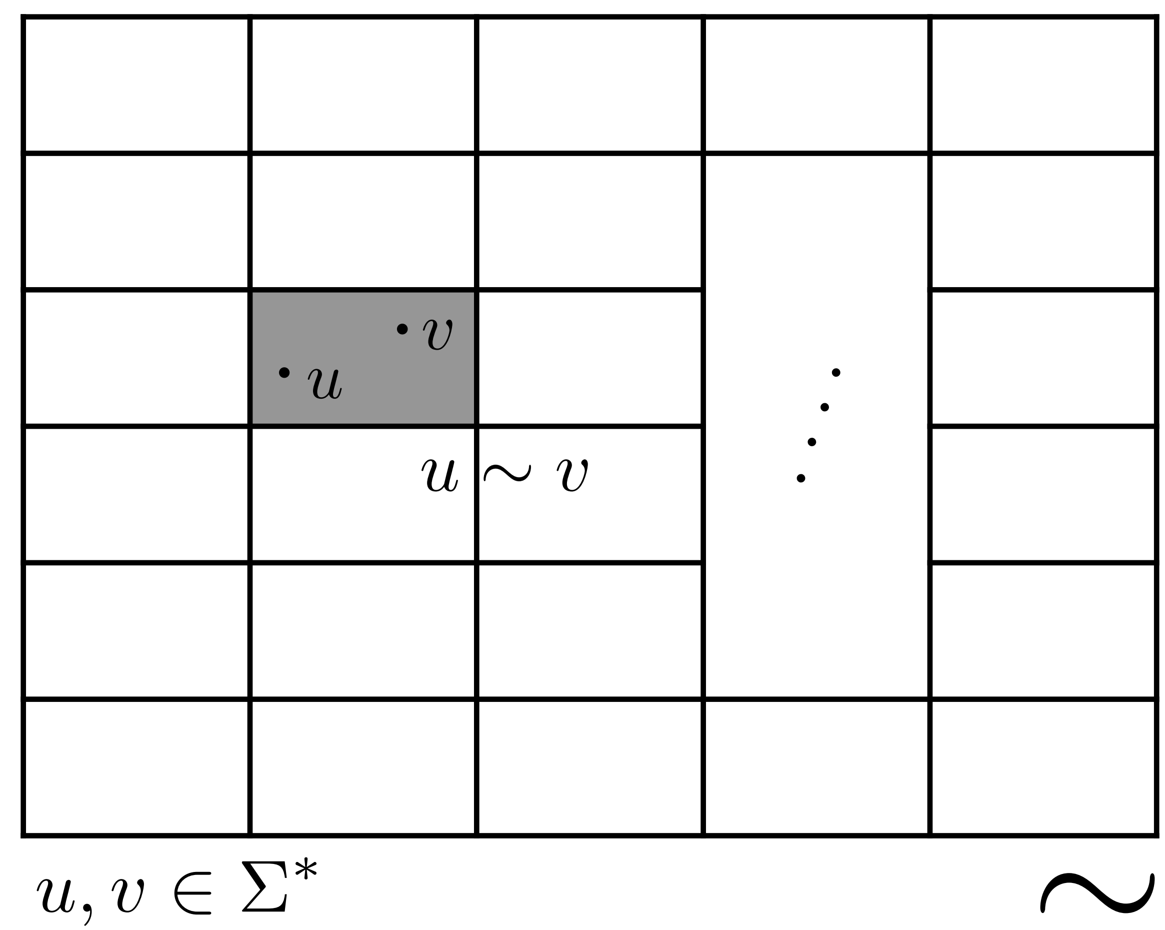
Language-theoretical Perspective

Common purpose: Build **Nerode's equivalence relation on words**

Given an automaton and its language L:

$$u \sim_L v \Leftrightarrow \underbrace{u^{-1}L}_{\{w \mid uw \in L\}} = v^{-1}L$$

$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ



E. Gutiérrez, IMDEA Software, Madrid

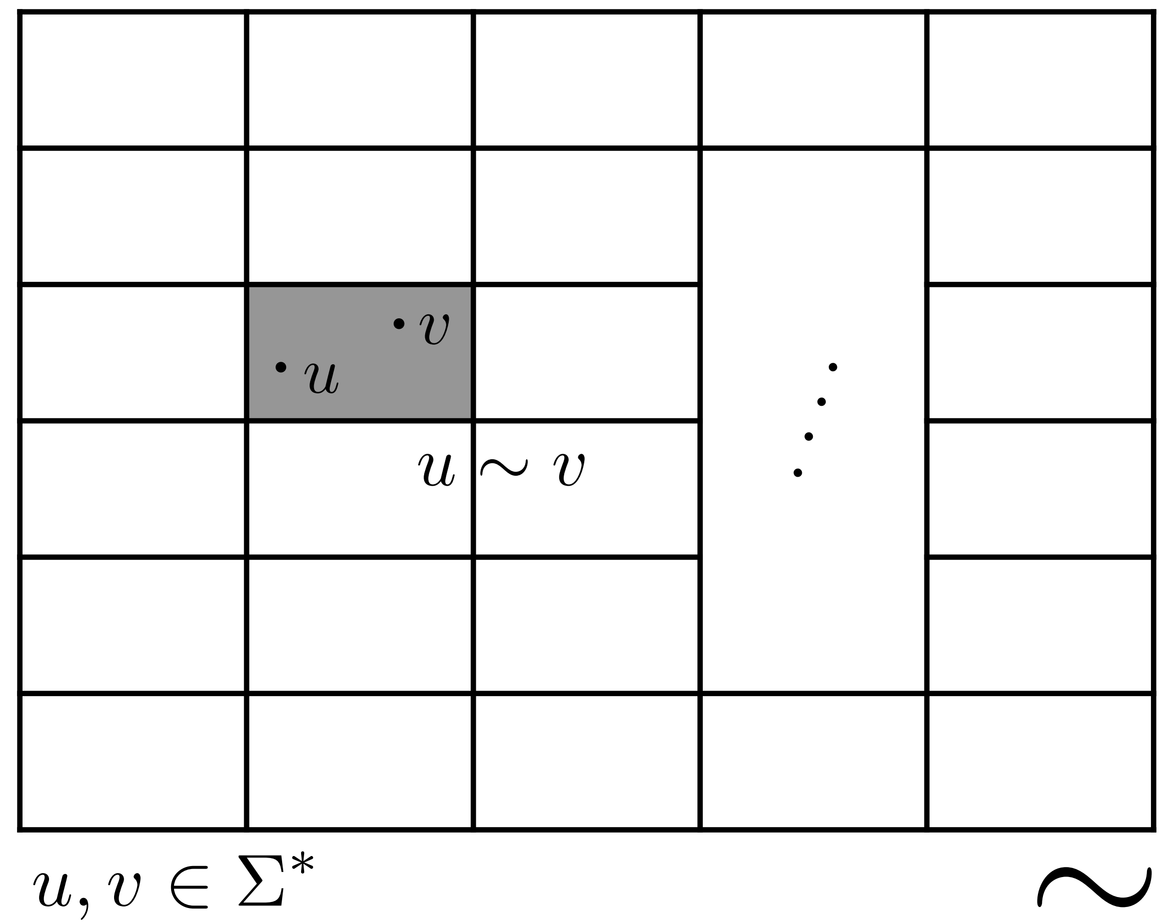
Language-theoretical Perspective

Common purpose: Build **Nerode's equivalence relation on words**

Given an automaton and its language L:

$$u \sim_L v \Leftrightarrow u^{-1}L = v^{-1}L$$

$\Sigma^* \overset{\text{def}}{=}$ *set of all words over the alphabet* Σ



Language-theoretical Perspective

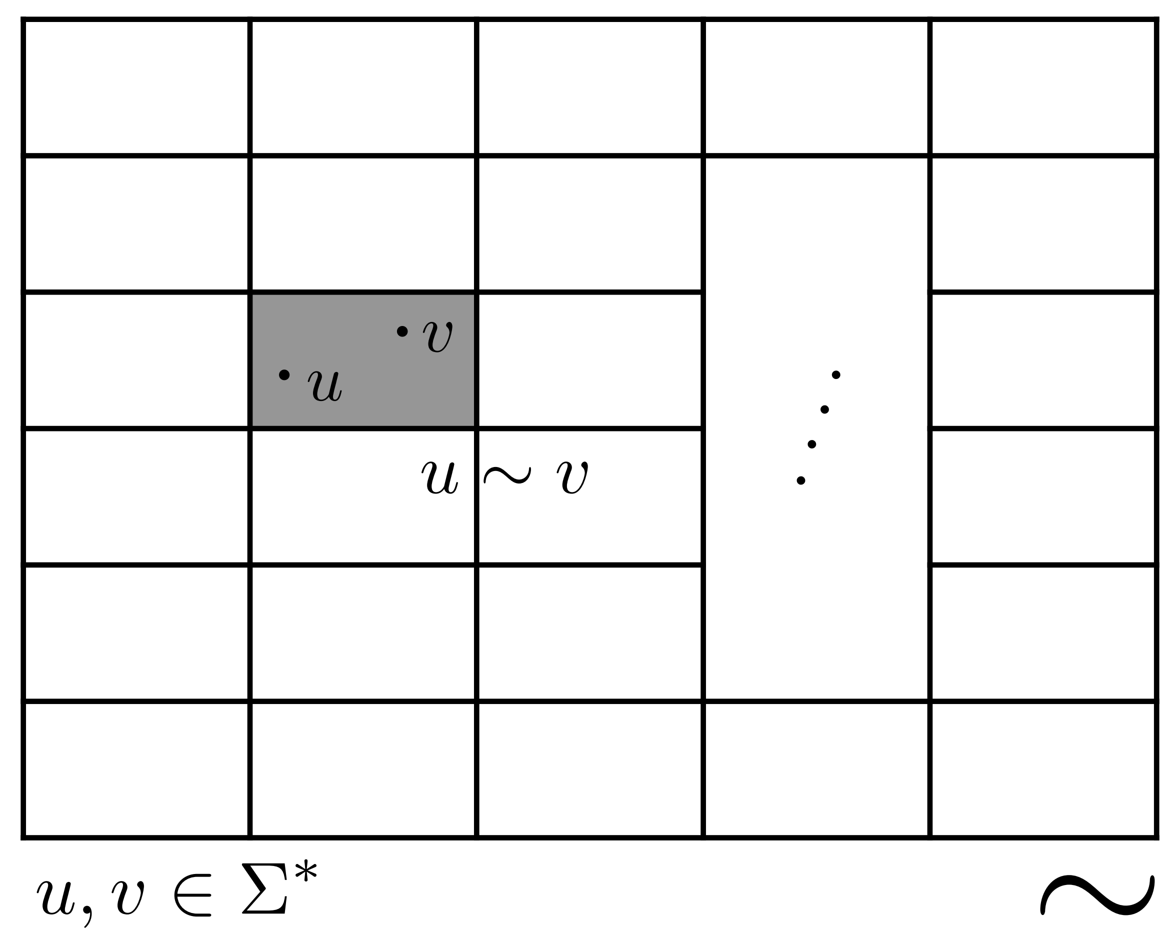
Common purpose: Build **Nerode's equivalence relation on words**

Given an automaton and its language L:

$$u \sim_L v \Leftrightarrow u^{-1}L = v^{-1}L$$

- **Finite** number of equivalence classes

$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ



Language-theoretical Perspective

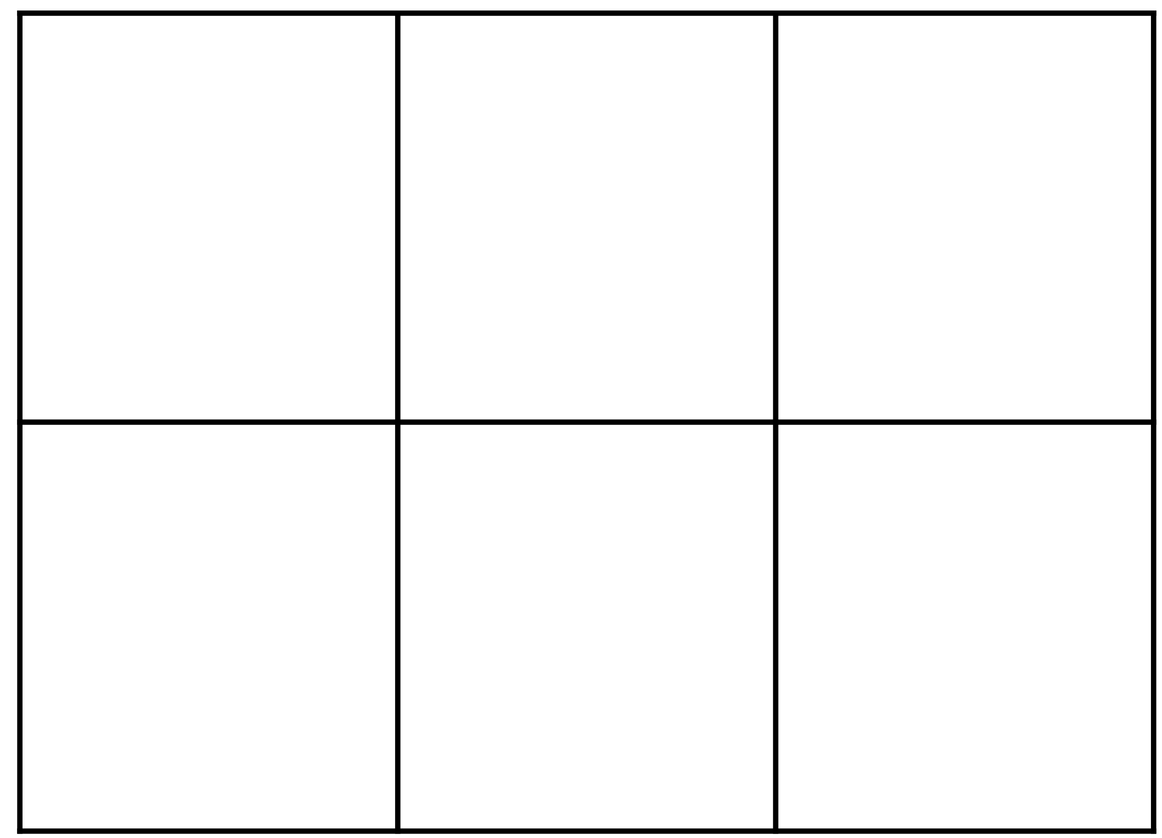
Common purpose: Build **Nerode's equivalence relation on words**

Given an automaton and its language L:

$$u \sim_L v \Leftrightarrow u^{-1}L = v^{-1}L$$

- **Finite** number of equivalence classes

$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ



$\sim$

Language-theoretical Perspective

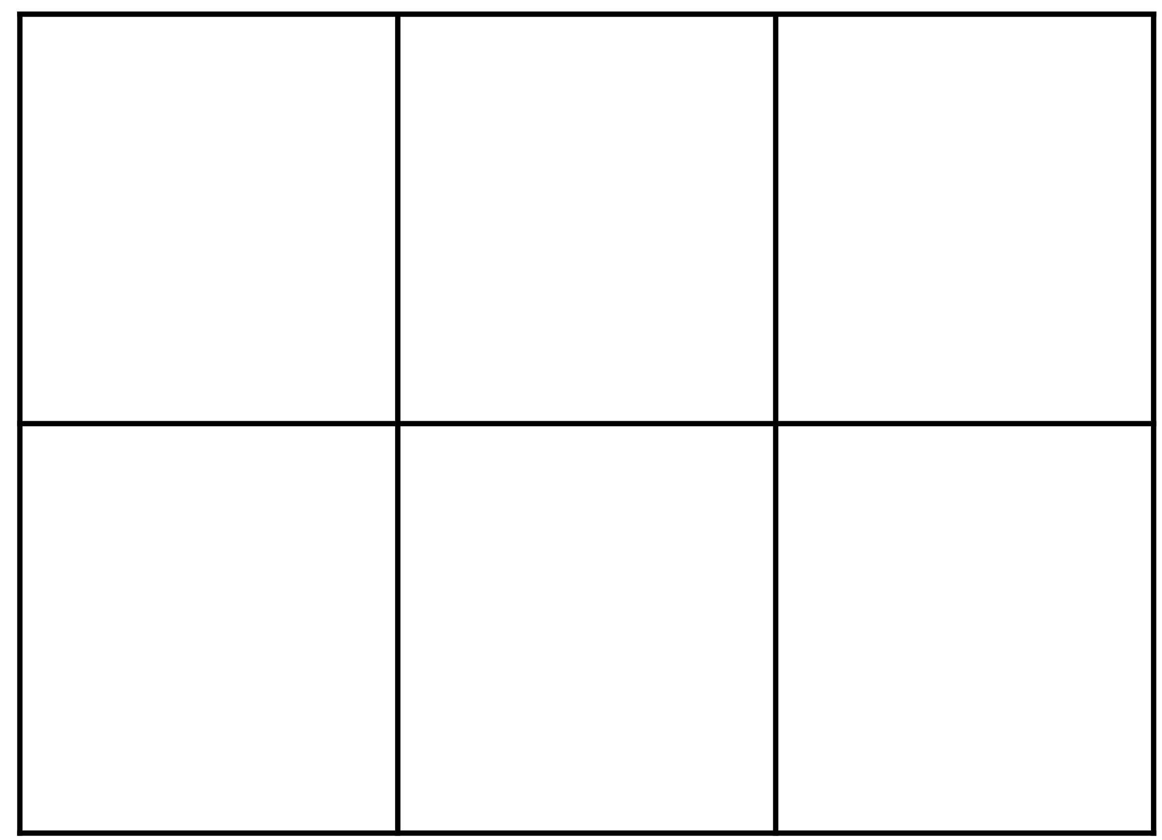
Common purpose: Build **Nerode's equivalence relation on words**

Given an automaton and its language L:

$$u \sim_L v \Leftrightarrow u^{-1}L = v^{-1}L$$

- **Finite** number of equivalence classes
- **Congruence**

$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ



$\sim$

E. Gutiérrez, IMDEA Software, Madrid

Language-theoretical Perspective

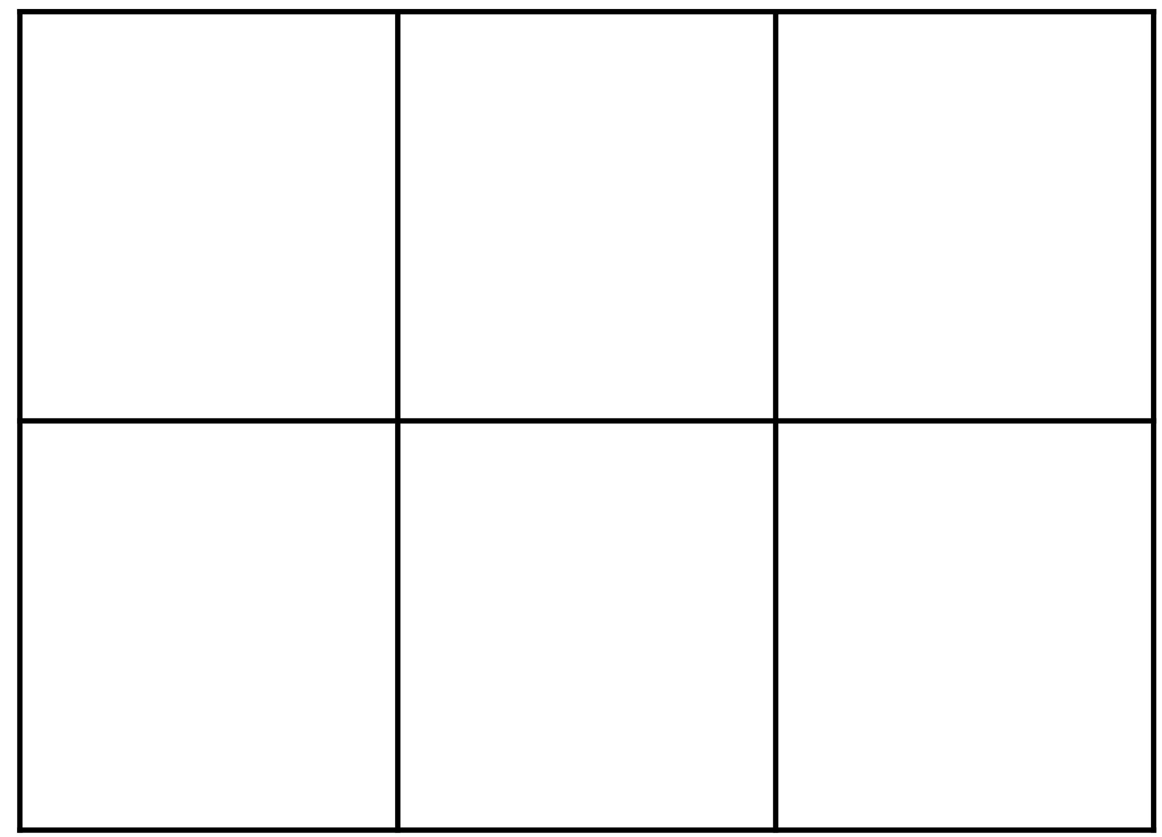
Common purpose: Build **Nerode's equivalence relation on words**

Given an automaton and its language L:

$$u \sim_L v \Leftrightarrow u^{-1}L = v^{-1}L$$

- **Finite** number of equivalence classes
- **Congruence**
- **Precisely represents** L

$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ



$\sim$

E. Gutiérrez, IMDEA Software, Madrid

Language-theoretical Perspective

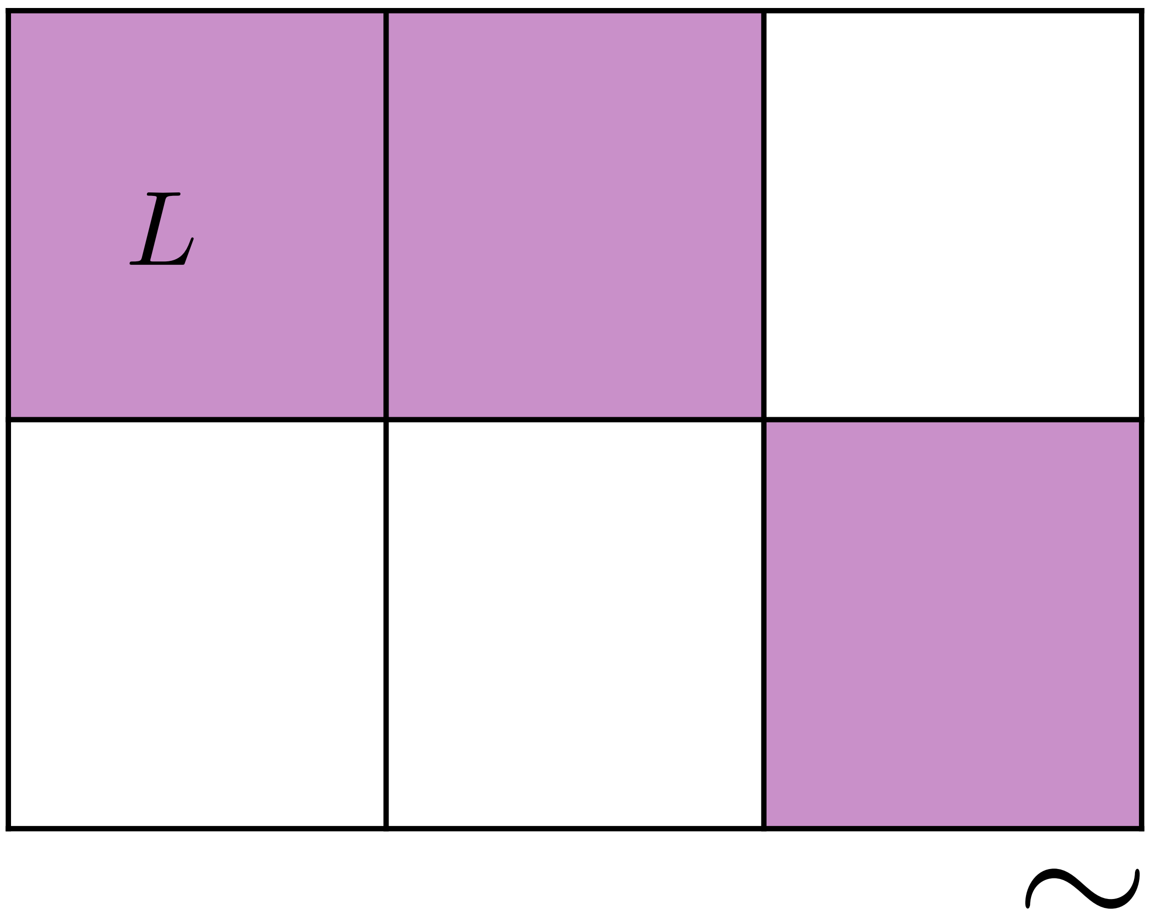
Common purpose: Build **Nerode's equivalence relation on words**

Given an automaton and its language L:

$$u \sim_L v \Leftrightarrow u^{-1}L = v^{-1}L$$

- **Finite** number of equivalence classes
- **Congruence**
- **Precisely represents** L

$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ



E. Gutiérrez, IMDEA Software, Madrid

Language-theoretical Perspective

Common purpose: Build **Nerode's equivalence relation on words**

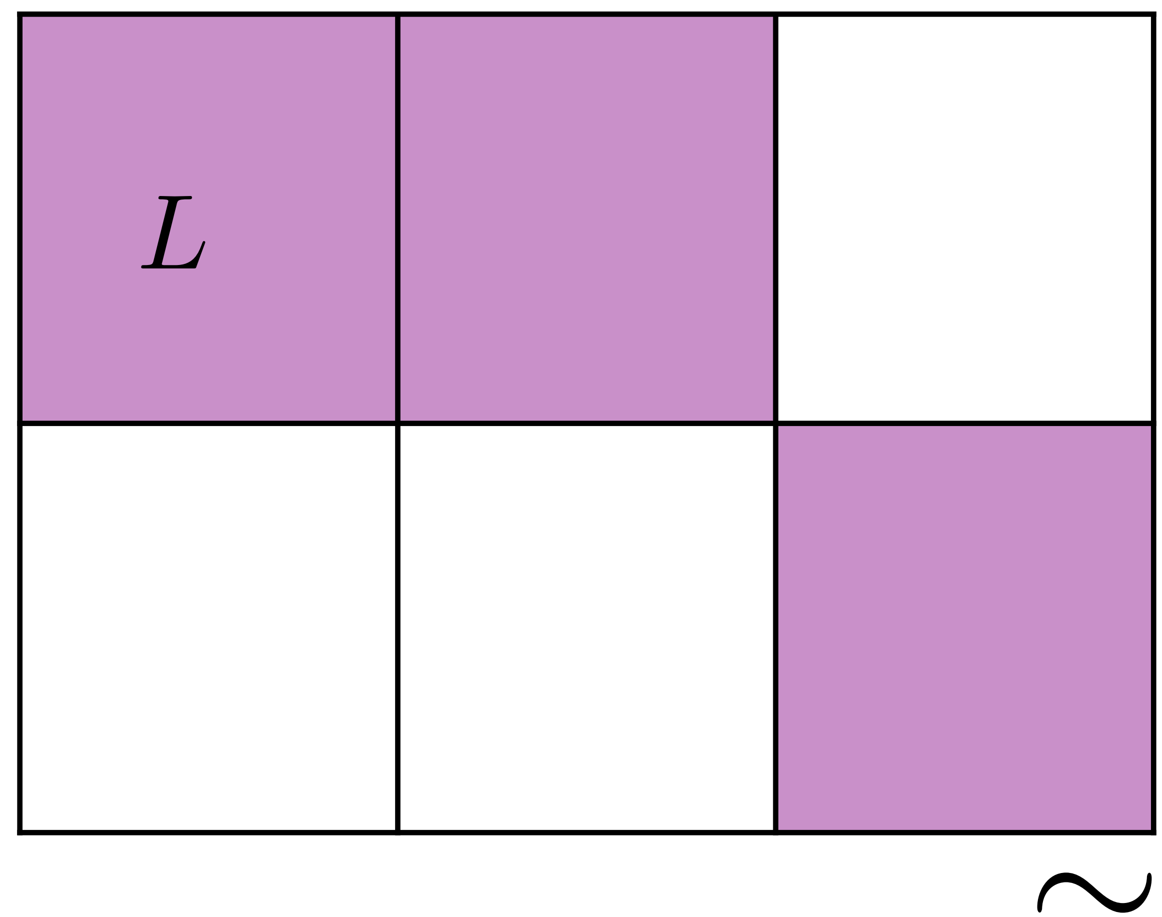
Given an automaton and its language L:

$$u \sim_L v \Leftrightarrow u^{-1}L = v^{-1}L$$

- **Finite** number of equivalence classes
- **Congruence**
- **Precisely represents** L

Build a deterministic automaton for L

$\Sigma^* \stackrel{\mathrm{def}}{=}$ *set of all words over the alphabet* Σ



E. Gutiérrez, IMDEA Software, Madrid

Language-theoretical Perspective

Common purpose: Build **Nerode's equivalence relation on words**

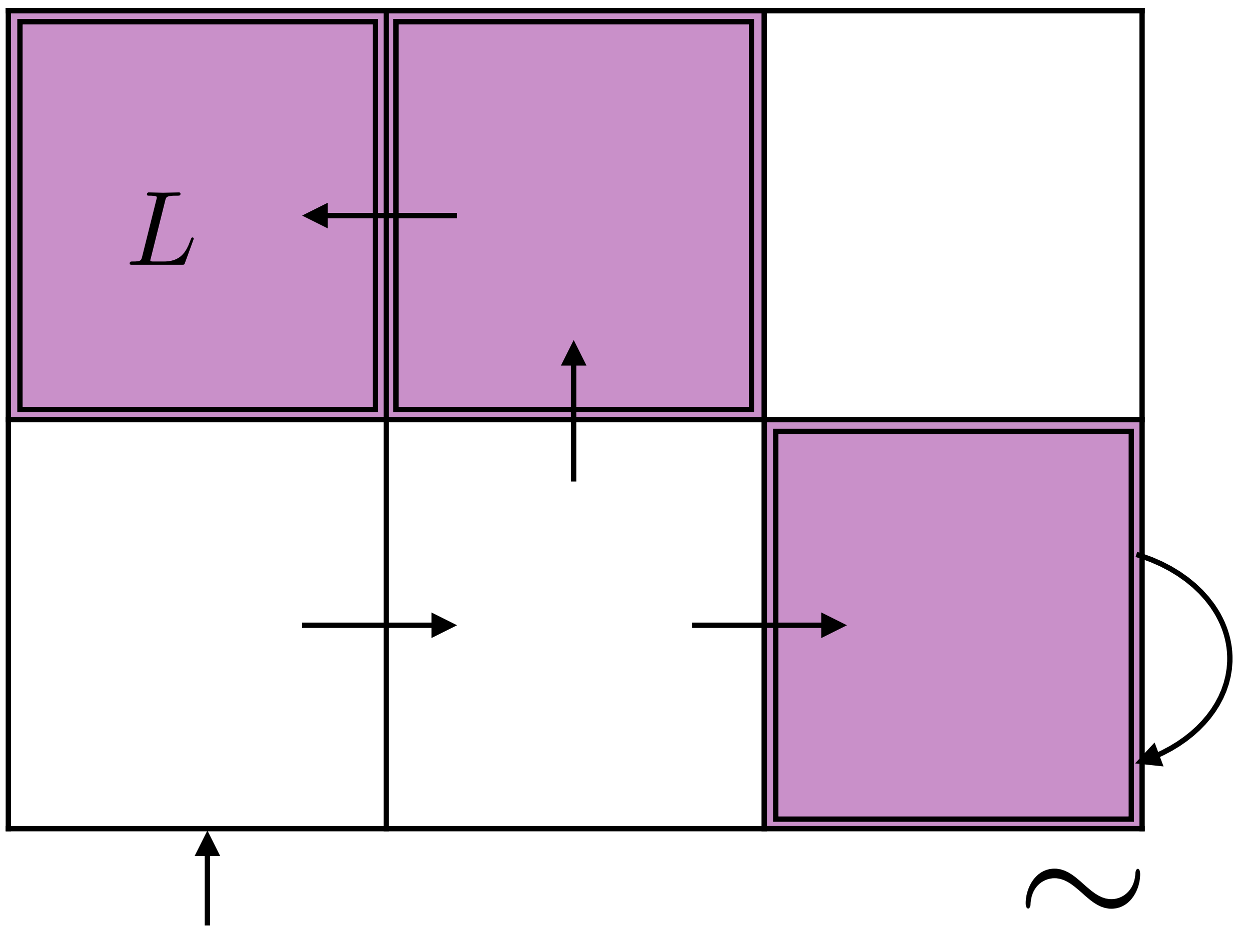
Given an automaton and its language L:

$$u \sim_L v \Leftrightarrow u^{-1}L = v^{-1}L$$

- **Finite** number of equivalence classes
- **Congruence**
- **Precisely represents** L

Build a deterministic automaton for L

$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ



E. Gutiérrez, IMDEA Software, Madrid

Language-theoretical Perspective

Common purpose: Build **Nerode's equivalence relation on words**

Given an automaton and its language L:

$$u \sim_L v \Leftrightarrow u^{-1}L = v^{-1}L$$

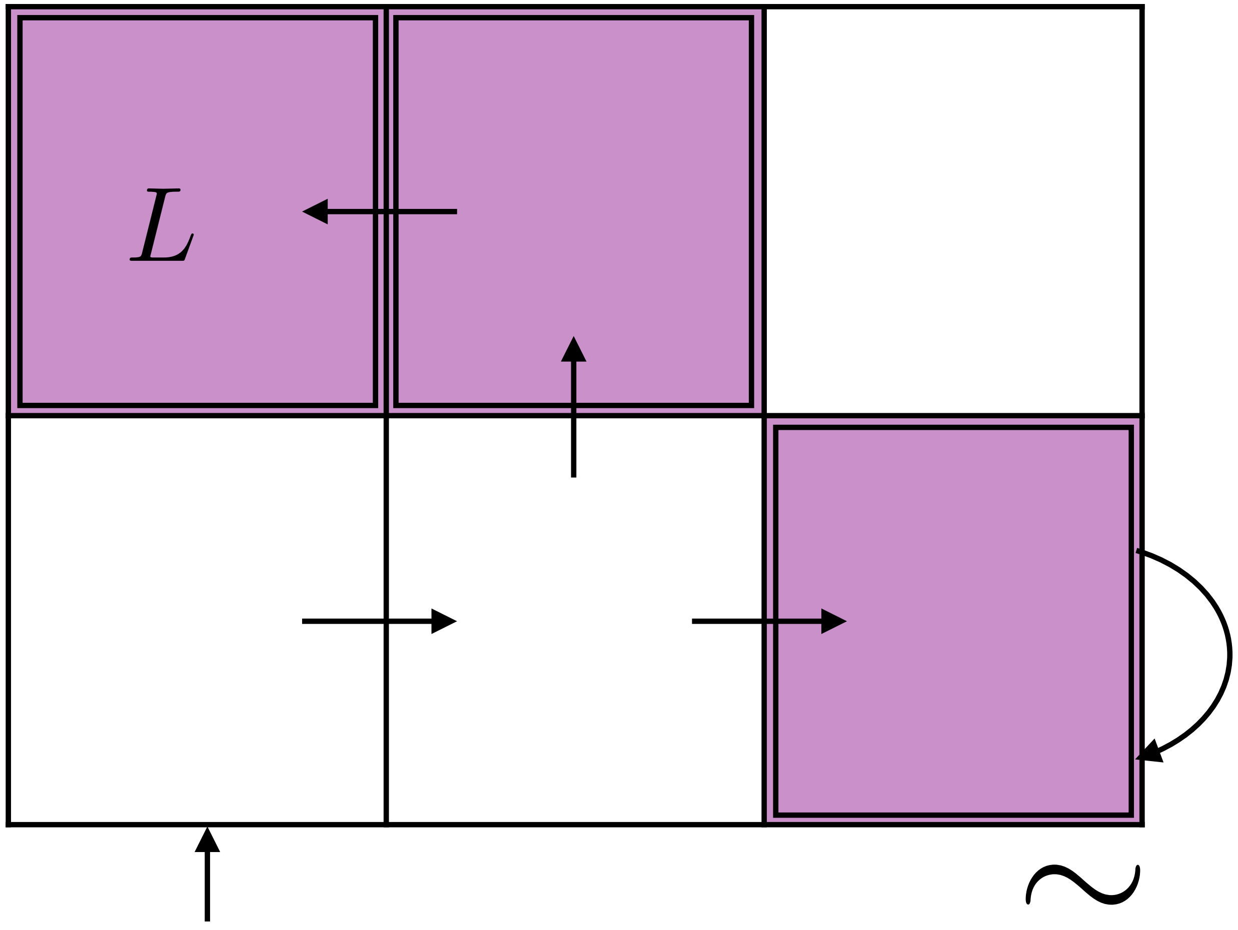
- **Finite** number of equivalence classes
- **Congruence**
- **Precisely represents** L

Build a deterministic automaton for L

- **Coarsest congruence** satisfying the latter properties

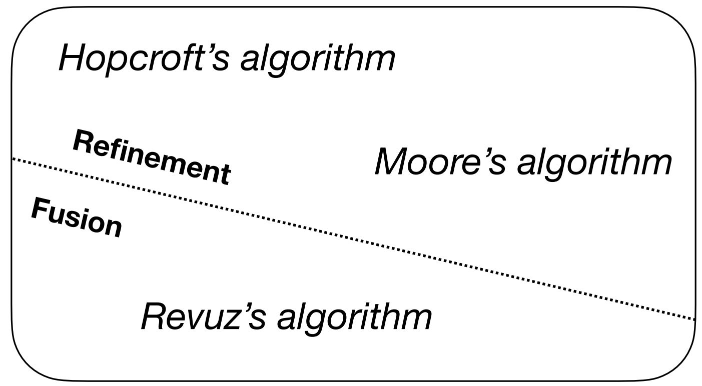
Build the ***minimal*** *deterministic automaton for* L

$\Sigma^* \stackrel{\text{def}}{=}$ *set of all words over the alphabet* Σ

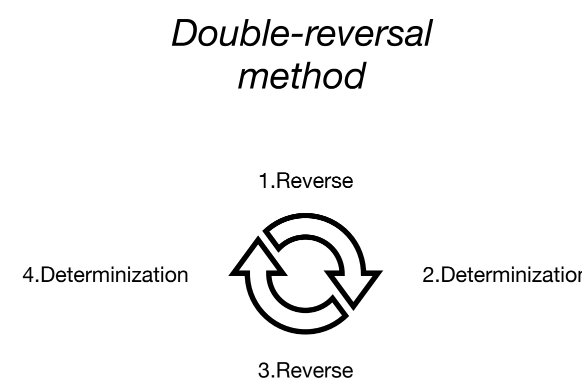


E. Gutiérrez, IMDEA Software, Madrid

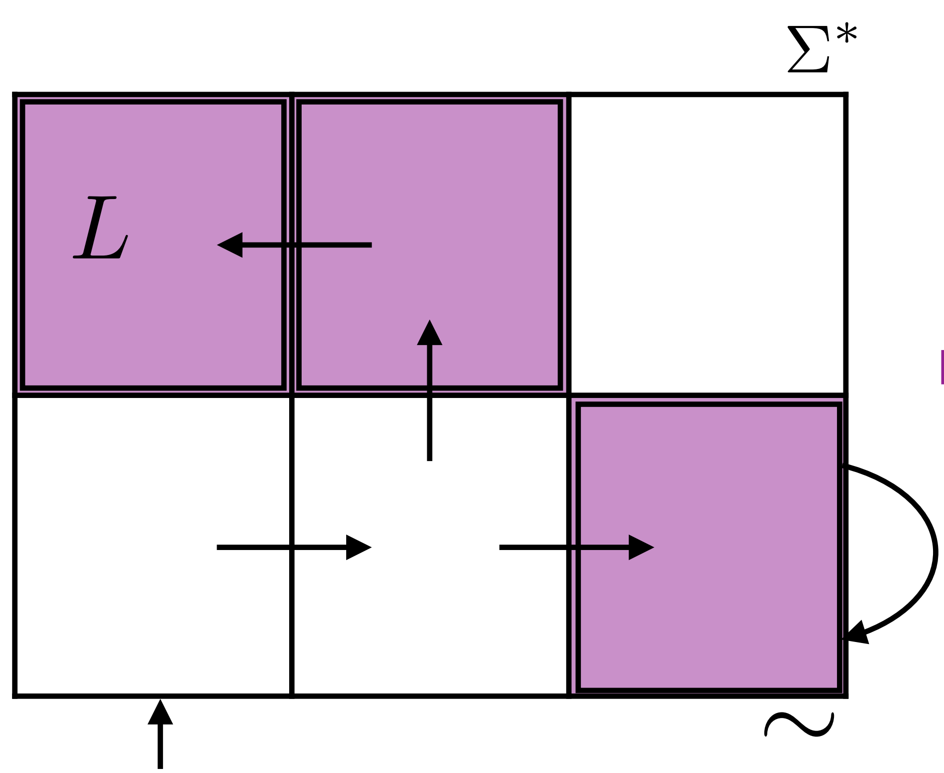
Language-theoretical Perspective



Partition of the set of states

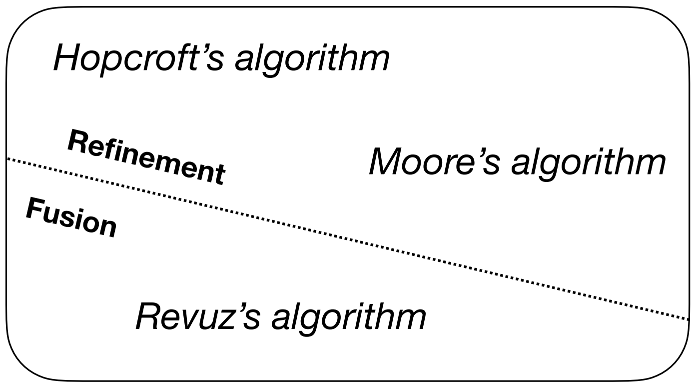


Combination of automata constructions



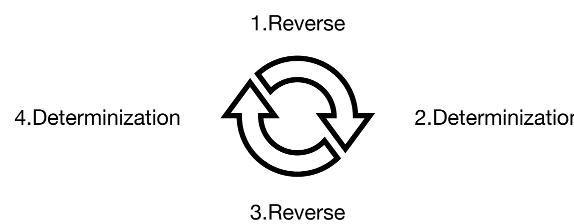
Nerode's equivalence relation on words

Language-theoretical Perspective

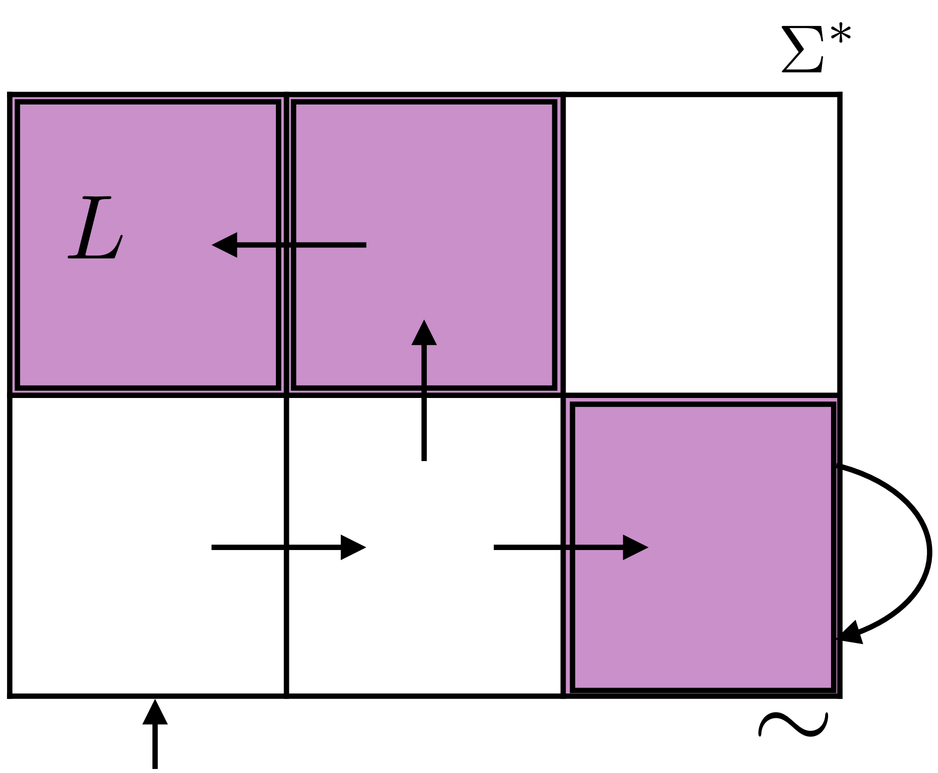


Partition of the set of states

Double-reversal method

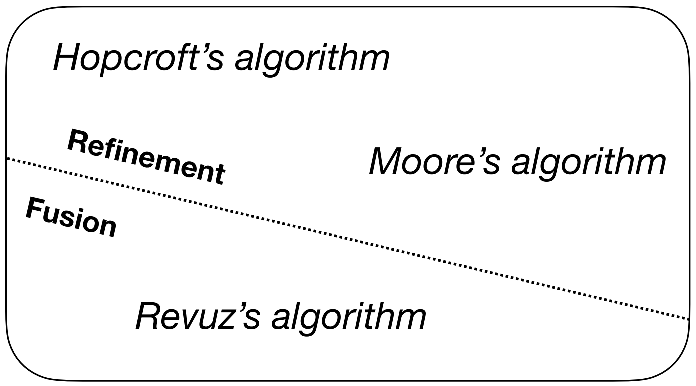


Combination of automata constructions



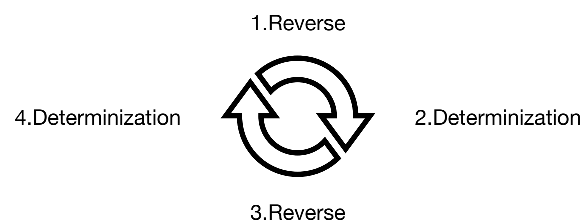
E. Gutiérrez, IMDEA Software, Madrid

Language-theoretical Perspective

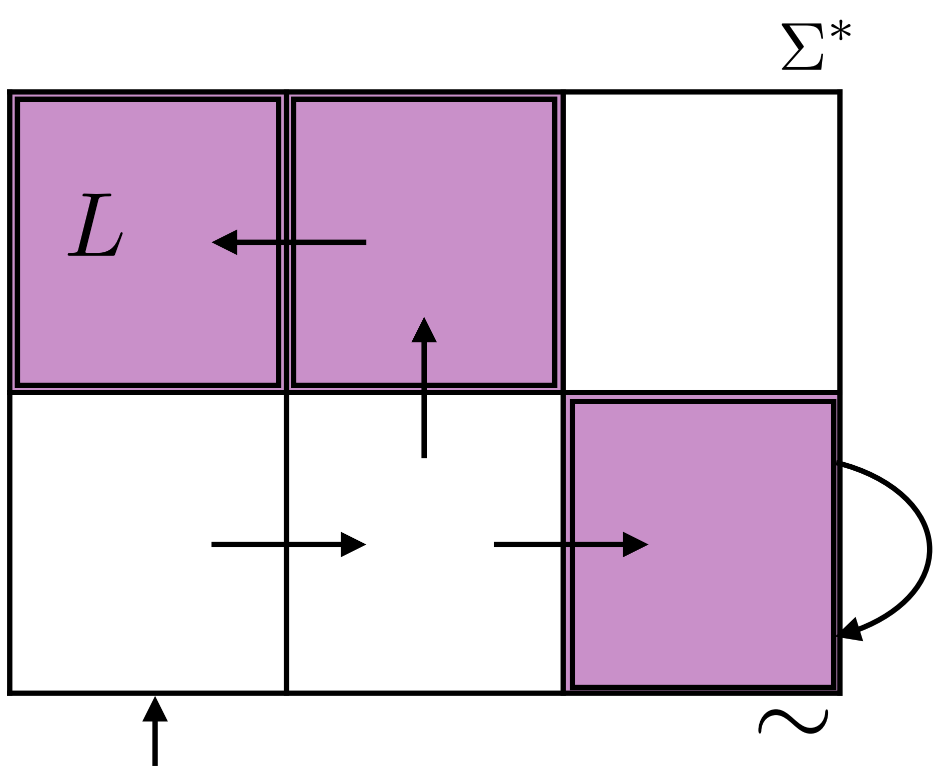


Partition of the set of states

Double-reversal method



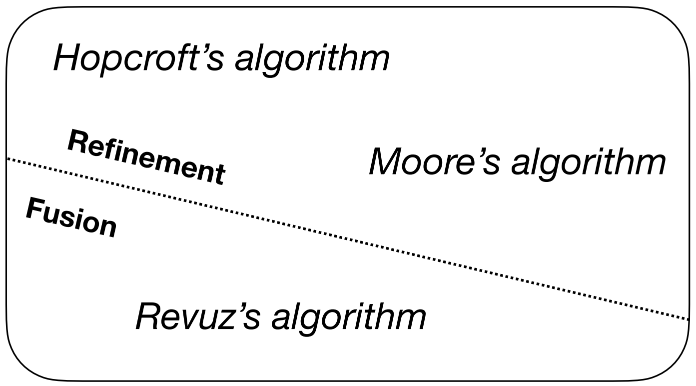
Combination of automata constructions



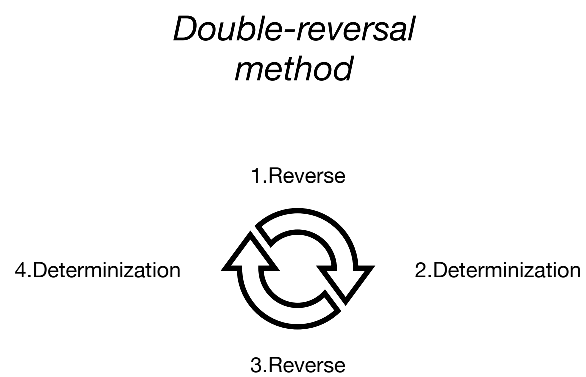
Contributions

- Automata constructions based on equivalences
- New simple proof of double-reversal method
- Revisit generalization of the double-reversal method
- Invariant of Moore's algorithm

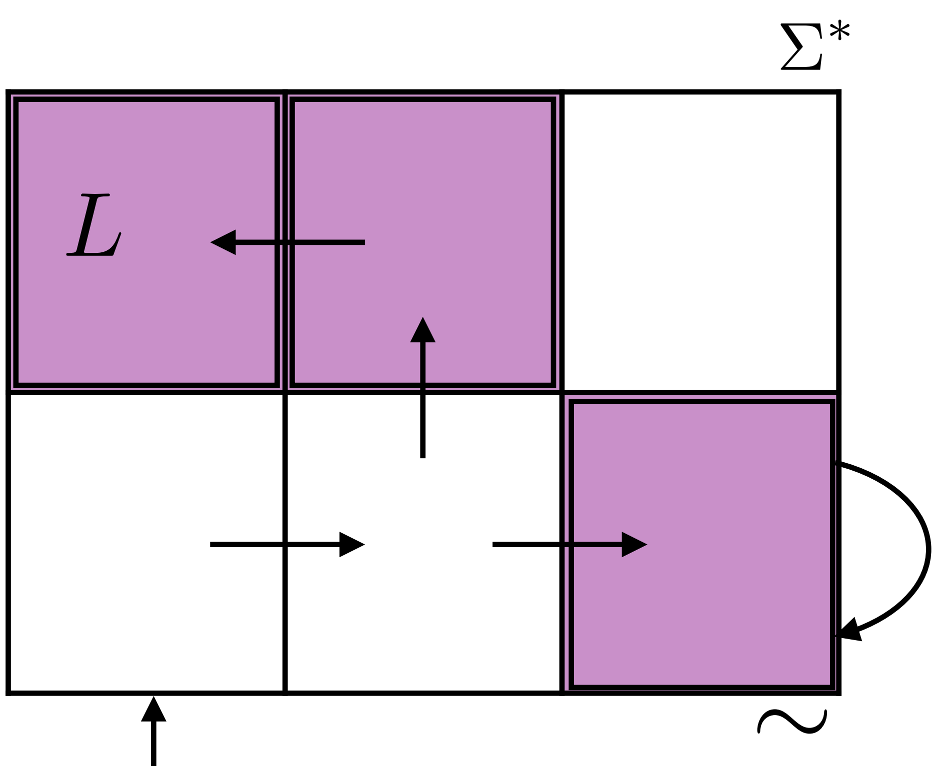
Language-theoretical Perspective



Partition of the set of states



Combination of automata constructions



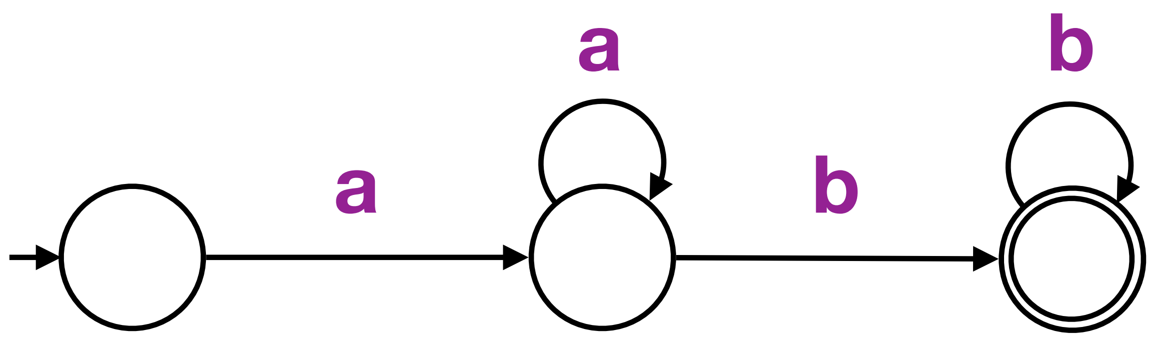
In this talk

- Automata constructions based on equivalences
- New simple proof of double-reversal method
- Revisit generalization of the double-reversal method
- Invariant of Moore's algorithm

The Basics

$\Sigma = \{ \mathbf{a}, \mathbf{b} \}$

$\mathbf{a}^+ \mathbf{b}^+$

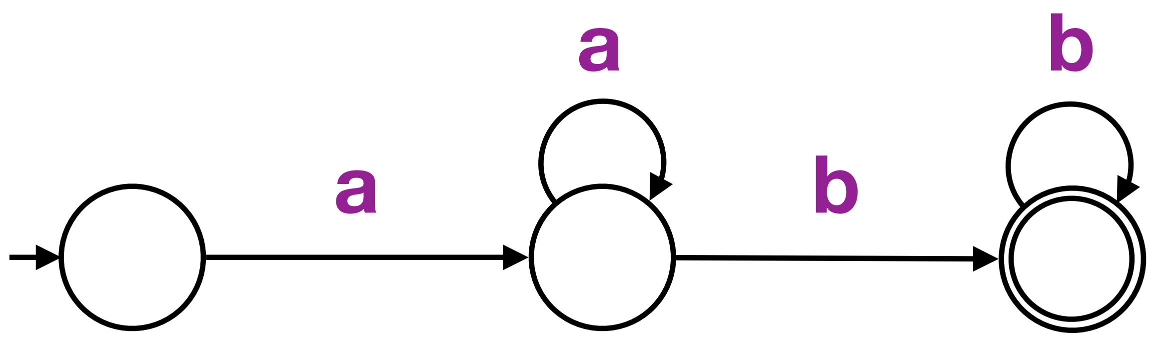


Deterministic
(DFA)

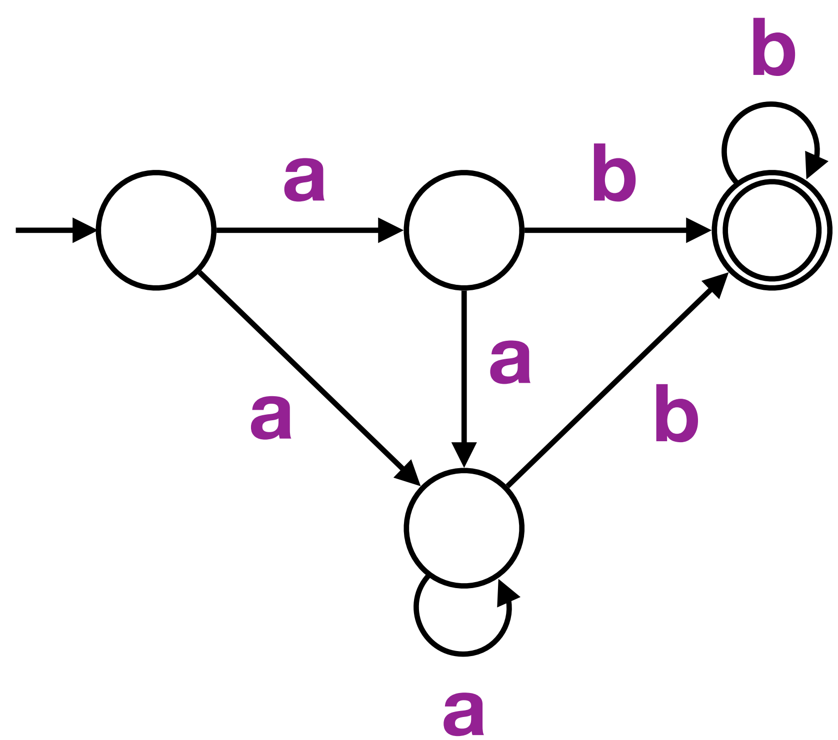
The Basics

$\Sigma = \{\, \mathsf{a} , \mathsf{b} \,\}$

$\mathsf{a}^{+}\mathsf{b}^{+}$



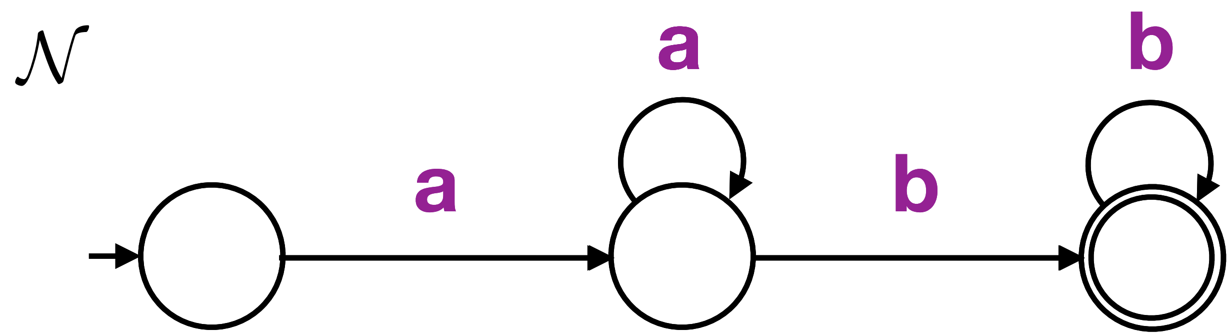
Deterministic
(DFA)



Nondeterministic
(NFA)

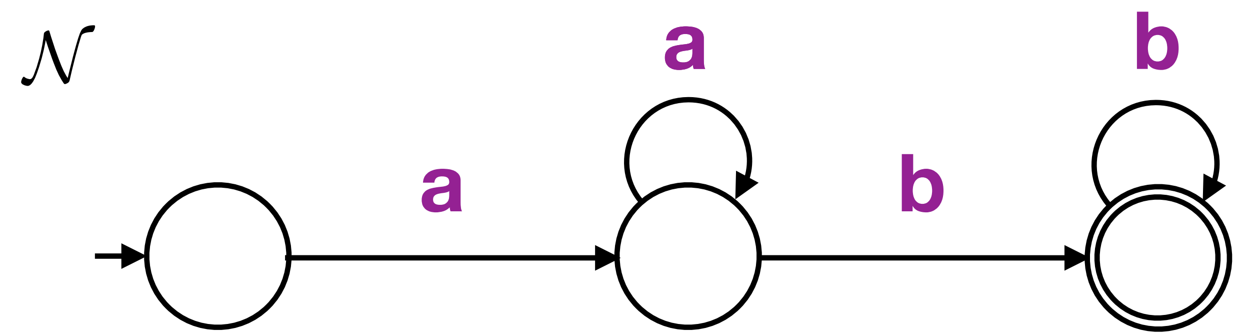
The Basics

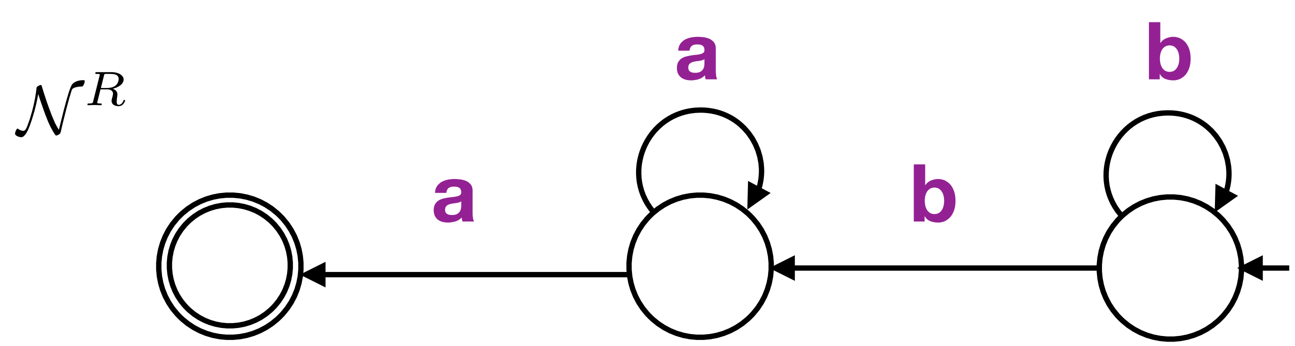
Reverse Construction



The Basics

Reverse Construction



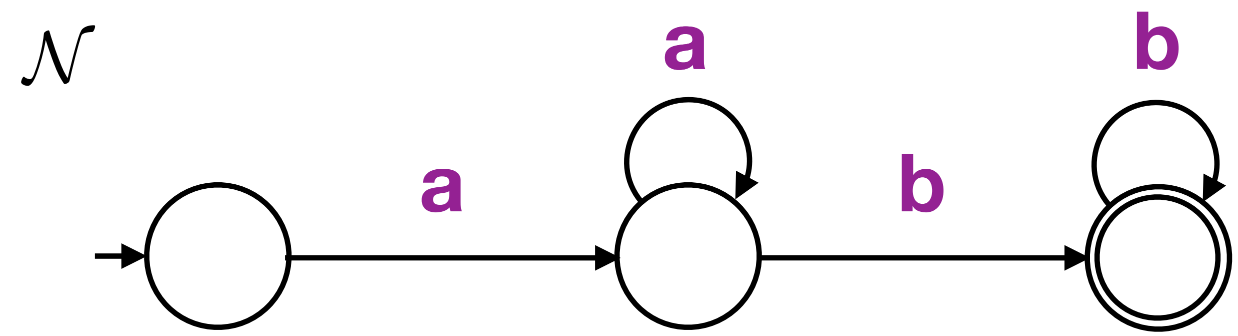


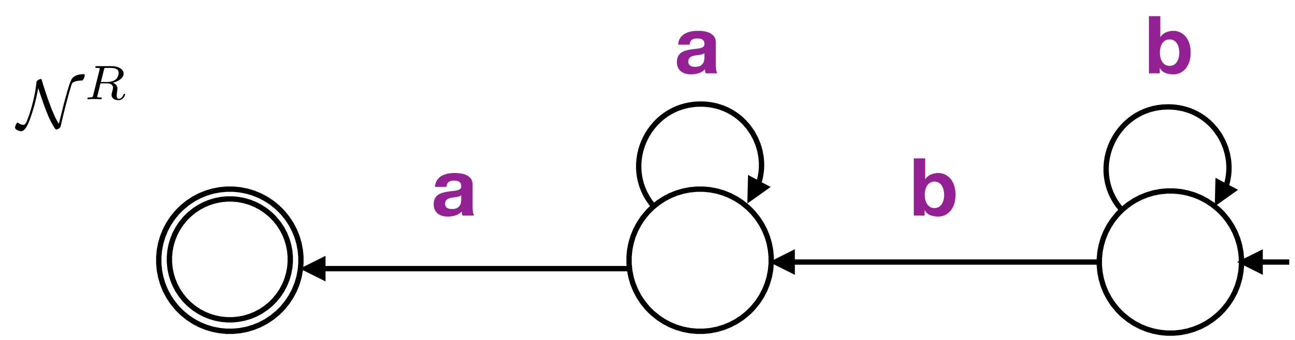
Reverse Automaton

$$\mathcal{L}(\mathcal{N}^R) = \{\, \mathbf{ba},\ \mathbf{baa},\ \mathbf{bbaa},\ \mathbf{bbaaa},\ \ldots \} = \mathbf{b}^+\mathbf{a}^+$$

The Basics

Co-deterministic automaton



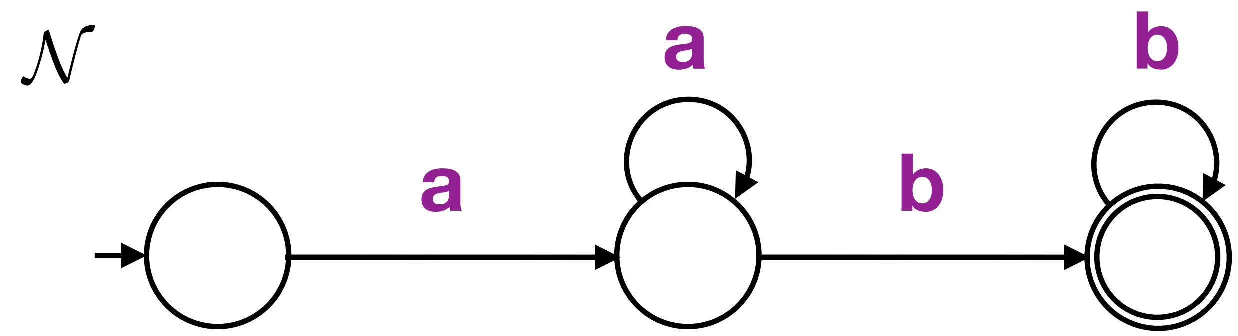


Reverse Automaton

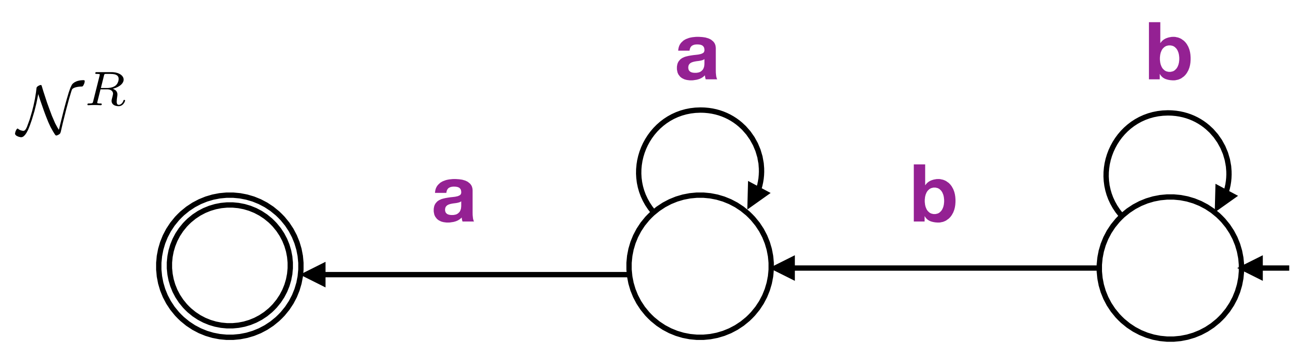
$$\mathcal{L}(\mathcal{N}^R) = \{\, \mathsf{ba}, \mathsf{baa}, \mathsf{bbaa}, \mathsf{bbaaa}, \ \ldots \,\} = \mathsf{b}^+\mathsf{a}^+$$

The Basics

Co-deterministic automaton



Co-deterministic (co-DFA)



Reverse Automaton

$$\mathcal{L}(\mathcal{N}^R) = \{\, \mathbf{ba},\ \mathbf{baa},\ \mathbf{bbaa},\ \mathbf{bbaaa},\ \ldots \,\} = \mathbf{b}^+\mathbf{a}^+$$

Congruences

Equivalences on words with good properties w.r.t. concatenation of symbols

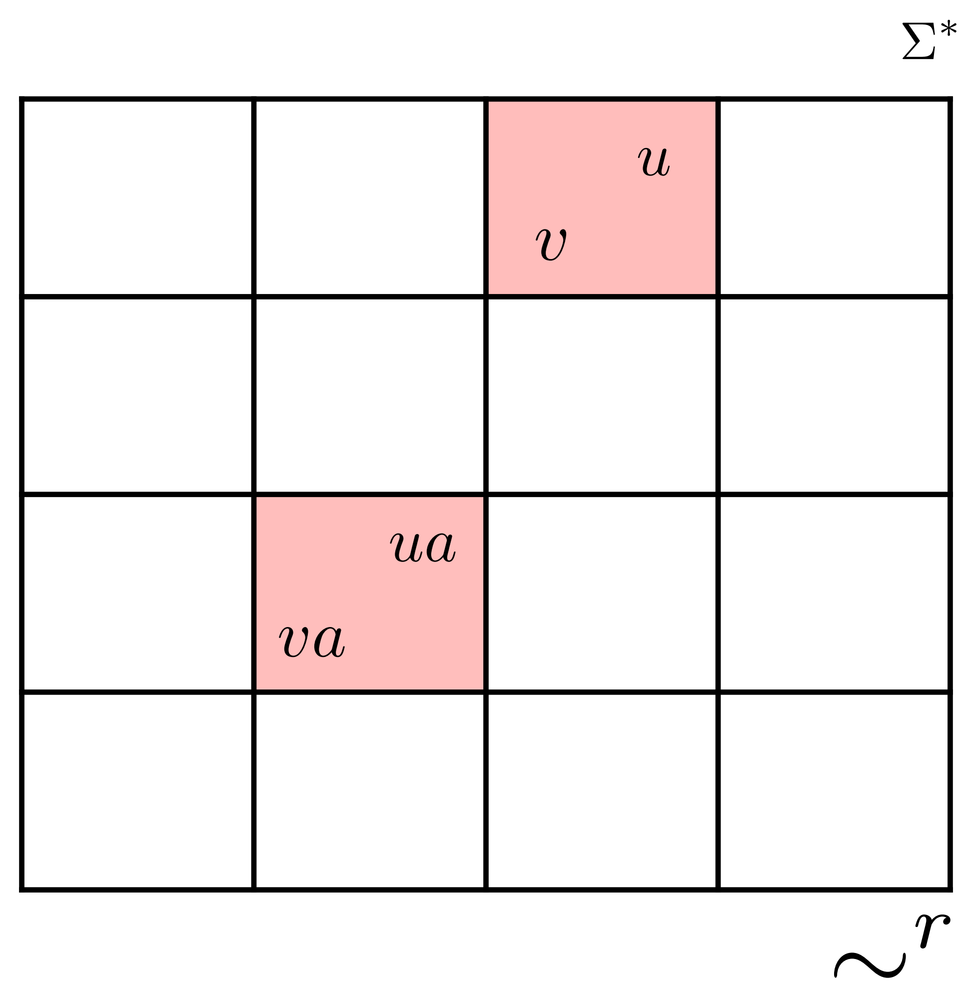
Congruences

Equivalences on words with good properties w.r.t. concatenation of symbols

Right congruences:

$$\forall a \in \Sigma :$$

$$u \sim^r v \Rightarrow ua \sim^r va$$



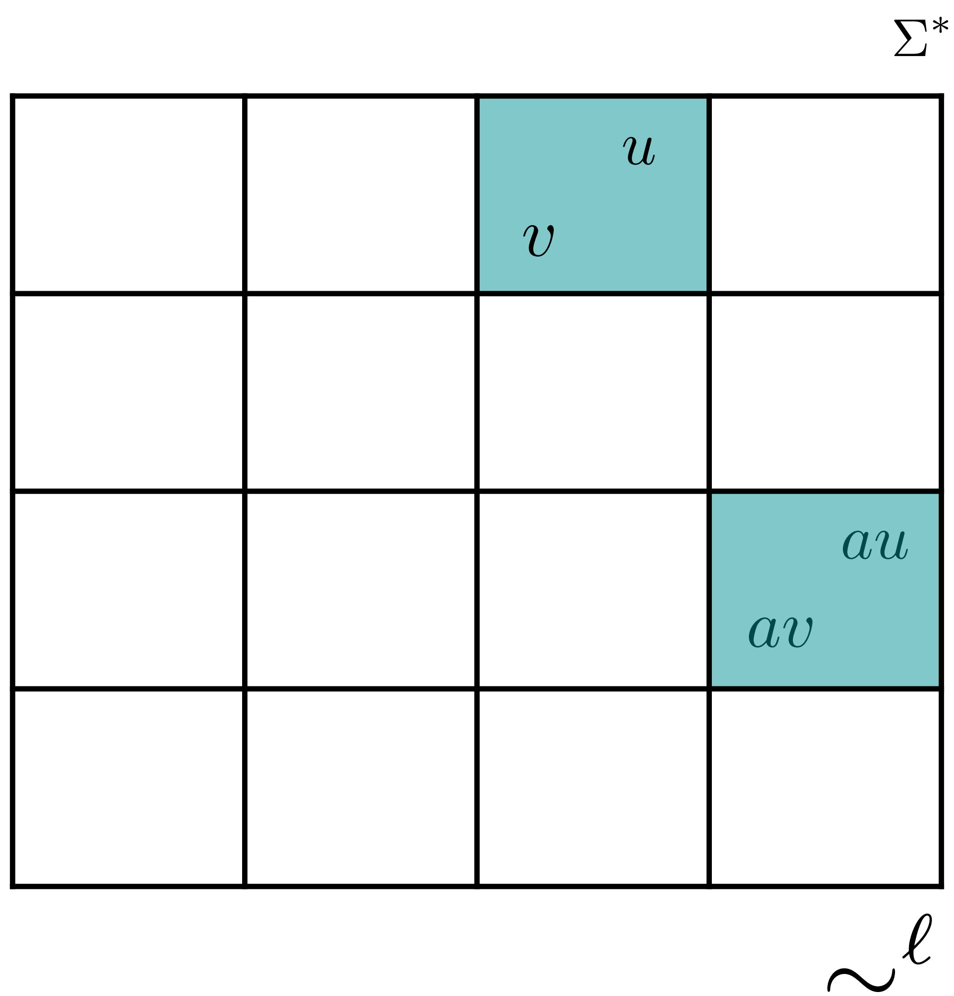
E. Gutiérrez, IMDEA Software, Madrid

Congruences

Equivalences on words with good properties w.r.t. concatenation of symbols

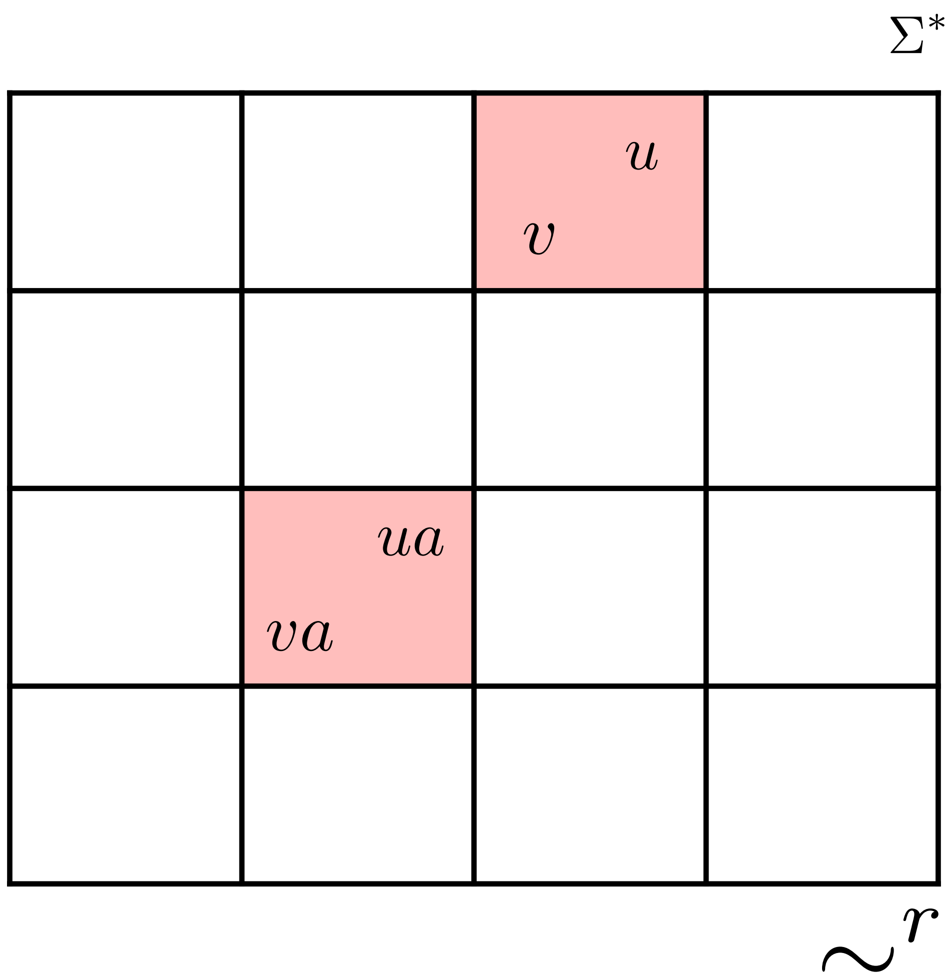
Left congruences:

$$\forall a \in \Sigma :$$
$$u \sim^{\ell} v \Rightarrow au \sim^{\ell} av$$



Right congruences:

$$\forall a \in \Sigma :$$
$$u \sim^{r} v \Rightarrow ua \sim^{r} va$$

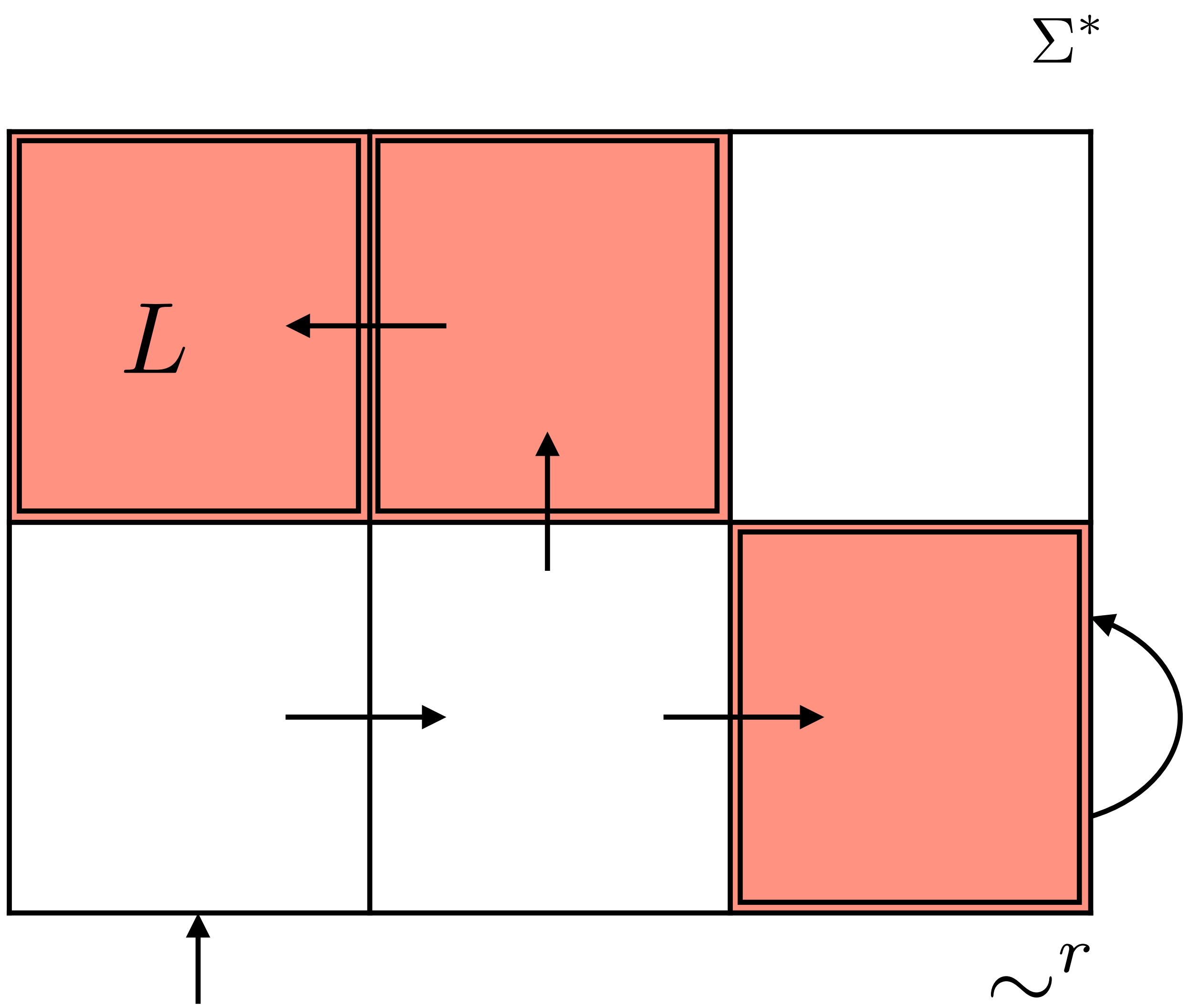


How to build a deterministic automaton from a right congruence

[Gutiérrez et. al, MFCS 2019]

[Hopcroft and Ullman, 1979]

- $\sim^r$ is a **finite** right congruence
- $P_{\sim^r}(L) = L$ ($\sim^r$ precisely represents L)

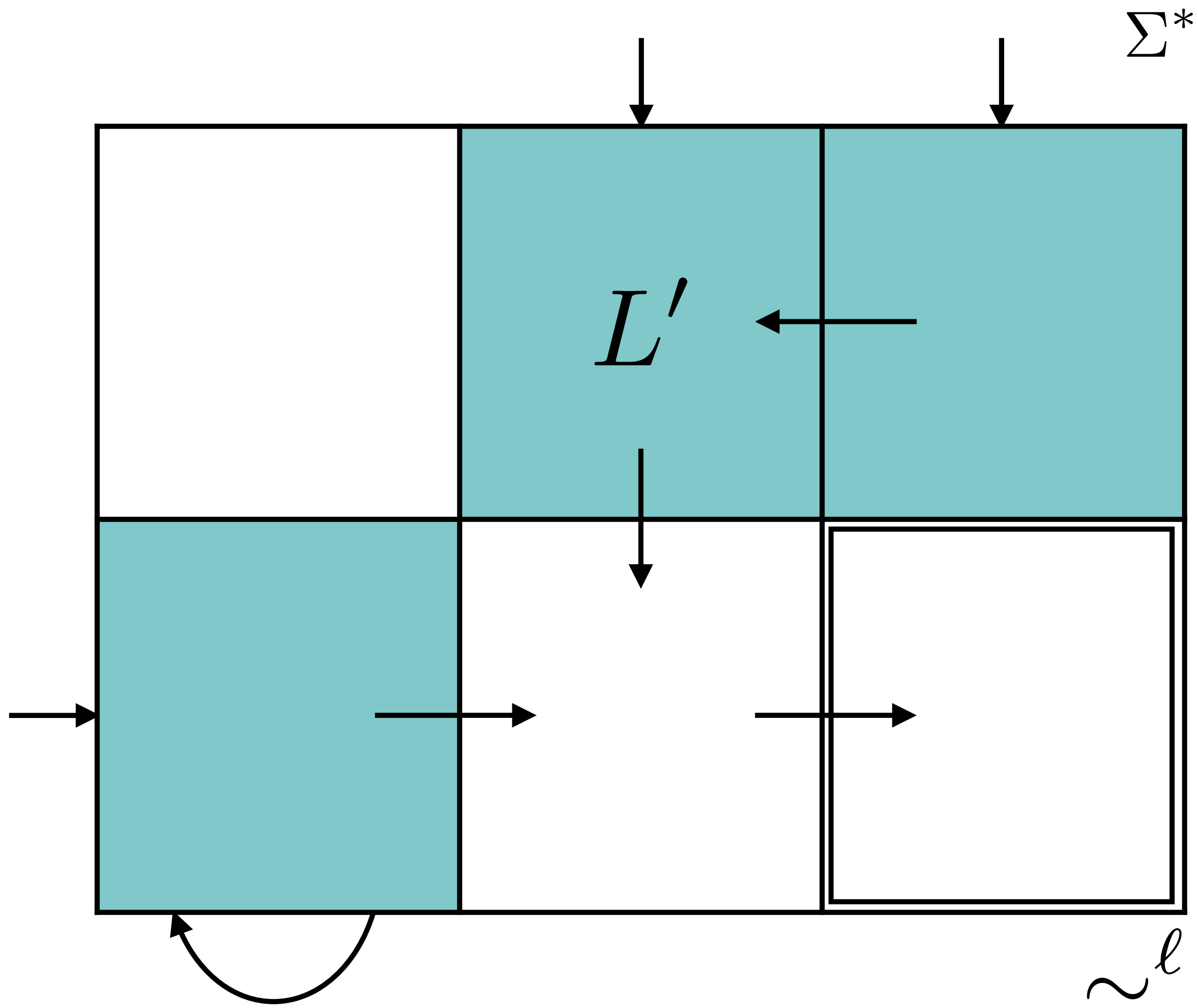


The DFA accepts L

How to build a co-deterministic automata from a left congruence

[Gutiérrez et. al, MFCS 2019]

- $\sim^{\ell}$ is a **finite** left congruence
- $P_{\sim^{\ell}}(L') = L'$ ($\sim^{\ell}$ precisely represents L')

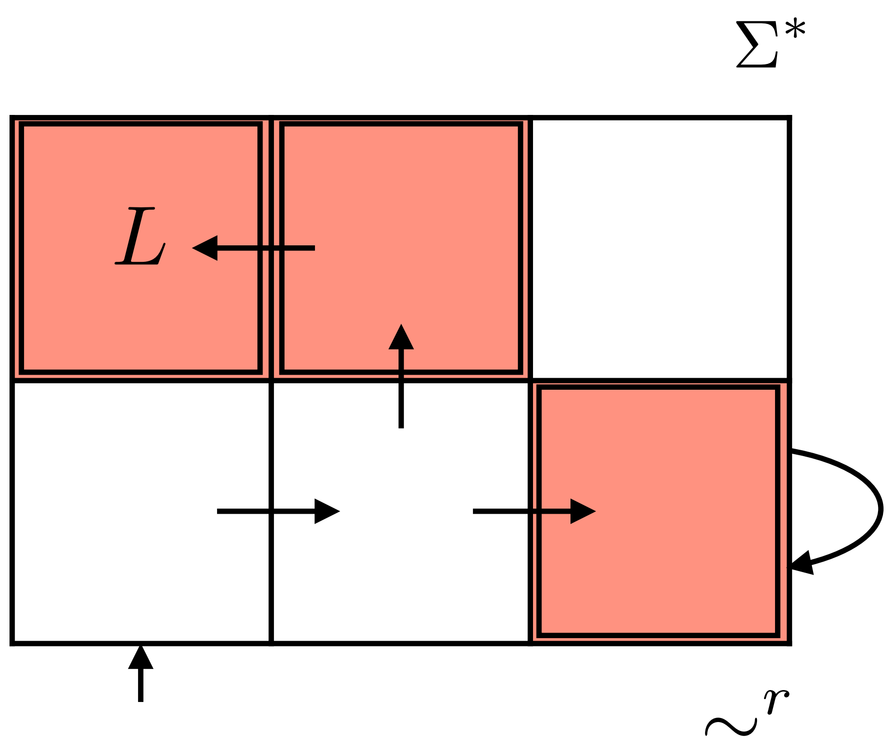


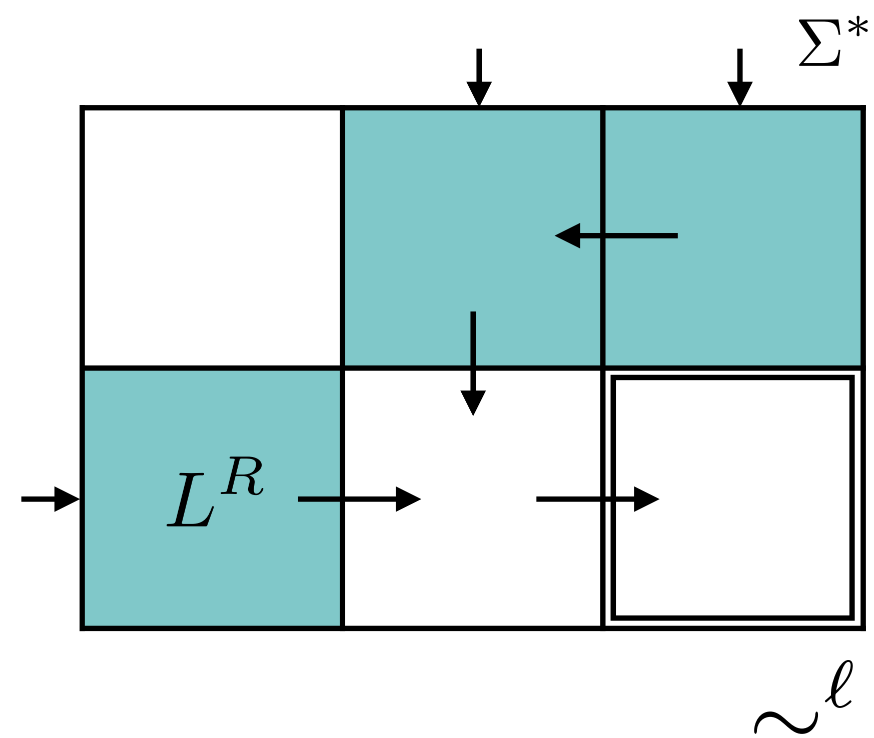
The co-DFA accepts L'

A property of dual congruences $\sim^r$ and $\sim^\ell$

[Gutiérrez et. al, MFCS 2019]

Lemma:



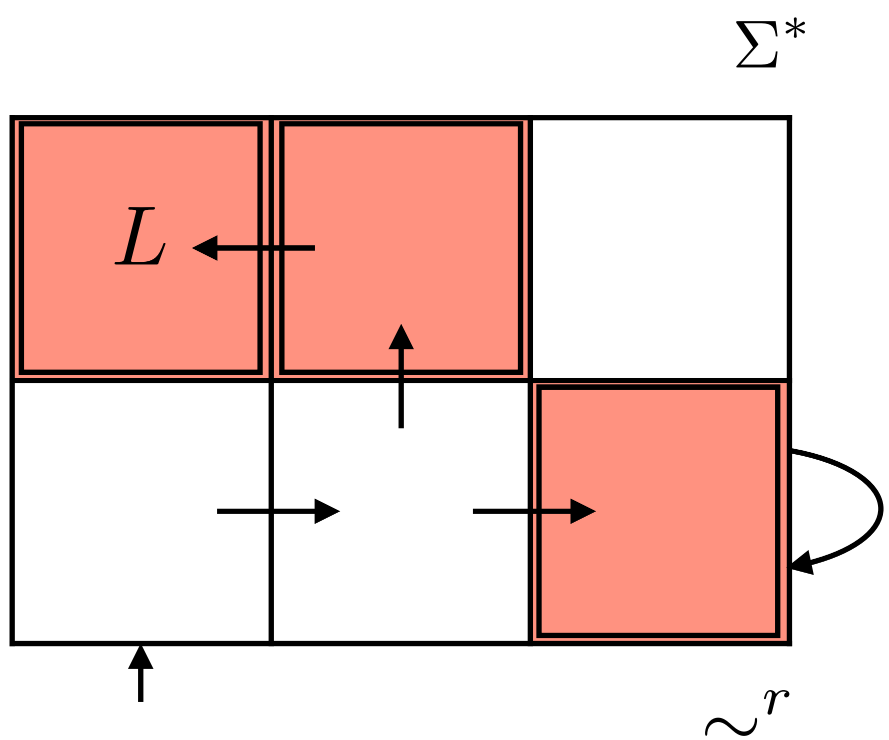


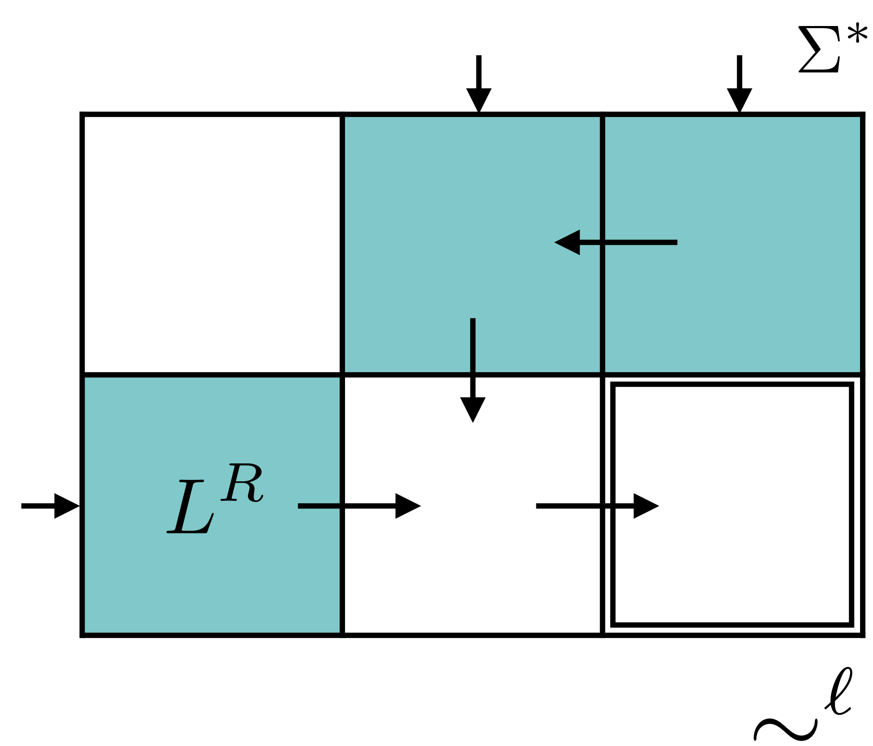
E. Gutiérrez, IMDEA Software, Madrid

A property of dual congruences $\sim^r$ and $\sim^\ell$

[Gutiérrez et. al, MFCS 2019]

Lemma: Given $u \sim^r v \Leftrightarrow u^R \sim^\ell v^R$

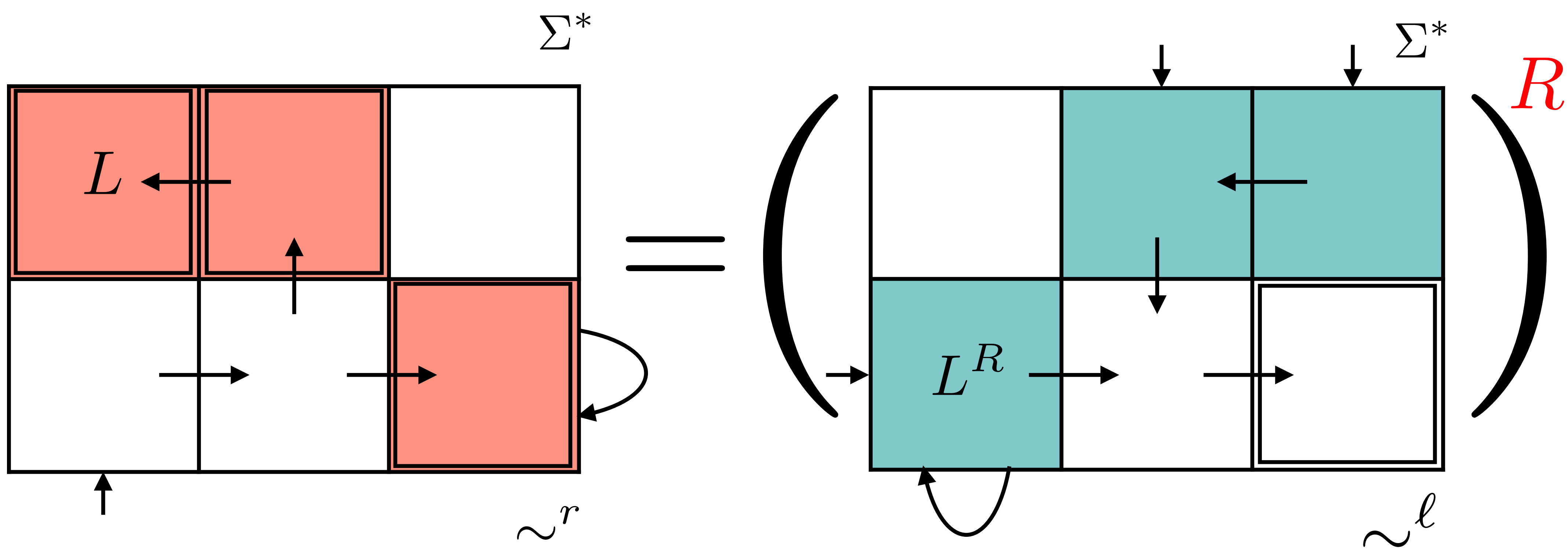




A property of dual congruences $\sim^r$ and $\sim^\ell$

[Gutiérrez et. al, MFCS 2019]

Lemma: Given $u \sim^r v \Leftrightarrow u^R \sim^\ell v^R$ then



Instances of right congruences

- $\sim^r$ is a **finite** right congruence
- $P_{\sim^r}(L) = L$

[Gutiérrez et. al, MFCS 2019]

Instances of right congruences

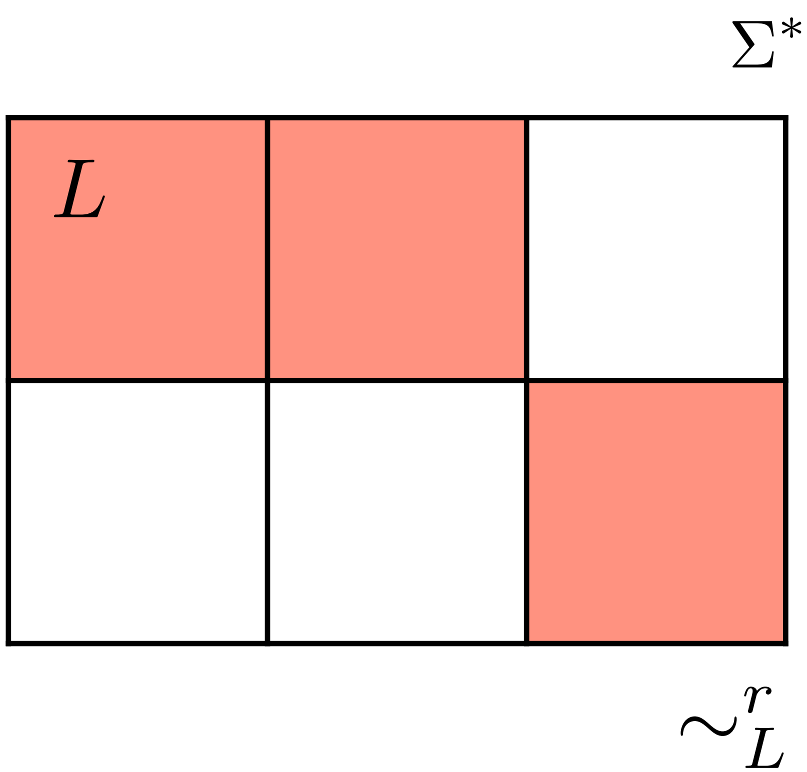
- $\sim^r$ is a **finite** right congruence
- $P_{\sim^r}(L) = L$

[Gutiérrez et. al, MFCS 2019]

Language-based [Büchi, 1989; Khoussainov and Nerode, 2001]

Given a regular language L

$$u \sim_L^r v \Leftrightarrow u^{-1}L = v^{-1}L$$



The minimal DFA for L

Instances of right congruences

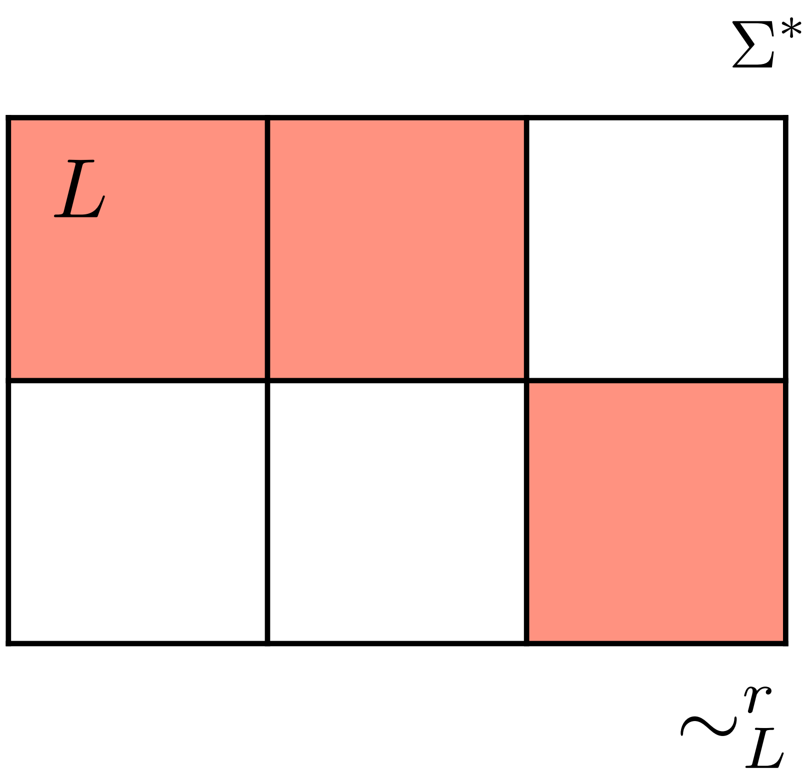
- $\sim^r$ is a **finite** right congruence
- $P_{\sim^r}(L) = L$

[Gutiérrez et. al, MFCS 2019]

Language-based [Büchi, 1989; Khoussainov and Nerode, 2001]

Given a regular language L

$$u \sim_L^r v \Leftrightarrow u^{-1}L = v^{-1}L$$



The minimal DFA for L

$$\mathsf{Min}^r(L)$$

Instances of right congruences

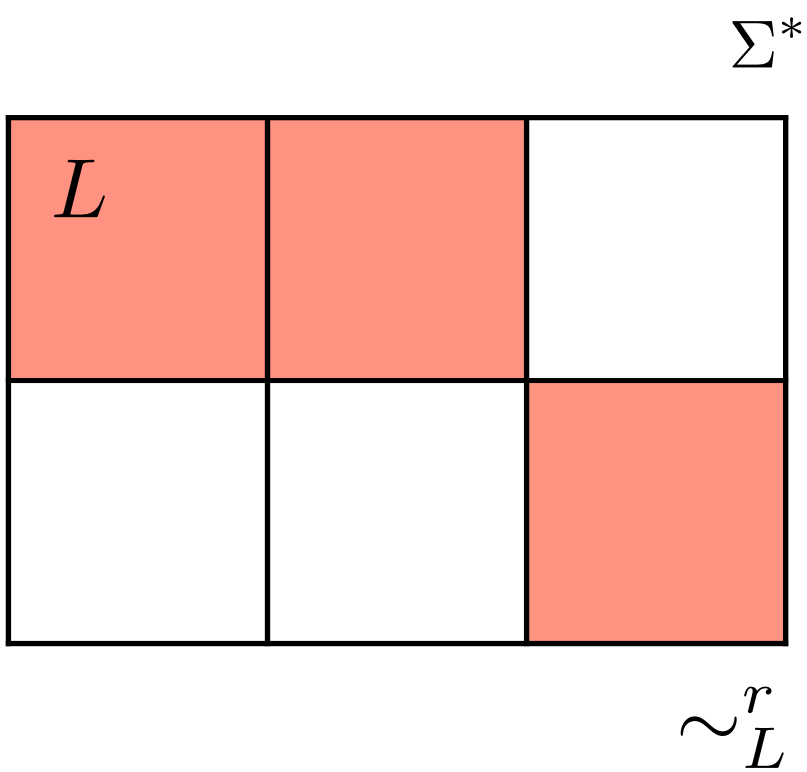
- $\sim^r$ is a **finite** right congruence
- $P_{\sim^r}(L) = L$

[Gutiérrez et. al, MFCS 2019]

Language-based [Büchi, 1989; Khoussainov and Nerode, 2001]

Given a regular language L

$$u \sim_L^r v \Leftrightarrow u^{-1}L = v^{-1}L$$



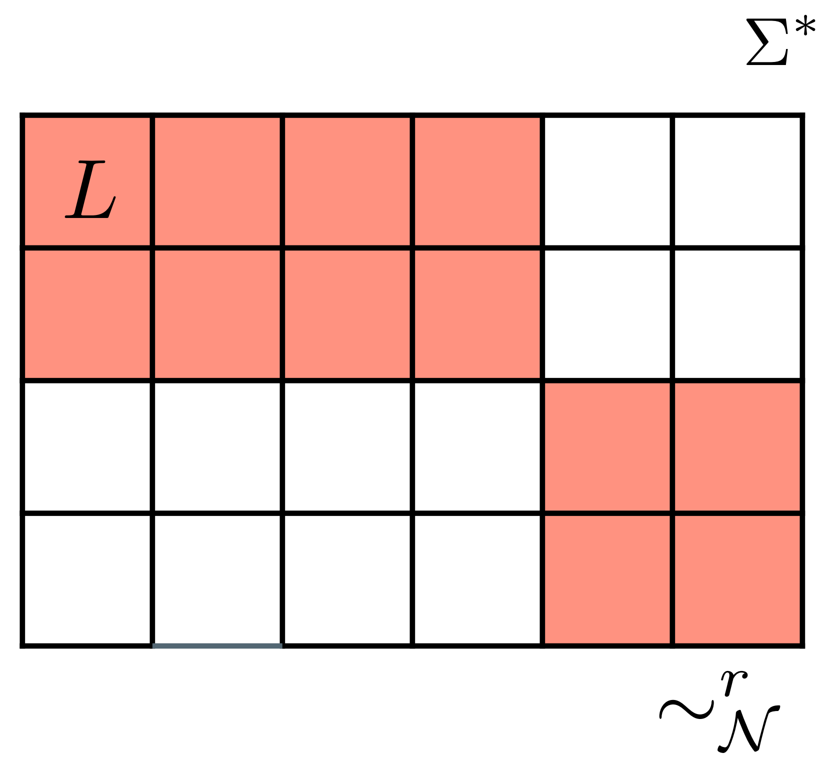
The minimal DFA for L

$\mathsf{Min}^r(L)$

Automata-based

Given an NFA $\mathcal{N}$ with $\mathcal{L}(\mathcal{N}) = L$

$$u \sim_{\mathcal{N}}^r v \Leftrightarrow \mathrm{post}_{\mathcal{N}}(u) = \mathrm{post}_{\mathcal{N}}(v)$$



A DFA for L : the usual "determinization" operation

Instances of right congruences

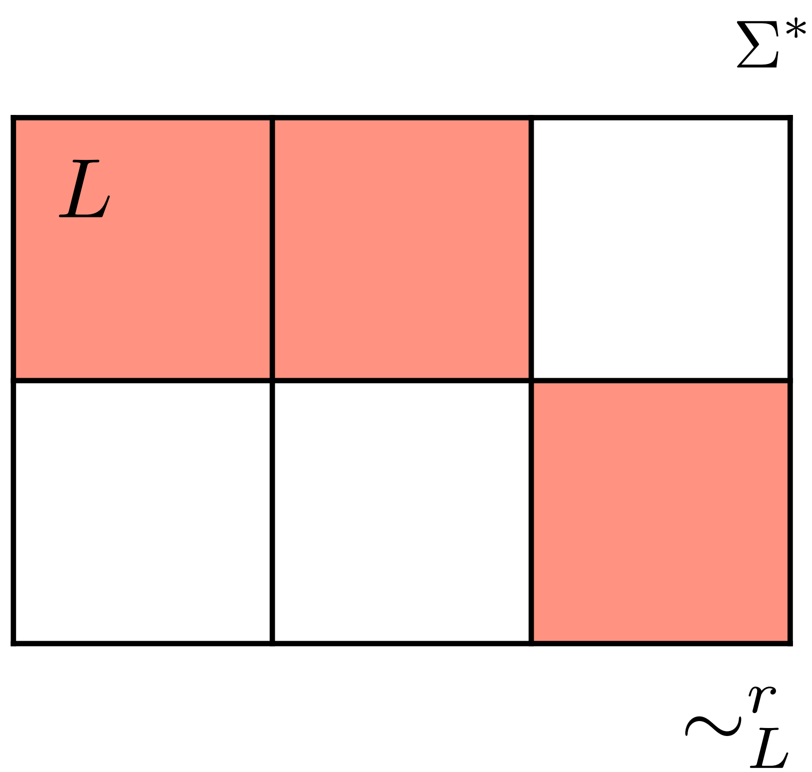
- $\sim^r$ is a **finite** right congruence
- $P_{\sim^r}(L) = L$

[Gutiérrez et. al, MFCS 2019]

Language-based [Büchi, 1989; Khoussainov and Nerode, 2001]

Given a regular language L

$$u \sim_L^r v \Leftrightarrow u^{-1}L = v^{-1}L$$



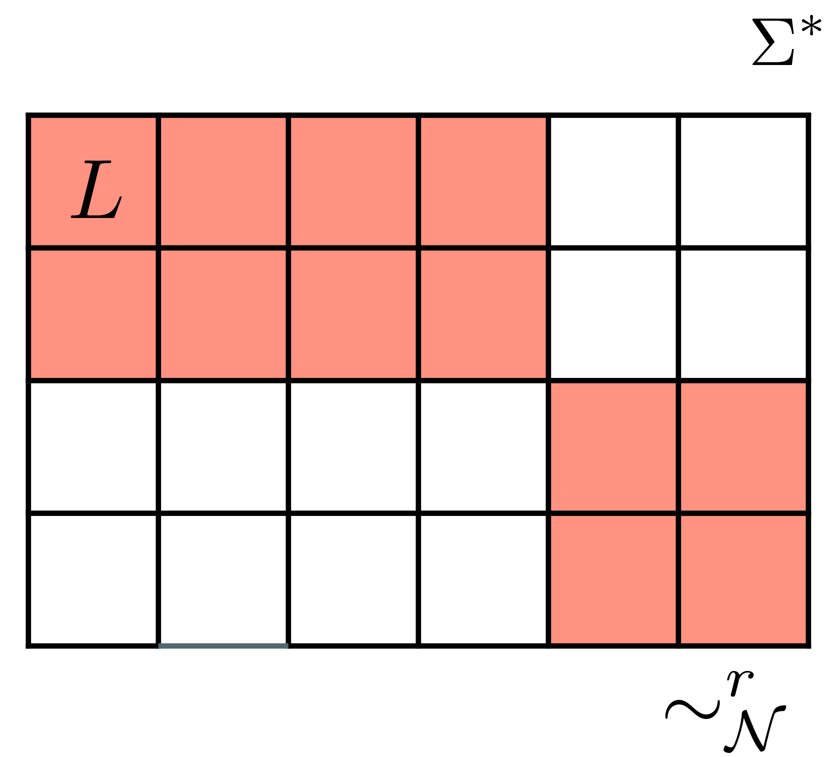
The minimal DFA for L

$\mathsf{Min}^r(L)$

Automata-based

Given an NFA $\mathcal{N}$ with $\mathcal{L}(\mathcal{N}) = L$

$$u \sim_{\mathcal{N}}^r v \Leftrightarrow \mathrm{post}_{\mathcal{N}}(u) = \mathrm{post}_{\mathcal{N}}(v)$$



A DFA for L : the usual "determinization" operation

$\mathsf{Det}^r(\mathcal{N})$

Instances of right congruences

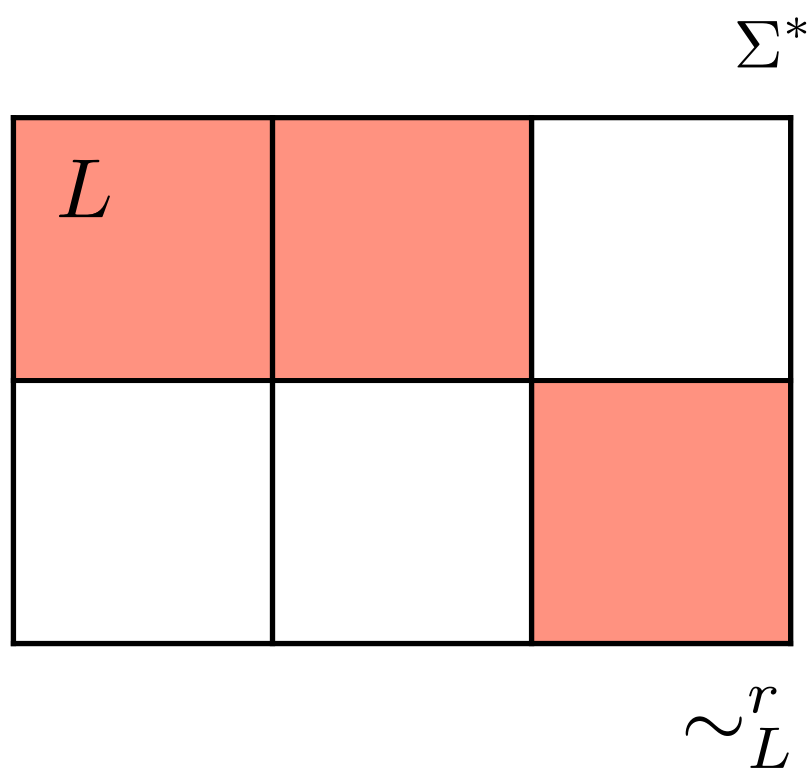
- $\sim^r$ is a **finite** right congruence
- $P_{\sim^r}(L) = L$

[Gutiérrez et. al, MFCS 2019]

Language-based [Büchi, 1989; Khoussainov and Nerode, 2001]

Given a regular language L

$$u \sim^r_L v \Leftrightarrow u^{-1}L = v^{-1}L$$



The minimal DFA for L

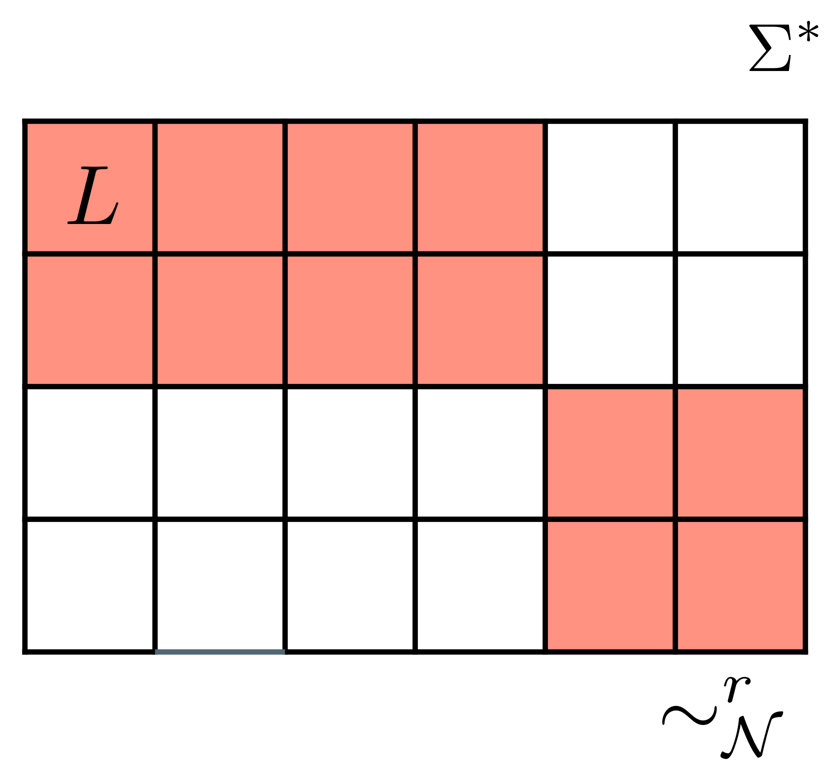
$\mathsf{Min}^r(L)$

$$\succeq$$

Automata-based

Given an NFA $\mathcal{N}$ with $\mathcal{L}(\mathcal{N}) = L$

$$u \sim^r_{\mathcal{N}} v \Leftrightarrow \mathrm{post}_{\mathcal{N}}(u) = \mathrm{post}_{\mathcal{N}}(v)$$



A DFA for L : the usual "determinization" operation

$\mathsf{Det}^r(\mathcal{N})$

Instances of right congruences

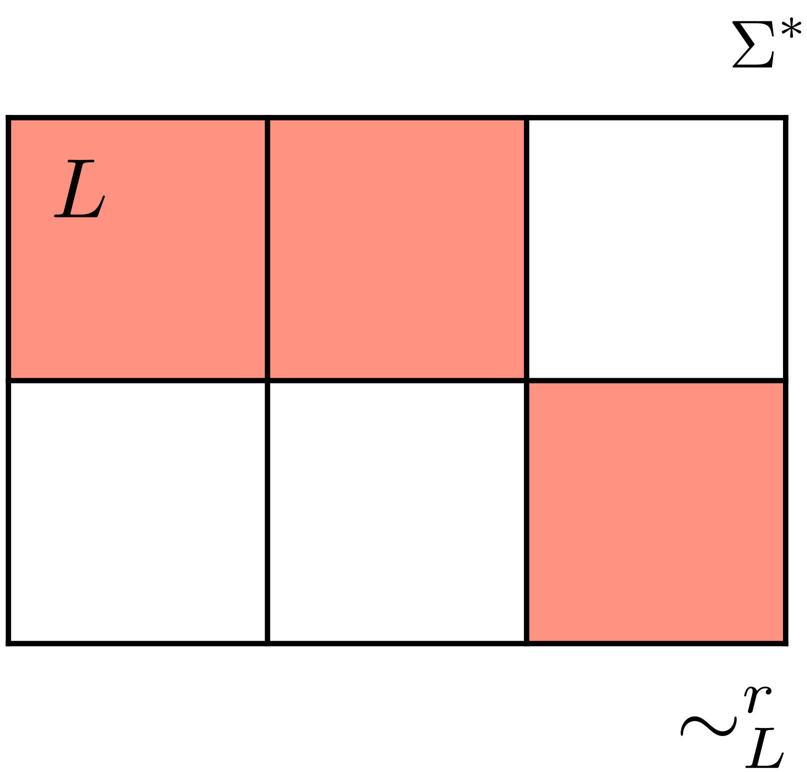
- $\sim^r$ is a **finite** right congruence
- $P_{\sim^r}(L) = L$

[Gutiérrez et. al, MFCS 2019]

Language-based [Büchi, 1989; Khoussainov and Nerode, 2001]

Given a regular language L

$$u \sim_L^r v \Leftrightarrow u^{-1}L = v^{-1}L$$

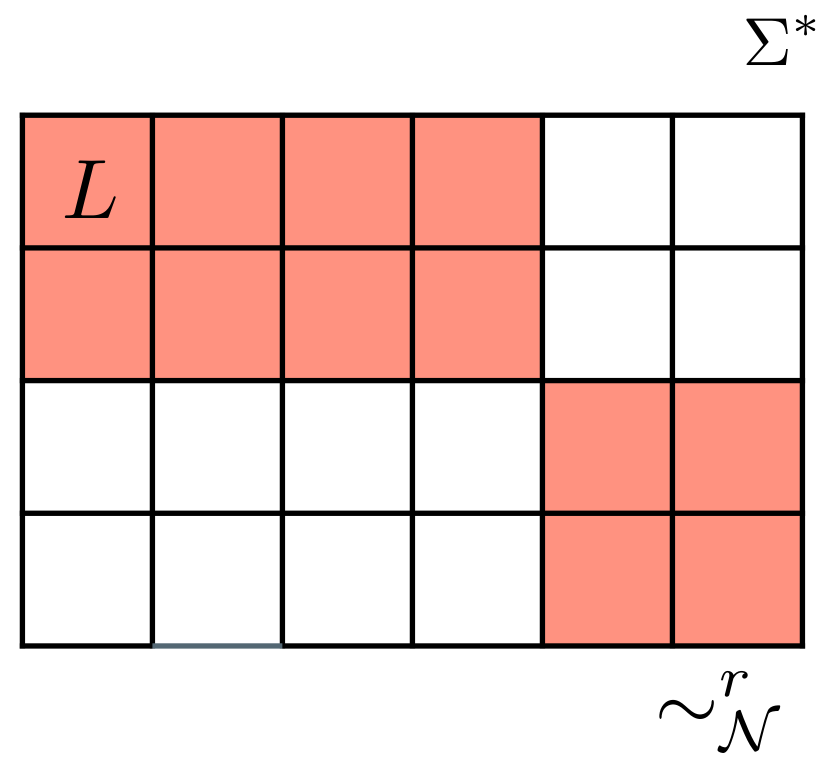


$$\stackrel{?}{=}$$

Automata-based

Given an NFA $\mathcal{N}$ with $\mathcal{L}(\mathcal{N}) = L$

$$u \sim_{\mathcal{N}}^r v \Leftrightarrow \mathrm{post}_{\mathcal{N}}(u) = \mathrm{post}_{\mathcal{N}}(v)$$



Instances of right congruences

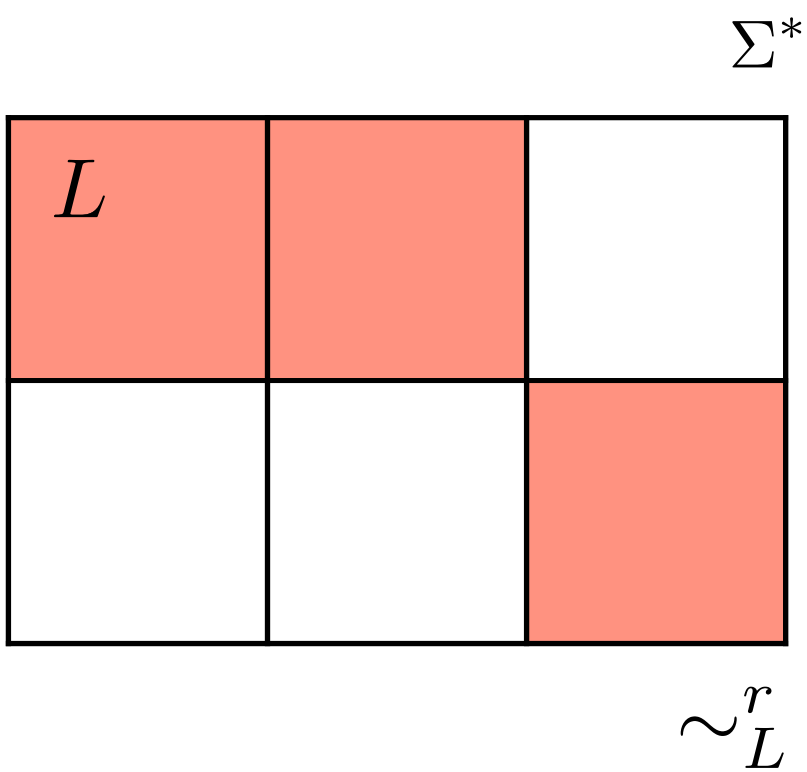
- $\sim^r$ is a **finite** right congruence
- $P_{\sim^r}(L) = L$

[Gutiérrez et. al, MFCS 2019]

Language-based [Büchi, 1989; Khoussainov and Nerode, 2001]

Given a regular language L

$$u \sim_L^r v \Leftrightarrow u^{-1}L = v^{-1}L$$

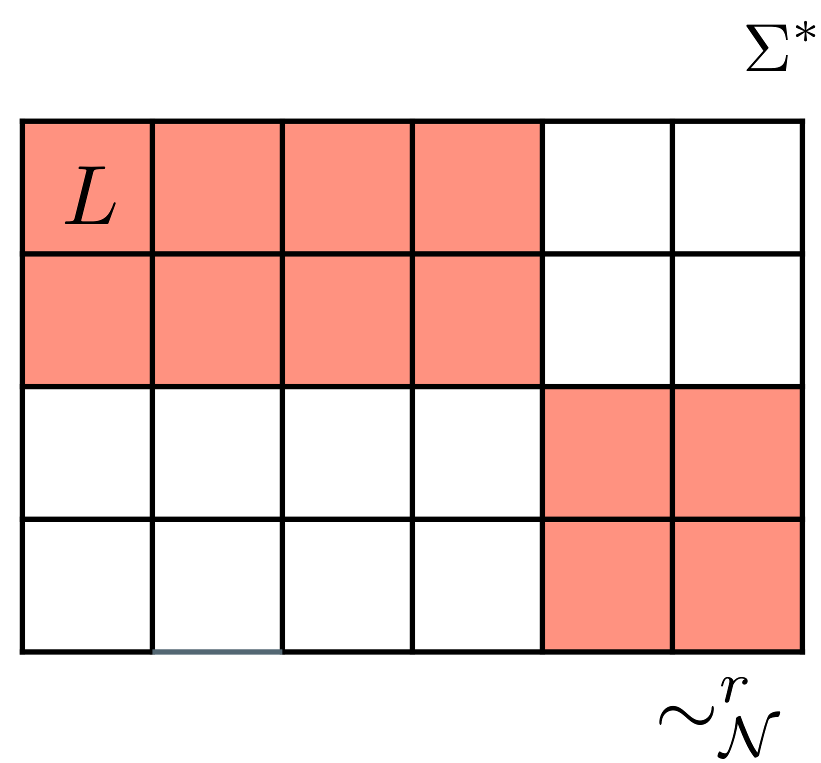


$$\stackrel{?}{=}$$

Automata-based

Given an NFA $\mathcal{N}$ with $\mathcal{L}(\mathcal{N}) = L$

$$u \sim_{\mathcal{N}}^r v \Leftrightarrow \mathrm{post}_{\mathcal{N}}(u) = \mathrm{post}_{\mathcal{N}}(v)$$



Lemma:

$$\sim_L^r = \sim_{\mathcal{N}}^r \text{ iff } u^{-1}L = v^{-1}L \Leftrightarrow \mathrm{post}_{\mathcal{N}}(u) = \mathrm{post}_{\mathcal{N}}(v)$$

Instances of left congruences

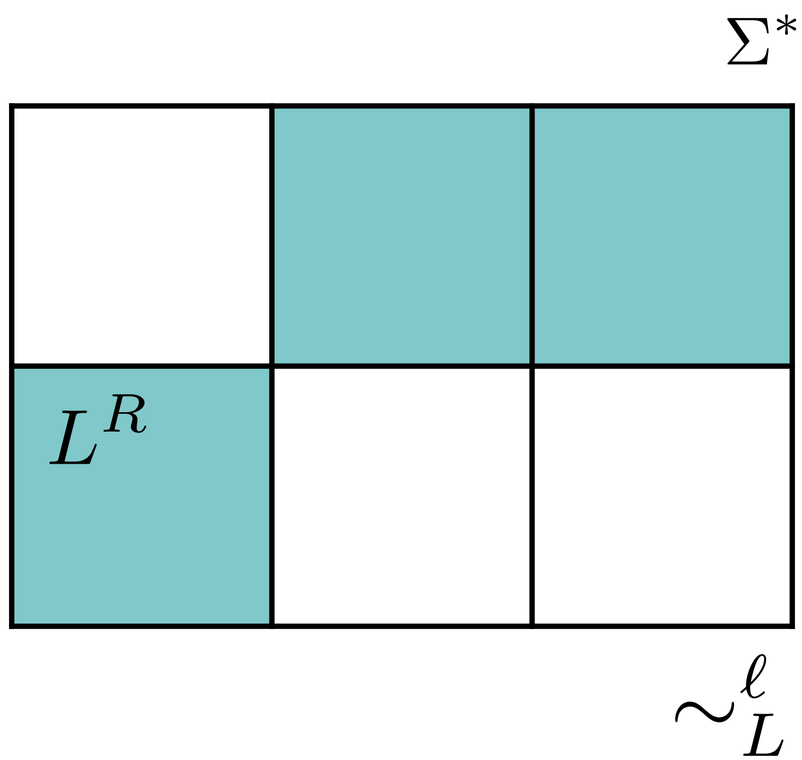
- $\sim_\ell$ is a **finite** left congruence
- $P_{\sim_\ell}(L) = L$

[Gutiérrez et. al, MFCS 2019]

Language-based

Given a regular language L

$$u \sim^\ell_L v \Leftrightarrow Lu^{-1} = Lv^{-1}$$



The minimal co-DFA for L

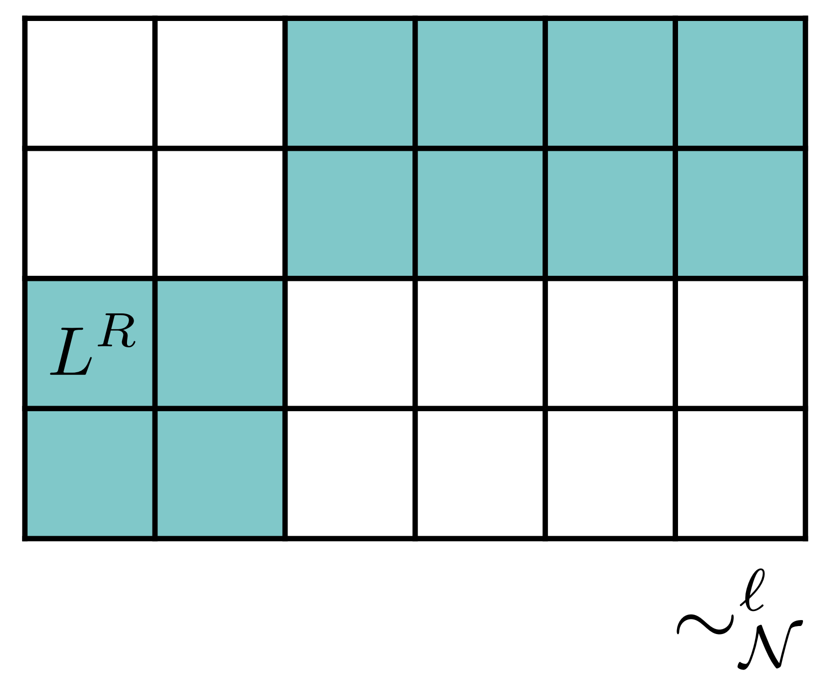
$\mathsf{Min}^\ell(L)$

$$\succeq$$

Automata-based

Given an NFA $\mathcal{N}$ with $\mathcal{L}(\mathcal{N}) = L$

$$u \sim^\ell_{\mathcal{N}} v \Leftrightarrow \mathrm{pre}_{\mathcal{N}}(u) = \mathrm{pre}_{\mathcal{N}}(v)$$



A co-DFA for L

$\mathsf{Det}^\ell(\mathcal{N})$

Double-reversal Method

[Gutiérrez et. al, MFCS 2019]

$$
\begin{tikzcd}
\mathcal{N} \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^R
\end{tikzcd}
$$

Double-reversal Method

[Gutiérrez et. al, MFCS 2019]

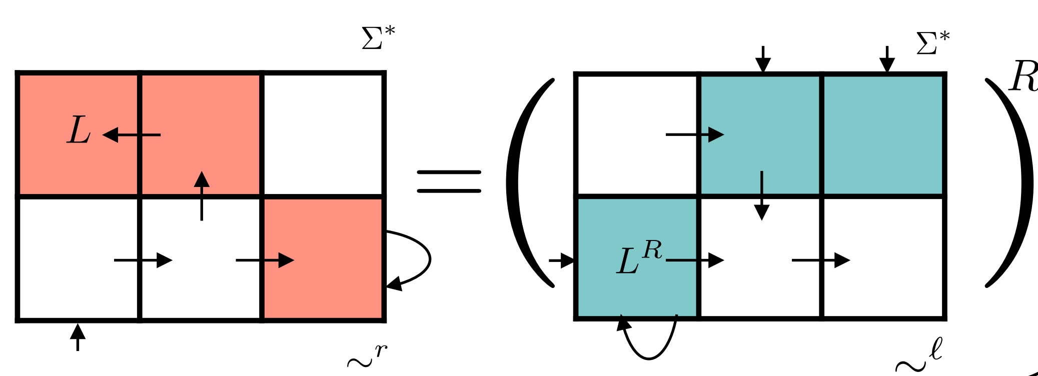
$$
\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{\ell}(\mathcal{N}) \\
\mathcal{N}^R \arrow[r, "\mathsf{Det}^{r}"] & \mathsf{Det}^{r}(\mathcal{N}^R)
\end{tikzcd}
$$

Double-reversal Method

[Gutiérrez et. al, MFCS 2019]

$$\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{\ell}(\mathcal{N}) \\
\mathcal{N}^R \arrow[r, "\mathsf{Det}^{r}"] & \mathsf{Det}^{r}(\mathcal{N}^R)
\end{tikzcd}$$

Lemma: Given $u \sim^r v \Leftrightarrow u^R \sim^\ell v^R$ then

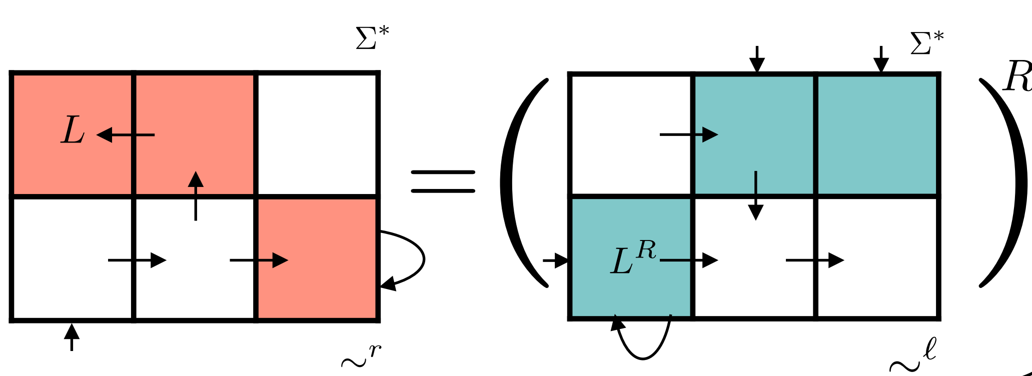


Double-reversal Method

[Gutiérrez et. al, MFCS 2019]

$$\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^R \arrow[r, "\mathsf{Det}^{r}"] & \mathsf{Det}^{r}(\mathcal{N}^R)
\end{tikzcd}$$

Lemma: Given $u \sim^r v \Leftrightarrow u^R \sim^\ell v^R$ then



Double-reversal Method

[Gutiérrez et. al, MFCS 2019]

$$\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^{R} \arrow[r, "\mathsf{Det}^{r}"] & \mathsf{Det}^{r}(\mathcal{N}^{R})
\end{tikzcd}$$

Double-reversal Method

[Gutiérrez et. al, MFCS 2019]

$$\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^R \arrow[r, "\mathsf{Det}^{r}"] & \mathsf{Det}^{r}(\mathcal{N}^R)
\end{tikzcd}$$

Lemma:

$$\sim^r_L = \sim^r_{\mathcal{N}} \;\text{ iff }\; u^{-1}L = v^{-1}L \Leftrightarrow \mathrm{post}_{\mathcal{N}}(u) = \mathrm{post}_{\mathcal{N}}(v)$$

Double-reversal Method

[Gutiérrez et. al, MFCS 2019]

$$
\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^R \arrow[r, "\mathsf{Det}^{r}"] & \mathsf{Det}^{r}(\mathcal{N}^R)
\end{tikzcd}
$$

Lemma:

True if $\mathcal{N}$ is a co-DFA

$$\sim_L^r = \sim_{\mathcal{N}}^r \quad \text{iff } u^{-1}L = v^{-1}L \Leftrightarrow \mathrm{post}_{\mathcal{N}}(u) = \mathrm{post}_{\mathcal{N}}(v)$$

Double-reversal Method

[Gutiérrez et. al, MFCS 2019]

$$
\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[r, "\mathsf{Det}^{r}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{r}(\mathsf{Det}^{\ell}(\mathcal{N})) \\
\mathcal{N}^R \arrow[r, "\mathsf{Det}^{r}"] & \mathsf{Det}^{r}(\mathcal{N}^R) &
\end{tikzcd}
$$

Lemma:

True if $\mathcal{N}$ is a co-DFA

$$\sim^r_L = \sim^r_{\mathcal{N}} \quad \text{iff } u^{-1}L = v^{-1}L \Leftrightarrow \mathrm{post}_{\mathcal{N}}(u) = \mathrm{post}_{\mathcal{N}}(v)$$

Double-reversal Method

[Gutiérrez et. al, MFCS 2019]

$$
\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] \arrow[rr, bend left, dotted, "\mathsf{Min}^{r}"] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[r, "\mathsf{Det}^{r}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{r}(\mathsf{Det}^{\ell}(\mathcal{N})) \\
\mathcal{N}^{R} \arrow[r, "\mathsf{Det}^{r}"] & \mathsf{Det}^{r}(\mathcal{N}^{R}) &
\end{tikzcd}
$$

Lemma:

True if $\mathcal{N}$ is a co-DFA

$$\sim^r_L = \sim^r_{\mathcal{N}} \quad \text{iff } u^{-1}L = v^{-1}L \Leftrightarrow \text{post}_{\mathcal{N}}(u) = \text{post}_{\mathcal{N}}(v)$$

Double-reversal Method

[Gutiérrez et. al, MFCS 2019]

$$\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] \arrow[rr, bend left, dotted, "\mathsf{Min}^{r}"] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[r, "\mathsf{Det}^{r}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{r}(\mathsf{Det}^{\ell}(\mathcal{N})) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^{R} \arrow[r, "\mathsf{Det}^{r}"] \arrow[rr, bend right, dotted, "\mathsf{Min}^{\ell}"'] & \mathsf{Det}^{r}(\mathcal{N}^{R}) \arrow[r, "\mathsf{Det}^{\ell}"] & \mathsf{Det}^{\ell}(\mathsf{Det}^{r}(\mathcal{N}^{R}))
\end{tikzcd}$$

Lemma:

True if $\mathcal{N}$ is a co-DFA

$$\sim^{r}_{L} = \sim^{r}_{\mathcal{N}} \quad \text{iff } u^{-1}L = v^{-1}L \Leftrightarrow \mathrm{post}_{\mathcal{N}}(u) = \mathrm{post}_{\mathcal{N}}(v)$$

Double-reversal Method

[Gutiérrez et. al, MFCS 2019]

$$
\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] \arrow[rr, bend left, dotted, "\mathsf{Min}^{r}"] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[r, "\mathsf{Det}^{r}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{r}(\mathsf{Det}^{\ell}(\mathcal{N})) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^{R} \arrow[r, "\mathsf{Det}^{r}"] \arrow[rr, bend right, dotted, "\mathsf{Min}^{\ell}"'] & \mathsf{Det}^{r}(\mathcal{N}^{R}) \arrow[r, "\mathsf{Det}^{\ell}"] & \mathsf{Det}^{\ell}(\mathsf{Det}^{r}(\mathcal{N}^{R}))
\end{tikzcd}
$$

<u>**Thm:**</u> **[Brzozowski, 1962]** Let $\mathcal{N}$ be an NFA. Then $\mathsf{Det}^{r}((\mathsf{Det}^{r}(\mathcal{N}^{R}))^{R})$ is isomorphic to the minimal DFA for $\mathcal{L}(\mathcal{N})$

Double-reversal Method

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\mathcal{N} \arrow[r, "\mathsf{Det}^\ell"] \arrow[d, leftrightarrow, "R"'] \arrow[rr, bend left, dotted, "\mathsf{Min}^r"] & \color{red}{\mathsf{Det}^\ell(\mathcal{N})} \arrow[r, "\mathsf{Det}^r"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^r(\mathsf{Det}^\ell(\mathcal{N})) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^R \arrow[r, "\mathsf{Det}^r"] \arrow[rr, bend right, dotted, "\mathsf{Min}^\ell"'] & \mathsf{Det}^r(\mathcal{N}^R) \arrow[r, "\mathsf{Det}^\ell"] & \mathsf{Det}^\ell(\mathsf{Det}^r(\mathcal{N}^R))
\end{tikzcd}
$$

Lemma:

True if $\mathcal{N}$ is a co-DFA

$$\sim^r_L = \sim^r_{\mathcal{N}} \quad \text{iff } u^{-1}L = v^{-1}L \Leftrightarrow \mathrm{post}_{\mathcal{N}}(u) = \mathrm{post}_{\mathcal{N}}(v)$$

Double-reversal Method

$$
\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] \arrow[rr, bend left, dotted, "\mathsf{Min}^{r}"] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[r, "\mathsf{Det}^{r}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{r}(\mathsf{Det}^{\ell}(\mathcal{N})) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^R \arrow[r, "\mathsf{Det}^{r}"] \arrow[rr, bend right, dotted, "\mathsf{Min}^{\ell}"'] & \mathsf{Det}^{r}(\mathcal{N}^R) \arrow[r, "\mathsf{Det}^{\ell}"] & \mathsf{Det}^{\ell}(\mathsf{Det}^{r}(\mathcal{N}^R))
\end{tikzcd}
$$

$$\sim^{\ell}_{L} \quad \succeq \quad ? \quad \succeq \quad \sim^{\ell}_{\mathcal{N}}$$

Language-based　　　　Automata-based

Double-reversal Method

$$
\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[rr, dotted, bend left, "\mathsf{Min}^{r}"] \arrow[d, leftrightarrow, "R"'] & \color{red}{\mathsf{Det}^{\ell}(\mathcal{N})} \arrow[r, "\mathsf{Det}^{r}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{r}(\mathsf{Det}^{\ell}(\mathcal{N})) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^{R} \arrow[r, "\mathsf{Det}^{r}"] \arrow[rr, dotted, bend right, "\mathsf{Min}^{\ell}"] & \mathsf{Det}^{r}(\mathcal{N}^{R}) \arrow[r, "\mathsf{Det}^{\ell}"] & \mathsf{Det}^{\ell}(\mathsf{Det}^{r}(\mathcal{N}^{R}))
\end{tikzcd}
$$

$$\sim^{\ell}_{L} \quad \succeq \quad \sim^{\ell}_{\leftarrow} \quad \succeq \quad \color{red}{\sim^{\ell}_{\mathcal{N}}}$$

Language-based Simulation-based Automata-based

Double-reversal Method

$$
\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] \arrow[rr, bend left, dotted, "\mathsf{Min}^{r}"] & \color{red}{\mathsf{Sim}^{\ell}(\mathcal{N}, \leftarrow)} \arrow[r, "\mathsf{Det}^{r}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{r}(\mathsf{Det}^{\ell}(\mathcal{N})) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^{R} \arrow[r, "\mathsf{Det}^{r}"] \arrow[rr, bend right, dotted, "\mathsf{Min}^{\ell}"'] & \mathsf{Det}^{r}(\mathcal{N}^{R}) \arrow[r, "\mathsf{Det}^{\ell}"] & \mathsf{Det}^{\ell}(\mathsf{Det}^{r}(\mathcal{N}^{R}))
\end{tikzcd}
$$

$$\sim^{\ell}_{L} \quad \succeq \quad \sim^{\ell}_{\leftarrow} \quad \succeq \quad \color{red}{\sim^{\ell}_{\mathcal{N}}}$$

Language-based Simulation-based Automata-based

Contributions of This Work

[Gutiérrez et. al, MFCS 2019]

$\mathcal{N}$: NFA
L : language of $\mathcal{N}$

[Brzozowski and Tamm, 2014] **Generalization** of the Double-reversal Method	
[Brzozowski, 1962] Double-reversal Method	$\mathsf{Det}^r(\mathsf{Det}^\ell(\mathcal{N})) \equiv$ Minimal DFA for L
[Moore, 1956] Moore's algorithm	

Contributions of This Work

[Gutiérrez et. al, MFCS 2019]

$\mathcal{N}$: NFA
L : language of $\mathcal{N}$ L_q : *left* language of $\mathcal{N}$ w.r.t. q

[Brzozowski and Tamm, 2014] **Generalization** of the Double-reversal Method	$\mathsf{Det}^r(\mathcal{N}) \equiv$ Minimal DFA for L iff $\forall q: \; P_{\sim^r_L}(L_q) = L_q$
[Brzozowski, 1962] Double-reversal Method	$\mathsf{Det}^r(\mathsf{Det}^\ell(\mathcal{N})) \equiv$ Minimal DFA for L
[Moore, 1956] Moore's algorithm	

Contributions of This Work

[Gutiérrez et. al, MFCS 2019]

$\mathcal{N}$: NFA
L : language of $\mathcal{N}$ L_q : *left* language of $\mathcal{N}$ w.r.t. q

[Brzozowski and Tamm, 2014] **Generalization** of the Double-reversal Method	$\mathsf{Det}^r(\mathcal{N}) \equiv$ Minimal DFA for L iff $\forall q : \; P_{\sim^r_L}(L_q) = L_q$
[Brzozowski, 1962] Double-reversal Method	$\mathsf{Det}^r(\mathsf{Det}^\ell(\mathcal{N})) \equiv$ Minimal DFA for L
[Moore, 1956] Moore's algorithm	At each step n of Moore's partition refinement: $\forall q \in \mathcal{Q}^{(n)} : \; P^{(n)}_{\sim^r_L}(L_q) = L_q$

Conclusions

- **Left-right duality** between our automata-based congruences to explain **double-reversal method**

$$
\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[rr, bend left, dotted, "\mathsf{Min}^{r}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[r, "\mathsf{Det}^{r}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{r}(\mathsf{Det}^{\ell}(\mathcal{N})) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^{R} \arrow[r, "\mathsf{Det}^{r}"] \arrow[rr, bend right, dotted, "\mathsf{Min}^{\ell}"'] & \mathsf{Det}^{r}(\mathcal{N}^{R}) \arrow[r, "\mathsf{Det}^{\ell}"] & \mathsf{Det}^{\ell}(\mathsf{Det}^{r}(\mathcal{N}^{R}))
\end{tikzcd}
$$

- More **general** view on the method

- Congruences as **language abstractions**

$$P_{\sim_L^r}(L_q) = L_q \ , \forall q \in Q$$

Conclusions

- **Left-right duality** between our automata-based congruences to explain **double-reversal method**

$$\mathsf{Sim}^{\ell}(\mathcal{N}, \leftarrow)$$

$$\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] \arrow[rr, bend left, dotted, "\mathsf{Min}^{r}"] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[r, "\mathsf{Det}^{r}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{r}(\mathsf{Det}^{\ell}(\mathcal{N})) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^{R} \arrow[r, "\mathsf{Det}^{r}"] \arrow[rr, bend right, dotted, "\mathsf{Min}^{\ell}"'] & \mathsf{Det}^{r}(\mathcal{N}^{R}) \arrow[r, "\mathsf{Det}^{\ell}"] & \mathsf{Det}^{\ell}(\mathsf{Det}^{r}(\mathcal{N}^{R}))
\end{tikzcd}$$

- More **general** view on the method

- Congruences as **language abstractions**

$$P_{\sim^{r}_{L}}(L_q) = L_q \ \ , \forall q \in Q$$

Conclusions

- **Left-right duality** between our automata-based congruences to explain **double-reversal method**

$$
\begin{tikzcd}
\mathsf{Sim}^{\ell}(\mathcal{N}, \leftarrow) \arrow[dr, squiggly] & & \\
\mathcal{N} \arrow[r, "\mathsf{Det}^{\ell}"] \arrow[d, leftrightarrow, "R"'] \arrow[rr, bend left, dotted, "\mathsf{Min}^{r}"] & \mathsf{Det}^{\ell}(\mathcal{N}) \arrow[r, "\mathsf{Det}^{r}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Det}^{r}(\mathsf{Det}^{\ell}(\mathcal{N})) \arrow[d, leftrightarrow, "R"'] \\
\mathcal{N}^{R} \arrow[r, "\mathsf{Det}^{r}"] \arrow[rr, bend right, dotted, "\mathsf{Min}^{\ell}"'] & \mathsf{Det}^{r}(\mathcal{N}^{R}) \arrow[r, "\mathsf{Det}^{\ell}"] & \mathsf{Det}^{\ell}(\mathsf{Det}^{r}(\mathcal{N}^{R}))
\end{tikzcd}
$$

- More **general** view on the method

- Congruences as **language abstractions**

$$P_{\sim^r_L}(L_q) = L_q \quad , \forall q \in Q$$

Questions?

E. Gutiérrez, IMDEA Software, Madrid

Auxiliary Material

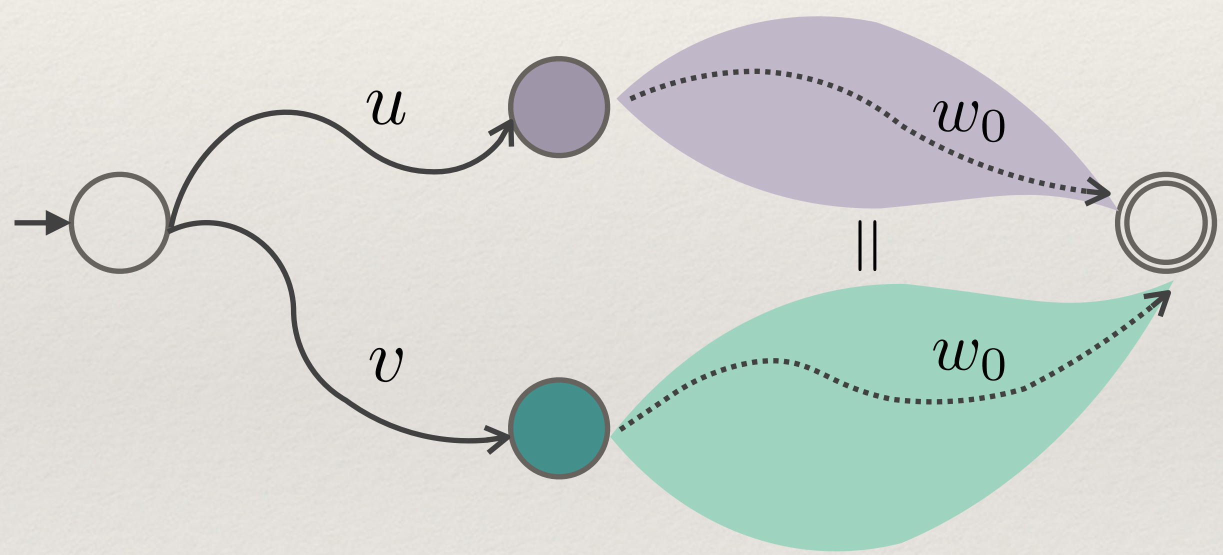
Theorem: Let $\mathcal{N}$ be a co-DFA. Then, for each $u, v \in \Sigma^*$:

$$u^{-1}L = v^{-1}L \iff \mathrm{post}_u^{\mathcal{N}}(I) = \mathrm{post}_v^{\mathcal{N}}(I)$$

<u>*Proof:*</u>

$\Leftarrow$) *Trivial (since* $\sim^r_{\mathcal{N}}$ *always finer than* $\sim^r_L$ *).*

$\Rightarrow$) *Assume* $\mathrm{post}_u^{\mathcal{N}}(I) \neq \mathrm{post}_v^{\mathcal{N}}(I)$. Then,



But  =  contradicts that $\mathcal{N}$ is a co-DFA!

There cannot be one word w0 that allows me to reach the final state from two different states in a co-DFA

Therefore, $\mathrm{post}_u^{\mathcal{N}}(I) = \mathrm{post}_v^{\mathcal{N}}(I)$