A Quasiorder-Based Perspective on Residual Automata

Pierre Ganty, Elena Gutiérrez, Pedro Valero

IMDEA Software Institute, Madrid, Spain

MFCS 2020

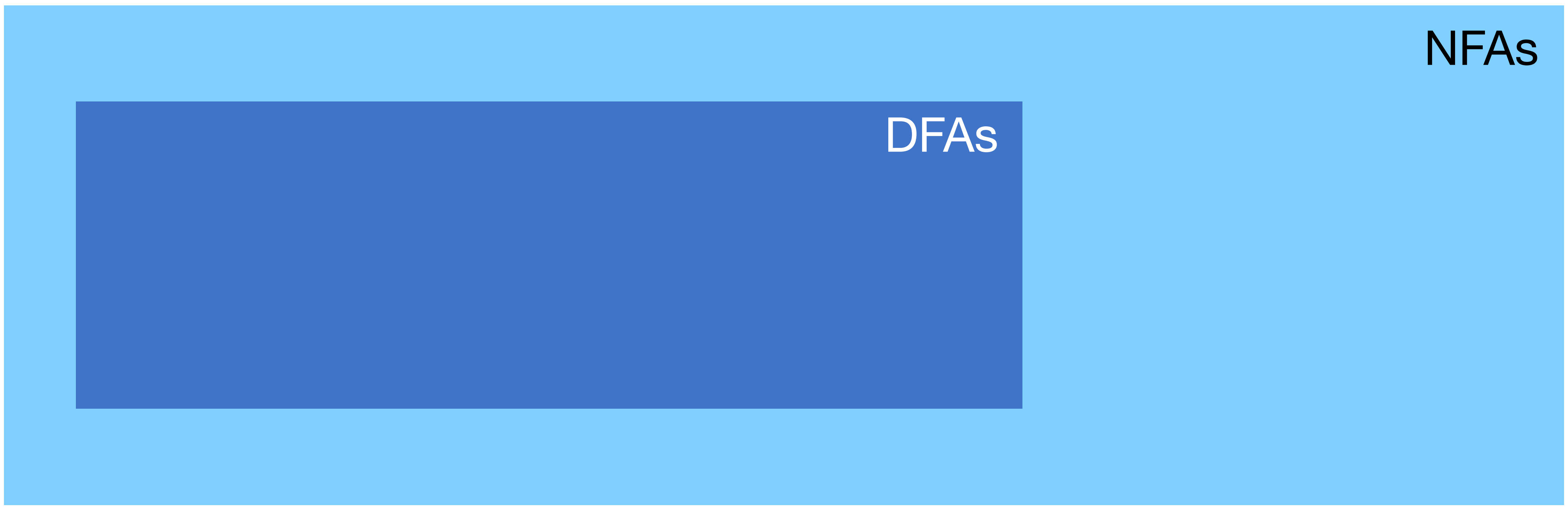
August 25th, 2020

Motivation

Finite representation of regular languages

NFA: **N**ondeterministic
DFA: **D**eterministic

finite-state **a**utomata

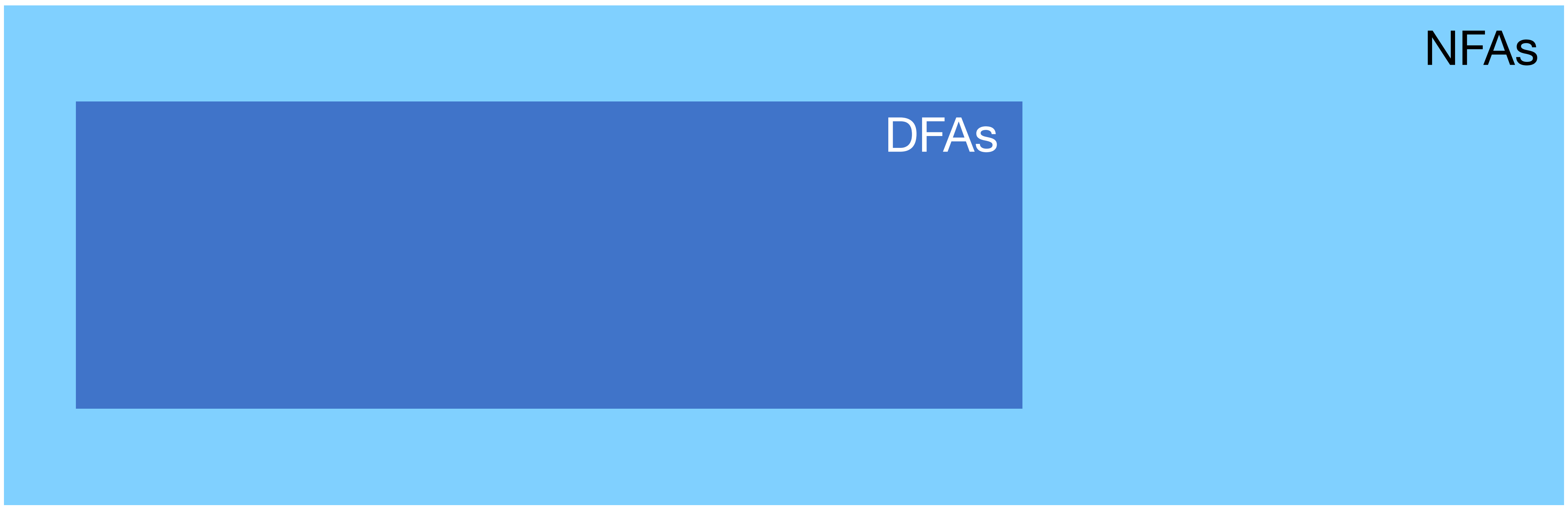


Motivation

Finite representation of regular languages

NFA: **N**ondeterministic
DFA: **D**eterministic

finite-state **a**utomata



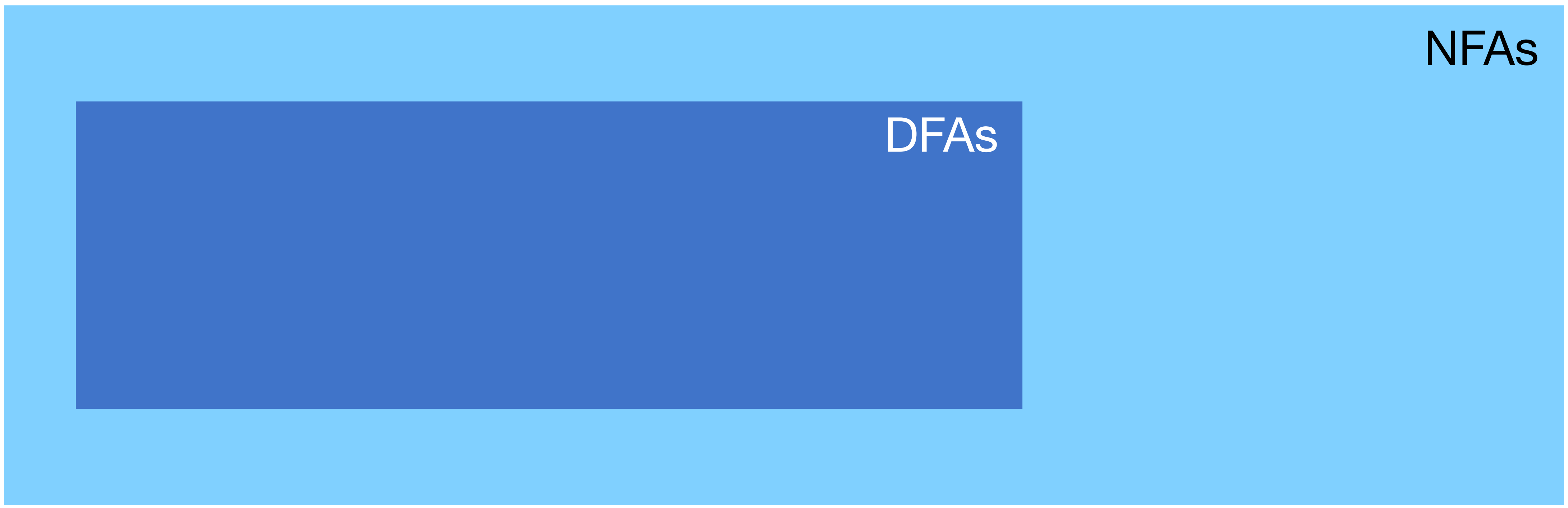
<u>DFAs</u>

- ✚ Polynomial-time decision problems
- ✚ States $\leftrightarrow$ *natural components* of the language
- ✚ *Minimal* DFA
- − Exponentially large in the worst case

Motivation

Finite representation of regular languages

NFA: **N**ondeterministic
DFA: **D**eterministic
finite-state **a**utomata



DFAs

- \+ Polynomial-time decision problems
- \+ States $\leftrightarrow$ *natural components* of the language
- \+ *Minimal* DFA
- − Exponentially large in the worst case

NFAs

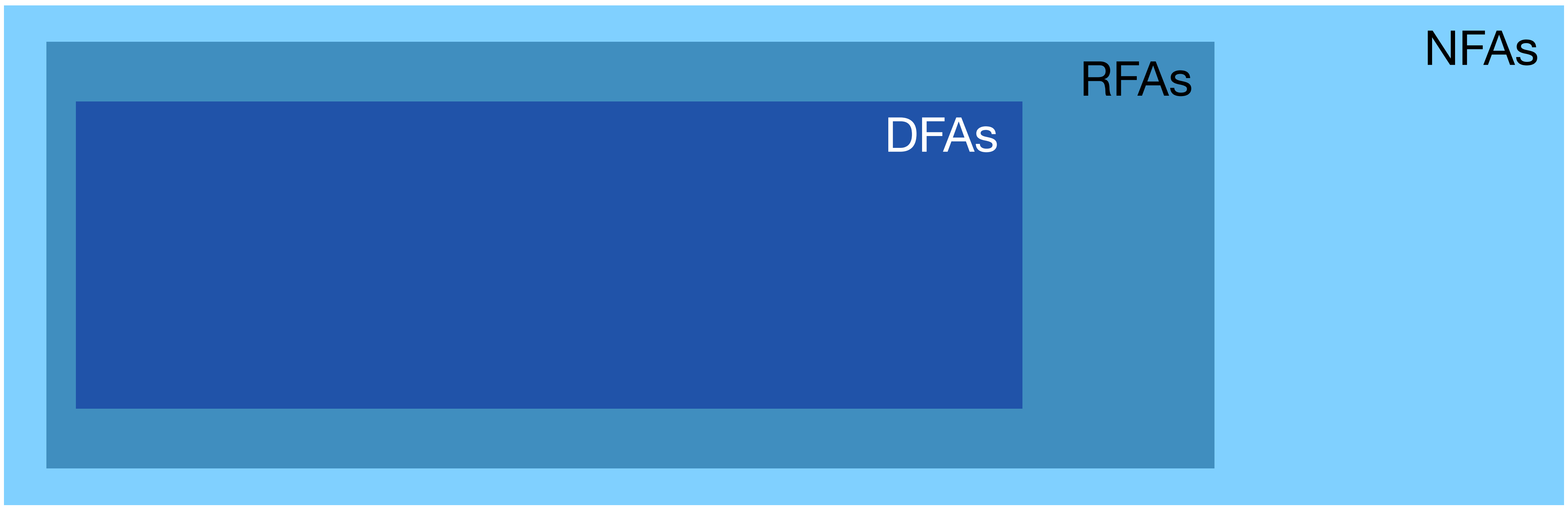
- − PSPACE-complete decision problems
- − States $\not\leftrightarrow$ *natural components*
- − No *minimal* NFA
- \+ Concise representation

Motivation

Finite representation of regular languages

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

finite-state **a**utomata



DFAs

- ➕ Polynomial-time decision problems
- ➕ States $\leftrightarrow$ *natural components* of the language
- ➕ Minimal form
- ➖ Exponentially large in the worst case

NFAs

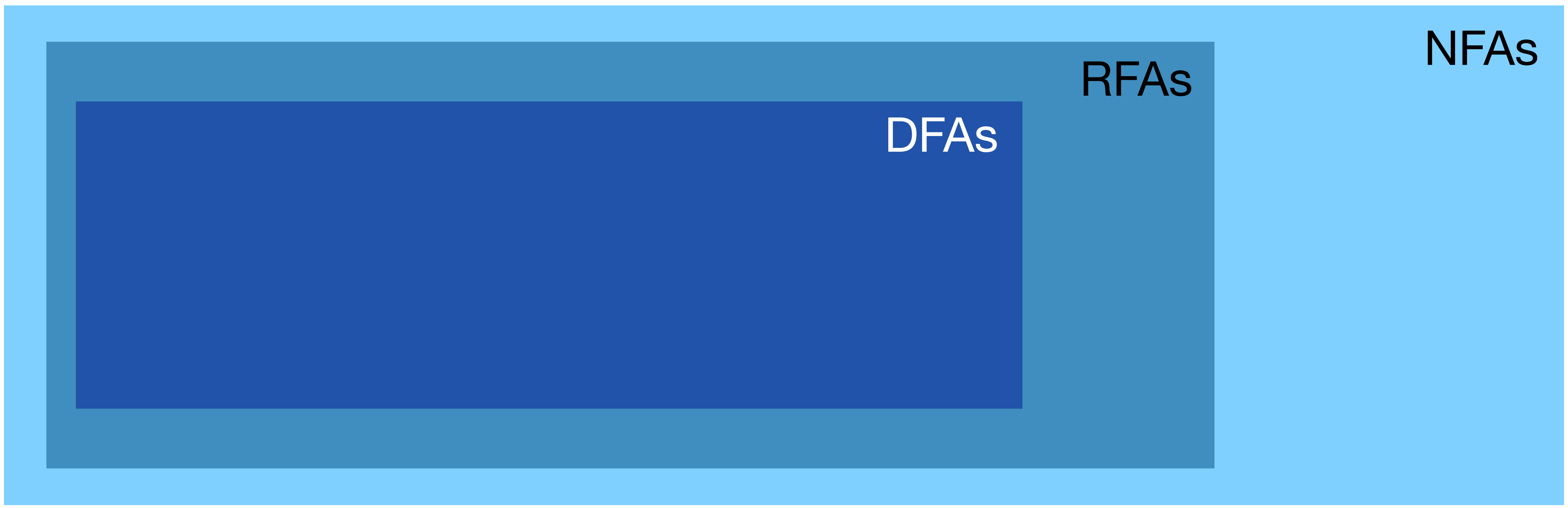
- ➖ PSPACE-complete decision problems
- ➖ States $\not\leftrightarrow$ *natural components*
- ➖ No minimal form
- ➕ Concise representation

Motivation

Finite representation of regular languages

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

finite-state **a**utomata



RFAs **[Denis et al., 2000]**

- States $\leftrightarrow$ *natural components* of the language
- Concise representation
- Existence of a **minimal** form

P. Valero, IMDEA Software, Madrid

Motivation

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

finite-state
automata

Minimal DFA

Canonical RFA

Motivation

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

finite-state
automata

Minimal DFA

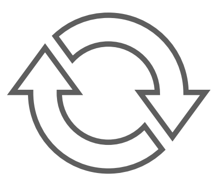
Double-reversal method

1.Reverse

2.Determinization

3.Reverse

4.Determinization



Canonical RFA

P. Valero, IMDEA Software, Madrid

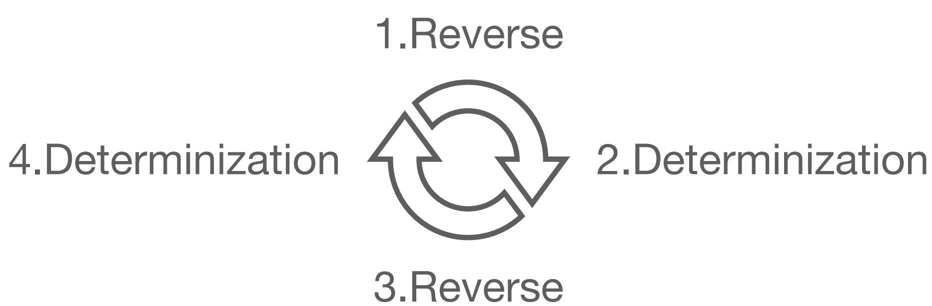
Motivation

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

finite-state **a**utomata

Minimal DFA

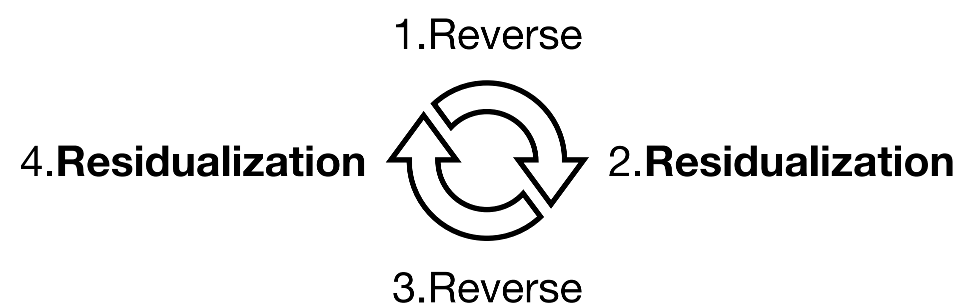
Double-reversal method



Canonical RFA

Double-reversal method

[Denis et al., 2002]



Motivation

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

finite-state
automata

NFAs

DFAs

RFAs

Minimal DFA

Double-reversal method

1.Reverse
2.Determinization
3.Reverse
4.Determinization

Canonical RFA

Double-reversal method

[Denis et al., 2002]

1.Reverse
2.**Residualization**
3.Reverse
4.**Residualization**

Goal

Give new language-theoretical insights on:

- the **residualization** operation, and
- the **double-reversal method** for the canonical RFA

Motivation

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

finite-state automata

NFAs

DFAs

RFAs

Minimal DFA

Double-reversal method

1.Reverse

2.Determinization

3.Reverse

4.Determinization

Canonical RFA

Double-reversal method

[Denis et al., 2002]

1.Reverse

2.**Residualization**

3.Reverse

4.**Residualization**

Goal

Give new language-theoretical insights on:

- the **residualization** operation, and
- the **double-reversal method** for the canonical RFA

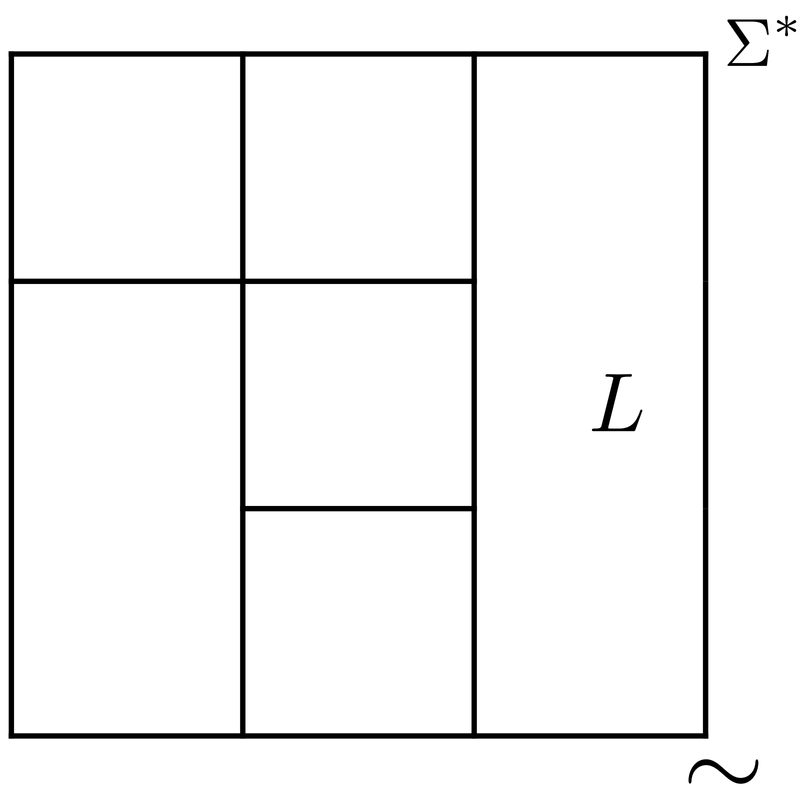
Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

In previous chapters…

[Valero et al., 2019]

DFA



- Finite-index **equivalence relations** on words

Today

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

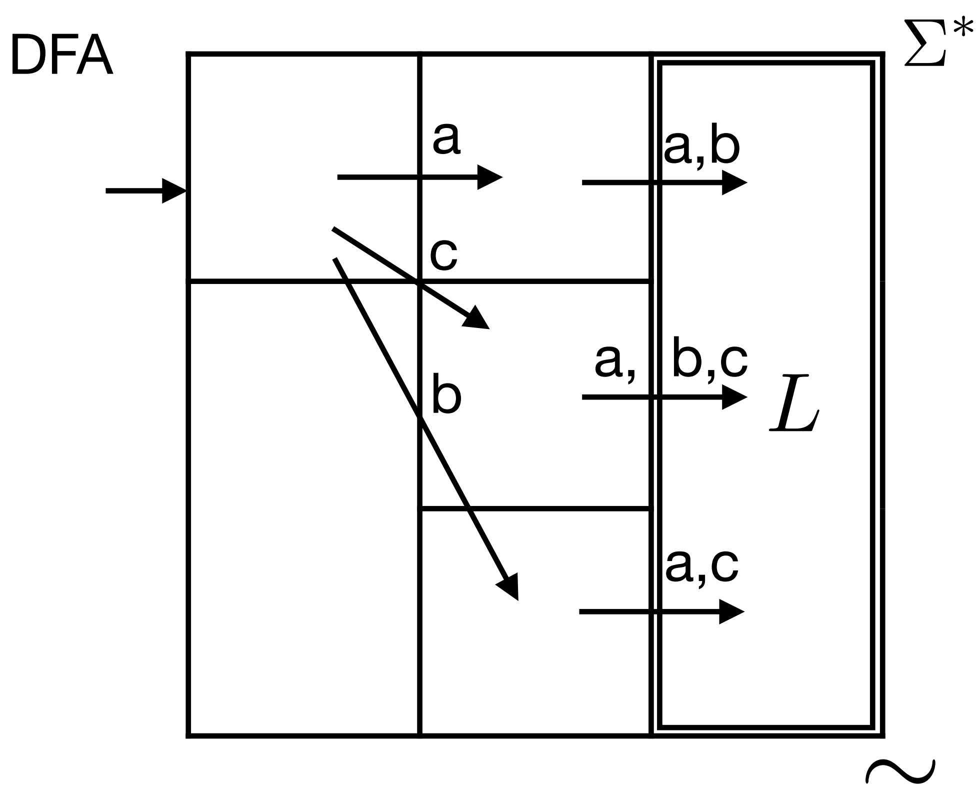
NFAs

DFAs

RFAs

In previous chapters...

[Valero et al., 2019]



- Finite-index **equivalence relations** on words

— DFA constructions

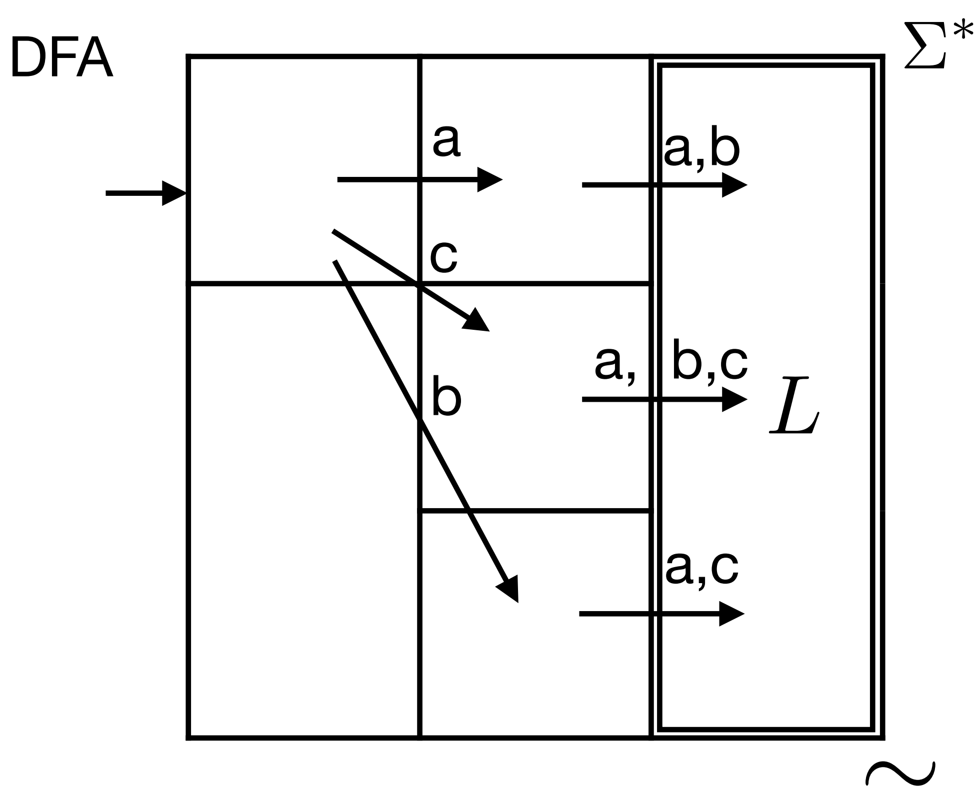
Today

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

In previous chapters…

[Valero et al., 2019]



- Finite-index **equivalence relations** on words
  - DFA constructions
  - New simple proof of double-reversal method

Today

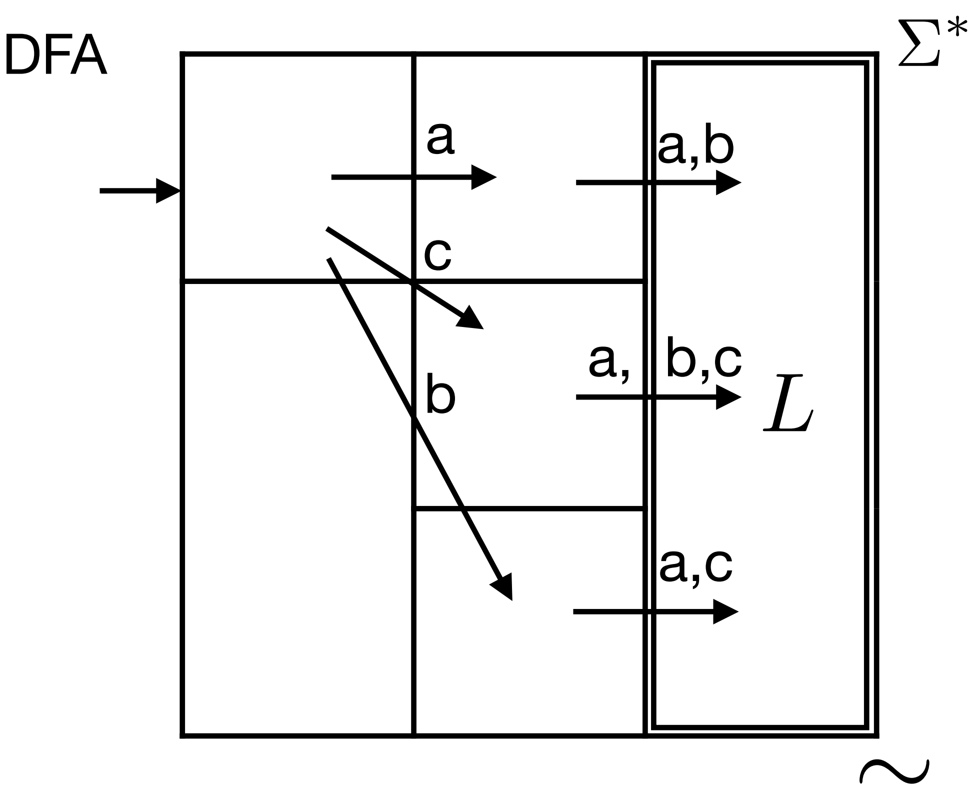
P. Valero, IMDEA Software, Madrid

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

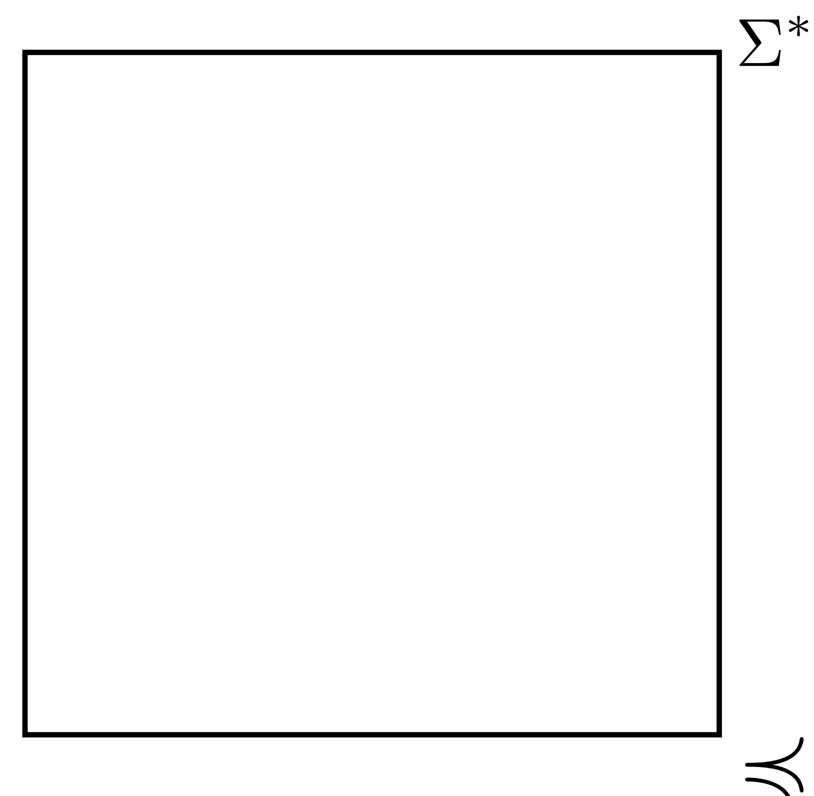
In previous chapters…

[Valero et al., 2019]



- Finite-index **equivalence relations** on words
  - DFA constructions
  - New simple proof of double-reversal method

Today



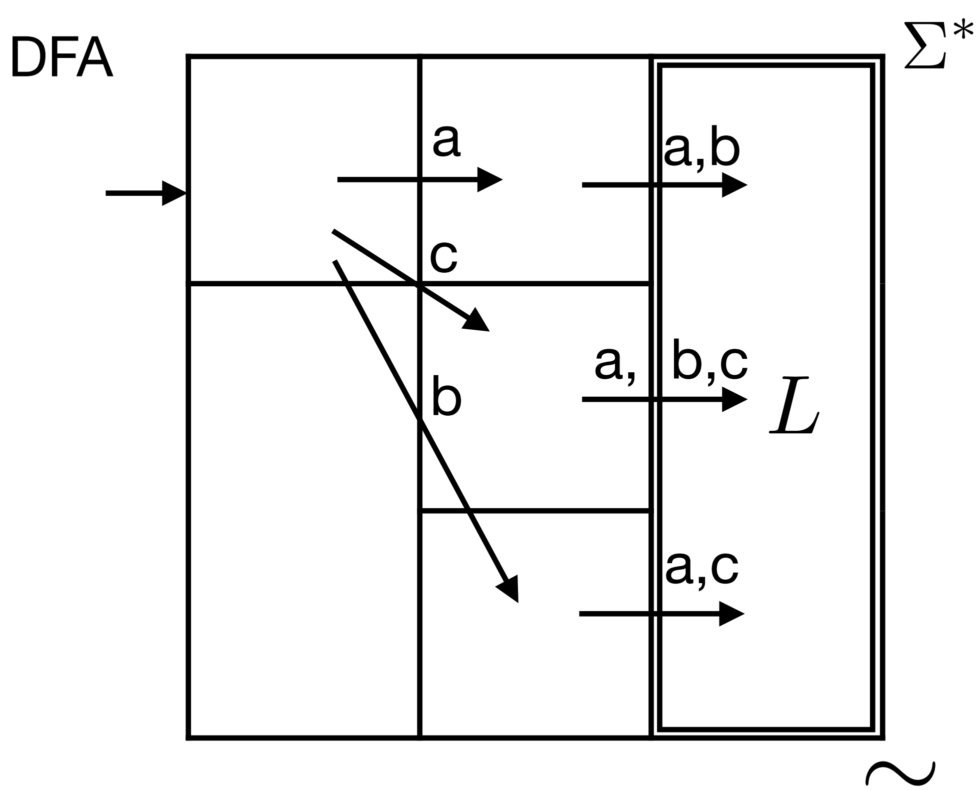
- Finite **quasiorders** on words

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

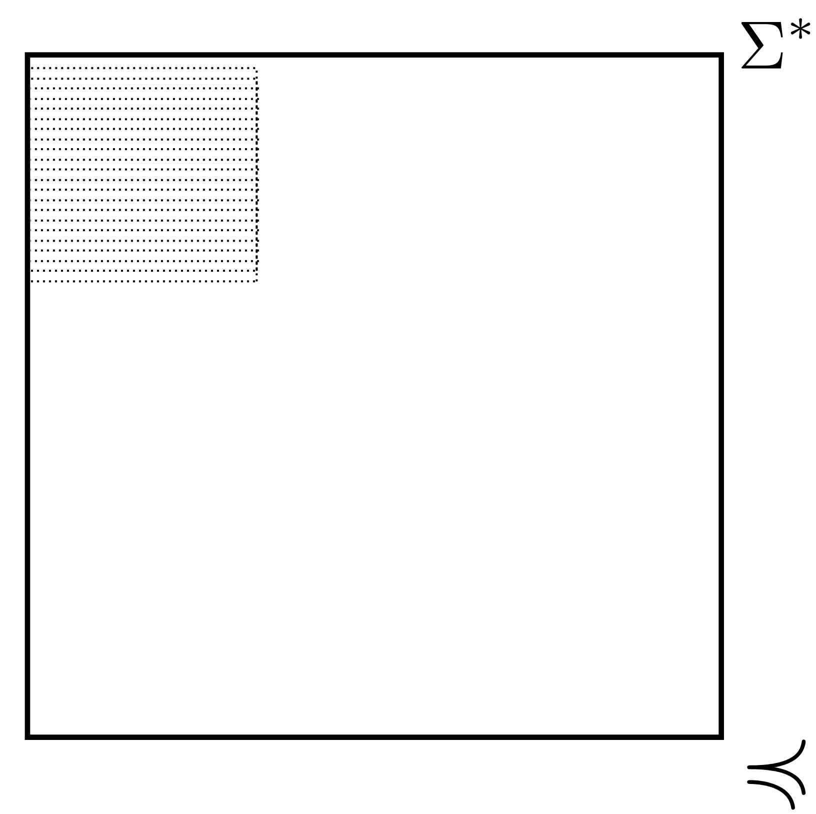
In previous chapters…

[Valero et al., 2019]



- Finite-index **equivalence relations** on words
  - DFA constructions
  - New simple proof of double-reversal method

Today



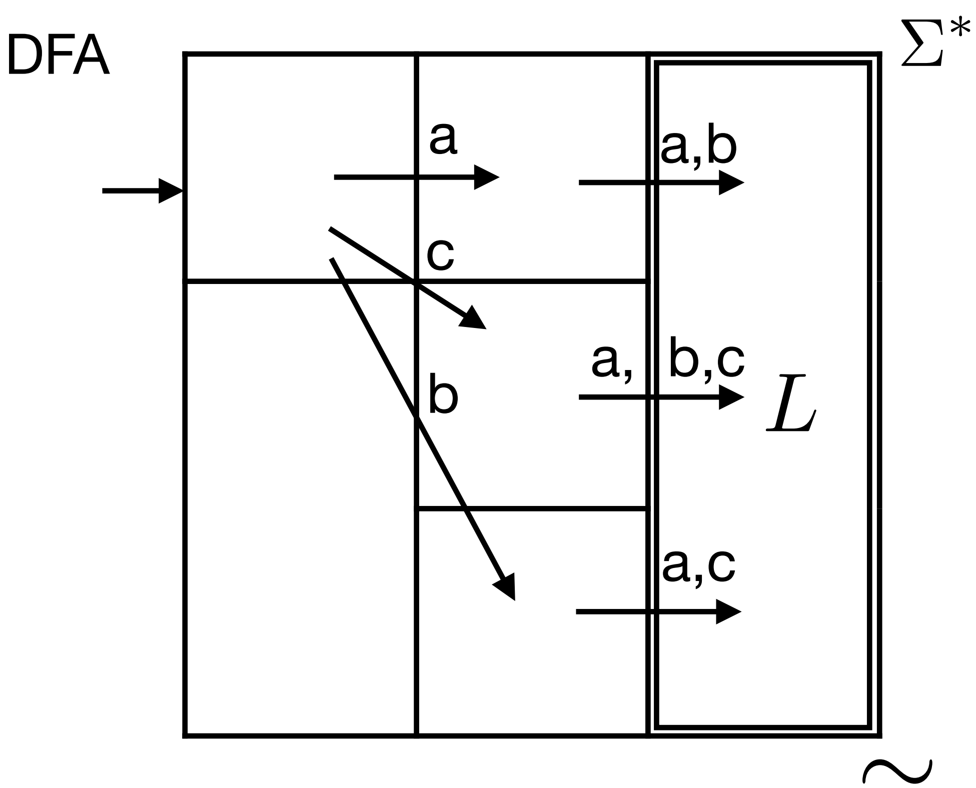
- Finite **quasiorders** on words

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

In previous chapters...

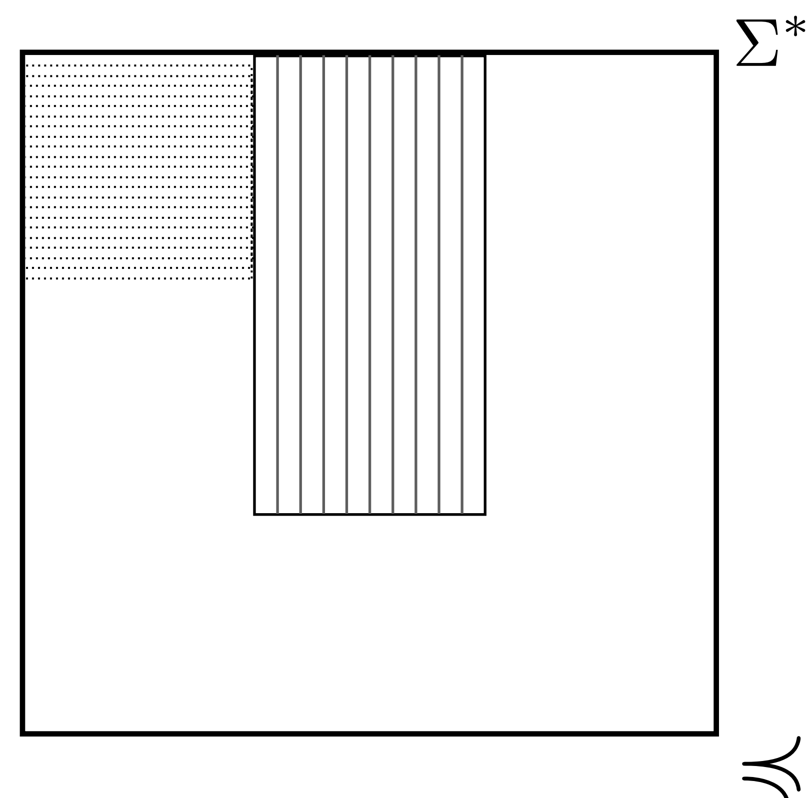
[Valero et al., 2019]



- Finite-index **equivalence relations** on words

  - DFA constructions
  - New simple proof of double-reversal method

Today



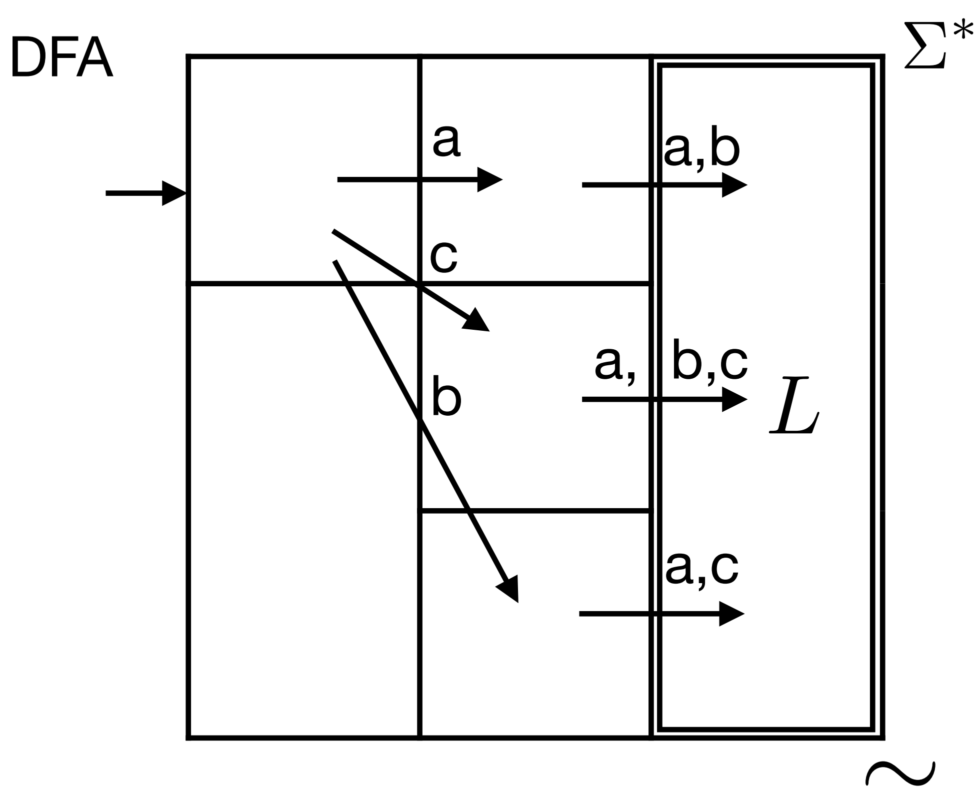
- Finite **quasiorders** on words

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

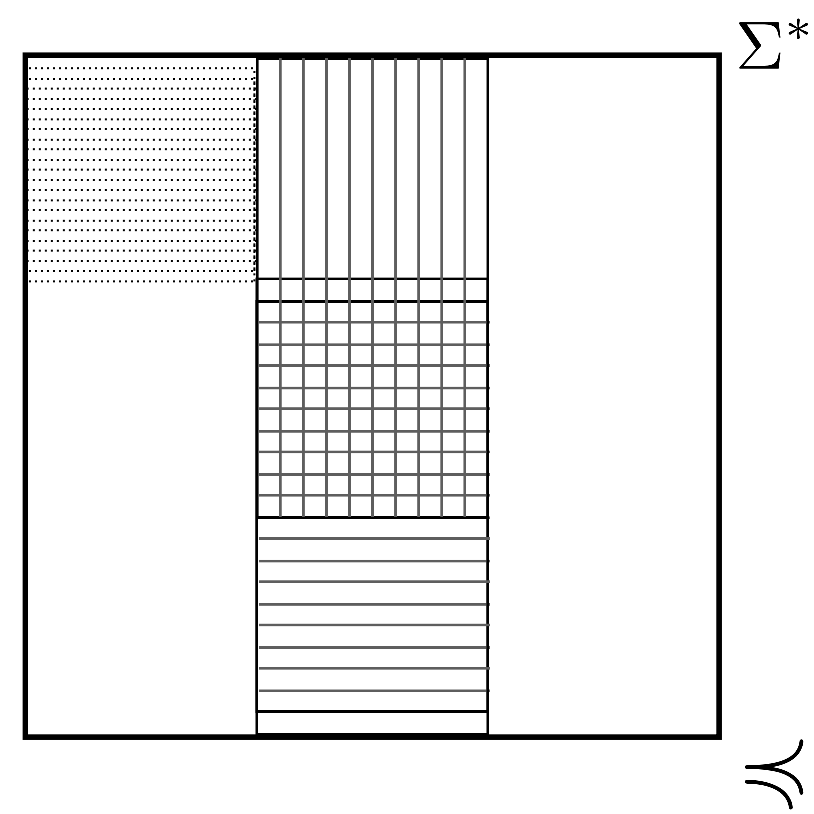
In previous chapters…

[Valero et al., 2019]



- Finite-index **equivalence relations** on words
  - DFA constructions
  - New simple proof of double-reversal method

Today



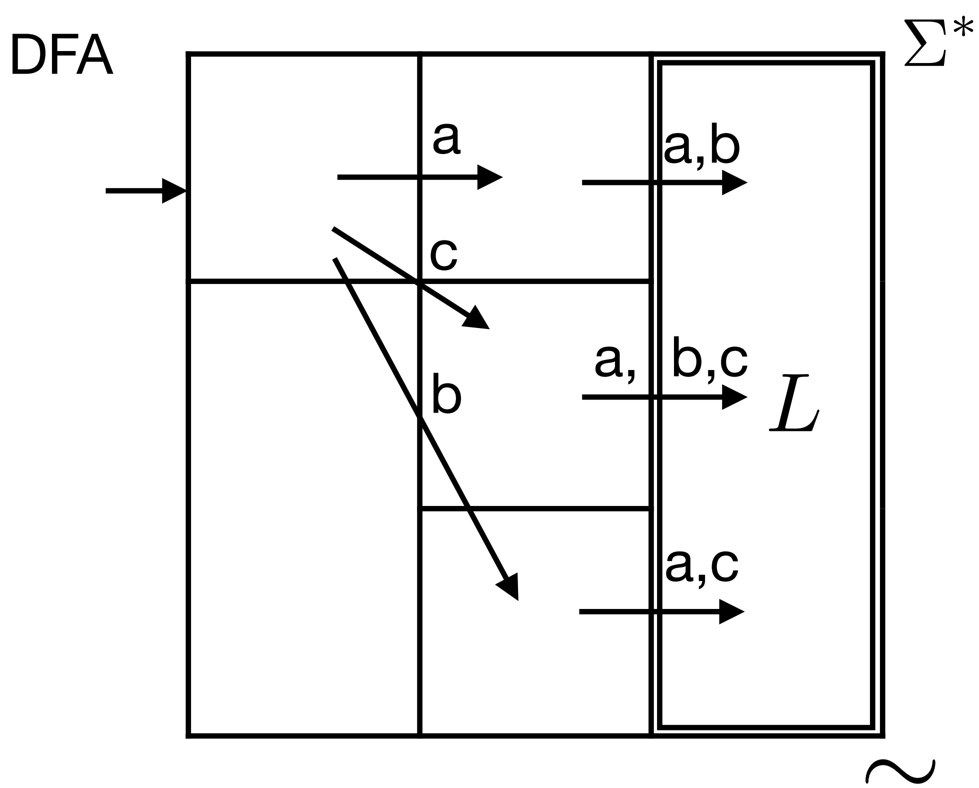
- Finite **quasiorders** on words

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

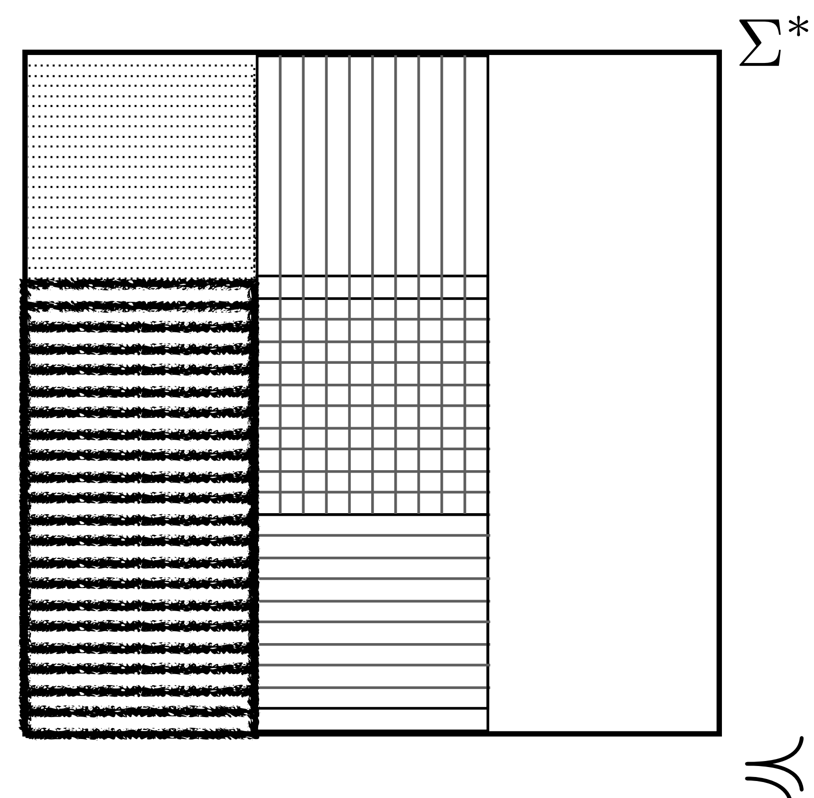
In previous chapters…

[Valero et al., 2019]



- Finite-index **equivalence relations** on words
  - DFA constructions
  - New simple proof of double-reversal method

Today



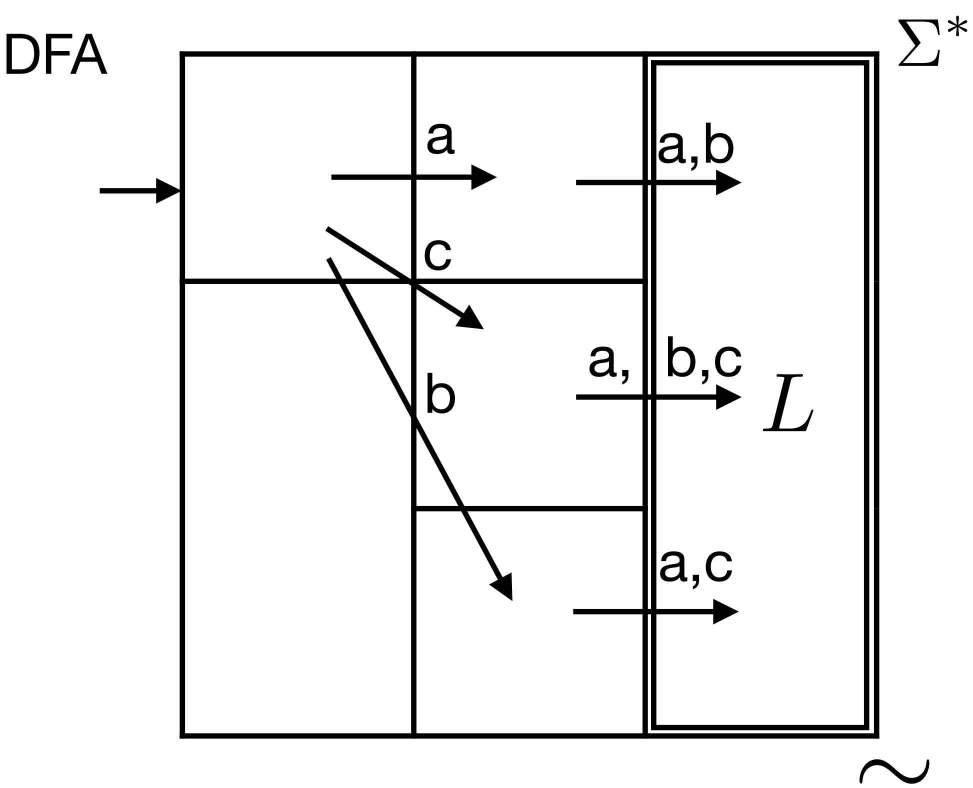
- Finite **quasiorders** on words

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

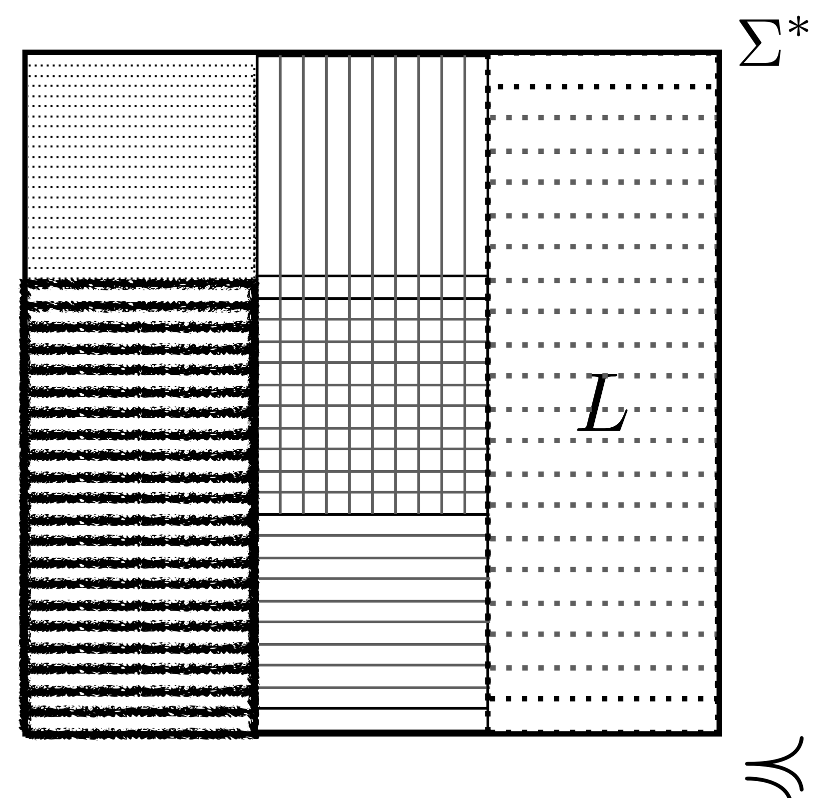
In previous chapters…

[Valero et al., 2019]



- Finite-index **equivalence relations** on words
  - DFA constructions
  - New simple proof of double-reversal method

Today



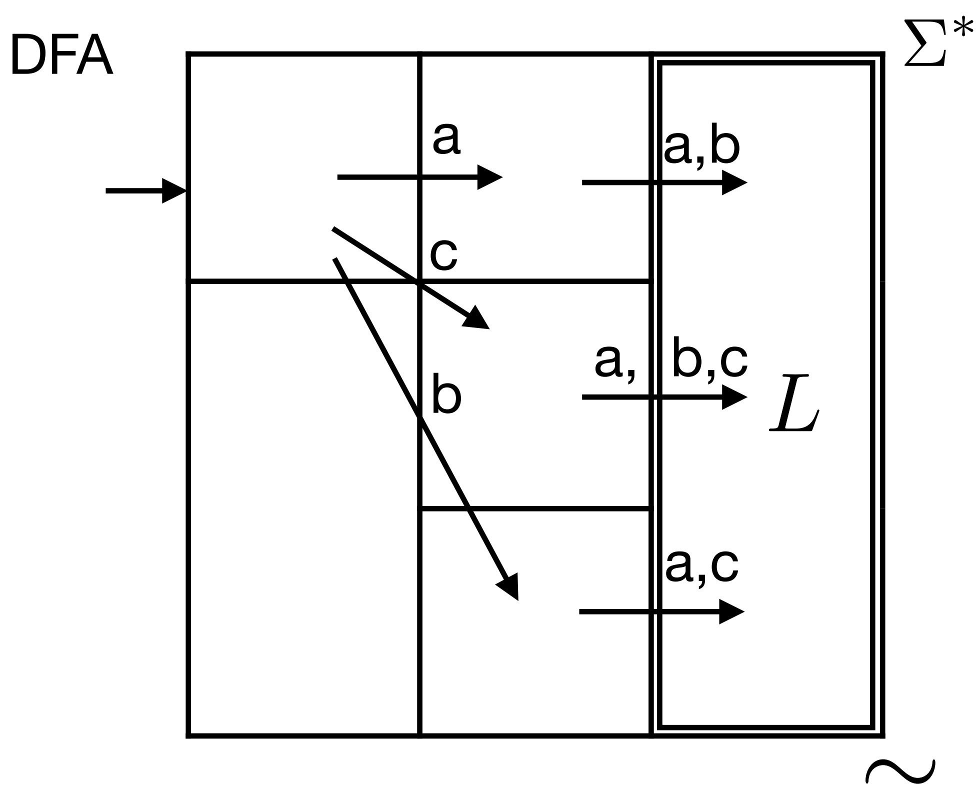
- Finite **quasiorders** on words

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

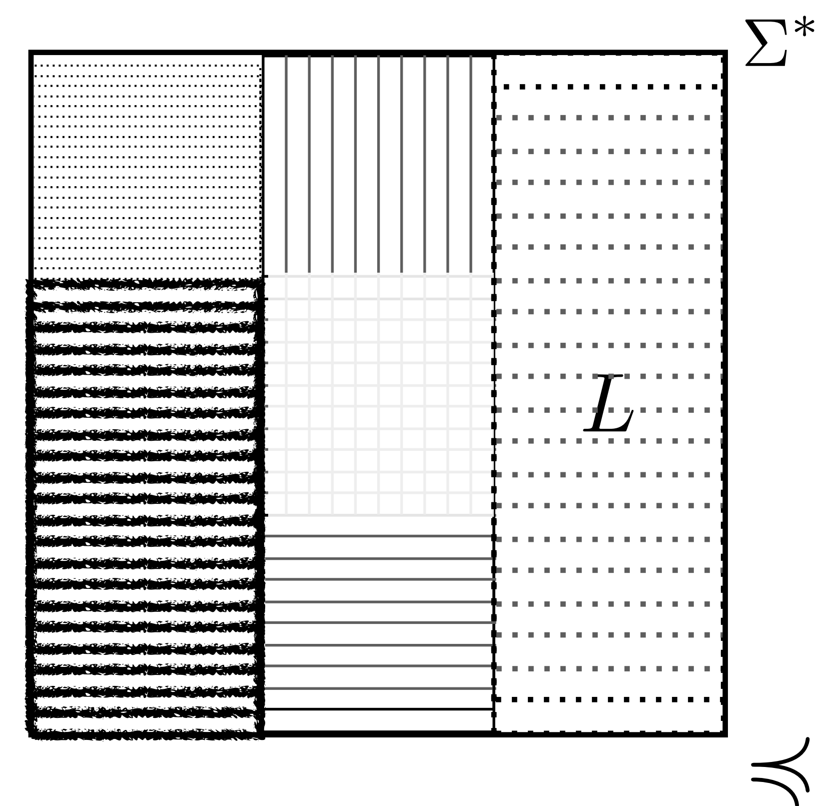
In previous chapters…

[Valero et al., 2019]



- Finite-index **equivalence relations** on words
  - DFA constructions
  - New simple proof of double-reversal method

Today



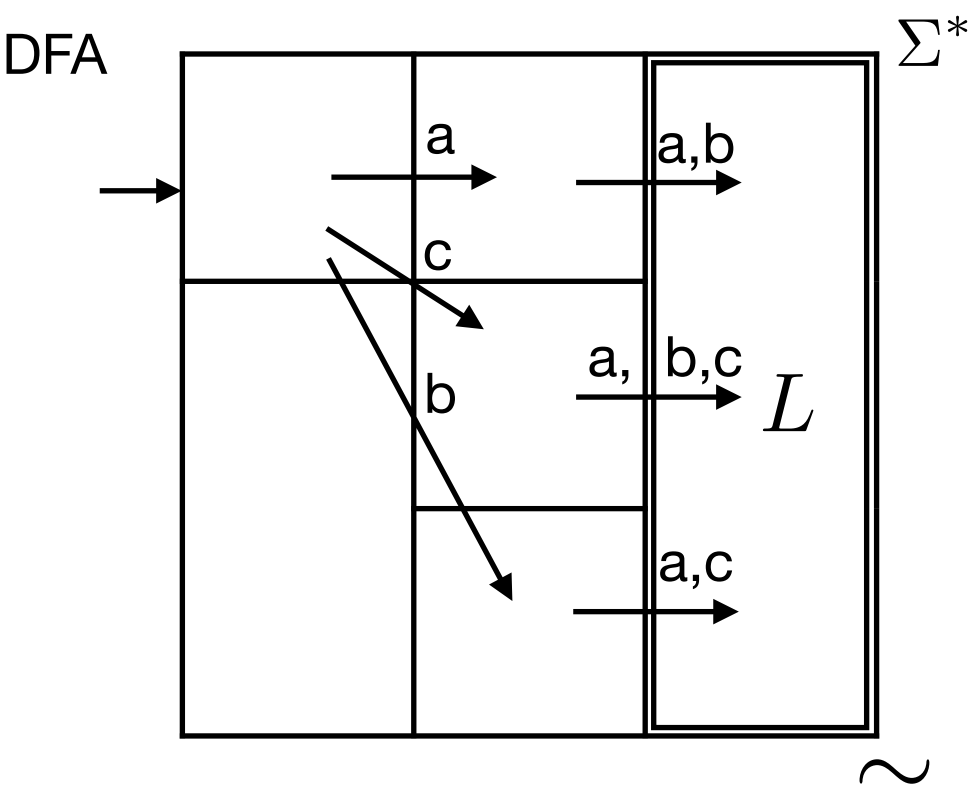
- Finite **quasiorders** on words

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

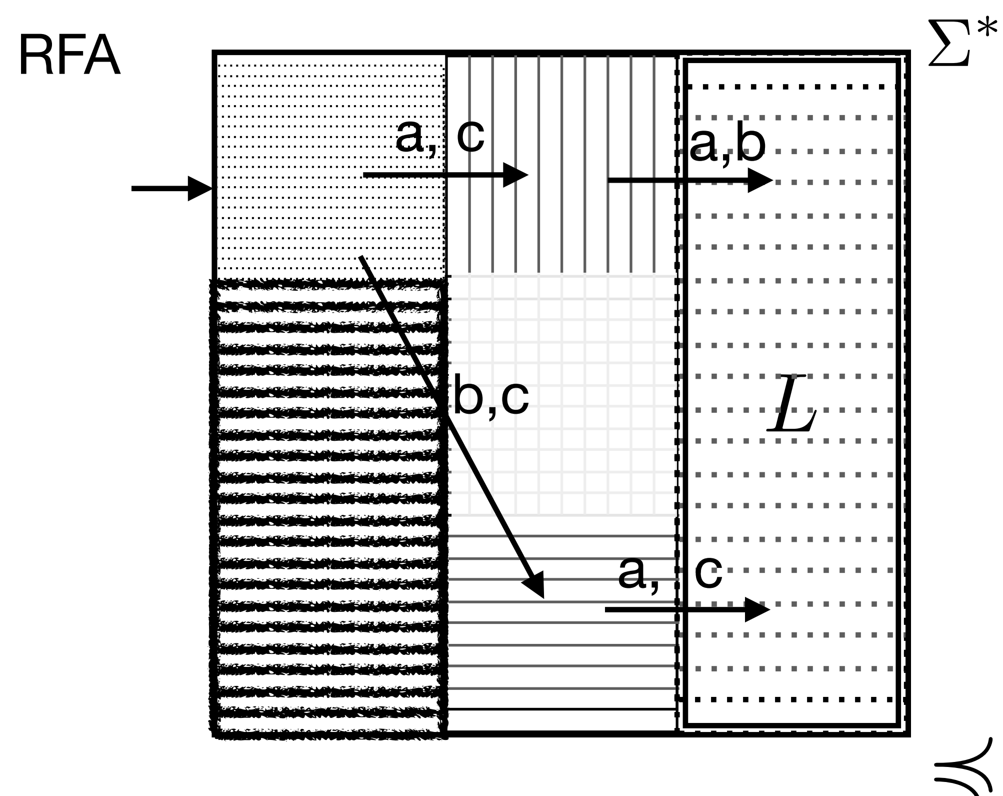
In previous chapters…

[Valero et al., 2019]



- Finite-index **equivalence relations** on words
  - DFA constructions
  - New simple proof of double-reversal method

Today



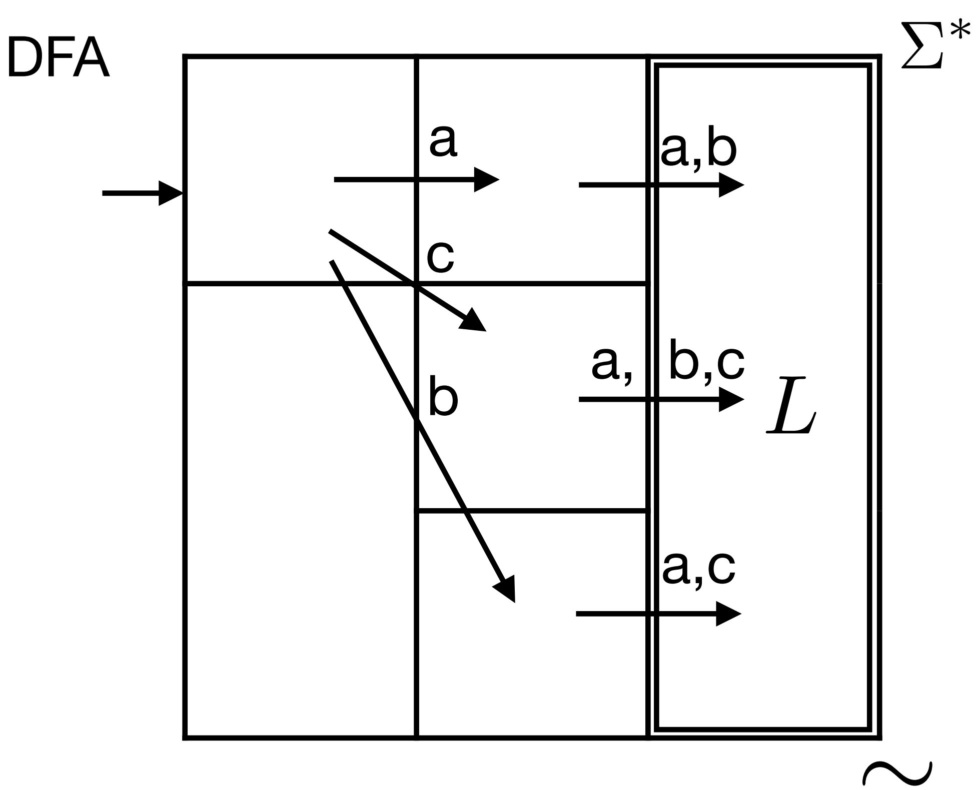
- Finite **quasiorders** on words

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

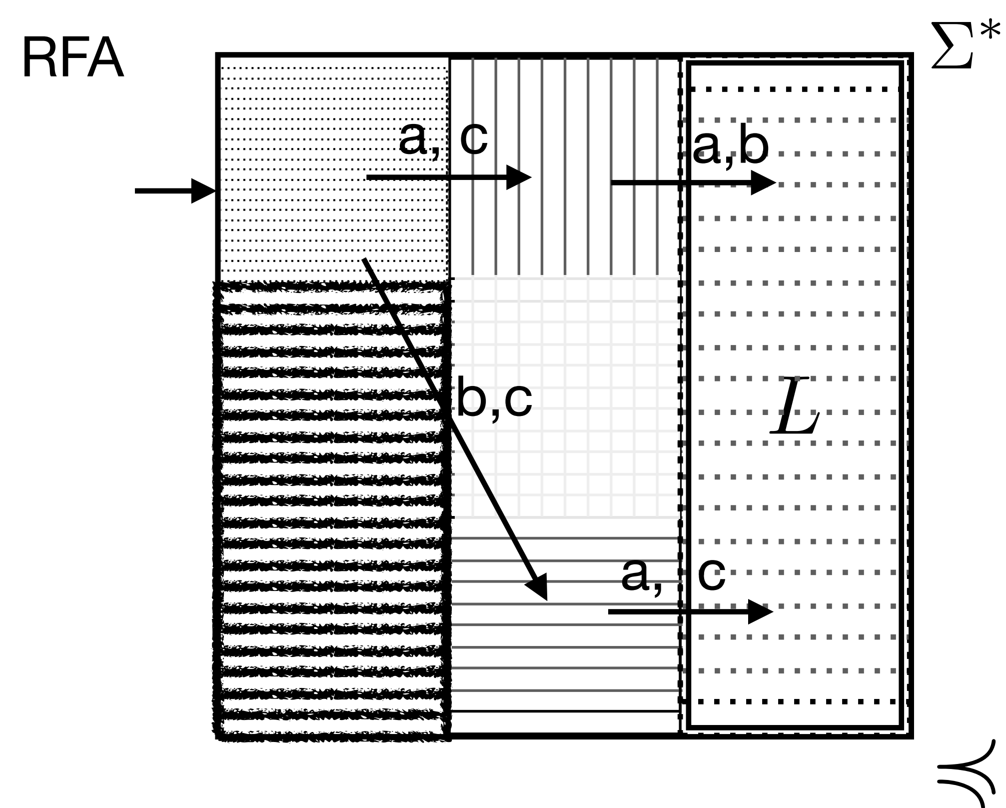
In previous chapters…

[Valero et al., 2019]



- Finite-index **equivalence relations** on words
  - DFA constructions
  - New simple proof of double-reversal method

Today



- Finite **quasiorders** on words

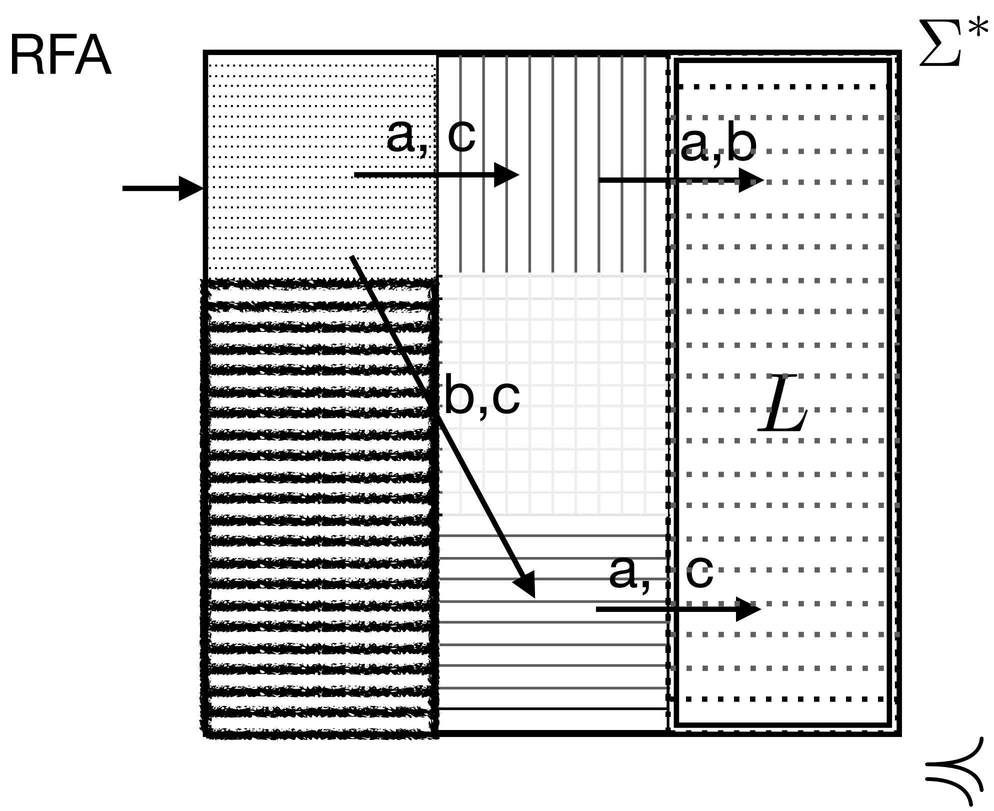
Uniform framework of RFA constructions

P. Valero, IMDEA Software, Madrid

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

Today



- Finite **quasiorders** on words

Uniform framework of RFA constructions

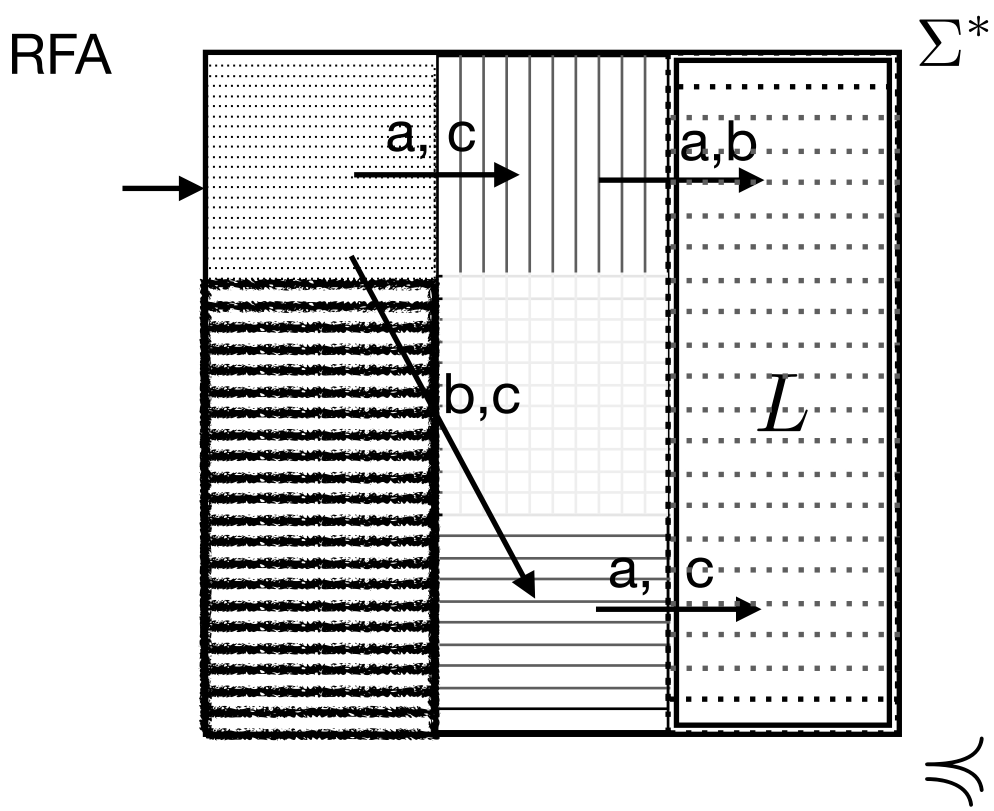
Contributions

- **RFA** constructions based on **quasiorders**
- New simple proof of **double-reversal** method for the **canonical RFA**
- Revisit **generalization** of the double-reversal method
- Quasiorder-based perspective on **NL***

Language-theoretical Perspective

NFA: **N**ondeterministic
DFA: **D**eterministic
RFA: **R**esidual

Today



- Finite **quasiorders** on words

Uniform framework of RFA constructions

Contributions

- **RFA** constructions based on **quasiorders**
- New simple proof of **double-reversal** method for the **canonical RFA**
- Revisit **generalization** of the double-reversal method
- Quasiorder-based perspective on **NL***

Preliminaries: Residual Automaton

Intuition: States $\leftrightarrow$ **natural components** of the language

Preliminaries: Residual Automaton

Intuition: States $\leftrightarrow$ **natural components** of the language

Definition: An NFA for which all states define **left quotients** of the language it accepts

$$||$$

residuals

Preliminaries: Residual Automaton

Intuition: States $\leftrightarrow$ **natural components** of the language

Definition: An NFA for which all states define **left quotients** of the language it accepts

||

residuals

u : word

L : language

$$u^{-1}L \stackrel{\text{def}}{=} \{w \mid uw \in L\}$$

Preliminaries: Residual Automaton

Intuition: States $\leftrightarrow$ **natural components** of the language

Definition: An NFA for which all states define **left quotients** of the language it accepts

||

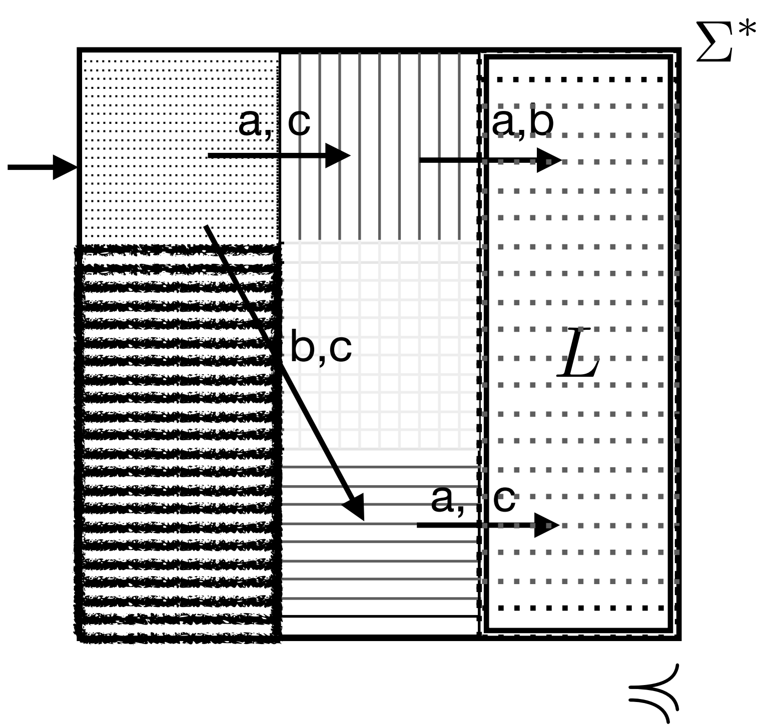
residuals

u : word

L : language

$$u^{-1}L \stackrel{\text{def}}{=} \{w \mid uw \in L\}$$

Example:



Preliminaries: Residual Automaton

Intuition: States $\leftrightarrow$ **natural components** of the language

Definition: An NFA for which all states define **left quotients** of the language it accepts

||

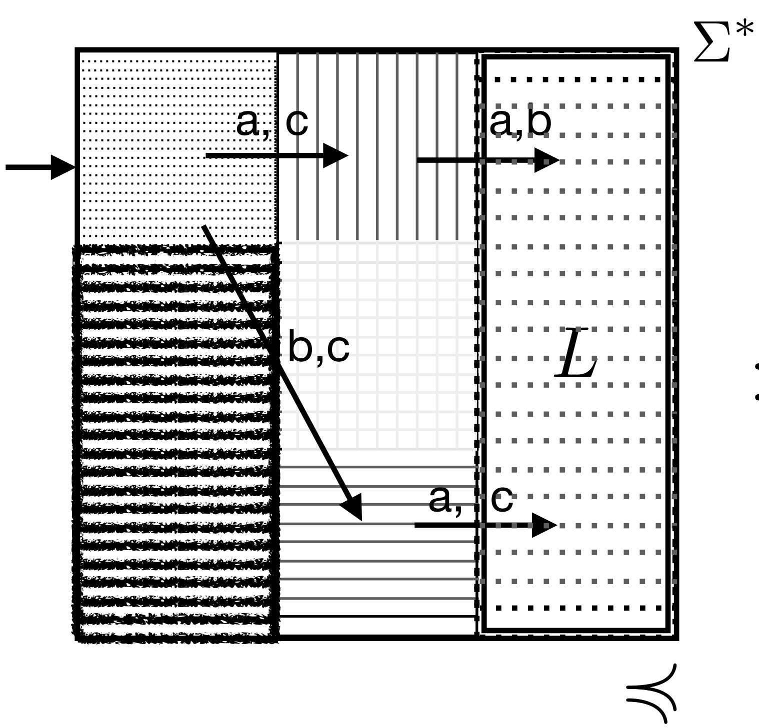
residuals

u : word

L : language

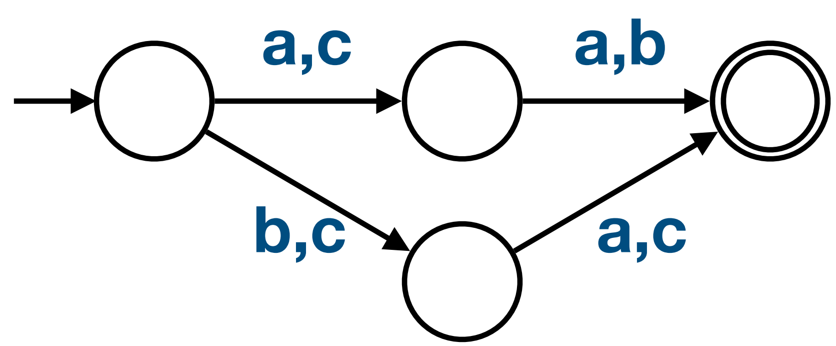
$$u^{-1}L \overset{\text{def}}{=} \{w \mid uw \in L\}$$

Example:



$$L = \{\textbf{aa, ab, ba, bc, ca, cb, cc}\}$$

$$=$$



Preliminaries: Residual Automaton

Intuition: States $\leftrightarrow$ **natural components** of the language

Definition: An NFA for which all states define **left quotients** of the language it accepts

||

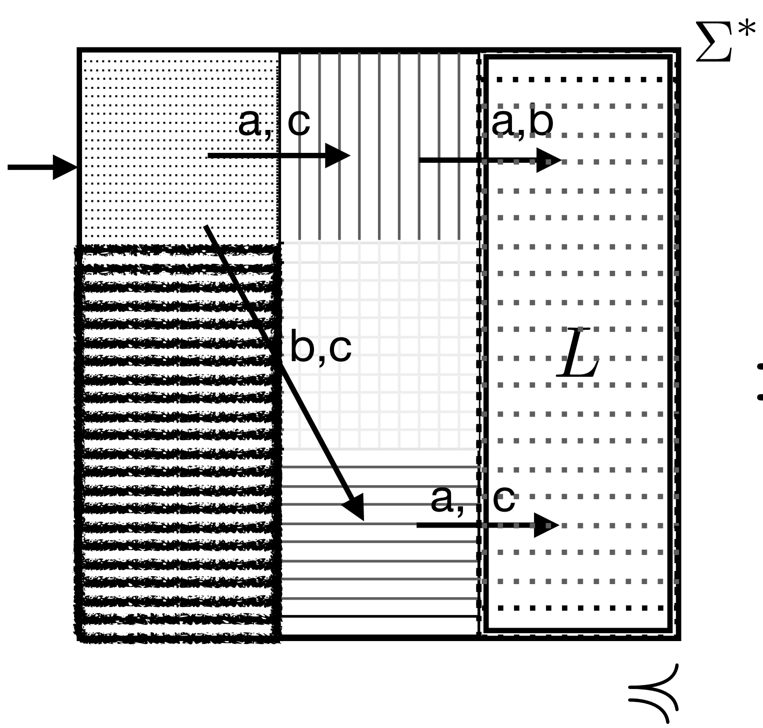
residuals

u : word

L : language

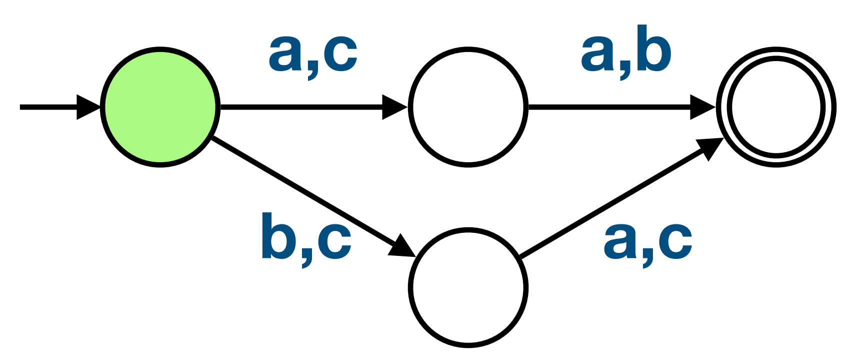
$$u^{-1}L \stackrel{\text{def}}{=} \{w \mid uw \in L\}$$

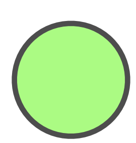
Example:



$$L = \{\textbf{aa, ab, ba, bc, ca, cb, cc}\}$$

$=$



 defines $L = \varepsilon^{-1}L$

Preliminaries: Residual Automaton

Intuition: States $\leftrightarrow$ **natural components** of the language

Definition: An NFA for which all states define **left quotients** of the language it accepts

||

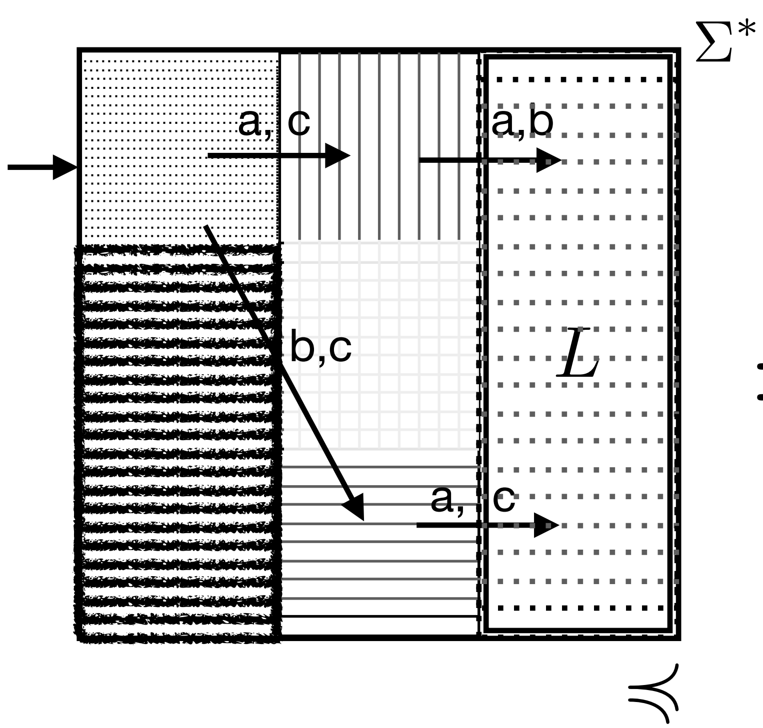
residuals

u : word

L : language

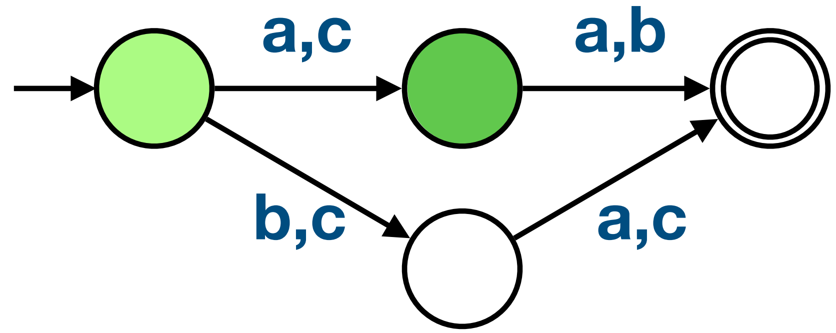
$$u^{-1}L \overset{\text{def}}{=} \{w \mid uw \in L\}$$

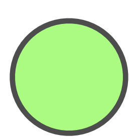
Example:

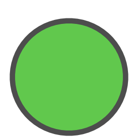


$$L = \{\textbf{aa, ab, ba, bc, ca, cb, cc}\}$$

$=$



 defines $L = \varepsilon^{-1}L$

 defines $\{\textbf{a,b}\} = \textbf{a}^{-1}L$

Preliminaries: Residual Automaton

Intuition: States $\leftrightarrow$ **natural components** of the language

Definition: An NFA for which all states define **left quotients** of the language it accepts

||

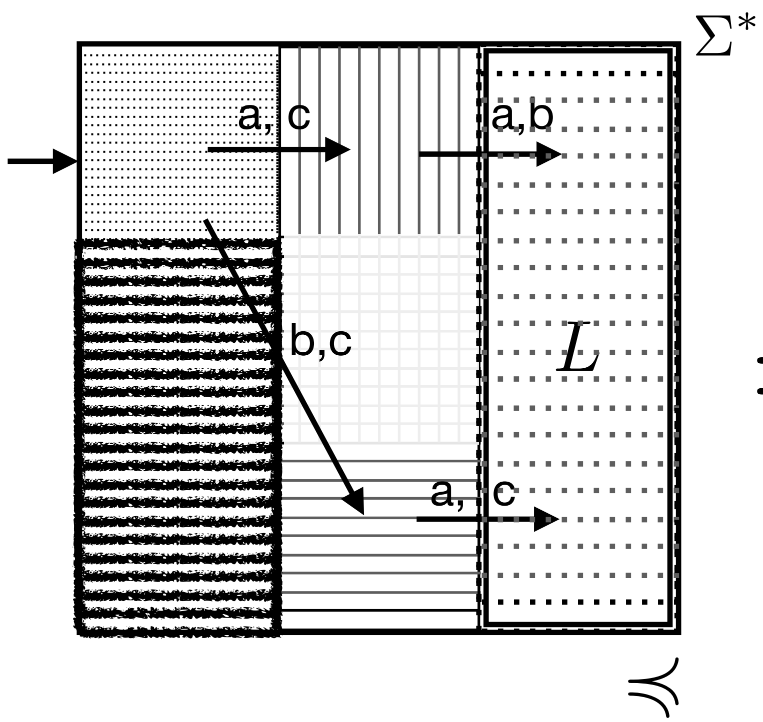
residuals

u : word

L : language

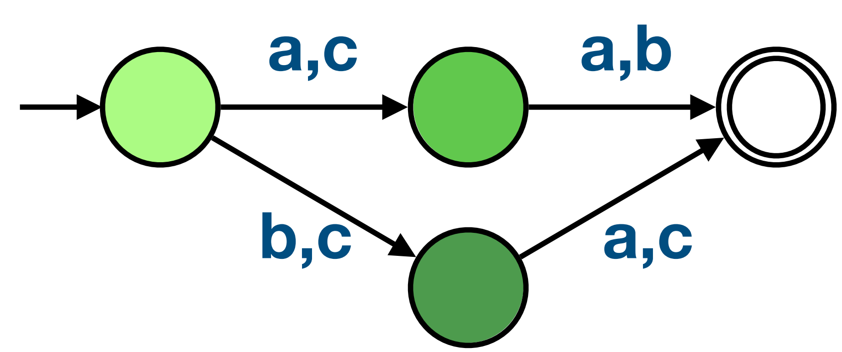
$$u^{-1}L \overset{\text{def}}{=} \{w \mid uw \in L\}$$

Example:



$$L = \{\textbf{aa, ab, ba, bc, ca, cb, cc}\}$$

$=$



defines $L = \varepsilon^{-1}L$

defines $\{\textbf{a,b}\} = \textbf{a}^{-1}L$

defines $\{\textbf{a,c}\} = \textbf{b}^{-1}L$

Preliminaries: Residual Automaton

Intuition: States $\leftrightarrow$ **natural components** of the language

Definition: An NFA for which all states define **left quotients** of the language it accepts

||

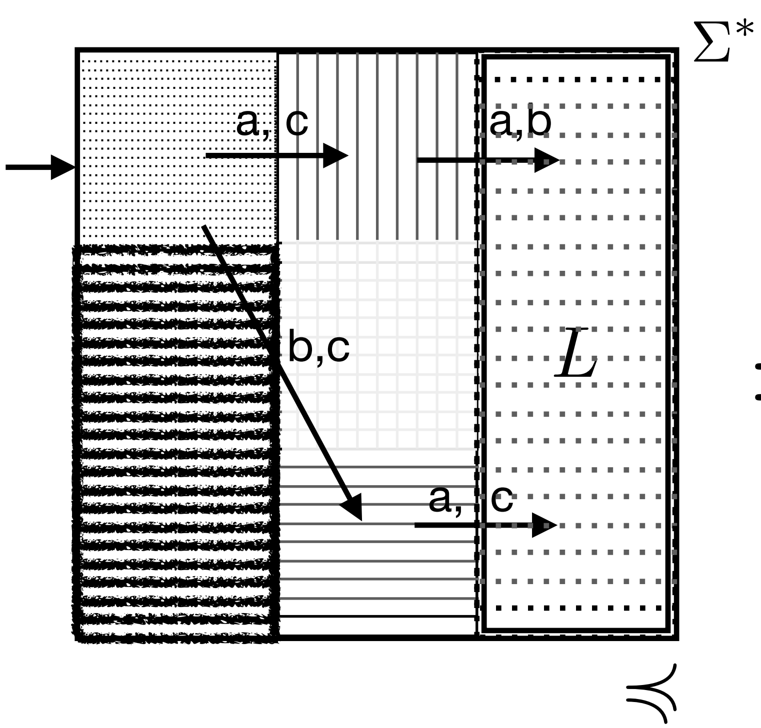
residuals

u : word

L : language

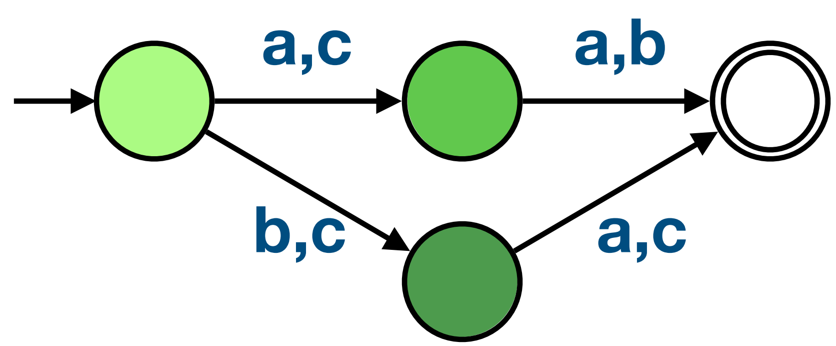
$$u^{-1}L \overset{\text{def}}{=} \{w \mid uw \in L\}$$

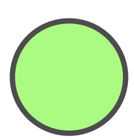
Example:

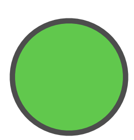


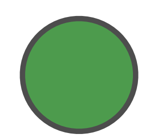
$$L = \{\textbf{aa, ab, ba, bc, ca, cb, cc}\}$$

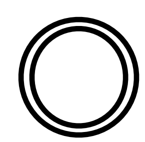
$=$



 defines $L = \varepsilon^{-1}L$

 defines $\{\textbf{a,b}\} = \textbf{a}^{-1}L$

 defines $\{\textbf{a,c}\} = \textbf{b}^{-1}L$

 defines $\{\varepsilon\} = \textbf{aa}^{-1}L$

Preliminaries: Residual Automaton

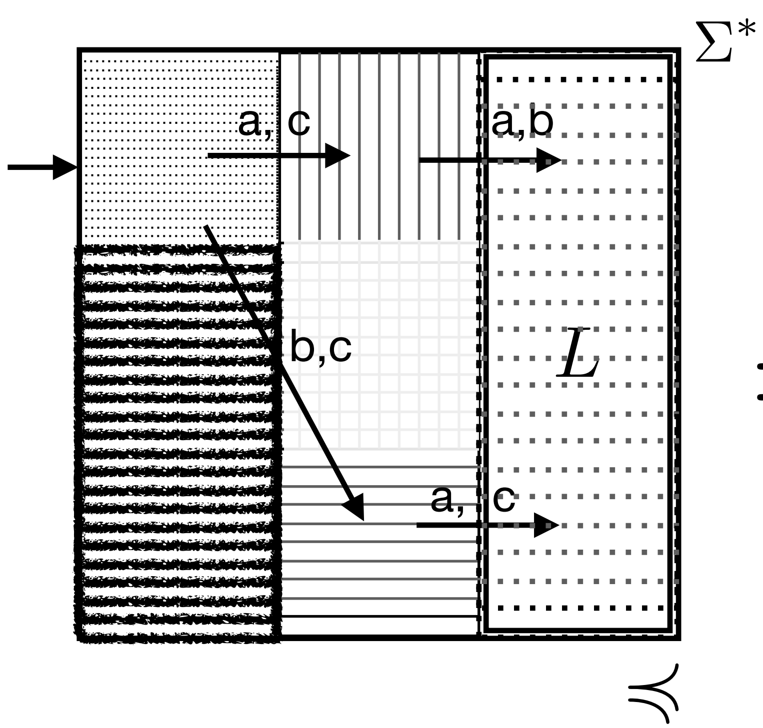
Intuition: States $\leftrightarrow$ **natural components** of the language

Definition: An NFA for which all states define **left quotients** of the language it accepts
||
residuals

u : word
L : language

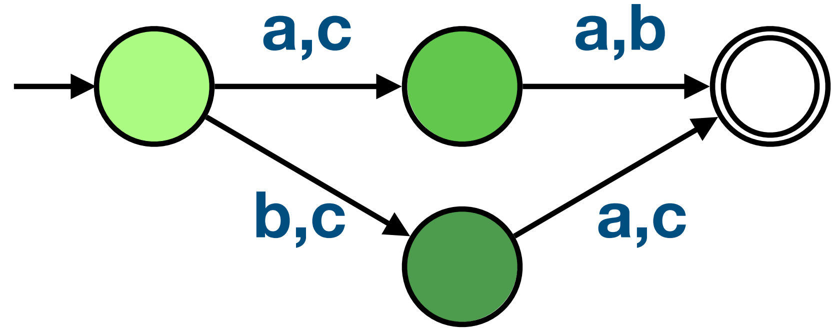
$$u^{-1}L \stackrel{\text{def}}{=} \{w \mid uw \in L\}$$

Example:

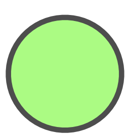


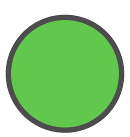
$$L = \{\text{aa, ab, ba, bc, ca, cb, cc}\}$$

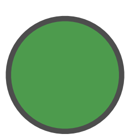
$=$

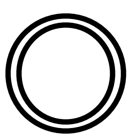


RFA 

 defines $L = \varepsilon^{-1}L$

 defines $\{\text{a,b}\} = \text{a}^{-1}L$

 defines $\{\text{a,c}\} = \text{b}^{-1}L$

 defines $\{\varepsilon\} = \text{aa}^{-1}L$

Preliminaries: Residual Automata

Intuition: States $\leftrightarrow$ **natural components** of the language

Definition: An NFA for which all states define **left quotients** of the language it accepts

||

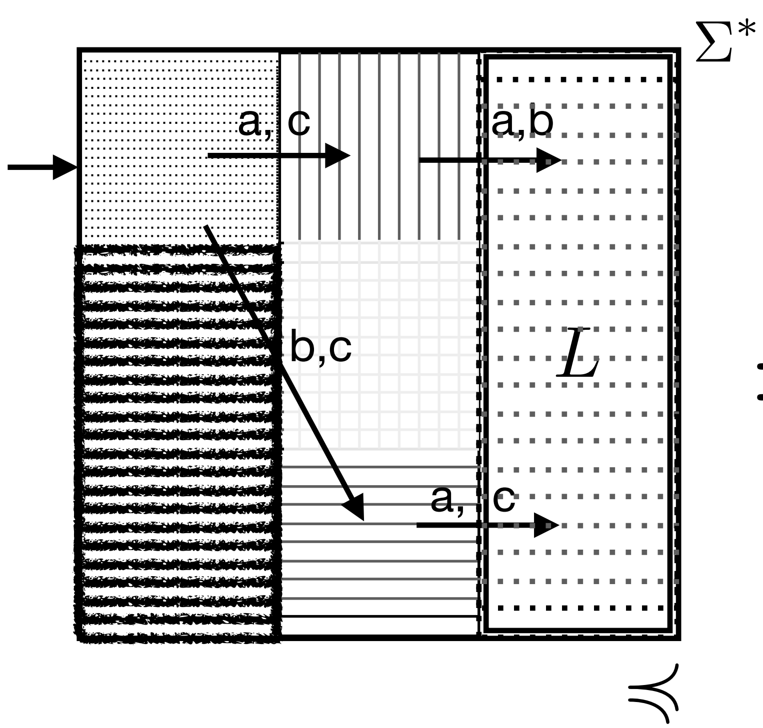
residuals

u : word

L : language

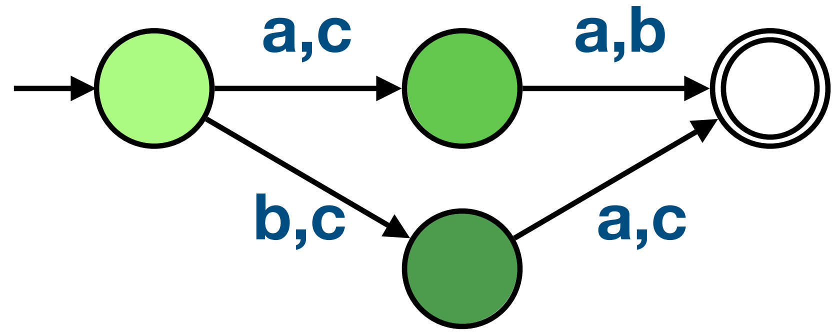
$$u^{-1}L \overset{\text{def}}{=} \{w \mid uw \in L\}$$

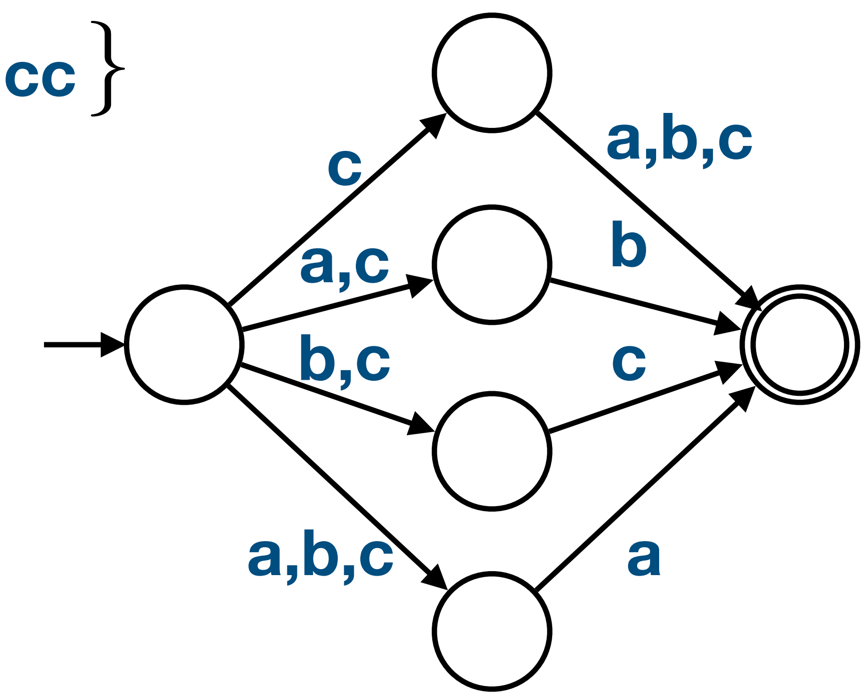
Example:



$$L = \{\textbf{aa, ab, ba, bc, ca, cb, cc}\}$$

$=$





P. Valero, IMDEA Software, Madrid

Preliminaries: Residual Automata

Intuition: States $\leftrightarrow$ **natural components** of the language

Definition: An NFA for which all states define **left quotients** of the language it accepts

||

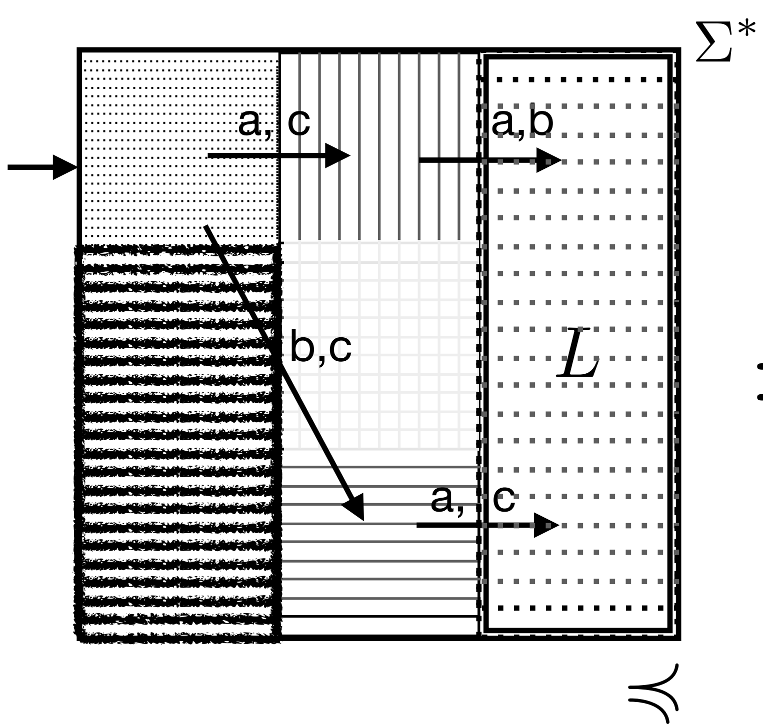
residuals

u : word

L : language

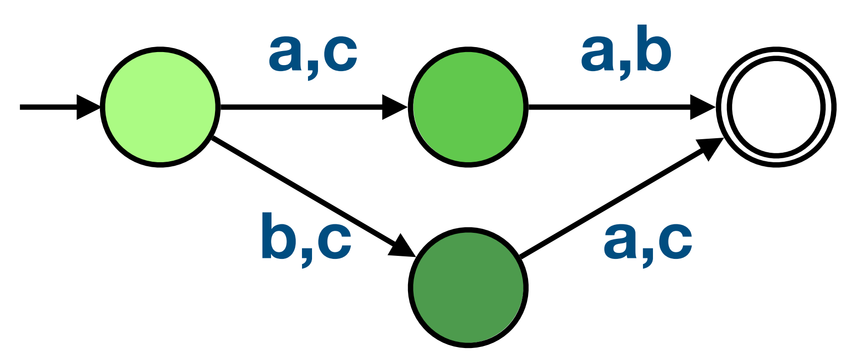
$$u^{-1}L \stackrel{\text{def}}{=} \{w \mid uw \in L\}$$

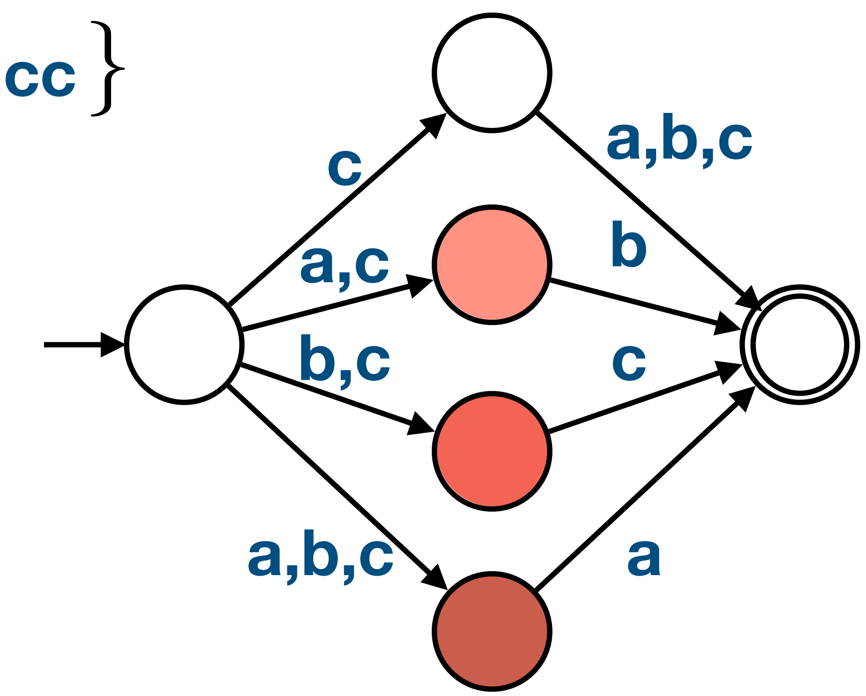
Example:

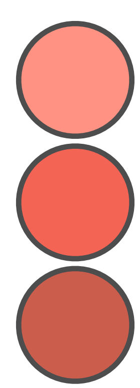


$=$

$L = \{$**aa, ab, ba, bc, ca, cb, cc**$\}$







defines $\{$**b**$\}$

defines $\{$**c**$\}$

defines $\{$**a**$\}$

Not **residuals** for any u

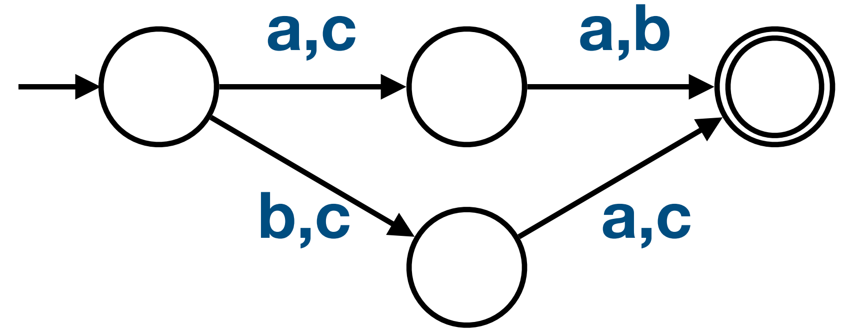
not RFA

Preliminaries: Reverse

Reverse Construction

$\mathcal{L}(\mathcal{N}) = \{$ aa, ab, ba, bc, ca, cb, cc $\}$

$\mathcal{N}$

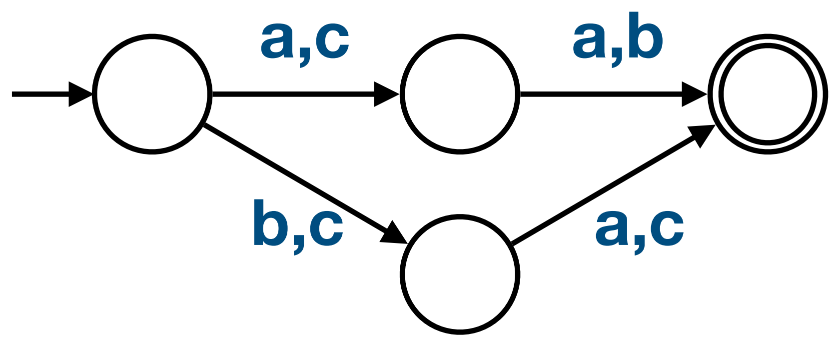


Preliminaries: Reverse

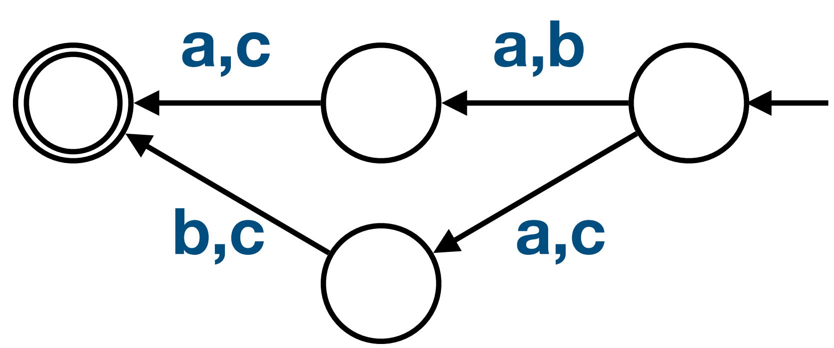
Reverse Construction

$\mathcal{L}(\mathcal{N}) = \{$ **aa, ab, ba, bc, ca, cb, cc** $\}$

$\mathcal{N}$



$\mathcal{N}^R$



Reverse Automaton

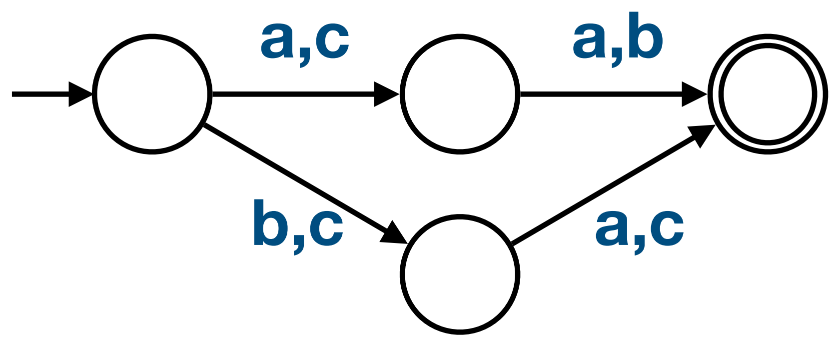
$$\mathcal{L}(\mathcal{N}^R) = \{\, \textbf{aa}, \textbf{ba}, \textbf{ab}, \textbf{cb}, \textbf{bc}, \textbf{cc} \}$$

Preliminaries: Reverse

Reverse Construction

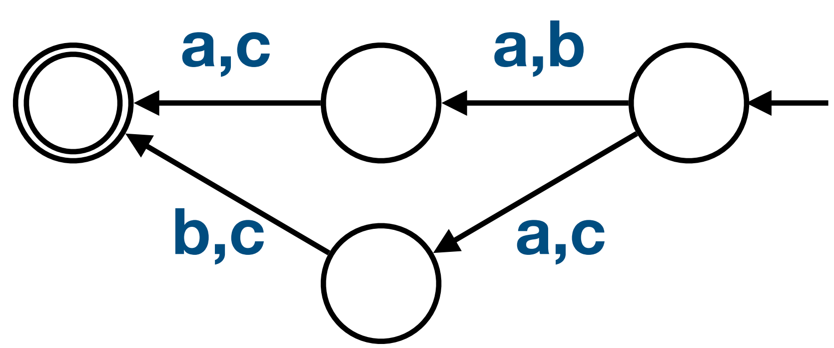
$\mathcal{L}(\mathcal{N}) = \{$ **aa, ab, ba, bc, ca, cb, cc** $\}$

$\mathcal{N}$



$\mathcal{N}^R$

Co-residual (co-RFA)



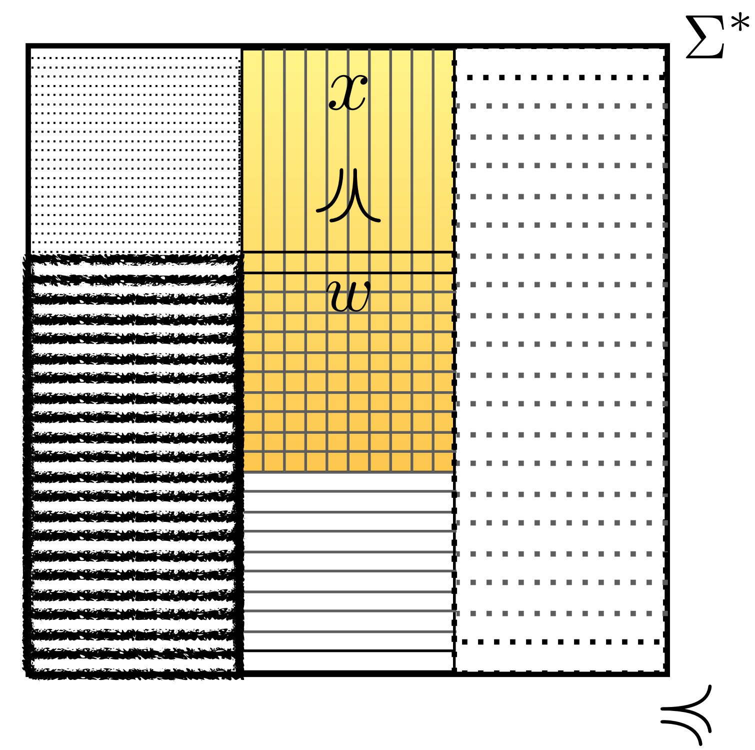
Reverse Automaton

$$\mathcal{L}(\mathcal{N}^R) = \{\, \textbf{aa}, \textbf{ba}, \textbf{ab}, \textbf{cb}, \textbf{bc}, \textbf{cc} \}$$

Preliminaries: Quasiorders

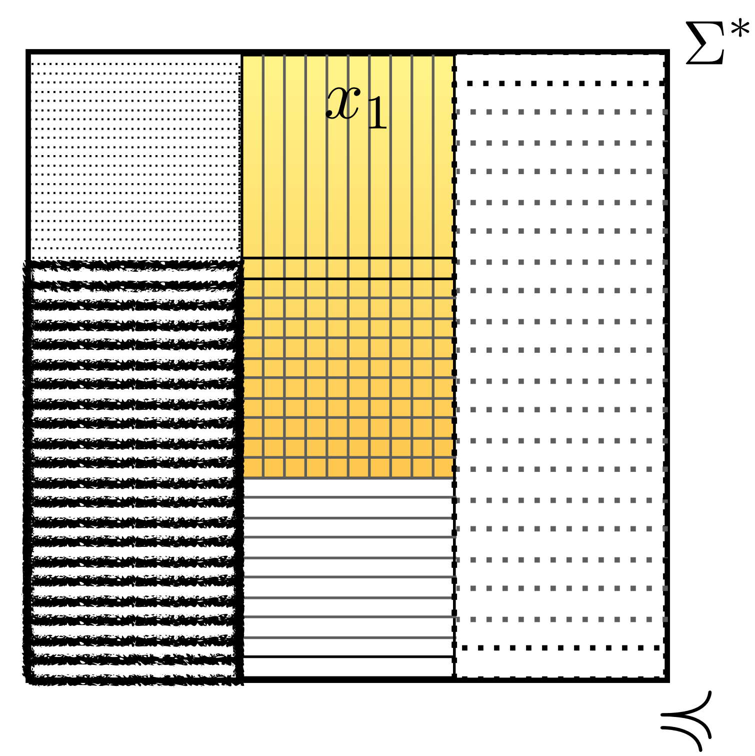
<u>Reflexive</u> and <u>transitive</u> relation on words with good properties w.r.t. concatenation of symbols

- **Closure of** x **w.r.t.** $\preccurlyeq$ **:** $\mathrm{cl}_{\preccurlyeq}(x) \stackrel{\text{def}}{=} \{w \mid x \preccurlyeq w\}$
  (a.k.a. closed set)



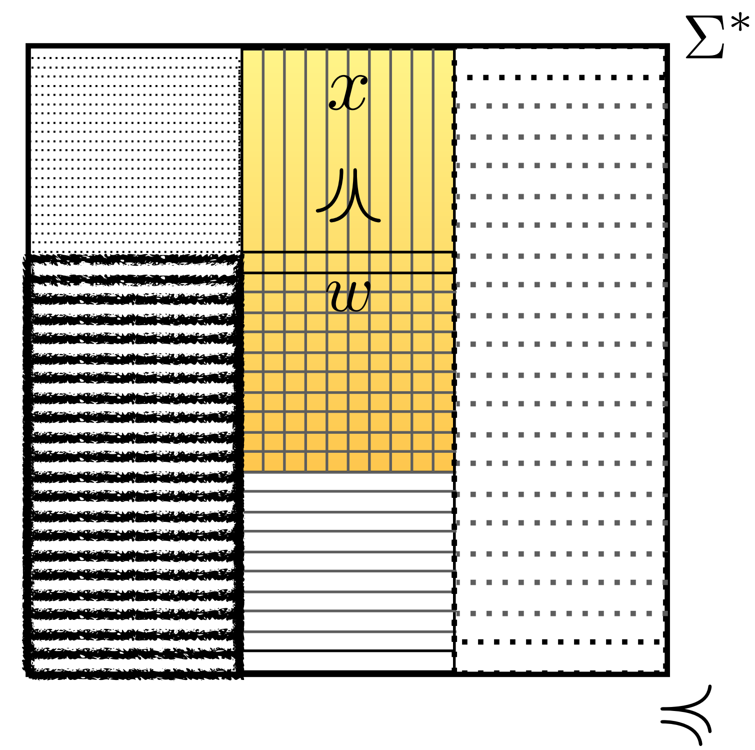
- **Composite** $\mathrm{cl}_{\preccurlyeq}(u)$ **iff** *(simplification)*

$$\mathrm{cl}_{\preccurlyeq}(u) = \bigcap_{\mathrm{cl}_{\preccurlyeq}(u) \subset \mathrm{cl}_{\preccurlyeq}(x)} \mathrm{cl}_{\preccurlyeq}(x)$$



Preliminaries: Quasiorders

<u>Reflexive</u> and <u>transitive</u> relation on words with good properties w.r.t. concatenation of symbols

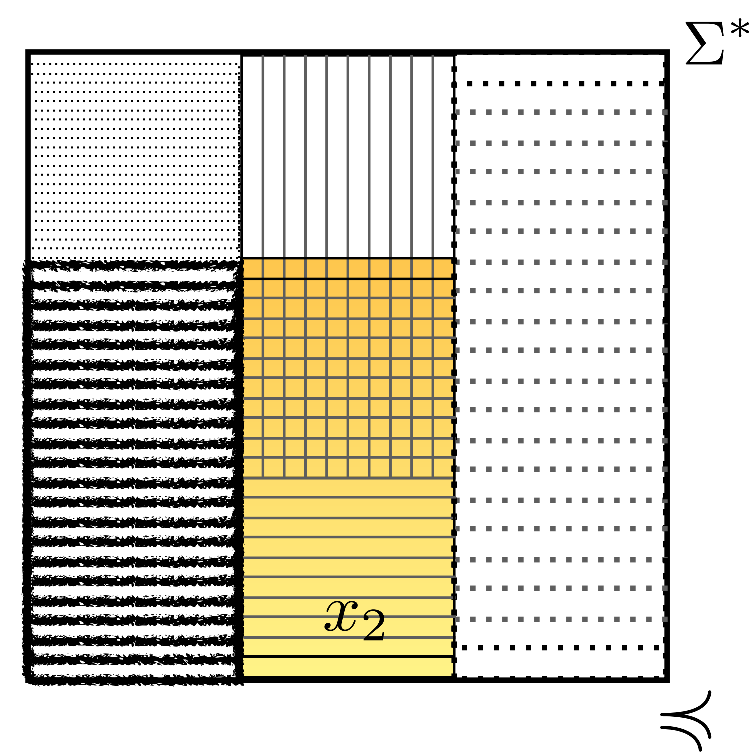


- **Closure of** x **w.r.t.** $\preccurlyeq$ **:** $\mathrm{cl}_{\preccurlyeq}(x) \stackrel{\text{def}}{=} \{w \mid x \preccurlyeq w\}$

  (a.k.a. closed set)

- **Composite** $\mathrm{cl}_{\preccurlyeq}(u)$ **iff** *(simplification)*

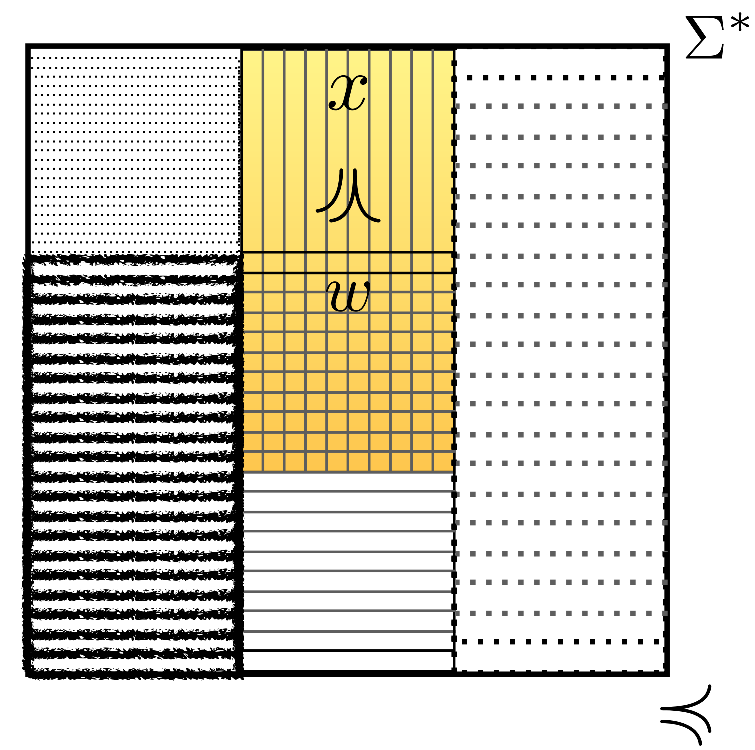
$$\mathrm{cl}_{\preccurlyeq}(u) = \bigcap_{\mathrm{cl}_{\preccurlyeq}(u) \subset \mathrm{cl}_{\preccurlyeq}(x)} \mathrm{cl}_{\preccurlyeq}(x)$$



Preliminaries: Quasiorders

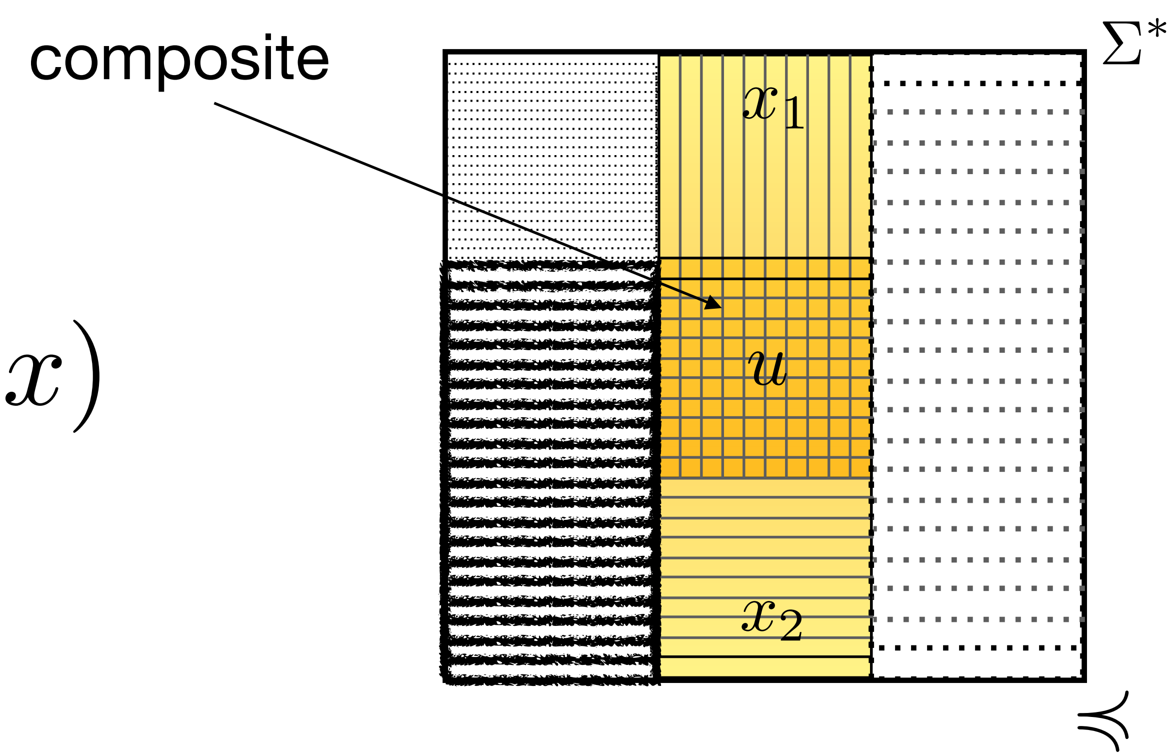
<u>Reflexive</u> and <u>transitive</u> relation on words with good properties w.r.t. concatenation of symbols

- **Closure of x w.r.t. $\preccurlyeq$:** $\mathrm{cl}_{\preccurlyeq}(x) \stackrel{\text{def}}{=} \{w \mid x \preccurlyeq w\}$
  (a.k.a. closed set)



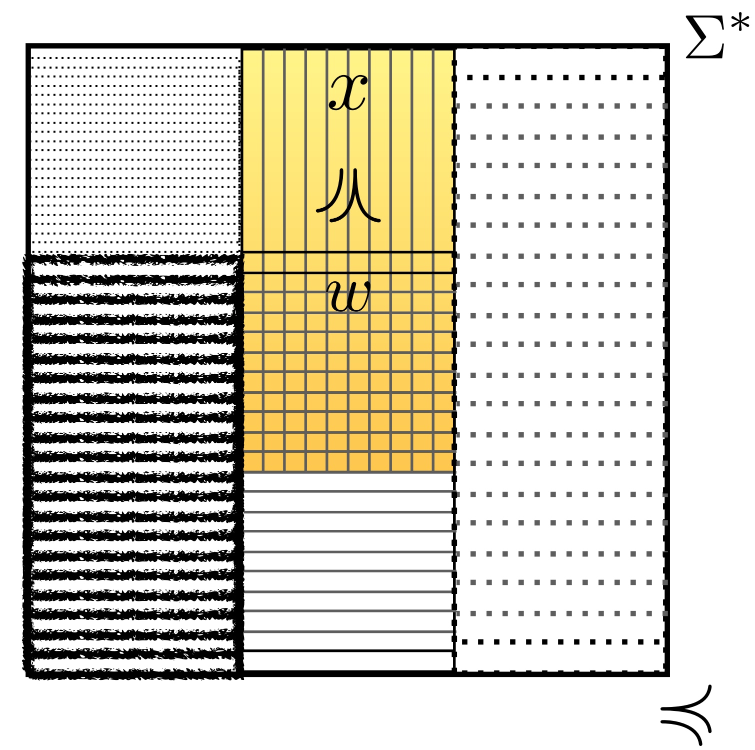
- **Composite** $\mathrm{cl}_{\preccurlyeq}(u)$ **iff** *(simplification)*

$$\mathrm{cl}_{\preccurlyeq}(u) = \bigcap_{\mathrm{cl}_{\preccurlyeq}(u) \subset \mathrm{cl}_{\preccurlyeq}(x)} \mathrm{cl}_{\preccurlyeq}(x)$$



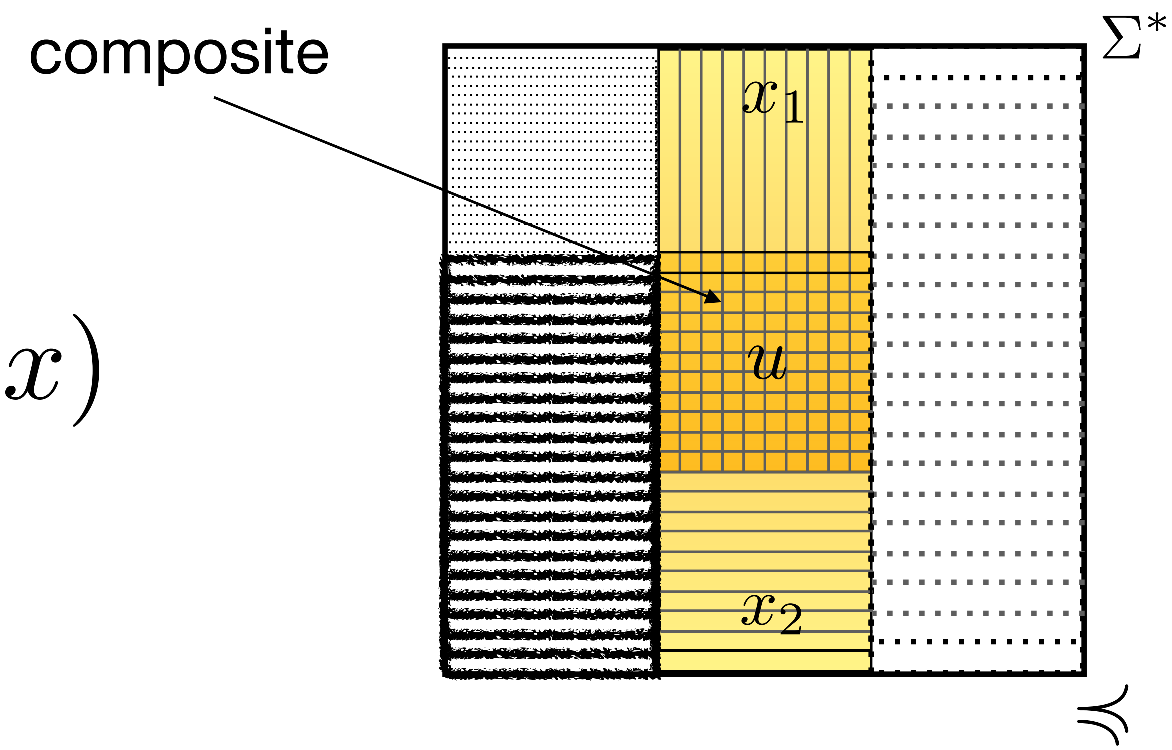
Preliminaries: Quasiorders

<u>Reflexive</u> and <u>transitive</u> relation on words with good properties w.r.t. concatenation of symbols



- **Closure of** x **w.r.t.** $\preccurlyeq$ **:** $\mathrm{cl}_{\preccurlyeq}(x) \stackrel{\text{def}}{=} \{w \mid x \preccurlyeq w\}$
  (a.k.a. closed set)

- **Composite** $\mathrm{cl}_{\preccurlyeq}(u)$ **iff** *(simplification)*



$$\mathrm{cl}_{\preccurlyeq}(u) = \bigcap_{\mathrm{cl}_{\preccurlyeq}(u) \subset \mathrm{cl}_{\preccurlyeq}(x)} \mathrm{cl}_{\preccurlyeq}(x)$$

Otherwise, **prime**

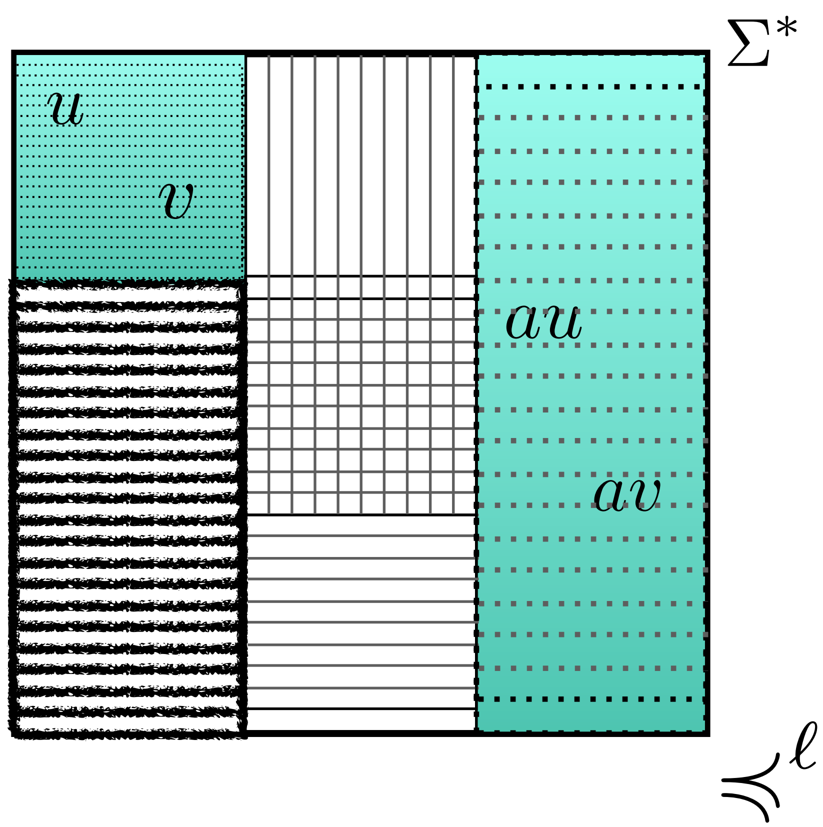
Preliminaries: Quasiorders

Reflexive and transitive relation on words with good properties w.r.t. concatenation of symbols

Left quasiorders:

$\forall a \in \Sigma :$

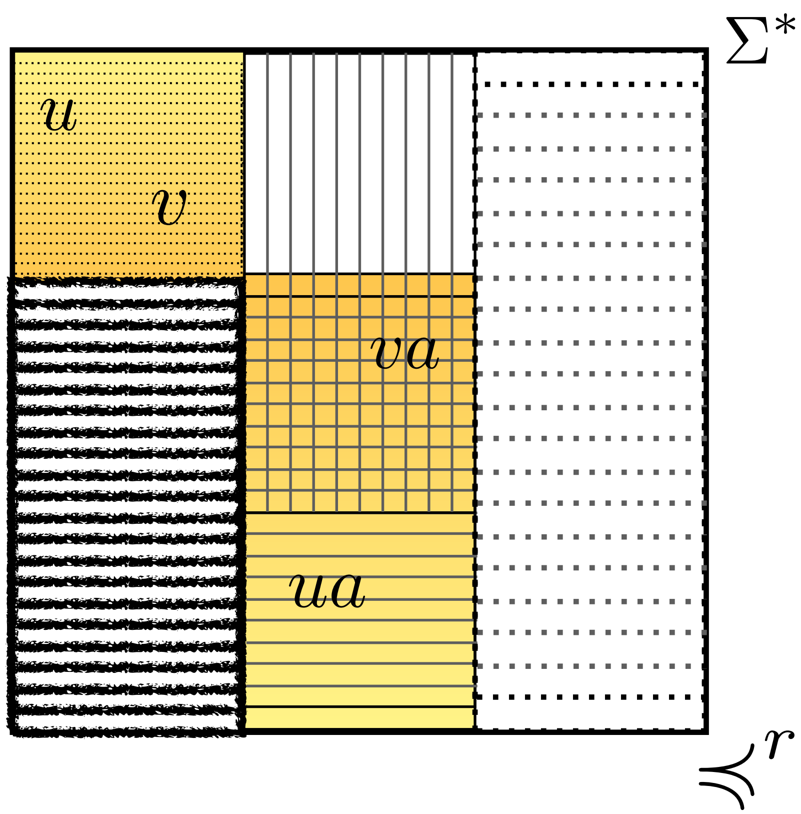
$$u \preccurlyeq^{\ell} v \implies au \preccurlyeq^{\ell} av$$



Right quasiorders:

$\forall a \in \Sigma :$

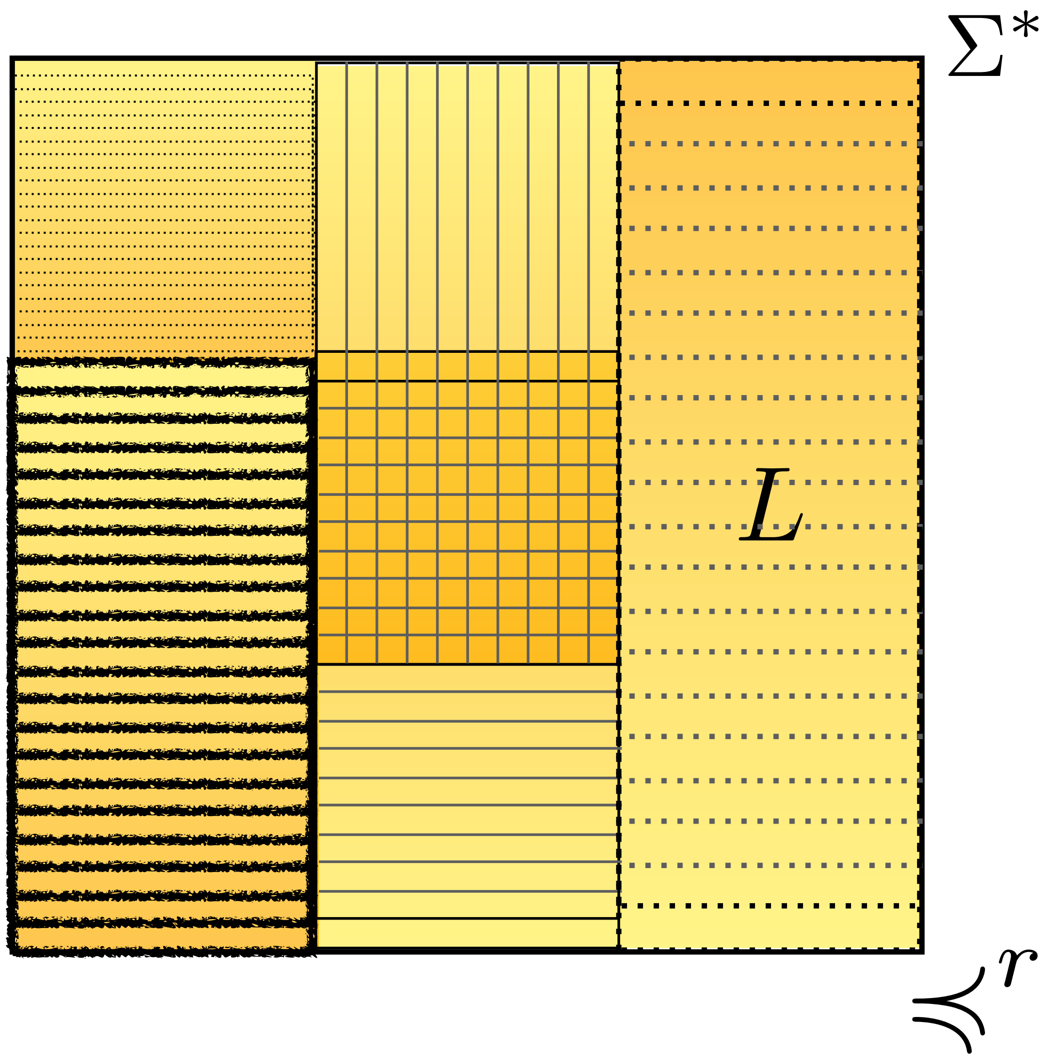
$$u \preccurlyeq^{r} v \implies ua \preccurlyeq^{r} va$$



How to build a residual automaton from a right quasiorder?

[Valero et. al, MFCS 2020]

- $\preccurlyeq^r$ is a **finite** right quasiorder
- $\mathrm{cl}_{\preccurlyeq^r}(L) = L$ ($\preccurlyeq^r$ precisely represents L)

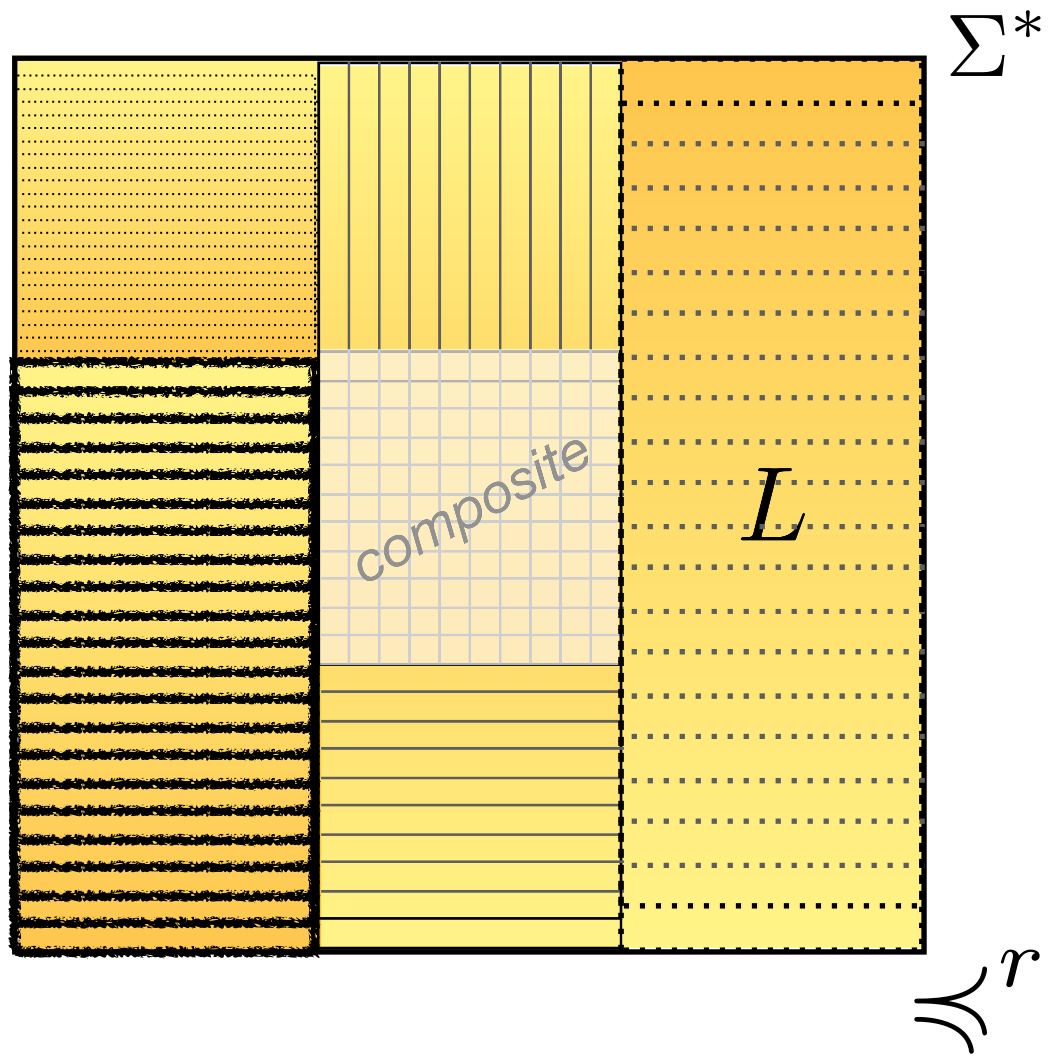


The RFA accepts L

How to build a residual automaton from a right quasiorder?

[Valero et. al, MFCS 2020]

- $\preccurlyeq^r$ is a **finite** right quasiorder
- $\mathrm{cl}_{\preccurlyeq^r}(L) = L$ ($\preccurlyeq^r$ precisely represents L)

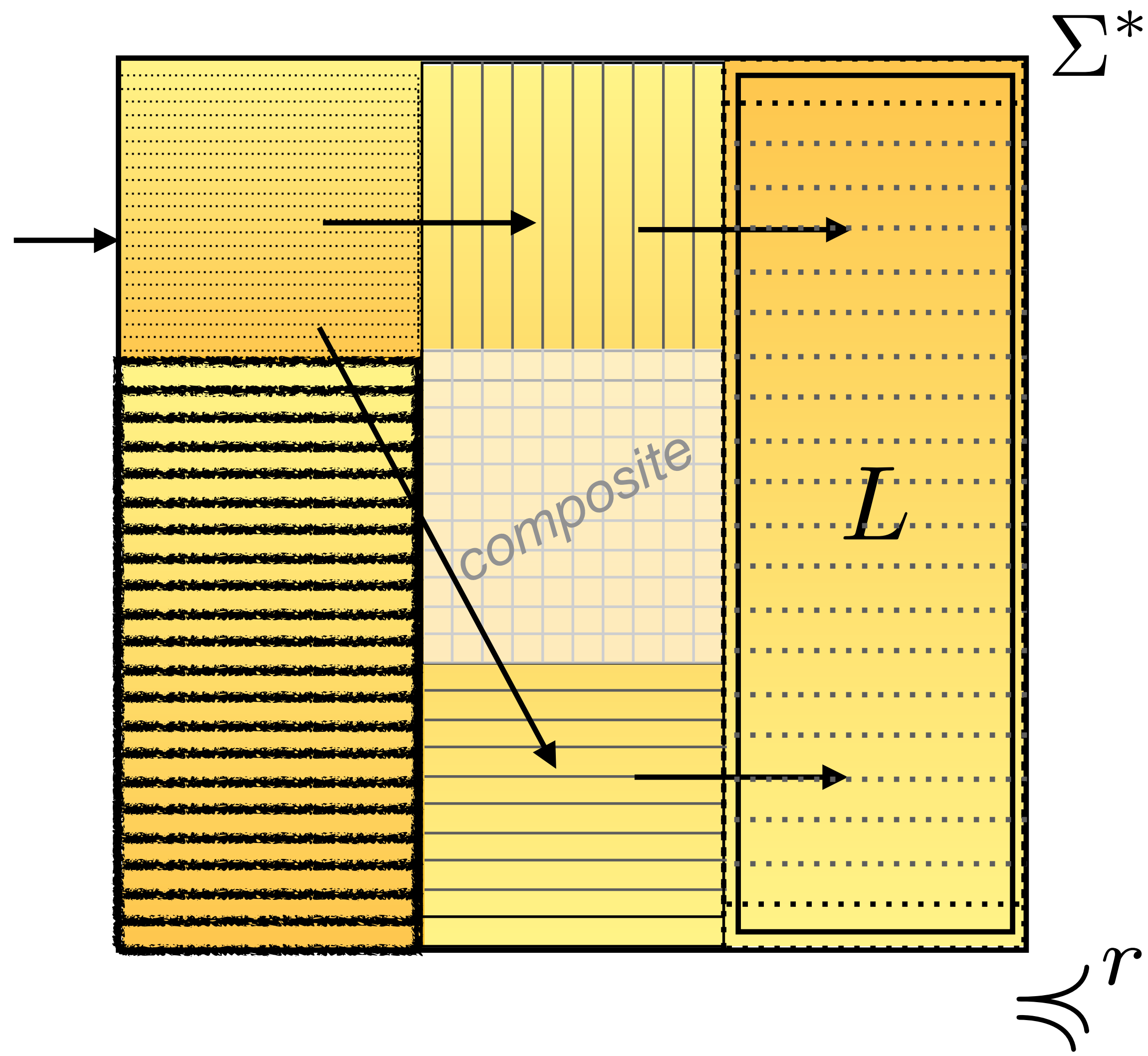


The RFA accepts L

How to build a residual automaton from a right quasiorder?

[Valero et. al, MFCS 2020]

- $\preccurlyeq^r$ is a **finite** right quasiorder
- $\mathrm{cl}_{\preccurlyeq^r}(L) = L$ ($\preccurlyeq^r$ precisely represents L)

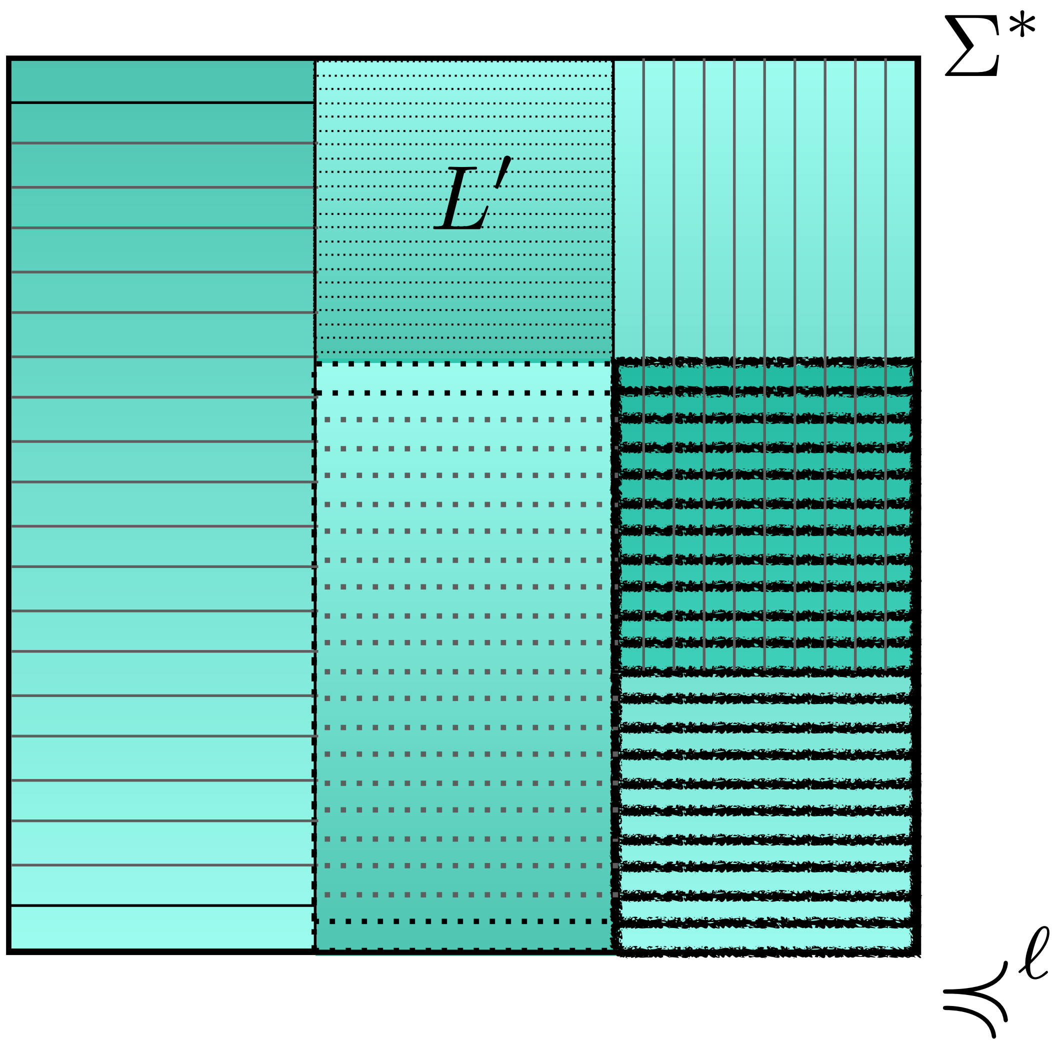


The RFA accepts L

How to build a co-residual automaton from a left quasiorder?

[Valero et. al, MFCS 2020]

- $\preccurlyeq^{\ell}$ is a **finite left** quasiorder
- $\mathrm{cl}_{\preccurlyeq^{\ell}}(L') = L'$ ($\preccurlyeq^{\ell}$ precisely represents L')

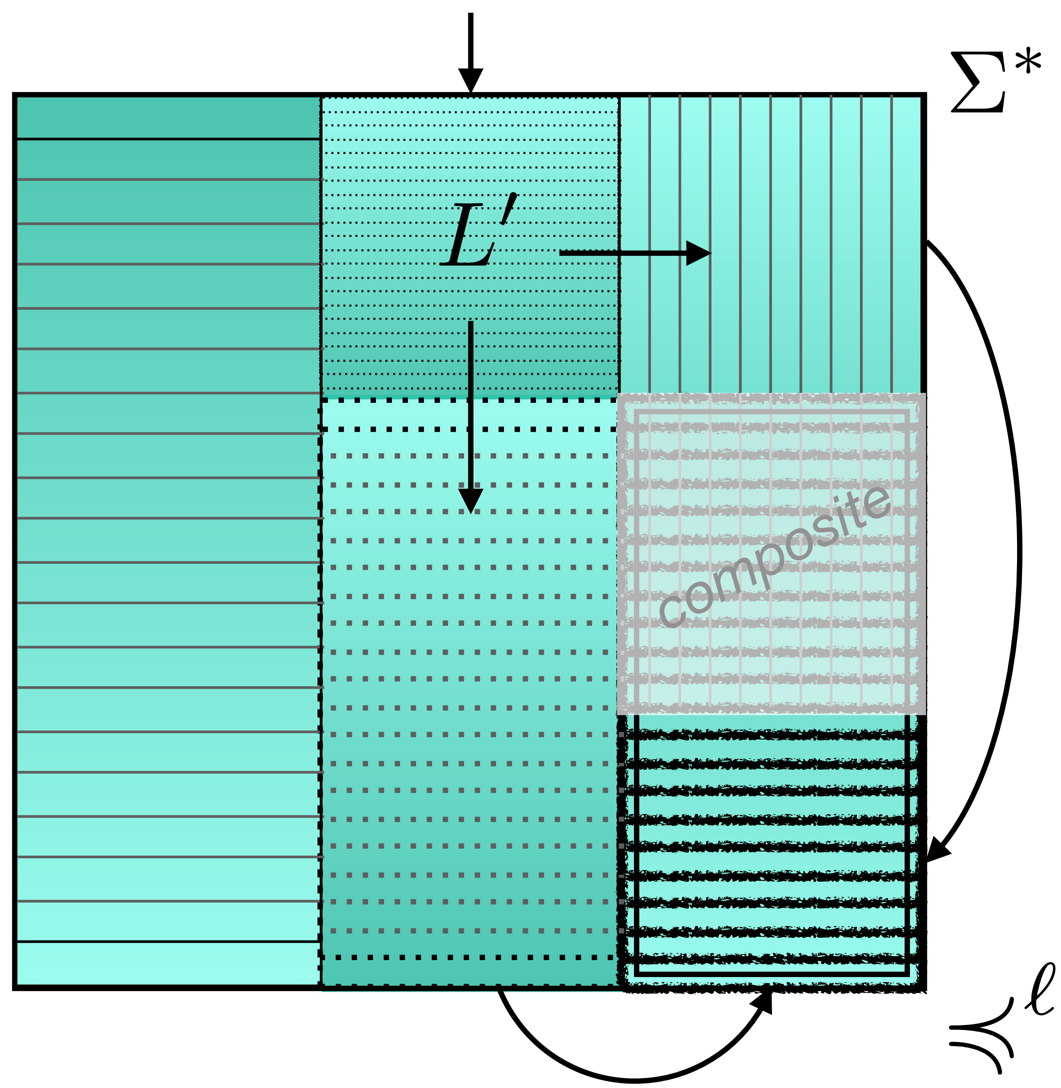


The co-RFA accepts L'

How to build a co-residual automaton from a left quasiorder?

[Valero et. al, MFCS 2020]

- $\preccurlyeq^{\ell}$ is a **finite** left quasiorder
- $\mathrm{cl}_{\preccurlyeq^{\ell}}(L') = L'$ ($\preccurlyeq^{\ell}$ precisely represents L')

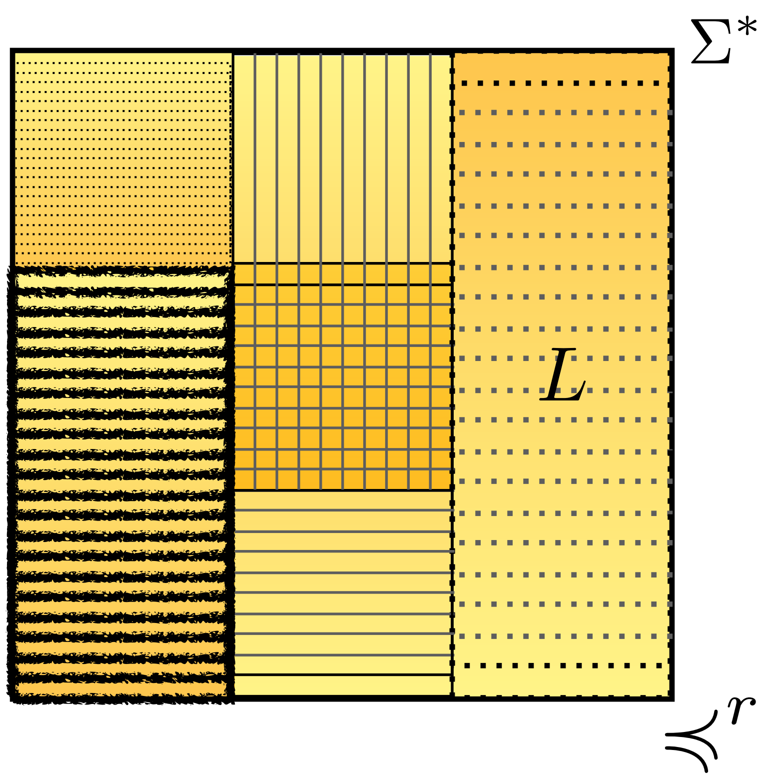


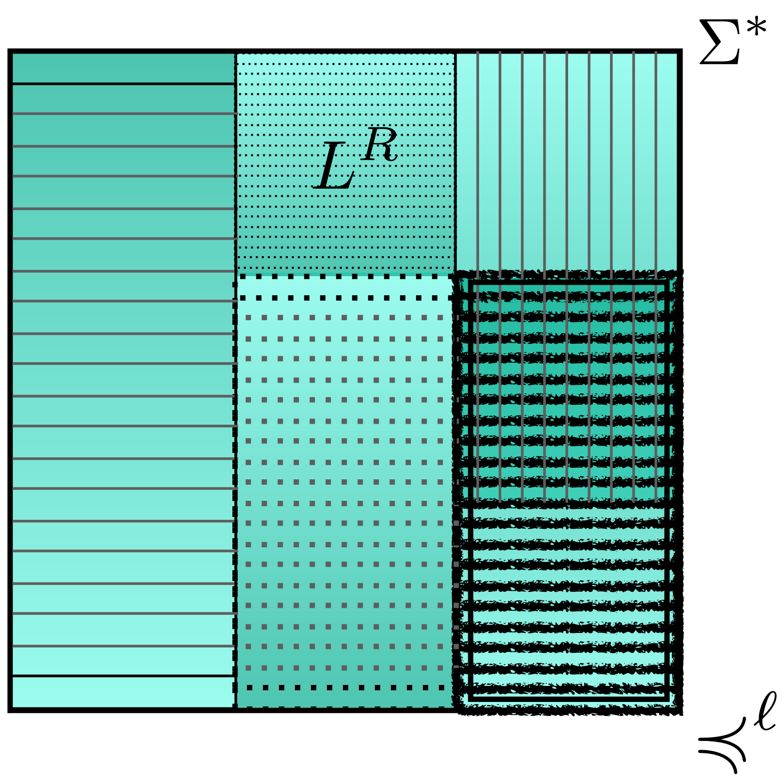
The co-RFA accepts L'

A property of dual quasiorders $\preccurlyeq^r$ and $\preccurlyeq^\ell$

[Valero et. al, MFCS 2020]

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

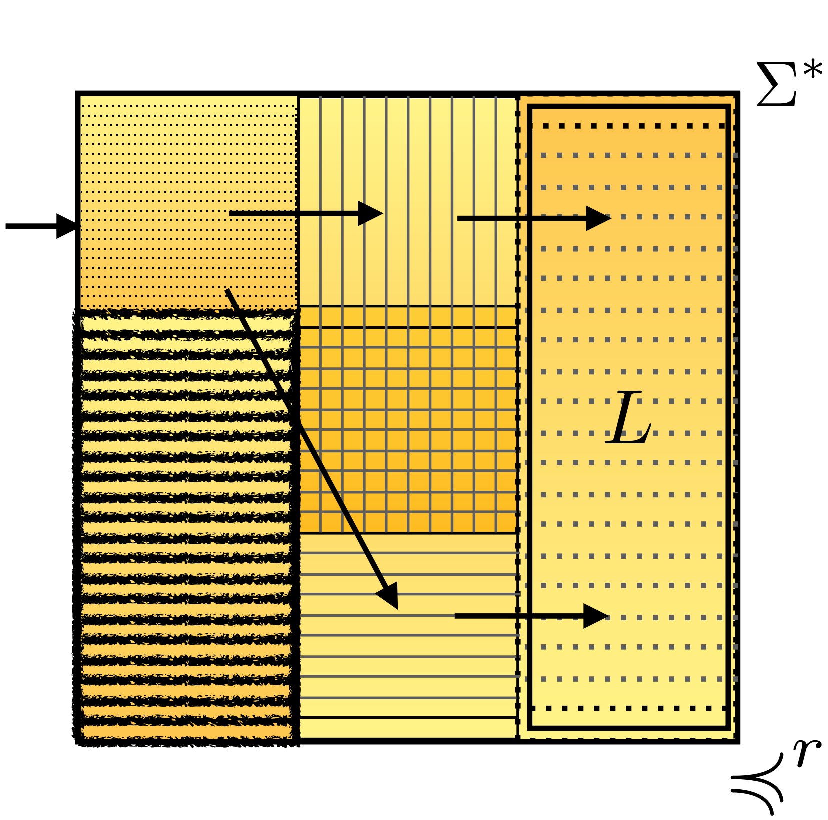


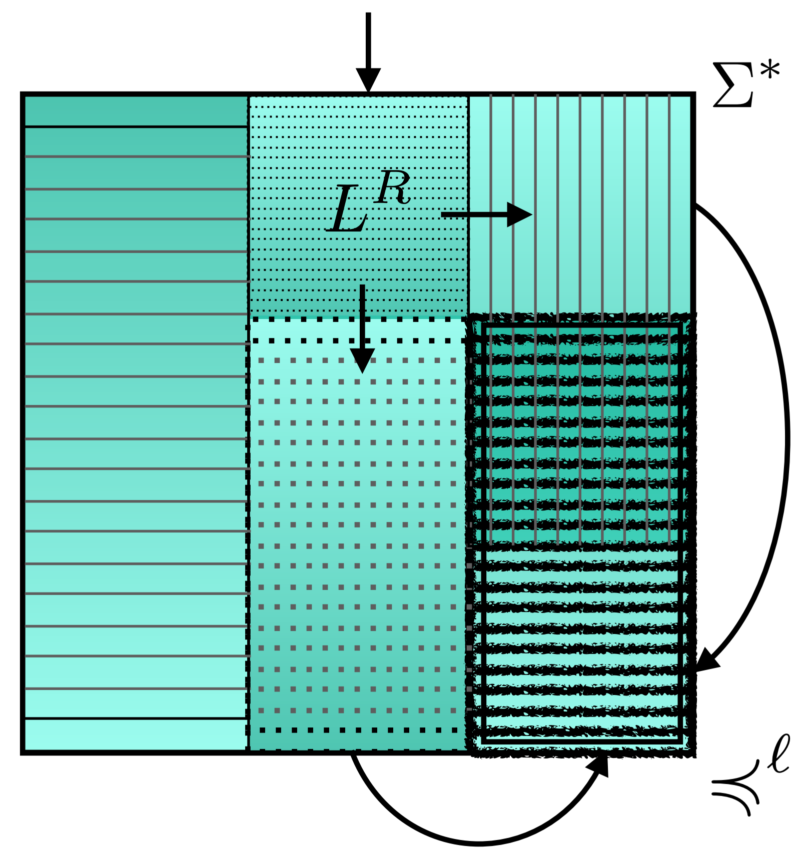


A property of dual quasiorders $\preccurlyeq^r$ and $\preccurlyeq^\ell$

[Valero et. al, MFCS 2020]

<u>**Lemma:**</u> Given $u \preccurlyeq^r v \Leftrightarrow u^R \preccurlyeq^\ell v^R$

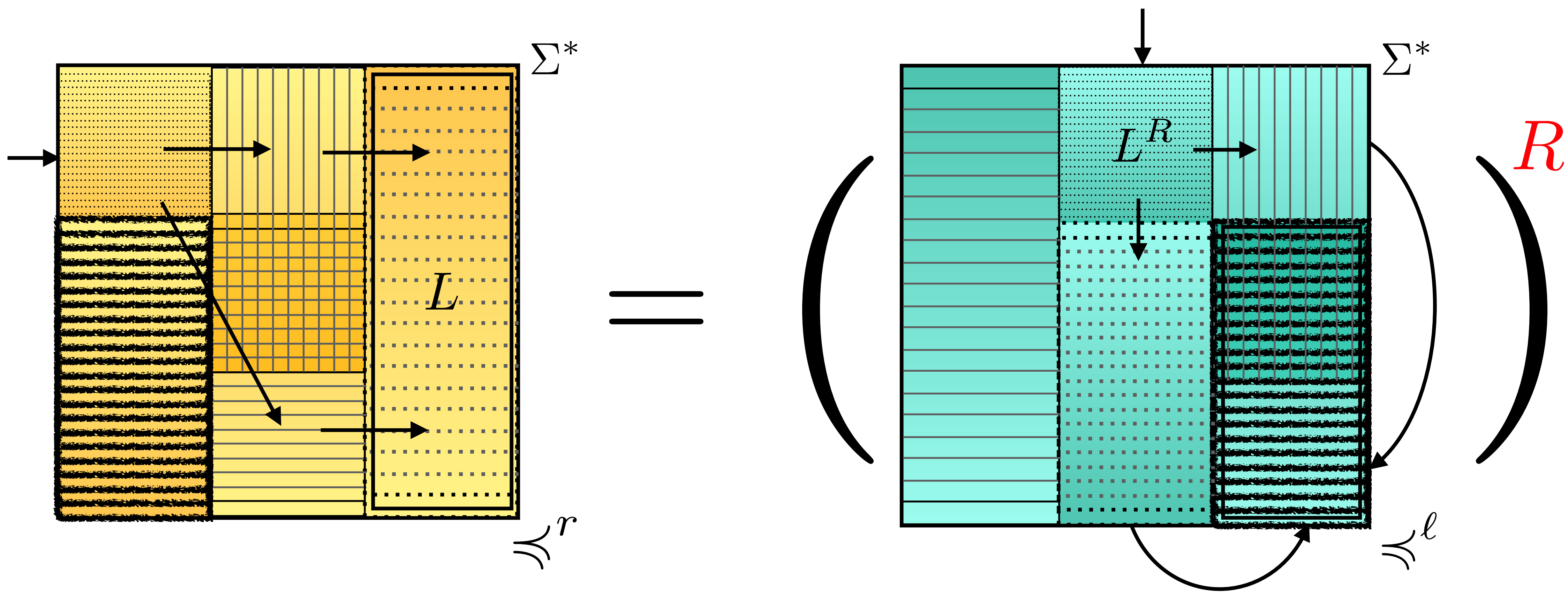




A property of dual quasiorders $\preccurlyeq^r$ and $\preccurlyeq^\ell$

[Valero et. al, MFCS 2020]

<u>**Lemma:**</u> Given $u \preccurlyeq^r v \Leftrightarrow u^R \preccurlyeq^\ell v^R$ then



Instances of right quasiorders

- $\preccurlyeq^r$ is a **finite** right quasiorder
- $\mathrm{cl}_{\preccurlyeq^r}(L) = L$

[Valero et. al, MFCS 2020]

Instances of right quasiorders

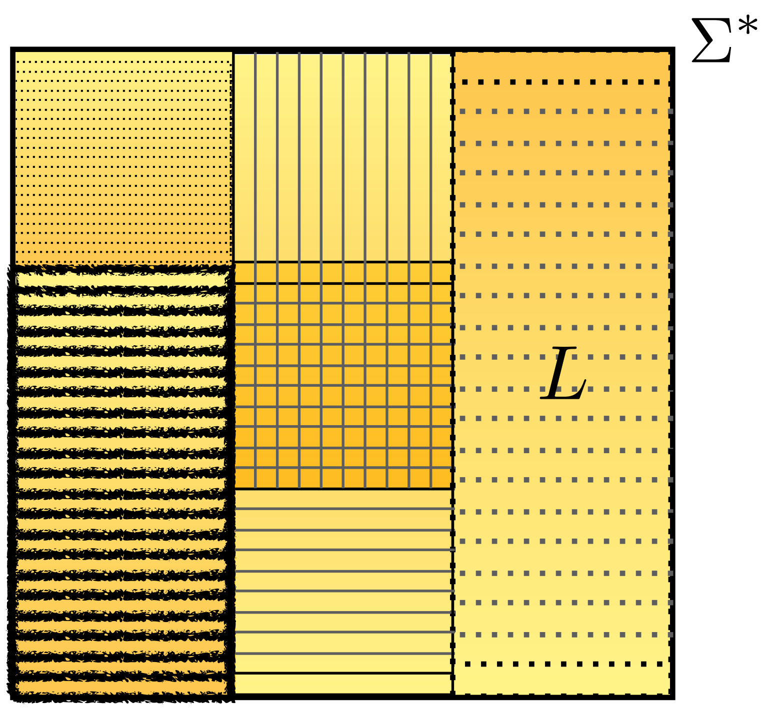
- $\preccurlyeq^r$ is a **finite** right quasiorder
- $\mathrm{cl}_{\preccurlyeq^r}(L) = L$

[Valero et. al, MFCS 2020]

Language-based [de Luca and Varricchio, 1994]

Given a regular language L

$$u \preccurlyeq_L^r v \Leftrightarrow u^{-1}L \subseteq v^{-1}L$$



The canonical RFA for L

Instances of right quasiorders

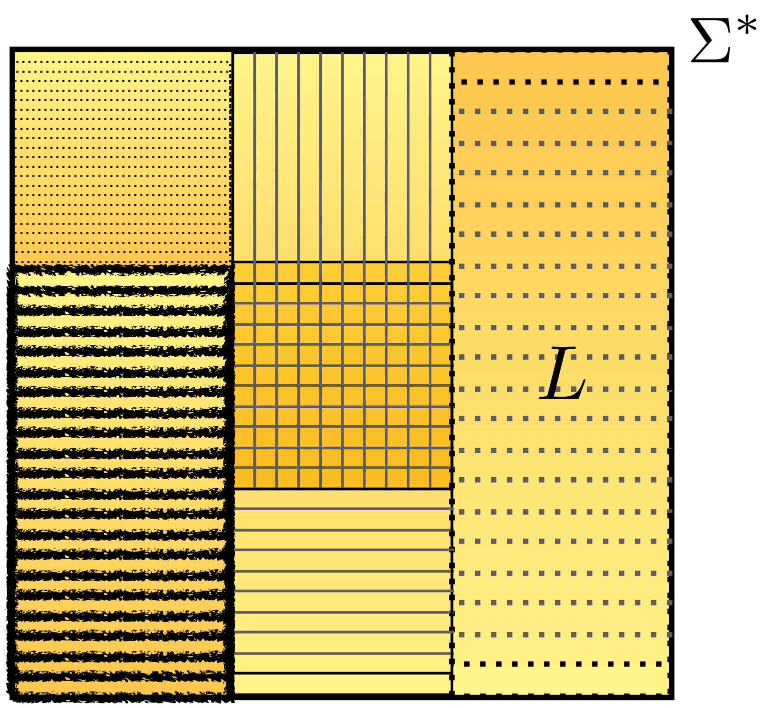
- $\preccurlyeq^r$ is a **finite** right quasiorder
- $\mathrm{cl}_{\preccurlyeq^r}(L) = L$

[Valero et. al, MFCS 2020]

Language-based [de Luca and Varricchio, 1994]

Given a regular language L

$$u \preccurlyeq^r_L v \Leftrightarrow u^{-1}L \subseteq v^{-1}L$$



The canonical RFA for L

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

Instances of right quasiorders

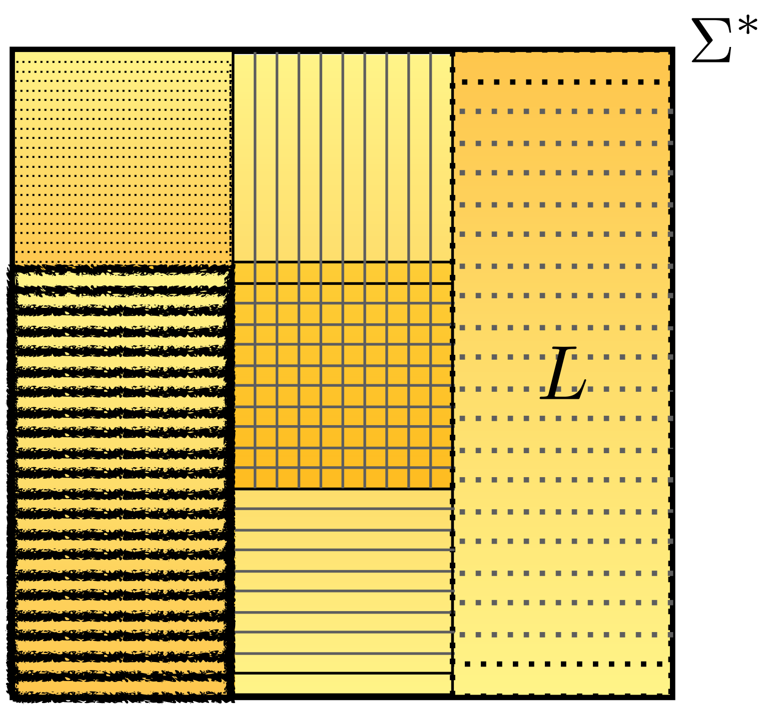
- $\preccurlyeq^r$ is a **finite** right quasiorder
- $\mathrm{cl}_{\preccurlyeq^r}(L) = L$

[Valero et. al, MFCS 2020]

Language-based [de Luca and Varricchio, 1994]

Given a regular language L

$$u \preccurlyeq^r_L v \Leftrightarrow u^{-1}L \subseteq v^{-1}L$$



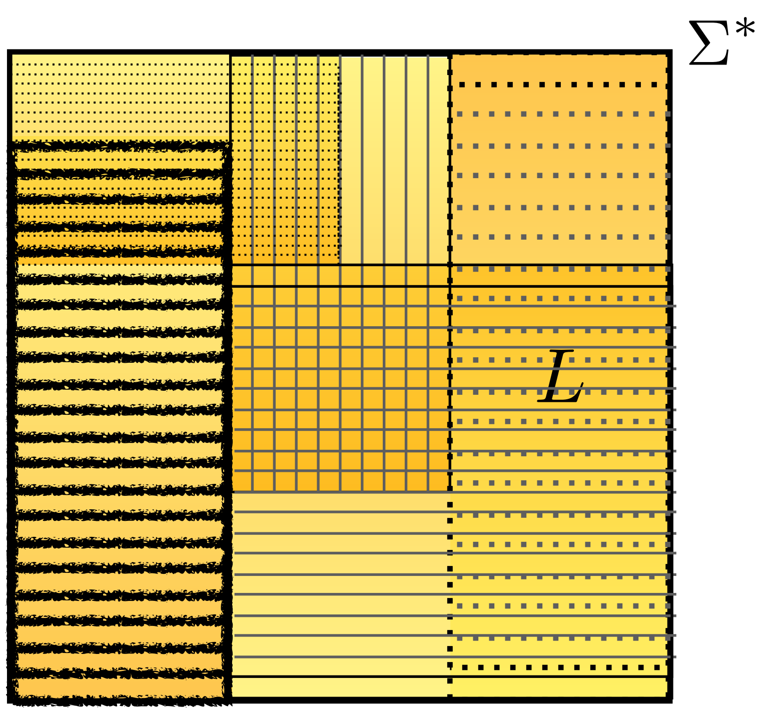
The canonical RFA for L

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

Automata-based

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

$$u \preccurlyeq^r_{\mathcal{N}} v \Leftrightarrow \mathrm{post}^{\mathcal{N}}_u(I) \subseteq \mathrm{post}^{\mathcal{N}}_v(I)$$



An RFA for L : new residualization operation

Instances of right quasiorders

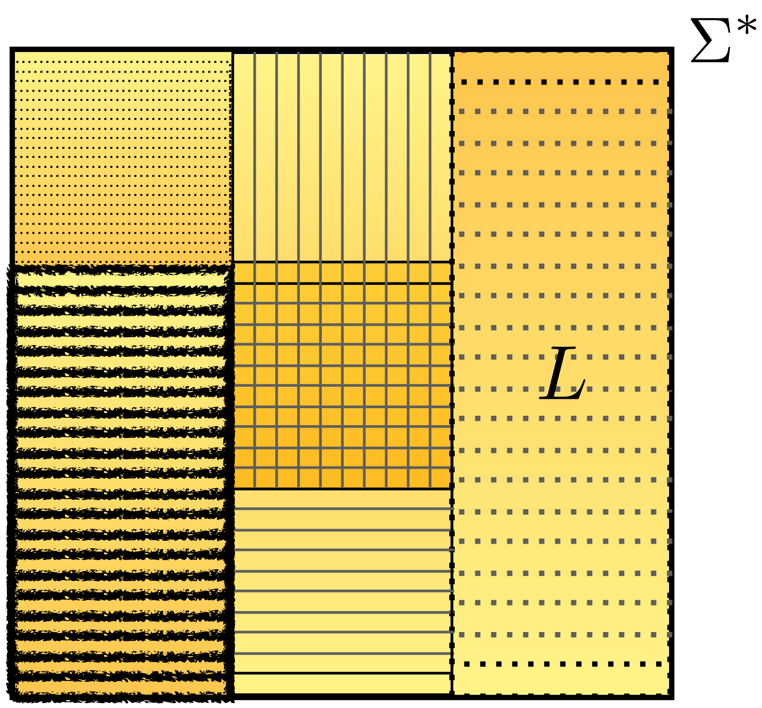
- $\preccurlyeq^r$ is a **finite right** quasiorder
- $\mathrm{cl}_{\preccurlyeq^r}(L) = L$

[Valero et. al, MFCS 2020]

Language-based [de Luca and Varricchio, 1994]

Given a regular language L

$$u \preccurlyeq^r_L v \Leftrightarrow u^{-1}L \subseteq v^{-1}L$$



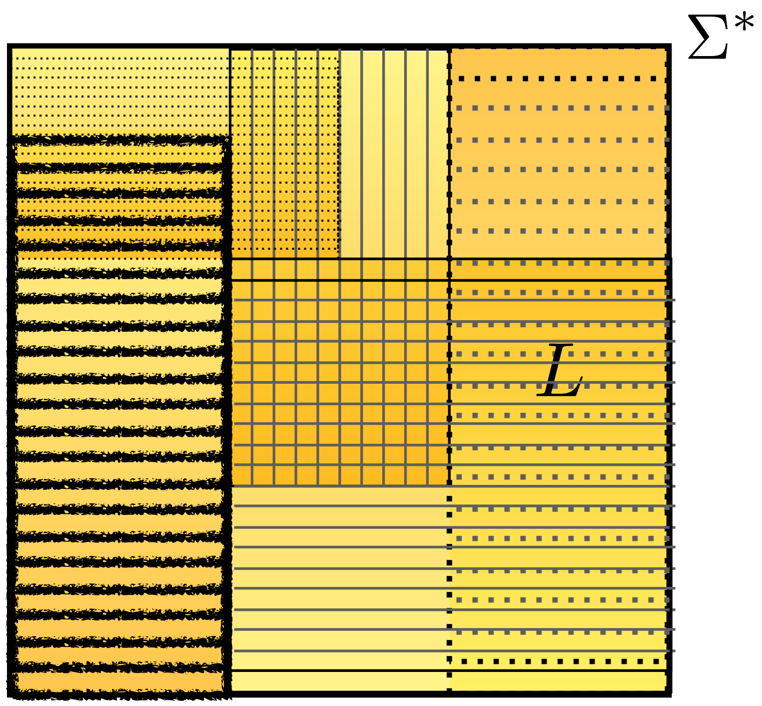
The canonical RFA for L

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

Automata-based

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

$$u \preccurlyeq^r_{\mathcal{N}} v \Leftrightarrow \mathrm{post}^{\mathcal{N}}_u(I) \subseteq \mathrm{post}^{\mathcal{N}}_v(I)$$



An RFA for L : new residualization operation

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

Instances of right quasiorders

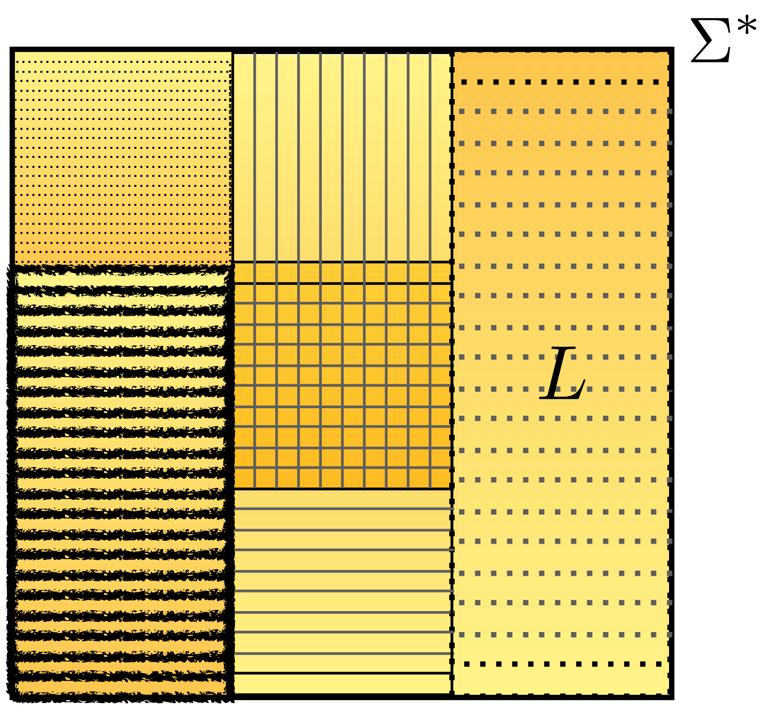
- $\preccurlyeq^r$ is a **finite** right quasiorder
- $\mathrm{cl}_{\preccurlyeq^r}(L) = L$

[Valero et. al, MFCS 2020]

Language-based [de Luca and Varricchio, 1994]

Given a regular language L

$$u \preccurlyeq^r_L v \Leftrightarrow u^{-1}L \subseteq v^{-1}L$$



The canonical RFA for L

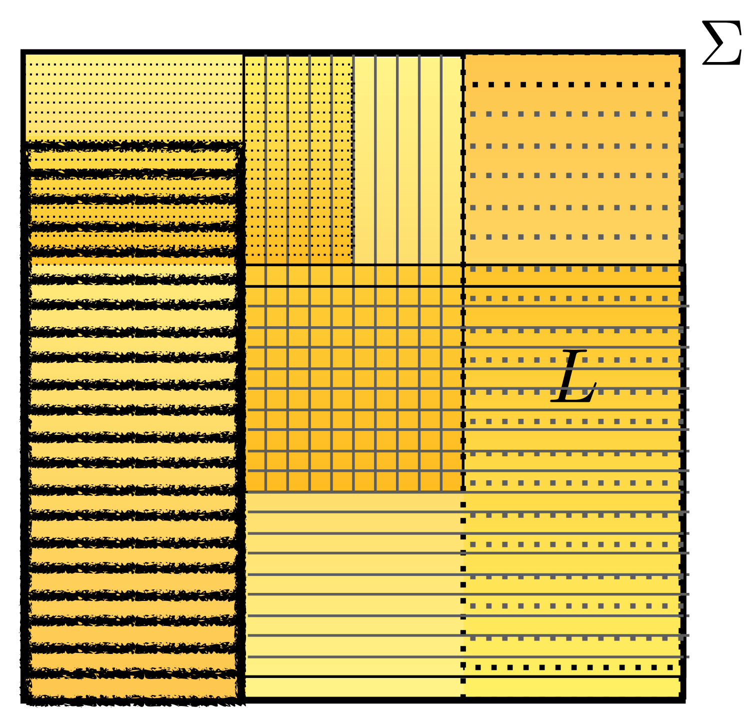
$\mathsf{Can}^r(L)$

$\succeq$

Automata-based

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

$$u \preccurlyeq^r_{\mathcal{N}} v \Leftrightarrow \mathrm{post}^{\mathcal{N}}_u(I) \subseteq \mathrm{post}^{\mathcal{N}}_v(I)$$



An RFA for L : new residualization operation

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

Instances of right quasiorders

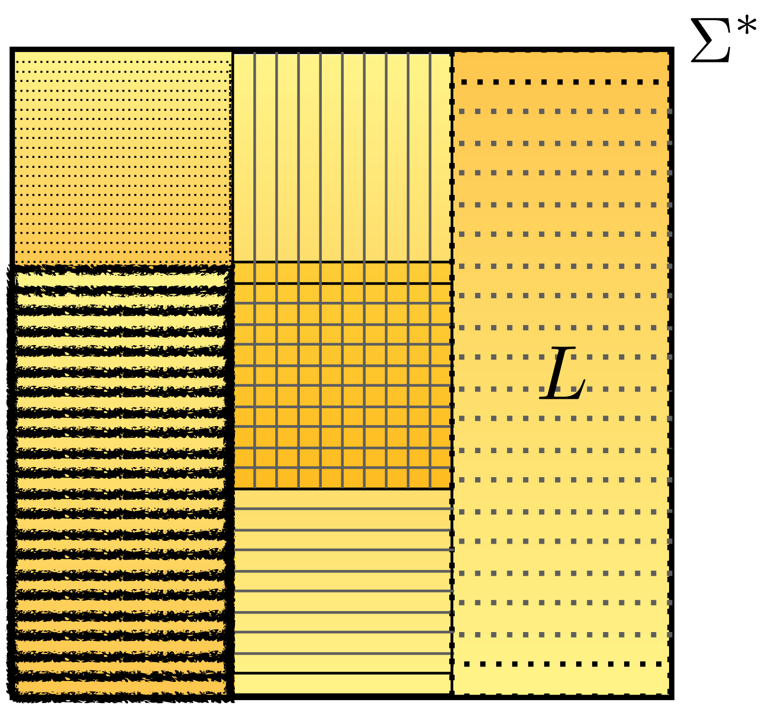
- $\preccurlyeq^r$ is a **finite** right quasiorder
- $\mathrm{cl}_{\preccurlyeq^r}(L) = L$

[Valero et. al, MFCS 2020]

Language-based [de Luca and Varricchio, 1994]

Given a regular language L

$$u \preccurlyeq^r_L v \Leftrightarrow u^{-1}L \subseteq v^{-1}L$$

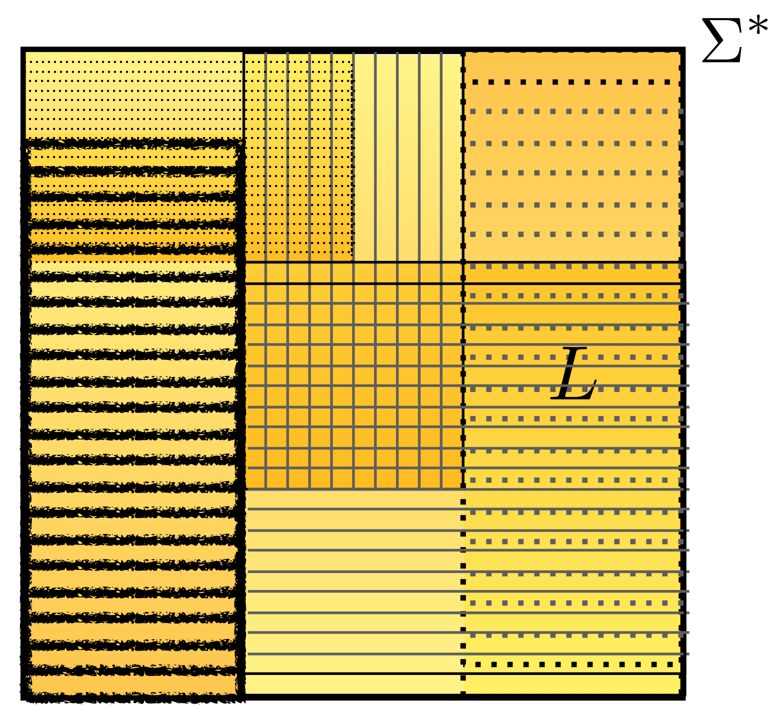


$$\stackrel{?}{=}$$

Automata-based

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

$$u \preccurlyeq^r_{\mathcal{N}} v \Leftrightarrow \mathrm{post}^{\mathcal{N}}_u(I) \subseteq \mathrm{post}^{\mathcal{N}}_v(I)$$



Instances of right quasiorders

- $\preccurlyeq^r$ is a **finite right** quasiorder
- $\mathrm{cl}_{\preccurlyeq^r}(L) = L$

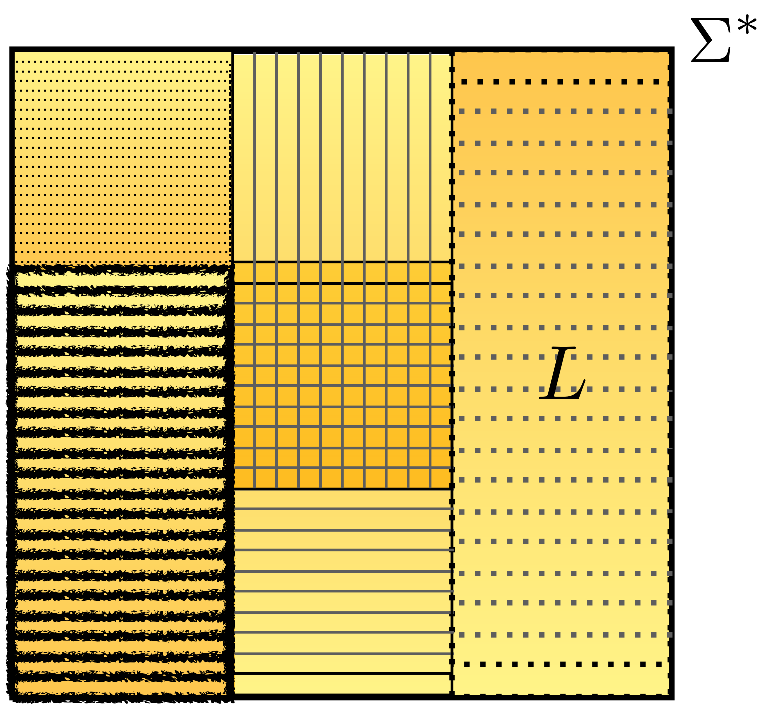
[Valero et. al, MFCS 2020]

Language-based

[de Luca and Varricchio, 1994]

Given a regular language L

$$u \preccurlyeq^r_L v \Leftrightarrow u^{-1}L \subseteq v^{-1}L$$

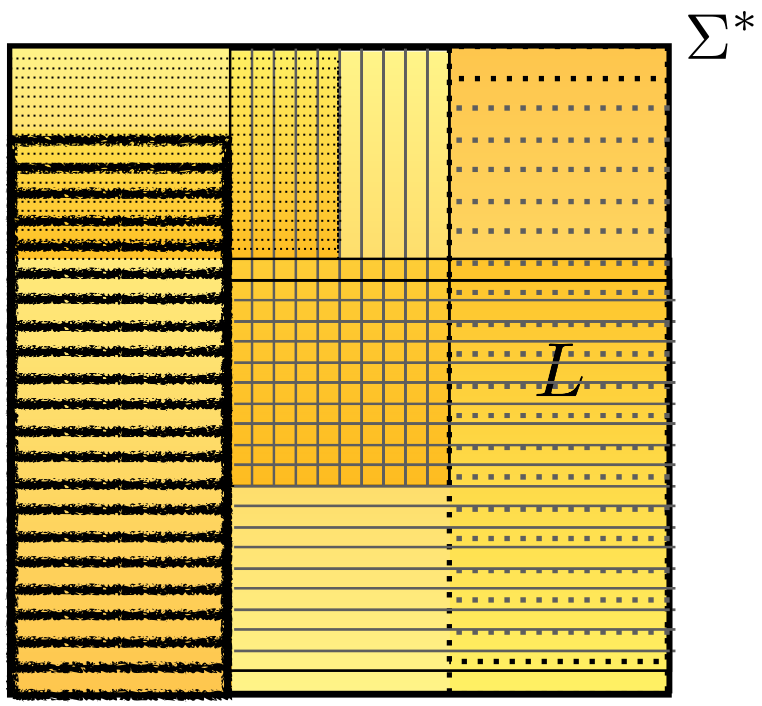


$\stackrel{?}{=}$

Automata-based

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

$$u \preccurlyeq^r_{\mathcal{N}} v \Leftrightarrow \mathrm{post}^{\mathcal{N}}_u(I) \subseteq \mathrm{post}^{\mathcal{N}}_v(I)$$



Lemma:

$$\preccurlyeq^r_L = \preccurlyeq^r_{\mathcal{N}} \text{ iff } u^{-1}L \subseteq v^{-1}L \Leftrightarrow \mathrm{post}^{\mathcal{N}}_u(I) \subseteq \mathrm{post}^{\mathcal{N}}_v(I)$$

Instances of left quasiorders

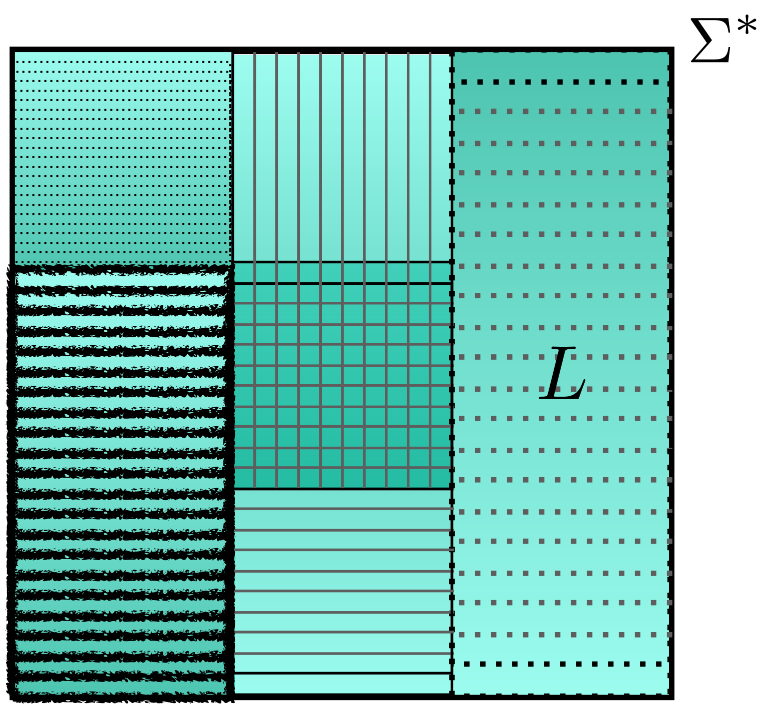
- $\preccurlyeq^{\ell}$ is a **finite** left quasiorder
- $\mathrm{cl}_{\preccurlyeq^{\ell}}(L) = L$

[Valero et. al, MFCS 2020]

Language-based [de Luca and Varricchio, 1994]

Given a regular language L

$$u \preccurlyeq^{\ell}_{L} v \Leftrightarrow Lu^{-1} \subseteq Lv^{-1}$$



The canonical co-RFA for L :

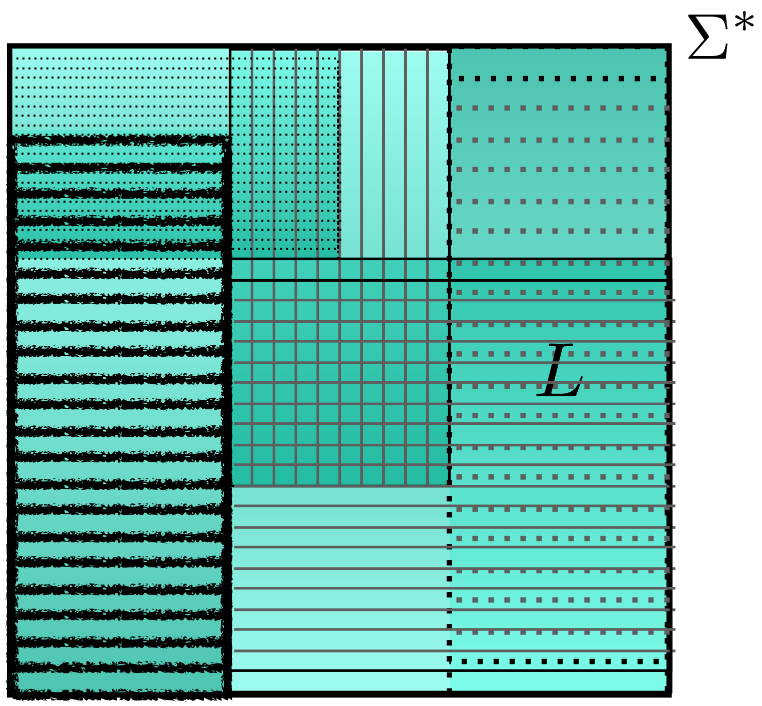
$\mathsf{Can}^{\ell}(L)$

$$\succeq$$

Automata-based

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

$$u \preccurlyeq^{\ell}_{\mathcal{N}} v \Leftrightarrow \mathrm{pre}^{\mathcal{N}}_{u}(F) \subseteq \mathrm{pre}^{\mathcal{N}}_{v}(F)$$



A co-RFA for L:

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

Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

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

Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

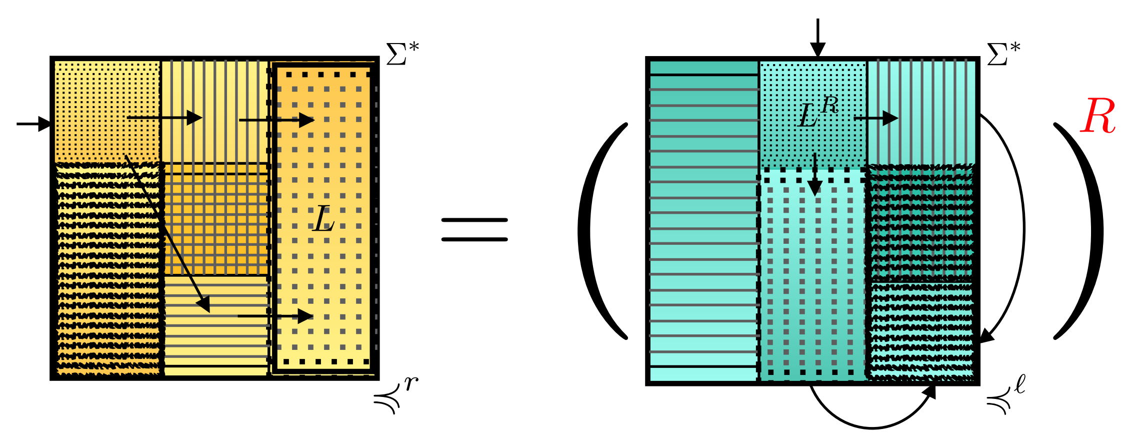
$$\begin{tikzcd}
\mathcal{N} \arrow[r, "\mathsf{Res}^{\ell}"] \arrow[d, leftrightarrow, "R"'] & \mathsf{Res}^{\ell}(\mathcal{N}) \\
\mathcal{N}^R \arrow[r, "\mathsf{Res}^{r}"] & \mathsf{Res}^{r}(\mathcal{N}^R)
\end{tikzcd}$$

Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

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

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

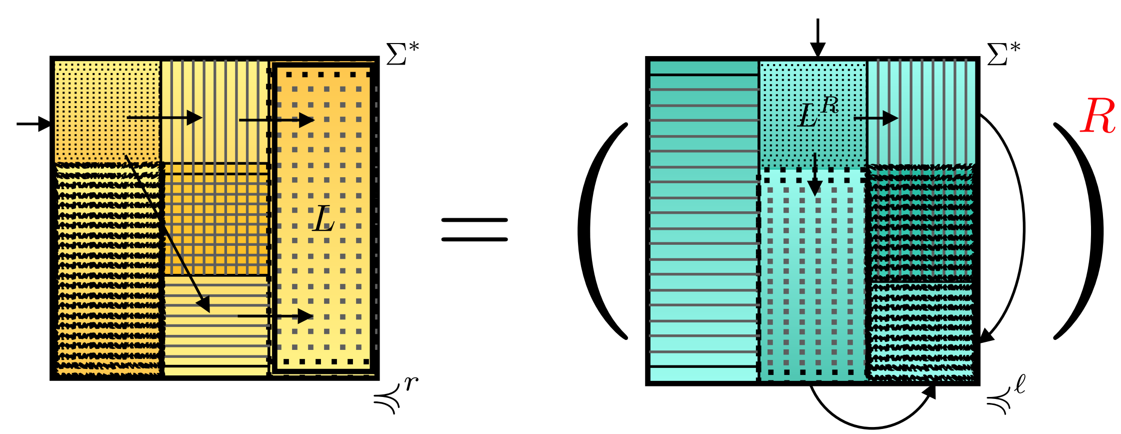


Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

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

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



Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

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

Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

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

Lemma:

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

$$\preccurlyeq^r_L = \preccurlyeq^r_{\mathcal{N}} \ \text{ iff } u^{-1}L \subseteq v^{-1}L \Leftrightarrow \mathrm{post}^{\mathcal{N}}_u(I) \subseteq \mathrm{post}^{\mathcal{N}}_v(I)$$

Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

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

Lemma:

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

$$\preccurlyeq^r_L = \preccurlyeq^r_{\mathcal{N}} \ \text{ iff } u^{-1}L \subseteq v^{-1}L \Leftrightarrow \mathrm{post}^{\mathcal{N}}_u(I) \subseteq \mathrm{post}^{\mathcal{N}}_v(I)$$

Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

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

Lemma:

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

$$\preccurlyeq^r_L = \preccurlyeq^r_{\mathcal{N}} \text{ iff } u^{-1}L \subseteq v^{-1}L \Leftrightarrow \mathrm{post}^{\mathcal{N}}_u(I) \subseteq \mathrm{post}^{\mathcal{N}}_v(I)$$

Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

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

<u>Lemma:</u>

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

$$\preccurlyeq^r_L = \preccurlyeq^r_{\mathcal{N}} \ \text{ iff } u^{-1}L \subseteq v^{-1}L \Leftrightarrow \mathrm{post}^{\mathcal{N}}_u(I) \subseteq \mathrm{post}^{\mathcal{N}}_v(I)$$

Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

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

Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

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

Double-reversal Method for the canonical RFA

[Valero et. al, MFCS 2020]

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

Thm (Double-reversal): Let $\mathcal{N}$ be an NFA. Then $\mathsf{Res}^{r}((\mathsf{Res}(\mathcal{N}^{R}))^{R})$ is isomorphic to the canonical RFA for $\mathcal{L}(\mathcal{N})$

Contributions of This Work

[Valero et. al, MFCS 2020]

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

[Tamm, 2015] **Generalization** of the Double-reversal Method	
[Denis et al., 2002] Double-reversal Method	$\mathrm{Res}^r(\mathrm{Res}^\ell(\mathcal{N})) \equiv$ Canonical RFA for L
[Bollig, 2009] NL* algorithm	

Contributions of This Work

[Valero et. al, MFCS 2020]

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

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

[Tamm, 2015] **Generalization** of the Double-reversal Method	$\mathsf{Res}^r(\mathcal{N}) \equiv$ Canonical RFA for L iff $\forall q : \mathrm{cl}_{\preccurlyeq^r_L}(L_q) = L_q$
[Denis et al., 2002] Double-reversal Method	$\mathsf{Res}^r(\mathsf{Res}^\ell(\mathcal{N})) \equiv$ Canonical RFA for L
[Bollig, 2009] NL* algorithm	

Contributions of This Work

[Valero et. al, MFCS 2020]

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

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

[Tamm, 2015] **Generalization** of the Double-reversal Method	$\text{Res}^r(\mathcal{N}) \equiv$ Canonical RFA for L iff $\forall q : \ \mathrm{cl}_{\preccurlyeq^r_L}(L_q) = L_q$
[Denis et al., 2002] Double-reversal Method	$\text{Res}^r(\text{Res}^\ell(\mathcal{N})) \equiv$ Canonical RFA for L
[Bollig, 2009] NL* algorithm	$\mathcal{S}$: suffix-closed **finite** set of Σ^* $u \ \preccurlyeq^r_{\mathrm{NL}*} \ v \overset{\text{def}}{\Leftrightarrow} u^{-1}L \subseteq_{\mathcal{S}} v^{-1}L$

From DFAs to RFAs

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

	Quasiorders	Congruences **[Valero et.al, 2019]**
[Tamm, 2015] **Generalization** of the Double-reversal Method	$\text{Res}^r(\mathcal{N}) \equiv$ Canonical RFA for L iff $\forall q: \ \mathrm{cl}_{\preccurlyeq^r_L}(L_q) = L_q$	$\text{Det}^r(\mathcal{N}) \equiv$ Minimal DFA for L iff $\forall q: \mathrm{cl}_{\sim^r_L}(L_q) = L_q$
[Denis et al., 2002] Double-reversal Method	$\text{Res}^r(\text{Res}^\ell(\mathcal{N})) \equiv$ Canonical RFA for L	$\text{Det}^r(\text{Det}^\ell(\mathcal{N})) \equiv$ Minimal DFA for L

- Quasiorders are for RFAs as congruences are for DFAs

From DFAs to RFAs

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

	Quasiorders	Congruences **[Valero et.al, 2019]**
[Tamm, 2015] **Generalization** of the Double-reversal Method	$\mathsf{Res}^r(\mathcal{N}) \equiv$ Canonical RFA for L iff $\forall q : \mathrm{cl}_{\preccurlyeq^r_L}(L_q) = L_q$	$\mathsf{Det}^r(\mathcal{N}) \equiv$ Minimal DFA for L iff $\forall q : \mathrm{cl}_{\sim^r_L}(L_q) = L_q$
[Denis et al., 2002] Double-reversal Method	$\mathsf{Res}^r(\mathsf{Res}^\ell(\mathcal{N})) \equiv$ Canonical RFA for L	$\mathsf{Det}^r(\mathsf{Det}^\ell(\mathcal{N})) \equiv$ Minimal DFA for L

- Quasiorders are for RFAs as congruences are for DFAs

Questions?