Nominal G-Automata

Steven Bronsveld

Supervisor: **dr. J.C. Rot**

Radboud University

Part I

Automata Theory in Nominal Sets (2012)

Mikołaj Bojańczyk, Bartek Klin and Sławomir Lasota

Logical Methods in Computer Science,
August 15, 2014, Volume 10, Issue 3
doi: 10.2168/LMCS-10(3:4)2014

Part II

Residual Nominal Automata (2020)

Joshua Moerman and Matteo Sammartino

31st International Conference on
Concurrency Theory (CONCUR 2020)
doi: 10.4230/LIPIcs.CONCUR.2020.44

Part I

Defining nominal automata

Motivation

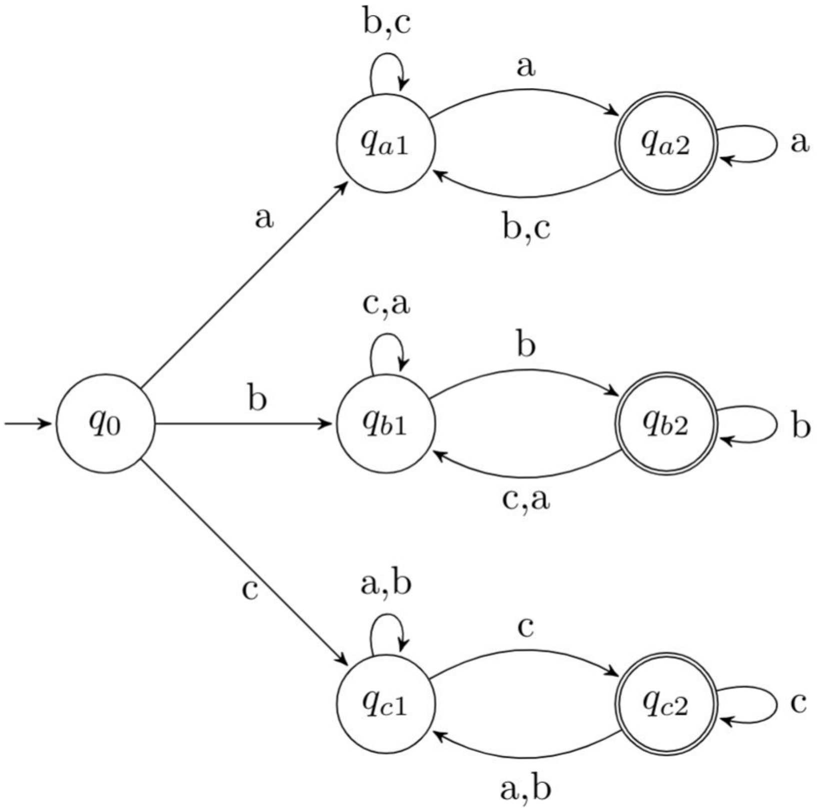
$$A := \{a, b, c\}$$

$$\mathcal{L} := \{\ xwx \ \mid\ x \in A, w \in A^*\ \}$$

Motivation

$A := \{a, b, c\}$

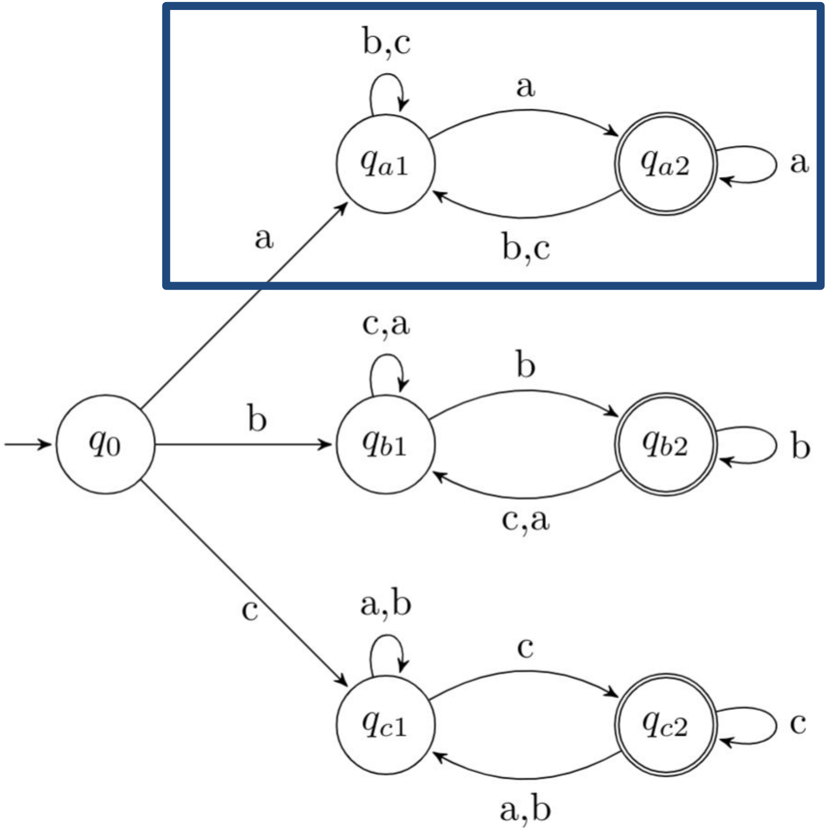
$\mathcal{L} := \{\ xwx \mid x \in A, w \in A^*\ \}$



Motivation

$$A := \{a, b, c\}$$

$$\mathcal{L} := \{\ xwx \ \mid \ x \in A, w \in A^*\ \}$$

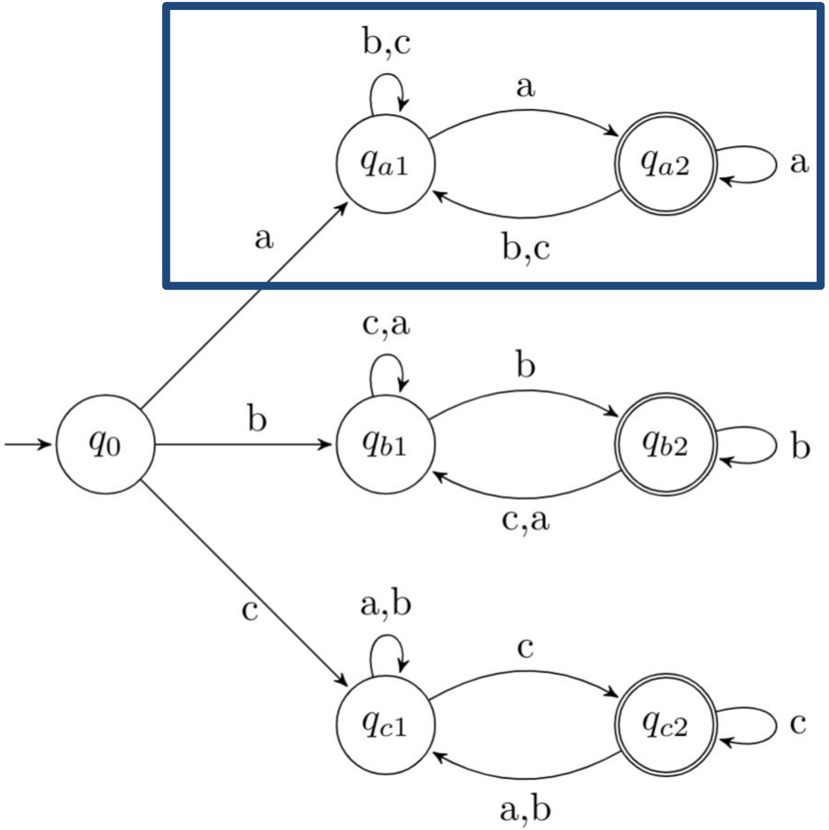


Motivation

$$A := \{a, b, c\}$$

$$\mathcal{L} := \{\ xwx \mid x \in A, w \in A^*\ \}$$

$$\sigma := \begin{cases} a \mapsto b \\ b \mapsto c \\ c \mapsto a \end{cases}$$

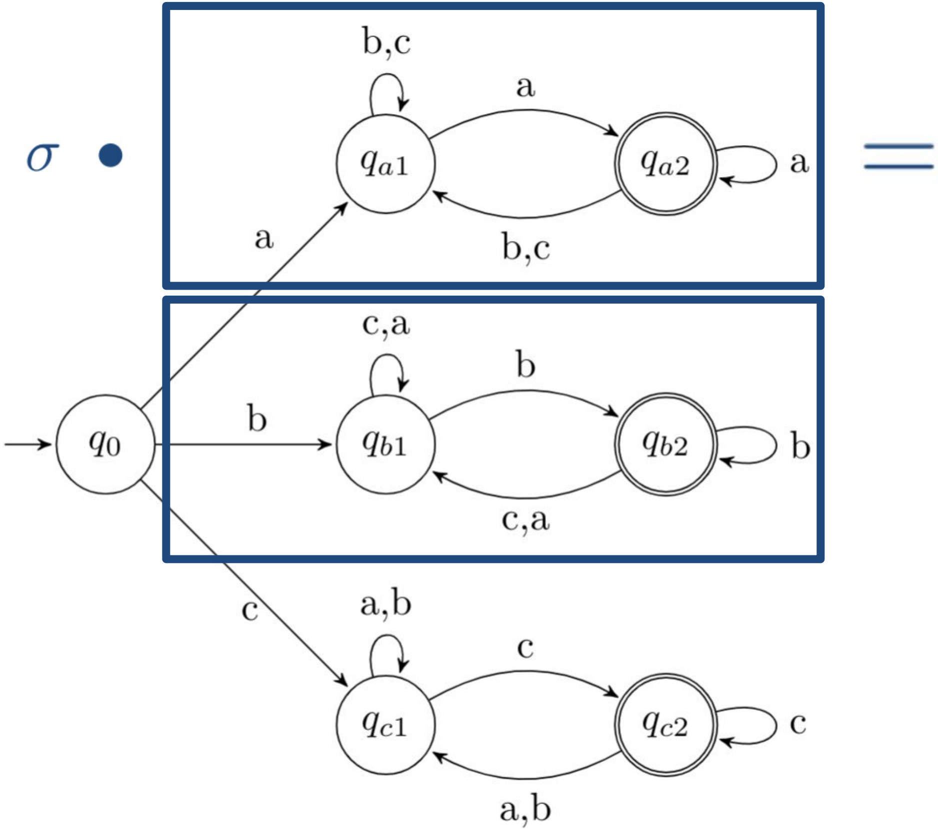


Motivation

$$A := \{a, b, c\}$$

$$\mathcal{L} := \{\ xwx \ \mid \ x \in A, w \in A^* \ \}$$

$$\sigma := \begin{cases} a \mapsto b \\ b \mapsto c \\ c \mapsto a \end{cases}$$

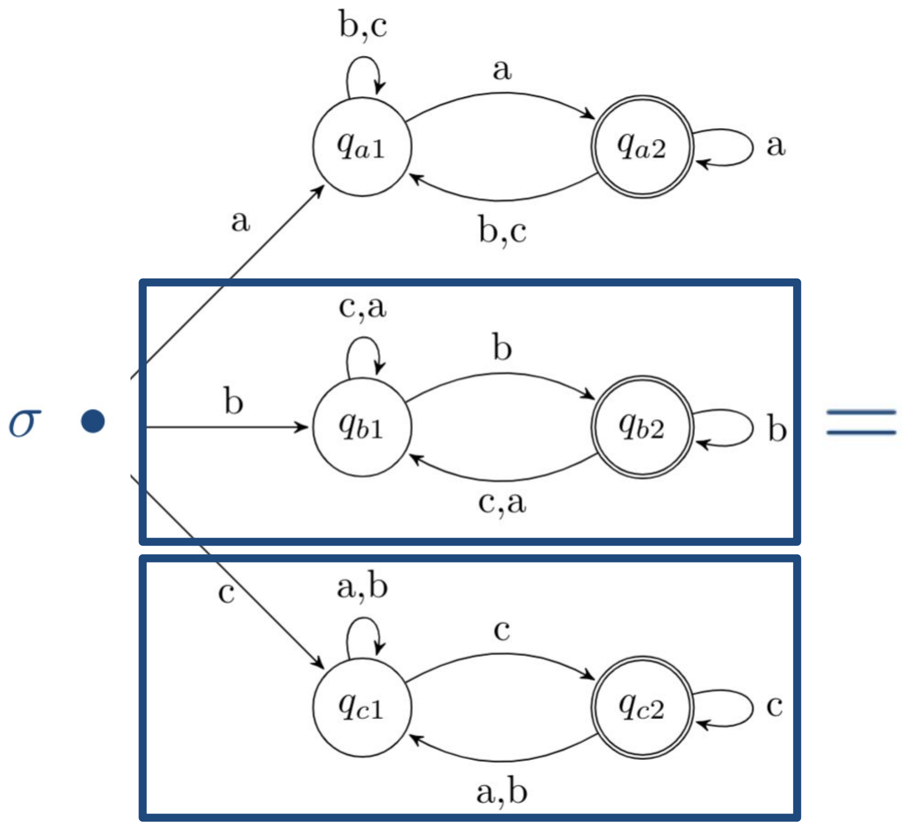


Motivation

$$A := \{a, b, c\}$$

$$\mathcal{L} := \{\ xwx \mid x \in A, w \in A^*\ \}$$

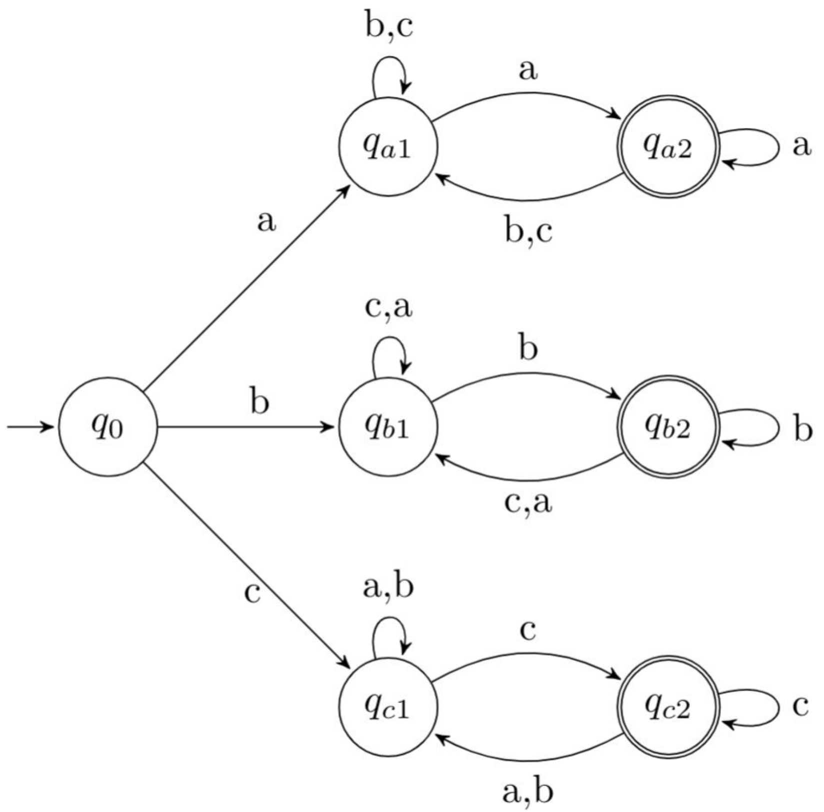
$$\sigma := \begin{cases} a \mapsto b \\ b \mapsto c \\ c \mapsto a \end{cases}$$



Motivation

$A := \{a, b, c\}$

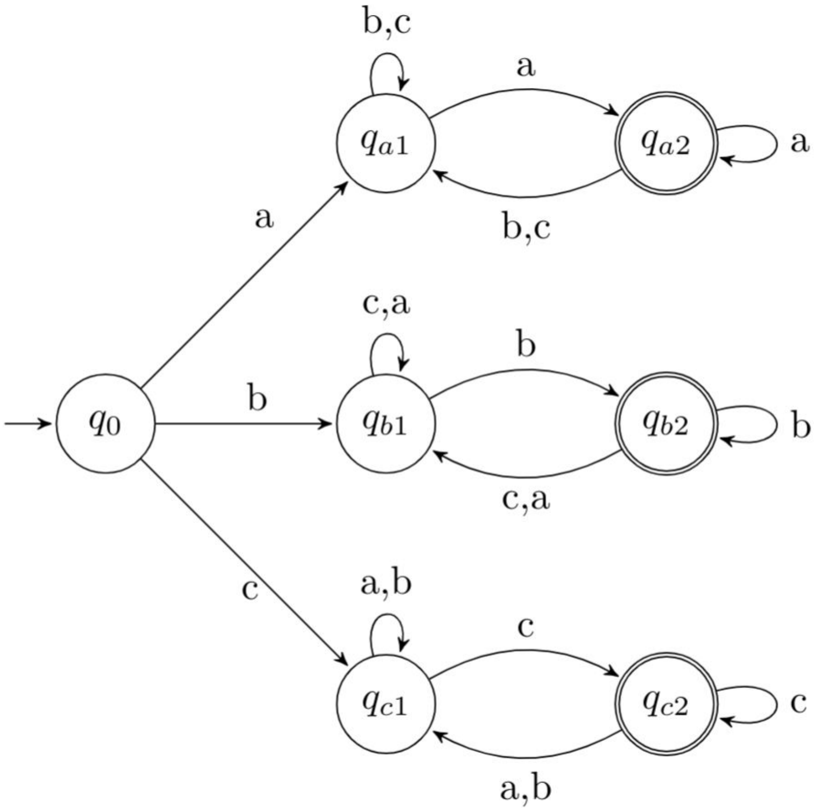
$\mathcal{L} := \{\ xwx \ \mid \ x \in A, w \in A^*\ \}$



Motivation

$$\mathbb{A} := \{a_0, a_1, a_2, \dots\}$$

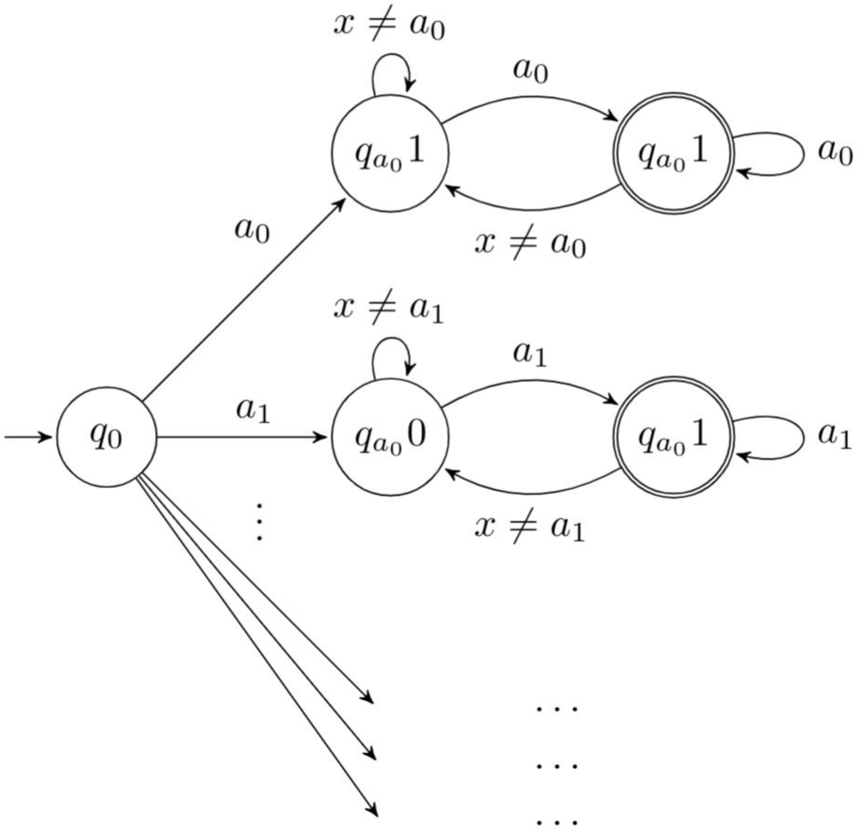
$$\mathcal{L} := \{\ xwx \ \mid \ x \in \mathbb{A}, w \in \mathbb{A}^*\ \}$$



Motivation

$\mathbb{A} := \{a_0, a_1, a_2, \dots\}$

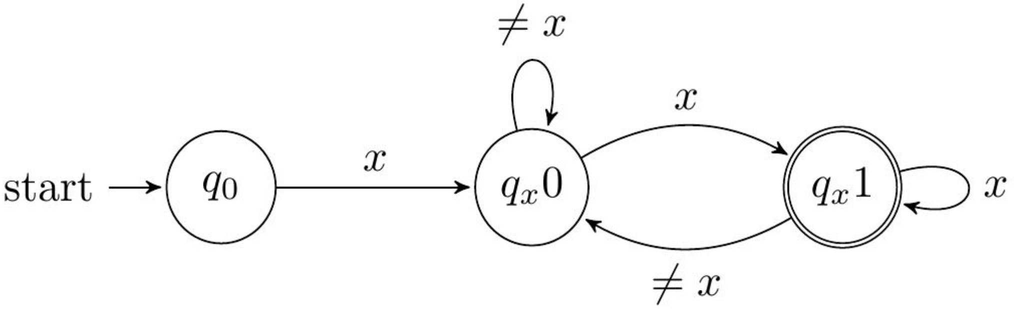
$\mathcal{L} := \{\ xwx \mid x \in \mathbb{A}, w \in \mathbb{A}^*\ \}$



Motivation

$\mathbb{A} := \{a_0, a_1, a_2, \dots\}$

$\mathcal{L} := \{\ xwx \ \mid\ x \in \mathbb{A}, w \in \mathbb{A}^*\ \}$

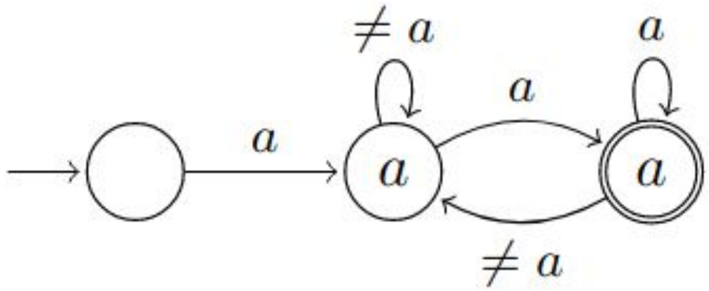


Motivation

$$\mathbb{A} := \{a_0, a_1, a_2, \dots\}$$

$$\mathcal{L} := \{\ xwx \ \mid\ x \in \mathbb{A}, w \in \mathbb{A}^*\ \}$$

$\mathcal{A}_\mathrm{d} =$



Formal definition

Group Actions

Group G, set X

$$\bullet : G \times X \to X$$

$$e \bullet x = x \qquad \forall x \in X$$

$$(\pi \cdot \sigma) \bullet x = \pi \bullet (\sigma \bullet x) \qquad \forall x \in X\ \forall \pi, \sigma \in G$$

Group Actions (G-Sets)

$$G := S(\mathbb{A}), \qquad X := \mathbb{A} = \{a_0, a_1, a_2, \ldots\}$$

$$\bullet : S(\mathbb{A}) \times \mathbb{A} \to \mathbb{A}$$

$$e \bullet x = x \qquad \forall x \in \mathbb{A}$$

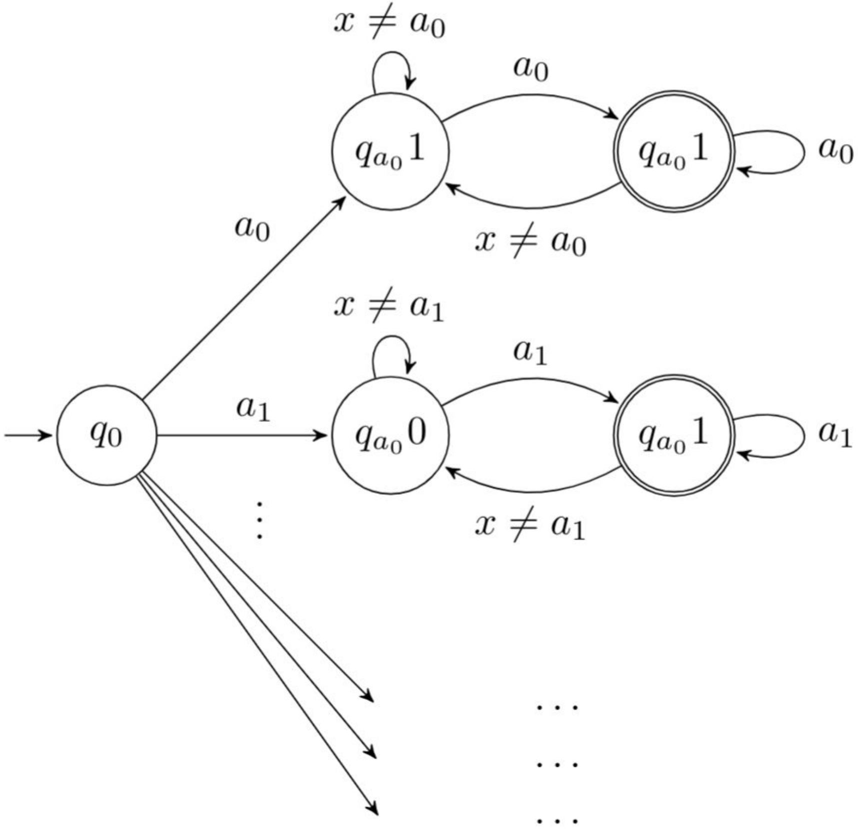
$$(\pi \cdot \sigma) \bullet x = \pi \bullet (\sigma \bullet x) \qquad \forall x \in \mathbb{A}\ \forall \pi, \sigma \in S(\mathbb{A})$$

Orbits

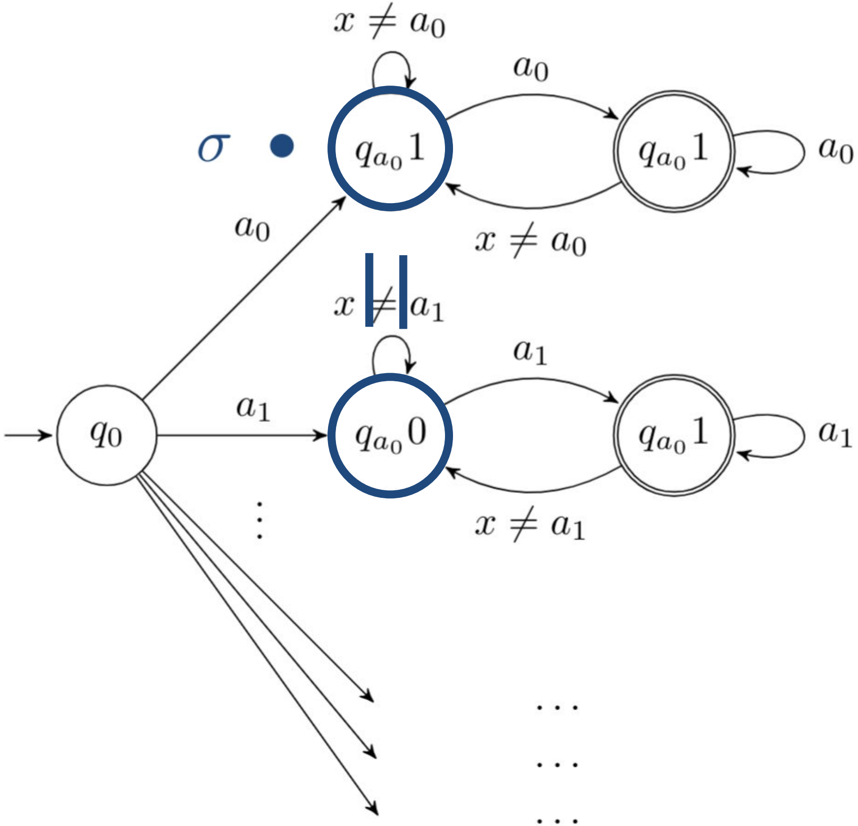
$$\text{Group } G, \quad \text{G-Set } X, \quad x \in X$$

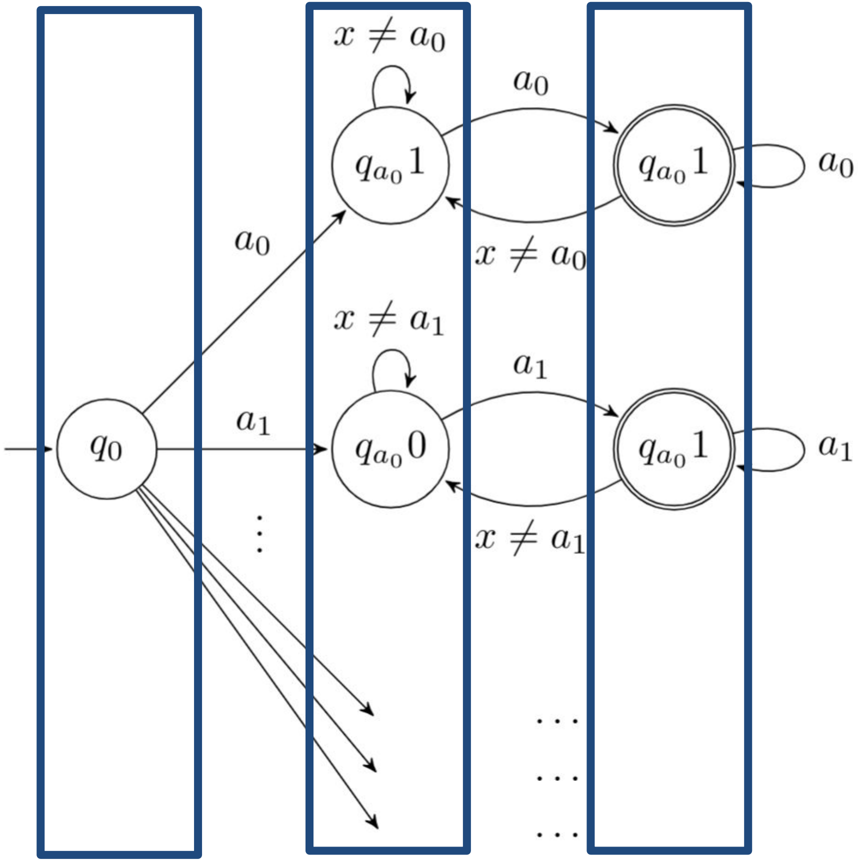
$$G \bullet x := \{ \sigma \bullet x \mid \sigma \in G \}$$

Lemma any G-set is partitioned into orbits in a unique way



$x \neq a_0$
a_0
$q_{a_0}1$
$q_{a_0}1$
a_0
a_0
$x \neq a_0$
$x \neq a_1$
a_1
q_0
a_1
$q_{a_0}0$
$q_{a_0}1$
a_1
$x \neq a_1$


$x \neq a_0$
a_0
σ
$q_{a_0}1$
$q_{a_0}1$
a_0
a_0
$x \neq a_0$
$x \neq a_1$
a_1
q_0
a_1
$q_{a_0}0$
$q_{a_0}1$
a_1
$x \neq a_1$
...
...
...



$x \neq a_0$
a_0
$q_{a_0}1$
$q_{a_0}1$
a_0
a_0
$x \neq a_0$
$x \neq a_1$
a_1
q_0
a_1
$q_{a_0}0$
$q_{a_0}1$
a_1
$x \neq a_1$

Equivariance

A set $A \subseteq X$ is equivariant if for all $\sigma \in G$ we have $\sigma \bullet A := \{ \sigma \bullet x \mid x \in A \} = A$.

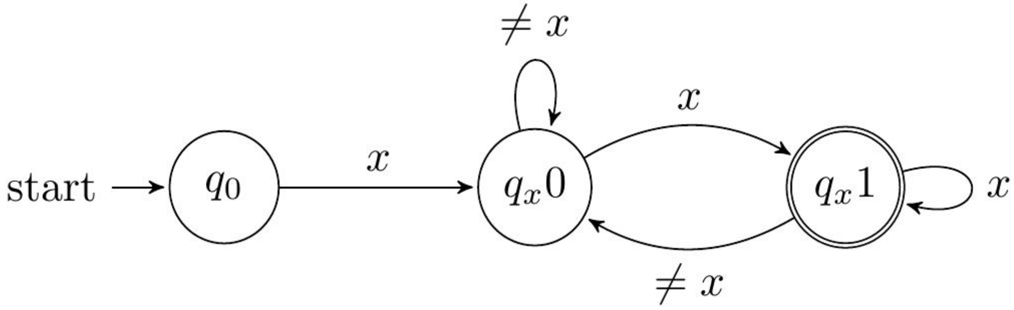
Equivariant relation

A relation $R \subseteq X \times X$ is equivariant if for all $\sigma \in G$ we have

$$xRy \iff (\sigma \bullet x)R(\sigma \bullet y)$$

G-Automata

$(Q, \mathbb{A}, I, F, \delta)$
$\mathbb{A}, Q$ are **orbit finite G-Sets**
I, F, δ are **equivariant**



G-Language

$\mathcal{L} \subseteq \mathbb{A}^*$ is **equivariant**

Nominal G-Automata

Finite support

$A \subseteq \mathbb{A}$ *supports* $x \in X$ if
$$\sigma \mid_A = Id_A \implies \sigma \bullet x = x \qquad \forall \sigma \in S(\mathbb{A})$$

Nominal G-Set

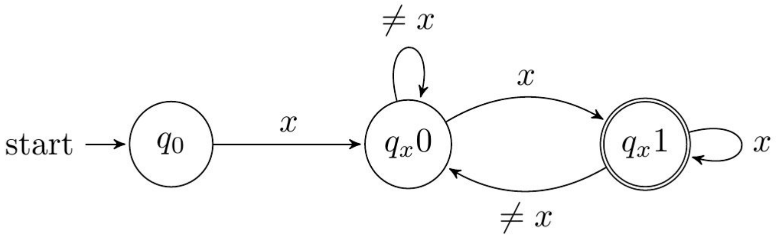
Every element $x \in X$ has a finite support $supp(x)$

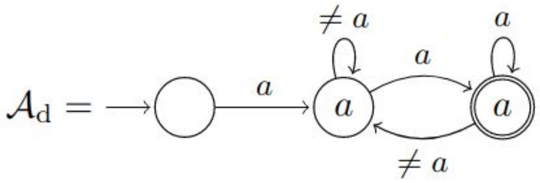
Nominal G-Automata

Q and $\mathbb{A}$ are nominal G-Sets.

Nominal G-Automata

- **Infinite** number of states
- Some kind of symmetry
- But **finitely** many orbits
- Initial and final states equivariant
- Transition function equivariant
- **We can represent them finitely**
- Same expressiveness as *register automata*





Derivative languages

Derivative languages

$$u^{-1}\mathcal{L} := \{\, w \;\mid\; uw \in \mathcal{L} \,\}$$

Derivative languages

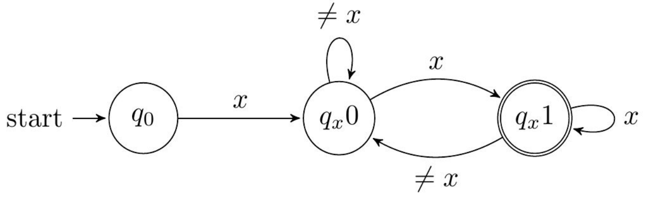
$$u^{-1}\mathcal{L} := \{ \, w \mid uw \in \mathcal{L} \, \}$$

$$\epsilon^{-1}\mathcal{L} = \mathcal{L}$$

Derivative languages

$$u^{-1}\mathcal{L} := \{\ w \mid uw \in \mathcal{L}\ \}$$
$$\epsilon^{-1}\mathcal{L} = \mathcal{L}$$

$$\mathbb{A} := \{a, b, c, \ldots\}$$
$$\mathcal{L} := \{\ xwx \mid x \in \mathbb{A}, w \in \mathbb{A}^*\ \}$$



Derivative languages

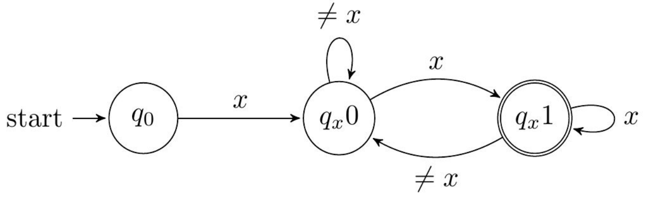
$$u^{-1}\mathcal{L} := \{\ w \mid uw \in \mathcal{L}\ \}$$

$$\epsilon^{-1}\mathcal{L} = \mathcal{L}$$

$$a^{-1}\mathcal{L} = \{\ w \mid aw \in \mathcal{L}\ \}$$

$$\mathbb{A} := \{a, b, c, \dots\}$$

$$\mathcal{L} := \{\ xwx \mid x \in \mathbb{A}, w \in \mathbb{A}^*\ \}$$



Derivative languages

$$u^{-1}\mathcal{L} := \{\ w \mid uw \in \mathcal{L}\ \}$$

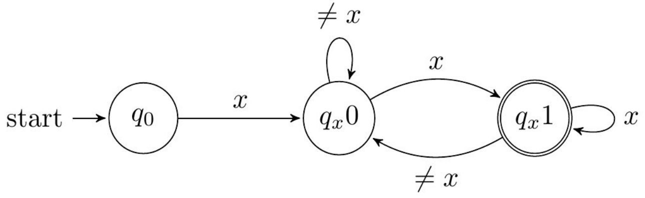
$$\epsilon^{-1}\mathcal{L} = \mathcal{L}$$

$$a^{-1}\mathcal{L} = \{\ w \mid aw \in \mathcal{L}\ \}$$

$$= \{a, aa, ba, ca, \ldots\}$$

$$\mathbb{A} := \{a, b, c, \ldots\}$$

$$\mathcal{L} := \{\ xwx \mid x \in \mathbb{A}, w \in \mathbb{A}^*\ \}$$



Derivative languages

$$u^{-1}\mathcal{L} := \{\ w \mid uw \in \mathcal{L}\ \}$$

$$\epsilon^{-1}\mathcal{L} = \mathcal{L}$$

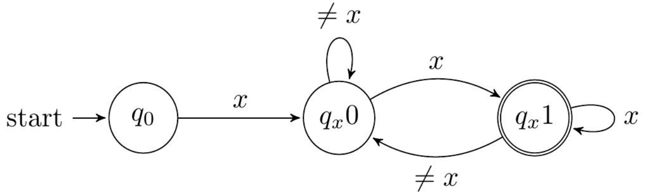
$$a^{-1}\mathcal{L} = \{\ w \mid aw \in \mathcal{L}\ \}$$

$$= \{a, aa, ba, ca, \ldots\}$$

$$(aa)^{-1}\mathcal{L} = \{\epsilon, a, ba, \ldots\}$$

$$\mathbb{A} := \{a, b, c, \ldots\}$$

$$\mathcal{L} := \{\ xwx \mid x \in \mathbb{A}, w \in \mathbb{A}^*\ \}$$



Derivative languages

$$u^{-1}\mathcal{L} := \{\ w \mid uw \in \mathcal{L}\ \}$$

$$\epsilon^{-1}\mathcal{L} = \mathcal{L}$$

$$a^{-1}\mathcal{L} = \{\ w \mid aw \in \mathcal{L}\ \}$$

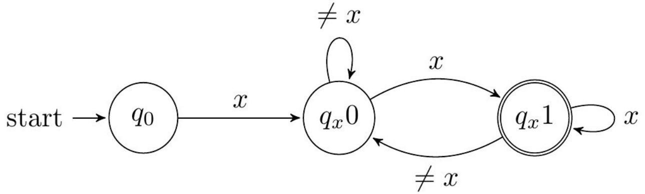
$$= \{a, aa, ba, ca, \ldots\}$$

$$(aa)^{-1}\mathcal{L} = \{\epsilon, a, ba, \ldots\}$$

$$Der(\mathcal{L}) := \{\ x^{-1}\mathcal{L} \mid x \in \mathbb{A}^*\ \}$$

$$\mathbb{A} := \{a, b, c, \ldots\}$$

$$\mathcal{L} := \{\ xwx \mid x \in \mathbb{A}, w \in \mathbb{A}^*\ \}$$



Derivative languages

$$u^{-1}\mathcal{L} := \{\ w \mid uw \in \mathcal{L}\ \}$$

$$\epsilon^{-1}\mathcal{L} = \mathcal{L}$$

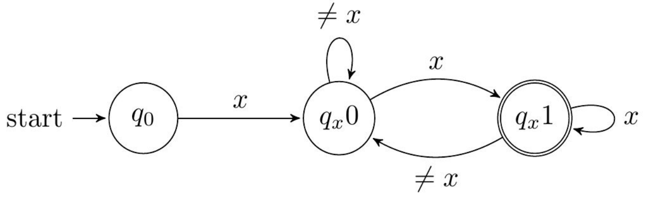
$$a^{-1}\mathcal{L} = \{\ w \mid aw \in \mathcal{L}\ \}$$

$$= \{a, aa, ba, ca, \dots\}$$

$$(aa)^{-1}\mathcal{L} = \{\epsilon, a, ba, \dots\}$$

$$\mathbb{A} := \{a, b, c, \dots\}$$

$$\mathcal{L} := \{\ xwx \mid x \in \mathbb{A}, w \in \mathbb{A}^*\ \}$$



$$Der(\mathcal{L}) := \{\ x^{-1}\mathcal{L} \mid x \in \mathbb{A}^*\ \}$$

$$= \{\ \mathcal{L},\ x^{-1}\mathcal{L},\ (xx)^{-1}\mathcal{L} \mid x \in \mathbb{A}\ \}$$

What are the orbits

$$\begin{aligned} Der(\mathcal{L}) &:= \{\ x^{-1}\mathcal{L} \ \mid\ x \in \mathbb{A}\ \} \\ &= \{\ \mathcal{L},\ \ x^{-1}\mathcal{L},\ \ (xx)^{-1}\mathcal{L}\ \mid\ x \in \mathbb{A}\ \} \end{aligned}$$

What are the orbits

$$Der(\mathcal{L}) := \{\ x^{-1}\mathcal{L}\ \mid\ x \in \mathbb{A}\ \}$$
$$= \{\ \mathcal{L},\ \ x^{-1}\mathcal{L},\ \ (xx)^{-1}\mathcal{L}\ \mid\ x \in \mathbb{A}\ \}$$

$$\sigma := \begin{cases} a \mapsto b \\ \dots \end{cases}$$

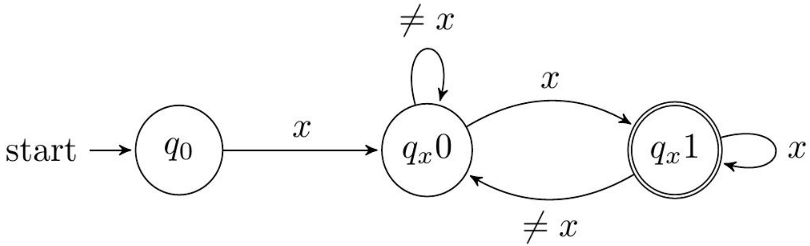
$$\sigma \bullet (aa)^{-1}\mathcal{L} = (bb)^{-1}\mathcal{L}$$

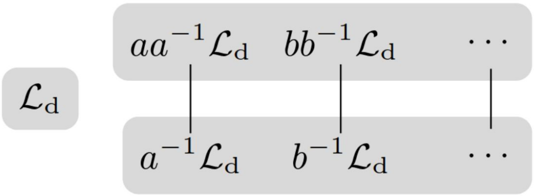
What are the orbits?

$$Der(\mathcal{L}) := \{\, x^{-1}\mathcal{L} \mid x \in \mathbb{A} \,\}$$
$$= \{\, \mathcal{L},\ x^{-1}\mathcal{L},\ (xx)^{-1}\mathcal{L} \mid x \in \mathbb{A} \,\}$$

$$\sigma := \begin{cases} a \mapsto b \\ \ldots \end{cases}$$

$$\sigma \bullet (aa)^{-1}\mathcal{L} = (bb)^{-1}\mathcal{L}$$





Myhill-Nerode Theorem

Myhill-Nerode Theorem (DFA)

Finite alphabet A

Language $\mathcal{L} \subseteq A^*$

$\mathcal{L}$ is recognized by a deterministic finite automata

$$\Longleftrightarrow$$

$Der(\mathcal{L})$ is finite

Myhill-Nerode Theorem (G-Automata)

Orbit finite G-Set $\mathbb{A}$

G-Language $\mathcal{L} \subseteq \mathbb{A}^*$

$\mathcal{L}$ is recognized by a **deterministic G-**Automaton

$$\iff$$

$Der(\mathcal{L})$ is **orbit** finite

Myhill-Nerode Theorem (Nominal G-Automata)

Orbit finite nominal G-Set $\mathbb{A}$

G-Language $\mathcal{L} \subseteq \mathbb{A}^*$

$\mathcal{L}$ is recognized by a **deterministic nominal G-**Automaton

$$\Longleftrightarrow$$

$Der(\mathcal{L})$ is **orbit** finite

Myhill-Nerode Theorem (DFA)

$\mathcal{L}$ is recognized by a deterministic finite automata

$\iff$

$|\, Der(\mathcal{L}) \,|$ is finite

Proof (⇒)

Automaton $D = (Q, A, \{q_0\}, F, \delta)$ such that $\mathcal{L}(D) = \mathcal{L}$

Myhill-Nerode Theorem (DFA)

$\mathcal{L}$ is recognized by a deterministic finite automata

$\Longleftrightarrow$

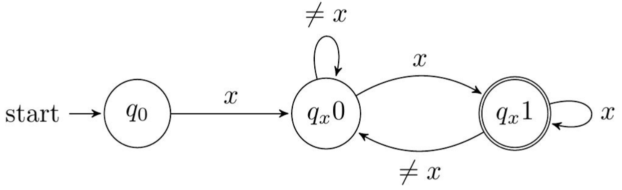
$|\ Der(\mathcal{L})\ |$ is finite

Proof ($\Rightarrow$)

Automaton $D = (Q, A, \{q_0\}, F, \delta)$ such that $\mathcal{L}(D) = \mathcal{L}$

$$\mathcal{L}(D, q) = \{\ w \in A^* \ |\ \delta(q, w) \in F\ \}$$

$$\mathcal{L}(D, q_0) = \mathcal{L}(D)$$



Myhill-Nerode Theorem (DFA)

$\mathcal{L}$ is recognized by a deterministic finite automata

$$\iff$$

$| \, Der(\mathcal{L}) \, |$ is finite

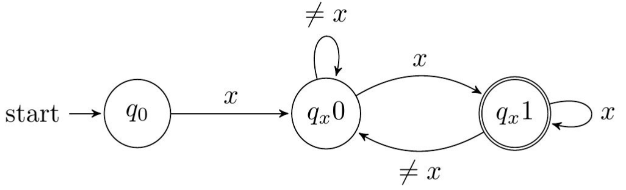
Proof ($\Rightarrow$)

Automaton $D = (Q, A, \{q_0\}, F, \delta)$ such that $\mathcal{L}(D) = \mathcal{L}$

$$C := \{ \, \mathcal{L}(D, q) \mid q \in Q \, \}$$

$$\mathcal{L}(D, q) = \{ \, w \in A^* \mid \delta(q, w) \in F \, \}$$

$$\mathcal{L}(D, q_0) = \mathcal{L}(D)$$



Myhill-Nerode Theorem (DFA)

$\mathcal{L}$ is recognized by a deterministic finite automata

$$\Longleftrightarrow$$

$|\, Der(\mathcal{L}) \,|$ is finite

Proof ($\Rightarrow$)

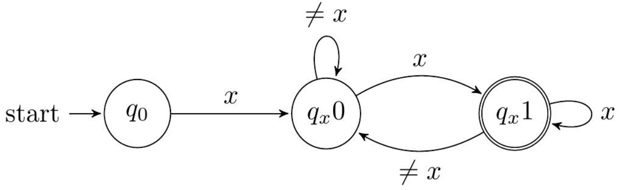
Automaton $D = (Q, A, \{q_0\}, F, \delta)$ such that $\mathcal{L}(D) = \mathcal{L}$

$$C := \{\ \mathcal{L}(D, q) \ \mid\ q \in Q\ \}$$

$$|\, C \,| \ \leq\ |\, Q \,| < \infty$$

$$\mathcal{L}(D, q) = \{\ w \in A^* \ \mid\ \delta(q, w) \in F\ \}$$

$$\mathcal{L}(D, q_0) = \mathcal{L}(D)$$



Myhill-Nerode Theorem (DFA)

$\mathcal{L}$ is recognized by a deterministic finite automata

$$\iff$$

$| \, Der(\mathcal{L}) \, |$ is finite

Proof ($\Rightarrow$)

Automaton $D = (Q, A, \{q_0\}, F, \delta)$ such that $\mathcal{L}(D) = \mathcal{L}$

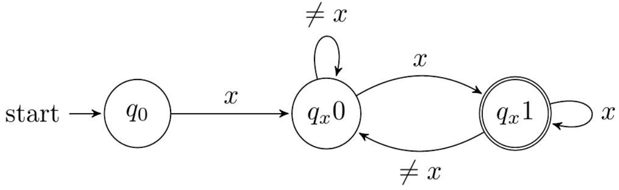
$$C := \{ \, \mathcal{L}(D, q) \ | \ q \in Q \, \}$$

$$| \, C \, | \ \leq \ | \, Q \, | < \infty$$

To prove: $Der(\mathcal{L}) \subseteq C$

$$\mathcal{L}(D, q) = \{ \, w \in A^* \ | \ \delta(q, w) \in F \, \}$$

$$\mathcal{L}(D, q_0) = \mathcal{L}(D)$$



Myhill-Nerode Theorem (DFA)

$\mathcal{L}$ is recognized by a deterministic finite automata

$$\iff$$

$$|\,Der(\mathcal{L})\,| \text{ is finite}$$

Proof (⇒)

Automaton $D = (Q, A, \{q_0\}, F, \delta)$ such that $\mathcal{L}(D) = \mathcal{L}$

$$C := \{\ \mathcal{L}(D, q)\ \mid\ q \in Q\ \}$$

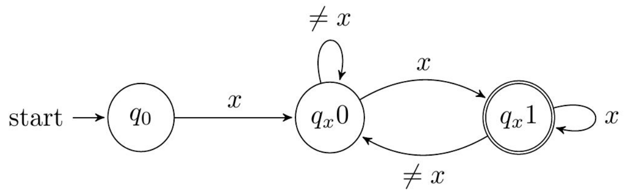
$$|\,C\,| \ \leq\ |\,Q\,| < \infty$$

To prove: $Der(\mathcal{L}) \subseteq C$

$$Der(\mathcal{L}) = \{\ w^{-1}\mathcal{L}\ \mid\ w \in A^*\ \}$$

$$\mathcal{L}(D, q) = \{\ w \in A^*\ \mid\ \delta(q, w) \in F\ \}$$

$$\mathcal{L}(D, q_0) = \mathcal{L}(D)$$



Myhill-Nerode Theorem (DFA)

$\mathcal{L}$ is recognized by a deterministic finite automata

$$\iff$$

$$|\, Der(\mathcal{L}) \,| \text{ is finite}$$

Proof (⇒)

Automaton $D = (Q, A, \{q_0\}, F, \delta)$ such that $\mathcal{L}(D) = \mathcal{L}$

$$C := \{\, \mathcal{L}(D, q) \mid q \in Q \,\}$$

$$|\, C \,| \leq |\, Q \,| < \infty$$

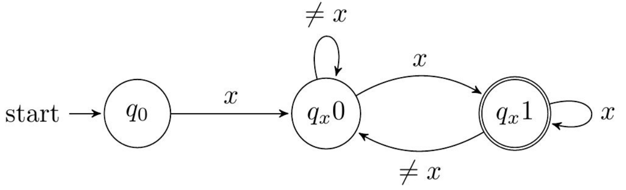
To prove: $Der(\mathcal{L}) \subseteq C$

$$Der(\mathcal{L}) = \{\, w^{-1}\mathcal{L} \mid w \in A^* \,\}$$

$$= \{\, w^{-1}\mathcal{L}(D, q_0) \mid w \in A^* \,\}$$

$$\mathcal{L}(D, q) = \{\, w \in A^* \mid \delta(q, w) \in F \,\}$$

$$\mathcal{L}(D, q_0) = \mathcal{L}(D)$$



Myhill-Nerode Theorem (DFA)

$$\mathcal{L} \text{ is recognized by a deterministic finite automata}$$
$$\iff$$
$$|\, Der(\mathcal{L}) \,| \text{ is finite}$$

Proof (⇒)

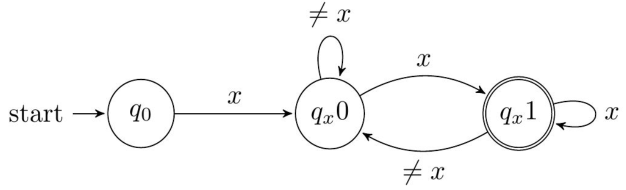
$$\text{Automaton } D = (Q, A, \{q_0\}, F, \delta) \text{ such that } \mathcal{L}(D) = \mathcal{L}$$

$$C := \{\, \mathcal{L}(D, q) \mid q \in Q \,\}$$

$$|\, C \,| \leq |\, Q \,| < \infty$$

$$\mathcal{L}(D, q) = \{\, w \in A^* \mid \delta(q, w) \in F \,\}$$

$$\mathcal{L}(D, q_0) = \mathcal{L}(D)$$



To prove: $Der(\mathcal{L}) \subseteq C$

$$\begin{aligned} Der(\mathcal{L}) &= \{\, w^{-1}\mathcal{L} \mid w \in A^* \,\} \\ &= \{\, w^{-1}\mathcal{L}(D, q_0) \mid w \in A^* \,\} \\ &= \{\, \mathcal{L}(D, \delta(q_0, w)) \mid w \in A^* \,\} \end{aligned}$$

Myhill-Nerode Theorem (DFA)

$\mathcal{L}$ is recognized by a deterministic finite automata

$\iff$

$|\ Der(\mathcal{L})\ |$ is finite

Proof (⇒)

Automaton $D = (Q, A, \{q_0\}, F, \delta)$ such that $\mathcal{L}(D) = \mathcal{L}$

$$C := \{\ \mathcal{L}(D, q)\ \mid\ q \in Q\ \}$$

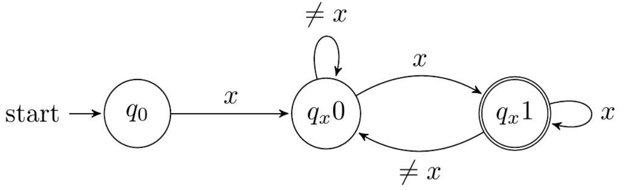
$$|\ C\ |\ \leq\ |\ Q\ | < \infty$$

To prove: $Der(\mathcal{L}) \subseteq C$

$$\begin{aligned} Der(\mathcal{L}) &= \{\ w^{-1}\mathcal{L}\ \mid\ w \in A^*\ \} \\ &= \{\ w^{-1}\mathcal{L}(D, q_0)\ \mid\ w \in A^*\ \} \\ &= \{\ \mathcal{L}(D, \delta(q_0, w))\ \mid\ w \in A^*\ \} \\ &\subseteq \{\ \mathcal{L}(D, q')\ \mid\ q' \in Q\ \} = C \end{aligned}$$

$$\mathcal{L}(D, q) = \{\ w \in A^*\ \mid\ \delta(q, w) \in F\ \}$$

$$\mathcal{L}(D, q_0) = \mathcal{L}(D)$$



Myhill-Nerode Theorem (DFA)

$\mathcal{L}$ is recognized by a deterministic finite automata

$\Longleftrightarrow$

$|\ Der(\mathcal{L})\ |$ is finite

Proof (⇐)

We create a **syntactic automaton**:

$$Q := Der(\mathcal{L})$$

$$I := \epsilon^{-1}\mathcal{L}$$

$$F := \{\ p^{-1}\mathcal{L} \mid \epsilon \in p^{-1}\mathcal{L}\ \}$$

$$\delta(p^{-1}\mathcal{L}, t) := (pt)^{-1}\mathcal{L}$$

Myhill-Nerode Theorem (DFA)

$$\mathcal{L} \text{ is recognized by a deterministic finite automata} \iff |\, Der(\mathcal{L}) \,| \text{ is finite}$$

Proof (⇐)

We create a **syntactic automaton**:

- It is well-defined
- It accepts $\mathcal{L}$

$$Q := Der(\mathcal{L})$$

$$I := \epsilon^{-1}\mathcal{L}$$

$$F := \{\, p^{-1}\mathcal{L} \mid \epsilon \in p^{-1}\mathcal{L} \,\}$$

$$\delta(p^{-1}\mathcal{L}, t) := (pt)^{-1}\mathcal{L}$$

The other theorems generalize straightforwardly

- Does the syntactic automaton still work?
- Is everything orbit finite?

Part II

Residual Nominal Automata
Hierarchy

Determination fails

A **nondeterministic** nominal G-automaton can (in general) not be turned into a **deterministic** nominal G-automaton

Exact learning

Constructing an automaton of a unknown language $\mathcal{L}$

There is an exact learning algorithm **L*** which uses residual languages

Which nominal languages admit an exact learning algorithm?

Residual

If the language of each state is a derivative of $\mathcal{L}$

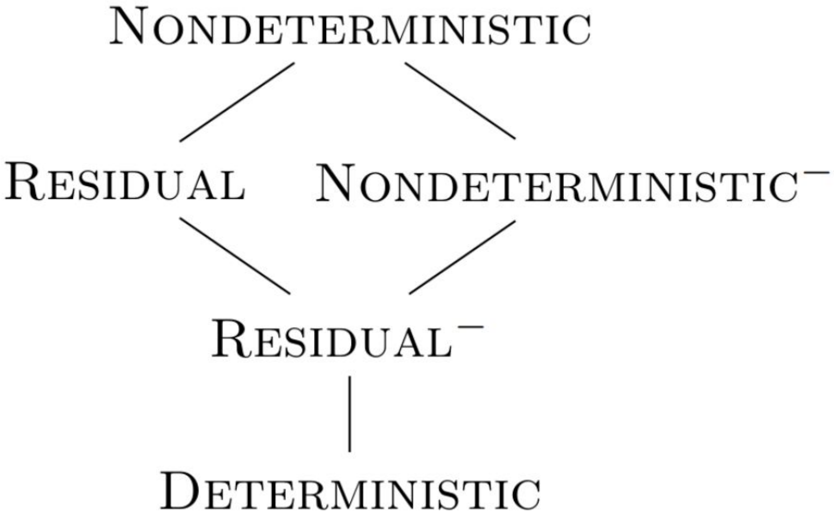
$$\mathcal{L}(A, q) = w^{-1}\mathcal{L} \text{ for some } w \in \mathbb{A}^*$$

Non-Guessing

May **not** store symbols in registers without explicitly reading them

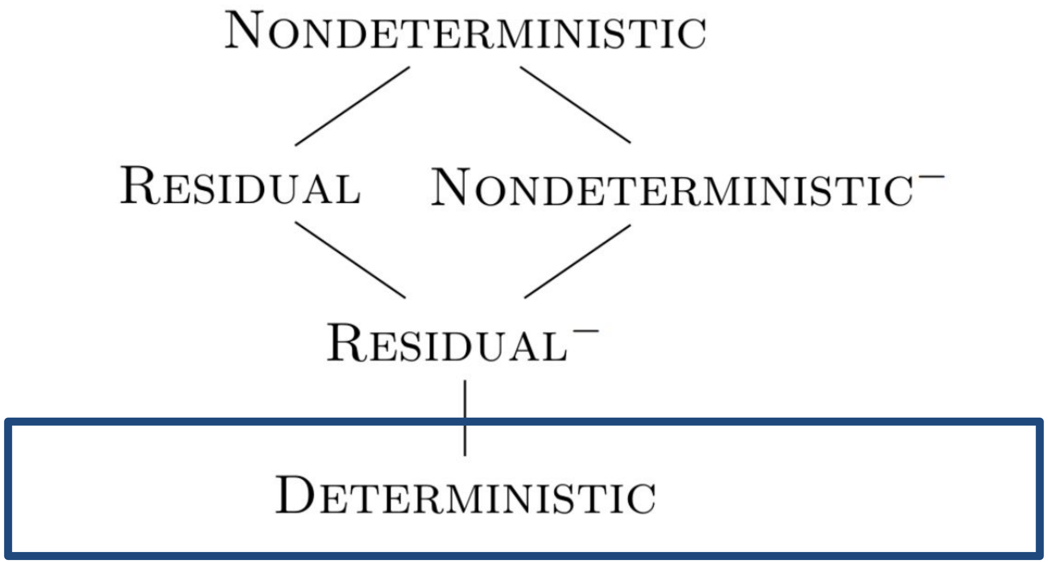
$$\text{if } supp(q_0) = \varnothing \text{ and}$$
$$supp(q') \subseteq supp(q) \cup supp(a)$$
$$\text{for each } (q, a, q') \in \delta$$

Nominal G-Automata



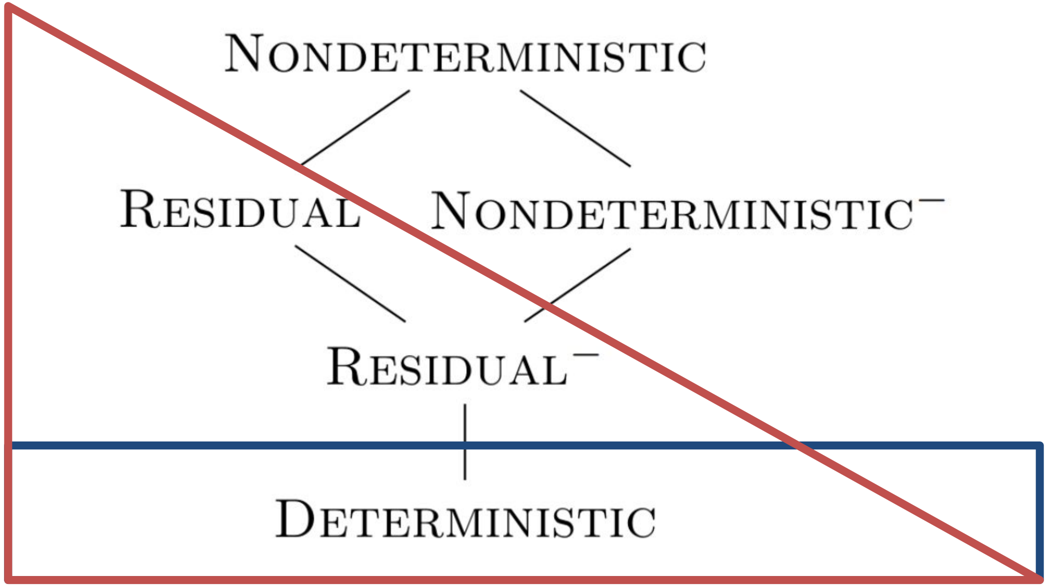
$^{-}$ = non-guessing

Nominal G-Automata



$^{-}$ = non-guessing

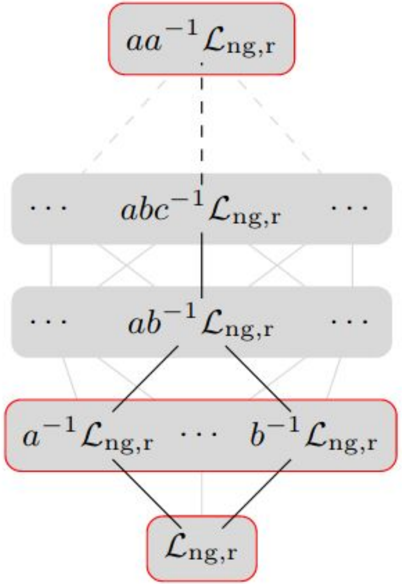
Nominal G-Automata



⁻ = non-guessing

Join-irreducible derivative languages

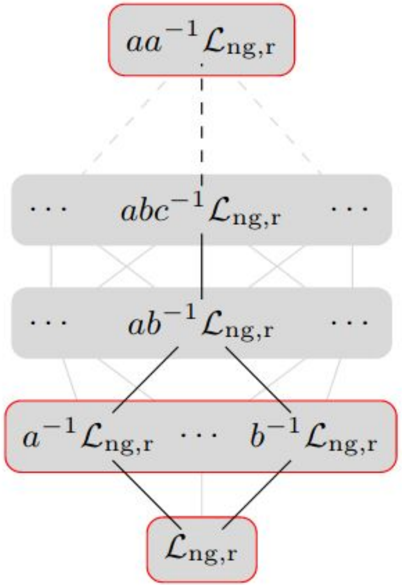
$$\mathcal{L}_{\mathrm{ng,r}} \coloneqq \{uavaw \mid u, v, w \in \mathbb{A}^*, a \in \mathbb{A}\}.$$



Join-irreducible derivative languages

$$\mathcal{L}_{\mathrm{ng,r}} := \{uavaw \mid u, v, w \in \mathbb{A}^*, a \in \mathbb{A}\}.$$

Is this language recognised by a deterministic nominal G-automaton?

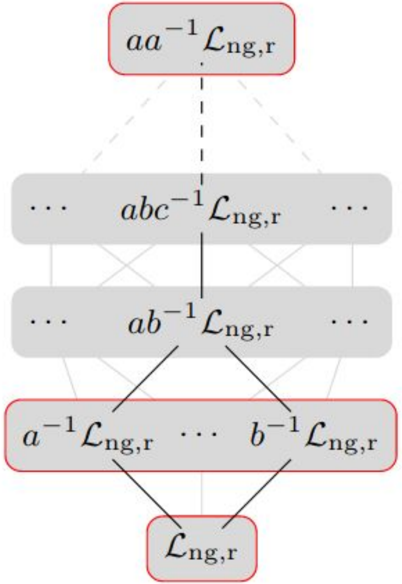


Join-irreducible derivative languages

$$\mathcal{L}_{\mathrm{ng,r}} := \{uavaw \mid u, v, w \in \mathbb{A}^*, a \in \mathbb{A}\}.$$

Is this language recognised by a deterministic nominal G-automaton?

NO



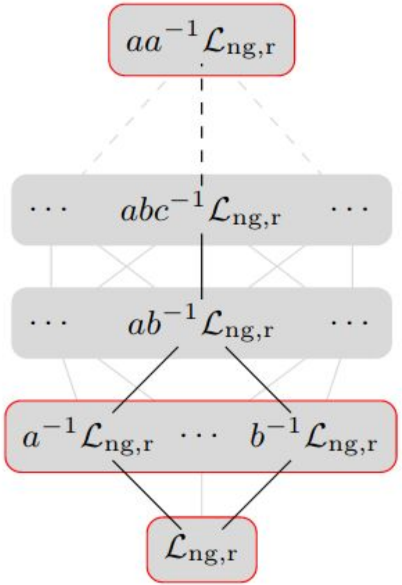
Join-irreducible derivative languages

$$\mathcal{L}_{\mathrm{ng,r}} := \{uavaw \mid u, v, w \in \mathbb{A}^*, a \in \mathbb{A}\}.$$

Is this language recognised by a deterministic nominal G-automaton?

NO

There are orbit finite join-irreducible derivative languages



Join-irreducible derivative languages

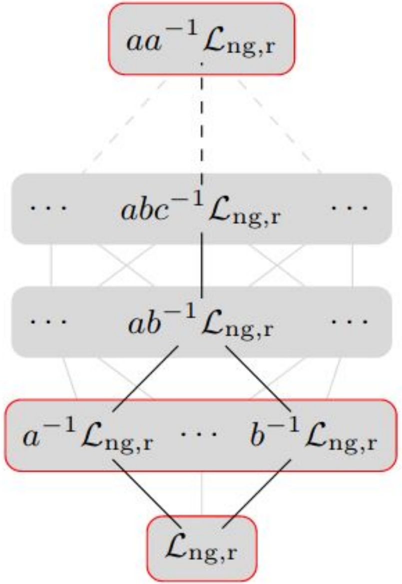
$$\mathcal{L}_{\mathrm{ng,r}} := \{uavaw \mid u, v, w \in \mathbb{A}^*, a \in \mathbb{A}\}.$$

Is this language recognised by a deterministic nominal G-automaton?

NO

There are orbit finite join-irreducible derivative languages

We can apply the Residual Automata Theorem!



Residual Automata Theorem

Orbit finite nominal G-Set $\mathbb{A}$

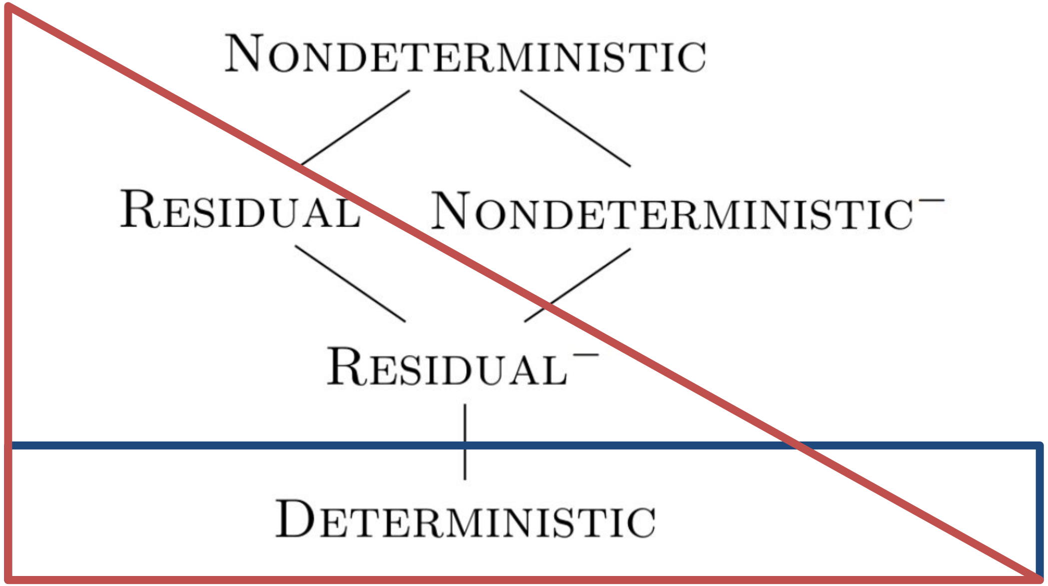
G-Language $\mathcal{L} \subseteq \mathbb{A}^*$

$\mathcal{L}$ is recognized by a **residual nominal G-**Automaton

$$\Longleftrightarrow$$

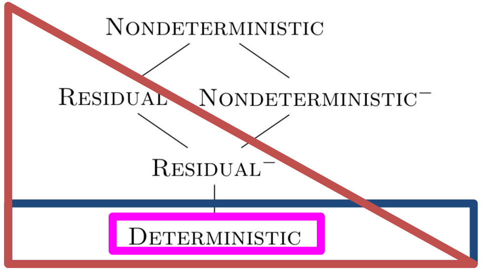
JI($Der(\mathcal{L})$) is **orbit** finite and generates $Der(\mathcal{L})$

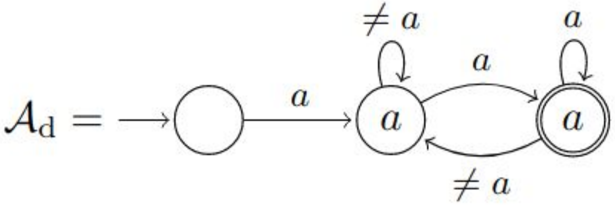
Nominal G-Automata

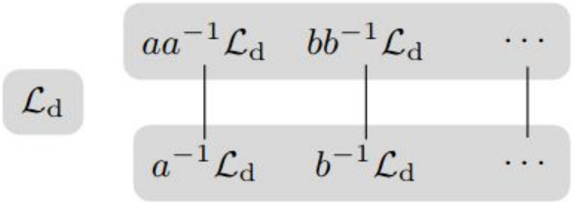


‾ = non-guessing

Deterministic

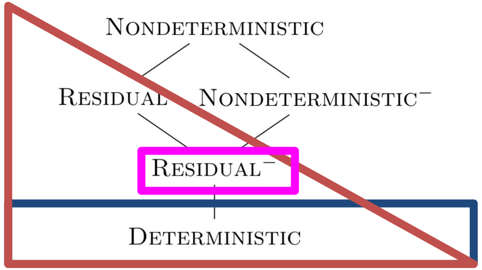


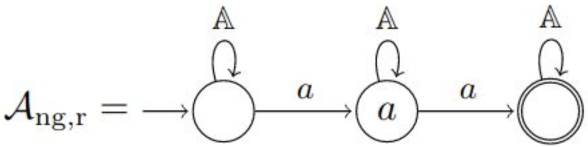


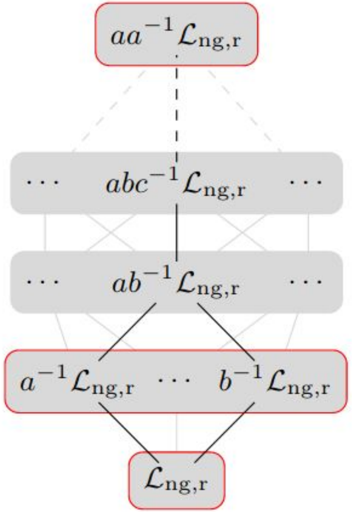


$$\mathcal{L}_d := \{awa \mid a \in \mathbb{A}, w \in \mathbb{A}^*\}$$

Non-guessing residual

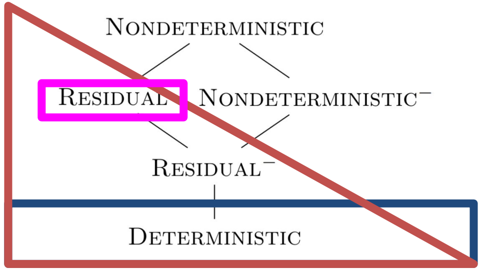


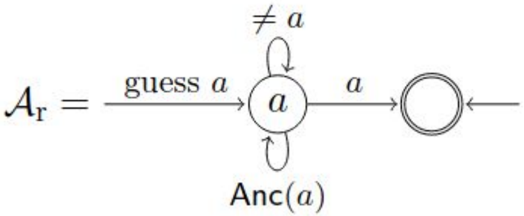


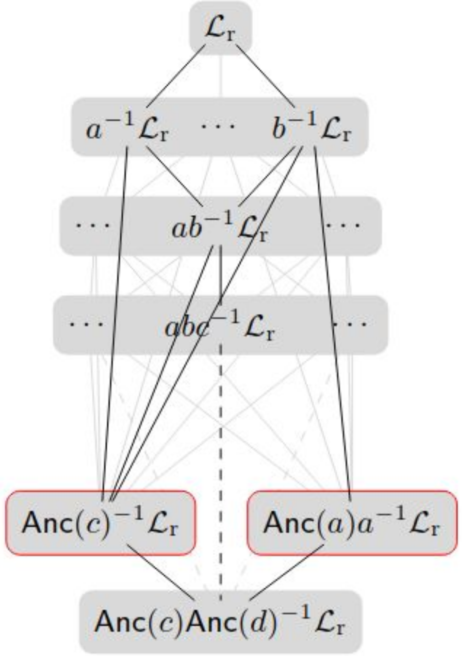


$$\mathcal{L}_{ng,r} := \{uavaw \mid u, v, w \in \mathbb{A}^*, a \in \mathbb{A}\}$$

Guessing residual

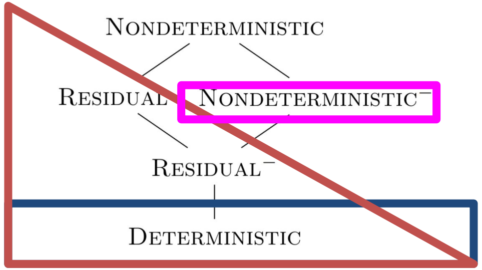


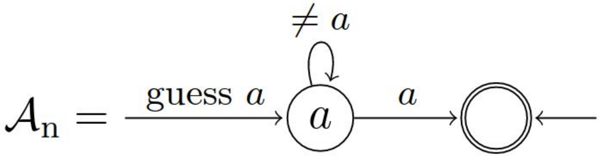


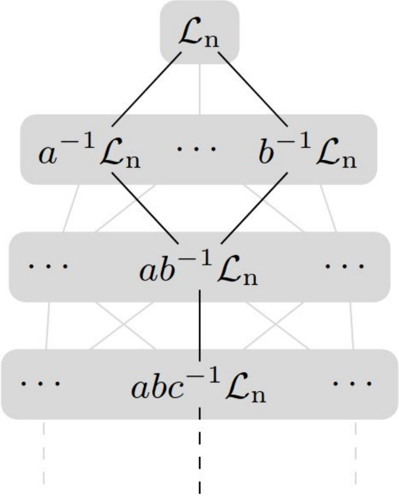


"Last letter is unique but anchored"

Nondeterministic non-guessing

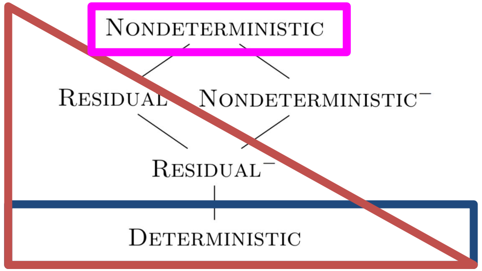


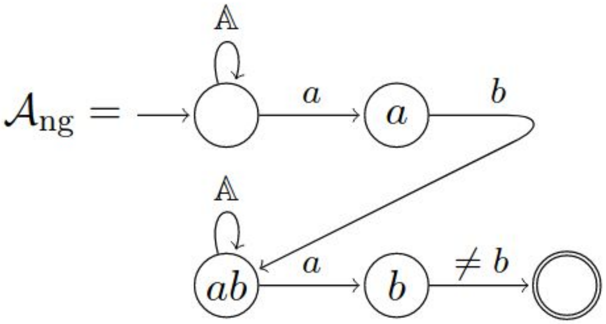


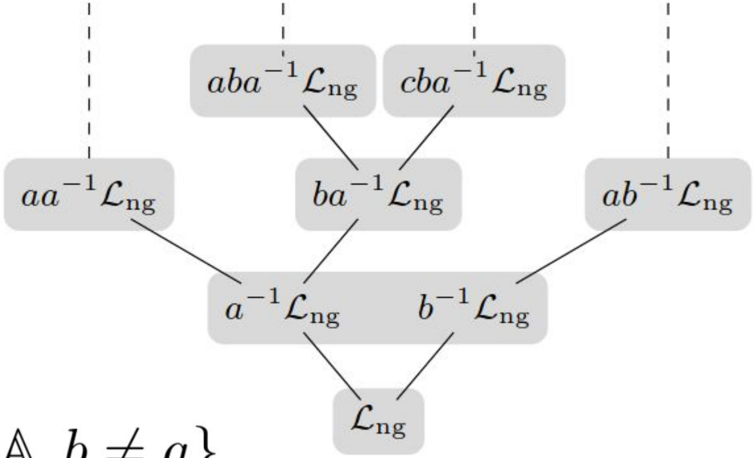


$$\mathcal{L}_n := \{wa \mid a \text{ not in } w\} \cup \{\epsilon\}$$

Nondeterministic non-guessing







$$\mathcal{L}_{\text{ng}} := \{uabvac \mid u, v \in \mathbb{A}^*, a, b, c \in \mathbb{A}, b \neq a\}$$

Summary

- Nominal G-Automata
- Myhill-Nerode Theorem
- Residual Automaton Theorem
- Some languages of each class

Questions?

“What about the learning algorithm?”

- L* for deterministic languages ⟶ deterministic nominal languages (νL*)
- Not to non-deterministic nominal automata (they are more expressive!)
- NL* for residual automata ⟶ νNL* for nominal residual automata

- It works by constructing an observation table of derivative languages
  - which is orbit-finite for the nominal deterministic case
  - for residual automata we can find the canonical representation

MODIFIED $\nu\mathsf{NL}^\star$ LEARNER

$S, E = \{\epsilon\}$

repeat

  while (S, E) is not residually-closed or not residually-consistent

  if (S, E) is not residually-closed

    find $s \in S, a \in A$ such that $row(sa) \in \mathsf{JI}(\mathrm{Rows}(S, E)) \setminus \mathrm{Rows}^\top(S, E)$

    k = length of the word sa

    $S = S \cup \Sigma^{\leq k}$

  if (S, E) is not residually-consistent

    find $s_1, s_2 \in S$, $a \in A$, and $e \in E$ such that $\mathrm{row}(s_1) \sqsubseteq \mathrm{row}(s_2)$ and $\mathcal{L}(s_1 ae) = 1$, $\mathcal{L}(s_2 ae) = 0$

    $E = E \cup \mathsf{orb}(ae)$

  Make the conjecture $N(S, E)$

  if the Teacher replies **no**, with a counter-example t

    $E = E \cup \{\mathsf{orb}(t_0) \mid t_0 \text{ is a suffix of } t\}$

until the Teacher replies **yes** to the conjecture $N(S, E)$.

return $N(S, E)$

Radboud University