Nominal Automata

Mikolaj Bojanczyk[1], Bartek Klin[12], Slawomir Lasota[1]
[1] *Warsaw University* [2] *University of Cambridge*

IFIP WG1.3, Aussois, 08/01/11

$$\lambda X.\ (\mathrm{nominal}\ X)$$

Nominal Automata

Mikolaj Bojanczyk[1], Bartek Klin[12], Slawomir Lasota[1]
[1] *Warsaw University* [2] *University of Cambridge*

IFIP WG1.3, Aussois, 08/01/11

Finite automata

An automaton:

- set Q of states
- alphabet A
- transition function $\delta : Q \times A \to Q$
- initial state $q_0 \in Q$
- accepting states $Q_a \subseteq Q$

Finite automata

An automaton:

- set Q of states
- alphabet A
- transition function $\delta : Q \times A \to Q$
- initial state $q_0 \in Q$
- accepting states $Q_a \subseteq Q$

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ Q \times A \arrow[r, "\delta"'] & Q \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

Finite automata

An automaton:

- set Q of states

finite

- alphabet A
- transition function $\delta : Q \times A \to Q$
- initial state $q_0 \in Q$
- accepting states $Q_a \subseteq Q$

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ Q \times A \arrow[r, "\delta"'] & Q \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

Finite automata

An automaton:

- set Q of states
- alphabet A
- transition function $\delta : Q \times A \to Q$
- initial state $q_0 \in Q$
- accepting states $Q_a \subseteq Q$

finite

?

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ Q \times A \arrow[r, "\delta"'] & Q \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

Finite memory automata [FK]

$A = \mathbb{N}$

Idea: store numbers in configurations...

Finite memory automata [FK]

$A = \mathbb{N}$

Idea: store numbers in configurations...
... but only compare them for equality

Finite memory automata [FK]

$A = \mathbb{N}$

- finite set Q of states

Finite memory automata [FK]

$A = \mathbb{N}$

- finite set Q of states
- finite store R of registers

Finite memory automata [FK]

$A = \mathbb{N}$

- finite set Q of states
- finite store R of registers

configurations: $X = Q \times (A+1)^R$

Finite memory automata [FK]

$A = \mathbb{N}$

- finite set Q of states
- finite store R of registers

configurations: $X = Q \times (A + 1)^R$

- transition function $\delta : X \times A \to X$
  - only checks letters for equality
  - new store $\subseteq$ old store $\cup$ $\{$letter just read$\}$

Finite memory automata [FK]

$A = \mathbb{N}$

- finite set Q of states
- finite store R of registers

  configurations: $X = Q \times (A + 1)^R$

- transition function $\delta : X \times A \to X$
  - only checks letters for equality
  - new store $\subseteq$ old store $\cup$ $\{$letter just read$\}$
- initial state, accepting states

Example

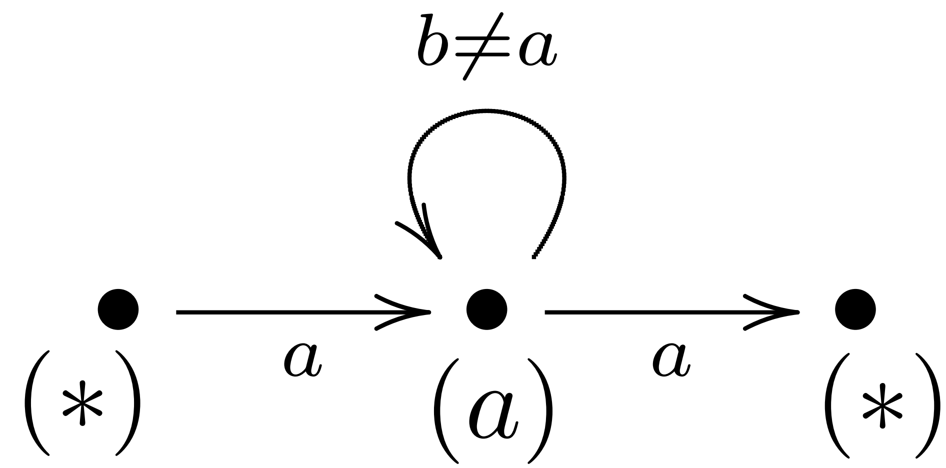
"The first letter appears again"

$$L = \{a_1 a_2 \dots a_n \in \mathbb{N}^* \mid \exists k > 1.\ a_1 = a_k\}$$

Example

"The first letter appears again"

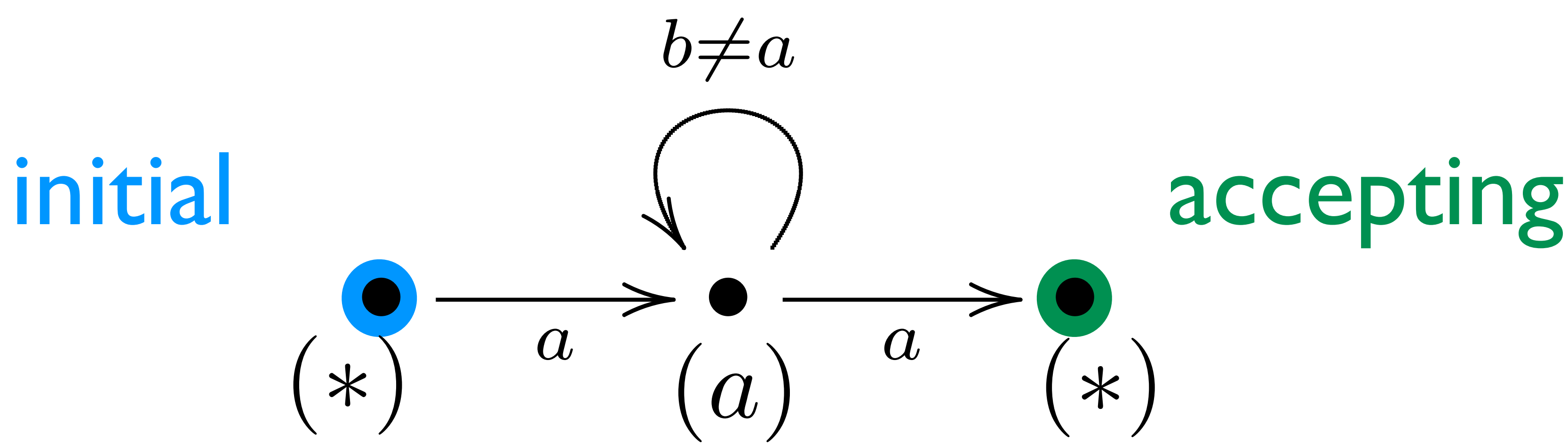
$$L = \{a_1 a_2 \dots a_n \in \mathbb{N}^* \mid \exists k > 1.\ a_1 = a_k\}$$



Example

"The first letter appears again"

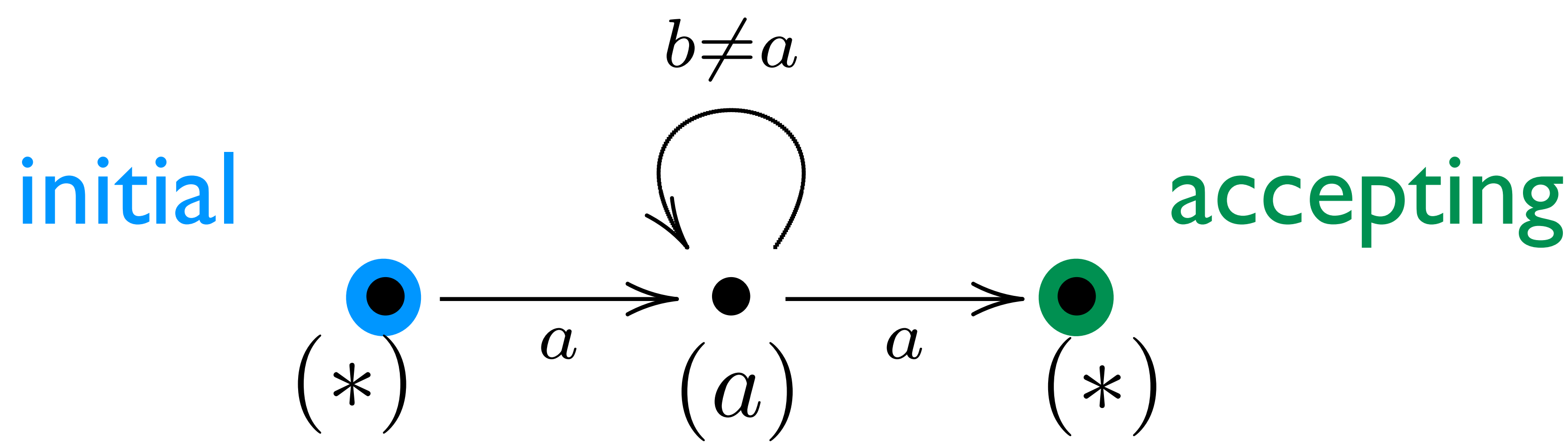
$$L = \{a_1 a_2 \ldots a_n \in \mathbb{N}^* \mid \exists k > 1.\ a_1 = a_k\}$$



Example

"The first letter appears again"

$$L = \{a_1 a_2 \ldots a_n \in \mathbb{N}^* \mid \exists k > 1.\ a_1 = a_k\}$$



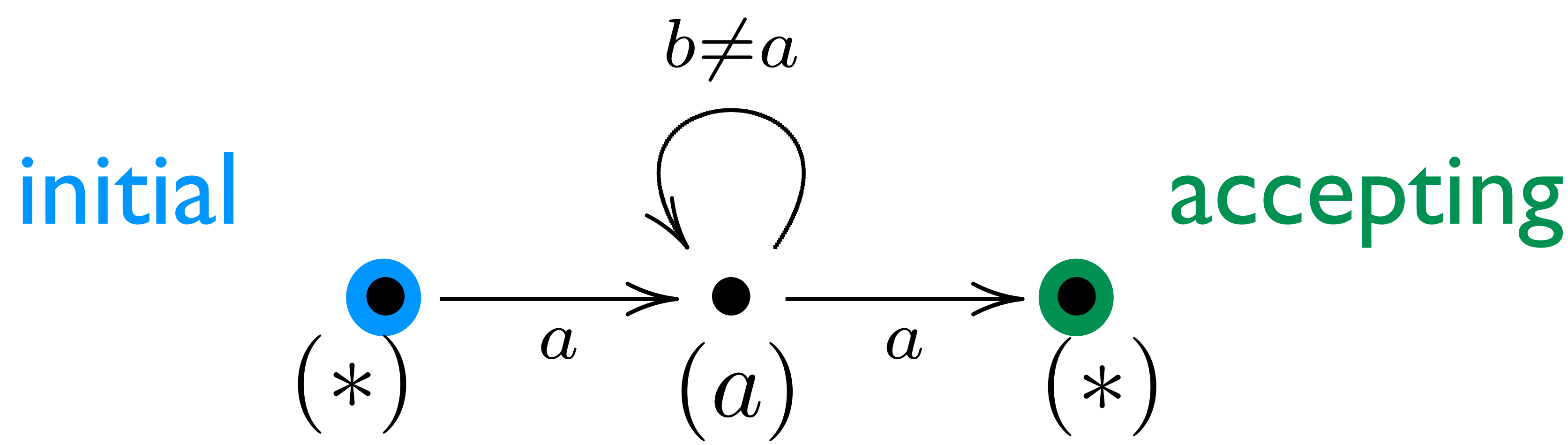
"The last letter appears before"

$$L = \{a_1 a_2 \ldots a_n \in \mathbb{N}^* \mid \exists k < n.\ a_k = a_n\}$$

Example

"The first letter appears again"

$$L = \{a_1 a_2 \dots a_n \in \mathbb{N}^* \mid \exists k > 1.\ a_1 = a_k\}$$



"The last letter appears before"

$$L = \{a_1 a_2 \dots a_n \in \mathbb{N}^* \mid \exists k < n.\ a_k = a_n\}$$

not recognizable

Finite memory automata [FK]

- finite set Q of states
- finite store R of registers

configurations: $X = Q \times (A+1)^R$

- transition function $\delta : X \times A \to X$
  - only checks letters for equality
  - new store $\subseteq$ old store $\cup$ $\{$letter just read$\}$
- initial state, accepting states

Finite memory automata [FK]

- finite set Q of states
- finite store R of registers

configurations: $X = Q \times (A+1)^R$

- transition function $\delta : X \times A \to X$
  - only checks letters for equality
  - new store $\subseteq$ old store $\cup$ $\{$letter just read$\}$
- initial state, accepting states

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

Syntactic automata

Myhill-Nerode equivalence: $L \subseteq A^*$

$$v \equiv_L w \iff \forall u \in A^*.\ (vu \in L \iff wu \in L)$$

Syntactic automata

Myhill-Nerode equivalence: $L \subseteq A^*$

$$v \equiv_L w \iff \forall u \in A^*.\ (vu \in L \iff wu \in L)$$

Fact: $v \equiv_L w \implies vu \equiv_L wu$

Syntactic automata

Myhill-Nerode equivalence: $L \subseteq A^*$

$$v \equiv_L w \iff \forall u \in A^*.\ (vu \in L \iff wu \in L)$$

Fact: $v \equiv_L w \implies vu \equiv_L wu$

$$\begin{tikzcd}
 & & 1 \arrow[d, "\iota"] & \\
A^*/_{\equiv_L} \times A \arrow[rr, "_\cdot_"'] & & A^*/_{\equiv_L} \arrow[r, "\alpha"'] & 2
\end{tikzcd}$$

Initial: $[\epsilon]_{\equiv_L}$

Accepting: $\{[w]_{\equiv_L} \mid w \in L\}$

Syntactic automata

Myhill-Nerode equivalence: $L \subseteq A^*$

$$v \equiv_L w \iff \forall u \in A^*.\ (vu \in L \iff wu \in L)$$

Fact: $v \equiv_L w \implies vu \equiv_L wu$

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ A^*/_{\equiv_L} \times A \arrow[r, "\_\cdot\_"'] & A^*/_{\equiv_L} \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

Syntactic automaton

Initial: $[\epsilon]_{\equiv_L}$

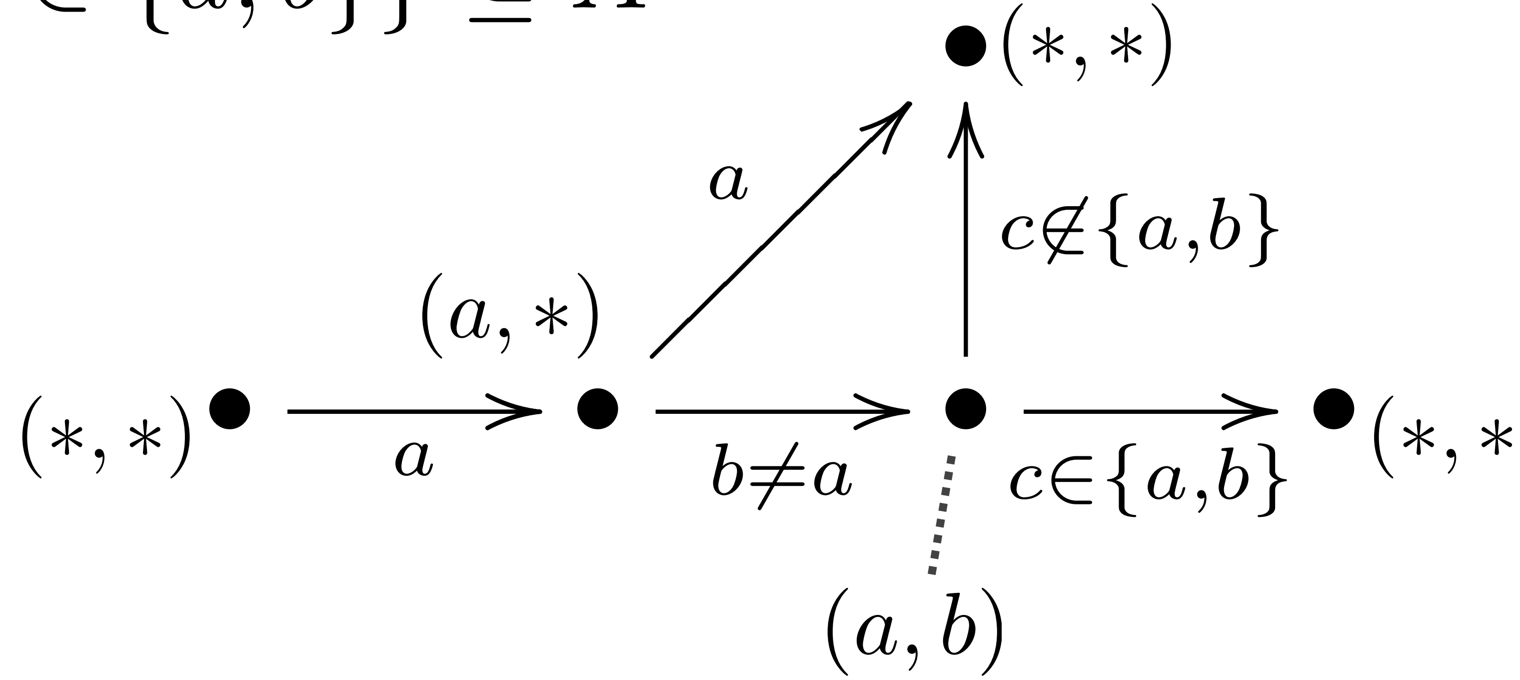
Accepting: $\{[w]_{\equiv_L} \mid w \in L\}$

Minimization problems for F.M.A.

$L = \{abc \mid a \neq b, c \in \{a, b\}\} \subseteq A^3$

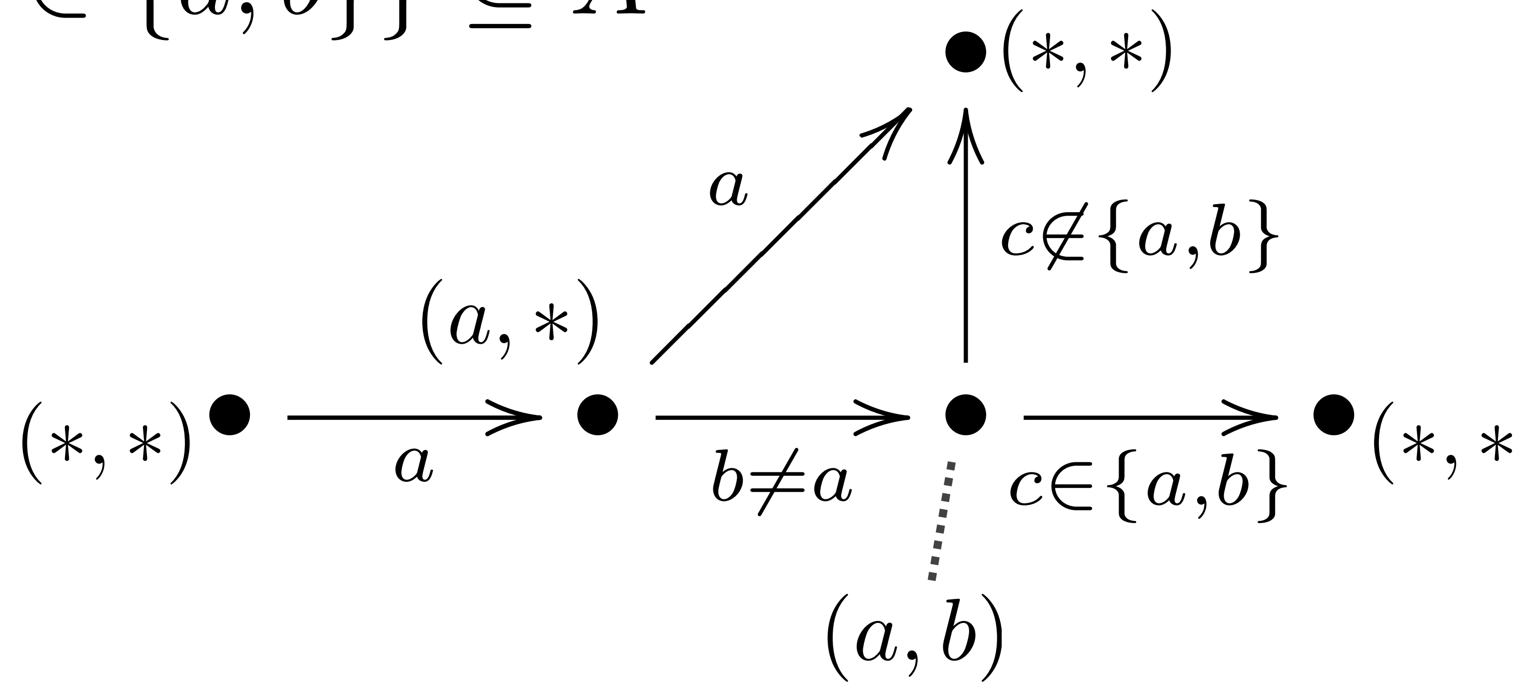
Minimization problems for F.M.A.

$L = \{abc \mid a \neq b, c \in \{a,b\}\} \subseteq A^3$



Minimization problems for F.M.A.

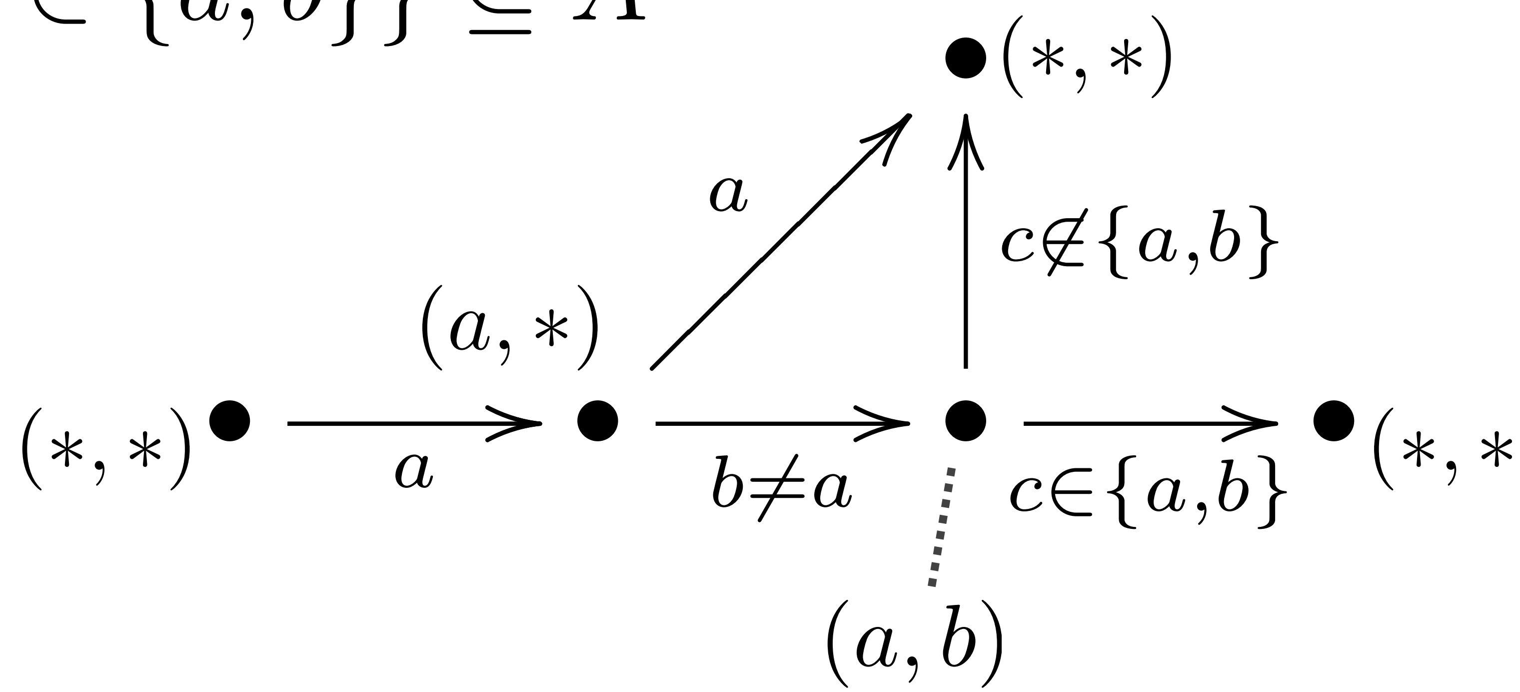
$$L = \{abc \mid a \neq b, c \in \{a,b\}\} \subseteq A^3$$



$$(*,*) \xrightarrow{ab} (a,b)$$
$$(*,*) \xrightarrow{ba} (b,a)$$

Minimization problems for F.M.A.

$$L = \{abc \mid a \neq b, c \in \{a,b\}\} \subseteq A^3$$

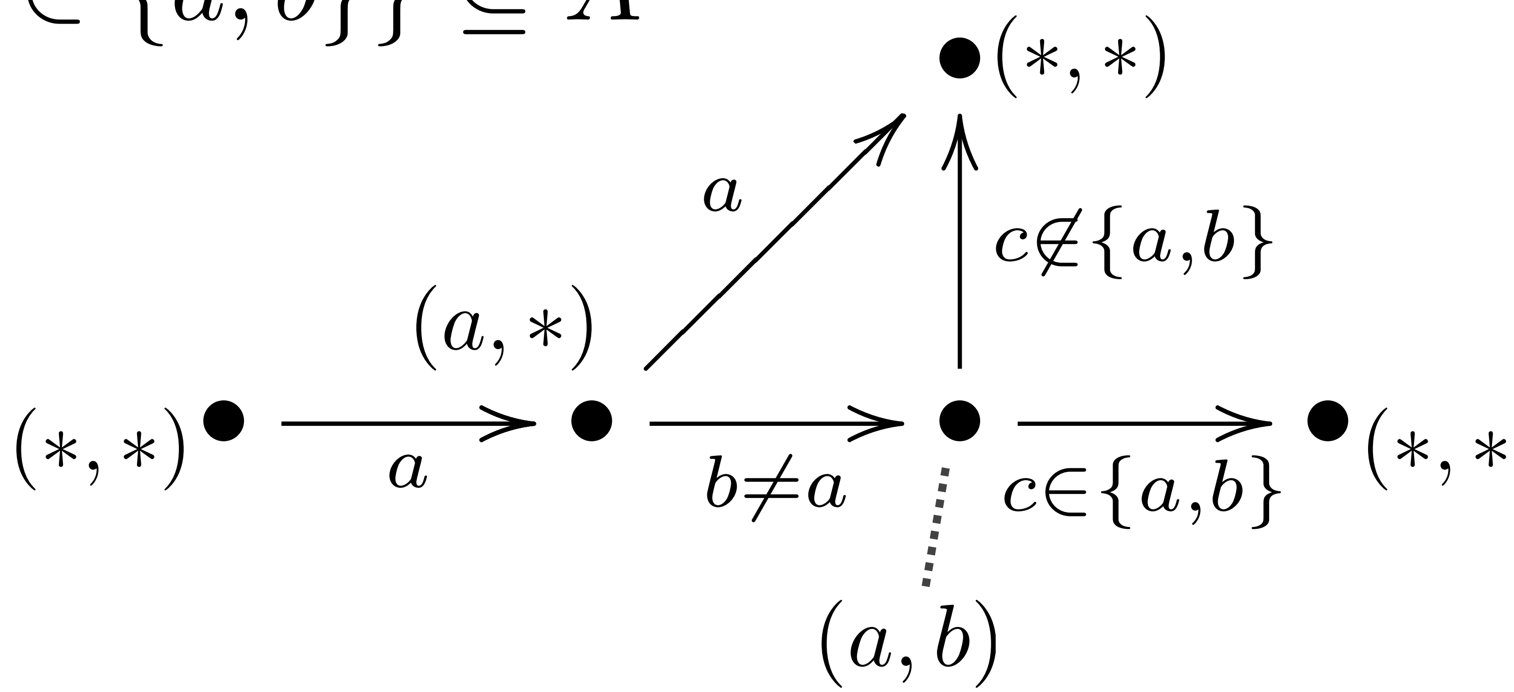


$$(*,*) \xrightarrow{ab} (a,b)$$
$$(*,*) \xrightarrow{ba} (b,a)$$

but $\quad ab \equiv_L ba$

Minimization problems for F.M.A.

$$L = \{abc \mid a \neq b, c \in \{a,b\}\} \subseteq A^3$$



$$(*,*) \xrightarrow{ab} (a,b)$$
$$(*,*) \xrightarrow{ba} (b,a)$$

but $ab \equiv_L ba$

Worse: for any $G \leq Sym(\{1,2,\ldots,n\})$,

$$L = \{a_1 \cdots a_n b_1 \cdots b_n \mid \exists \pi \in G.\ \forall i = 1..n.\ a_i = b_{\pi(i)}\}$$

F.M.A. are equivariant

Fix $G = Sym(A)$.

Defn.: A G-set is:

- a set X
- an action $\_ \cdot \_ : X \times G \to X$ (+ axioms)

Defn.: Function $f : X \to Y$ is equivariant if

$$f(x \cdot \pi) = f(x) \cdot \pi$$

F.M.A. are equivariant

Fix $G = Sym(A)$.

Defn.: A G-set is:

- a set X
- an action $\_ \cdot \_ : X \times G \to X$ (+ axioms)

Defn.: Function $f : X \to Y$ is equivariant if

$$f(x \cdot \pi) = f(x) \cdot \pi$$

$\boxed{\text{category } G\text{-Set}}$

F.M.A. are equivariant

Fix $G = Sym(A)$.

Defn.: A G-set is:

- a set X
- an action $\_ \cdot \_ : X \times G \to X$ (+ axioms)

Defn.: Function $f : X \to Y$ is equivariant if

$$f(x \cdot \pi) = f(x) \cdot \pi$$

category G-**Set**

Fact: In a F.M.A.:

- configurations form a G-set
- $\delta : X \times A \to X$ is equivariant

G-set automata

Idea: study diagrams

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

in G-$\mathbf{Set}$.

G-set automata

Idea: study diagrams

finite?

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

in G-$\mathbf{Set}$.

G-set automata

Idea: study diagrams

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

finite?

in G-**Set**.

Defn.: The orbit of $x \in X$:

$$x \cdot G = \{x \cdot \pi \mid \pi \in G\}$$

G-set automata

Idea: study diagrams

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

finite?

in G-**Set**.

Defn.: The orbit of $x \in X$:

$$x \cdot G = \{x \cdot \pi \mid \pi \in G\}$$

In an F.M.A., $Q \cong orbits(X)$

G-set automata

Idea: study diagrams

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

finite?

in G-**Set**.

Defn.: The orbit of $x \in X$:

$$x \cdot G = \{x \cdot \pi \mid \pi \in G\}$$

In an F.M.A., $Q \cong orbits(X)$

So we require X orbit-finite.

G-set automata

Idea: study diagrams

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$ in G-**Set**.

finite?

Defn.: The orbit of $x \in X$:

$$x \cdot G = \{x \cdot \pi \mid \pi \in G\}$$

In an F.M.A., $Q \cong orbits(X)$

So we require X orbit-finite.

Can we model finiteness of the store?

Nominal sets [GP]

Defn.: $C \subseteq A$ supports $x \in X$ if

$$\forall c \in C.\ \pi(c) = c \qquad \Longrightarrow \qquad x \cdot \pi = x$$

Nominal sets [GP]

Defn.: $C \subseteq A$ supports $x \in X$ if

$$\forall c \in C.\ \pi(c) = c \qquad \Longrightarrow \qquad x \cdot \pi = x$$

Nominal set: every element has a finite support.

Nominal sets [GP]

Defn.: $C \subseteq A$ supports $x \in X$ if

$$\forall c \in C.\ \pi(c) = c \qquad \Longrightarrow \qquad x \cdot \pi = x$$

Nominal set: every element has a finite support.

G-$\mathbf{Nom}$: nominal sets and equivariant functions

Nominal sets [GP]

Defn.: $C \subseteq A$ supports $x \in X$ if

$$\forall c \in C.\ \pi(c) = c \qquad \Longrightarrow \qquad x \cdot \pi = x$$

Nominal set: every element has a finite support.

$G\text{-}\mathbf{Nom}$: nominal sets and equivariant functions

Fact: In a nominal set, every x has the least support

Nominal sets [GP]

Defn.: $C \subseteq A$ supports $x \in X$ if

$$\forall c \in C.\ \pi(c) = c \qquad \Longrightarrow \qquad x \cdot \pi = x$$

Nominal set: every element has a finite support.

G-$\mathbf{Nom}$: nominal sets and equivariant functions

Fact: In a nominal set, every x has the least support

$$\mathrm{supp}(x) = \{a \in A \mid \{b \in A \mid x \cdot (ab) \neq x\} \text{ is infinite}\}$$

Nominal automata

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

G-$\mathbf{Nom}$

Nominal automata

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

G-$\mathbf{Nom}$

- finite set Q of states

Nominal automata

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

G-$\mathbf{Nom}$

- finite set Q of states
- for each state: finite set R_q, group $S_q \leq Sym(R_q)$

Nominal automata

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

G-**Nom**

- finite set Q of states
- for each state: finite set R_q, group $S_q \leq Sym(R_q)$
- transition: $\theta : \coprod_{q \in Q}(R_q + 1) \to \coprod_{p \in Q}((R_q + 1)^{R_p})$

Nominal automata

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

G-**Nom**

- finite set Q of states
- for each state: finite set R_q, group $S_q \leq Sym(R_q)$
- transition: $\theta : \coprod_{q\in Q}(R_q + 1) \to \coprod_{p\in Q}((R_q + 1)^{R_p})$

+ a condition involving S_q, S_p

Nominal automata

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

G-**Nom**

- finite set Q of states
- for each state: finite set R_q, group $S_q \leq Sym(R_q)$
- transition: $\theta : \coprod_{q \in Q}(R_q + 1) \to \coprod_{p \in Q}((R_q + 1)^{R_p})$

  \+ a condition involving S_q, S_p
- initial, accepting states

Nominal automata

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

G-**Nom**

$$X = \coprod_{q \in Q} (A^{R_q} \text{ up to } S_q)$$

- finite set Q of states
- for each state: finite set R_q, group $S_q \leq Sym(R_q)$
- transition: $\theta : \coprod_{q \in Q}(R_q + 1) \rightarrow \coprod_{p \in Q}((R_q + 1)^{R_p})$

  + a condition involving S_q, S_p
- initial, accepting states

Nominal automata

$$\begin{tikzcd} & 1 \arrow[d, "\iota"] & \\ X \times A \arrow[r, "\delta"'] & X \arrow[r, "\alpha"'] & 2 \end{tikzcd}$$

G-**Nom**

$$Q = orbits(X)$$
$$R_q = \mathrm{supp}(x) \quad S_q = G_x \qquad (x \in q)$$

- finite set Q of states
- for each state: finite set R_q, group $S_q \leq Sym(R_q)$
- transition: $\theta : \coprod_{q \in Q}(R_q + 1) \to \coprod_{p \in Q}((R_q + 1)^{R_p})$

  \+ a condition involving S_q, S_p
- initial, accepting states

Structured alphabets

Structured alphabets

Q: What if the alphabet is equipped with

- a total order,
- a partial order,
- a graph structure,
- ...

which we can check for?

Structured alphabets

Q: What if the alphabet is equipped with

- a total order,
- a partial order,
- a graph structure,
- ...

which we can check for?

A: Repeat the theory with some $G \leq Sym(A)$

E.g. $G =$ monotone bijections of $\mathbb{Q}$

G-nominal sets

Defn.: $C \subseteq A$ supports $x \in X$ if

$$\forall \pi \in G \Big(\forall c \in C.\ \pi(c) = c \qquad \Longrightarrow \qquad x \cdot \pi = x \Big)$$

Nominal set: every element has a finite support.

$G\text{-}\mathbf{Nom}$: G-nominal sets and equivariant functions

G-nominal sets

Defn.: $C \subseteq A$ supports $x \in X$ if

$$\forall \pi \in G\big(\forall c \in C.\ \pi(c) = c \qquad \Longrightarrow \qquad x \cdot \pi = x\big)$$

Nominal set: every element has a finite support.

G-$\mathbf{Nom}$: G-nominal sets and equivariant functions

Caution: least supports might not exist.

Representing G-nominal sets

Automaton:

- orbit-finite G-set X
- ...

Representing G-nominal sets

Step 1: every G-set is a sum of single-orbit ones.

Automaton:

- orbit-finite G-set X

- ...

Representing G-nominal sets

Step 1: every G-set is a sum of single-orbit ones.

Automaton:

- finite set Q of states
- for each state: single-orbit nominal set X_q
- ...

Representing G-nominal sets

Step 1: every G-set is a sum of single-orbit ones.

Step 2: for any single-orbit X,

$$X \cong G/H^r \quad \text{for some } H \leq G.$$

Automaton:

- finite set Q of states
- for each state: single-orbit nominal set X_q
- ...

Representing G-nominal sets

Step 1: every G-set is a sum of single-orbit ones.

Step 2: for any single-orbit X,

$$X \cong G/H^r \quad \text{for some } H \leq G.$$

Automaton:

- finite set Q of states
- for each state: group $H_q \leq G$
- ...

Representing G-nominal sets

Step 1: every G-set is a sum of single-orbit ones.

Step 2: for any single-orbit X,

$X \cong G/H^r$ for some $H \leq G$.

Step 3: X nominal iff H open:

$G_C = \{\pi \in G \mid \pi_C = \mathrm{id}\} \leq H$ for some $C \subseteq_{\mathrm{fin}} A$

Automaton:

- finite set Q of states
- for each state: group $H_q \leq G$
- ...

Representing G-nominal sets

Step 1: every G-set is a sum of single-orbit ones.

Step 2: for any single-orbit X,

$$X \cong G/H^r \quad \text{for some } H \leq G.$$

Step 3: X nominal iff H open:

$$G_C = \{\pi \in G \mid \pi_C = \mathrm{id}\} \leq H \text{ for some } C \subseteq_{\mathrm{fin}} A$$

Automaton:

- finite set Q of states
- for each state: open group $H_q \leq G$
- ...

Representing G-nominal sets

Step 1: every G-set is a sum of single-orbit ones.

Step 2: for any single-orbit X,

$$X \cong G/H^r \quad \text{for some } H \leq G.$$

Step 3: X nominal iff H open:

$$G_C = \{\pi \in G \mid \pi_C = \mathrm{id}\} \leq H \ \text{ for some } C \subseteq_{\mathrm{fin}} A$$

Step 4: ...

Automaton:

- finite set Q of states
- for each state: open group $H_q \leq G$
- ...

Fraïssé limits

Idea: - $\mathbb{N}$ universally embeds finite sets

- $\mathbb{Q}$ universally embeds finite total orders
- universal partial order? graph? etc.

Fraïssé limits

Idea:
- $\mathbb{N}$ universally embeds finite sets
- $\mathbb{Q}$ universally embeds finite total orders
- universal partial order? graph? etc.

Fraïssé construction:

a class of finite rel. structures closed under
- isomorphisms
- substructures
- amalgamation

has a countable "limit" $\mathcal{U}$ that embeds them.

Fraïssé limits

Idea: - $\mathbb{N}$ universally embeds finite sets

- $\mathbb{Q}$ universally embeds finite total orders
- universal partial order? graph? etc.

Fraïssé construction:

a class of finite rel. structures closed under

- isomorphisms
- substructures
- amalgamation

has a countable "limit" $\mathcal{U}$ that embeds them.

We use $G = Aut(\mathcal{U}) \leq Sym(|\mathcal{U}|)$

Fraïssé automata

Fix a class $\mathcal{K}$ of relational structures

(subject to conditions)

Fraïssé automata

Fix a class $\mathcal{K}$ of relational structures

(subject to conditions)

Automaton:

- finite set Q of states
- for each state: fin. structure R_q,

  group $S_q \leq Aut(R_q)$

Fraïssé automata

Fix a class $\mathcal{K}$ of relational structures

(subject to conditions)

Automaton:

- finite set Q of states
- for each state: fin. structure R_q,

  group $S_q \leq Aut(R_q)$

configurations: embeddings of R_q in $\mathcal{U}$

up to S_q

Fraïssé automata

Fix a class $\mathcal{K}$ of relational structures

(subject to conditions)

Automaton:

- finite set Q of states
- for each state: fin. structure R_q,

  group $S_q \leq Aut(R_q)$

  configurations: embeddings of R_q in $\mathcal{U}$

  up to S_q

- transition function: ....
- initial, accepting states

More fun

More fun

In Fraïssé situations, we represent in a finite way:

- G-nominal sets
- subsets, products
- equivariant functions and relations
- 1^{st}-order logic is decidable on orbit-finite sets

More fun

In Fraïssé situations, we represent in a finite way:

- G-nominal sets
- subsets, products
- equivariant functions and relations
- 1st-order logic is decidable on orbit-finite sets

So we can represent:

- automata
- push-down automata
- Turing machines
- ...

More fun

In Fraïssé situations, we represent in a finite way:

- G-nominal sets
- subsets, products
- equivariant functions and relations
- 1st-order logic is decidable on orbit-finite sets

So we can represent:

- automata
- push-down automata
- Turing machines
- ...

Is
$\mathbf{P} = \mathbf{NP}$
in nominal sets?