Computation Theory over Sets with Atoms

Bartek Klin
University of Oxford

MOVEP Summer School 2022, Aalborg

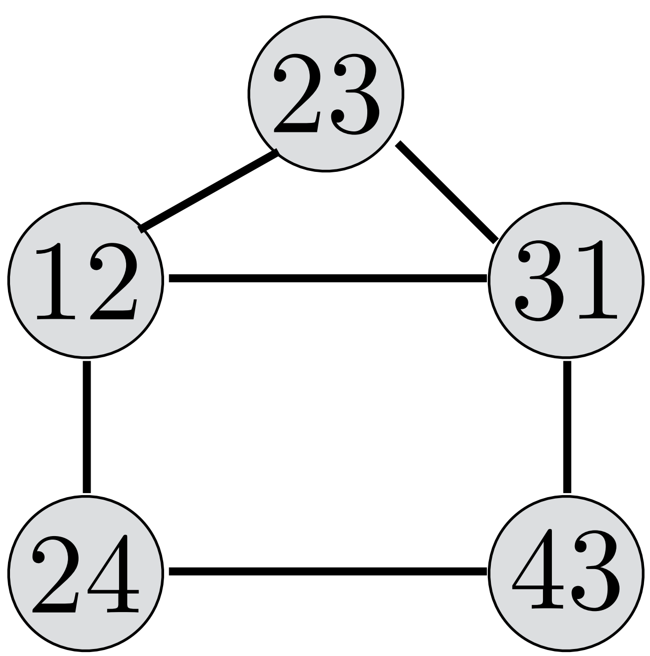
Puzzle 1 : a graph

- nodes: ordered pairs of distinct natural numbers

$$\{\, nm \mid n \neq m \in \mathbb{N} \,\}$$

- edges (undirected):

$$\{\, nm \text{ — } mk \mid n \neq k \,\}$$



Is it 3-colorable?

Puzzle II : linear equations

- variables: ordered pairs of distinct natural numbers

$$\{ nm \mid n \neq m \in \mathbb{N} \}$$

- equations:

$$nm + mk + kn = 0$$

$$12 + 21 = 1$$

Does it have a solution in $\mathbb{Z}_2$?

General theme

Replace finite structures
with infinite, but highly symmetric ones
in:

- automata theory
- computability theory
- modelling / verification
- algorithms

...

- all the way down to set theory

Plan

1. Register automata
2. Sets with atoms
3. μ-calculus with atoms

-- Turing machines with atoms

-- Constraint satisfaction problems with atoms

-- Programming with atoms

1
Register automata

A finite automaton is:

finite

- a set Q of states
- an alphabet Σ
- initial state $q_0 \in Q$, accepting states $F \subseteq Q$
- transition function $\delta : Q \times \Sigma \to Q$

(or relation $\delta \subseteq Q \times \Sigma \times Q$)

Example language: $$\bigcup_{a \in \Sigma} a(\Sigma \setminus a)^*$$

What about infinite alphabets?

Idea 1: keep the definition as it is

- a set Q of states ← finite
- an alphabet Σ ← infinite
- initial state $q_0 \in Q$, accepting states $F \subseteq Q$
- transition function $\delta : Q \times \Sigma \to Q$

(or relation $\delta \subseteq Q \times \Sigma \times Q$)

Problem: does not recognize $\bigcup_{a \in \Sigma} a(\Sigma \setminus a)^*$

What about infinite alphabets?

Idea II: allow infinitely many states

- a set Q of states
- an alphabet Σ

infinite

- initial state $q_0 \in Q$, accepting states $F \subseteq Q$
- transition function $\delta : Q \times \Sigma \to Q$

(or relation $\delta \subseteq Q \times \Sigma \times Q$)

Problem: every language is recognized

Register automata

A register automaton is:

- a set Q of states
- a set R of registers
- an alphabet $\mathbb{A}$ (or $\Sigma \times \mathbb{A}$)
- initial state $q_0 \in Q$, accepting states $F \subseteq Q$
- configurations: $\Gamma = Q \times (\mathbb{A} \cup \{\perp\})^R$
- transition function $\delta : \Gamma \times \mathbb{A} \to \Gamma$

  (or relation $\delta \subseteq \Gamma \times \mathbb{A} \times \Gamma$)

finite

infinite

that only checks $\mathbb{A}$ for equality. ?

“Only checking for equality”, syntactically

Every transition:

$$q \xrightarrow{a} q'$$

is guarded by a Boolean combination of conditions:

$$a = r_i \qquad a = r'_j \qquad r_i = r_j \qquad r_i = r'_j$$

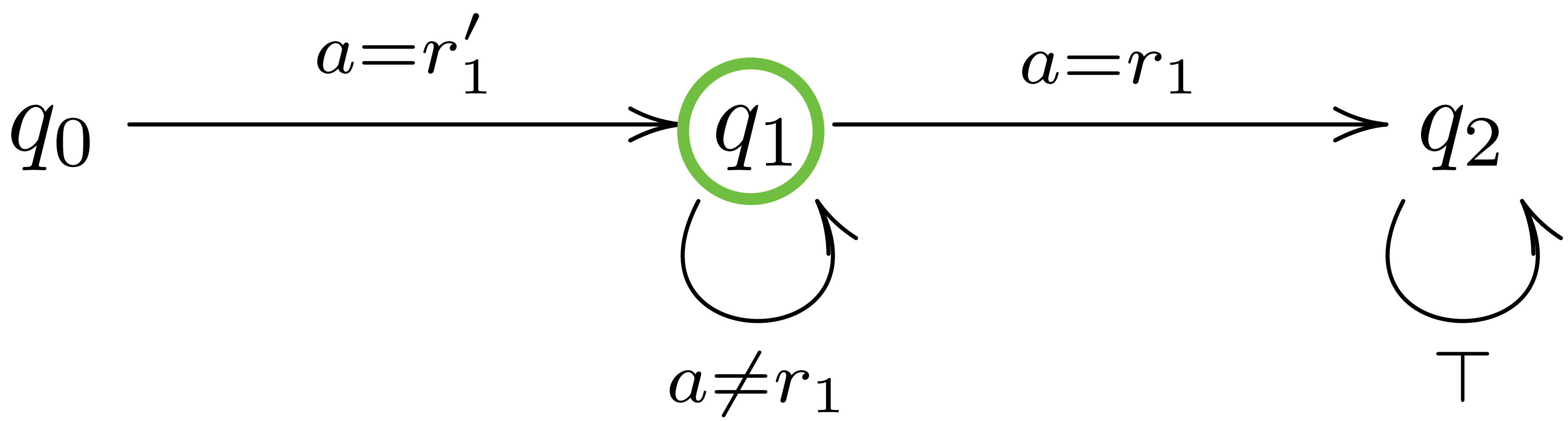
(so a is a “letter variable”, not an actual letter)

r_i - old i-th register

r'_i - new i-th register

Example

$$\bigcup_{a \in \mathbb{A}} a(\mathbb{A} \setminus a)^*$$



This is a deterministic register automaton.

"Only checking for equality", semantically

Every bijection $\pi : \mathbb{A} \to \mathbb{A}$ acts on configurations:

$$(q, a_1, \ldots, a_k) \cdot \pi = (q, \pi(a_1), \ldots, \pi(a_k))$$

This defines a group action of $\mathrm{Aut}(\mathbb{A})$ on Γ.

A group action of G on a set X :

$$\_ \cdot \_ : X \times G \to X$$

such that

$$x \cdot 1 = x$$

$$x \cdot (fg) = (x \cdot f) \cdot g$$

"Only checking for equality", semantically

Every bijection $\pi : \mathbb{A} \to \mathbb{A}$ acts on configurations:

$$(q, a_1, \ldots, a_k) \cdot \pi = (q, \pi(a_1), \ldots, \pi(a_k))$$

This defines a group action of $\mathrm{Aut}(\mathbb{A})$ on Γ.

We require δ to be equivariant:

if $(\gamma, a, \gamma') \in \delta$ then $(\gamma \cdot \pi, \pi(a), \gamma' \cdot \pi) \in \delta$

for all π.

Fact: The syntactic and the semantic conditions are equivalent.

It is tempting to write:

A register automaton is:

- a set Γ of configurations
- a group action of $\mathrm{Aut}(\mathbb{A})$ on Γ
- an alphabet $\mathbb{A}$ (or $\Sigma \times \mathbb{A}$)
- initial and accepting configurations
- transition function $\delta : \Gamma \times \mathbb{A} \to \Gamma$

  (or relation $\delta \subseteq \Gamma \times \mathbb{A} \times \Gamma$)

  that is equivariant.

infinite

This is too powerful

(we'll fix it later)

Questions

Q1: What about other computation models, logics, calculi etc?

Q2: What if we want to check for more than equality?

11
Sets with Atoms

Slogans

X = set, function, relation, automaton,
Turing machine, grammar, graph,
system of equations...

X with atoms

Infinite but with lots of symmetries

orbit-finite

Infinite but symbolically finitely presentable

We can compute on them

Von Neumann hierarchy

A hierarchy of universes:

$$\mathcal{U}_0 = \emptyset$$

$$\mathcal{U}_{\alpha+1} = \mathcal{P}\mathcal{U}_\alpha$$

$$\mathcal{U}_\beta = \bigcup_{\alpha<\beta} \mathcal{U}_\alpha$$

defined for every ordinal number.

Elements of sets are other sets,
in a well founded way

Every set sits somewhere in this hierarchy.

Sets with atoms

$\mathbb{A}$ - a countable set of atoms

A hierarchy of universes:

$$\mathcal{U}_0 = \emptyset$$
$$\mathcal{U}_{\alpha+1} = \mathcal{P}\mathcal{U}_\alpha + \mathbb{A}$$
$$\mathcal{U}_\beta = \bigcup_{\alpha<\beta} \mathcal{U}_\alpha$$

Elements of sets with atoms are atoms
or other sets with atoms, in a well founded way

A canonical group action:

$$\_ \cdot \_ : \mathcal{U} \times \mathrm{Aut}(\mathbb{A}) \to \mathcal{U}$$

Finite support

$S \subseteq \mathbb{A}$ supports X if

$\forall a \in S. \pi(a) = a$ implies $x \cdot \pi = x$

A legal set with atoms:

- has a finite support,
- every element has a finite support,
- and so on.

A set is equivariant if it has empty support.

Examples

$a \in \mathbb{A}$	is supported by	$\{a\}$
$\mathbb{A}$	is equivariant	
$S \subseteq \mathbb{A}$	is supported by	S
$\mathbb{A} \setminus S$	is supported by	S

Fact: $S \subseteq \mathbb{A}$ is fin. supp. iff it is finite or co-finite

$$\mathbb{A}^{(2)} = \{(d,e) \mid d,e \in \mathbb{A}, d \neq e\}$$ is equivariant

$$\binom{\mathbb{A}}{2} = \{\{d,e\} \mid d,e \in \mathbb{A}, d \neq e\}$$ is equivariant

Closure properties

Legal sets with atoms are closed under:

- unions, intersections, set differences
- Cartesian products
- taking finitely supported subsets
- quotienting by finitely supported equivalence relations

BUT not under powersets!

$\mathcal{P}(\mathbb{A})$ is equivariant but not legal.

They are closed under finite powersets $\mathcal{P}_{\mathrm{fin}}(\mathbb{A})$
and finitely supported powersets $\mathcal{P}_{\mathrm{fs}}(\mathbb{A})$

Relations and functions

Relations and functions are sets too, so:

$R \subseteq X \times Y$ is equivariant iff

xRy implies $(x \cdot \pi) R (y \cdot \pi)$ for all π

$f : X \to Y$ is equivariant iff

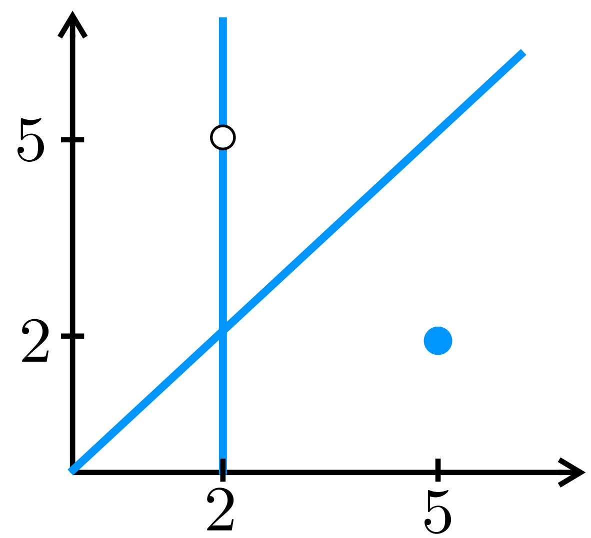
$f(x \cdot \pi) = f(x) \cdot \pi$ for all π

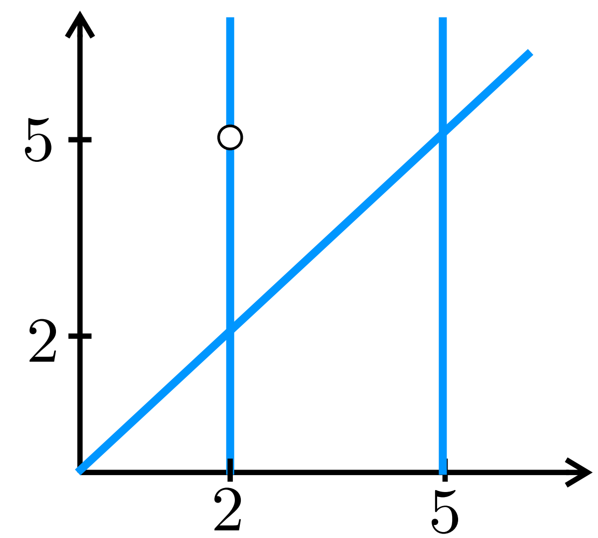
Examples

For fixed $2, 5 \in \mathbb{A}$:

$$R = \{(5,2)\} \cup \{(2,d) \mid d \neq 5\} \cup \{(d,d)\}$$

R





R^*

R , R^* are supported by $\{2, 5\}$

Examples ctd.

Equivariant binary relations on $\mathbb{A}$:

- empty
- equality
- total
- inequality

No equivariant function from $\binom{\mathbb{A}}{2}$ to $\mathbb{A}$, but

$$\{(\{a,b\},a) \mid a,b \in \mathbb{A}\}$$

is an equivariant relation.

Only equiv. functions from $\mathbb{A}^2$ to $\mathbb{A}$ are projections

Only equiv. function from $\mathbb{A}$ to $\mathbb{A}^2$ is the diagonal

Orbits

The orbit of x is the set $\{x \cdot \pi \mid \pi \in \mathrm{Aut}(\mathbb{A})\}$

Every equivariant set is a disjoint union of orbits.

Orbit-finite set if the union is finite.

More generally: the S-orbit of x is

$$\{x \cdot \pi \mid \pi \in \mathrm{Aut}_S(\mathbb{A})\}$$

Fact: An orbit-finite set is S-orbit-finite for every finite S.

Examples

Orbit-finite sets:

$$\mathbb{A} \qquad \mathbb{A}^n \qquad \binom{\mathbb{A}}{n}$$

$$\mathbb{A}^{\triangleleft} = \{\{(a,b,c),(b,c,a),(c,a,b)\} \mid a,b,c \in \mathbb{A}\}$$

- closed under finite union, intersection difference, finite Cartesian product
- but not under (even finite) powerset!

Not orbit-finite:

$$\mathbb{A}^* \qquad \mathcal{P}_{\mathrm{fin}}(\mathbb{A})$$

Automata with atoms

A automaton with atoms is:

- a set Q of states
- an alphabet Σ
- initial state $q_0 \in Q$, accepting states $F \subseteq Q$
- transition function $\delta : Q \times \Sigma \to Q$

  (or relation $\delta \subseteq Q \times \Sigma \times Q$)

orbit-finite

equivariant

Fact: these are expressively equivalent to reg. aut.

Finite presentation

A set-builder expression:

$$\{e \mid a_1, \ldots, a_n \in \mathbb{A},\ \phi[a_1, \ldots, a_n, b_1, \ldots, b_m]\}$$

expression bound variables FO(=)-formula free variables

Add also $\emptyset$ and $\cup$.

Fact: s.-b. e. + interpretation of free vars. as atoms
= a hereditarily orbit-finite set with atoms

Fact: Every h. o.-f. set is of this form.

Examples

The graph puzzle:

$$G = (V, E)$$
$$V = \{(a, b) \mid a, b \in \mathbb{A}, a \neq b\}$$
$$E = \{\{(a, b), (b, c)\} \mid a, b, c \in \mathbb{A}, a \neq b \neq c\}$$

(encode pairs with standard set-theoretic trickery)

Descriptions like this can be input to algorithms, for example:

Is 3-colorability of orbit-finite graphs decidable?

Set theory with atoms

Sets with atoms are a topos

A lot of mathematics can be done with atoms

set →	set with atoms
finite →	orbit-finite
function →	equivariant function

EXCEPT:

- axiom of choice fails, even orbit-finite choice
- powerset does not preserve orbit-finiteness

λX.(X with atoms)

A recipe for adding atoms to everything:

1. Take your favourite definition.
2. Replace all sets (relations, functions etc.)
   with sets with atoms (equivariant if you wish).
3. Replace every "finite" with "orbit-finite".
4. Check if your favourite theorems still hold.

(take with a pinch of salt)

Has been applied to: automata, grammars, Turing machines, while-programs, functional programs, CSPs, vector spaces, ...

Here: the μ-calculus.

III

μ-Calculus with Atoms

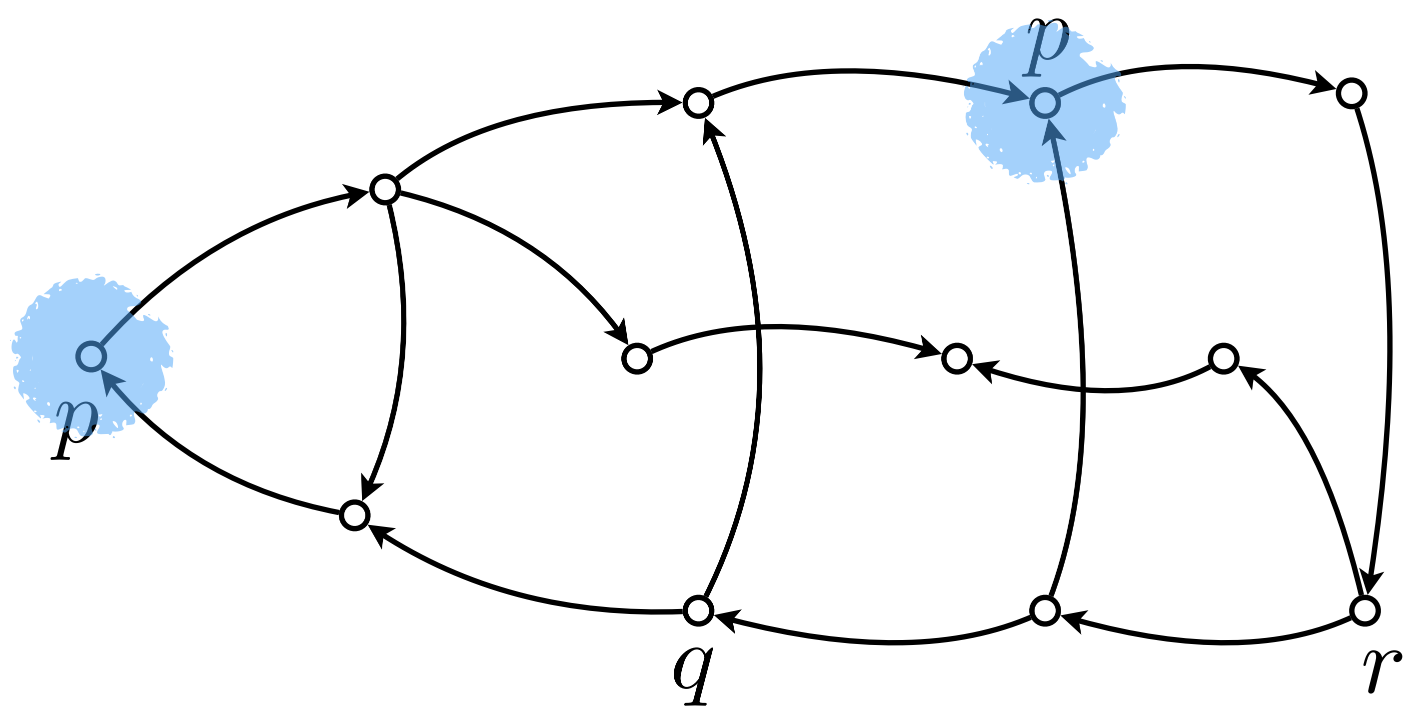
μ-calculus

$p, q, r, \ldots \in \mathbb{P}$

Formula: φ

p

Model: $\mathcal{K}$



Semantics:

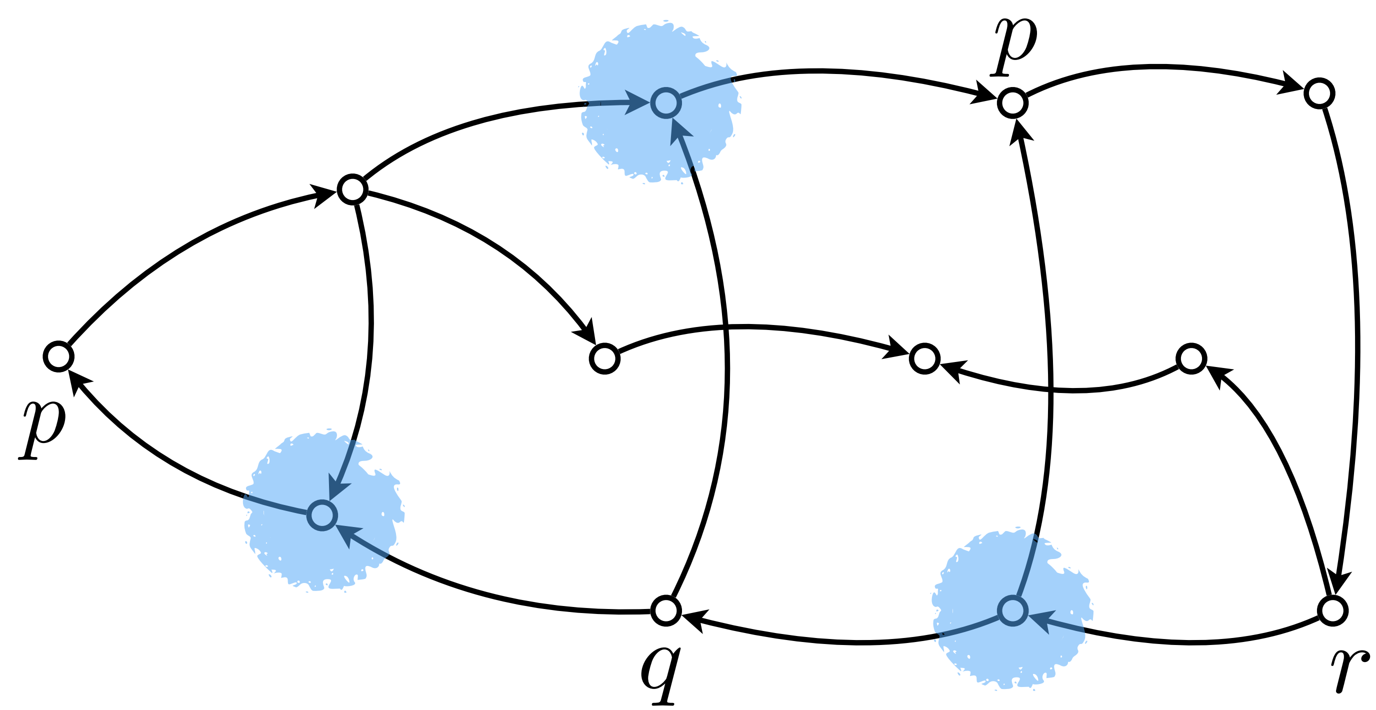
p holds now

μ-calculus

$p, q, r, \ldots \in \mathbb{P}$

Formula: φ $\Diamond p$

Model: $\mathcal{K}$



Semantics: p holds in some successor

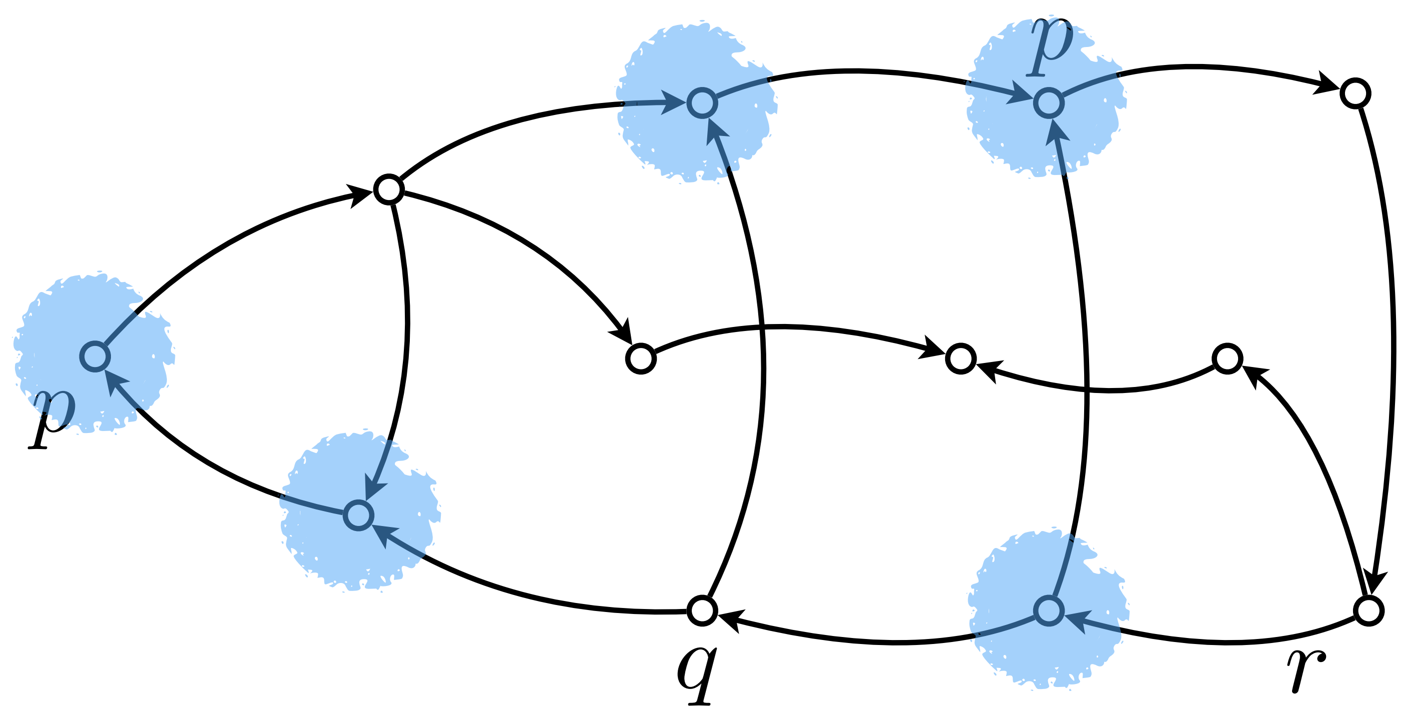
μ-calculus

$p, q, r, \ldots \in \mathbb{P}$

Formula: φ

$p \vee \Diamond p$

Model: $\mathcal{K}$



Semantics:

p holds now or in some successor

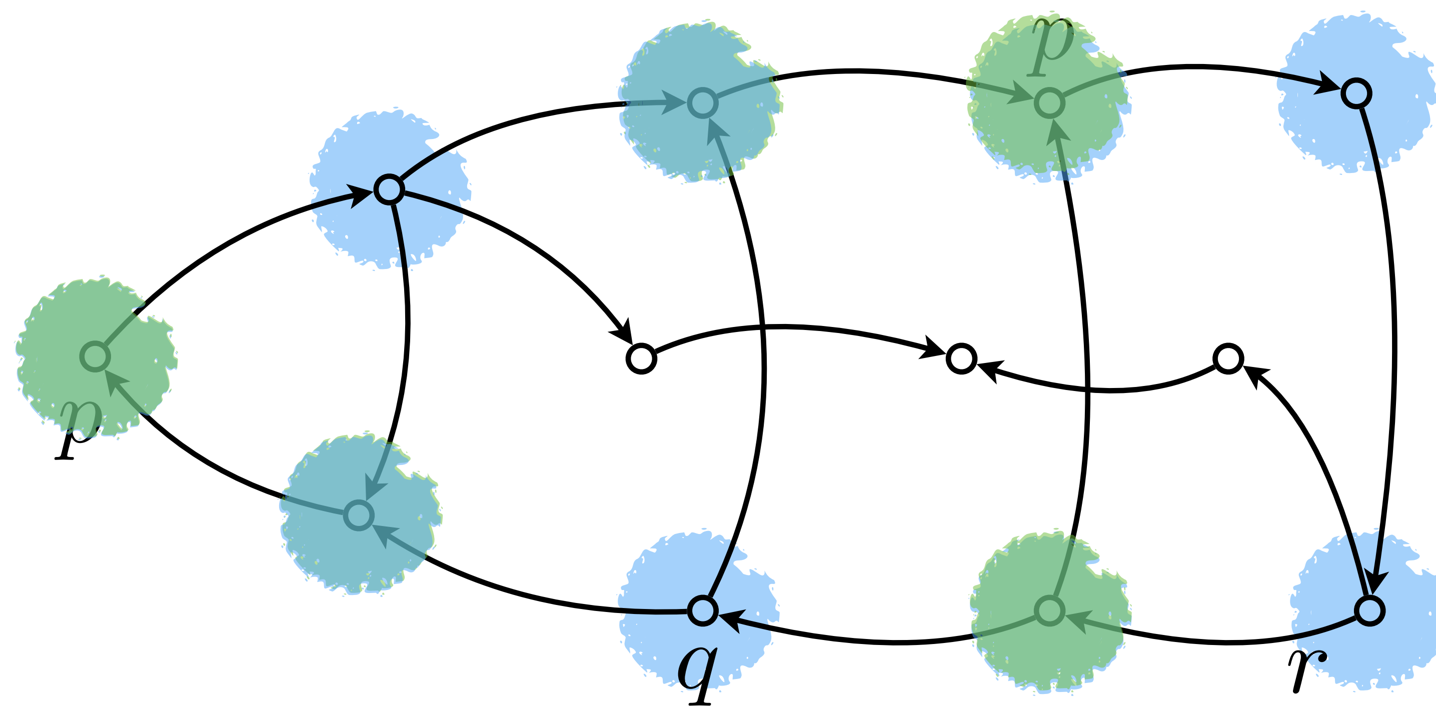
μ-calculus

$p, q, r, \ldots \in \mathbb{P}$

Formula: φ

$$\mu X.(p \vee \Diamond X)$$

Model: $\mathcal{K}$



Semantics:

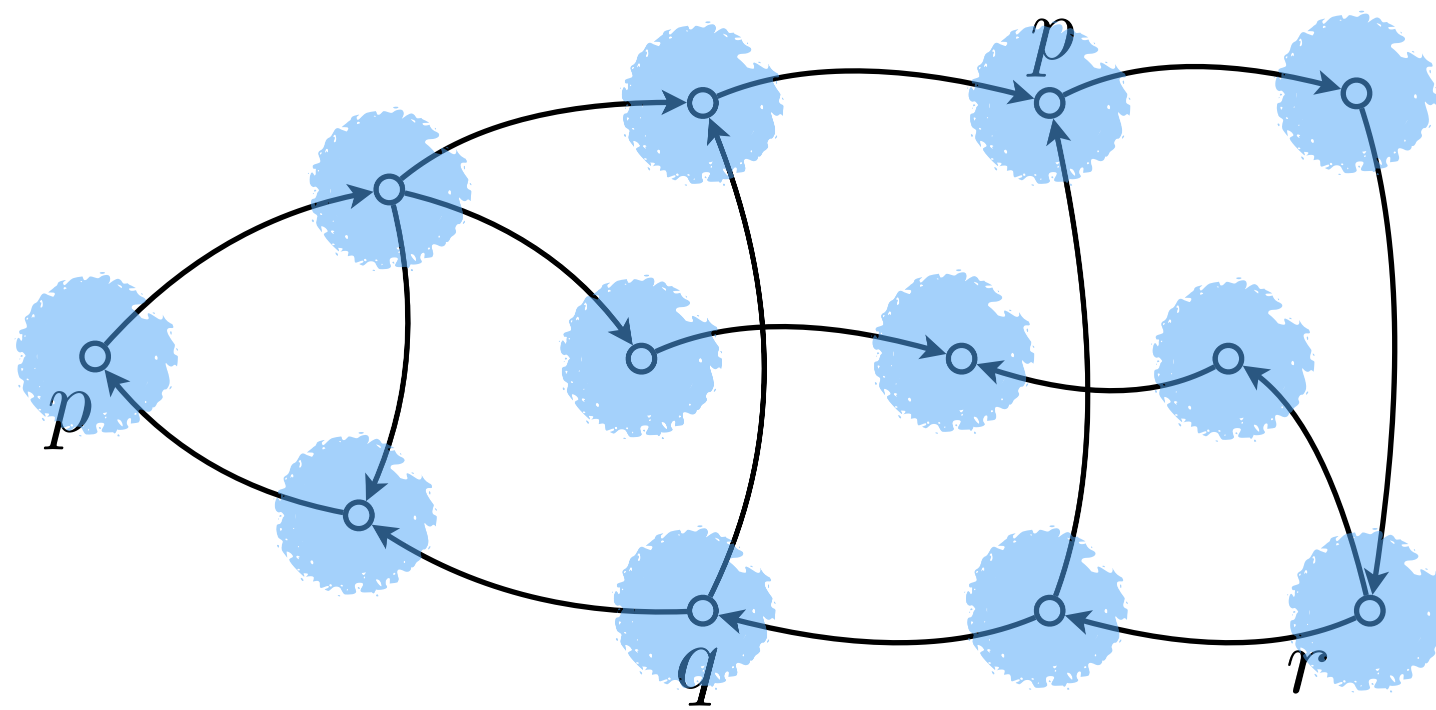
p holds in some future

μ-calculus

$p, q, r, \ldots \in \mathbb{P}$

Formula: φ $\qquad \nu X.(\neg p \wedge \Box X)$

Model: $\mathcal{K}$



Semantics:

p never holds in any future

Properties

Model checking:

parity games

Given $k \in \mathcal{K}$ and φ, does $\mathcal{K}, k \models \varphi$?

is decidable.

Satisfiability:

small model property

Given φ, are there $k \in \mathcal{K}$ s.t. $\mathcal{K}, k \models \varphi$?

is decidable.

Useful fragments, e.g. CTL*:

$$\Phi ::= p \mid \Phi \vee \Phi \mid \neg\Phi \mid \exists\phi$$

$$\phi ::= \Phi \mid \phi \vee \phi \mid \neg\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}\phi$$

state formulas

path formulas

Limitations

Consider an infinite set of basic predicates:

$$\mathbb{P} = \{p_0, p_1, p_2, \ldots\}$$

p_n: the number n has been input

Now let's define the property:

The current input number
is input again in some future

$$\bigvee_{n \in \mathbb{N}} (p_n \wedge \Diamond \mu X.(p_n \vee \Diamond X))$$

Problem: infinite disjunction

Practical motivation:

*The system never crashes**

* *unless the password generator generates the same password twice...*

Models with atoms

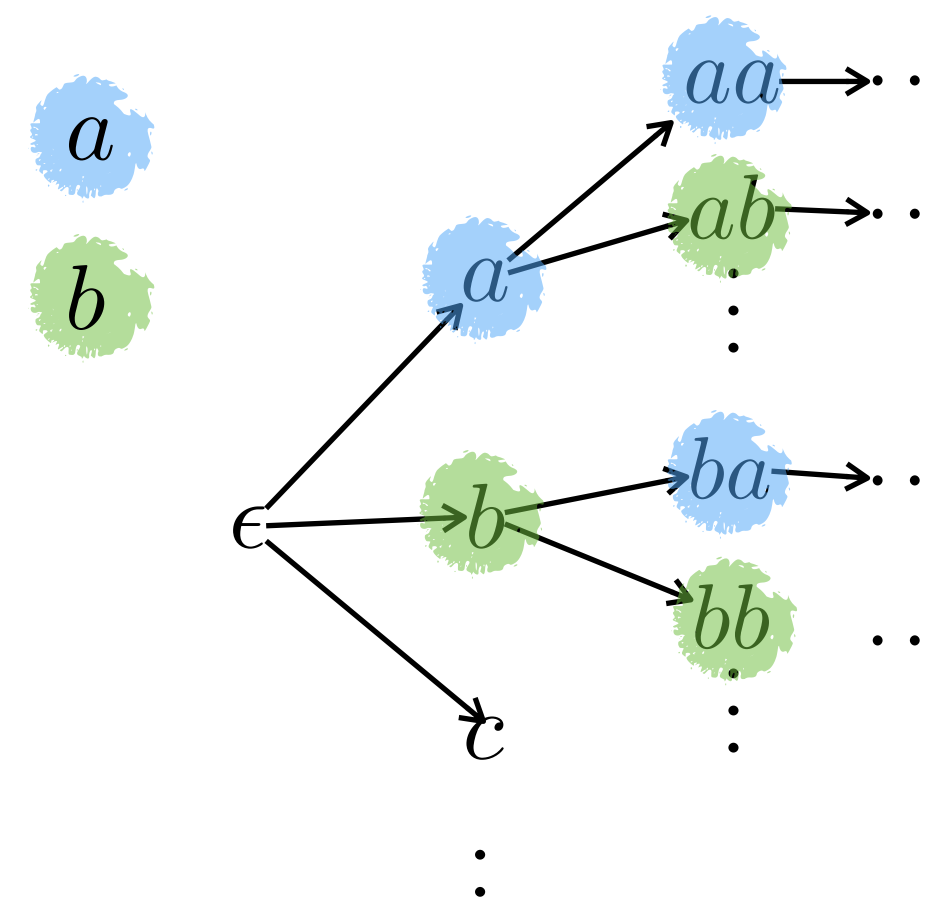
Fix an equivariant set $\mathbb{P}$ of basic predicates. (think $\mathbb{P} = \mathbb{A}$)

A model (with atoms): $\mathcal{K} = (K, \rightarrow, \mathrm{pred})$

- a set with atoms K,
- a finitely supported relation $\rightarrow \subseteq K \times K$,
- a finitely supported function $\mathrm{pred} : K \rightarrow \mathcal{P}_{\mathrm{fs}}\mathbb{P}$,

Example:

- $K = \mathbb{A}^*$,
- $w \rightarrow wa$ for $w \in \mathbb{A}^*, a \in \mathbb{A}$
- $\mathrm{pred}(a_1 a_2 \cdots a_n) = \{a_n\}$



Syntax

Formulas of $\mathcal{L}_\mu^{\mathbb{A}}$:

positive formula

$$\phi ::= p \mid X \mid \bigvee \Phi \mid \neg\phi \mid \Diamond\phi \mid \mu X.\phi$$

orbit-finite disjunction

we write e.g.

$\bigvee_{a\in\mathbb{A}} \phi_a$ for $\bigvee\{\phi_a \mid a \in \mathbb{A}\}$

Standard abbreviations:

$$\top := p \vee \neg p \qquad \bigwedge \Phi := \neg \bigvee\{\neg\phi \mid \phi \in \Phi\}$$

$$\Box\phi := \neg\Diamond\neg\phi \qquad \nu X.\phi := \neg\mu X.\neg\phi[{}^{\neg X}/{}_X]$$

Example: $\bigvee_{a\in\mathbb{A}}(a \wedge \Diamond\mu X.(a \vee \Diamond X))$

Semantics

For a formula ϕ and a model $\mathcal{K}$

(and a valuation $\rho : \mathit{Variables} \to \mathcal{P}_{\mathrm{fs}} K$)

define $[\![\phi]\!]_\rho \subseteq K$ by induction:

- $[\![p]\!]_\rho = \{x \in K \mid p \in \mathsf{pred}(x)\}$,
- $[\![X]\!]_\rho = \rho(X)$,
- $[\![\neg\phi]\!]_\rho = K \setminus [\![\phi]\!]_\rho$,
- $[\![\bigvee \Phi]\!]_\rho = \bigcup\{[\![\phi]\!]_\rho \mid \phi \in \Phi\}$,
- $[\![\Diamond\phi]\!]_\rho = \{k \in K \mid \exists s \in [\![\phi]\!]_\rho.\ k \to s\}$,
- $[\![\mu X.\phi]\!]_\rho = \mathrm{lfp}(F), \mathrm{where}\ F(A) = [\![\phi]\!]_{\rho[X \mapsto A]}$.

Examples

- $\bigvee_{a\in\mathbb{A}}(a \wedge \Diamond\mu X.(a \vee \Diamond X))$

some predicate that holds now, holds again in some future

- $\nu X.\,((\Diamond \bigvee_{a\in\mathbb{A}} a) \wedge \Box X)$

every reachable state has some successor for which some basic predicate holds

- $\neg(\mu X.(\psi \vee \Diamond X))$

  $\psi = \bigvee_{a\in\mathbb{A}}(a \wedge \Diamond\mu Y.(a \vee \Diamond Y))$

on every path,
no basic predicate holds more than once

Properties

Fact: Model checking on orbit-finite models is decidable.

(proof: direct computation of semantics, including fixpoints)

Fact: Satisfiability is undecidable.

(proof: direct encoding of Turing machine computations)

CTL* with atoms:

$$\Phi ::= p \mid \bigvee_a \Phi_a \mid \neg\Phi \mid \exists\phi \qquad \phi ::= \Phi \mid \bigvee_a \phi_a \mid \neg\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}\phi$$

Fact: This is not a fragment of $\mathcal{L}_\mu^{\mathbb{A}}$.

(and it has undecidable model checking)

The fresh path property

The property:

on some path
no basic predicate holds more
than once

is not expressible.

Note:

on every path
no basic predicate holds more
than once

is expressible: $\neg(\mu X.(\psi \vee \Diamond X))$

$$\psi = \bigvee_{a \in \mathbb{A}} (a \wedge \Diamond \mu Y.(a \vee \Diamond Y))$$

The history-dependent μ-calculus

Extend the syntax:

$$\phi ::= p \mid X \mid \sharp a \mid \bigvee \Phi \mid \neg\phi \mid \Diamond\phi \mid \mu X.\phi$$

Idea: $\sharp a$ says " a has never appeared in any predicate so far"

Semantics evaluated in the context of a history $H \subseteq_{\text{fin}} \mathbb{A}$:

$$x \in [\![\sharp a]\!]^H_\rho \iff a \notin H$$
$$x \in [\![\Diamond\phi]\!]^H_\rho \iff \exists y \in [\![\phi]\!]^{H \cup \mathsf{pred}(x)}_\rho \text{ s.t. } x \to y$$

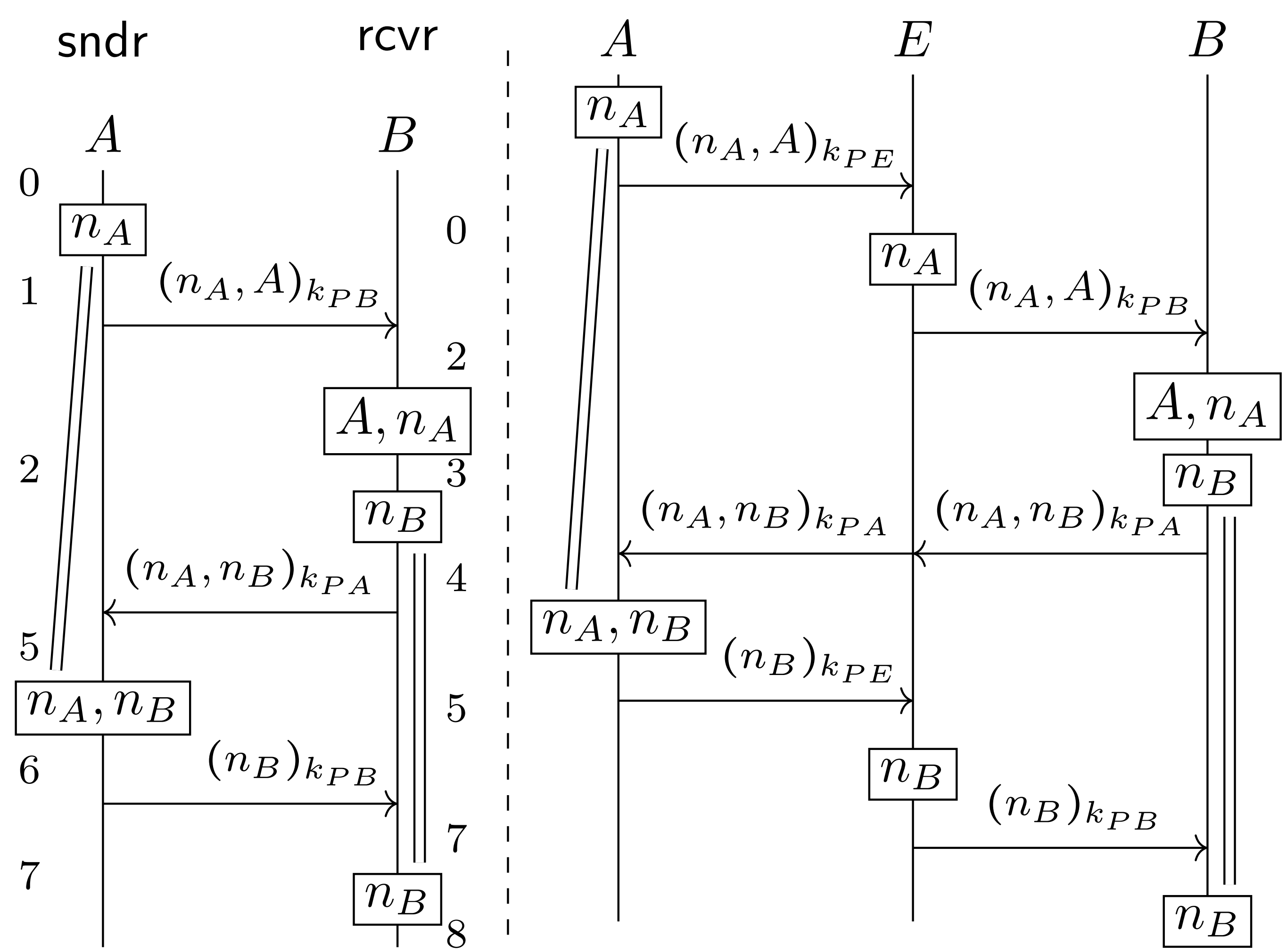
This expresses the fresh path property:

$$\nu X. \left(\bigwedge_{a \in \mathbb{A}} (a \to \sharp a) \wedge \Diamond X \right)$$

Fact: Model checking on orbit-finite models still decidable.

Application

The Needham-Schroeder public-key protocol:



The system (consisting of Alice, Bob and Eve) is represented as an orbit-finite model, and its security is expressed as a formula.

(which fails)

λX.(X with atoms)

A recipe for adding atoms to everything:

1. Take your favourite definition.
2. Replace all sets (relations, functions etc.) with sets with atoms (equivariant if you wish).
3. Replace every "finite" with "orbit-finite".
4. Check if your favourite theorems still hold.

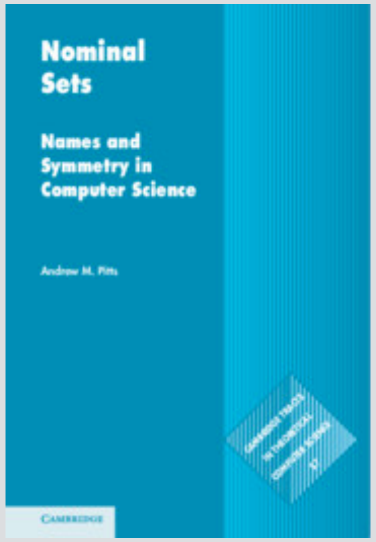
(take with a pinch of salt)

Further reading

Books:

- A. Pitts: *Nominal sets. Names and symmetry in Computer Science*

  Cambridge Univ. Press, 2013



- M. Bojańczyk: *Slightly infinite sets*

  to appear, available online:

  https://www.mimuw.edu.pl/~bojan/paper/atom-book