FoPSS 2019

3rd Summer School
on Foundations of Programming
and Software Systems

Nominal Techniques

Warsaw, 10-15 September, 2019

FoPSS

Summer Schools on Foundations of Programming and Software Systems

Supported by:

This time also by:

FoPSS

2017: Braga (Portugal)

Probabilistic Programming

FoPSS

2017: Braga (Portugal)

Probabilistic Programming

2018: Oxford (UK)

Logic and Learning

FoPSS

2017: Braga (Portugal)

Probabilistic Programming

2018: Oxford (UK)

Logic and Learning

2019: Warsaw (Poland)

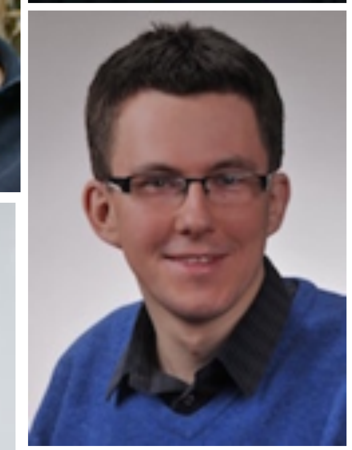
Nominal Techniques

FoPSS 2019

Our lecturers:

- Andrew M. Pitts:
  Nominal sets and functional programming
- Mikołaj Bojańczyk:
  Computation theory with atoms
- Andrzej Murawski:
  Nominal game semantics
- Maribel Fernández:
  Nominal rewriting and unification
- Johannes Borgström:
  Nominal process calculi and modal logics
- Murdoch J. Gabbay:
  Advanced nominal techniques
- Sławomir Lasota:
  Computation theory with atoms II







FoPSS 2019

[A]

Basic Nominal Techniques

Bartek Klin
University of Warsaw

Warsaw, 10-11 September, 2019

What is it all about?

Nominal techniques:

mathematics of, and computation with:

local names
and name dependence

What is it all about?

Nominal techniques:

mathematics of, and computation with:

local names
and name dependence

highly symmetrical
structures

What is it all about?

Nominal techniques:

mathematics of, and computation with:

local names
and name dependence

highly symmetrical
structures

“slightly infinite”
structures

What is it all about?

Nominal techniques:

mathematics of, and computation with:

local names
and name dependence

highly symmetrical
structures

“slightly infinite”
structures

structures
acessible via
limited interfaces

What is it all about?

Nominal techniques:

mathematics of, and computation with:

local names
and name dependence

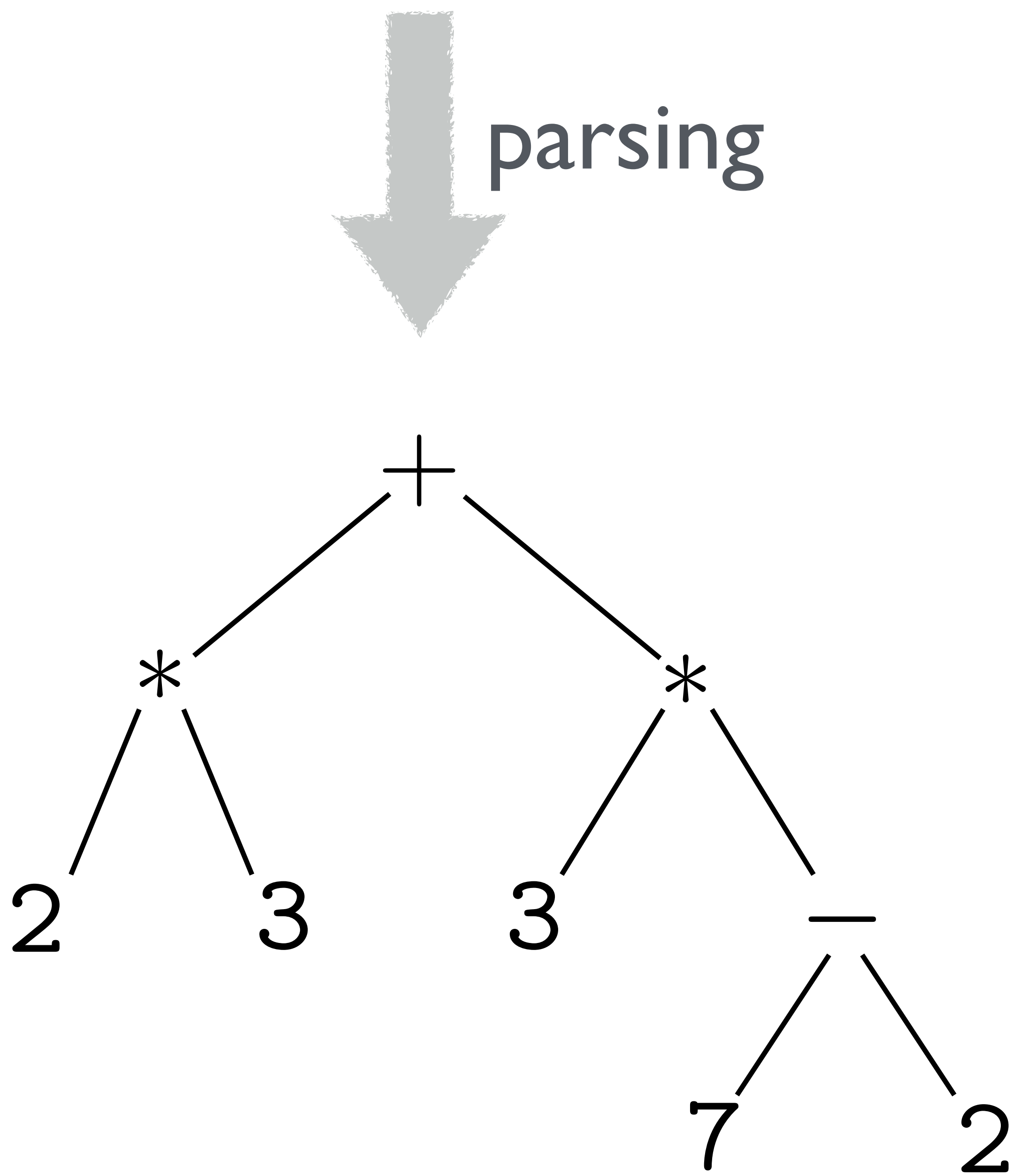
highly symmetrical
structures

“slightly infinite”
structures

structures
acessible via
limited interfaces

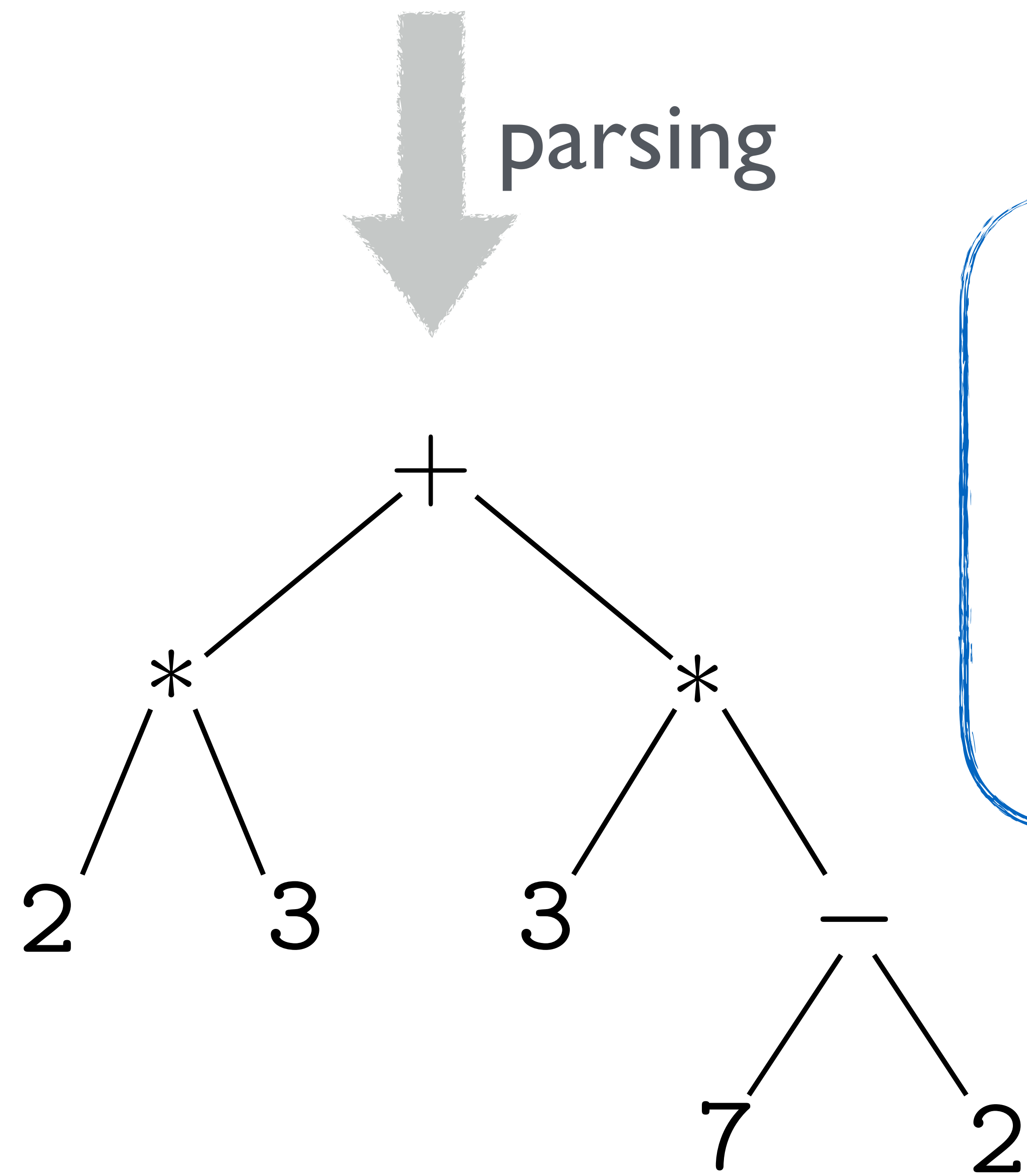
Concrete and abstract syntax

$$2 * 3 + 3 * (7 - 2)$$



Concrete and abstract syntax

$$2 * 3 + 3 * (7 - 2)$$

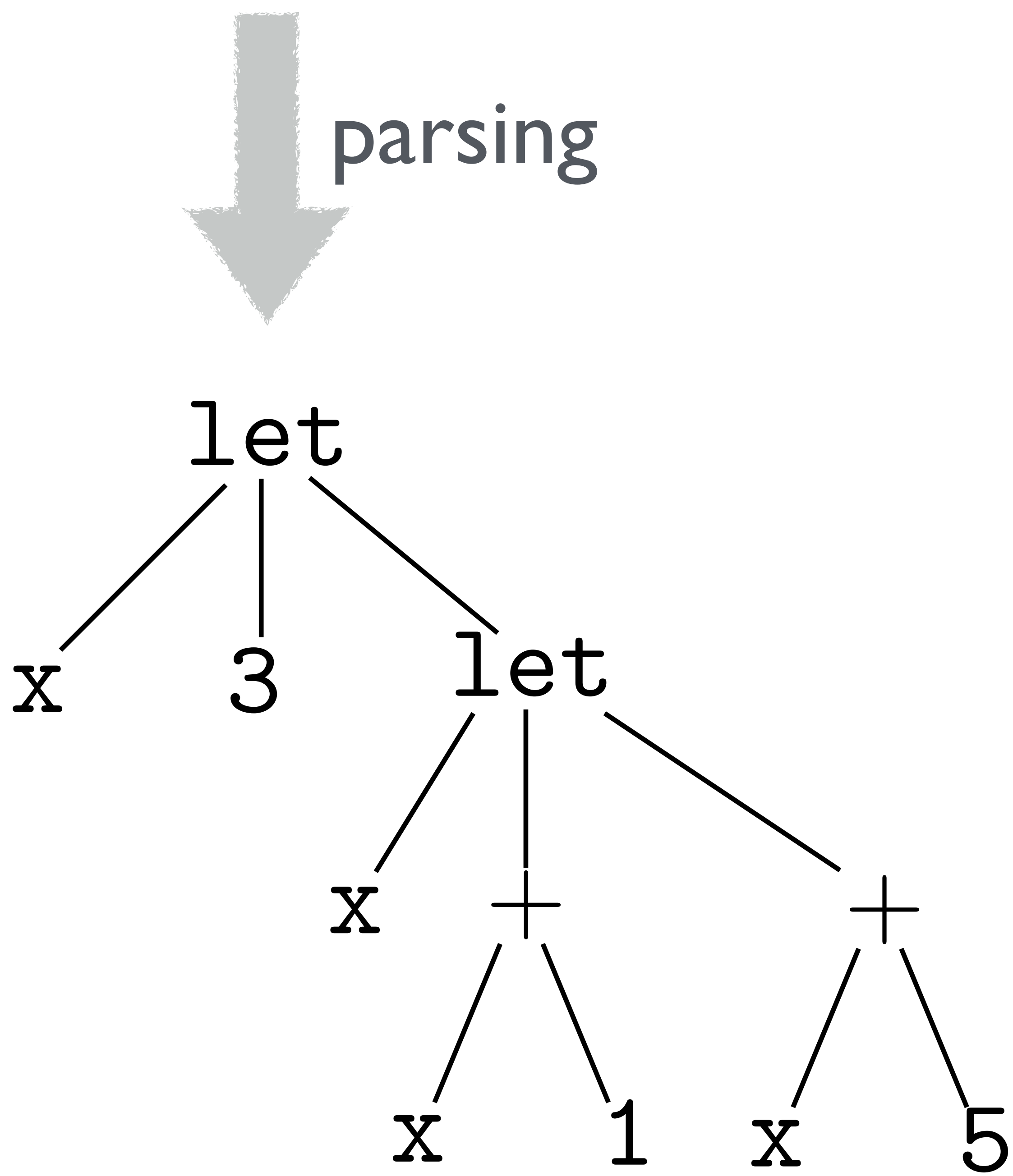


Algebraic features:

- definitions by recursion
- proofs by induction
- ...

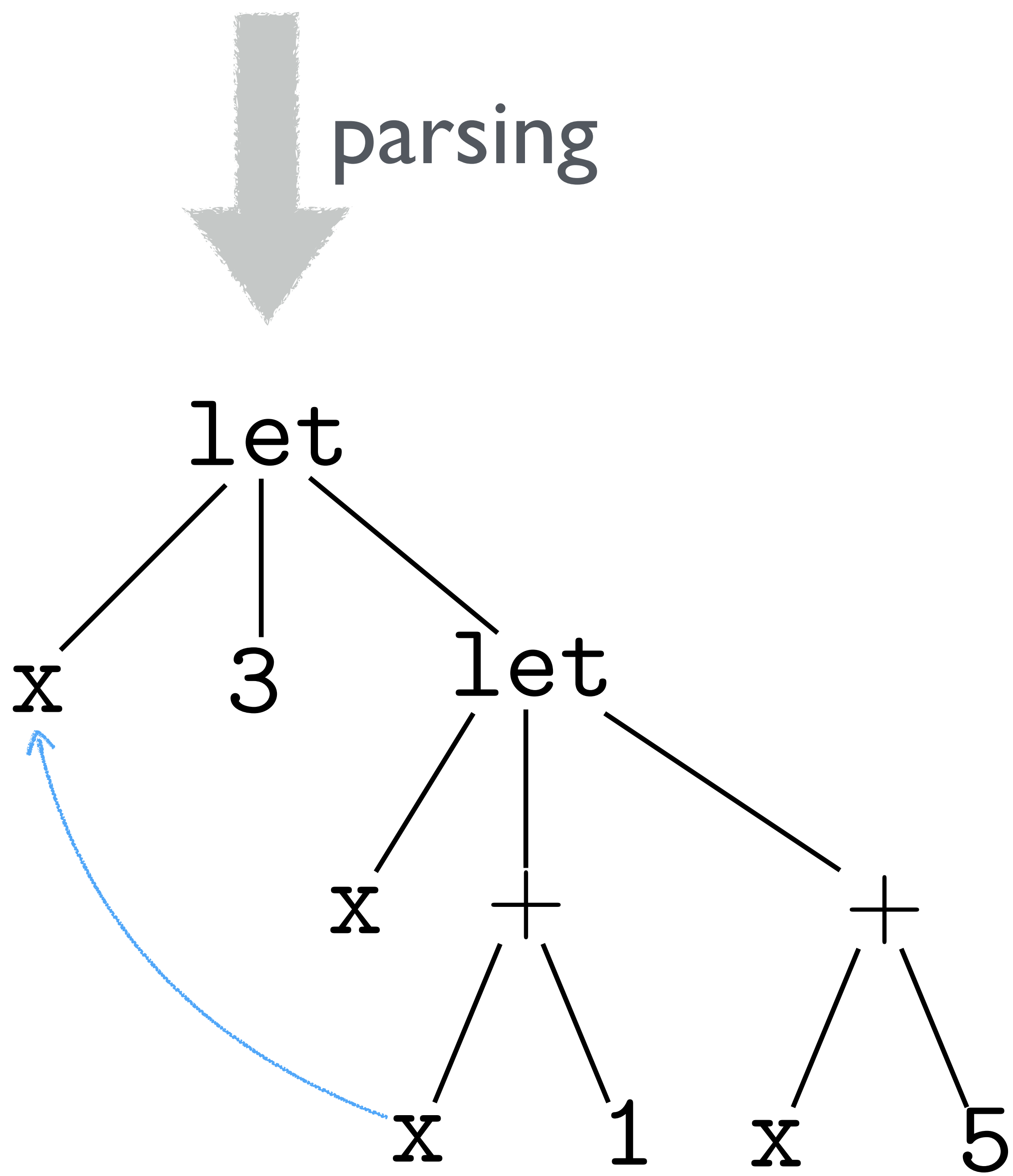
Complications with local names

$$\texttt{let x} = 3 \texttt{ in let x} = \texttt{x} + 1 \texttt{ in x} + 5$$



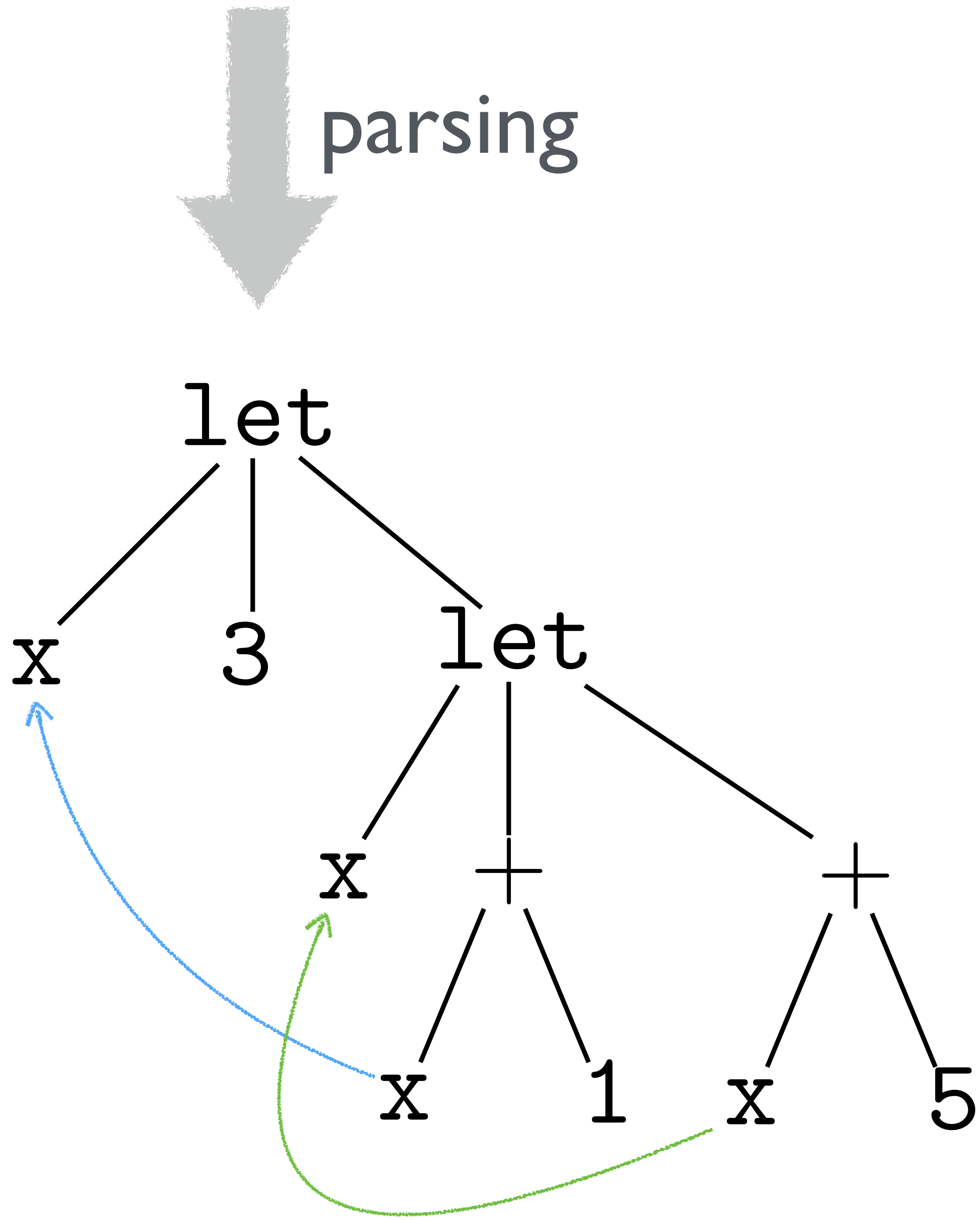
Complications with local names

$$\texttt{let x} = 3 \texttt{ in let x} = \texttt{x} + 1 \texttt{ in x} + 5$$



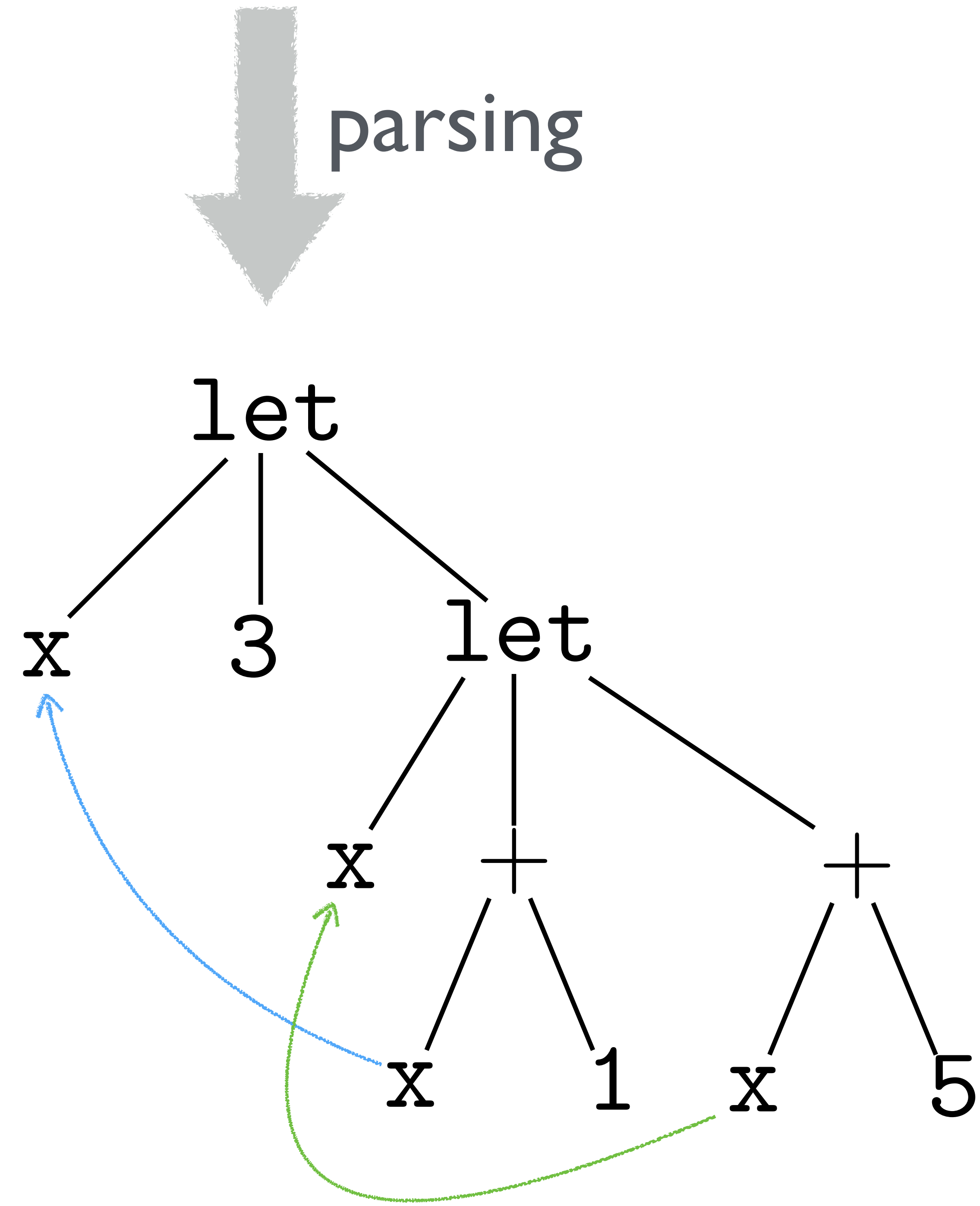
Complications with local names

$$\texttt{let x} = \texttt{3 in let x} = \texttt{x} + \texttt{1 in x} + \texttt{5}$$



Complications with local names

$$\texttt{let x} = 3\ \texttt{in let x} = \texttt{x} + 1\ \texttt{in x} + 5$$



Expressions depend on names!

Name dependence

Idea:

Let every expression come equipped
with an explicit dependence on some names
(or: atoms) that occur in it

Name dependence

Idea: nominal expressions

Let every expression come equipped with an explicit dependence on some names (or: atoms) that occur in it

Name dependence

Idea: nominal expressions

Let every expression come equipped
with an explicit dependence on some names
(or: atoms) that occur in it

More ambitious idea:

Let everything come equipped
with an explicit dependence on some names
(or: atoms) that occur in it

Name dependence

Idea: nominal expressions

Let every expression come equipped
with an explicit dependence on some names
(or: atoms) that occur in it

More ambitious idea: nominal sets

Let everything come equipped
with an explicit dependence on some names
(or: atoms) that occur in it

Name dependence

Q:

What does it mean to depend on a name?

Name dependence

Q:

What does it mean to depend on a name?

A:

X depends on a name a
if renaming a to any other name
would alter X

Name dependence

Q:

What does it mean to depend on a name?

A:

X depends on a name a
if renaming a to any other name
would alter X

Idea revisited: nominal sets

Let everything come equipped
with information on how renaming names
affects it

What is it all about?

Nominal techniques:

mathematics of, and computation with:

local names
and name dependence

highly symmetrical
structures

“slightly infinite”
structures

structures
acessible via
limited interfaces

A graph built of atoms

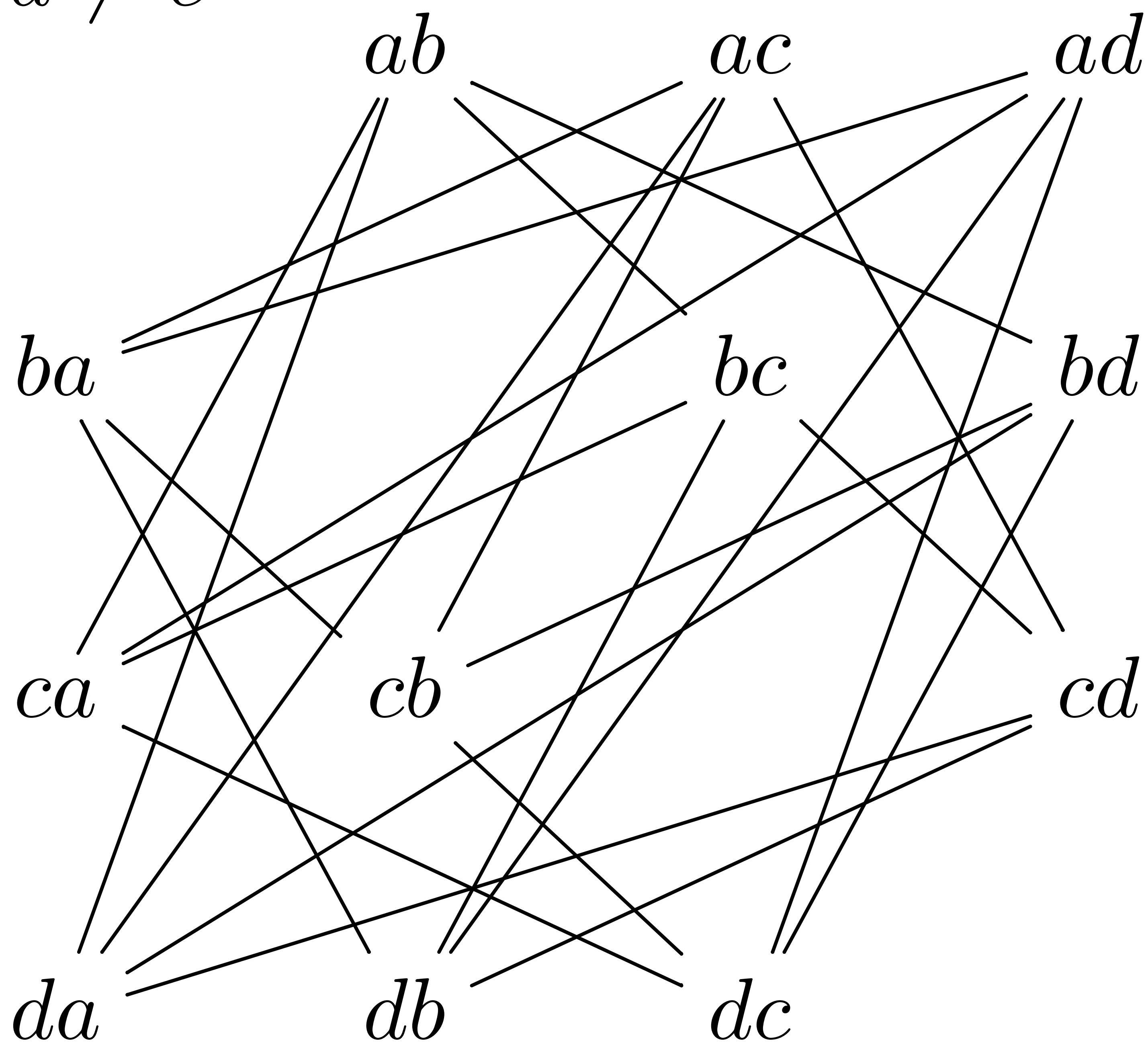
atomic names: $a, b, c, d, e, \ldots$

- nodes: ab $\quad a \neq b$
- edges: ab—bc $\quad a \neq c$

A graph built of atoms

atomic names: $a, b, c, d, e, \ldots$

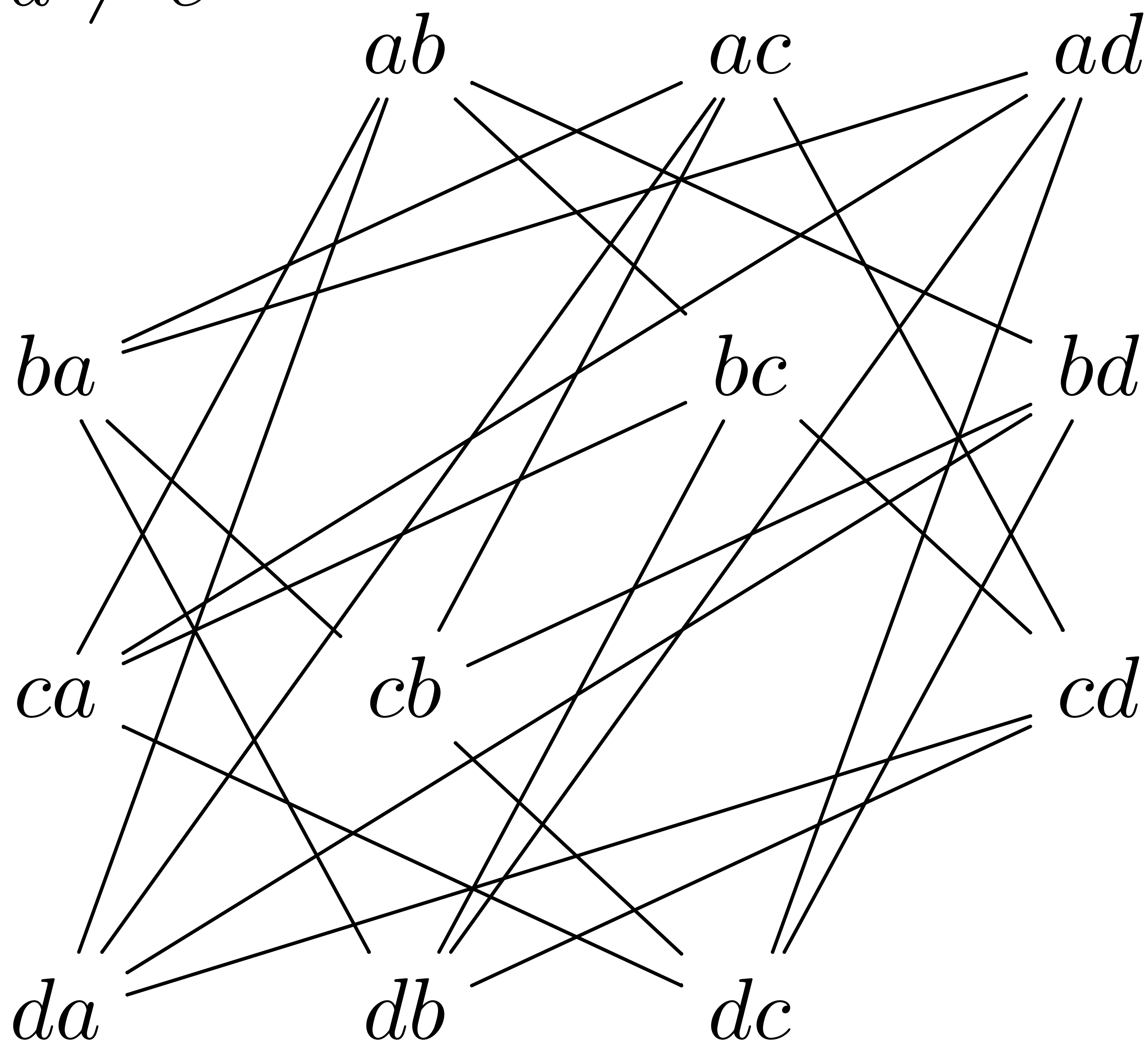
- nodes: ab $\quad a \neq b$
- edges: $ab—bc$ $\quad a \neq c$



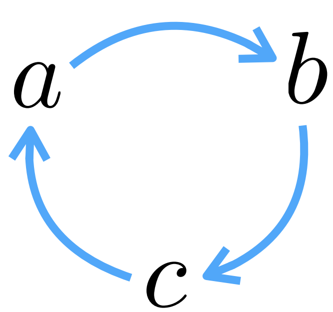
A graph built of atoms

atomic names: $a, b, c, d, e, \ldots$

- nodes: ab $\quad a \neq b$
- edges: ab—bc $\quad a \neq c$



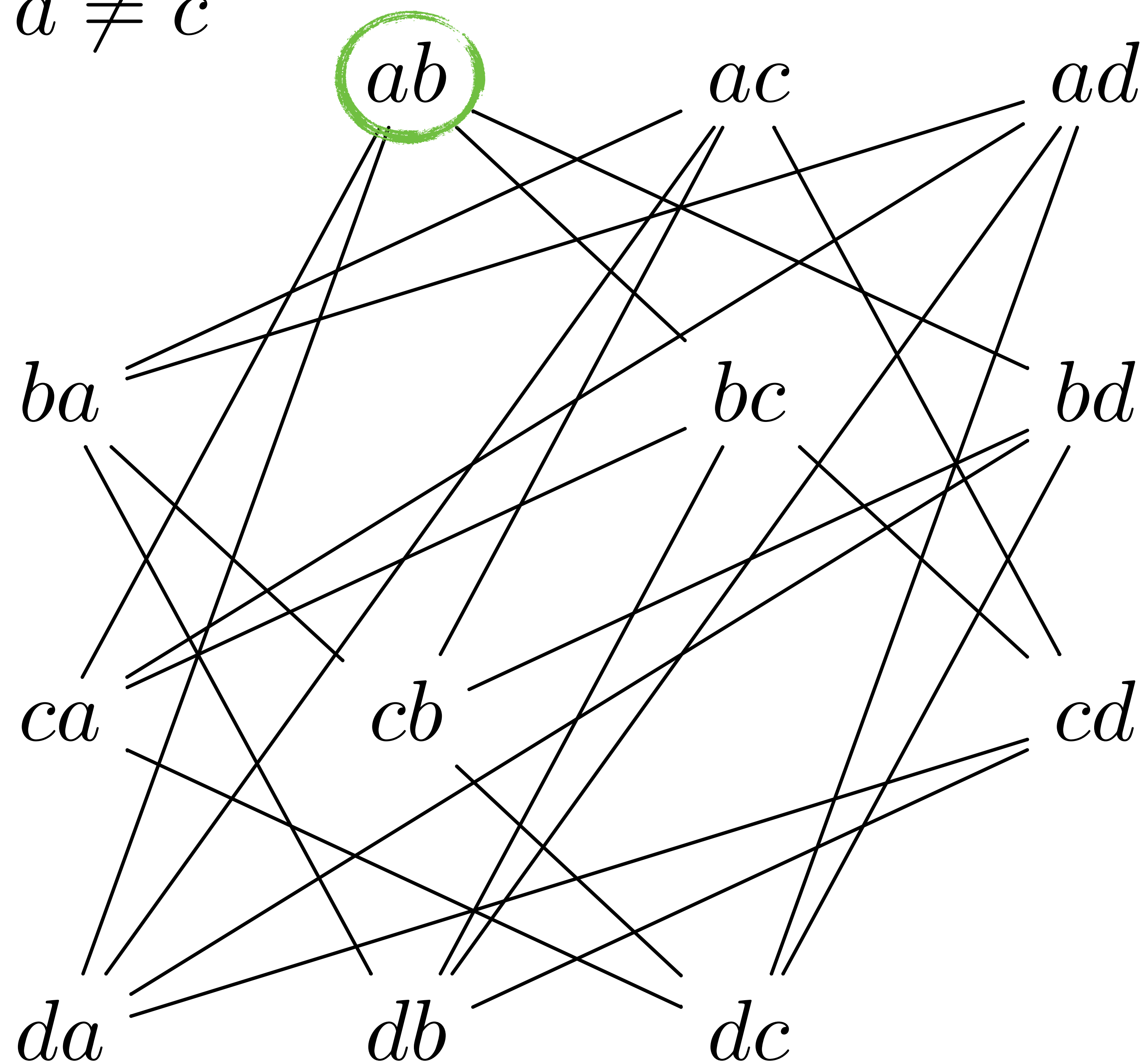
atom renaming:



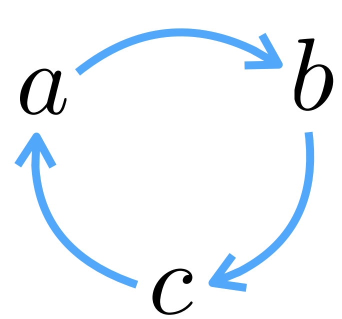
A graph built of atoms

atomic names: $a, b, c, d, e, \ldots$

- nodes: ab $\quad a \neq b$
- edges: ab—bc $\quad a \neq c$



atom renaming:

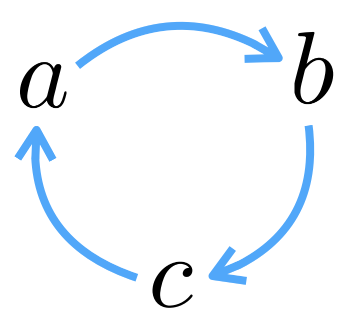


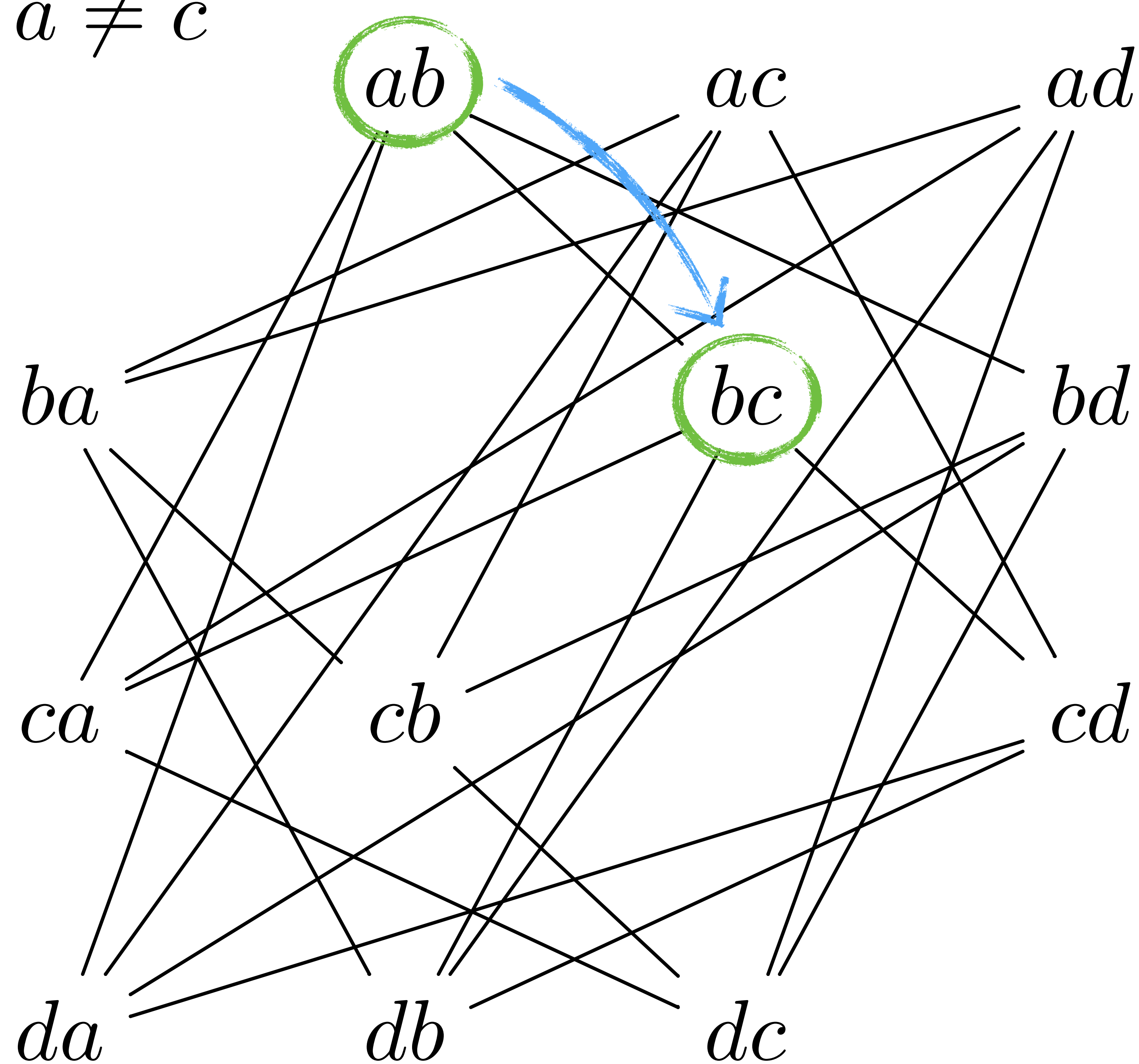
A graph built of atoms

atomic names: $a, b, c, d, e, \ldots$

- nodes: ab $\quad a \neq b$
- edges: ab—bc $\quad a \neq c$

atom renaming:

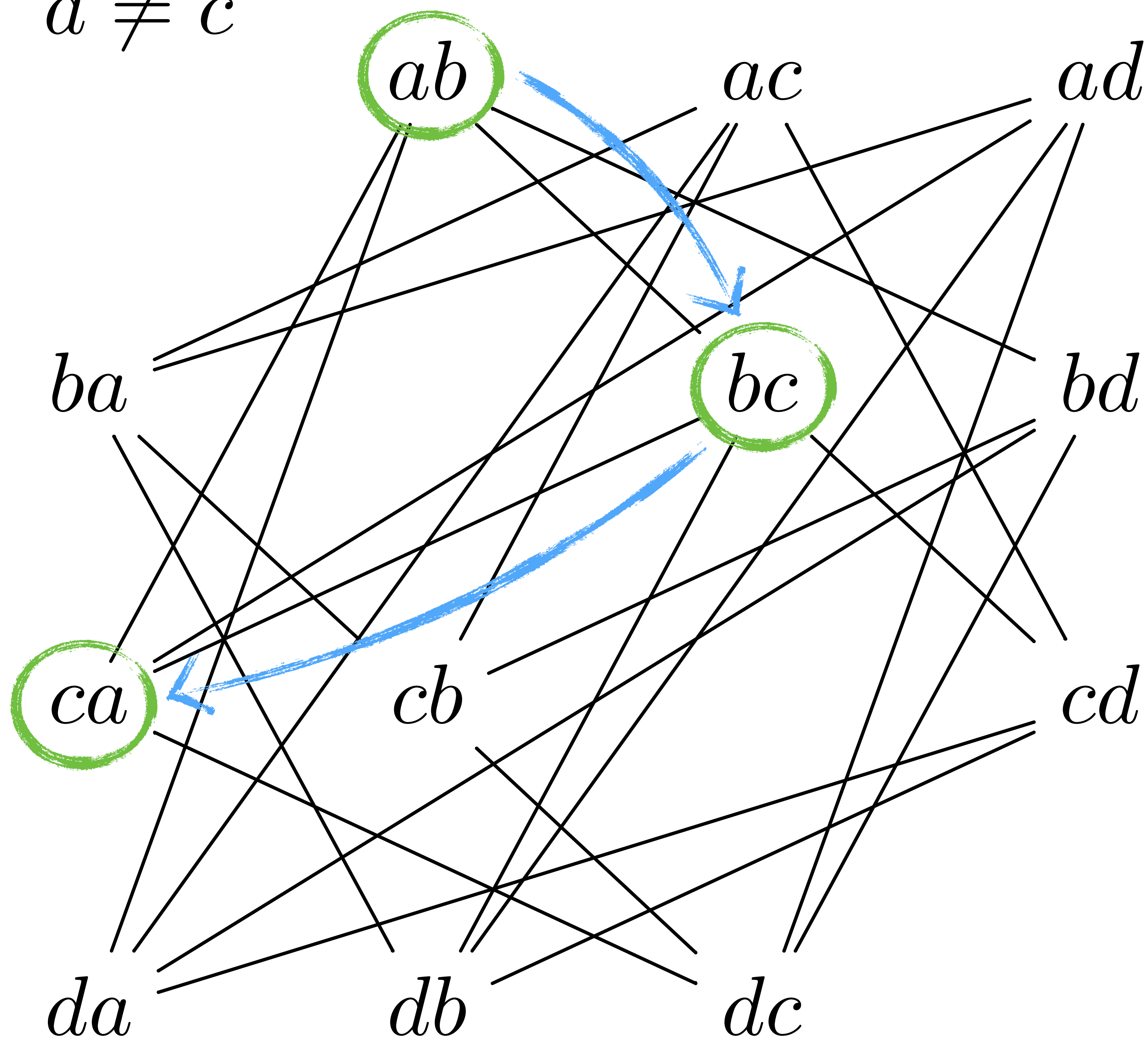




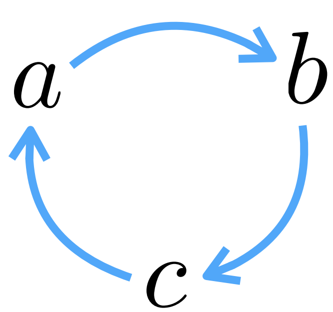
A graph built of atoms

atomic names: $a, b, c, d, e, \ldots$

- nodes: $ab \qquad a \neq b$
- edges: $ab — bc \qquad a \neq c$



atom renaming:

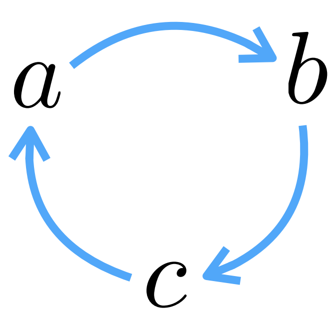


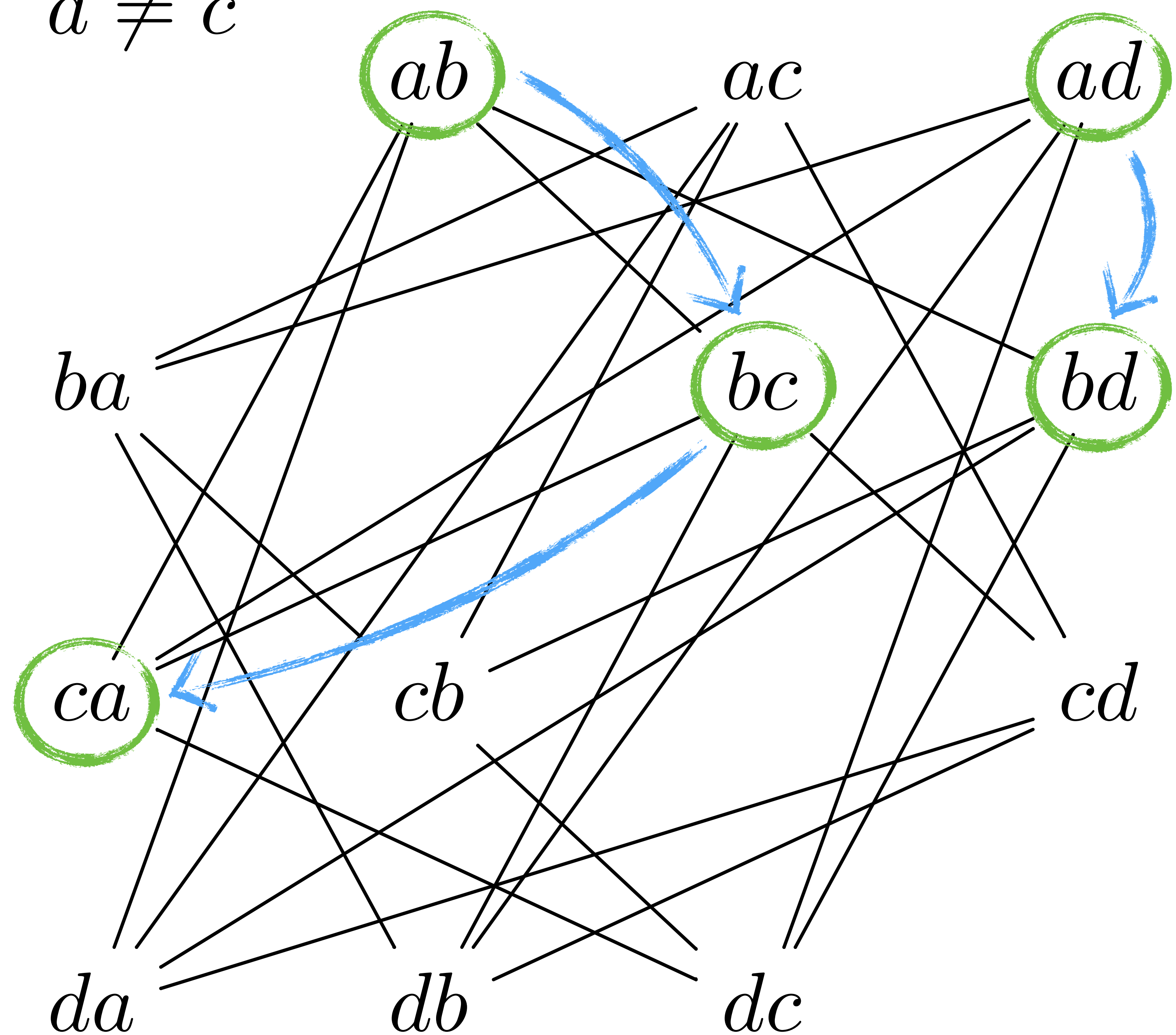
A graph built of atoms

atomic names: $a, b, c, d, e, \ldots$

- nodes: ab $\quad a \neq b$
- edges: ab—bc $\quad a \neq c$

atom renaming:

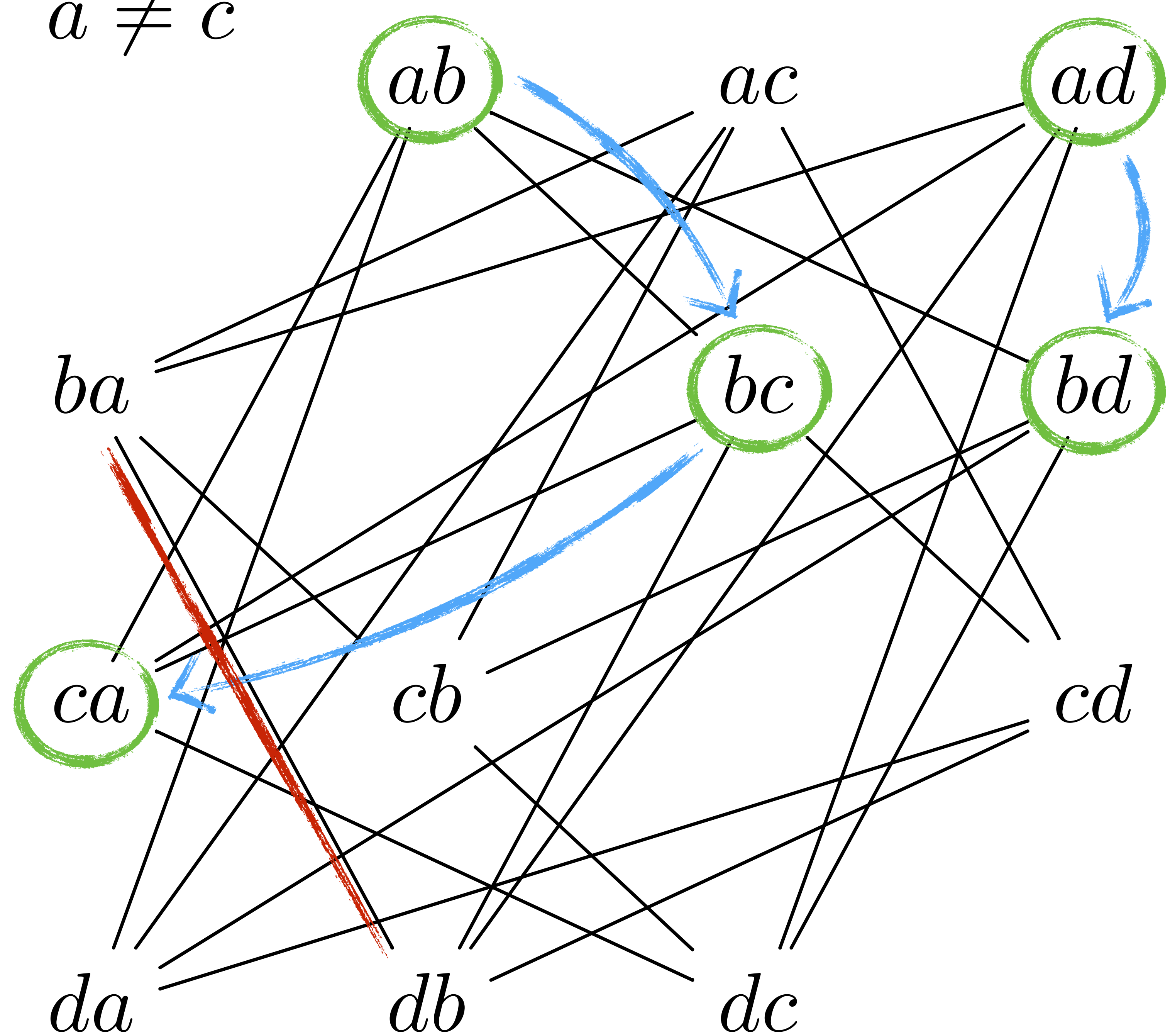




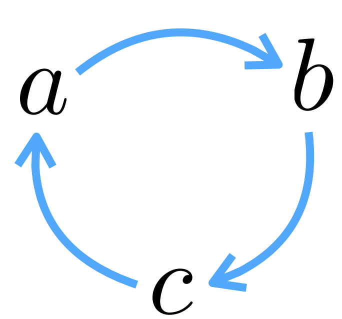
A graph built of atoms

atomic names: $a, b, c, d, e, \ldots$

- nodes: ab $\quad a \neq b$
- edges: ab—bc $\quad a \neq c$



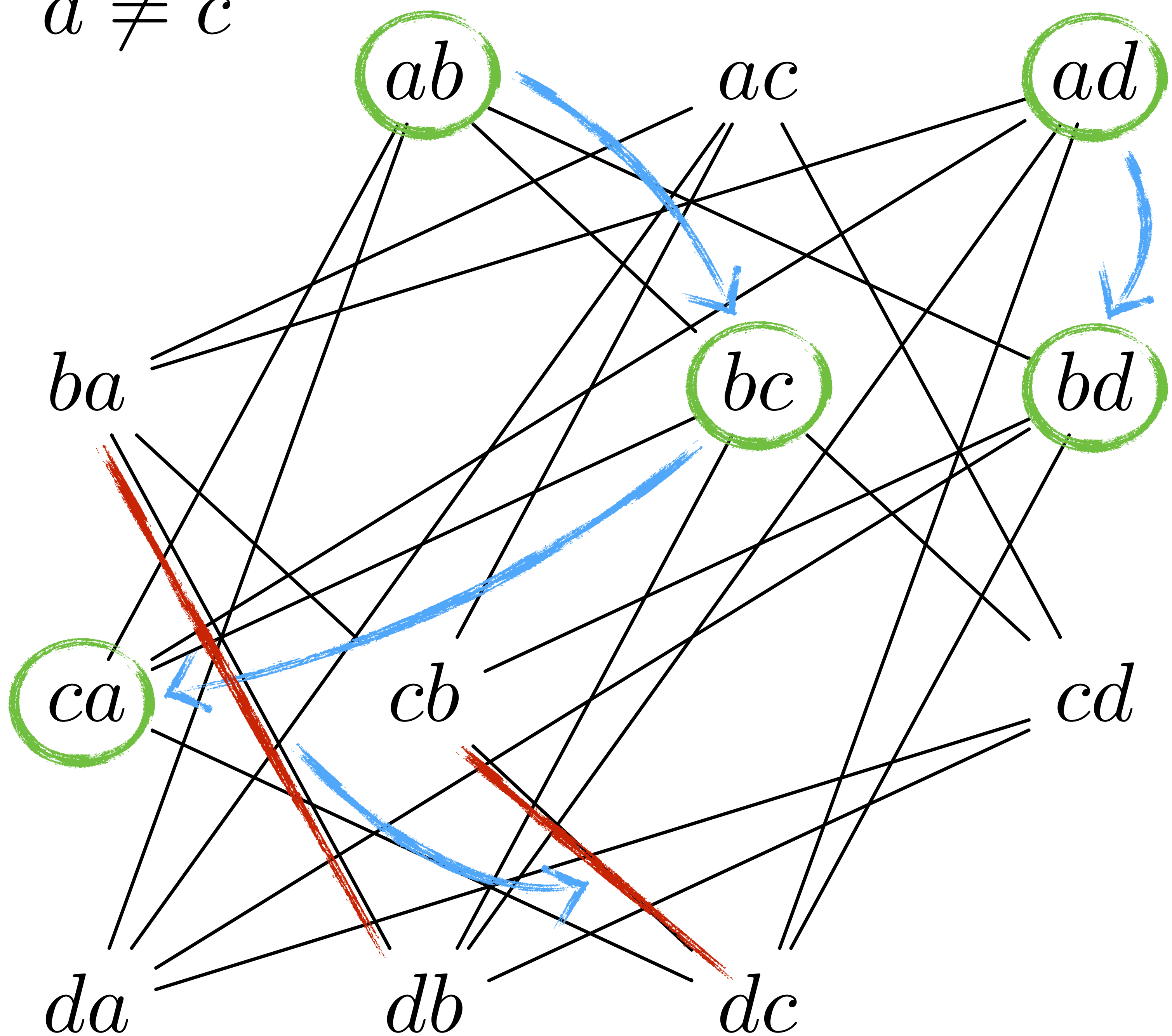
atom renaming:



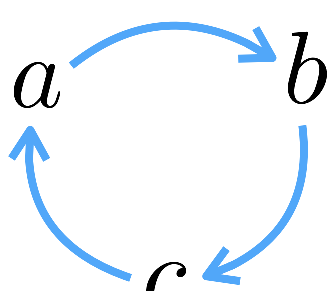
A graph built of atoms

atomic names: $a, b, c, d, e, \ldots$

- nodes: ab $\quad a \neq b$
- edges: ab—bc $\quad a \neq c$



atom renaming:



What is it all about?

Nominal techniques:

mathematics of, and computation with:

local names
and name dependence

highly symmetrical
structures

“slightly infinite”
structures

structures
acessible via
limited interfaces

Slightly infinite

The same graph:

- nodes: ab $\qquad a \neq b$
- edges: ab—bc $\qquad a \neq c$

Slightly infinite

The same graph:

- nodes: ab $a \neq b$
- edges: ab—bc $a \neq c$

- nodes: $\{(a,b) : a,b \in \mathbb{A} : a \neq b\}$
- edges: $\big\{\{(a,b),(b,c)\} : a,b,c \in \mathbb{A}$

  $: a \neq b \wedge b \neq c \wedge a \neq c\big\}$

Slightly infinite

The same graph:

- nodes: ab $a \neq b$
- edges: ab—bc $a \neq c$

- nodes: $\{(a,b) : a, b \in \mathbb{A} : a \neq b\}$
- edges: $\{\{(a,b),(b,c)\} : a,b,c \in \mathbb{A}$
$: a \neq b \wedge b \neq c \wedge a \neq c\}$

Infinite, but presented by finite means

An example problem

- nodes: ab $a \neq b$
- edges: ab—bc $a \neq c$

An example problem

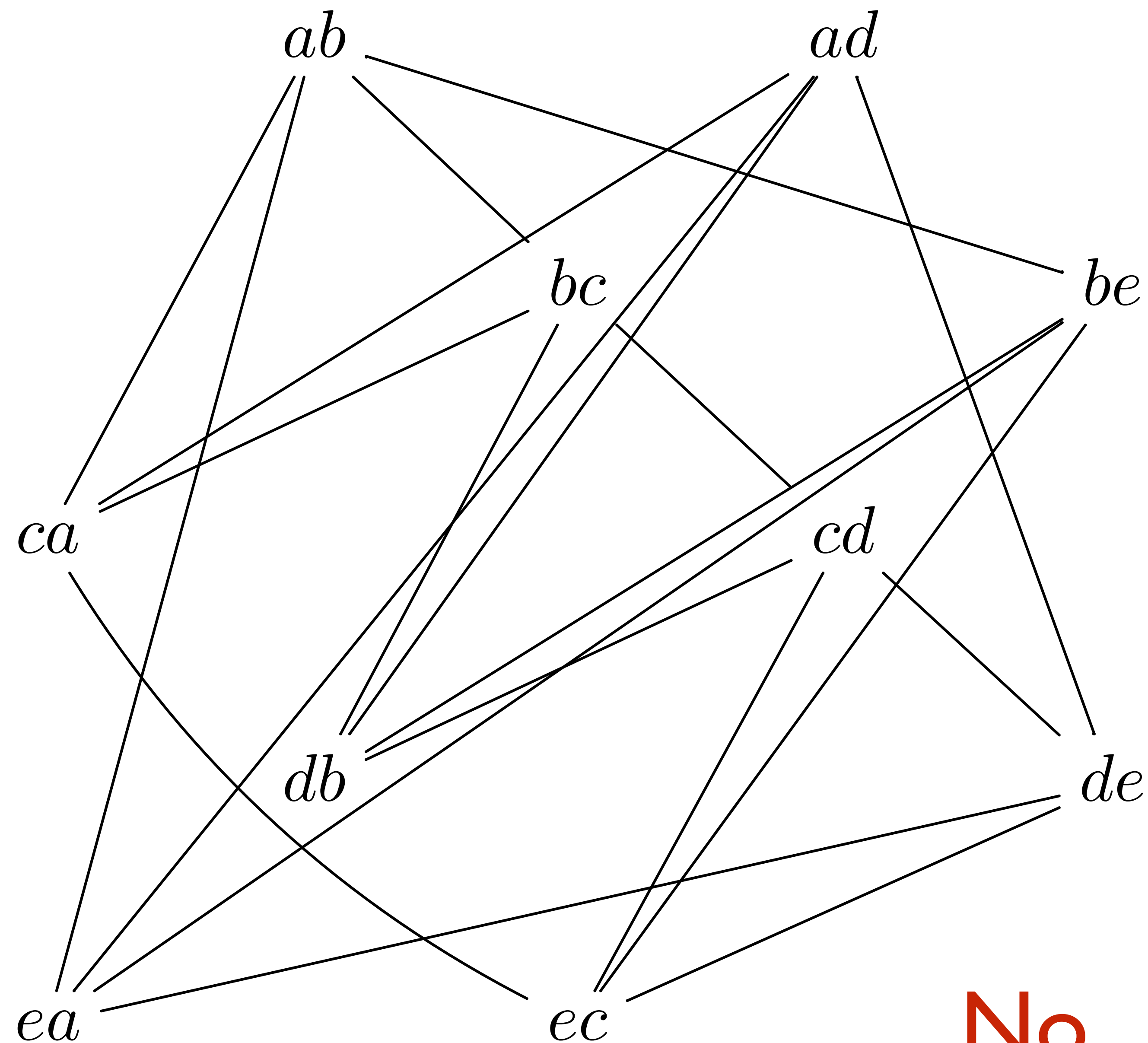
- nodes: ab $\qquad$ $a \neq b$
- edges: $ab—bc$ $\qquad$ $a \neq c$

Is it 3-colorable?

An example problem

- nodes: ab $\qquad a \neq b$
- edges: ab—bc $\qquad a \neq c$

Is it 3-colorable?

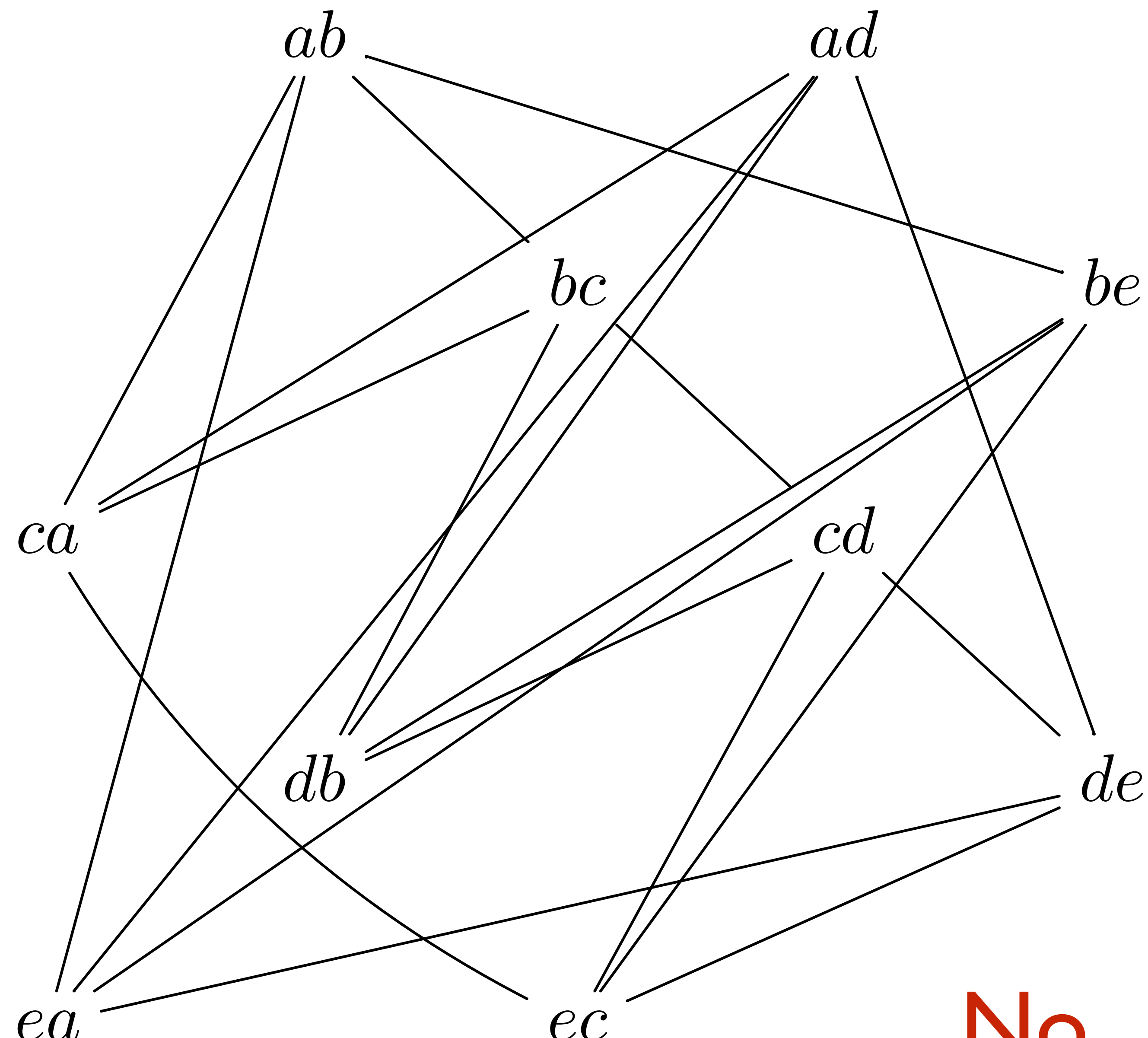


No.

An example problem

- nodes: ab $\quad$ $a \neq b$
- edges: ab—bc $\quad$ $a \neq c$

Is it 3-colorable?



No.

Is 3-colorability decidable?

What is it all about?

Nominal techniques:

mathematics of, and computation with:

local names
and name dependence

highly symmetrical
structures

“slightly infinite”
structures

structures
acessible via
limited interfaces

Computer Science 101

Theorem:

Every algorithm to sort n numbers must work in time $\Omega(n \log n)$.

Computer Science 101

Theorem:

Every algorithm to sort n numbers must work in time $\Omega(n \log n)$.

in the comparison model

Computer Science 101

Theorem:

Every algorithm to sort n numbers must work in time $\Omega(n \log n)$.

in the comparison model

Here, numbers are atoms accessible via relations:

$$= \qquad <$$

Computer Science 101

Theorem:

Every algorithm to sort n numbers must work in time $\Omega(n \log n)$.

in the comparison model

Here, numbers are atoms accessible via relations:

$$= \qquad <$$

This amounts to restricting the class of legal atom renamings.

Nominal Sets:
Basic Defnitions

$[\mathbb{A}]$

or: Sets with Atoms

Nominal Sets:
Basic Defnitions

Atoms

Let $\mathbb{A}$ be an infinite, countable set of atoms.

$$a, b, c, d, e, \ldots \in \mathbb{A}$$

Atoms

Let $\mathbb{A}$ be an infinite, countable set of atoms.

$$a, b, c, d, e, \ldots \in \mathbb{A}$$

$\mathrm{Aut}(\mathbb{A})$ - the group of all bijections of $\mathbb{A}$

Atoms

Let $\mathbb{A}$ be an infinite, countable set of atoms.

$$a, b, c, d, e, \ldots \in \mathbb{A}$$

$\mathrm{Aut}(\mathbb{A})$ - the group of all bijections of $\mathbb{A}$

$$(\pi \cdot \sigma) \cdot \rho = \pi \cdot (\sigma \cdot \rho)$$

$$\pi \cdot \pi^{-1} = \mathrm{id}$$

$$\pi \cdot \mathrm{id} = \pi = \mathrm{id} \cdot \pi$$

Atoms

Let $\mathbb{A}$ be an infinite, countable set of atoms.

$$a, b, c, d, e, \ldots \in \mathbb{A}$$

$\mathrm{Aut}(\mathbb{A})$ - the group of all bijections of $\mathbb{A}$

$$(\pi \cdot \sigma) \cdot \rho = \pi \cdot (\sigma \cdot \rho)$$
$$\pi \cdot \pi^{-1} = \mathrm{id}$$
$$\pi \cdot \mathrm{id} = \pi = \mathrm{id} \cdot \pi$$

the dot omitted frow now on

Atoms

Let $\mathbb{A}$ be an infinite, countable set of atoms.

$$a, b, c, d, e, \ldots \in \mathbb{A}$$

$\mathrm{Aut}(\mathbb{A})$ - the group of all bijections of $\mathbb{A}$

$$(\pi \cdot \sigma) \cdot \rho = \pi \cdot (\sigma \cdot \rho)$$
$$\pi \cdot \pi^{-1} = \mathrm{id}$$
$$\pi \cdot \mathrm{id} = \pi = \mathrm{id} \cdot \pi$$

the dot omitted frow now on

$(a\ b) \in \mathrm{Aut}(\mathbb{A})$ - the swap of a and b

Atoms

Let $\mathbb{A}$ be an infinite, countable set of atoms.

$$a, b, c, d, e, \ldots \in \mathbb{A}$$

$\mathrm{Aut}(\mathbb{A})$ - the group of all bijections of $\mathbb{A}$

$$(\pi \cdot \sigma) \cdot \rho = \pi \cdot (\sigma \cdot \rho)$$
$$\pi \cdot \pi^{-1} = \mathrm{id}$$
$$\pi \cdot \mathrm{id} = \pi = \mathrm{id} \cdot \pi$$

the dot omitted frow now on

$(a\ b) \in \mathrm{Aut}(\mathbb{A})$ - the swap of a and b

For example: $(a\ b)(b\ c)(c\ a) = (b\ c)$

$$(a\ b)^{-1} = (a\ b)$$

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.

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

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

Sets with atoms

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

A hierarchy of universes:

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

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

Renaming atoms

A canonical renaming action:

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

Renaming atoms

A canonical renaming action:

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

$$a \cdot \pi = \pi(a)$$

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

Renaming atoms

A canonical renaming action:

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

$$a \cdot \pi = \pi(a)$$

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

This is a group action of $\mathrm{Aut}(\mathbb{A})$:

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

$$x \cdot \mathrm{id} = x$$

Renaming atoms

A canonical renaming action:

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

$$a \cdot \pi = \pi(a)$$

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

This is a group action of $\mathrm{Aut}(\mathbb{A})$:

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

$$x \cdot \mathrm{id} = x$$

Fact: For every π , the function $\_ \cdot \pi$ is a bijection on $\mathcal{U}$.

Finite support

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

$$\forall a \in S. \pi(a) = a \quad \text{implies} \quad x \cdot \pi = x$$

Finite support

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

$$\forall a \in S.\pi(a) = a \quad \text{implies} \quad x \cdot \pi = x$$

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

Finite support

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

$$\forall a \in S. \pi(a) = a \quad \text{implies} \quad x \cdot \pi = x$$

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

A legal set with atoms, or nominal set:

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

Finite support

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

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

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

A legal set with atoms, or nominal set:

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

A set is equivariant if it has empty support.

Examples

$a \in \mathbb{A}$ is supported by $\{a\}$

Examples

$a \in \mathbb{A}$ is supported by $\{a\}$

$\mathbb{A}$ is equivariant

Examples

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

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

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

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

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

Basic Properties

Closure properties

Fact: if X and Y are legal sets then

$X \cup Y,\ \ X \cap Y,\ X + Y,\ X \setminus Y,\ X \times Y$ are legal.

Closure properties

Fact: if X and Y are legal sets then

$X \cup Y$, $X \cap Y$, $X + Y$, $X \setminus Y$, $X \times Y$ are legal.

Indeed: if

S supports X and T supports Y

then

$S \cup T$ supports $X \cup Y$, $X \cap Y$, ...

Closure properties

Fact: if X and Y are legal sets then

$X \cup Y$, $X \cap Y$, $X + Y$, $X \setminus Y$, $X \times Y$ are legal.

Indeed: if

S supports X and T supports Y

then

$S \cup T$ supports $X \cup Y$, $X \cap Y$, ...

(But: $S \cap T$ does not support $X \cap Y$!)

Closure properties

Fact: if X and Y are legal sets then

$X \cup Y$, $X \cap Y$, $X + Y$, $X \setminus Y$, $X \times Y$ are legal.

Indeed: if

S supports X and T supports Y

then

$S \cup T$ supports $X \cup Y$, $X \cap Y$, ...

(But: $S \cap T$ does not support $X \cap Y$!)

Fact: if X is legal and $Y \subseteq X$ is finitely supported then Y is legal.

Powersets

Fact: $\mathcal{P}\mathbb{A}$ is not legal (though it is equivariant).

Powersets

Fact: $\mathcal{P}\mathbb{A}$ is not legal (though it is equivariant).

Define:

$$\mathcal{P}_{\mathrm{fs}}X = \{Y \subseteq X \mid Y \text{ is finitely supported}\}$$

Powersets

Fact: $\mathcal{P}\mathbb{A}$ is not legal (though it is equivariant).

Define:

$$\mathcal{P}_{\mathrm{fs}}X = \{Y \subseteq X \mid Y \text{ is finitely supported } \}$$

Fact: if X is legal then $\mathcal{P}_{\mathrm{fs}}X$ is legal.

Powersets

Fact: $\mathcal{P}\mathbb{A}$ is not legal (though it is equivariant).

Define:

$$\mathcal{P}_{\mathrm{fs}}X = \{Y \subseteq X \mid Y \text{ is finitely supported}\}$$

Fact: if X is legal then $\mathcal{P}_{\mathrm{fs}}X$ is legal.

Key step: if S supports X
then $S \cdot \pi$ supports $X \cdot \pi$.

Powersets

Fact: $\mathcal{P}\mathbb{A}$ is not legal (though it is equivariant).

Define:

$$\mathcal{P}_{\mathrm{fs}}X = \{Y \subseteq X \mid Y \text{ is finitely supported}\}$$

Fact: if X is legal then $\mathcal{P}_{\mathrm{fs}}X$ is legal.

Key step: if S supports X
then $S \cdot \pi$ supports $X \cdot \pi$.

$$\sigma \in \mathrm{Aut}_{S\cdot\pi}(\mathbb{A}) \quad \Longrightarrow \quad \pi\sigma\pi^{-1} \in \mathrm{Aut}_S(\mathbb{A})$$

Powersets

Fact: $\mathcal{P}\mathbb{A}$ is not legal (though it is equivariant).

Define:

$$\mathcal{P}_{\mathrm{fs}}X = \{Y \subseteq X \mid Y \text{ is finitely supported}\}$$

Fact: if X is legal then $\mathcal{P}_{\mathrm{fs}}X$ is legal.

Key step: if S supports X then $S \cdot \pi$ supports $X \cdot \pi$.

$$\sigma \in \mathrm{Aut}_{S \cdot \pi}(\mathbb{A}) \quad \Longrightarrow \quad \pi\sigma\pi^{-1} \in \mathrm{Aut}_S(\mathbb{A})$$

$$X \cdot \pi = (X \cdot \pi\sigma\pi^{-1}) \cdot \pi = (X \cdot \pi) \cdot \sigma$$

Actions and supports

Fact: if S supports X and $\pi|_S = \sigma|_S$

then $X \cdot \pi = X \cdot \sigma$.

Actions and supports

Fact: if S supports X and $\pi|_S = \sigma|_S$

then $X \cdot \pi = X \cdot \sigma$.

Proof: if $\pi|_S = \sigma|_S$ then

$$\pi\sigma^{-1} \in \mathrm{Aut}_S(\mathbb{A})$$

so

$$X \cdot \sigma = (X \cdot \pi\sigma^{-1}) \cdot \sigma = X \cdot \pi$$

Actions and supports

Fact: if S supports X and $\pi|_S = \sigma|_S$
then $X \cdot \pi = X \cdot \sigma$.

Proof: if $\pi|_S = \sigma|_S$ then

$$\pi\sigma^{-1} \in \mathrm{Aut}_S(\mathbb{A})$$

so

$$X \cdot \sigma = (X \cdot \pi\sigma^{-1}) \cdot \sigma = X \cdot \pi$$

NB. these proofs are "easy".

Equivariant relations

A (binary) relation is a set of pairs.

Let's see what equivariance means for such sets:

$$R \cdot \pi = R \qquad \text{iff} \qquad (x, y) \in R \Longrightarrow (x, y) \cdot \pi \in R$$

Equivariant relations

A (binary) relation is a set of pairs.

Let's see what equivariance means for such sets:

$$R \cdot \pi = R \qquad \text{iff} \qquad (x, y) \in R \Longrightarrow (x, y) \cdot \pi \in R$$

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

$$xRy \ \text{ implies } \ (x \cdot \pi) R (y \cdot \pi) \ \text{ for all } \ \pi$$

Equivariant relations

A (binary) relation is a set of pairs.

Let's see what equivariance means for such sets:

$$R \cdot \pi = R \qquad \text{iff} \qquad (x, y) \in R \implies (x, y) \cdot \pi \in R$$

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

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

Similarly for S-supported relations, but for

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

Equivariant function

A function is a binary relation.

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

$$xRy \text{ implies } (x \cdot \pi)R(y \cdot \pi) \text{ for all } \pi$$

Equivariant function

A function is a binary relation.

$R \subseteq X \times Y$ 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

Equivariant function

A function is a binary relation.

$R \subseteq X \times Y$ 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

Similarly for S-supported functions, but for

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

Examples

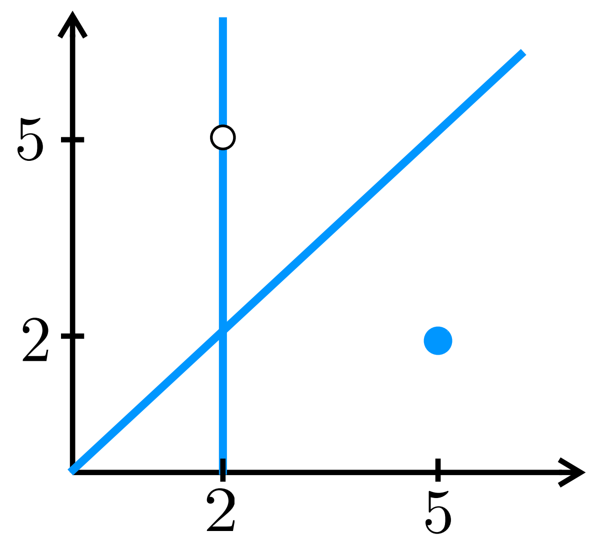
For fixed $2, 5 \in \mathbb{A}$:

Examples

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

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

R

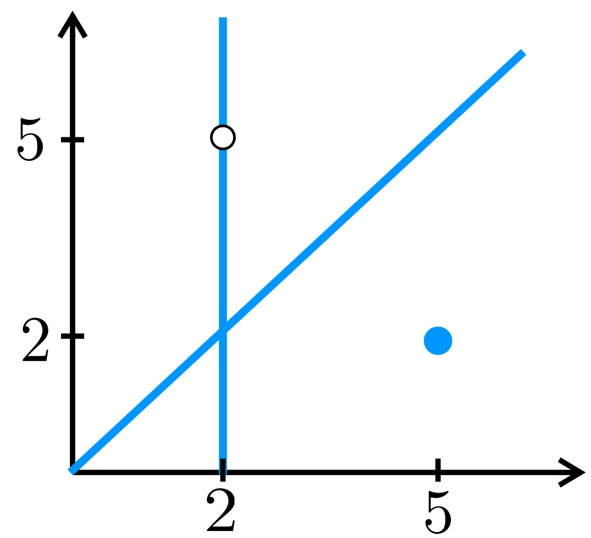


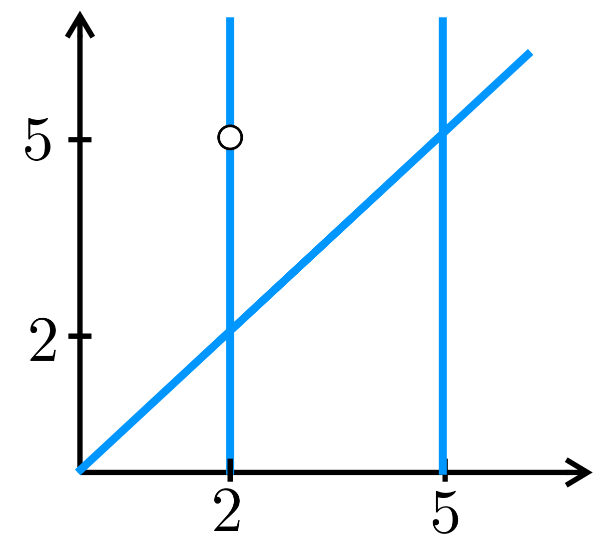
Examples

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

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

R





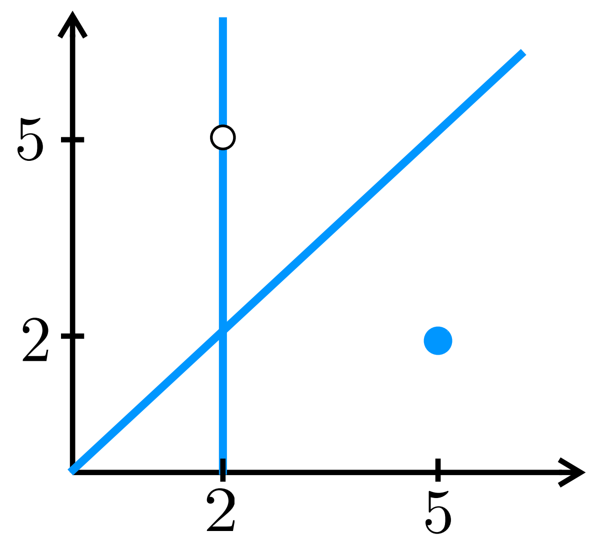
R^*

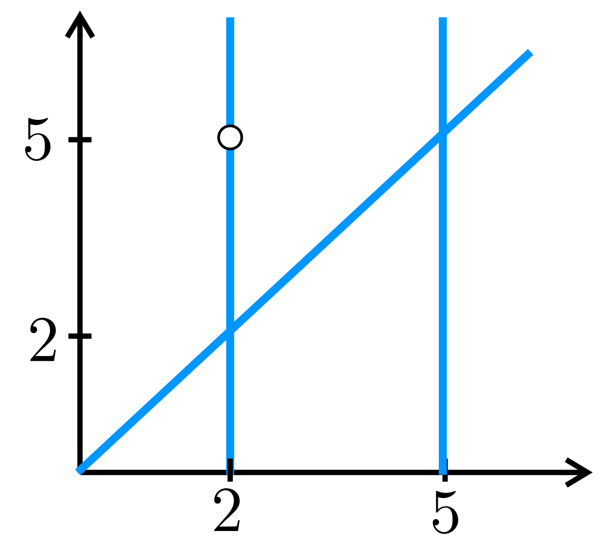
Examples

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 $\mathbb{A}$:

Examples ctd.

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

- empty
- equality
- total
- inequality

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.

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

Intuition

A relation/function/... is equivariant
iff
it only "checks" equality of atoms,
and does not mention specific atoms.

Intuition

A relation/function/... is equivariant
iff
it only "checks" equality of atoms,
and does not mention specific atoms.

A relation/function/... supported by S ,
may additionally mention
specific atoms from S .

Equivariant functions preserve supports

Fact: if S supports $x \in X$

and T supports $f : X \to Y$

then $S \cup T$ supports $f(x)$.

Equivariant functions preserve supports

Fact: if S supports $x \in X$
and T supports $f : X \to Y$
then $S \cup T$ supports $f(x)$.

Proof: $\mathrm{Aut}_{S \cup T}(\mathbb{A}) = \mathrm{Aut}_S(\mathbb{A}) \cap \mathrm{Aut}_T(\mathbb{A})$

so if $\pi \in \mathrm{Aut}_{S \cup T}(\mathbb{A})$

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

Equivariant functions preserve supports

Fact: if S supports $x \in X$

and T supports $f : X \to Y$

then $S \cup T$ supports $f(x)$.

Proof: $\mathrm{Aut}_{S\cup T}(\mathbb{A}) = \mathrm{Aut}_S(\mathbb{A}) \cap \mathrm{Aut}_T(\mathbb{A})$

so if $\pi \in \mathrm{Aut}_{S\cup T}(\mathbb{A})$

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

NB. another "easy" proof.

Least supports

Fact: for finite S and T,

if S supports X and T supports X

then $S \cap T$ supports X.

Least supports

Fact: for finite S and T,

if S supports X and T supports X

then $S \cap T$ supports X.

So: every legal X has the least support $\mathrm{supp}(X)$.

Least supports

Fact: for finite S and T,

if S supports X and T supports X

then $S \cap T$ supports X.

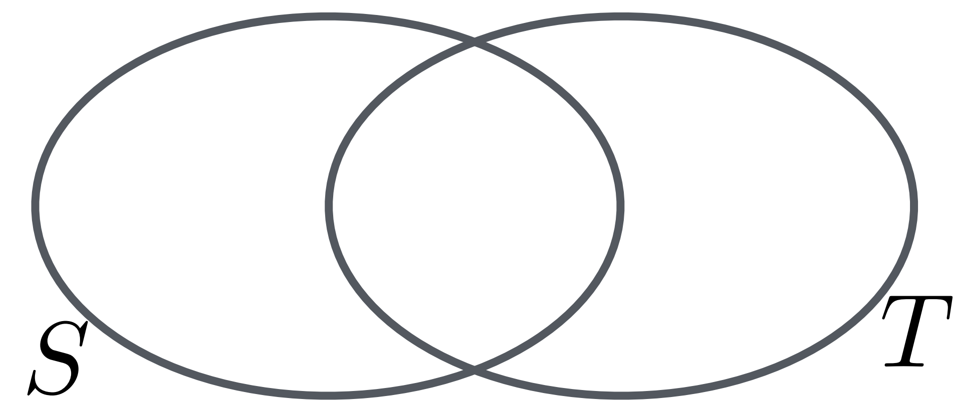
So: every legal X has the least support $\mathrm{supp}(X)$.

NB. This is harder to prove!

One way: induction on $|S \triangle T|$.

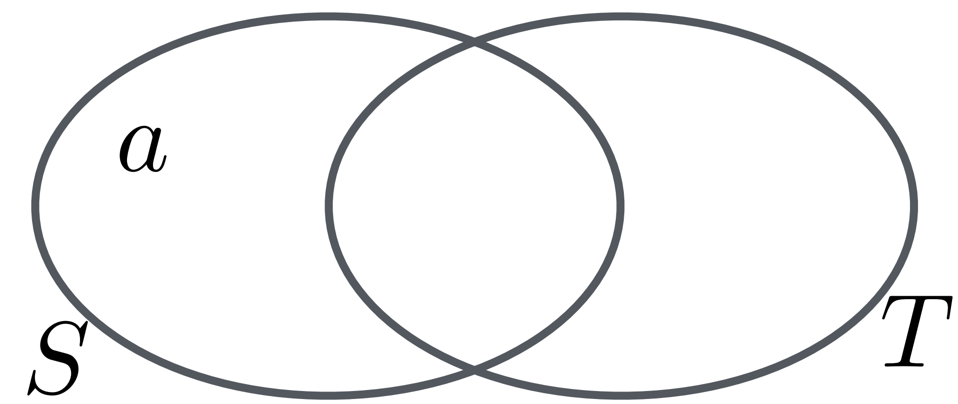
Proof

Assume S and T support X.



Proof

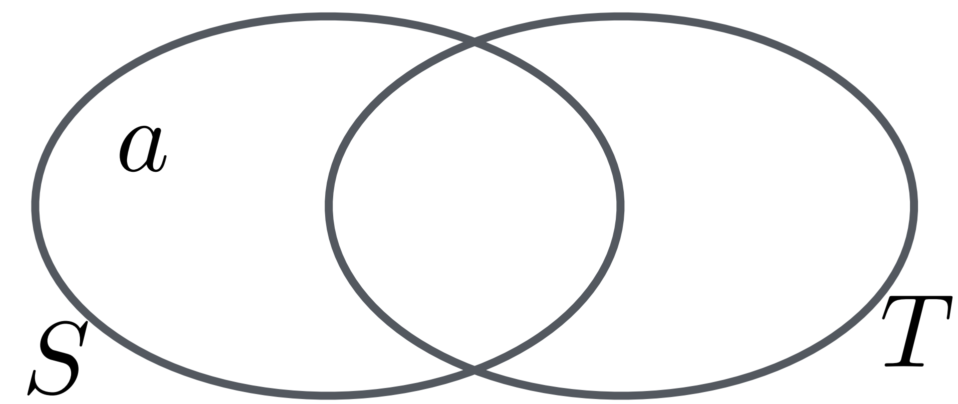
Assume S and T support X.



Proof

Assume S and T support X.

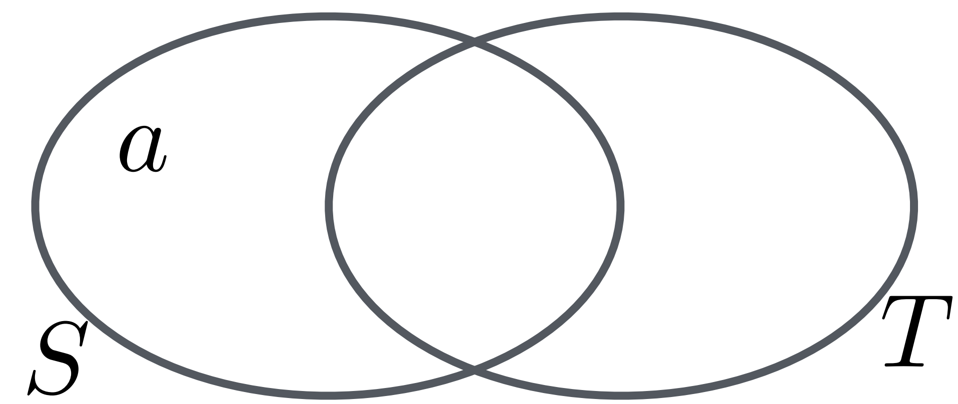
Goal: $S \setminus a$ supports X.



Proof

Assume S and T support X.

Goal: $S \setminus a$ supports X.



Take any $\pi \in \mathrm{Aut}_{S\setminus a}(\mathbb{A})$.

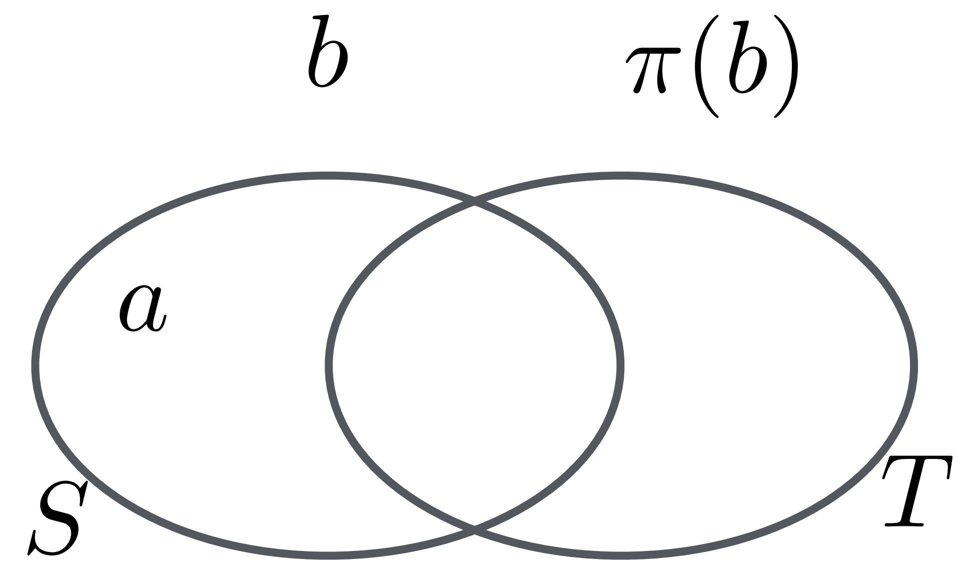
Proof

Assume S and T support X.

Goal: $S \setminus a$ supports X.

Take any $\pi \in \mathrm{Aut}_{S \setminus a}(\mathbb{A})$.

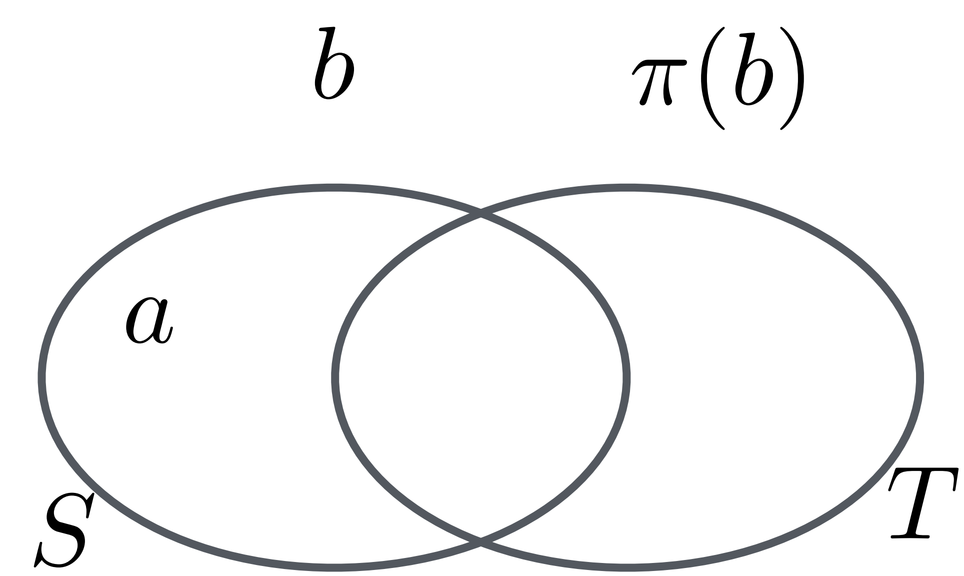
Pick a fresh b : $b, \pi(b) \notin S \cup T$.



Proof

Assume S and T support X.

Goal: $S \setminus a$ supports X.



Take any $\pi \in \mathrm{Aut}_{S \setminus a}(\mathbb{A})$.

Pick a fresh b : $b, \pi(b) \notin S \cup T$.

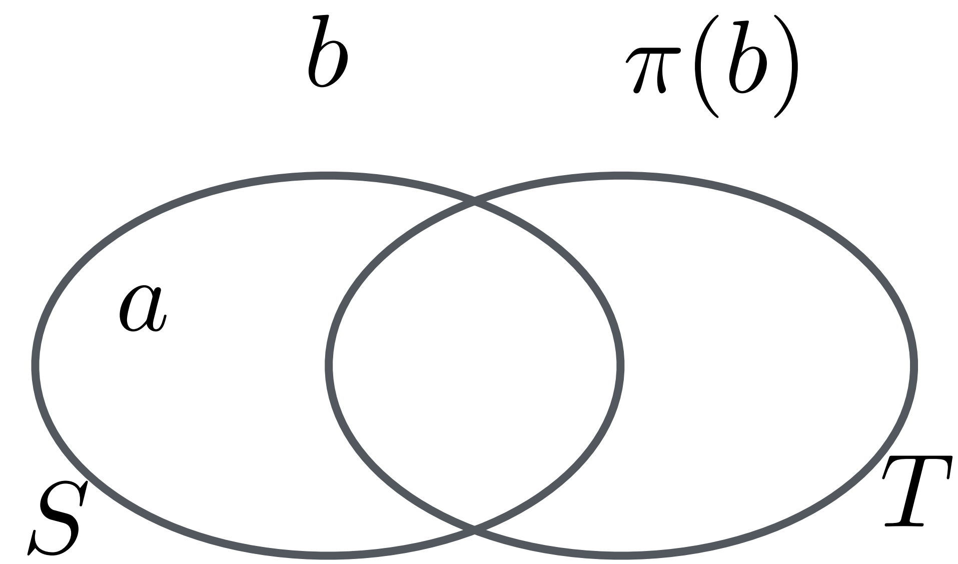
Put $\sigma = (a\ b)$, $\theta = (a\ \pi(b))$. Then:

$$\sigma, \theta = \mathrm{Aut}_T(\mathbb{A})$$

Proof

Assume S and T support X.

Goal: $S \setminus a$ supports X.



Take any $\pi \in \mathrm{Aut}_{S\setminus a}(\mathbb{A})$.

Pick a fresh b : $\quad b, \pi(b) \notin S \cup T$.

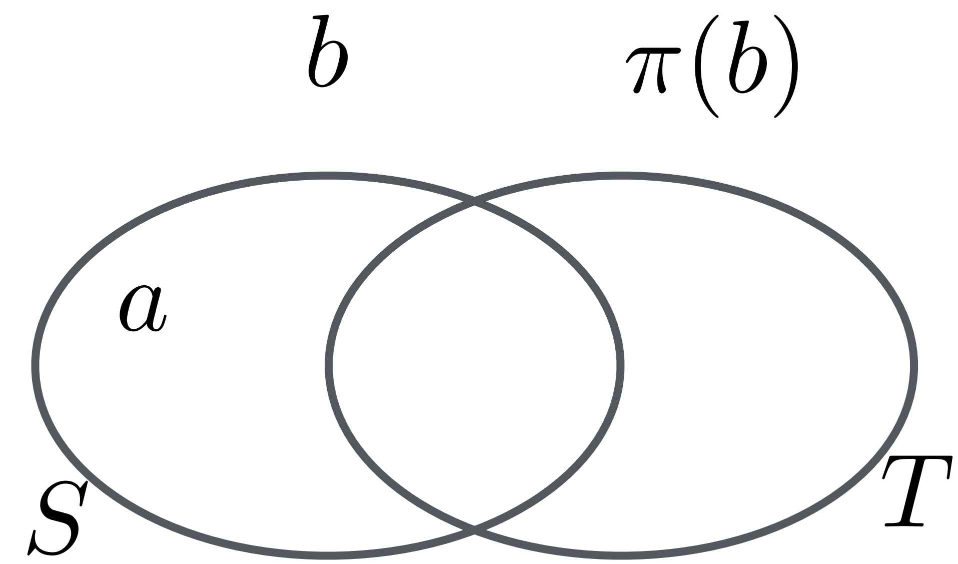
Put $\sigma = (a\ b)$, $\theta = (a\ \pi(b))$. Then:

$$\sigma, \theta = \mathrm{Aut}_T(\mathbb{A}) \qquad\qquad \sigma\pi\theta = \mathrm{Aut}_S(\mathbb{A})$$

Proof

Assume S and T support X.

Goal: $S \setminus a$ supports X.



Take any $\pi \in \mathrm{Aut}_{S\setminus a}(\mathbb{A})$.

Pick a fresh b : $b, \pi(b) \notin S \cup T$.

Put $\sigma = (a\ b)$, $\theta = (a\ \pi(b))$. Then:

$$\sigma, \theta = \mathrm{Aut}_T(\mathbb{A}) \qquad \sigma\pi\theta = \mathrm{Aut}_S(\mathbb{A})$$

so:

$$X \cdot \pi = ((X \cdot \sigma) \cdot \sigma\pi\theta) \cdot \theta = X$$

Name abstraction

For an (equivariant) set X,
define a relation $\approx$ on $\mathbb{A} \times X$ so:

$$(a, x) \approx (b, y) \iff x \cdot (a\ c) = y \cdot (b\ c)$$

for fresh c:

$$c \notin \{a, b\} \cup \mathrm{supp}(x, y)$$

Name abstraction

For an (equivariant) set X,
define a relation $\approx$ on $\mathbb{A} \times X$ so:

$$(a, x) \approx (b, y) \iff x \cdot (a\ c) = y \cdot (b\ c)$$

for fresh c:

$$c \notin \{a, b\} \cup \mathrm{supp}(x, y)$$

Fact: $\approx$ is an equivariant equivalence relation.

Name abstraction

For an (equivariant) set X,
define a relation $\approx$ on $\mathbb{A} \times X$ so:

$$(a, x) \approx (b, y) \iff x \cdot (a\ c) = y \cdot (b\ c)$$

for fresh c:

$$c \notin \{a, b\} \cup \mathrm{supp}(x, y)$$

Fact: $\approx$ is an equivariant equivalence relation.

Define: $$[\mathbb{A}]X = (\mathbb{A} \times X)/_{\approx}$$

Name abstraction

For an (equivariant) set X,
define a relation $\approx$ on $\mathbb{A} \times X$ so:

$$(a, x) \approx (b, y) \iff x \cdot (a\ c) = y \cdot (b\ c)$$

for fresh c:

$$c \notin \{a, b\} \cup \mathrm{supp}(x, y)$$

Fact: $\approx$ is an equivariant equivalence relation.

Define: $$[\mathbb{A}]X = (\mathbb{A} \times X)/_{\approx}$$

Fact: $[\mathbb{A}]X$ is an equivariant set.

$$\mathrm{supp}([a, x]_{\approx}) = \mathrm{supp}(x) \setminus \{a\}$$

Name abstraction

For an (equivariant) set X,
define a relation $\approx$ on $\mathbb{A} \times X$ so:

$$(a, x) \approx (b, y) \iff x \cdot (a\ c) = y \cdot (b\ c)$$

for fresh c:

$$c \notin \{a, b\} \cup \mathrm{supp}(x, y)$$

Fact: $\approx$ is an equivariant equivalence relation.

Define: $[\mathbb{A}]X = (\mathbb{A} \times X)/_{\approx}$

α-equivalence

Fact: $[\mathbb{A}]X$ is an equivariant set.

$$\mathrm{supp}([a, x]_{\approx}) = \mathrm{supp}(x) \setminus \{a\}$$

Exercises

1. If S supports $f : X \to Y$ and T supports $g : Y \to Z$ then $S \cup T$ supports $f ; g : X \to Z$.

Exercises

1. If S supports $f : X \to Y$ and T supports $g : Y \to Z$ then $S \cup T$ supports $f ; g : X \to Z$.
2. For an equivariant set X, the transitive closure function $(-)^* : \mathcal{P}_{\mathrm{fs}}(X \times X) \to \mathcal{P}_{\mathrm{fs}}(X \times X)$ is equivariant.

Exercises

1. If S supports $f : X \to Y$ and T supports $g : Y \to Z$ then $S \cup T$ supports $f ; g : X \to Z$.

2. For an equivariant set X, the transitive closure function $(-)^* : \mathcal{P}_{\mathrm{fs}}(X \times X) \to \mathcal{P}_{\mathrm{fs}}(X \times X)$ is equivariant.

3. For an equivariant set X, the least support function $\mathrm{supp} : X \to \mathcal{P}_{\mathrm{fin}}\mathbb{A}$ is equivariant.

Exercises

1. If S supports $f : X \to Y$ and T supports $g : Y \to Z$ then $S \cup T$ supports $f ; g : X \to Z$.

2. For an equivariant set X, the transitive closure function $(-)^{*} : \mathcal{P}_{\mathrm{fs}}(X \times X) \to \mathcal{P}_{\mathrm{fs}}(X \times X)$ is equivariant.

3. For an equivariant set X, the least support function $\mathrm{supp} : X \to \mathcal{P}_{\mathrm{fin}}\mathbb{A}$ is equivariant.

4. In a finite equivariant set, every element is equivariant.