Decidability border for Petri nets with data: wqo dichotomy conjecture

Sławomir Lasota
University of Warsaw

PN 2016 / ACSD 2016, Toruń, 2015.06.24

Outline

- (un)ordered data Petri nets
- standard decision problems
- Petri nets with homogeneous data
- undecidability
- decidability via wqo

unordered data Petri nets

- data domain $(\mathbb{N}, =)$

unordered data Petri nets

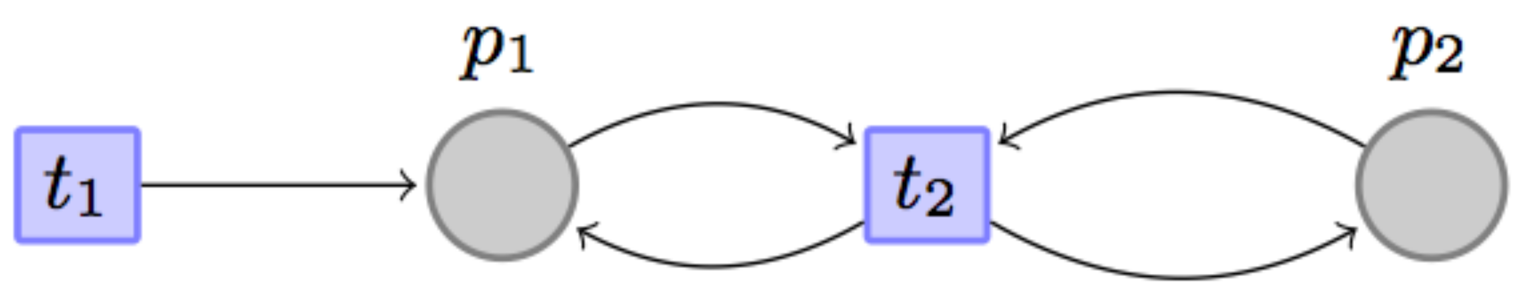
arbitrary countable infinite set

- data domain $(\mathbb{N}, =)$

unordered data Petri nets

arbitrary countable infinite set

- data domain $(\mathbb{N}, =)$

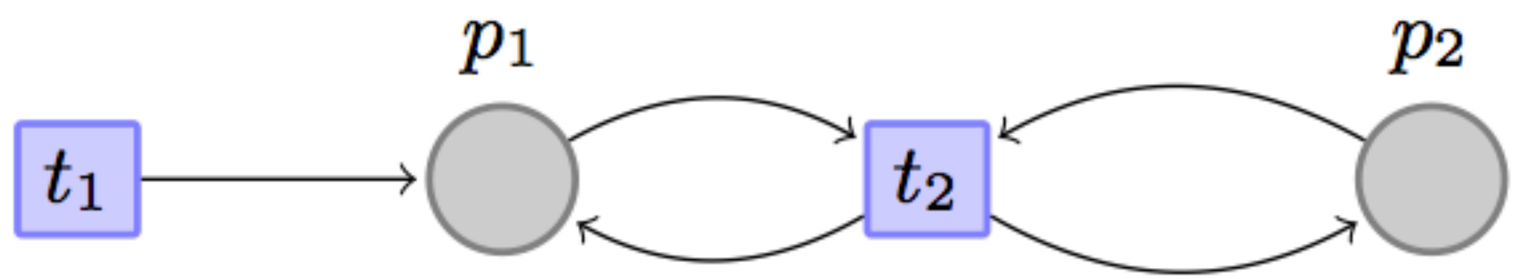


- finite sets of places P and transitions T, linked by arcs

unordered data Petri nets

arbitrary countable infinite set

- data domain $(\mathbb{N}, =)$

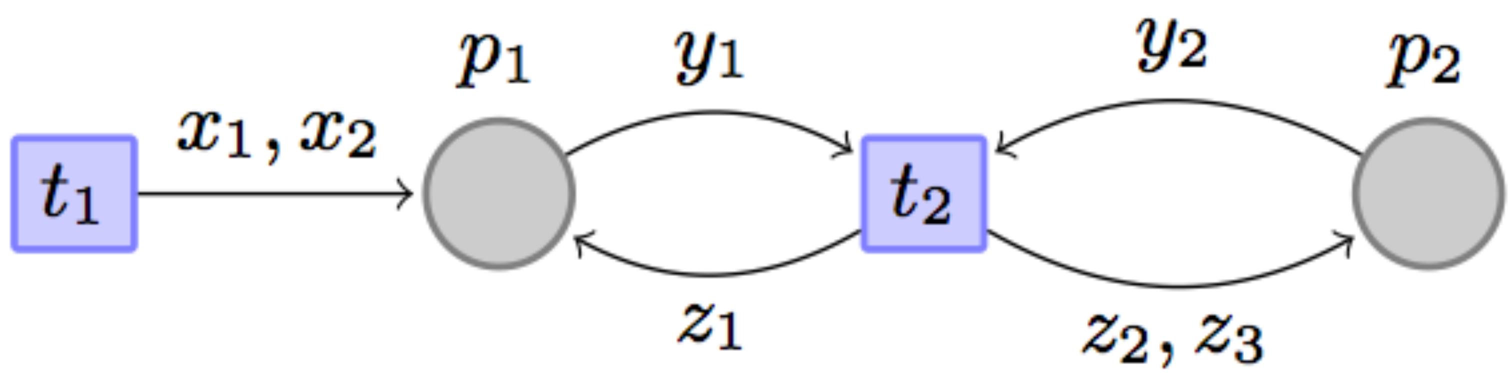


- finite sets of places P and transitions T, linked by arcs
- each arc labeled by a non-empty set of variables

unordered data Petri nets

arbitrary countable infinite set

- data domain $(\mathbb{N}, =)$

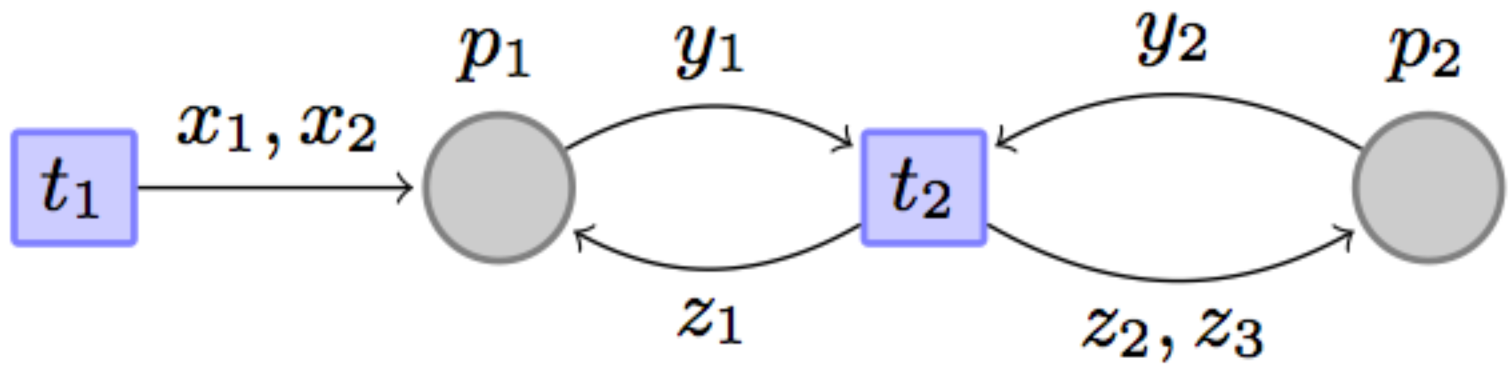


- finite sets of places P and transitions T, linked by arcs
- each arc labeled by a non-empty set of variables

unordered data Petri nets

arbitrary countable infinite set

- data domain $(\mathbb{N}, =)$

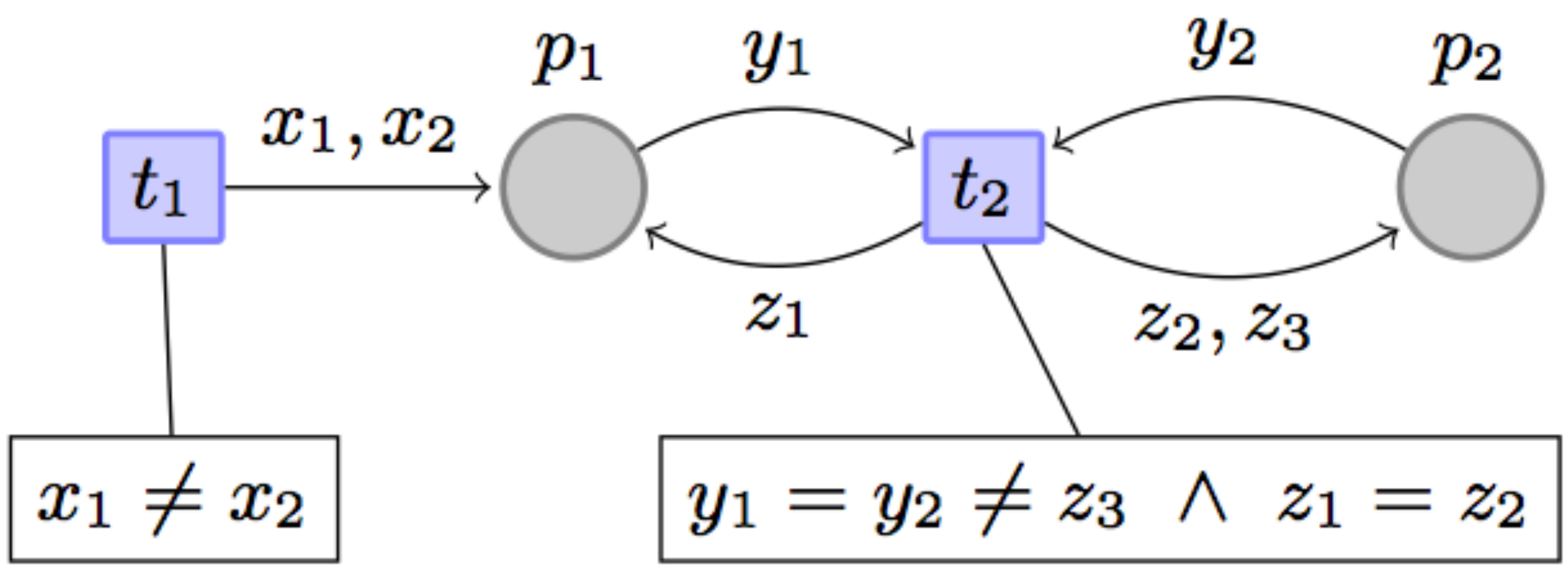


- finite sets of places P and transitions T, linked by arcs
- each arc labeled by a non-empty set of variables
- each transition labeled by a first-order formula with equality, whose free variables belong to the union of labels of adjacent arcs

unordered data Petri nets

arbitrary countable infinite set

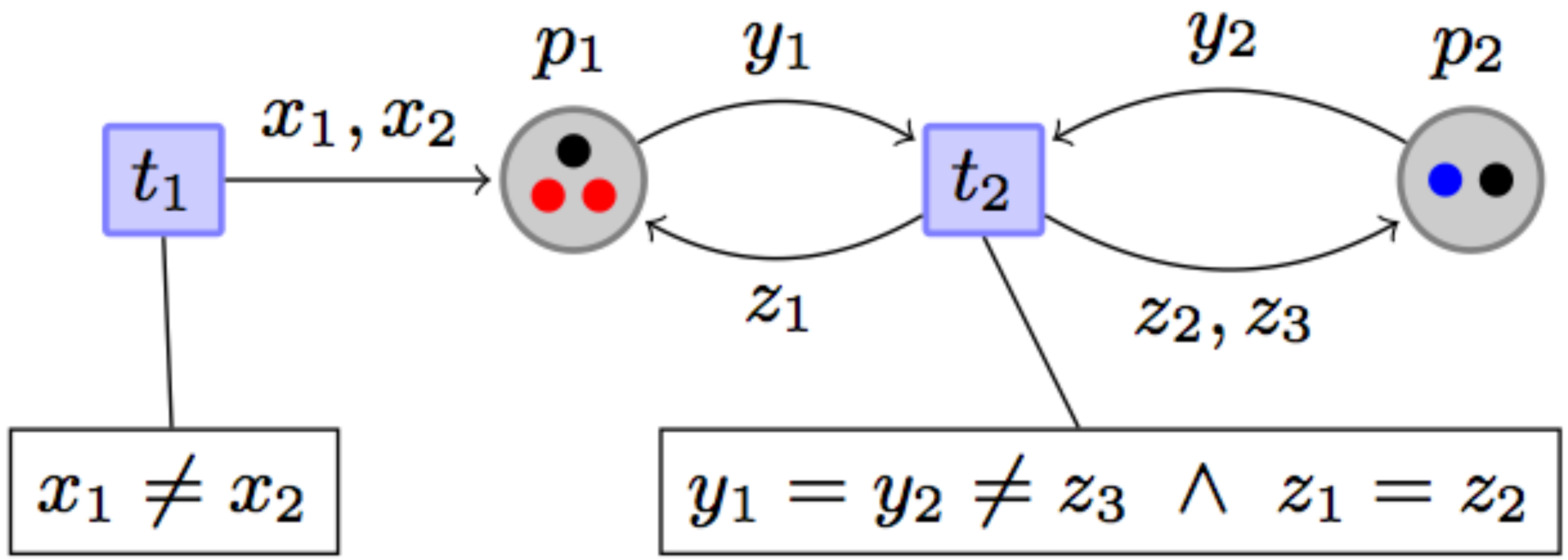
- data domain $(\mathbb{N}, =)$



- finite sets of places P and transitions T, linked by arcs
- each arc labeled by a non-empty set of variables
- each transition labeled by a first-order formula with equality, whose free variables belong to the union of labels of adjacent arcs

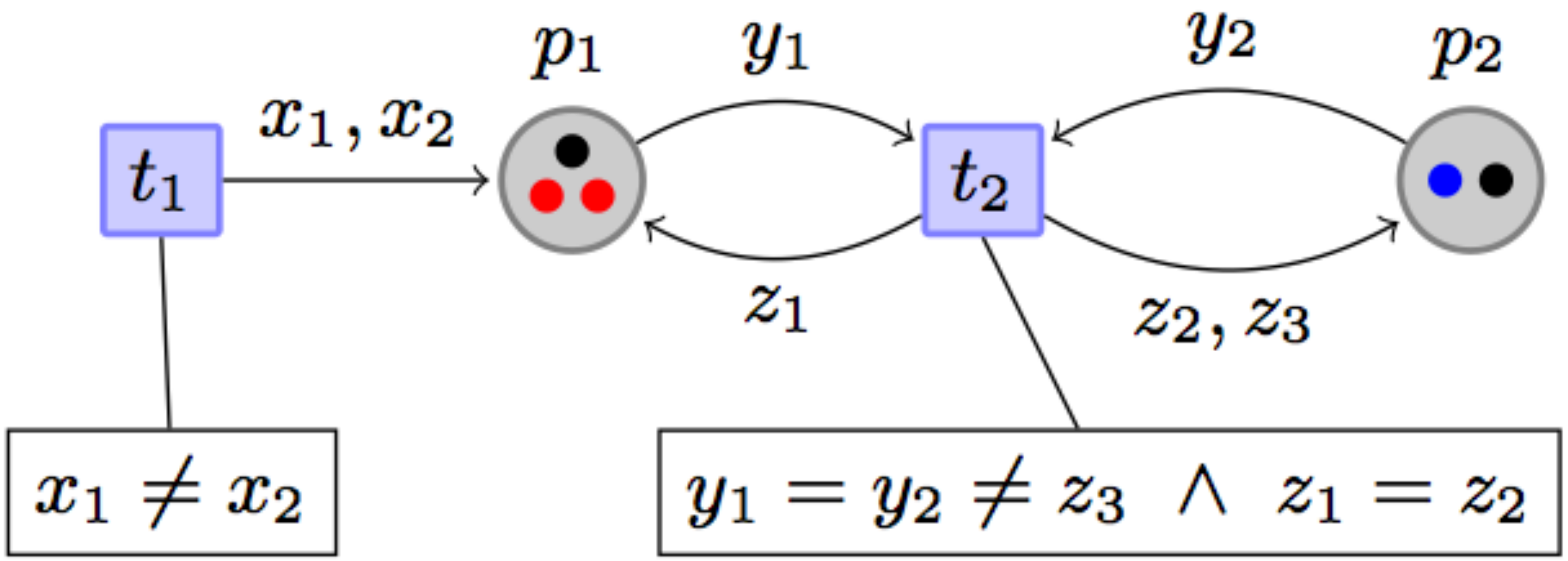
configurations (markings)

- data domain $(\mathbb{N}, =)$



configurations (markings)

- data domain $(\mathbb{N}, =)$

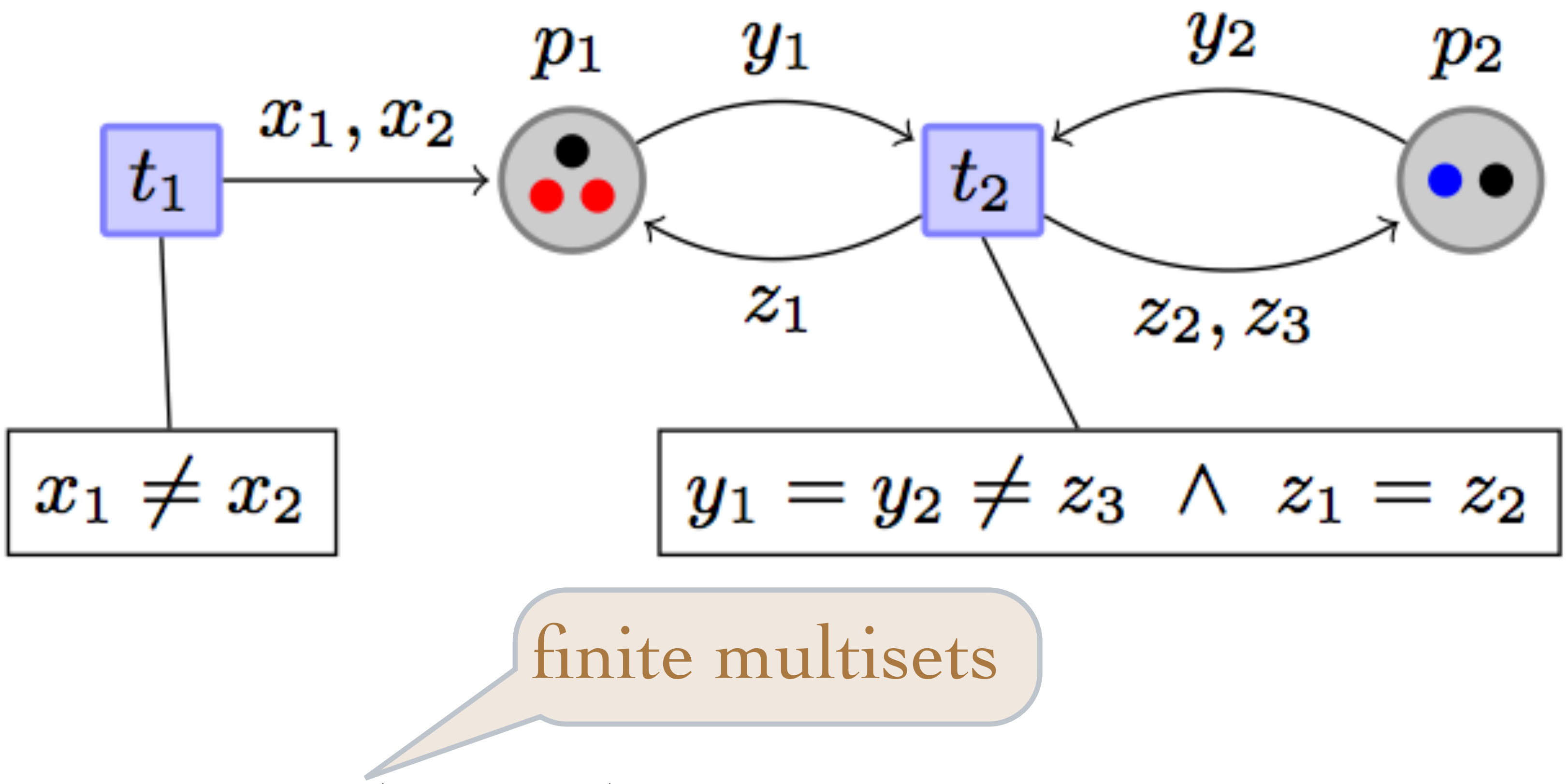


finite multisets

- configurations: $M(\mathrm{P} \times \mathbb{N})$

configurations (markings)

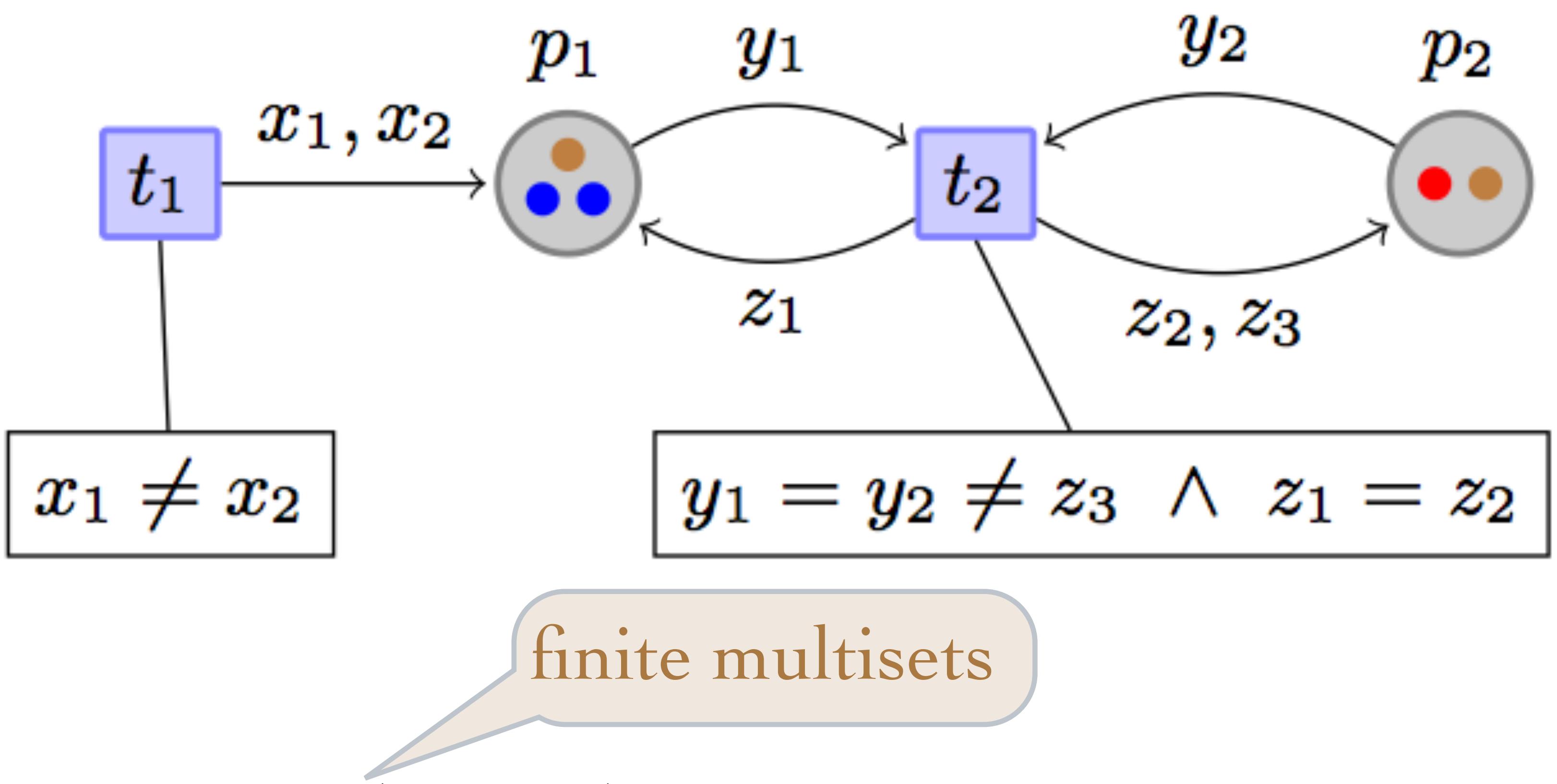
- data domain $(\mathbb{N}, =)$



- configurations: $M(\mathrm{P} \times \mathbb{N})$
- data automorphism = arbitrary permutation of $\mathbb{N}$

configurations (markings)

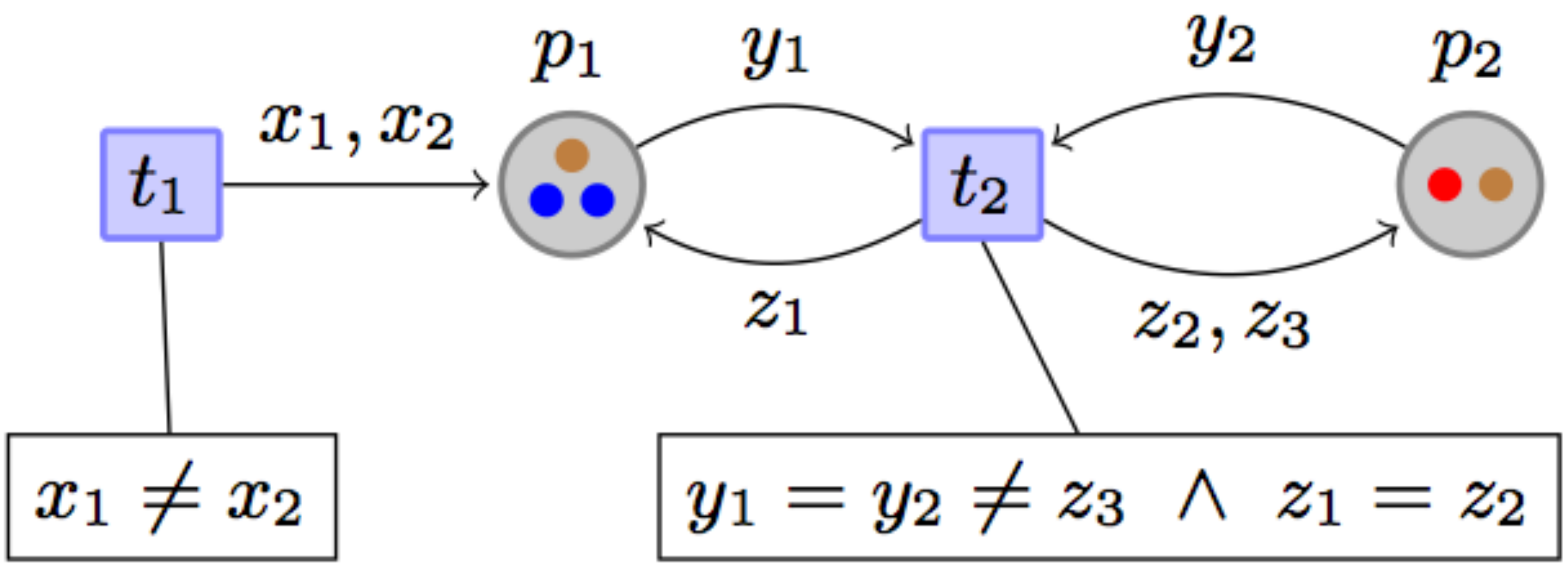
- data domain $(\mathbb{N}, =)$



- configurations: $M(\mathrm{P} \times \mathbb{N})$
- data automorphism = arbitrary permutation of $\mathbb{N}$

configurations (markings)

- data domain $(\mathbb{N}, =)$



finite multisets

- configurations: $M(\mathrm{P} \times \mathbb{N})$
- data automorphism = arbitrary permutation of $\mathbb{N}$
- configurations up to data automorphism: $M(M(\mathrm{P}))$

multiset rewriting

multiset rewriting

- a set $\mathcal{P}$

multiset rewriting

- a set $\mathcal{P}$
- a set of rewriting rules: $\mathcal{T} \quad \subseteq \quad M(\mathcal{P}) \times \mathcal{M}(\mathcal{P})$

multiset rewriting

- a set $\mathcal{P}$
- a set of rewriting rules: $\mathcal{T} \quad \subseteq \quad M(\mathcal{P}) \times \mathcal{M}(\mathcal{P})$

} multiset rewriting system

multiset rewriting

- a set $\mathcal{P}$
- a set of rewriting rules: $\mathcal{T} \quad \subseteq \quad M(\mathcal{P}) \times \mathcal{M}(\mathcal{P})$

} multiset rewriting system

- configurations: $\quad M(\mathcal{P})$

multiset rewriting

- a set $\mathcal{P}$
- a set of rewriting rules: $\mathcal{T} \quad \subseteq \quad M(\mathcal{P}) \times \mathcal{M}(\mathcal{P})$

} multiset rewriting system

- configurations: $M(\mathcal{P})$
- steps: $X + I \longrightarrow X + O, \qquad (I, O) \in \mathcal{T}$

multiset rewriting

- a set $\mathcal{P}$
- a set of rewriting rules: $\mathcal{T} \quad \subseteq \quad M(\mathcal{P}) \times \mathcal{M}(\mathcal{P})$

$\Big\}$ multiset rewriting system

- configurations: $M(\mathcal{P})$
- steps: $X + I \;\longrightarrow\; X + O, \qquad (I, O) \in \mathcal{T}$
- configurations reachable from an initial one

multiset rewriting

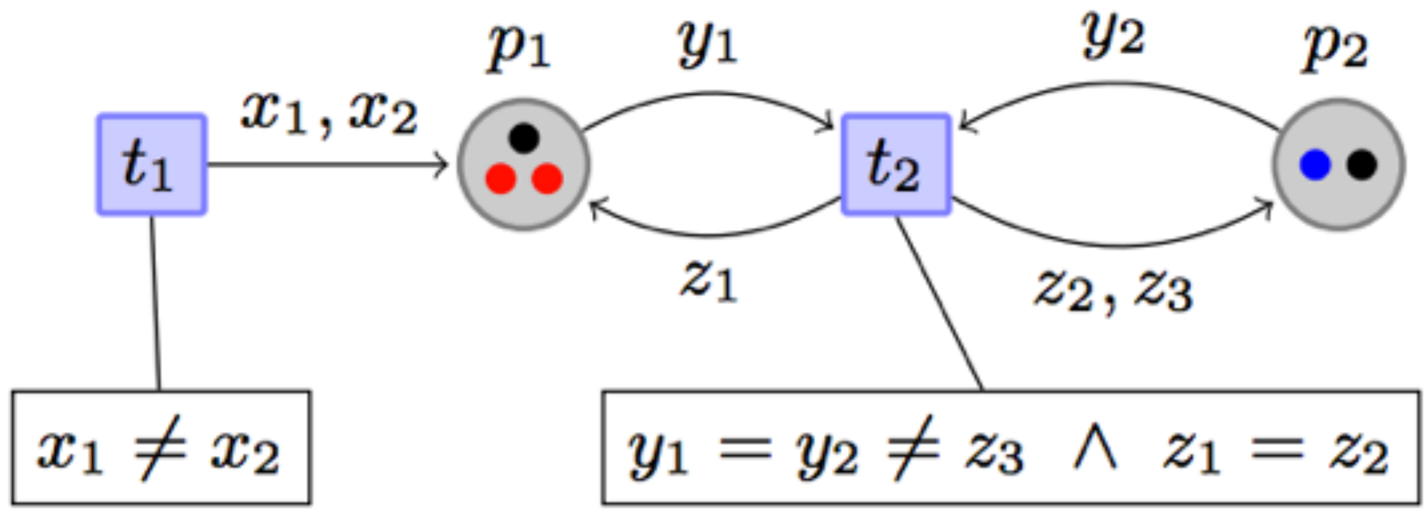
- a set $\mathcal{P}$
- a set of rewriting rules: $\mathcal{T} \quad \subseteq \quad M(\mathcal{P}) \times \mathcal{M}(\mathcal{P})$

} multiset rewriting system

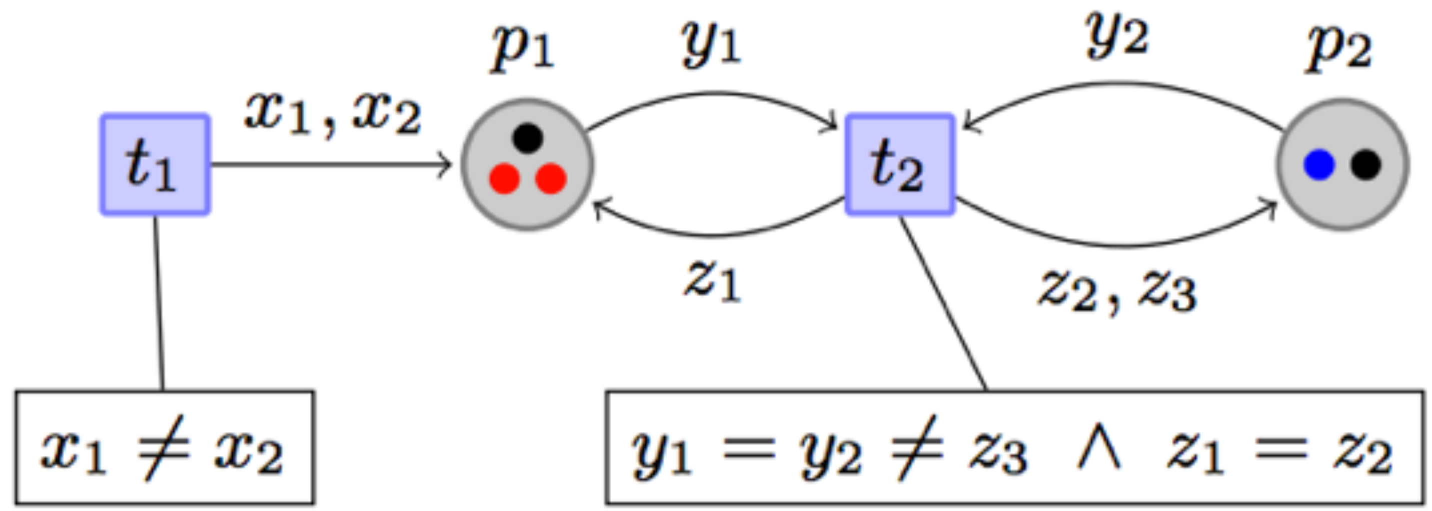
- configurations: $M(\mathcal{P})$
- steps: $X + I \longrightarrow X + O, \qquad (I, O) \in \mathcal{T}$
- configurations reachable from an initial one

For $\mathcal{P}$ and $\mathcal{T}$ finite, this is essentially classical Petri nets.

semantics via multiset rewriting

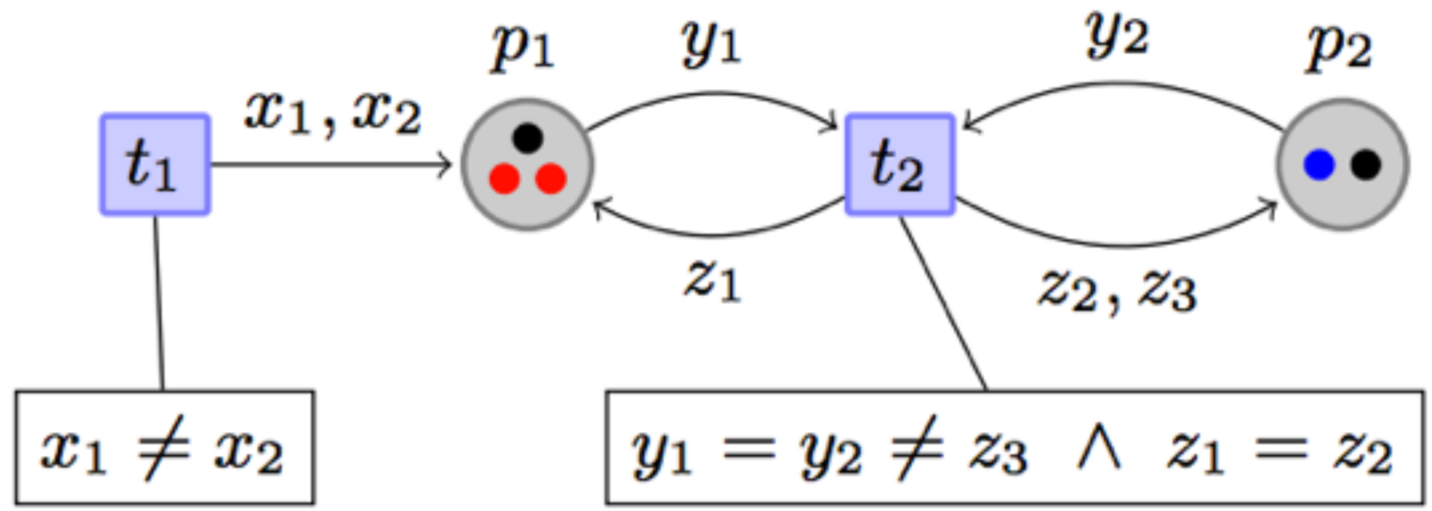


semantics via multiset rewriting



$$\mathcal{P} \ = \ \mathrm{P} \times \mathbb{N}$$

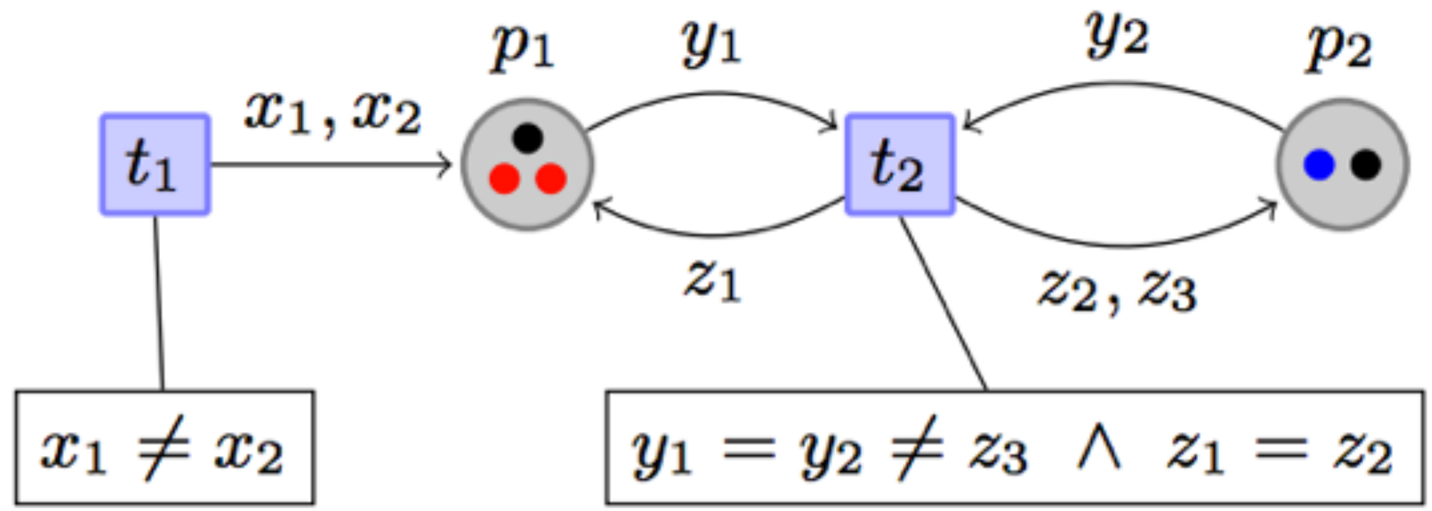
semantics via multiset rewriting



$$\mathcal{P} \ = \ \mathrm{P} \times \mathbb{N}$$

$$\mathcal{T} \ = \ \bigcup_{t \in \mathrm{T}} \mathcal{T}_t \qquad \mathcal{T}_t \subseteq M(\mathcal{P}) \times \mathcal{M}(\mathcal{P})$$

semantics via multiset rewriting

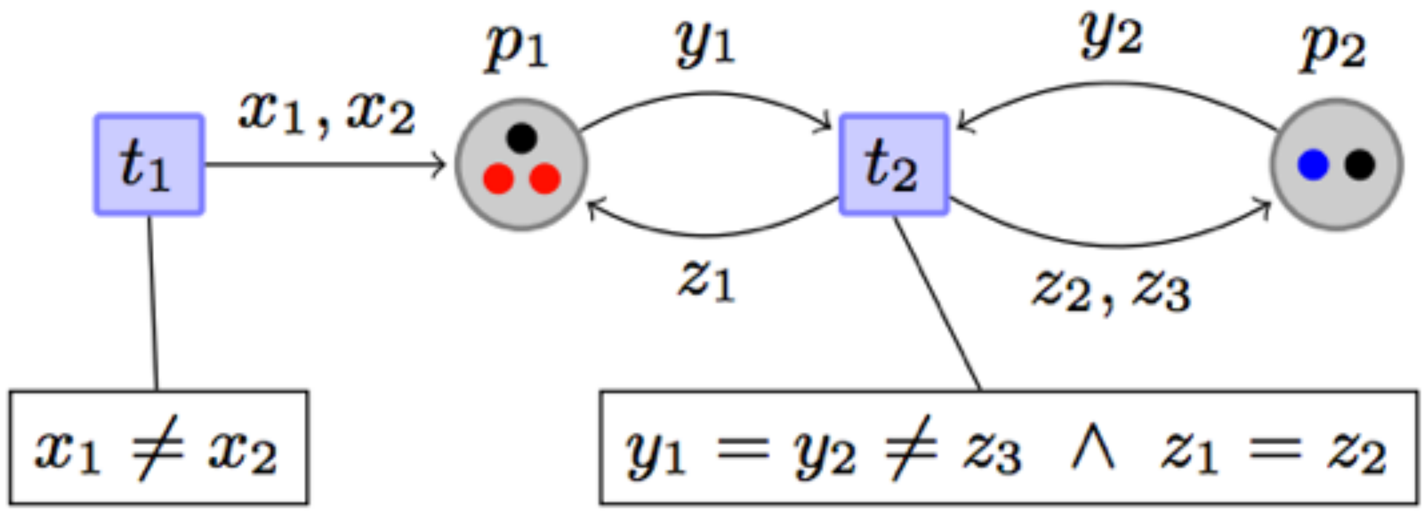


$$\mathcal{P} \;=\; \mathrm{P} \times \mathbb{N}$$

$$\mathcal{T} \;=\; \bigcup_{t \in \mathrm{T}} \mathcal{T}_t \qquad\qquad \mathcal{T}_t \subseteq M(\mathcal{P}) \times \mathcal{M}(\mathcal{P})$$

$$\mathcal{T}_{t_2} \;=\; \{\; (\{y_1, y_2\}, \{z_1, z_2, z_3\}) \;:\; y_1 = y_2 \neq z_3 \;\wedge\; z_1 = z_2 \}$$

semantics via multiset rewriting

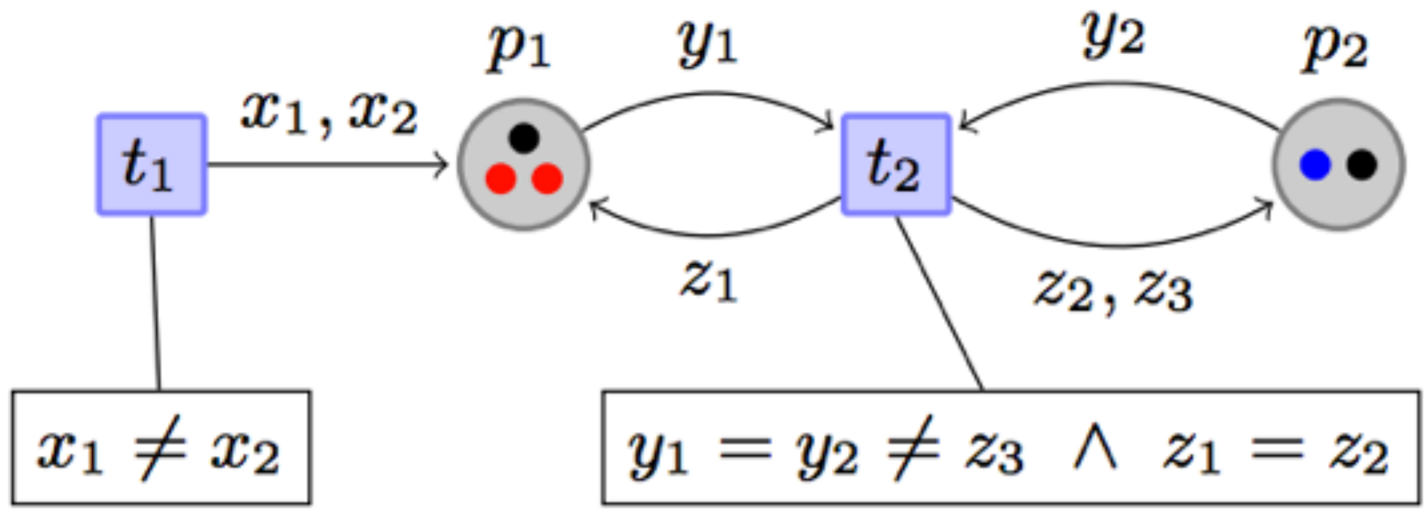


$$\mathcal{P} \ = \ \mathrm{P} \times \mathbb{N}$$

$$\mathcal{T} \ = \ \bigcup_{t \in \mathrm{T}} \mathcal{T}_t \qquad\qquad \mathcal{T}_t \subseteq M(\mathcal{P}) \times \mathcal{M}(\mathcal{P})$$

$$\mathcal{T}_t \ = \ \{(\{(p_1, y_1), (p_2, y_2)\}, \{(p_1, z_1), (p_2, z_2), (p_2, z_3)\}) \ : \ y_1 = y_2 \neq z_3 \ \wedge \ z_1 = z_2\}$$

semantics via multiset rewriting



Petri nets, where instead of finite sets of places and transitions, infinite ones but definable in first-order logic; this follows the lines of [Bojańczyk, Klin, L.'14]

$$\left.\begin{aligned} \mathcal{P} &= \mathrm{P} \times \mathbb{N} \\ \mathcal{T} &= \bigcup_{t \in \mathrm{T}} \mathcal{T}_t \end{aligned}\right\} \quad \mathcal{T}_t \subseteq M(\mathcal{P}) \times \mathcal{M}(\mathcal{P})$$

$$\mathcal{T}_t = \{(\{(p_1, y_1), (p_2, y_2)\}, \{(p_1, z_1), (p_2, z_2), (p_2, z_3)\}) \ : \ y_1 = y_2 \neq z_3 \ \wedge \ z_1 = z_2\}$$

ordered data Petri nets

- data domain $(\mathbb{Q}, <)$

ordered data Petri nets

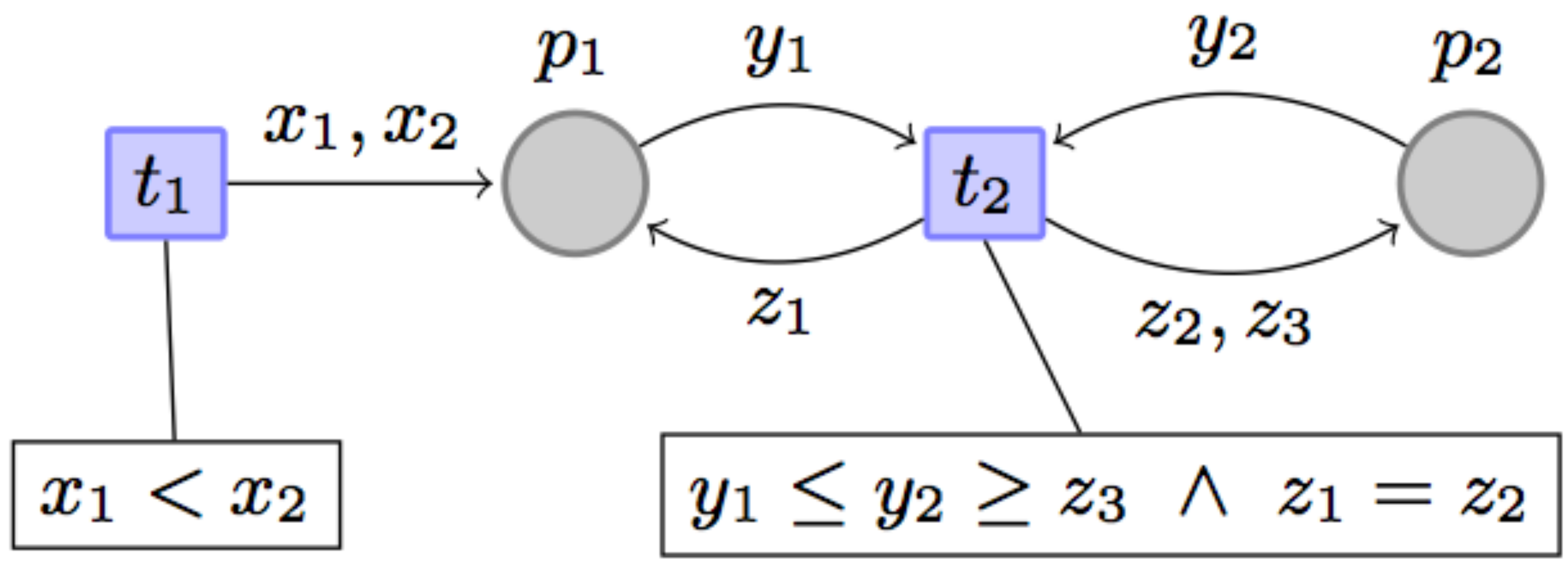
arbitrary countable infinite dense
total order without endpoints

- data domain $(\mathbb{Q}, <)$

ordered data Petri nets

arbitrary countable infinite dense total order without endpoints

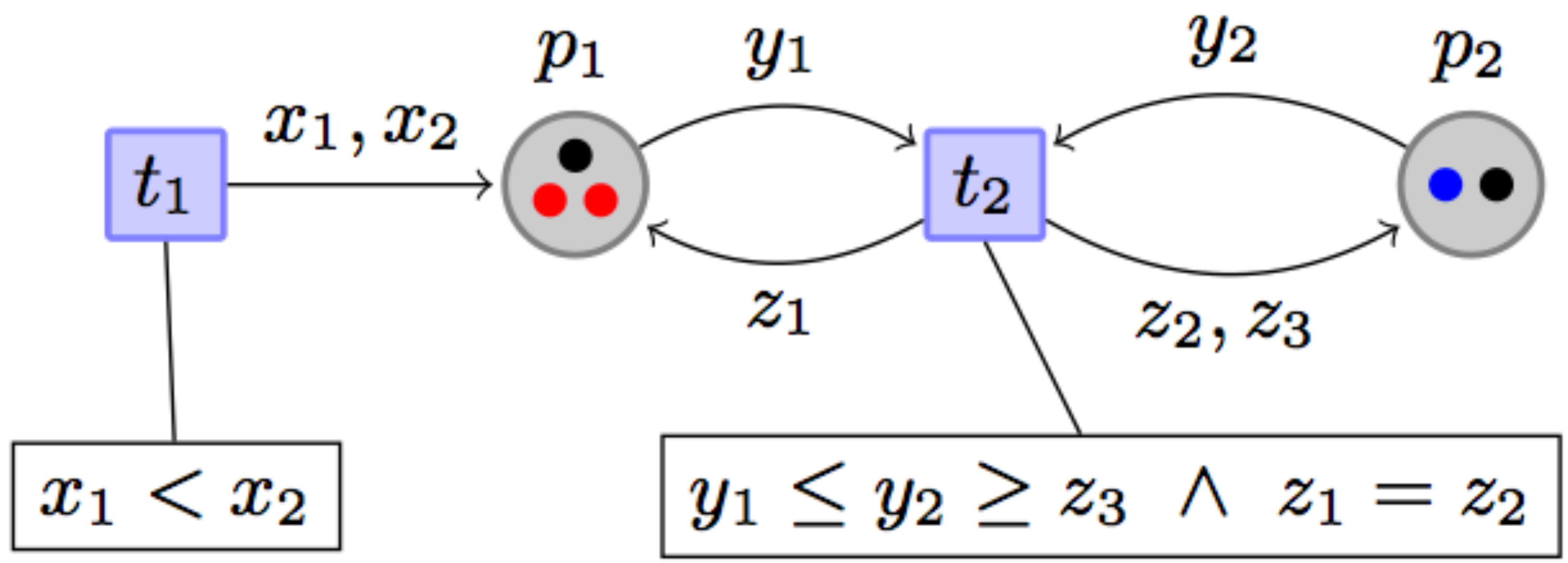
- data domain $(\mathbb{Q}, <)$



ordered data Petri nets

arbitrary countable infinite dense total order without endpoints

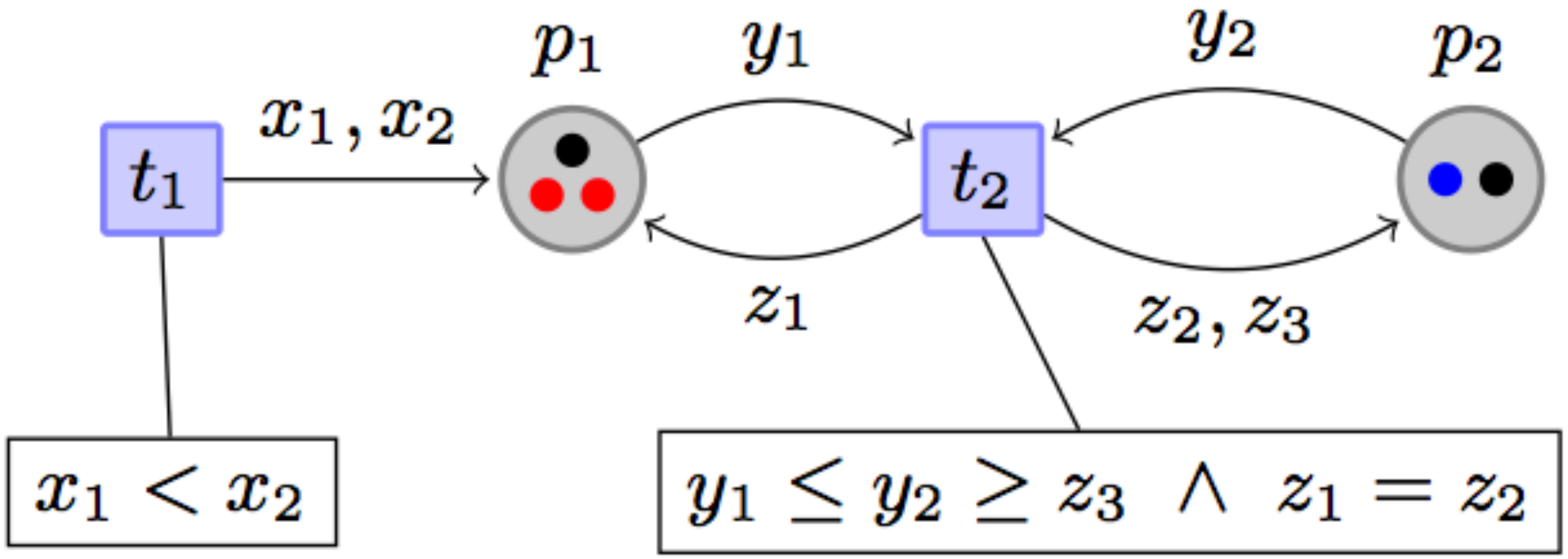
- data domain $(\mathbb{Q}, <)$



ordered data Petri nets

arbitrary countable infinite dense total order without endpoints

- data domain $(\mathbb{Q}, <)$

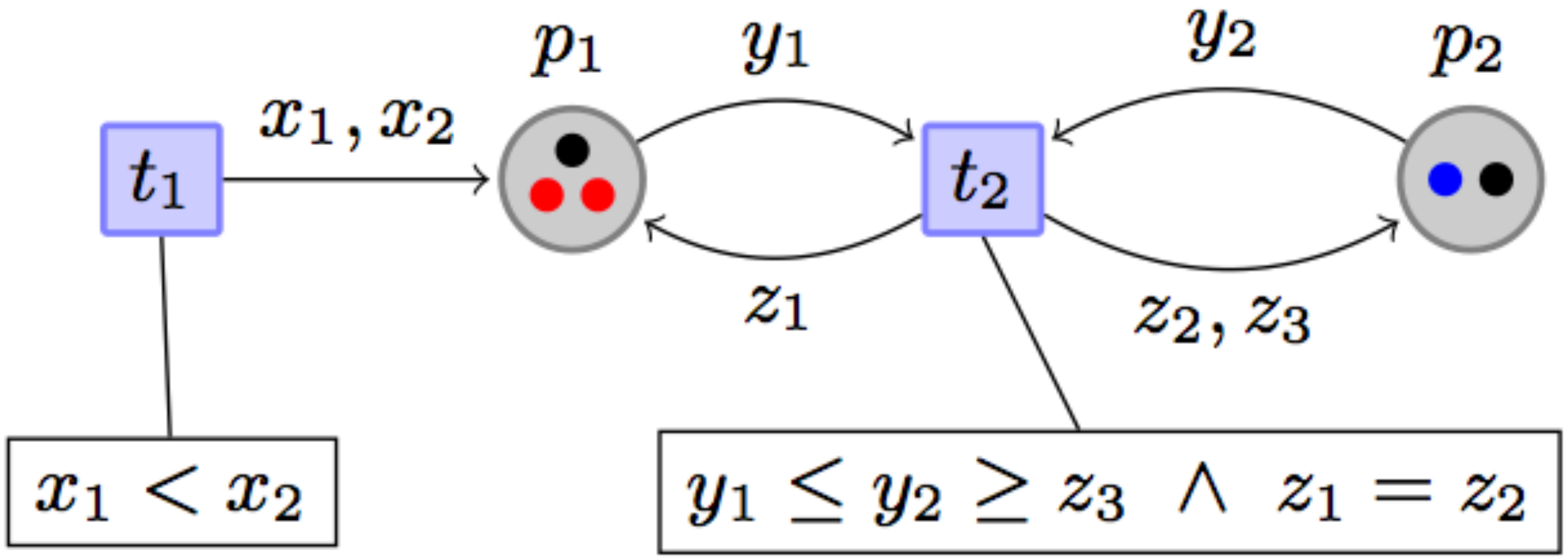


- configurations: $M(\mathrm{P} \times \mathbb{Q})$

ordered data Petri nets

- data domain $(\mathbb{Q}, <)$

arbitrary countable infinite dense total order without endpoints

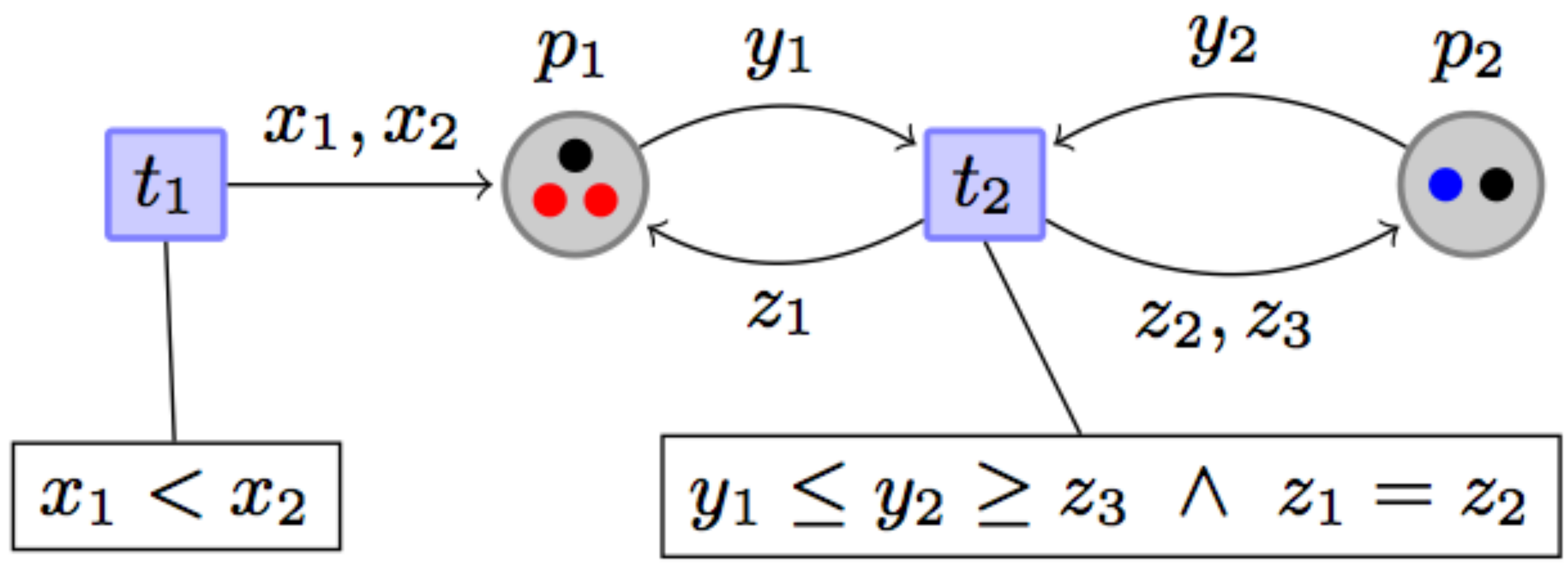


- configurations: $M(\mathrm{P} \times \mathbb{Q})$
- data automorphism = arbitrary monotonic permutation of $\mathbb{Q}$

ordered data Petri nets

- data domain $(\mathbb{Q}, <)$

arbitrary countable infinite dense total order without endpoints

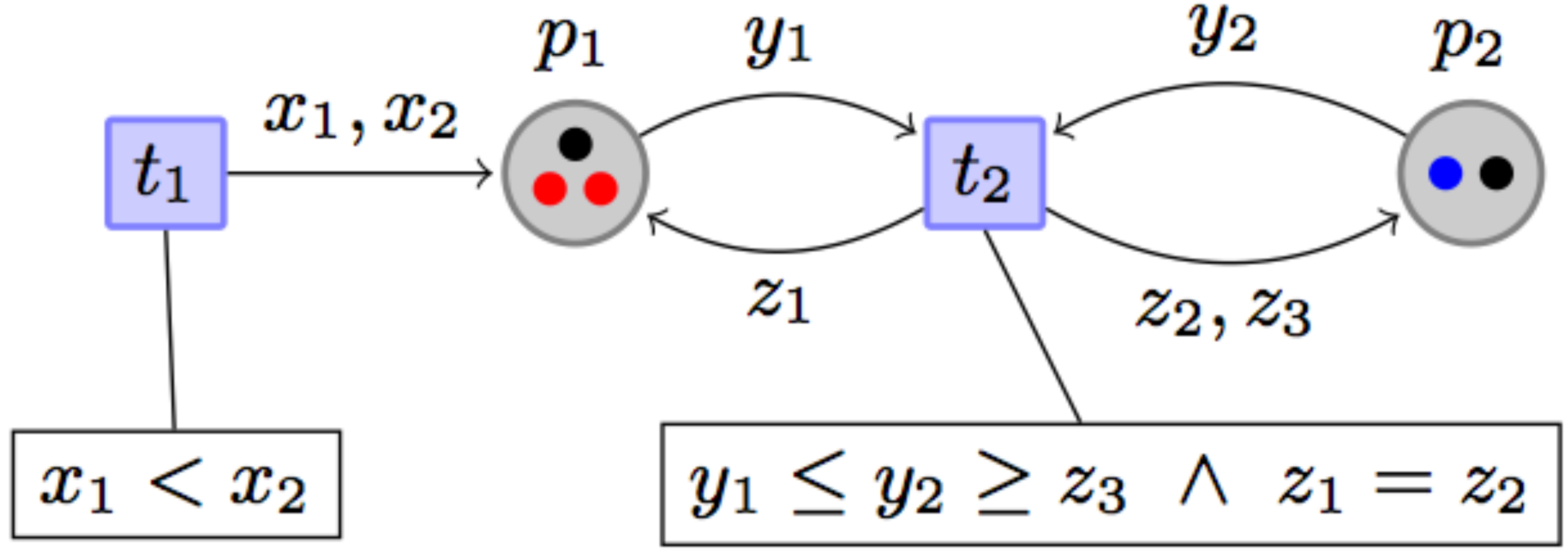


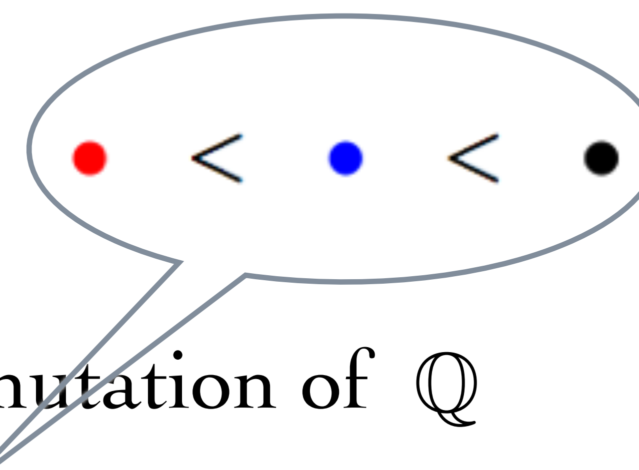
- configurations: $M(\mathrm{P} \times \mathbb{Q})$
- data automorphism = arbitrary monotonic permutation of $\mathbb{Q}$
- configurations up to data automorphism: $M(\mathrm{P})^*$

ordered data Petri nets

arbitrary countable infinite dense total order without endpoints

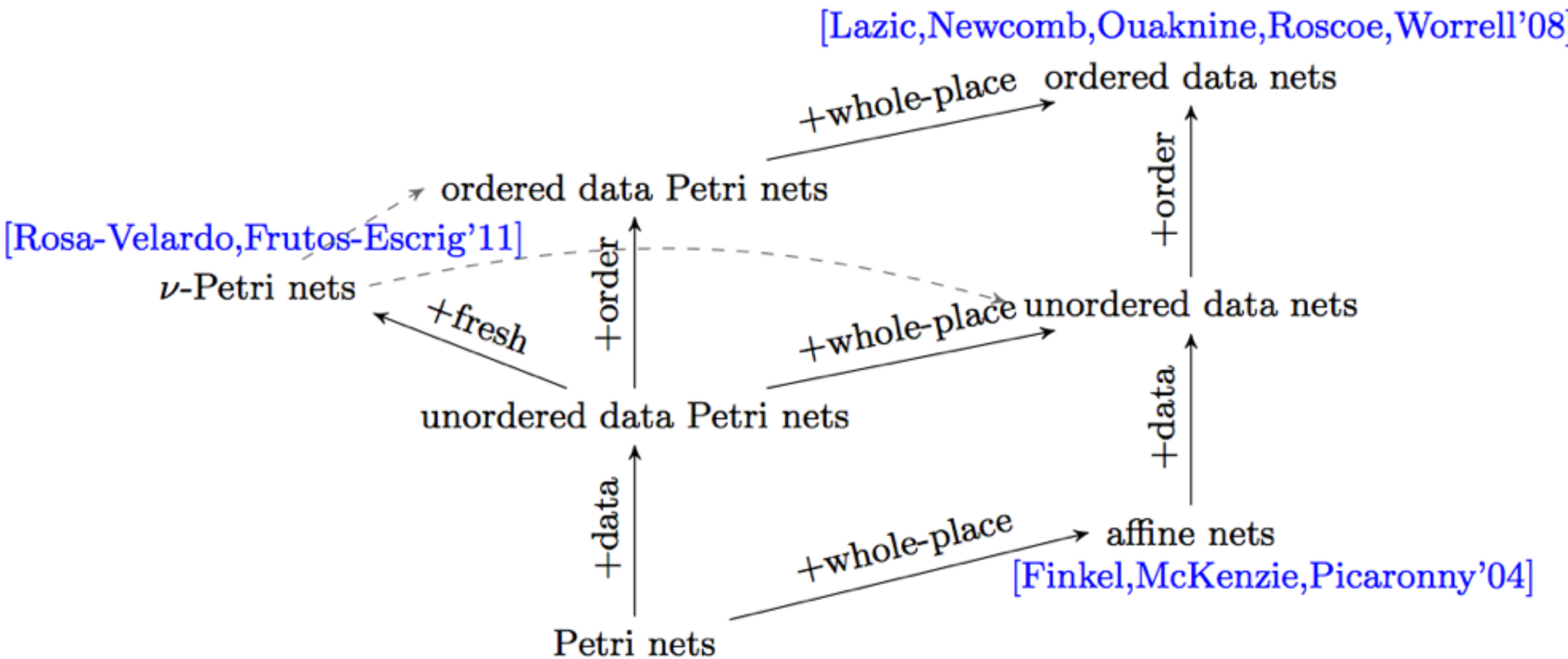
- data domain $(\mathbb{Q}, <)$



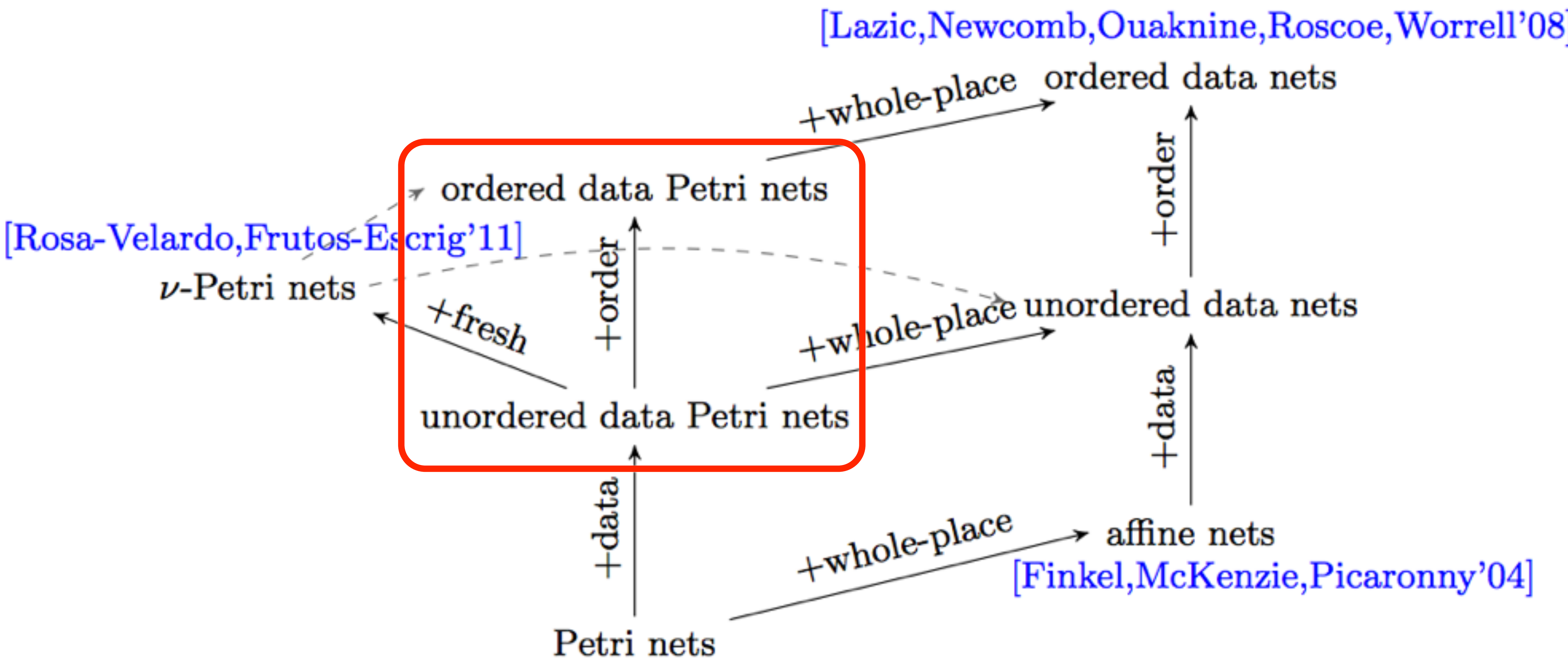


- configurations: $M(\mathrm{P} \times \mathbb{Q})$
- data automorphism = arbitrary monotonic permutation of $\mathbb{Q}$
- configurations up to data automorphism: $M(\mathrm{P})^*$

taxonomy of extensions



taxonomy of extensions



Outline

- (un)ordered data Petri nets
- **standard decision problems**
- Petri nets with homogeneous data
- undecidability
- decidability via wqo

standard decision problems

input: an (un)ordered Petri net
and an initial configuration.

standard decision problems

input: an (un)ordered Petri net
and an initial configuration.

- termination: are all runs finite?

standard decision problems

input: an (un)ordered Petri net and an initial configuration.

- termination: are all runs finite?
- place non-emptiness: does some reachable configuration put a token on a given place?

standard decision problems

input: an (un)ordered Petri net and an initial configuration.

- termination: are all runs finite?
- place non-emptiness: does some reachable configuration put a token on a given place?

} defined like classically

standard decision problems

input: an (un)ordered Petri net and an initial configuration.

- termination: are all runs finite?
- place non-emptiness: does some reachable configuration put a token on a given place?

} defined like classically

- boundedness: is the set of reachable configurations finite, up to data automorphism?

standard decision problems

input: an (un)ordered Petri net and an initial configuration.

- termination: are all runs finite?
- place non-emptiness: does some reachable configuration put a token on a given place?

} defined like classically

- boundedness: is the set of reachable configurations finite, up to data automorphism?
- coverability: does some reachable configuration cover a given configuration, up to data automorphism?

standard decision problems

input: an (un)ordered Petri net and an initial configuration.

- termination: are all runs finite?
- place non-emptiness: does some reachable configuration put a token on a given place?

} defined like classically

- boundedness: is the set of reachable configurations finite, up to data automorphism?
- coverability: does some reachable configuration cover a given configuration, up to data automorphism?

} defined up to automorphism

standard decision problems

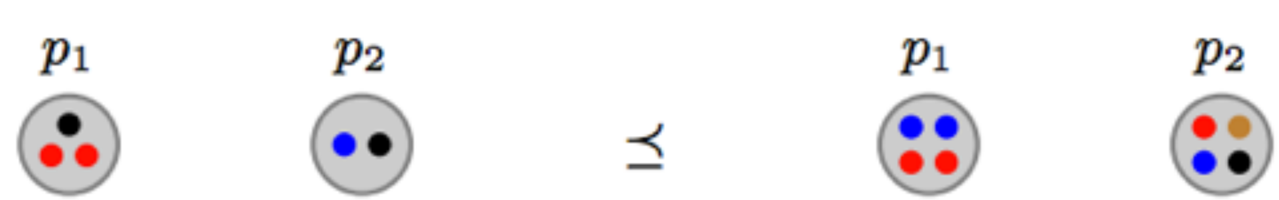
input: an (un)ordered Petri net and an initial configuration.

- termination: are all runs finite?
- place non-emptiness: does some reachable configuration put a token on a given place?

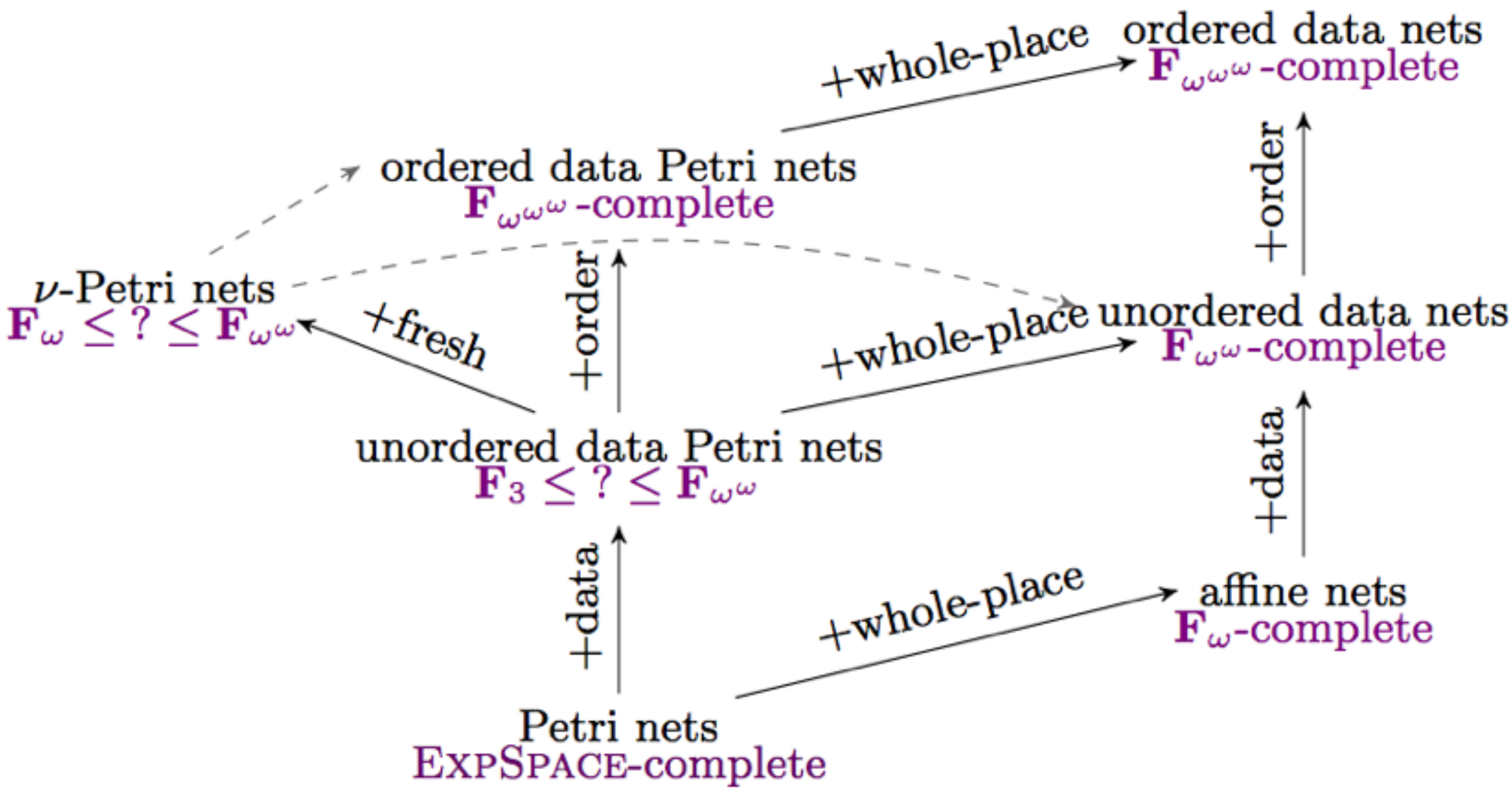
defined like classically

- boundedness: is the set of reachable configurations finite, up to data automorphism?
- coverability: does some reachable configuration cover a given configuration, up to data automorphism?

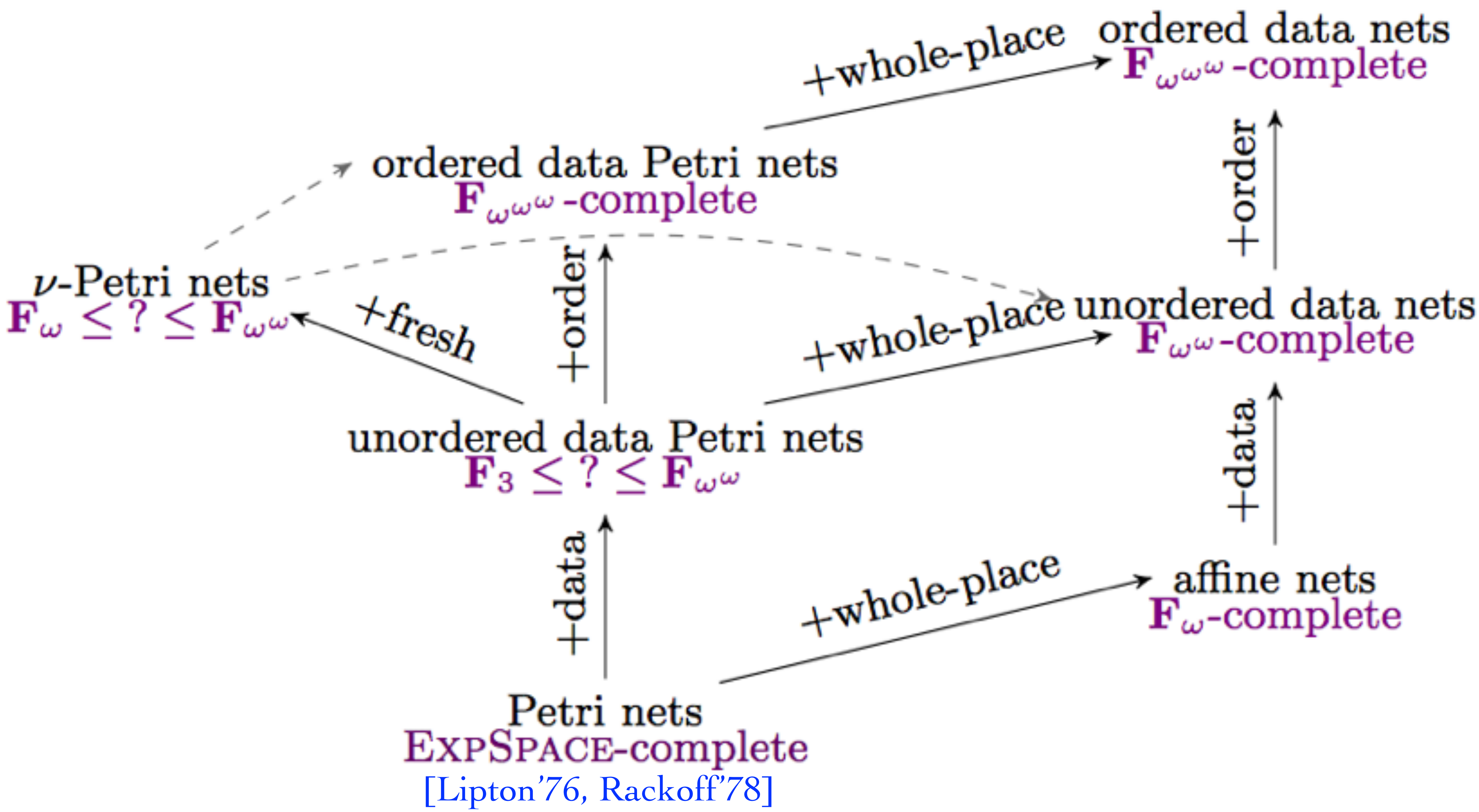
defined up to automorphism



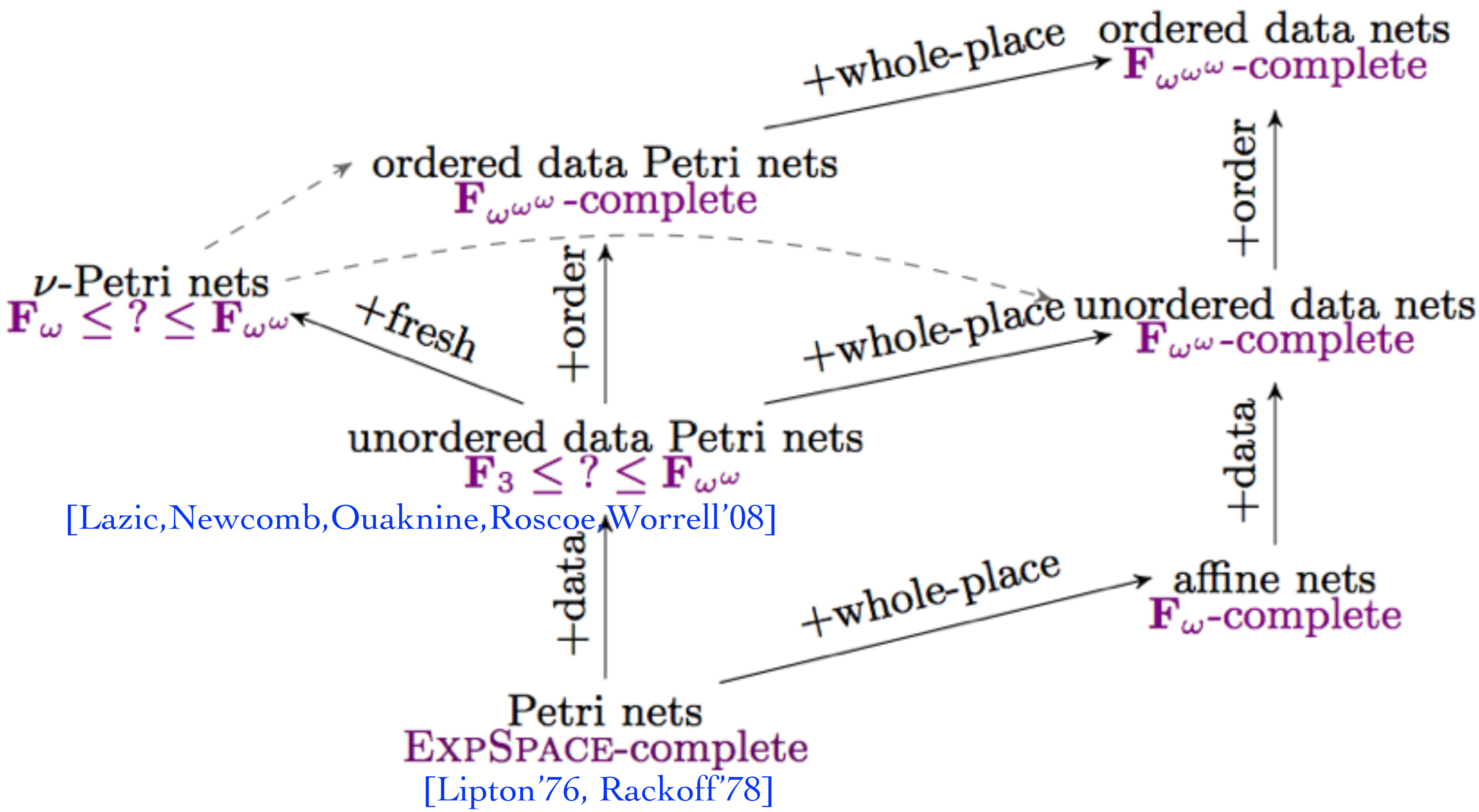
complexity of termination and coverability



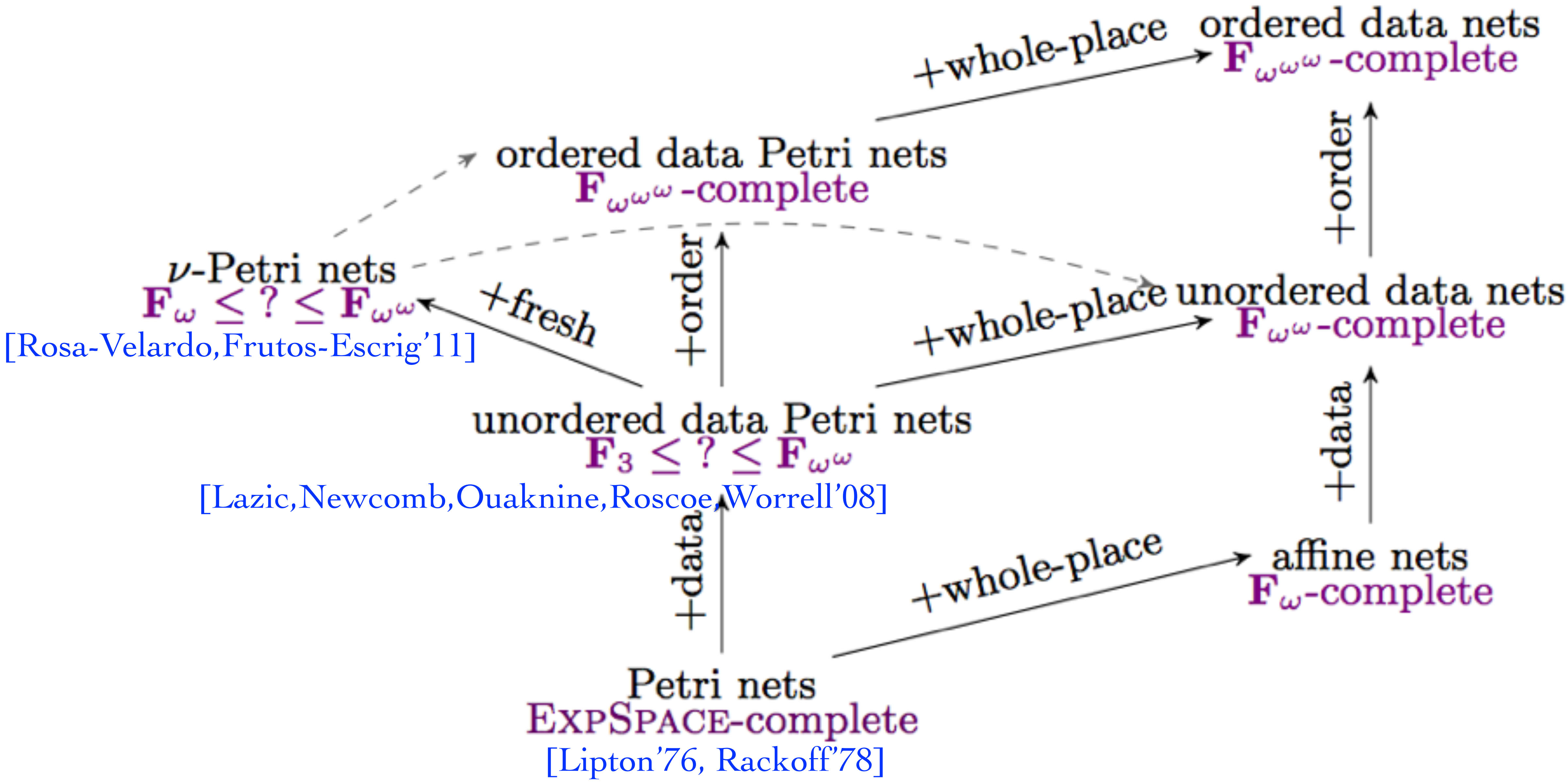
complexity of termination and coverability



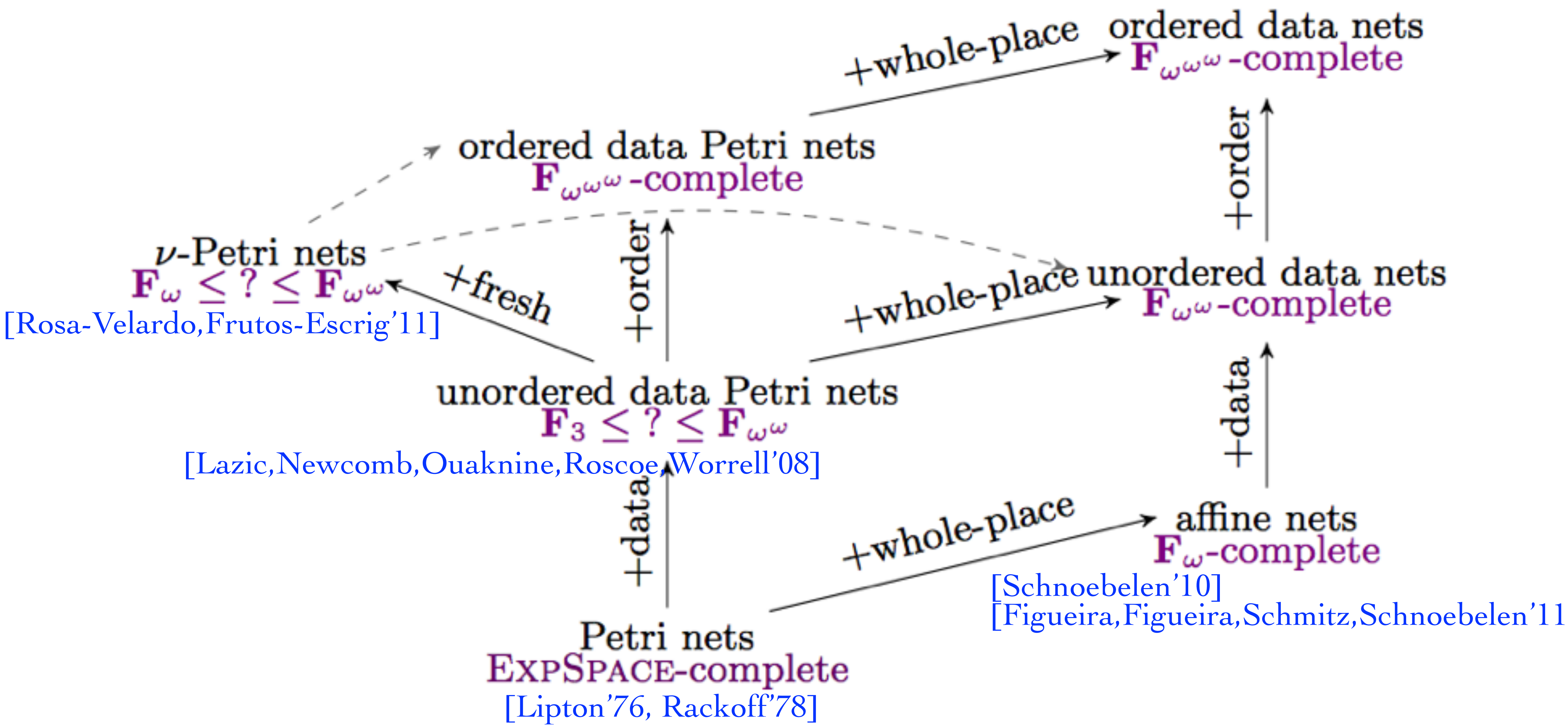
complexity of termination and coverability



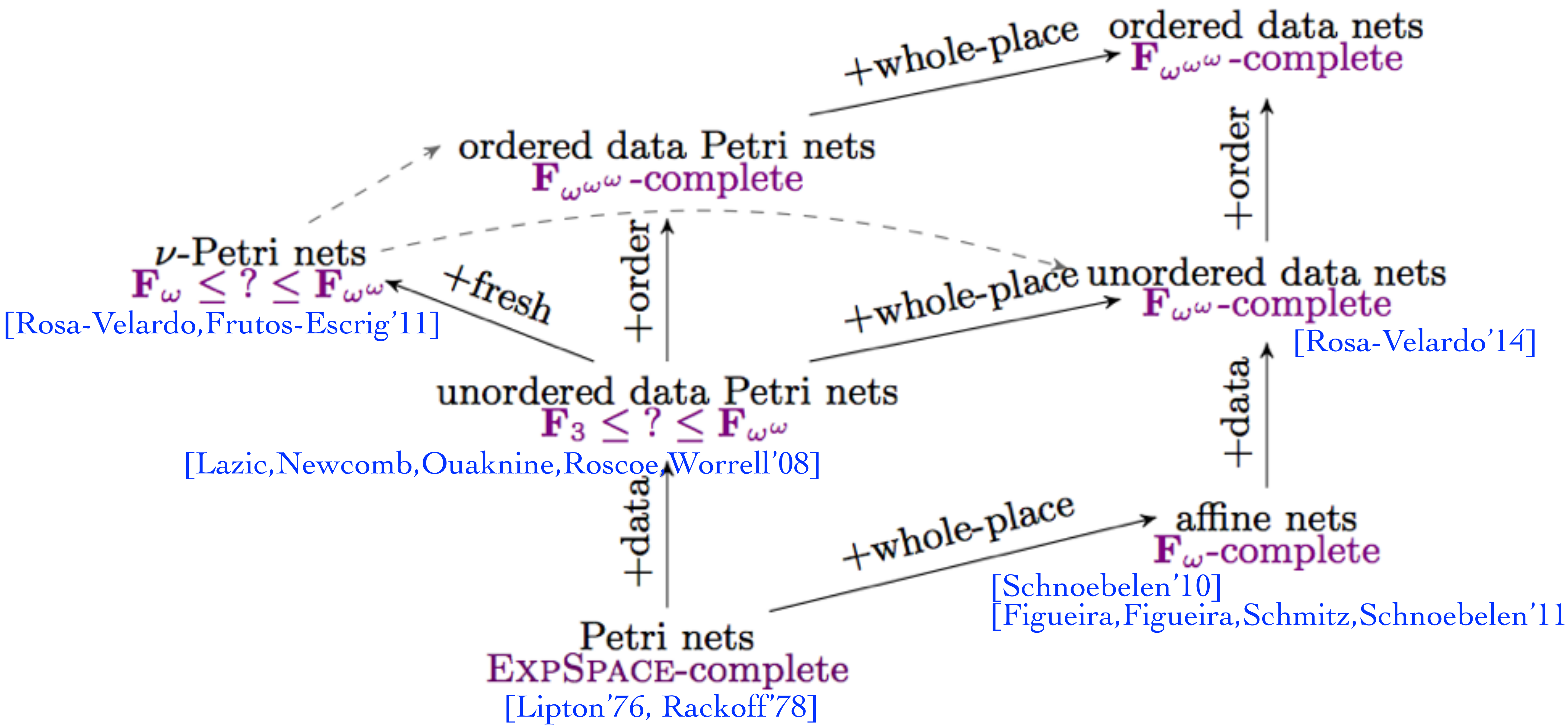
complexity of termination and coverability



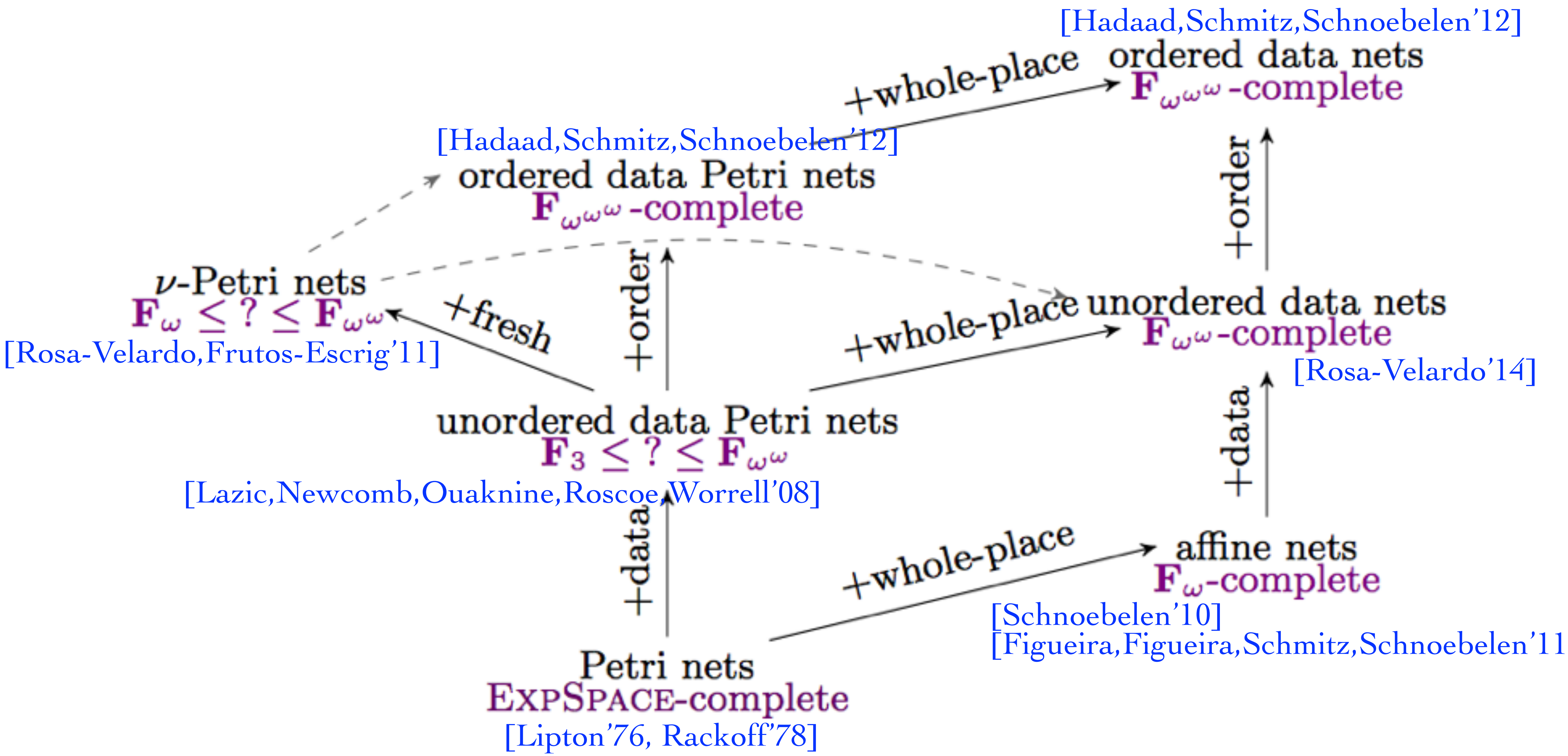
complexity of termination and coverability



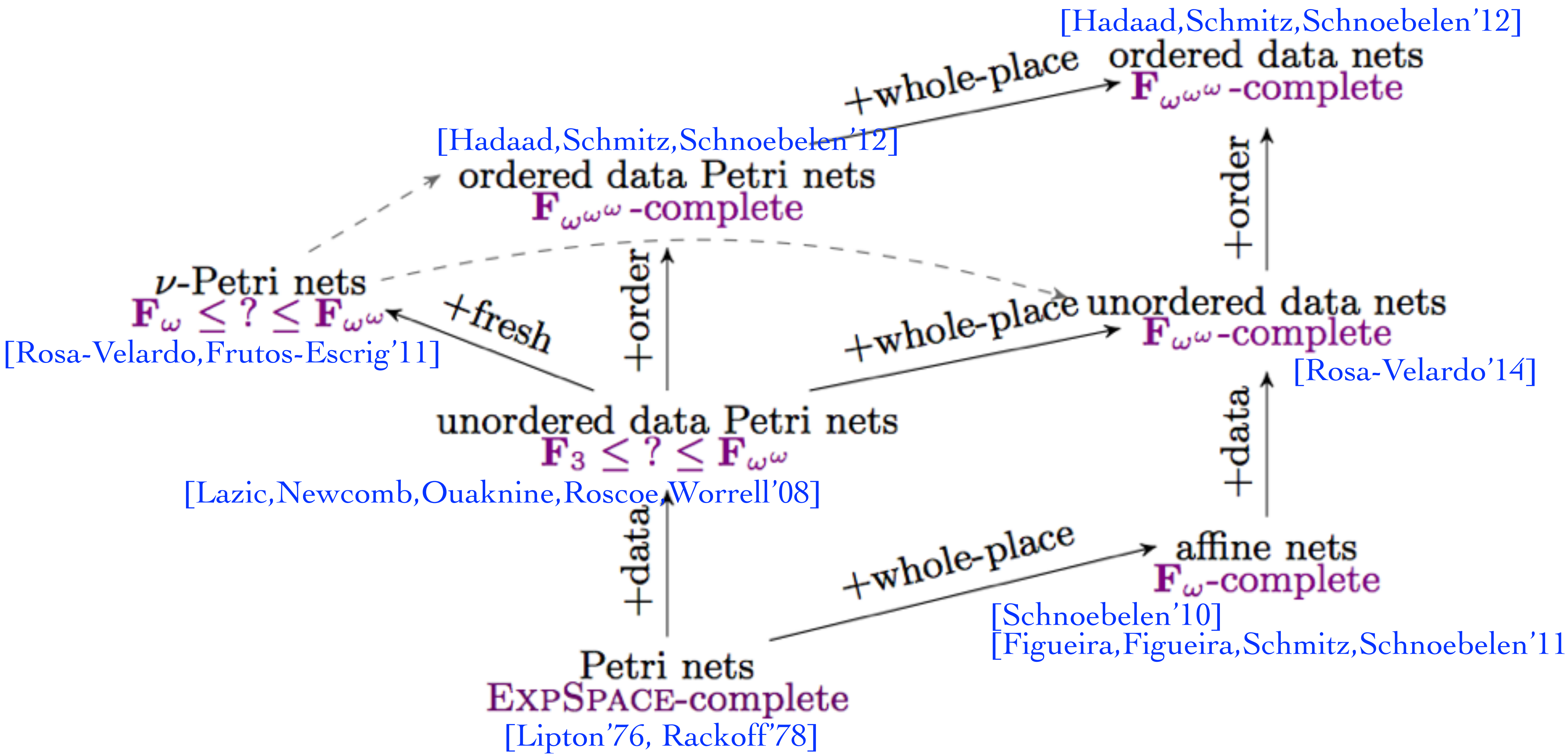
complexity of termination and coverability



complexity of termination and coverability



complexity of termination and coverability



Boundedness undecidable if resets are allowed.

beyond standard problems

input: an (un)ordered Petri net
and an initial configuration.

beyond standard problems

input: an (un)ordered Petri net
and an initial configuration.

- reachability: is a given configuration reachable, up automorphism?

beyond standard problems

input: an (un)ordered Petri net
and an initial configuration.

- reachability: is a given configuration reachable, up automorphism?
- place boundedness: is the set of reachable configurations finite, up to data automorphism, when projected to a given subset of places?

beyond standard problems

input: an (un)ordered Petri net and an initial configuration.

- reachability: is a given configuration reachable, up automorphism?
- place boundedness: is the set of reachable configurations finite, up to data automorphism, when projected to a given subset of places?

} defined up to automorphism

beyond standard problems

input: an (un)ordered Petri net and an initial configuration.

- reachability: is a given configuration reachable, up automorphism?
- place boundedness: is the set of reachable configurations finite, up to data automorphism, when projected to a given subset of places?
- width boundedness: is there a bound on the number of different data values in every reachable configuration?

} defined up to automorphism

beyond standard problems

input: an (un)ordered Petri net and an initial configuration.

- reachability: is a given configuration reachable, up automorphism?
- place boundedness: is the set of reachable configurations finite, up to data automorphism, when projected to a given subset of places?

} defined up to automorphism

- width boundedness: is there a bound on the number of different data values in every reachable configuration?
- depth boundedness: is there a bound on the cardinality of every data value in every reachable configuration?

beyond standard problems

input: an (un)ordered Petri net and an initial configuration.

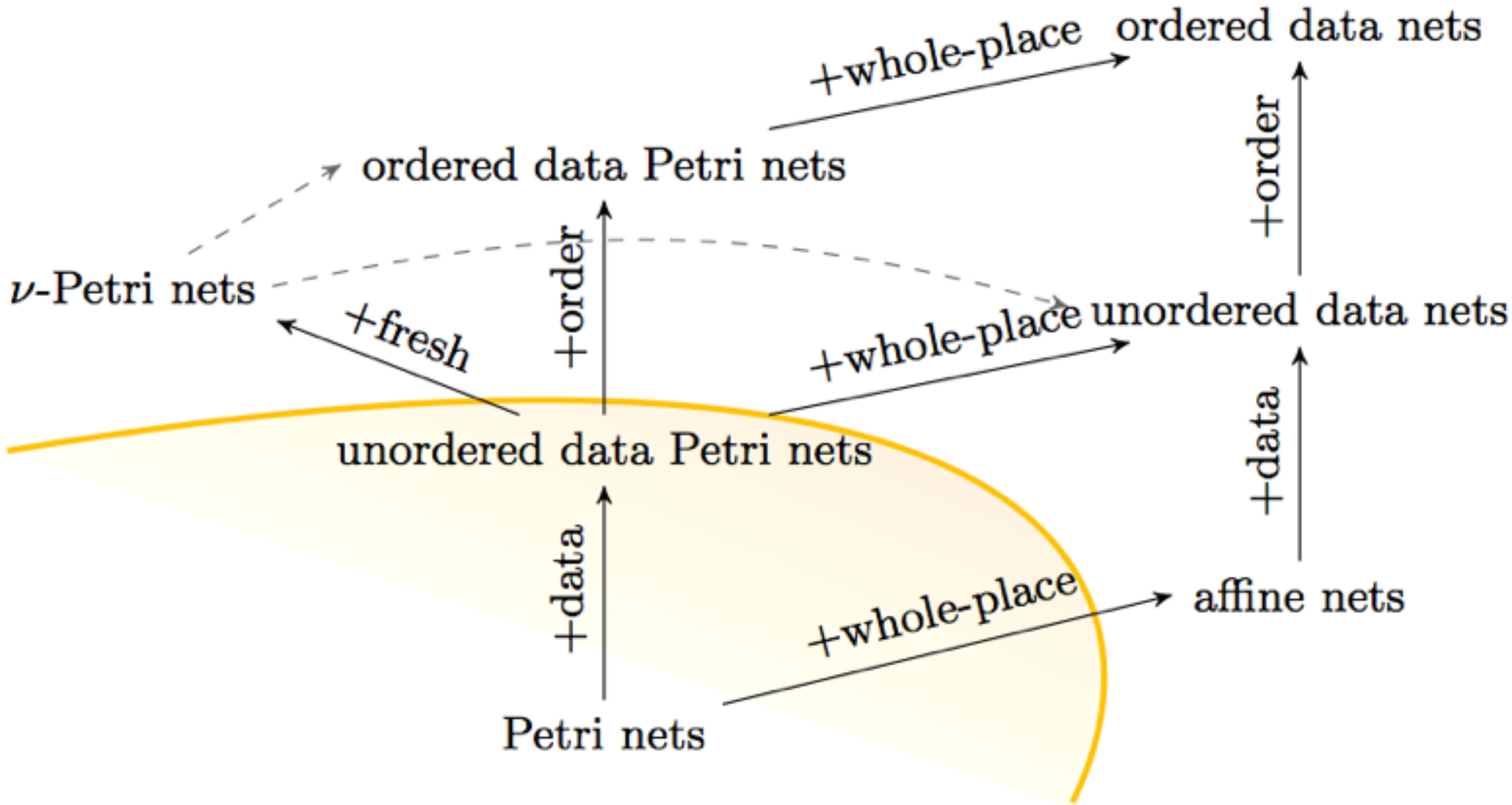
- reachability: is a given configuration reachable, up automorphism?
- place boundedness: is the set of reachable configurations finite, up to data automorphism, when projected to a given subset of places?

} defined up to automorphism

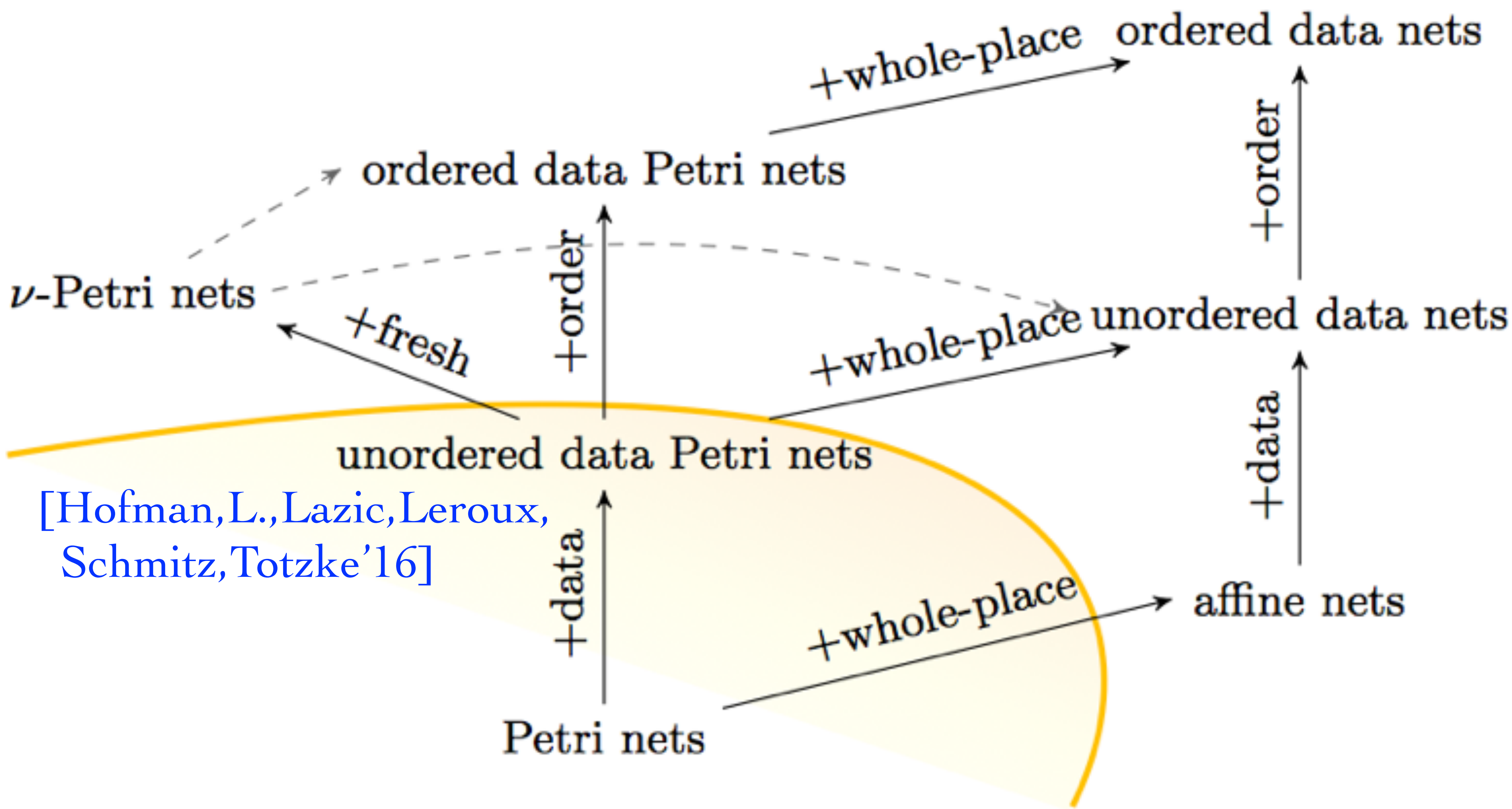
- width boundedness: is there a bound on the number of different data values in every reachable configuration?
- depth boundedness: is there a bound on the cardinality of every data value in every reachable configuration?

} only makes sense in data setting

decidability border for place boundedness

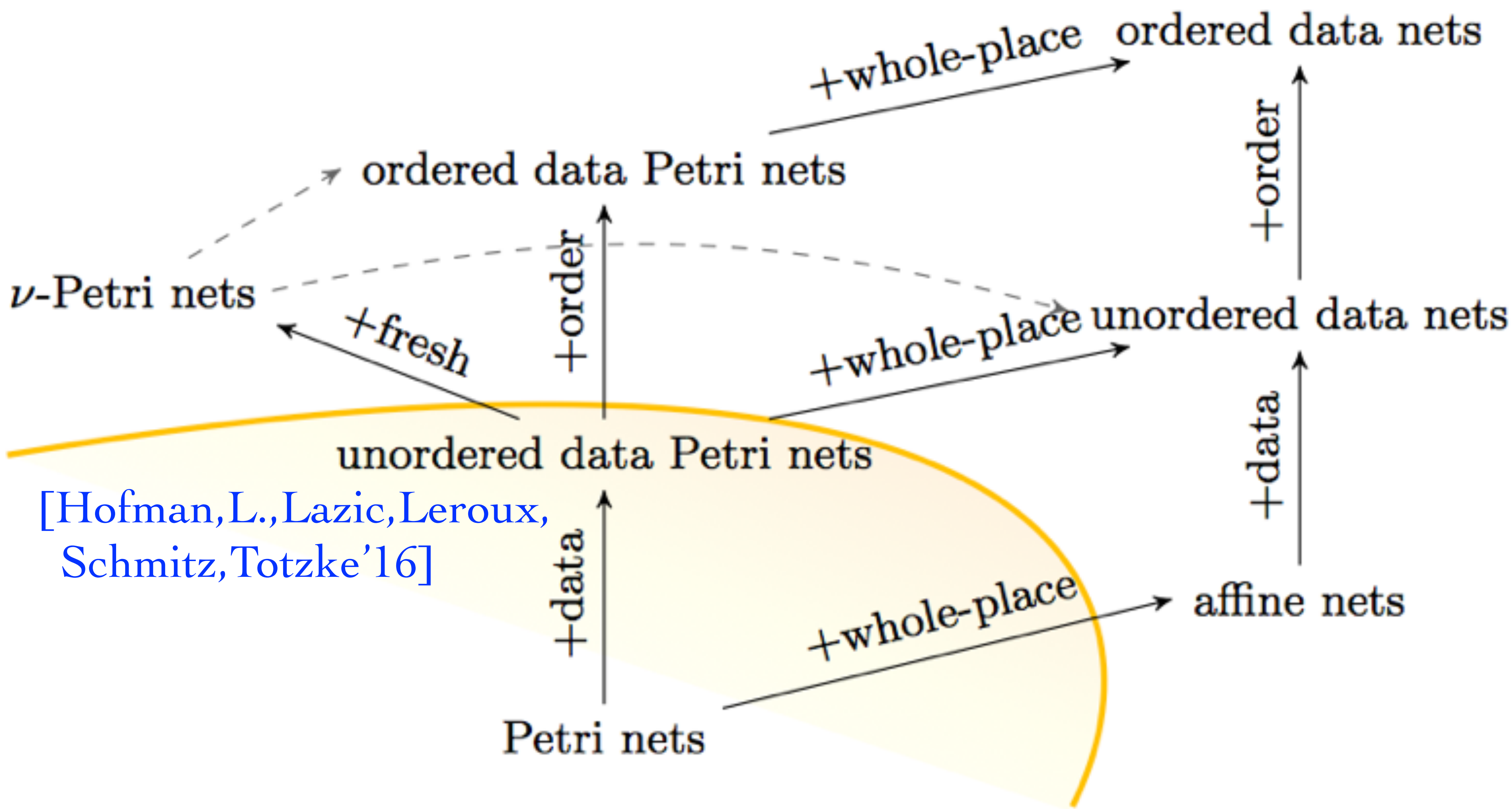


decidability border for place boundedness



[Hofman,L.,Lazic,Leroux,
Schmitz,Totzke'16]

decidability border for place boundedness



Reachability problem for unordered data Petri nets still open!

From now on we only consider standard problems like:

- termination
- coverability
- boundedness

Outline

- (un)ordered data Petri nets
- standard decision problems
- **Petri nets with homogeneous data**
- undecidability
- decidability via wqo

closely related models

closely related models

- high-level Petri nets [Genrich, Lautenbach'81]

closely related models

- high-level Petri nets [Genrich, Lautenbach'81]
- colored Petri nets [Jensen'81]

closely related models

- high-level Petri nets [Genrich, Lautenbach'81]
- colored Petri nets [Jensen'81]
- constraint multiset rewriting [Delzano'05]

closely related models

- high-level Petri nets [Genrich, Lautenbach'81]
- colored Petri nets [Jensen'81]
- constraint multiset rewriting [Delzano'05]
- ...

Petri nets with data

Petri nets with data

- unordered data Petri nets = Petri nets with data $(\mathbb{N}, =)$

Petri nets with data

- unordered data Petri nets = Petri nets with data $(\mathbb{N}, =)$
- ordered data Petri nets = Petri nets with data $(\mathbb{Q}, <)$

Petri nets with data

- unordered data Petri nets = Petri nets with data $(\mathbb{N}, =)$
- ordered data Petri nets = Petri nets with data $(\mathbb{Q}, <)$
- ...?

Petri nets with data

- unordered data Petri nets = Petri nets with data $(\mathbb{N}, =)$
- ordered data Petri nets = Petri nets with data $(\mathbb{Q}, <)$
- ...?

Fix a countably infinite relational structure $\mathbb{A}$ over a finite vocabulary, and call it data domain. This yields Petri nets with data $\mathbb{A}$.

Petri nets with data

- unordered data Petri nets = Petri nets with data $(\mathbb{N}, =)$
- ordered data Petri nets = Petri nets with data $(\mathbb{Q}, <)$
- ...?

Fix a countably infinite relational structure $\mathbb{A}$ over a finite vocabulary, and call it data domain. This yields Petri nets with data $\mathbb{A}$.

Data domain is a parameter in the following.

Petri nets with data

- unordered data Petri nets = Petri nets with data $(\mathbb{N}, =)$
- ordered data Petri nets = Petri nets with data $(\mathbb{Q}, <)$
- ...?

Fix a countably infinite relational structure $\mathbb{A}$ over a finite vocabulary, and call it data domain. This yields Petri nets with data $\mathbb{A}$.

Data domain is a parameter in the following.

Automorphisms of $\mathbb{A}$ we call data automorphisms.

example data domains

example data domains

data domain $\mathbb{A}$	data automorphisms $\mathrm{Aut}(\mathbb{A})$

example data domains

data domain $\mathbb{A}$	data automorphisms $\mathrm{Aut}(\mathbb{A})$
equality data $(\mathbb{N}, =)$	all bijections

example data domains

data domain $\mathbb{A}$	data automorphisms $\mathrm{Aut}(\mathbb{A})$
equality data $(\mathbb{N}, =)$	all bijections
total order data $(\mathbb{Q}, <)$	monotonic bijections

example data domains

data domain $\mathbb{A}$	data automorphisms $Aut(\mathbb{A})$
equality data $(\mathbb{N}, =)$	all bijections
total order data $(\mathbb{Q}, <)$	monotonic bijections
dense-timed data $(\mathbb{Q}, <, +1)$	monotonic bijections preserving integer differences

example data domains

data domain $\mathbb{A}$	data automorphisms $Aut(\mathbb{A})$
equality data $(\mathbb{N}, =)$	all bijections
total order data $(\mathbb{Q}, <)$	monotonic bijections
dense-timed data $(\mathbb{Q}, <, +1)$	monotonic bijections preserving integer differences
discrete-timed data $(\mathbb{Z}, <, +1)$	translations

example data domains

data domain $\mathbb{A}$	data automorphisms $\mathrm{Aut}(\mathbb{A})$
equality data $(\mathbb{N}, =)$	all bijections
total order data $(\mathbb{Q}, <)$	monotonic bijections
dense-timed data $(\mathbb{Q}, <, +1)$	monotonic bijections preserving integer differences
discrete-timed data $(\mathbb{Z}, <, +1)$	translations
universal equivalence $(\mathbb{A}, R, =)$	equivalence-preserving bijections

example data domains

data domain $\mathbb{A}$	data automorphisms Aut($\mathbb{A}$)
equality data ($\mathbb{N}$, =)	all bijections
total order data ($\mathbb{Q}$, <)	monotonic bijections
dense-timed data ($\mathbb{Q}$, <, +1)	monotonic bijections preserving integer differences
discrete-timed data ($\mathbb{Z}$, <, +1)	translations
universal equivalence ($\mathbb{A}$, R, =)	equivalence-preserving bijections
universal (random) graph	random graph automorphisms

example data domains

data domain $\mathbb{A}$	data automorphisms Aut($\mathbb{A}$)
equality data ($\mathbb{N}$, =)	all bijections
total order data ($\mathbb{Q}$, <)	monotonic bijections
dense-timed data ($\mathbb{Q}$, <, +1)	monotonic bijections preserving integer differences
discrete-timed data ($\mathbb{Z}$, <, +1)	translations
universal equivalence ($\mathbb{A}$, R, =)	equivalence-preserving bijections
universal (random) graph	random graph automorphisms
...	...

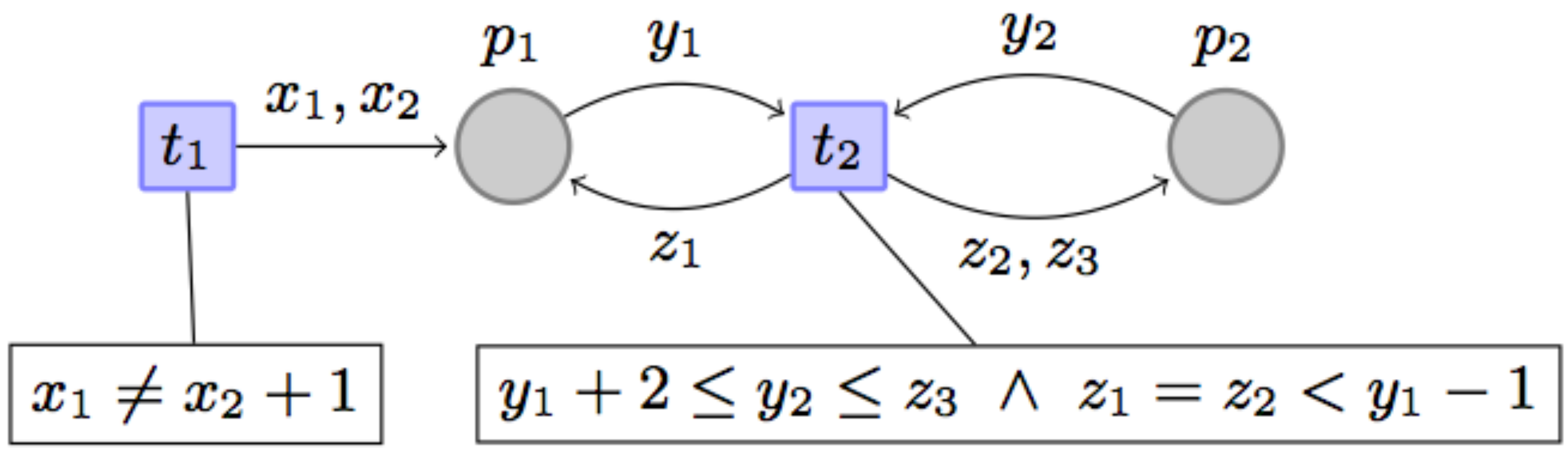
examples

examples

- data domain $(\mathbb{Q}, <, +1)$

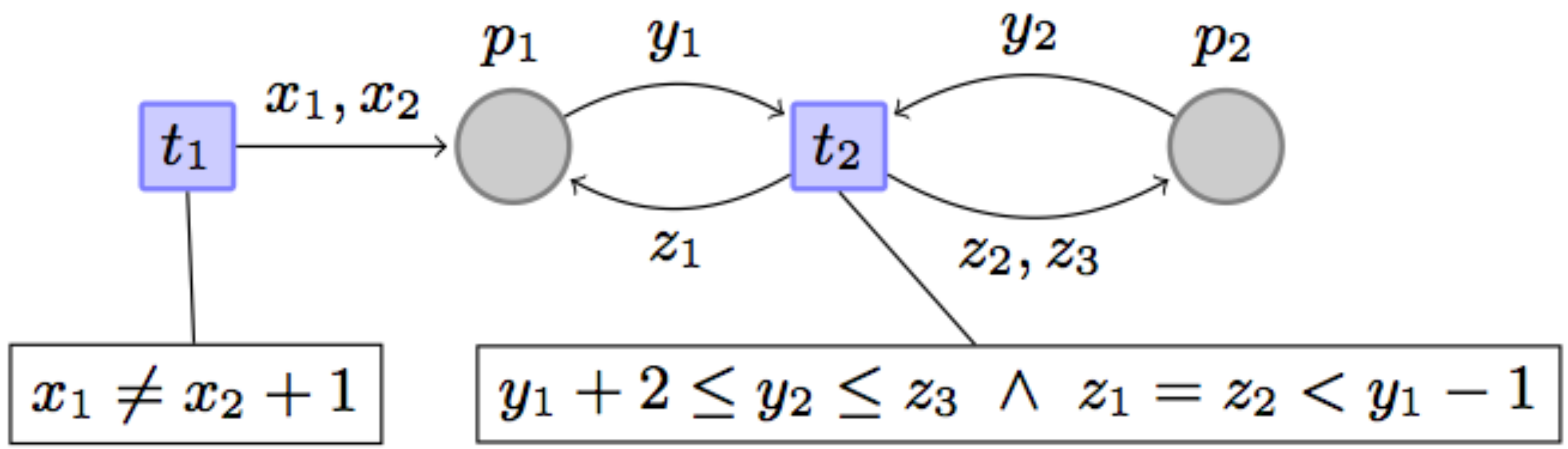
examples

- data domain $(\mathbb{Q}, <, +1)$



examples

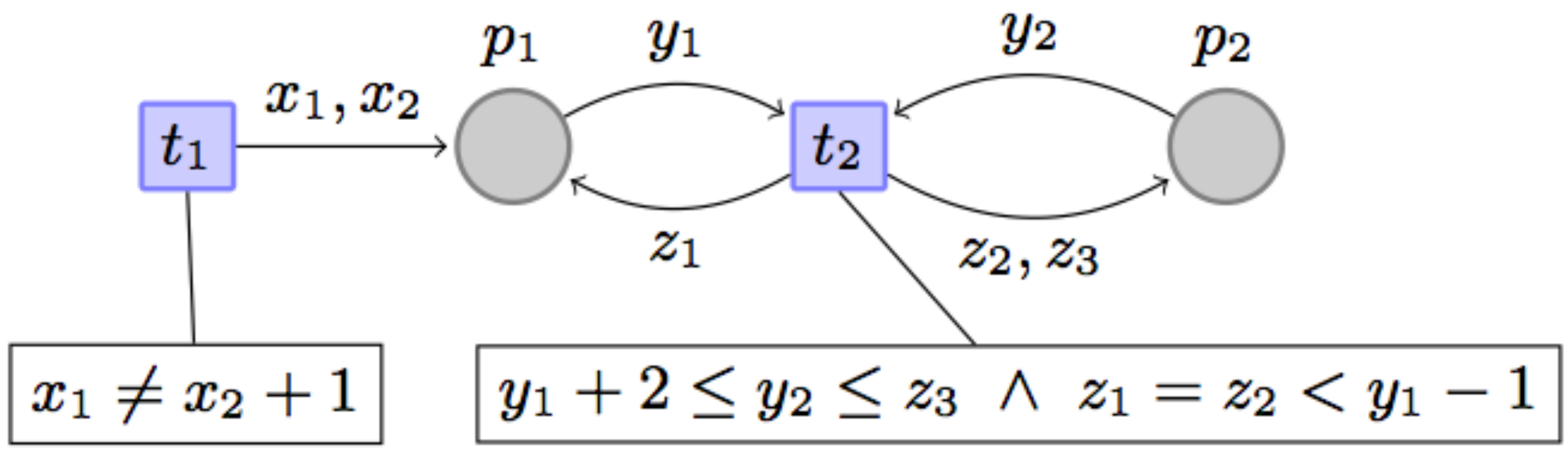
- data domain $(\mathbb{Q}, <, +1)$



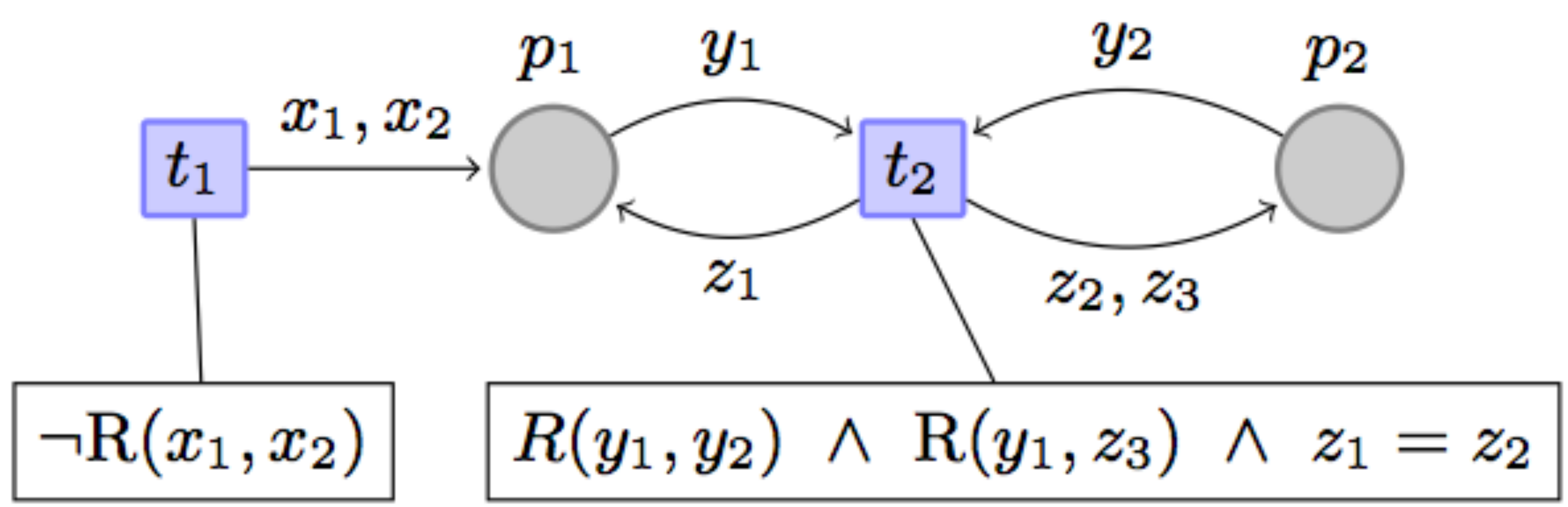
- data domain $(\mathbb{A}, \mathrm{R}, =) \;\simeq\; (\mathbb{N}^2, =_1, =)$

examples

- data domain $(\mathbb{Q}, <, +1)$



- data domain $(\mathbb{A}, \mathrm{R}, =) \;\simeq\; (\mathbb{N}^2, =_1, =)$



configurations are substructures of $\mathbb{A}$

configurations are substructures of $\mathbb{A}$

- configurations: $M(\mathrm{P} \times \mathbb{A})$

configurations are substructures of $\mathbb{A}$

- configurations: $M(\mathrm{P} \times \mathbb{A})$
- in other words, a configuration is a finite induced structure of $\mathbb{A}$, labeled by elements of $M(\mathrm{P})$

configurations are substructures of $\mathbb{A}$

restriction to a subset

- configurations: $M(\mathrm{P} \times \mathbb{A})$
- in other words, a configuration is a finite induced structure of $\mathbb{A}$, labeled by elements of $M(\mathrm{P})$

configurations are substructures of $\mathbb{A}$

restriction to a subset

- configurations: $M(\mathrm{P} \times \mathbb{A})$
- in other words, a configuration is a finite induced structure of $\mathbb{A}$, labeled by elements of $M(\mathrm{P})$
- denote by age($\mathbb{A}$) the class of all finite induced substructures of $\mathbb{A}$

configurations are substructures of $\mathbb{A}$

restriction to a subset

- configurations: $M(\mathrm{P} \times \mathbb{A})$
- in other words, a configuration is a finite induced structure of $\mathbb{A}$, labeled by elements of $M(\mathrm{P})$
- denote by age($\mathbb{A}$) the class of all finite induced substructures of $\mathbb{A}$
- semantics by multiset rewriting

homogeneous structures

homogeneous structures

A relational structure $\mathbb{A}$ is homogeneous

if

homogeneous structures

A relational structure $\mathbb{A}$ is homogeneous

if

every isomorphism of finite induced substructures of $\mathbb{A}$

homogeneous structures

A relational structure $\mathbb{A}$ is homogeneous

if

every isomorphism of finite induced substructures of $\mathbb{A}$ extends to an automorphism of the whole structure.

homogeneous structures

A relational structure $\mathbb{A}$ is homogeneous

if

every isomorphism of finite induced substructures of $\mathbb{A}$ extends to an automorphism of the whole structure.

Example: $(\mathbb{Q}, \leq)$

homogeneous structures

A relational structure $\mathbb{A}$ is homogeneous

if

every isomorphism of finite induced substructures of $\mathbb{A}$ extends to an automorphism of the whole structure.

Example: $(\mathbb{Q}, \leq)$

homogeneous structures

A relational structure $\mathbb{A}$ is homogeneous

if

every isomorphism of finite induced substructures of $\mathbb{A}$
extends to an automorphism of the whole structure.

Example: $(\mathbb{Q}, \leq)$

homogeneous structures

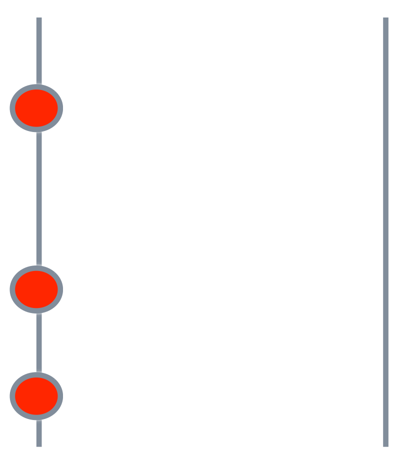
A relational structure $\mathbb{A}$ is homogeneous

if

every isomorphism of finite induced substructures of $\mathbb{A}$

extends to an automorphism of the whole structure.

Example: $(\mathbb{Q}, \leq)$



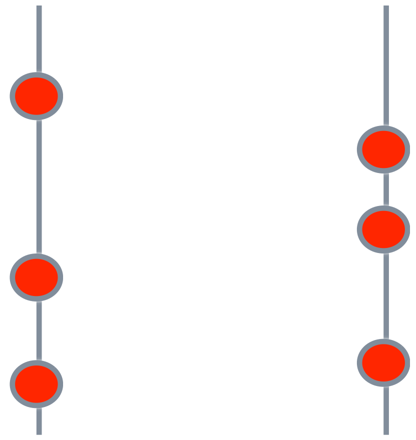
homogeneous structures

A relational structure $\mathbb{A}$ is homogeneous

if

every isomorphism of finite induced substructures of $\mathbb{A}$ extends to an automorphism of the whole structure.

Example: $(\mathbb{Q}, \leq)$



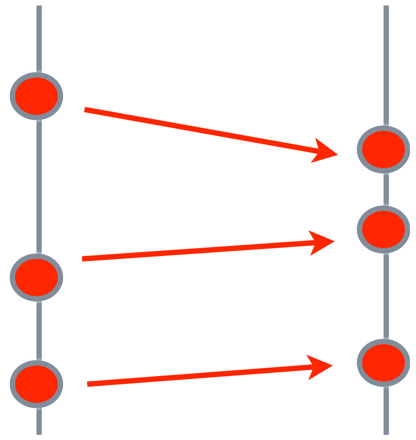
homogeneous structures

A relational structure $\mathbb{A}$ is homogeneous

if

every isomorphism of finite induced substructures of $\mathbb{A}$ extends to an automorphism of the whole structure.

Example: $(\mathbb{Q}, \leq)$



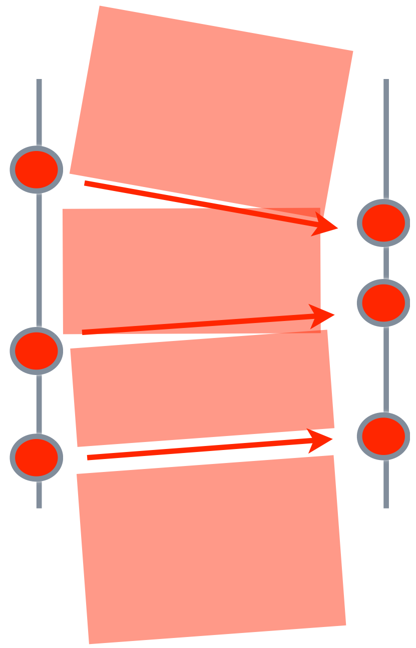
homogeneous structures

A relational structure $\mathbb{A}$ is homogeneous

if

every isomorphism of finite induced substructures of $\mathbb{A}$
extends to an automorphism of the whole structure.

Example: $(\mathbb{Q}, \leq)$



homogeneous structures

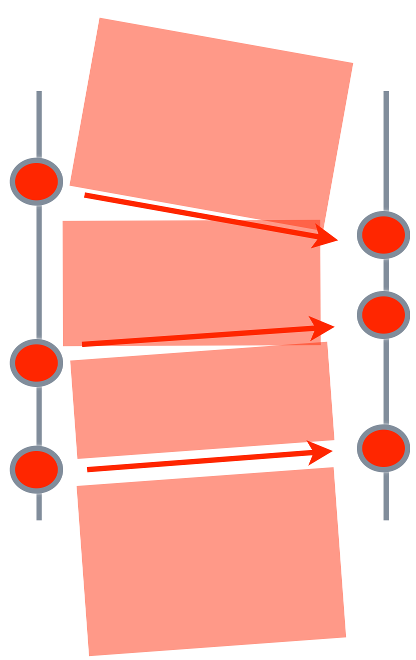
A relational structure $\mathbb{A}$ is homogeneous

if

every isomorphism of finite induced substructures of $\mathbb{A}$

extends to an automorphism of the whole structure.

Example: $(\mathbb{Q}, \leq)$



Fact:

Isomorphic configurations are equal up to data automorphism.

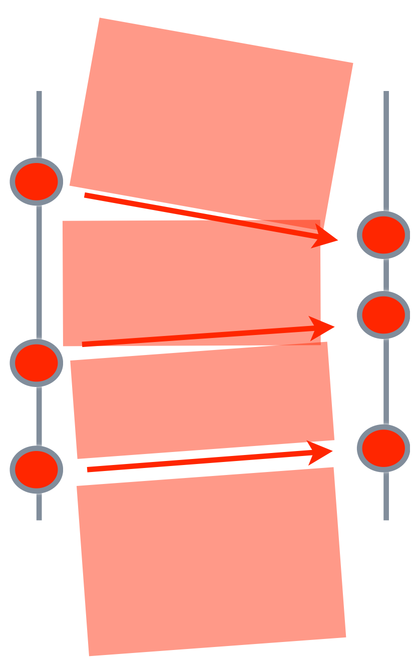
homogeneous structures

A relational structure $\mathbb{A}$ is homogeneous

if

every isomorphism of finite induced substructures of $\mathbb{A}$ extends to an automorphism of the whole structure.

Example: $(\mathbb{Q}, \leq)$

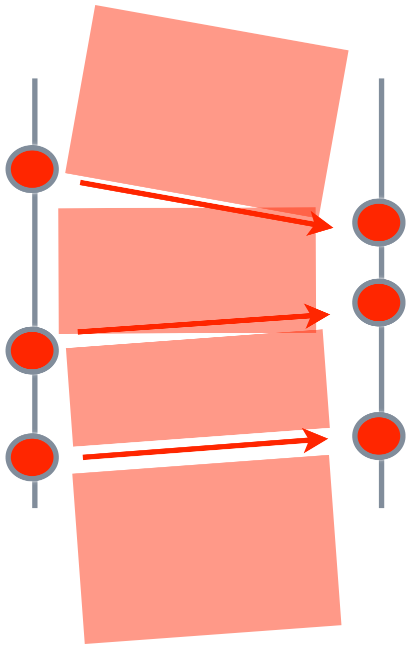


Fact:

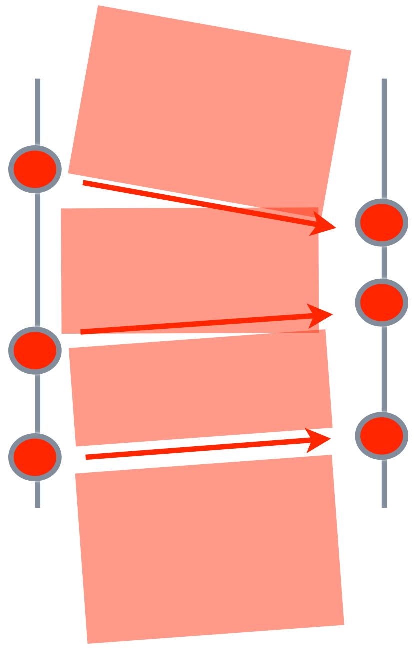
Isomorphic configurations are equal up to data automorphism.

Thus a configuration can be finitely represented by its isomorphism type.

homogeneous data domains

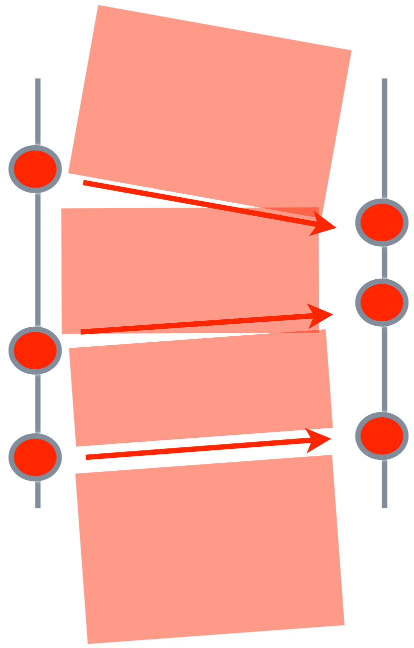


homogeneous data domains



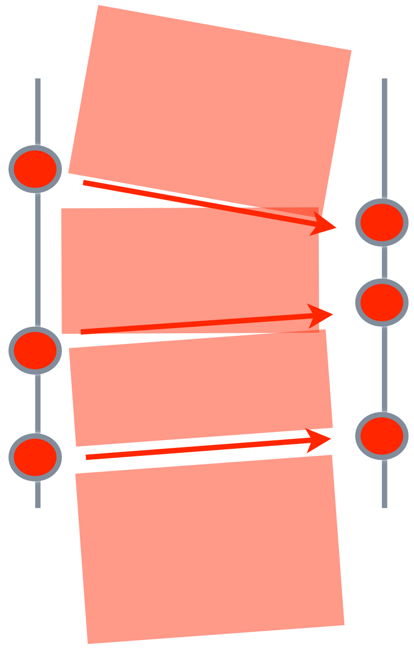
data domain $\mathbb{A}$	data automorphisms $Aut(\mathbb{A})$
equality data $(\mathbb{N}, =)$	all bijections
total order data $(\mathbb{Q}, <)$	monotonic bijections

homogeneous data domains



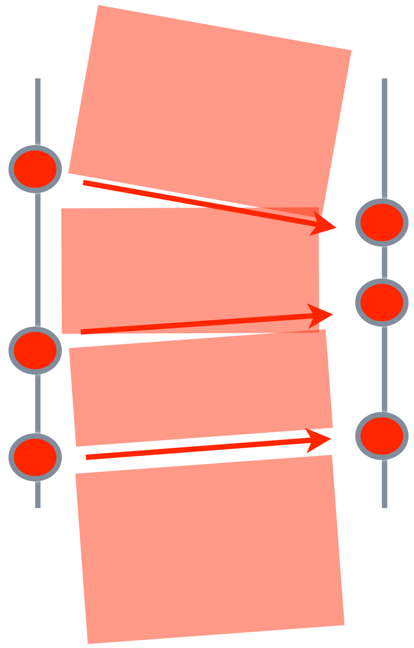
data domain $\mathbb{A}$	data automorphisms $\mathrm{Aut}(\mathbb{A})$
equality data $(\mathbb{N}, =)$	all bijections
total order data $(\mathbb{Q}, <)$	monotonic bijections
dense-timed data $(\mathbb{Q}, <, +1)$	monotonic bijections preserving difference 1

homogeneous data domains



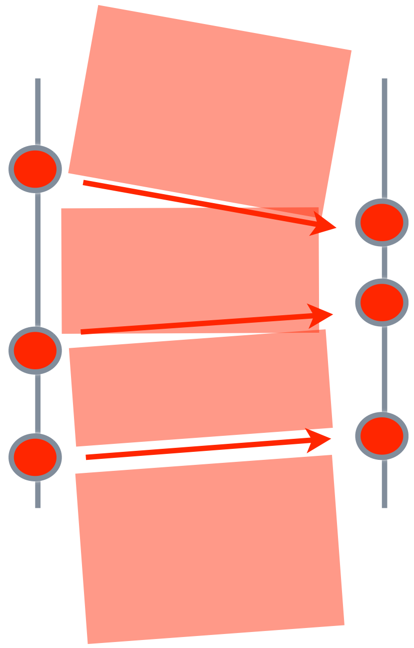
data domain $\mathbb{A}$	data automorphisms Aut($\mathbb{A}$)
equality data ($\mathbb{N}$, =)	all bijections
total order data ($\mathbb{Q}$, <)	monotonic bijections
~~dense-timed data ($\mathbb{Q}$, <, +1)~~	monotonic bijections preserving difference 1

homogeneous data domains



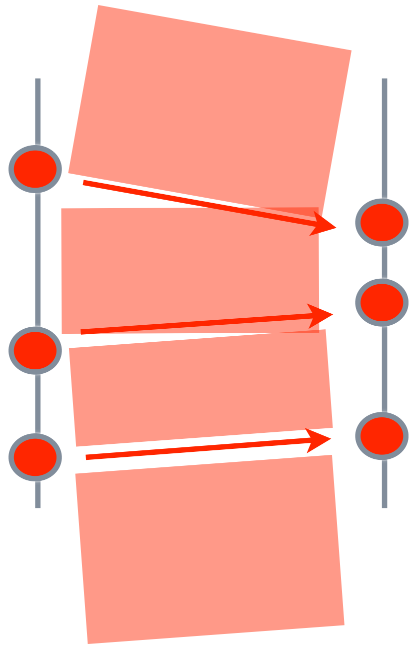
data domain $\mathbb{A}$	data automorphisms Aut($\mathbb{A}$)
equality data ($\mathbb{N}$, =)	all bijections
total order data ($\mathbb{Q}$, <)	monotonic bijections
~~dense-timed data ($\mathbb{Q}$, <, +1~~)	monotonic bijections preserving difference 1
discrete-timed data ($\mathbb{Z}$, <, +1)	translations

homogeneous data domains



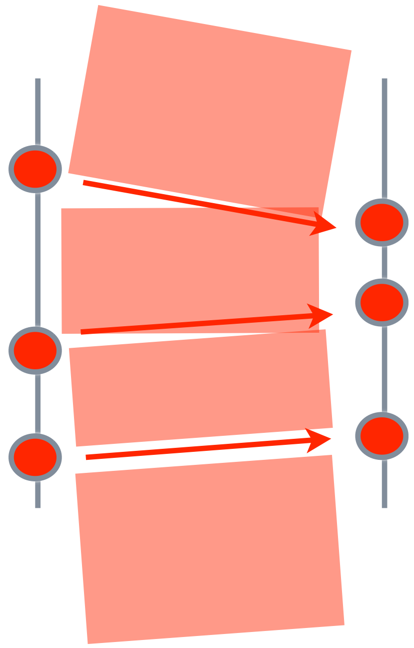
data domain $\mathbb{A}$	data automorphisms Aut($\mathbb{A}$)
equality data ($\mathbb{N}$, =)	all bijections
total order data ($\mathbb{Q}$, <)	monotonic bijections
~~dense-timed data ($\mathbb{Q}$, <, +1~~)	monotonic bijections preserving difference 1
~~discrete-timed data ($\mathbb{Z}$, <, +1~~)	translations

homogeneous data domains



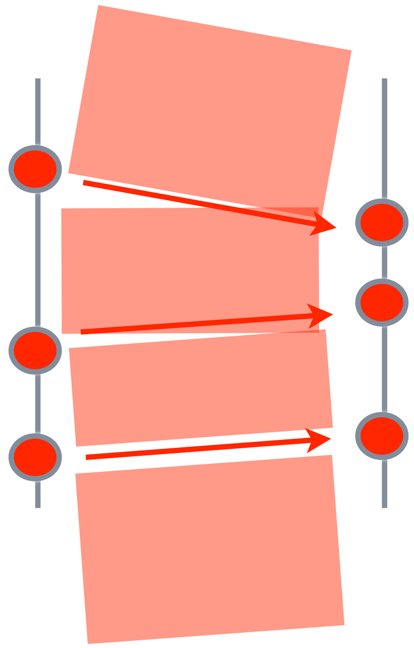
data domain $\mathbb{A}$	data automorphisms $Aut(\mathbb{A})$
equality data $(\mathbb{N}, =)$	all bijections
total order data $(\mathbb{Q}, <)$	monotonic bijections
~~dense-timed data $(\mathbb{Q}, <, +1)$~~	monotonic bijections preserving difference 1
~~discrete-timed data $(\mathbb{Z}, <, +1)$~~	translations
universal equivalence $(\mathbb{A}, R, =)$	equivalence-preserving bijections

homogeneous data domains



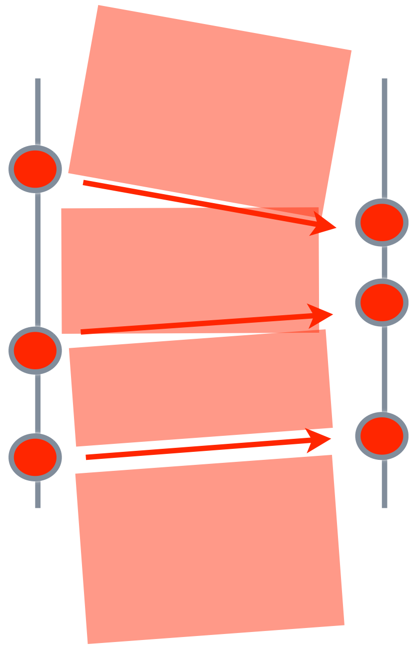
data domain $\mathbb{A}$	data automorphisms Aut($\mathbb{A}$)
equality data ($\mathbb{N}$, =)	all bijections
total order data ($\mathbb{Q}$, <)	monotonic bijections
~~dense-timed data ($\mathbb{Q}$, <, +1~~)	monotonic bijections preserving difference 1
~~discrete-timed data ($\mathbb{Z}$, <, +1~~)	translations
universal equivalence ($\mathbb{A}$, R, =)	equivalence-preserving bijections
universal (random) graph	random graph automorphisms

homogeneous data domains



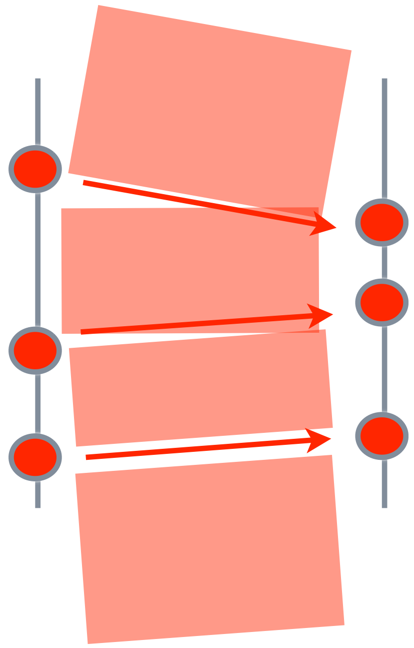
data domain $\mathbb{A}$	data automorphisms Aut($\mathbb{A}$)
equality data ($\mathbb{N}$, =)	all bijections
total order data ($\mathbb{Q}$, <)	monotonic bijections
~~dense-timed data ($\mathbb{Q}$, <, +1~~)	monotonic bijections preserving difference 1
~~discrete-timed data ($\mathbb{Z}$, <, +1~~)	translations
universal equivalence ($\mathbb{A}$, R, =)	equivalence-preserving bijections
universal (random) graph	random graph automorphisms
universal partial order	...

homogeneous data domains



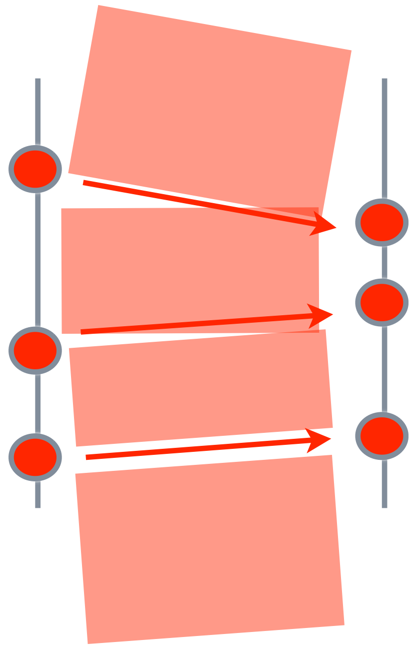
data domain $\mathbb{A}$	data automorphisms $Aut(\mathbb{A})$
equality data $(\mathbb{N}, =)$	all bijections
total order data $(\mathbb{Q}, <)$	monotonic bijections
~~dense-timed data $(\mathbb{Q}, <, +1$~~)	monotonic bijections preserving difference 1
~~discrete-timed data $(\mathbb{Z}, <, +1$~~)	translations
universal equivalence $(\mathbb{A}, R, =)$	equivalence-preserving bijections
universal (random) graph	random graph automorphisms
universal partial order	...
universal directed graph	...

homogeneous data domains



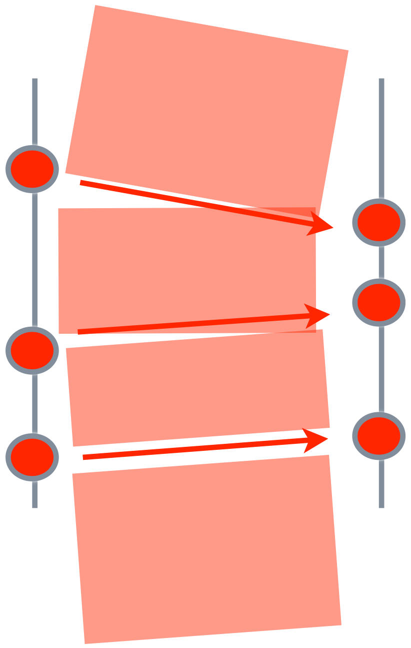
data domain $\mathbb{A}$	data automorphisms Aut($\mathbb{A}$)
equality data ($\mathbb{N}$, =)	all bijections
total order data ($\mathbb{Q}$, <)	monotonic bijections
~~dense-timed data ($\mathbb{Q}$, <, +1)~~	monotonic bijections preserving difference 1
~~discrete-timed data ($\mathbb{Z}$, <, +1)~~	translations
universal equivalence ($\mathbb{A}$, R, =)	equivalence-preserving bijections
universal (random) graph	random graph automorphisms
universal partial order	...
universal directed graph	...
universal tournament	...

homogeneous data domains

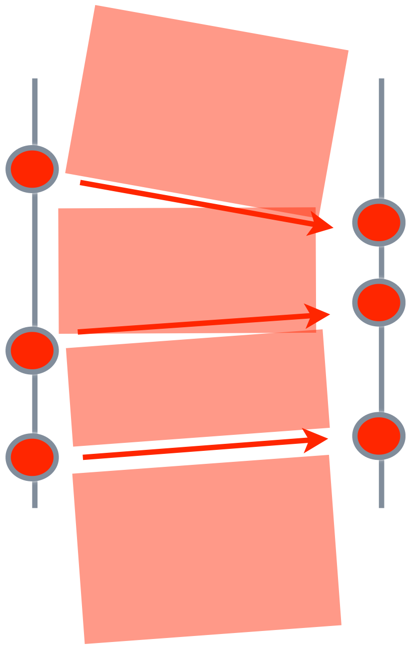


data domain $\mathbb{A}$	data automorphisms Aut($\mathbb{A}$)
equality data ($\mathbb{N}$, =)	all bijections
total order data ($\mathbb{Q}$, <)	monotonic bijections
~~dense-timed data ($\mathbb{Q}$, <, +1)~~	monotonic bijections preserving difference 1
~~discrete-timed data ($\mathbb{Z}$, <, +1)~~	translations
universal equivalence ($\mathbb{A}$, R, =)	equivalence-preserving bijections
universal (random) graph	random graph automorphisms
universal partial order	...
universal directed graph	...
universal tournament	...
...	...

quantifier elimination



quantifier elimination



Theorem:

Homogeneous structures admit quantifier elimination: every first-order formula is equivalent to a quantifier-free one.

amalgamation classes

amalgamation classes

A class of finite structures is amalgamation class

if

amalgamation classes

A class of finite structures is amalgamation class

if

it is closed under induced substructures

amalgamation classes

a way of glueing two structures with common part

A class of finite structures is amalgamation class

if

it is closed under induced substructures
and admits amalgamation.

amalgamation classes

a way of glueing two structures with common part

A class of finite structures is amalgamation class

if

it is closed under induced substructures
and admits amalgamation.

Theorem: [Freisse 1953] There is a one-to-one correspondence between

homogeneous structures

amalgamation classes

amalgamation classes

a way of glueing two structures with common part

A class of finite structures is amalgamation class

if

it is closed under induced substructures
and admits amalgamation.

the class of all finite induced substructures

Theorem: [Freisse 1953] There is a one-to-one correspondence between

homogeneous structures $\xrightarrow{\text{age}}$ amalgamation classes

amalgamation classes

a way of glueing two structures with common part

A class of finite structures is amalgamation class

if

it is closed under induced substructures
and admits amalgamation.

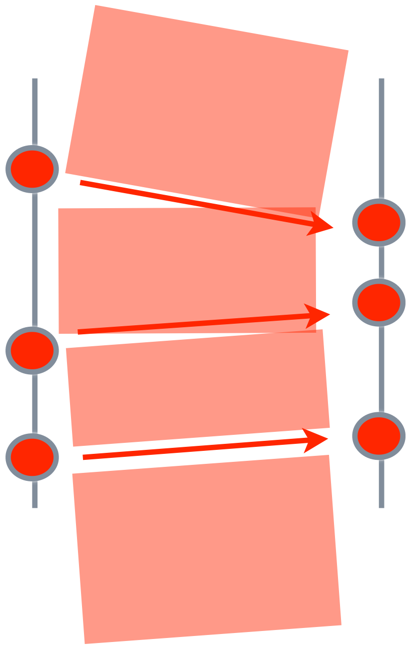
the class of all finite induced substructures

Theorem: [Freisse 1953] There is a one-to-one correspondence between

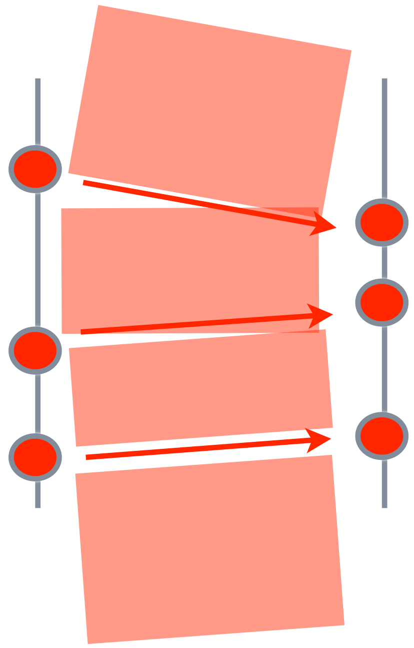
homogeneous structures —age→ amalgamation classes

homogeneous structures ←Freisse limit— amalgamation classes

homogeneous data domains

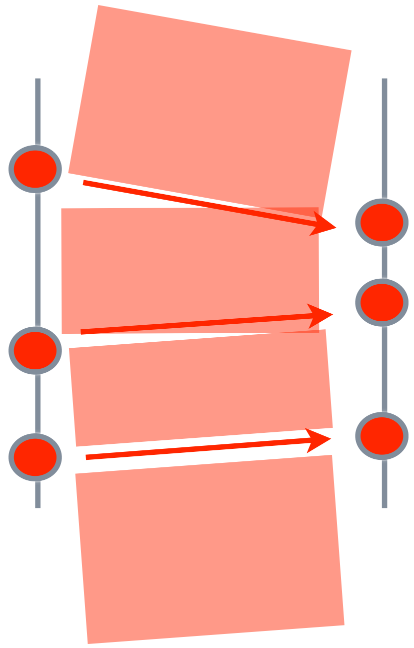


homogeneous data domains



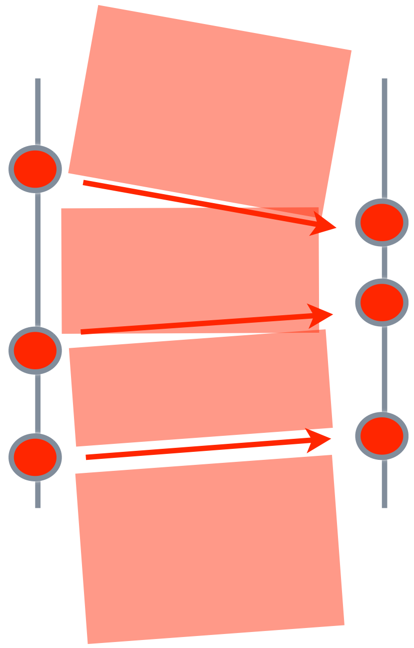
data domain $\mathbb{A}$	amalgamation class
equality data $(\mathbb{N}, =)$	finite pure sets
total order data $(\mathbb{Q}, <)$	finite total orders

homogeneous data domains



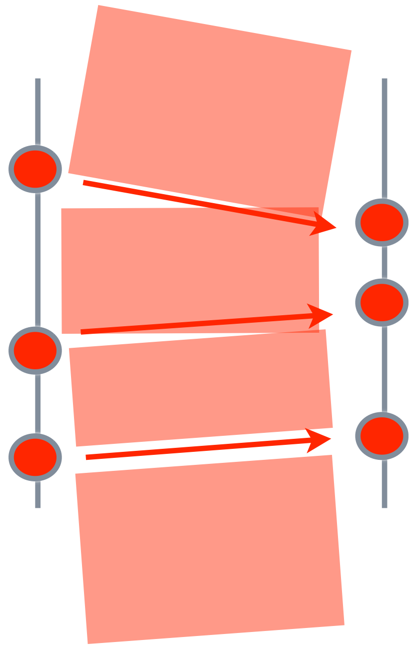
data domain $\mathbb{A}$	amalgamation class
equality data $(\mathbb{N}, =)$	finite pure sets
total order data $(\mathbb{Q}, <)$	finite total orders
~~dense-time data $(\mathbb{Q}, <, +1)$~~	
~~discrete-time data $(\mathbb{Z}, <, +1)$~~	

homogeneous data domains



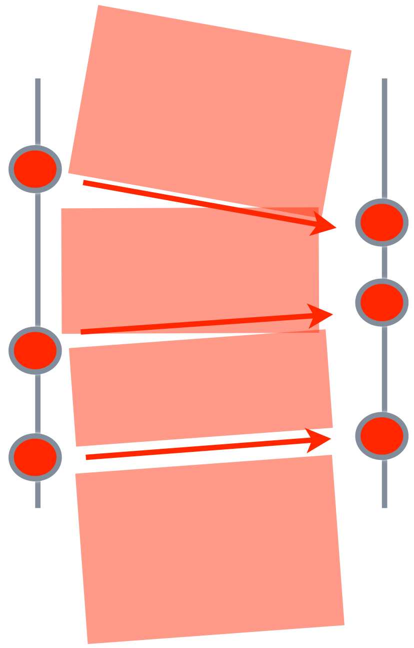
data domain $\mathbb{A}$	amalgamation class
equality data ($\mathbb{N}$, =)	finite pure sets
total order data ($\mathbb{Q}$, <)	finite total orders
~~dense-time data ($\mathbb{Q}$, <, +1)~~	
~~discrete-time data ($\mathbb{Z}$, <, +1)~~	
universal equivalence ($\mathbb{A}$, R, =)	finite equivalence relations

homogeneous data domains



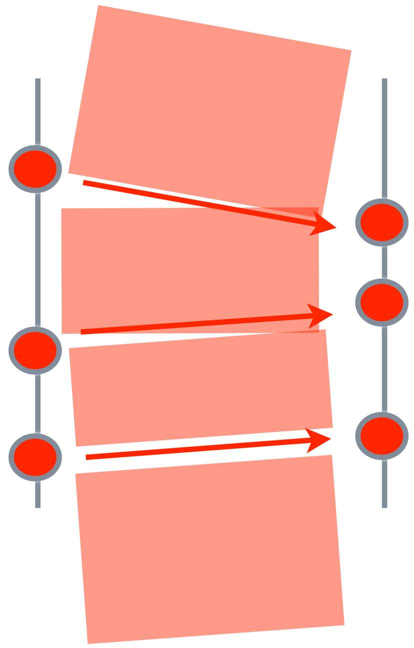
data domain $\mathbb{A}$	amalgamation class
equality data $(\mathbb{N}, =)$	finite pure sets
total order data $(\mathbb{Q}, <)$	finite total orders
~~dense-time data $(\mathbb{Q}, <, +1)$~~	
~~discrete-time data $(\mathbb{Z}, <, +1)$~~	
universal equivalence $(\mathbb{A}, R, =)$	finite equivalence relations
universal (random) graph	finite graphs

homogeneous data domains



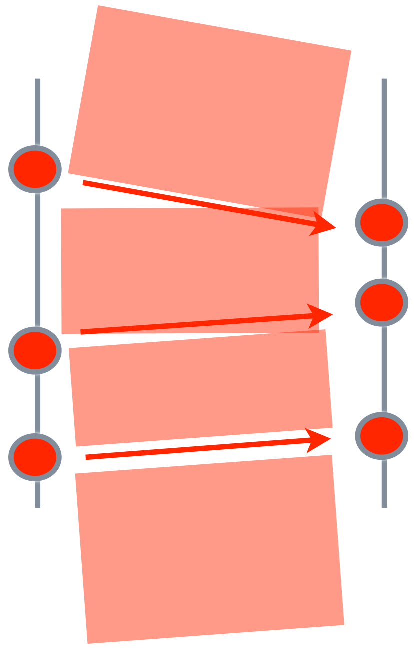
data domain $\mathbb{A}$	amalgamation class
equality data $(\mathbb{N}, =)$	finite pure sets
total order data $(\mathbb{Q}, <)$	finite total orders
~~dense-time data $(\mathbb{Q}, <, +1)$~~	
~~discrete-time data $(\mathbb{Z}, <, +1)$~~	
universal equivalence $(\mathbb{A}, R, =)$	finite equivalence relations
universal (random) graph	finite graphs
universal partial order	finite partial orders

homogeneous data domains



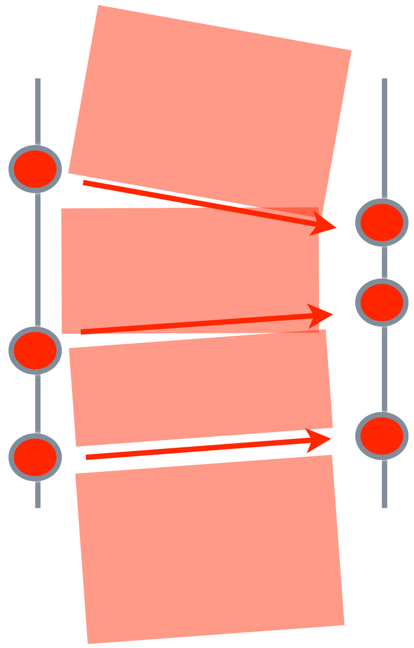
data domain $\mathbb{A}$	amalgamation class
equality data ($\mathbb{N}$, =)	finite pure sets
total order data ($\mathbb{Q}$, <)	finite total orders
~~dense-time data ($\mathbb{Q}$, <, +1)~~	
~~discrete-time data ($\mathbb{Z}$, <, +1)~~	
universal equivalence ($\mathbb{A}$, R, =)	finite equivalence relations
universal (random) graph	finite graphs
universal partial order	finite partial orders
universal directed graph	finite directed graphs

homogeneous data domains



data domain $\mathbb{A}$	amalgamation class
equality data $(\mathbb{N}, =)$	finite pure sets
total order data $(\mathbb{Q}, <)$	finite total orders
~~dense-time data $(\mathbb{Q}, <, +1)$~~	
~~discrete-time data $(\mathbb{Z}, <, +1)$~~	
universal equivalence $(\mathbb{A}, R, =)$	finite equivalence relations
universal (random) graph	finite graphs
universal partial order	finite partial orders
universal directed graph	finite directed graphs
universal tournament	finite tournaments

homogeneous data domains



data domain $\mathbb{A}$	amalgamation class
equality data $(\mathbb{N}, =)$	finite pure sets
total order data $(\mathbb{Q}, <)$	finite total orders
~~dense-time data $(\mathbb{Q}, <, +1)$~~	
~~discrete-time data $(\mathbb{Z}, <, +1)$~~	
universal equivalence $(\mathbb{A}, R, =)$	finite equivalence relations
universal (random) graph	finite graphs
universal partial order	finite partial orders
universal directed graph	finite directed graphs
universal tournament	finite tournaments
...	...

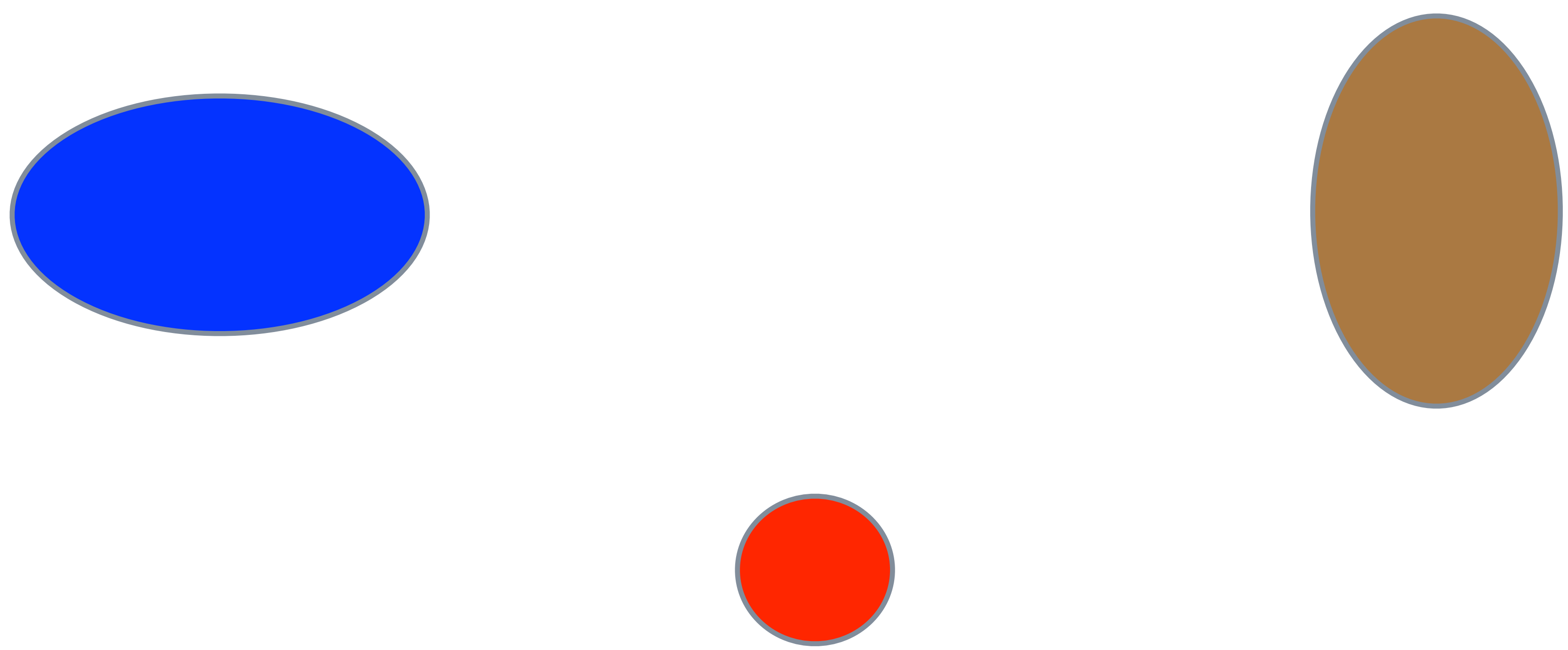
amalgamation

Def.: A embeds in B if A is isomorphic to an induced substructure of B.

amalgamation

Def.: A embeds in B if A is isomorphic to an induced substructure of B.

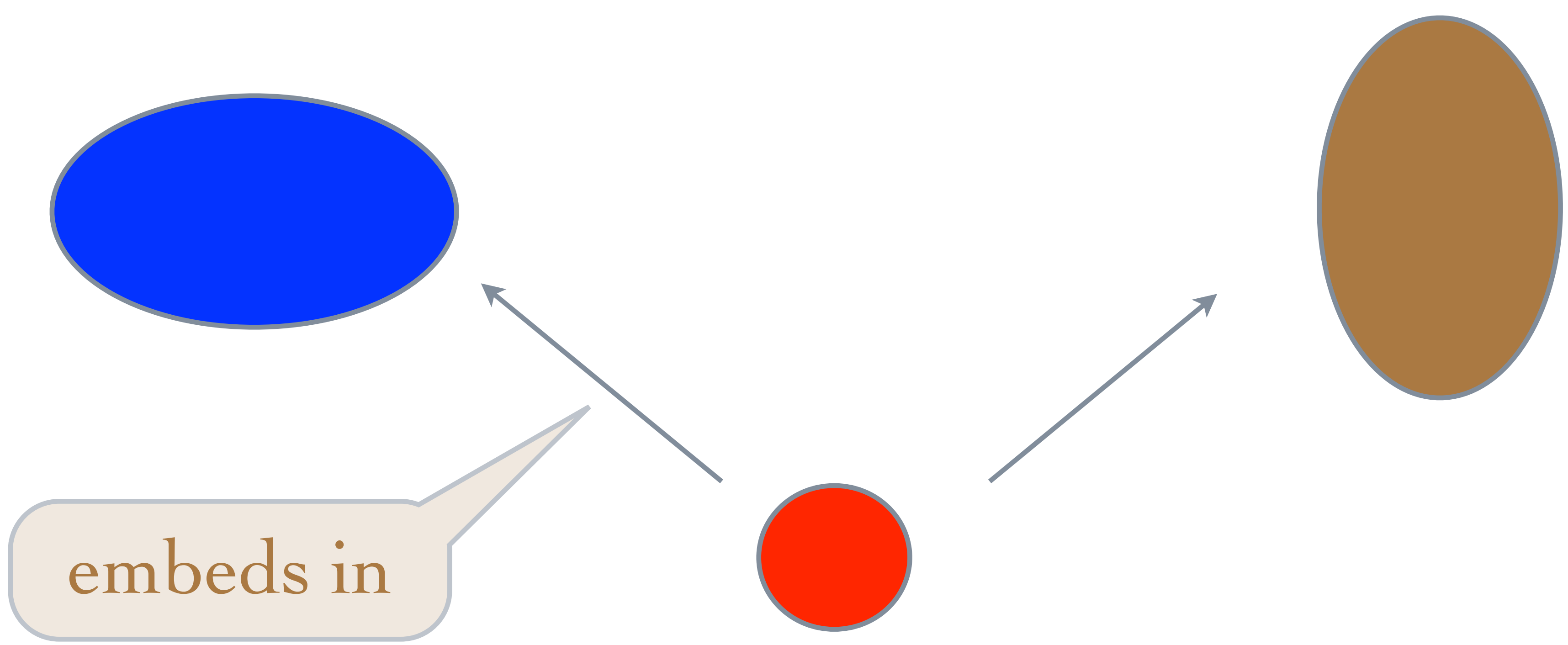
for every



amalgamation

Def.: A embeds in B if A is isomorphic to an induced substructure of B.

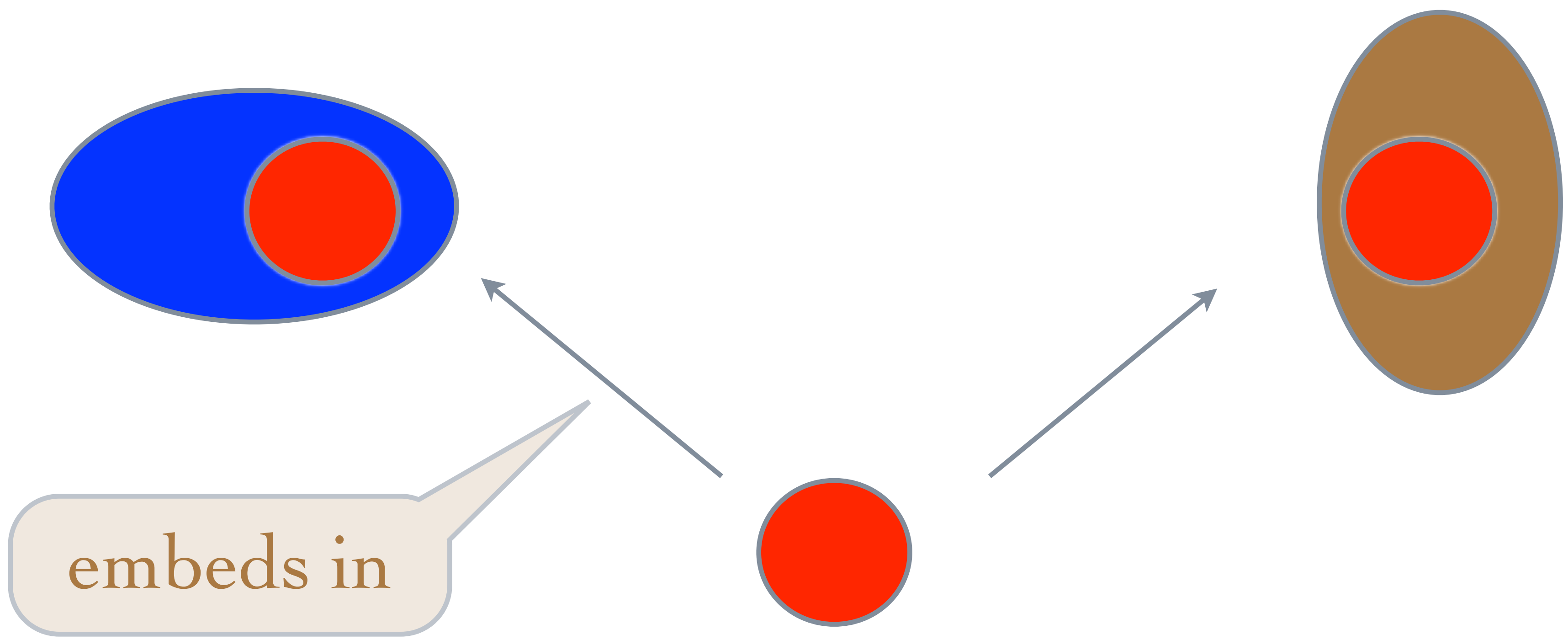
for every



amalgamation

Def.: A embeds in B if A is isomorphic to an induced substructure of B.

for every

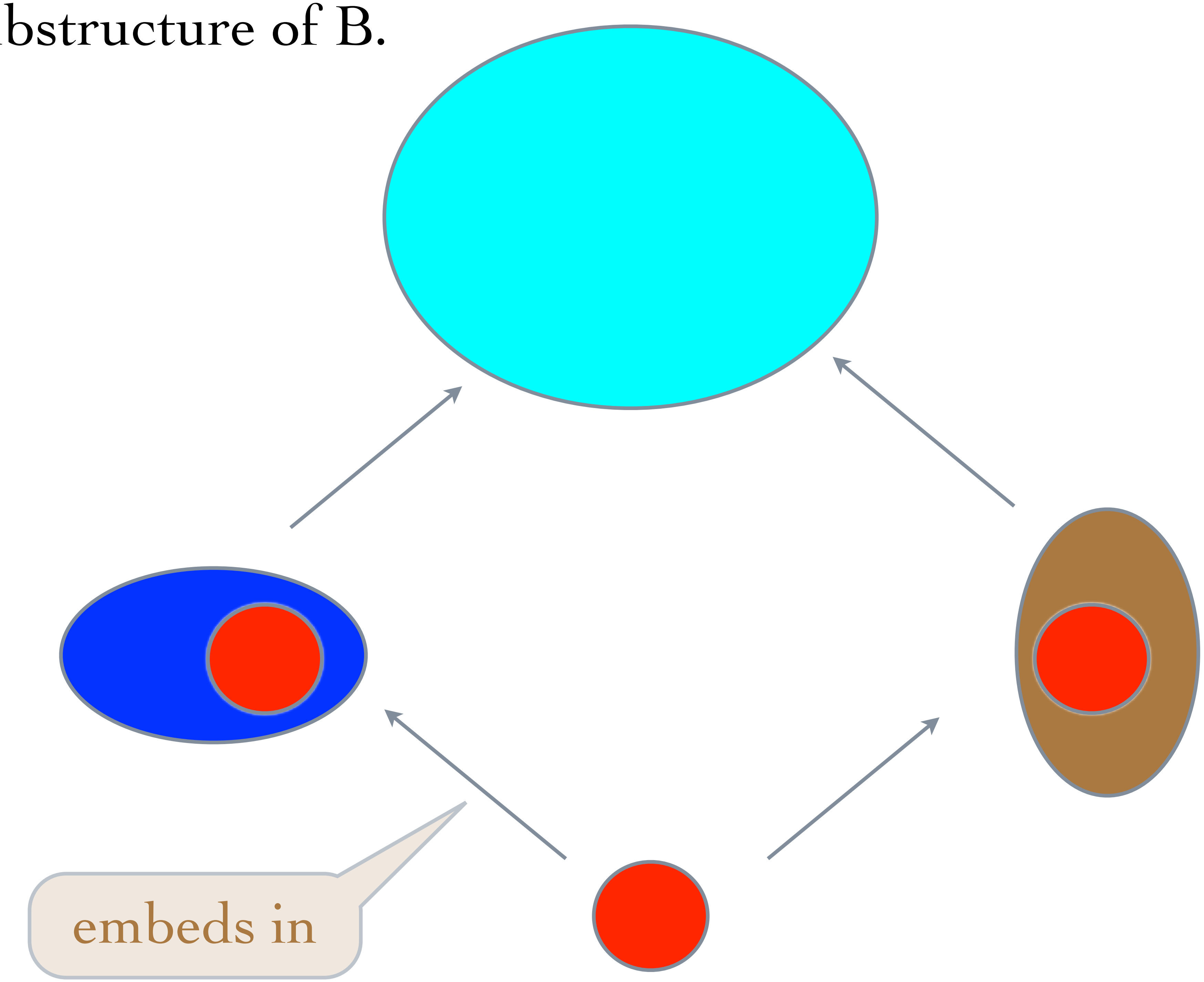


amalgamation

Def.: A embeds in B if A is isomorphic to an induced substructure of B.

there exists

for every

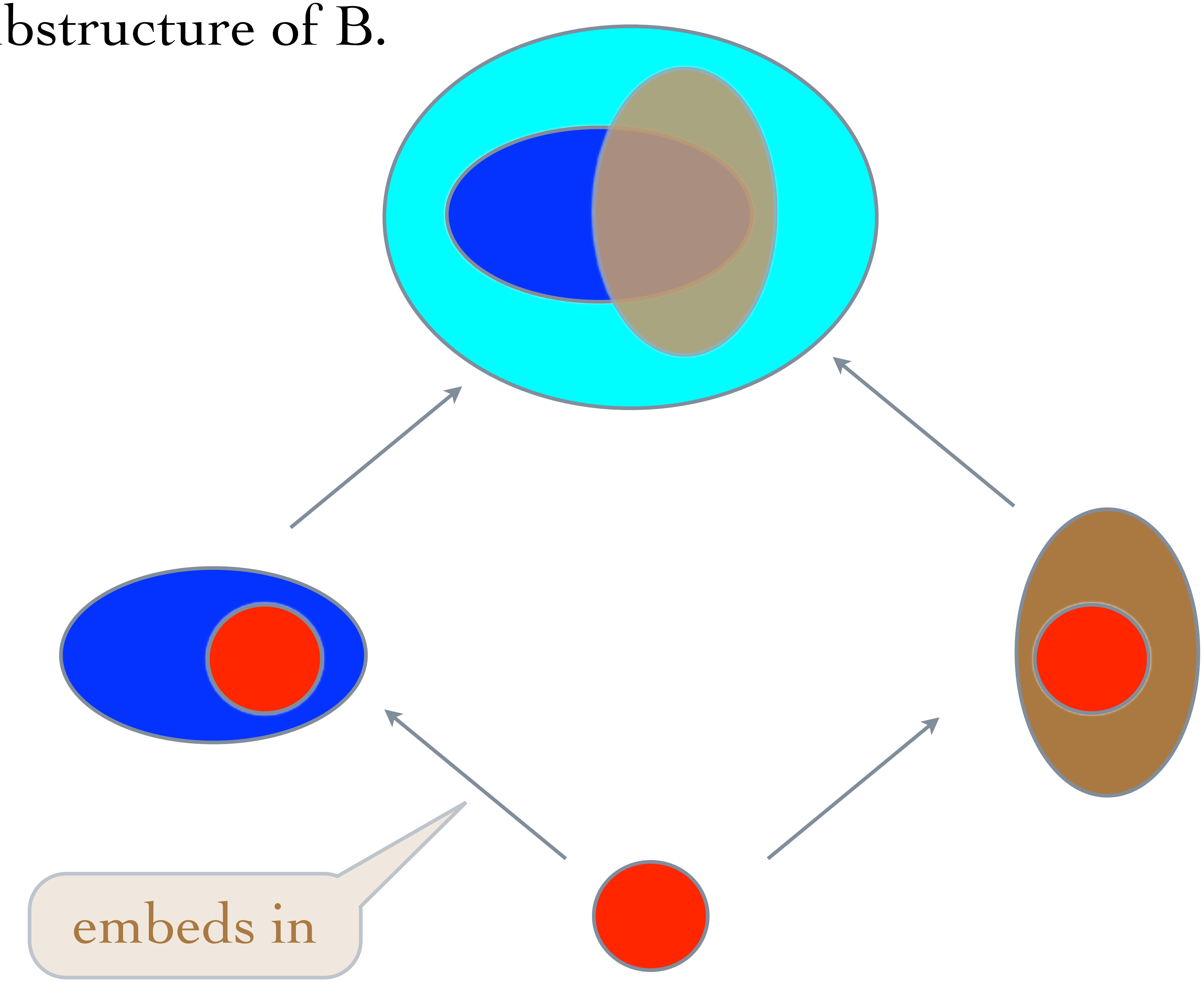


amalgamation

Def.: A embeds in B if A is isomorphic to an induced substructure of B.

there exists

for every

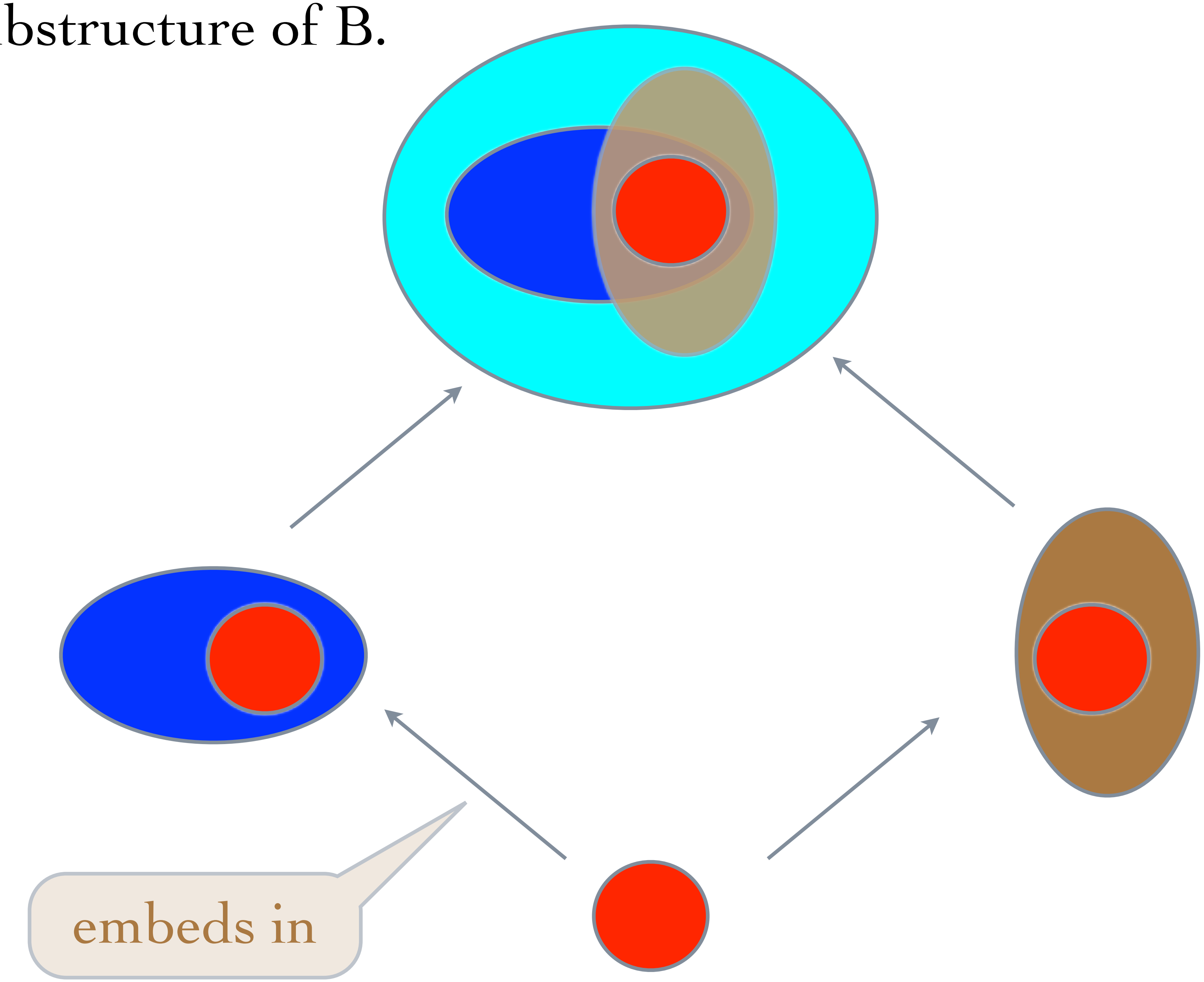


amalgamation

Def.: A embeds in B if A is isomorphic to an induced substructure of B.

there exists

for every

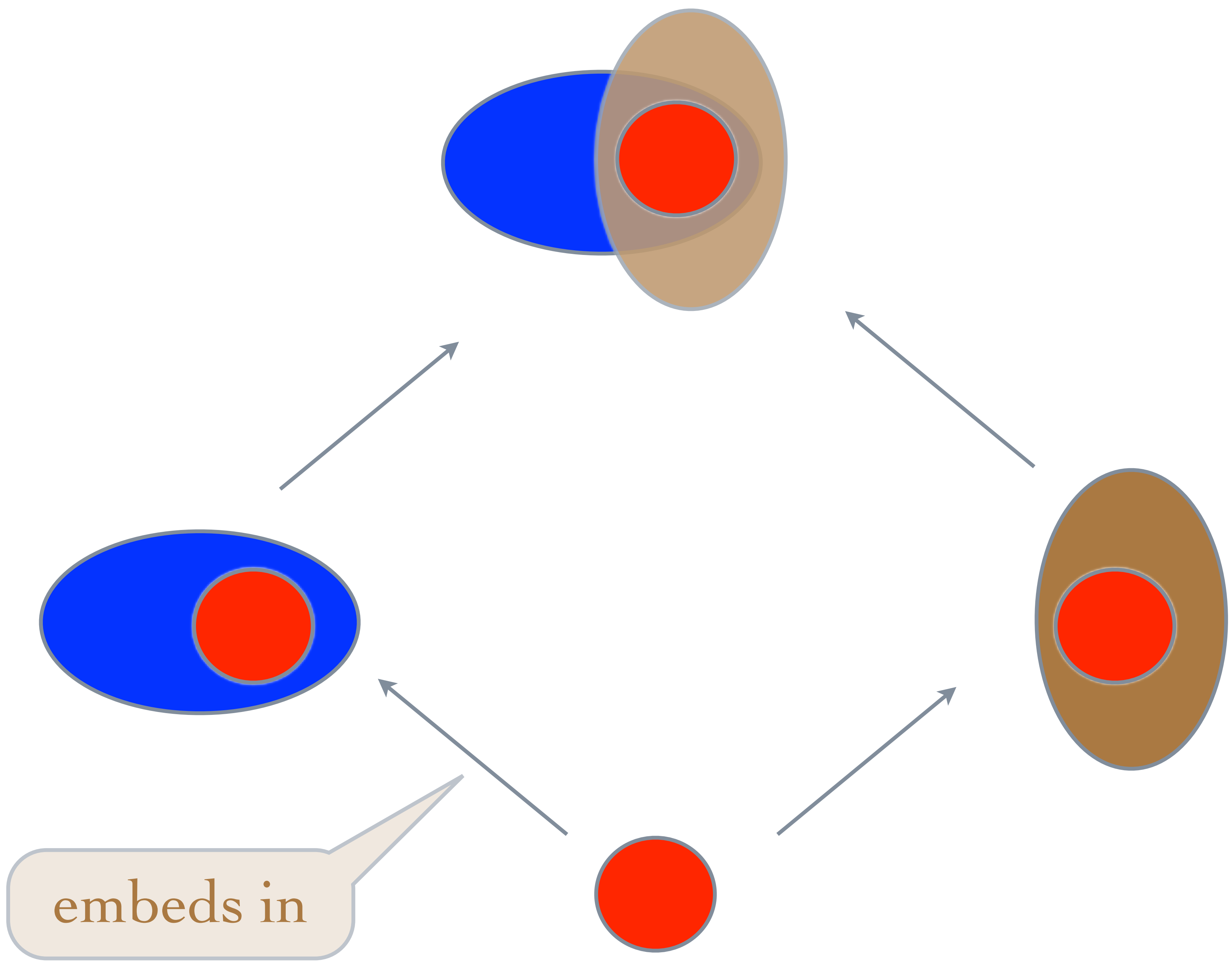


amalgamation

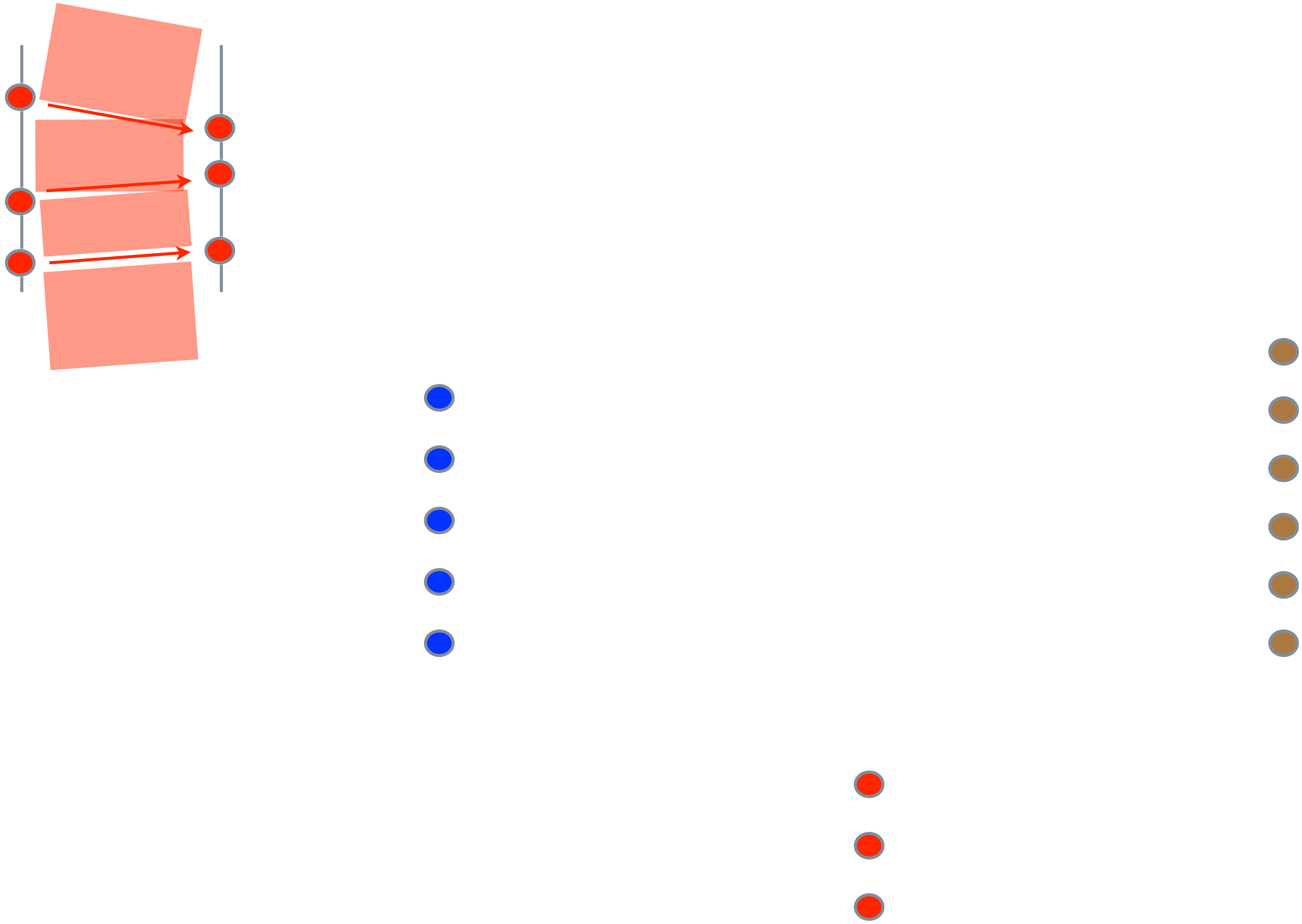
Def.: A embeds in B if A is isomorphic to an induced substructure of B.

there exists

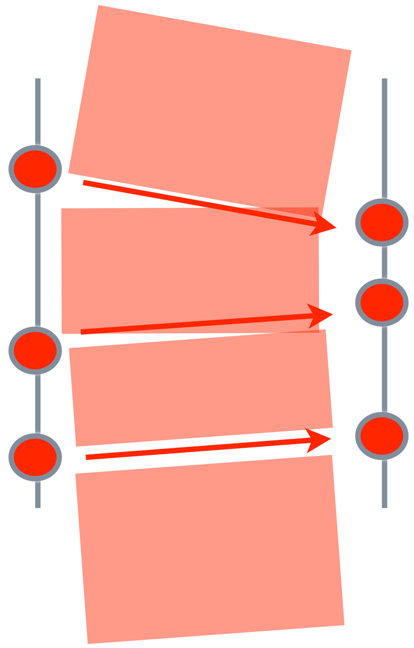
for every

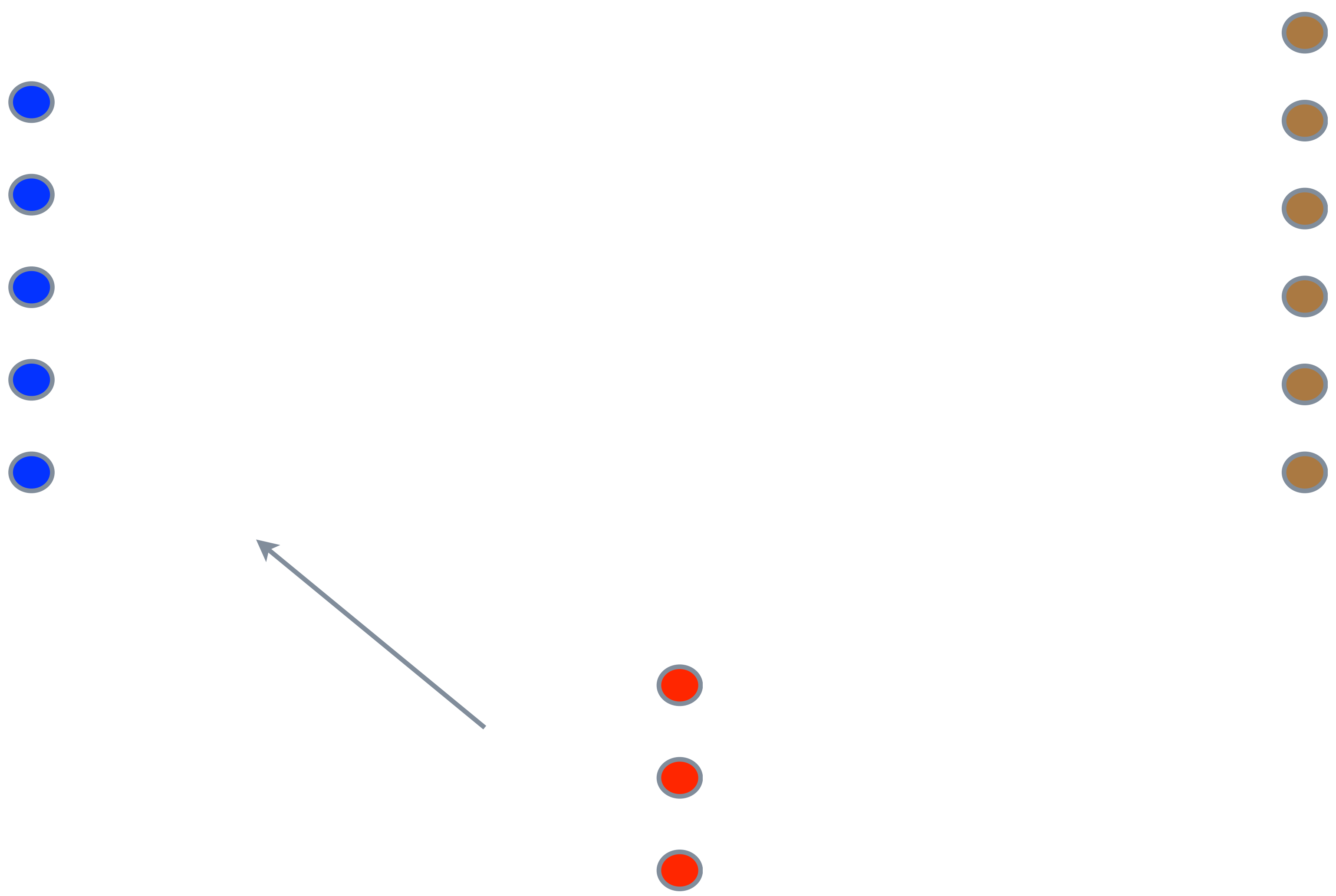


amalgamation example

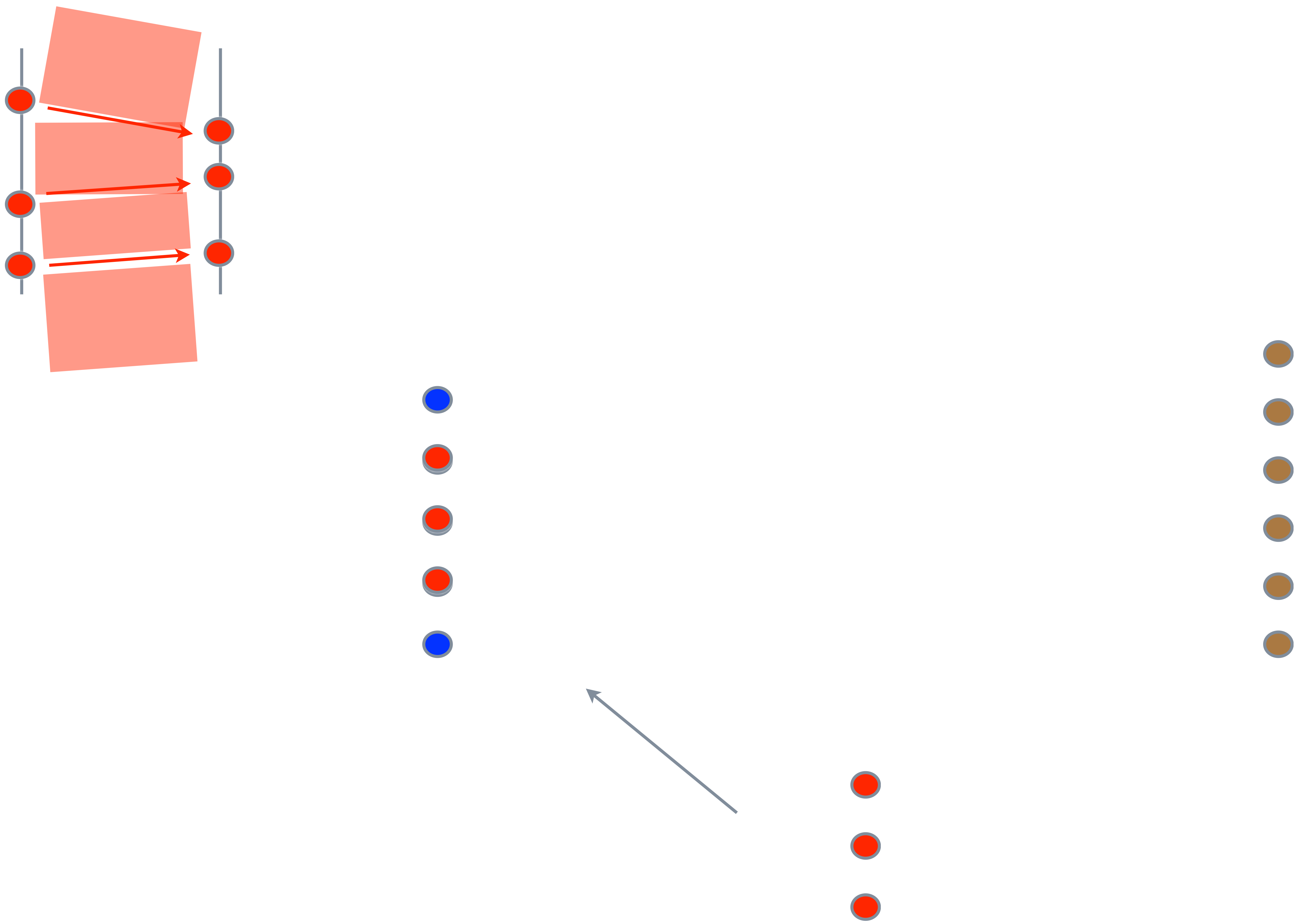


amalgamation example

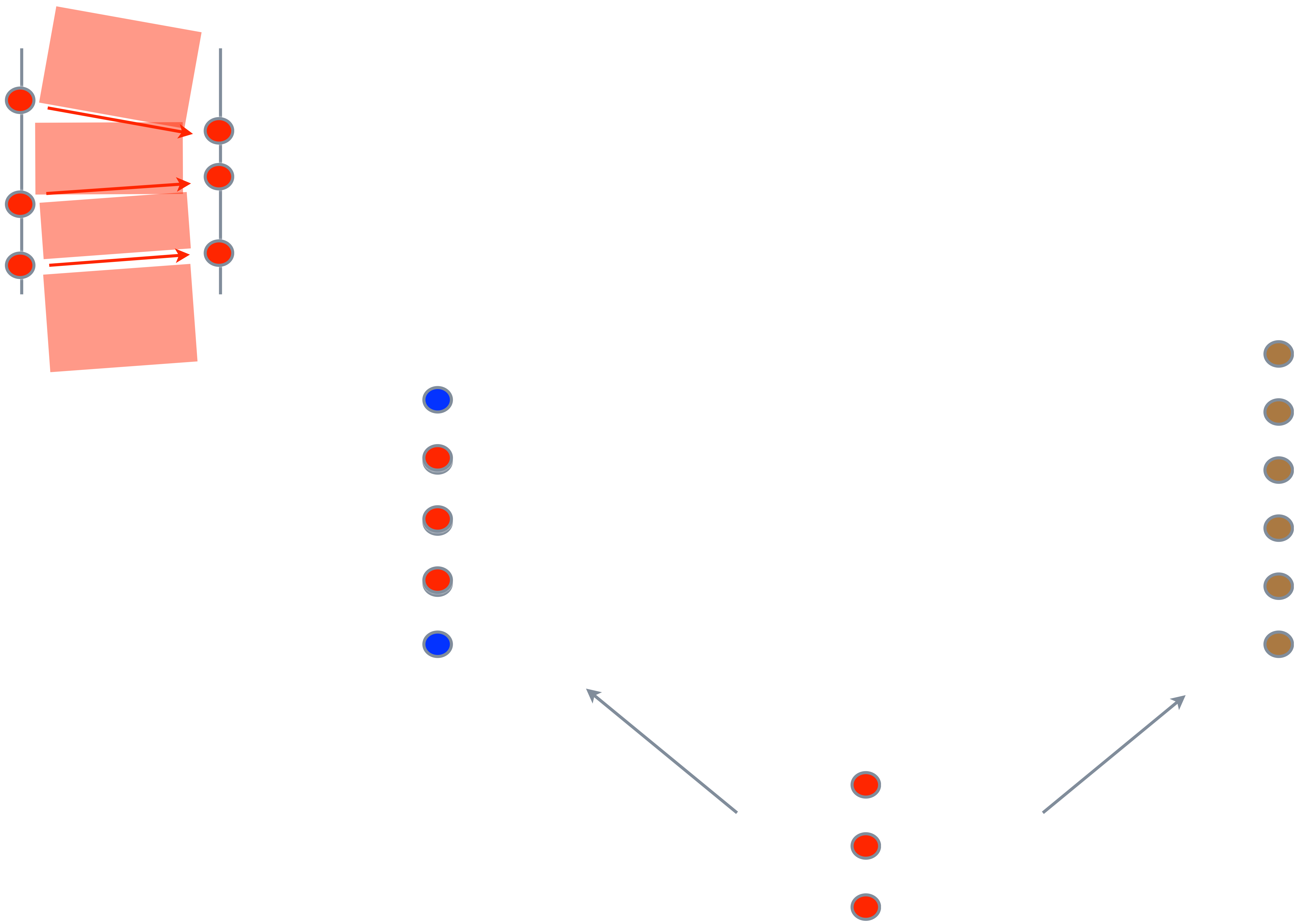




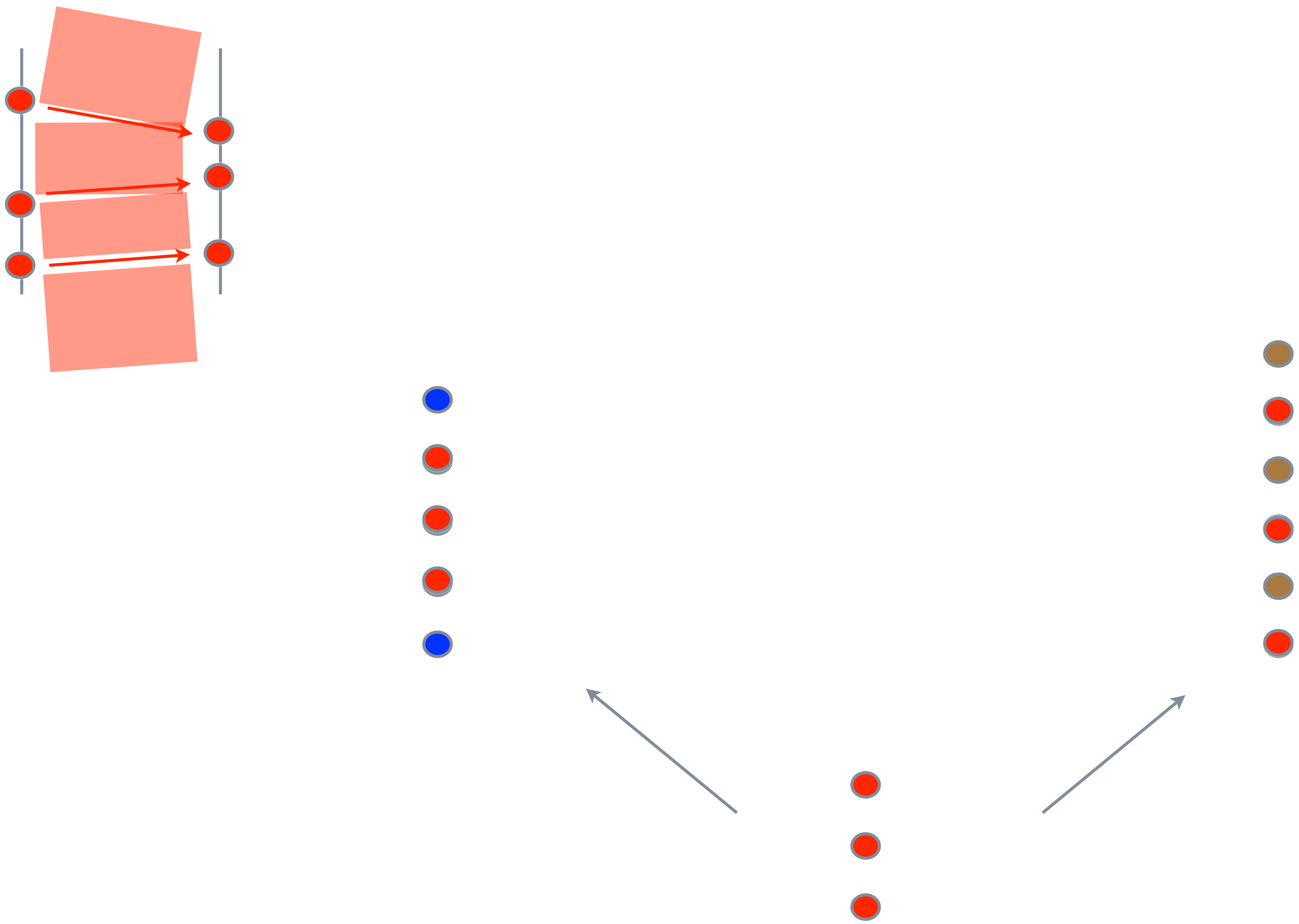
amalgamation example



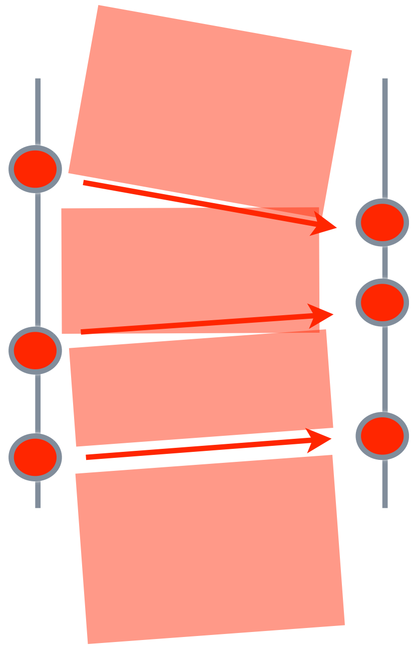
amalgamation example

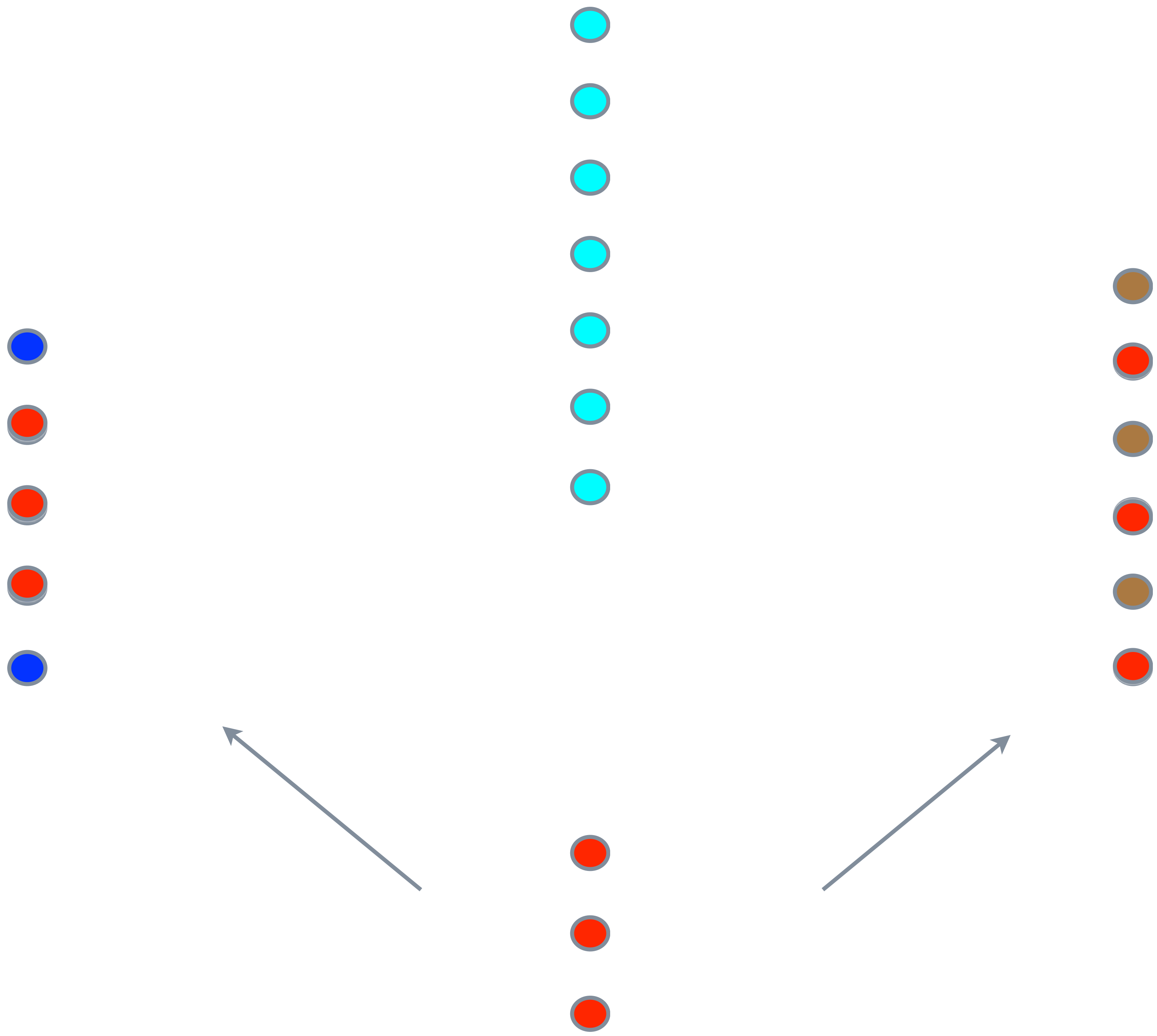


amalgamation example

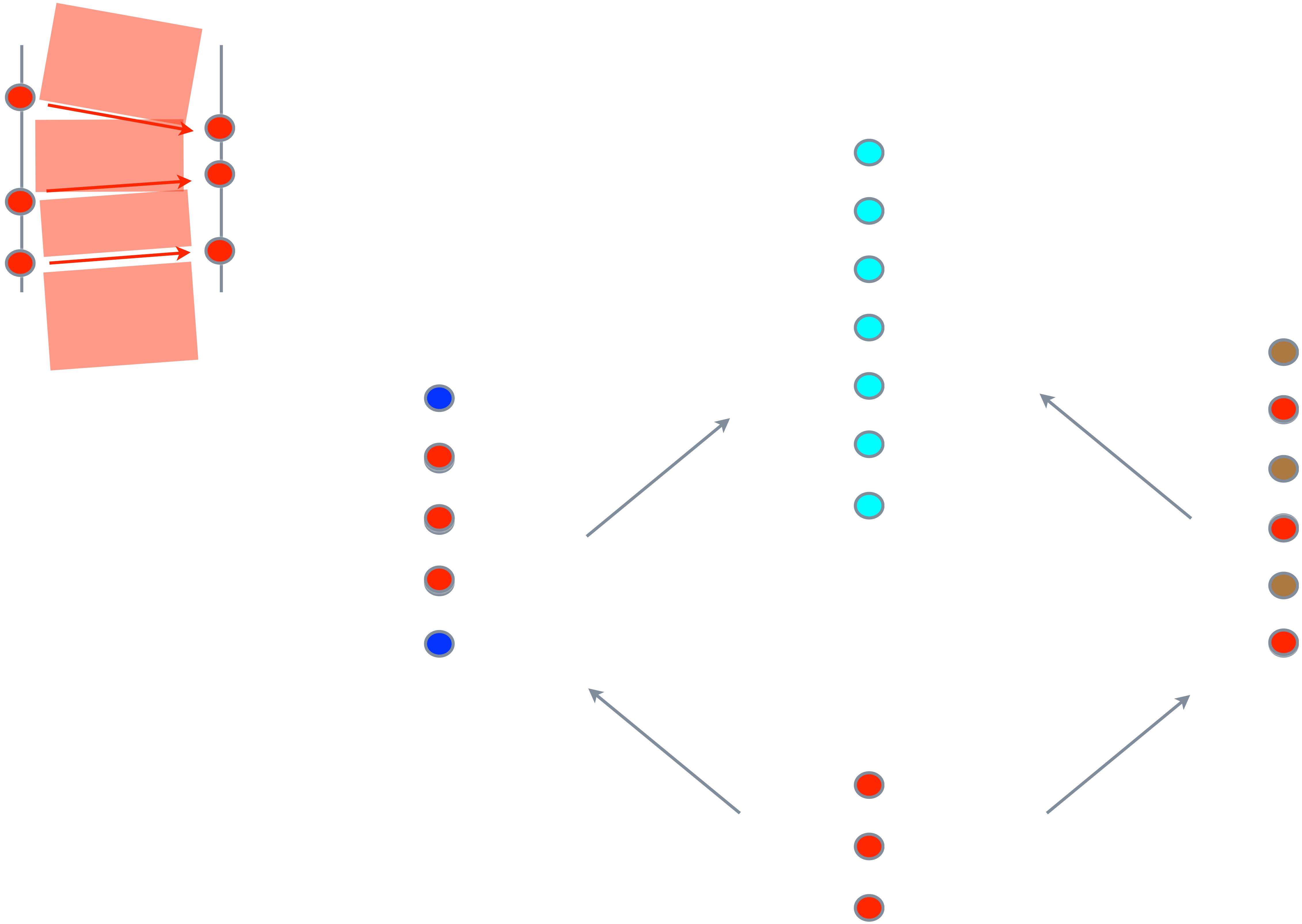


amalgamation example

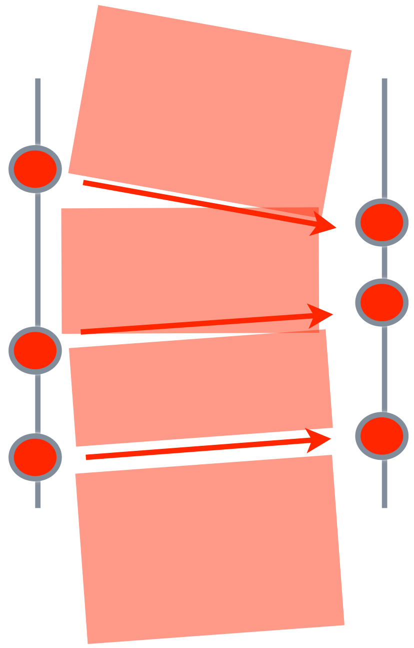


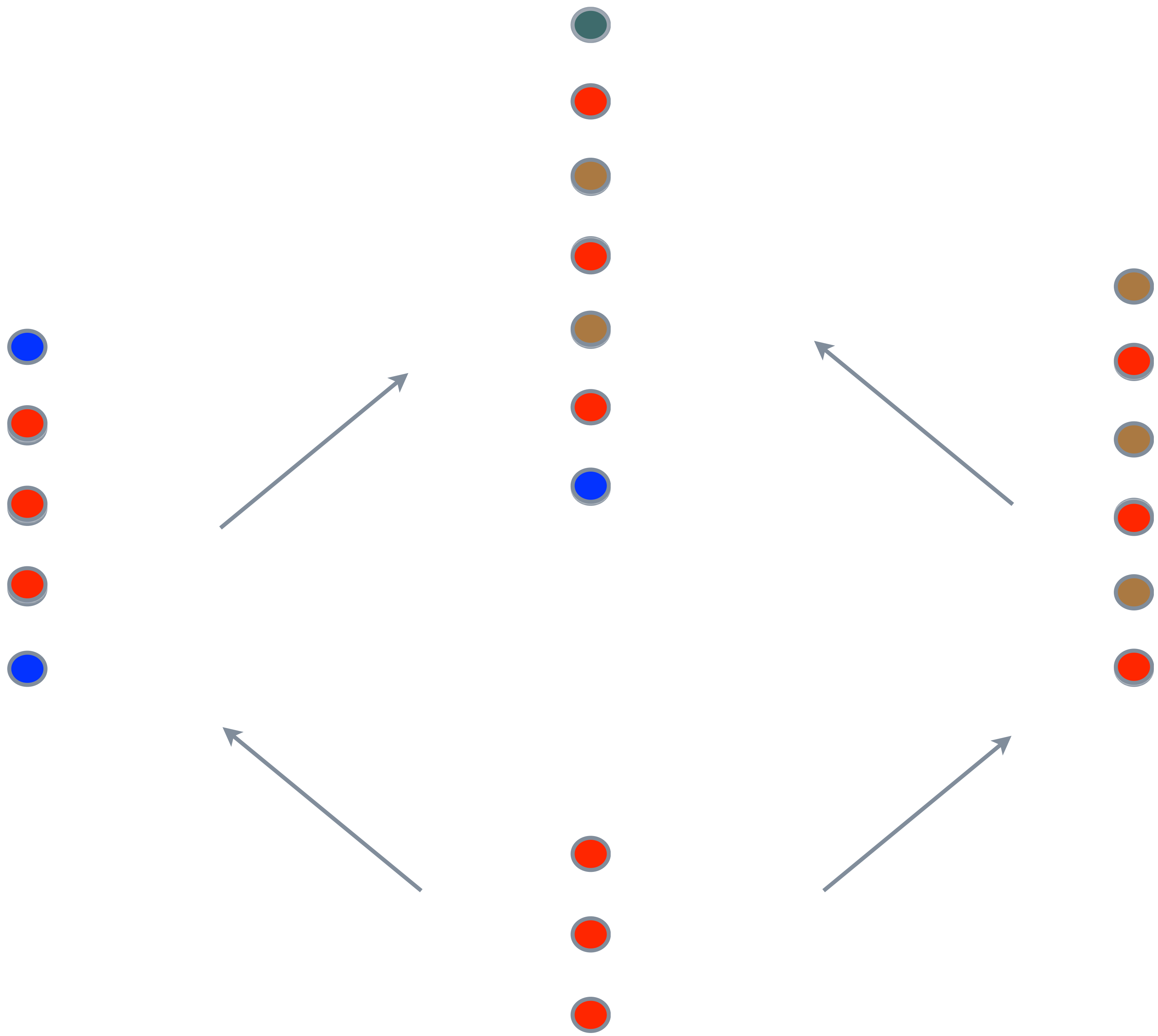


amalgamation example



amalgamation example





the question

For which data domains the standard problems are decidable?

the question

homogeneous

For which data domains the standard problems are decidable?

classifications

classifications

Theorem: [Lachlan,Woodrow'80] Let $\mathbb{A}$ be an infinite countable homogeneous graph. Then either $\mathbb{A}$ or its complement is isomorphic to one of:

classifications

Theorem: [Lachlan,Woodrow'80] Let $\mathbb{A}$ be an infinite countable homogeneous graph. Then either $\mathbb{A}$ or its complement is isomorphic to one of:

- universal (random) graph

classifications

Theorem: [Lachlan,Woodrow'80] Let $\mathbb{A}$ be an infinite countable homogeneous graph. Then either $\mathbb{A}$ or its complement is isomorphic to one of:

- universal (random) graph
- universal graph excluding n-clique, for some n

classifications

Theorem: [Lachlan,Woodrow'80] Let $\mathbb{A}$ be an infinite countable homogeneous graph. Then either $\mathbb{A}$ or its complement is isomorphic to one of:

- universal (random) graph
- universal graph excluding n-clique, for some n
- disjoint union of cliques of the same (finite or infinite) size

classifications

Theorem: [Lachlan,Woodrow'80] Let $\mathbb{A}$ be an infinite countable homogeneous graph. Then either $\mathbb{A}$ or its complement is isomorphic to one of:

- universal (random) graph
- universal graph excluding n-clique, for some n
- disjoint union of cliques of the same (finite or infinite) size

An analogous (but more complex) classification exists for directed graphs [Cherlin'98].

classifications

Theorem: [Lachlan,Woodrow'80] Let $\mathbb{A}$ be an infinite countable homogeneous graph. Then either $\mathbb{A}$ or its complement is isomorphic to one of:

- universal (random) graph
- universal graph excluding n-clique, for some n
- disjoint union of cliques of the same (finite or infinite) size

An analogous (but more complex) classification exists for directed graphs [Cherlin'98].

A classification of all homogeneous structures remains a great challenge.

Outline

- (un)ordered data Petri nets
- standard decision problems
- Petri nets with homogeneous data
- **undecidability**
- decidability via wqo

undecidability - example

undecidability - example

- data domain $(\mathbb{N}^2, =_1, =_2, =_{12})$

undecidability - example

- data domain $(\mathbb{N}^2, =_1, =_2, =_{12})$
- the proof adapts to $(\mathbb{N}^2, =_1, =_2)$

undecidability - proof idea

undecidability - proof idea

- data domain $(\mathbb{N}^2, =_1, =_2, =_{12})$

undecidability - proof idea

- data domain $(\mathbb{N}^2, =_1, =_2, =_{12})$
- simulate 2-counter machine

undecidability - proof idea

- data domain $(\mathbb{N}^2, =_1, =_2, =_{12})$
- simulate 2-counter machine
- counter value n is represented by a "cycle":

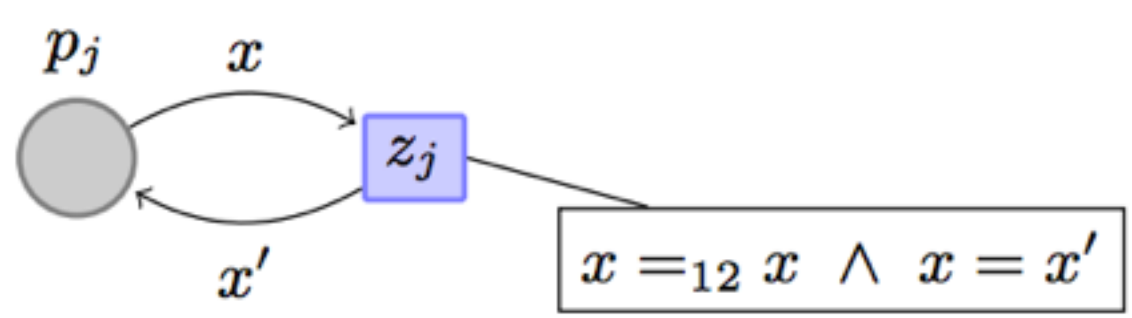
$$(a_1, a_2), (a_2, a_3), \ldots, (a_n, a_{n+1}), (a_{n+1}, a_1)$$

undecidability - proof idea

- data domain $(\mathbb{N}^2, =_1, =_2, =_{12})$
- simulate 2-counter machine
- counter value n is represented by a "cycle":

$$(a_1, a_2), (a_2, a_3), \ldots, (a_n, a_{n+1}), (a_{n+1}, a_1)$$

- zero test:

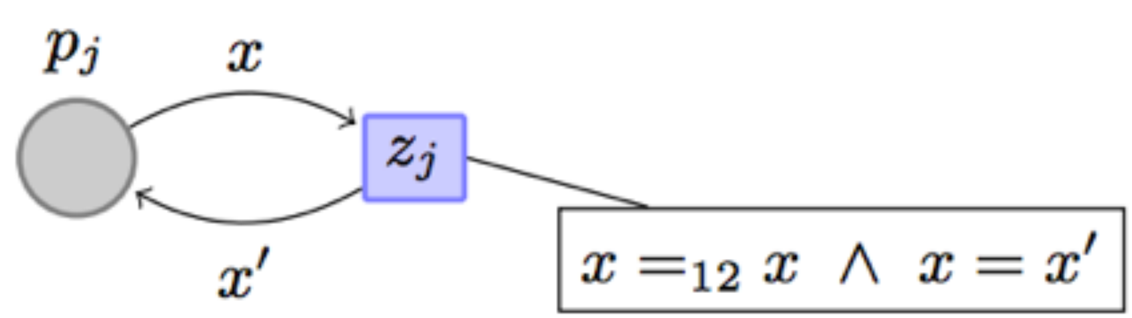


undecidability - proof idea

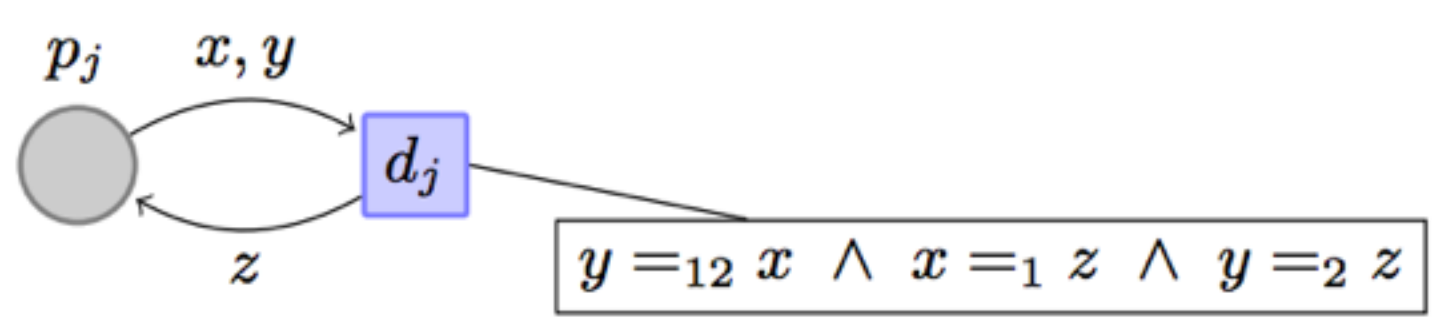
- data domain $(\mathbb{N}^2, =_1, =_2, =_{12})$
- simulate 2-counter machine
- counter value n is represented by a "cycle":

$$(a_1, a_2), (a_2, a_3), \ldots, (a_n, a_{n+1}), (a_{n+1}, a_1)$$

- zero test:

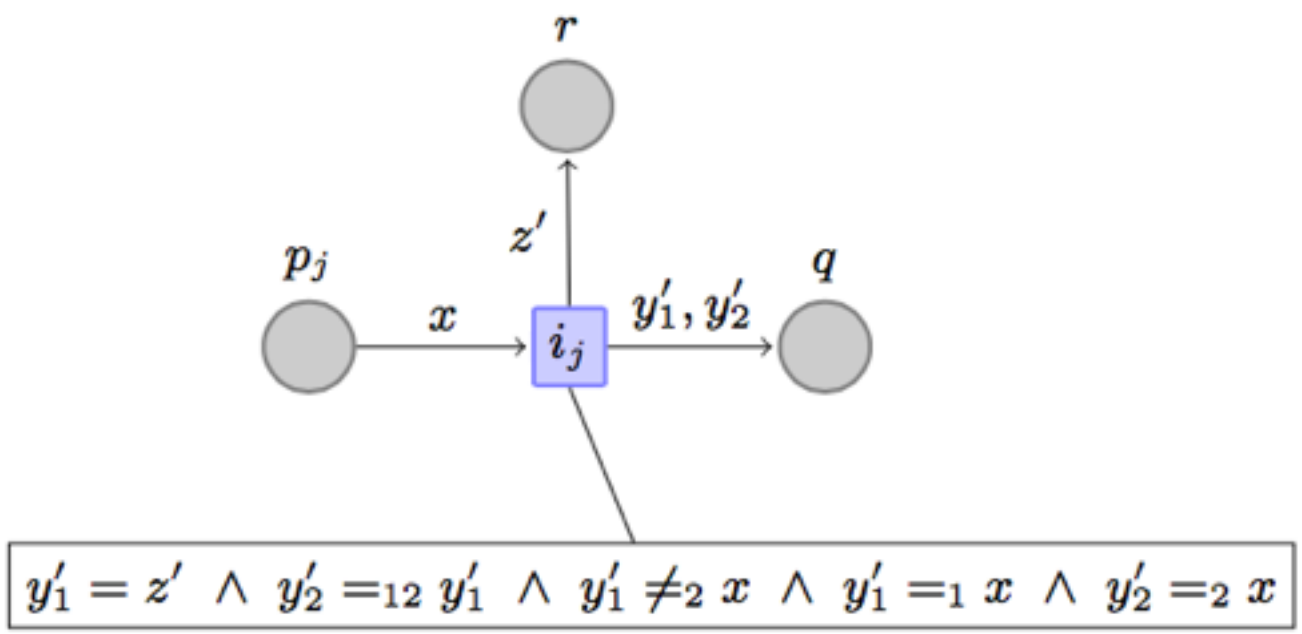


- decrement:



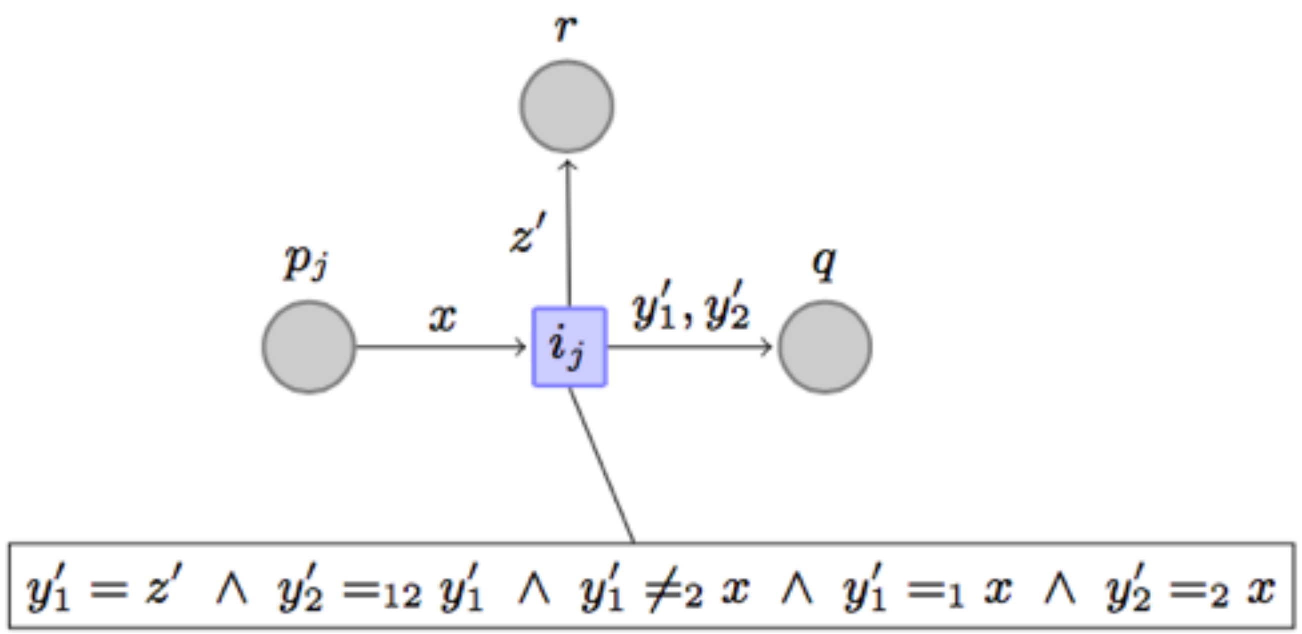
undecidability - proof idea

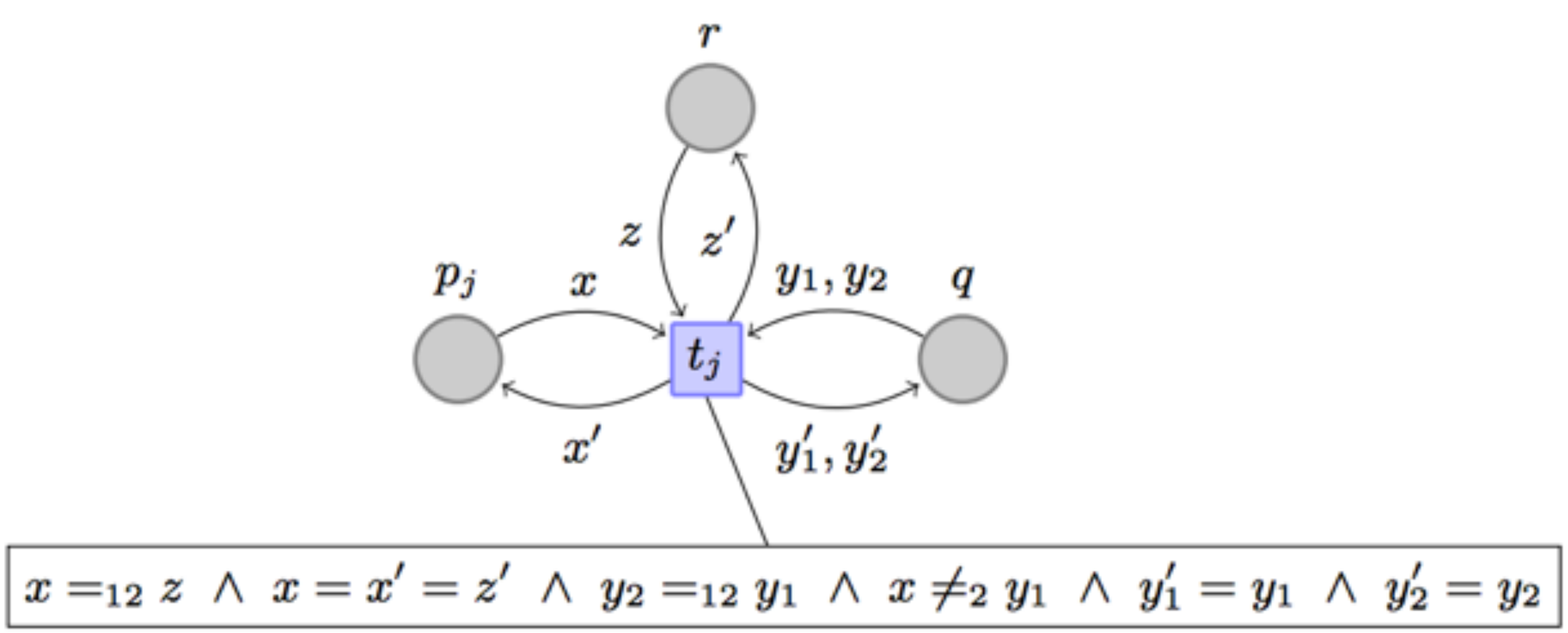
- increment:



undecidability - proof idea

- increment:





the question

For which homogeneous data domains the standard problems are decidable?

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

undecidable:

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

undecidable:

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

undecidable:

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

undecidable:

$(\mathbb{N}^2, =_1, =_2, =_{12})$

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

undecidable:

$(\mathbb{N}^2, =_1, =_2, =_{12})$

$(\mathbb{N}^2, =_1, =_2)$

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

$(\mathbb{N}^2, =_1, =)$ $\simeq$ universal equivalence relation

undecidable:

$(\mathbb{N}^2, =_1, =_2, =_{12})$

$(\mathbb{N}^2, =_1, =_2)$

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

$(\mathbb{N}^2, =_1, =) \simeq$ universal equivalence relation

$(\mathbb{N}^3, =_1, =_{1\wedge 2}, =)$

undecidable:

$(\mathbb{N}^2, =_1, =_2, =_{12})$

$(\mathbb{N}^2, =_1, =_2)$

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

$(\mathbb{N}^2, =_1, =)$ $\simeq$ universal equivalence relation

$(\mathbb{N}^3, =_1, =_{1\wedge 2}, =)$

undecidable:

$(\mathbb{N}^2, =_1, =_2, =_{12})$

$(\mathbb{N}^2, =_1, =_2)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =_2)$

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

$(\mathbb{N}^2, =_1, =)$ $\simeq$ universal equivalence relation

$(\mathbb{N}^3, =_1, =_{1\wedge 2}, =)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =)$

undecidable:

$(\mathbb{N}^2, =_1, =_2, =_{12})$

$(\mathbb{N}^2, =_1, =_2)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =_2)$

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

$(\mathbb{N}^2, =_1, =)$ $\simeq$ universal equivalence relation

$(\mathbb{N}^3, =_1, =_{1\wedge 2}, =)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =)$

undecidable:

$(\mathbb{N}^2, =_1, =_2, =_{12})$

$(\mathbb{N}^2, =_1, =_2)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =_2)$

universal (random) graph

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

$(\mathbb{N}^2, =_1, =)$ $\simeq$ universal equivalence relation

$(\mathbb{N}^3, =_1, =_{1\wedge 2}, =)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =)$

undecidable:

$(\mathbb{N}^2, =_1, =_2, =_{12})$

$(\mathbb{N}^2, =_1, =_2)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =_2)$

universal (random) graph

universal partial order

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

$(\mathbb{N}^2, =_1, =)$ $\simeq$ universal equivalence relation

$(\mathbb{N}^3, =_1, =_{1\wedge 2}, =)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =)$

undecidable:

$(\mathbb{N}^2, =_1, =_2, =_{12})$

$(\mathbb{N}^2, =_1, =_2)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =_2)$

universal (random) graph

universal partial order

universal tournament

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

$(\mathbb{N}^2, =_1, =)$ $\simeq$ universal equivalence relation

$(\mathbb{N}^3, =_1, =_{1\wedge 2}, =)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =)$

undecidable:

$(\mathbb{N}^2, =_1, =_2, =_{12})$

$(\mathbb{N}^2, =_1, =_2)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =_2)$

universal (random) graph

universal partial order

universal tournament

...

the question

For which homogeneous data domains the standard problems are decidable?

decidable:

$(\mathbb{N}, =)$

$(\mathbb{Q}, <)$

$(\mathbb{N}^2, =_1, =)$ $\simeq$ universal equivalence relation

$(\mathbb{N}^3, =_1, =_{1\wedge 2}, =)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =)$

...

undecidable:

$(\mathbb{N}^2, =_1, =_2, =_{12})$

$(\mathbb{N}^2, =_1, =_2)$

$(\mathbb{Q} \times \mathbb{N}, <_1, =_2)$

universal (random) graph

universal partial order

universal tournament

...

undecidability - further examples

undecidability - further examples

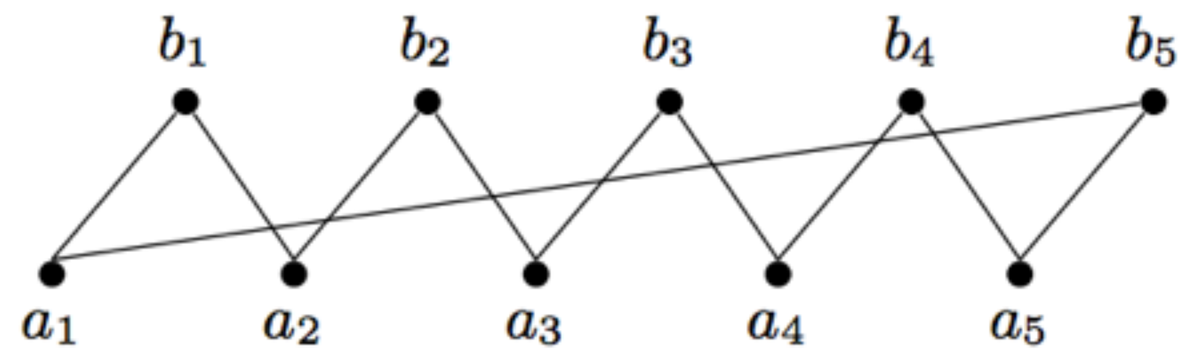
- universal (random) graph: cycle

undecidability - further examples

- universal (random) graph: cycle
- universal partial order:

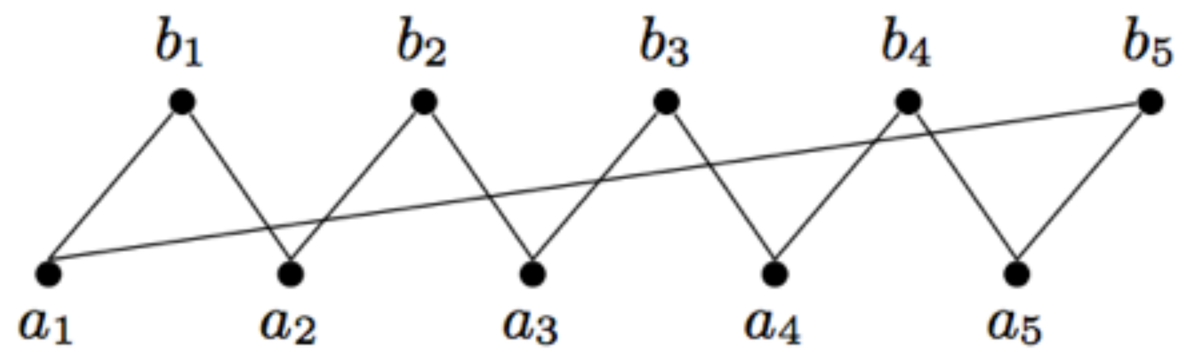
undecidability - further examples

- universal (random) graph: cycle
- universal partial order:



undecidability - further examples

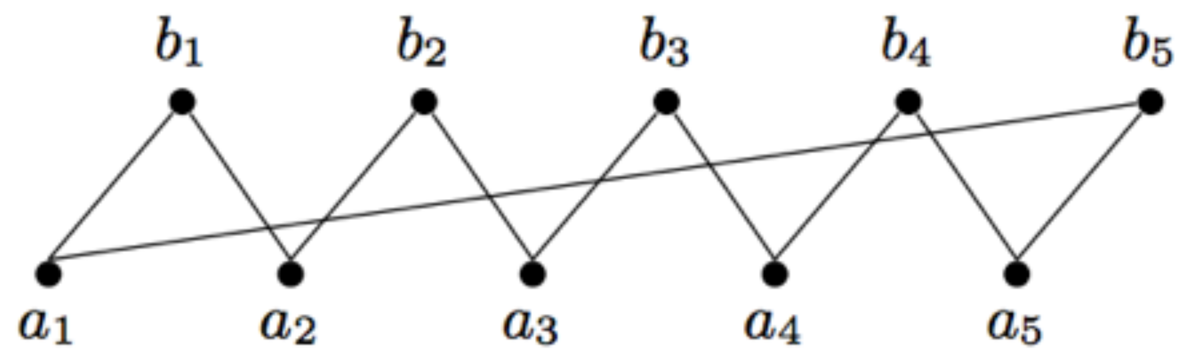
- universal (random) graph: cycle
- universal partial order:



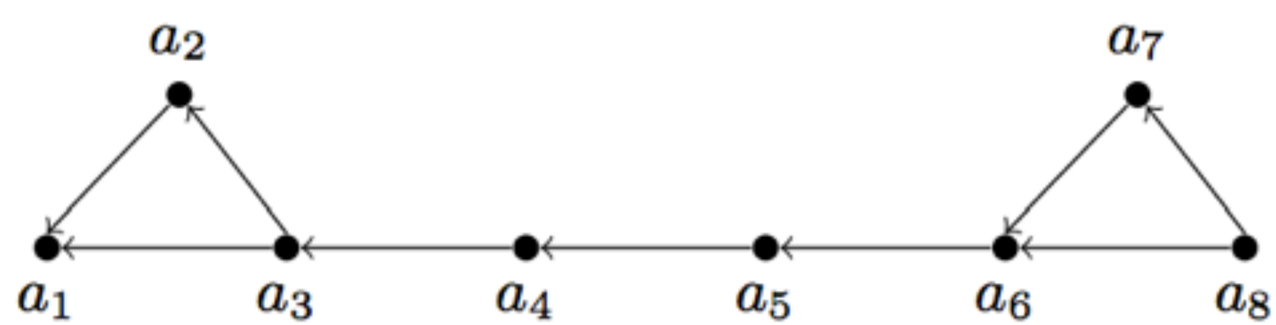
- universal tournament:

undecidability - further examples

- universal (random) graph: cycle
- universal partial order:

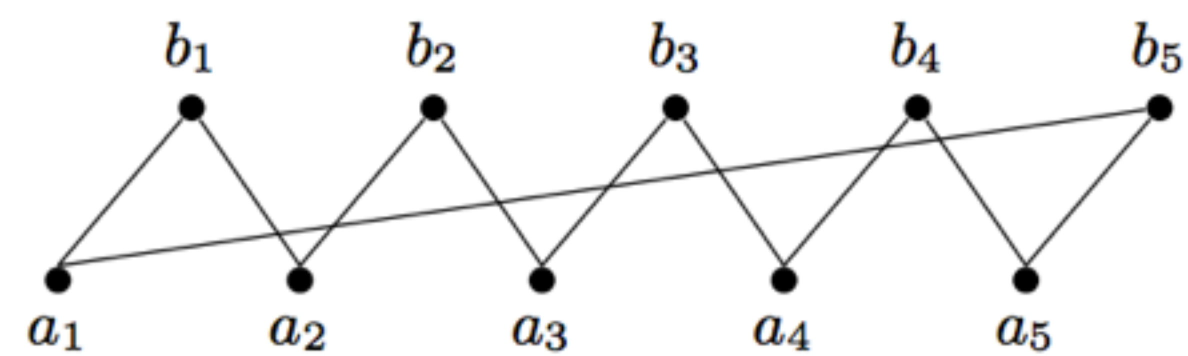


- universal tournament:

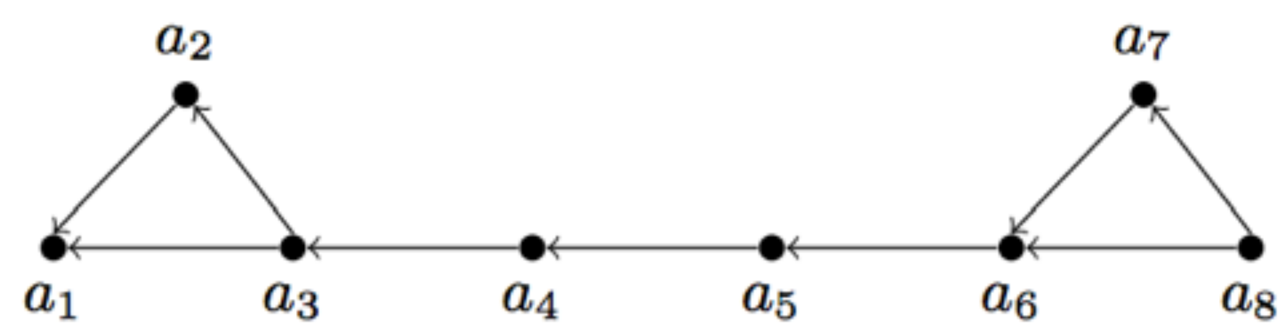


undecidability - further examples

- universal (random) graph: cycle
- universal partial order:



- universal tournament:



- ...

Outline

- (un)ordered data Petri nets
- standard decision problems
- Petri nets with homogeneous data
- undecidability
- **decidability via wqo**

well quasi-orders (wqos)

Definition: A quasi-order is a wqo if every infinite sequence

$$a_1, a_2, a_3, \ldots$$

contains a dominating pair, that is $a_i \leq a_j$, for some $i < j$.

well quasi-orders (wqos)

Definition: A quasi-order is a wqo if every infinite sequence

$$a_1, a_2, a_3, \ldots$$

contains a dominating pair, that is $a_i \leq a_j$, for some $i < j$.

Examples:

well quasi-orders (wqos)

Definition: A quasi-order is a wqo if every infinite sequence

$$a_1, a_2, a_3, \ldots$$

contains a dominating pair, that is $a_i \leq a_j$, for some $i < j$.

Examples:

$M(\mathrm{P})$ ordered by multiset inclusion (Dickson's lemma)

well quasi-orders (wqos)

Definition: A quasi-order is a wqo if every infinite sequence

$$a_1, a_2, a_3, \ldots$$

contains a dominating pair, that is $a_i \leq a_j$, for some $i < j$.

Examples:

$M(\mathrm{P})$ ordered by multiset inclusion (Dickson's lemma)

P^* ordered by subsequence ordering (Higman's lemma)

well quasi-orders (wqos)

Definition: A quasi-order is a wqo if every infinite sequence

$$a_1, a_2, a_3, \ldots$$

contains a dominating pair, that is $a_i \leq a_j$, for some $i < j$.

Examples:

$M(\mathrm{P})$ ordered by multiset inclusion (Dickson's lemma)

P^* ordered by subsequence ordering (Higman's lemma)

$M(M(\mathrm{P}))$

well quasi-orders (wqos)

Definition: A quasi-order is a wqo if every infinite sequence

$$a_1, a_2, a_3, \ldots$$

contains a dominating pair, that is $a_i \leq a_j$, for some $i < j$.

Examples:

$M(\mathrm{P})$ ordered by multiset inclusion (Dickson's lemma)

P^* ordered by subsequence ordering (Higman's lemma)

$M(M(\mathrm{P}))$

$M(\mathrm{P})^*$

well quasi-orders (wqos)

Definition: A quasi-order is a wqo if every infinite sequence

$$a_1, a_2, a_3, \ldots$$

contains a dominating pair, that is $a_i \leq a_j$, for some $i < j$.

Examples:

$M(\mathrm{P})$ ordered by multiset inclusion (Dickson's lemma)

P^* ordered by subsequence ordering (Higman's lemma)

$M(M(\mathrm{P}))$

$M(\mathrm{P})^*$

...

decidability

decidability

Consider age($\mathbb{A}$) ordered by embeddings.

decidability

Consider age($\mathbb{A}$) ordered by embeddings.

Recall that a configuration is a structure from age($\mathbb{A}$) labeled by elements of $M(\mathrm{P})$

decidability

Consider age($\mathbb{A}$) ordered by embeddings.
Recall that a configuration is a structure from age($\mathbb{A}$) labeled by elements of $M(\mathrm{P})$

Examples:

decidability

Consider age($\mathbb{A}$) ordered by embeddings.

Recall that a configuration is a structure from age($\mathbb{A}$) labeled by elements of $M(\mathrm{P})$

Examples:

$\mathbb{A} = (\mathbb{N}, =)$ configurations = $M(M(\mathrm{P}))$

decidability

Consider age($\mathbb{A}$) ordered by embeddings.

Recall that a configuration is a structure from age($\mathbb{A}$) labeled by elements of $M(\mathrm{P})$

Examples:

$\mathbb{A}$ = $(\mathbb{N}, =)$ configurations = $M(M(\mathrm{P}))$

$\mathbb{A}$ = $(\mathbb{Q}, <)$ configurations = $M(\mathrm{P})^*$

decidability

Consider age($\mathbb{A}$) ordered by embeddings.

Recall that a configuration is a structure from age($\mathbb{A}$) labeled by elements of $M(\mathrm{P})$

Examples:

$\mathbb{A}$ = $(\mathbb{N}, =)$	configurations = $M(M(\mathrm{P}))$	wqo
$\mathbb{A}$ = $(\mathbb{Q}, <)$	configurations = $M(\mathrm{P})^*$	

decidability

Consider age($\mathbb{A}$) ordered by embeddings.

Recall that a configuration is a structure from age($\mathbb{A}$) labeled by elements of $M(\mathrm{P})$

Examples:

$\mathbb{A}$ = $(\mathbb{N}, =)$	configurations = $M(M(\mathrm{P}))$	wqo
$\mathbb{A}$ = $(\mathbb{Q}, <)$	configurations = $M(\mathrm{P})^*$	wqo
$\mathbb{A}$ = random graph	configurations = finite graphs labeled by elements of $M(\mathrm{P})$	

decidability

Consider age($\mathbb{A}$) ordered by embeddings.
Recall that a configuration is a structure from age($\mathbb{A}$) labeled by elements of $M(\mathrm{P})$

Examples:

$\mathbb{A}$ = $(\mathbb{N}, =)$	configurations = $M(M(\mathrm{P}))$	wqo
$\mathbb{A}$ = $(\mathbb{Q}, <)$	configurations = $M(\mathrm{P})^*$	wqo
$\mathbb{A}$ = random graph	configurations = finite graphs labeled by elements of $M(\mathrm{P})$	no wqo!

decidability

Consider age($\mathbb{A}$) ordered by embeddings.
Recall that a configuration is a structure from age($\mathbb{A}$) labeled by elements of $M(\mathrm{P})$

Examples:

$\mathbb{A}$ = $(\mathbb{N}, =)$ configurations = $M(M(\mathrm{P}))$

$\mathbb{A}$ = $(\mathbb{Q}, <)$ configurations = $M(\mathrm{P})^*$

} wqo

$\mathbb{A}$ = random graph configurations = finite graphs labeled by elements of $M(\mathrm{P})$ no wqo!

not to be confused with Robertson-Seymour's Graph Minor Theorem!

decidability

Consider age($\mathbb{A}$) ordered by embeddings.
Recall that a configuration is a structure from age($\mathbb{A}$) labeled by elements of $M(\mathrm{P})$

Examples:

$\mathbb{A}$ = $(\mathbb{N}, =)$	configurations = $M(M(\mathrm{P}))$	wqo
$\mathbb{A}$ = $(\mathbb{Q}, <)$	configurations = $M(\mathrm{P})^*$	wqo
$\mathbb{A}$ = random graph	configurations = finite graphs labeled by elements of $M(\mathrm{P})$	no wqo!

Theorem:

Let $\mathbb{A}$ be an effective homogeneous data domain such that configurations, ordered by embeddings, are a wqo.
Then all the standard problems are decidable.

decidability

Consider age($\mathbb{A}$) ordered by embeddings.
Recall that a configuration is a structure from age($\mathbb{A}$) labeled by elements of $M(\mathrm{P})$

Examples:

$\mathbb{A} = (\mathbb{N}, =)$	configurations = $M(M(\mathrm{P}))$	wqo
$\mathbb{A} = (\mathbb{Q}, <)$	configurations = $M(\mathrm{P})^*$	wqo
$\mathbb{A}$ = random graph	configurations = finite graphs labeled by elements of $M(\mathrm{P})$	no wqo!

Theorem:

Let $\mathbb{A}$ be an effective homogeneous data domain such that configurations, ordered by embeddings, are a wqo.
Then all the standard problems are decidable.

Proof: Using the framework of well-structured transition systems of [Finkel,Schnoebelen'01].

WQO Dichotomy Conjecture:

For a homogeneous data domain $\mathbb{A}$, exactly one of the following conditions holds:

- age($\mathbb{A}$), ordered by embeddings, is a wqo
- all the standard problems are undecidable.

WQO Dichotomy Conjecture:

For a homogeneous data domain $\mathbb{A}$, exactly one of the following conditions holds:

- age($\mathbb{A}$), ordered by embeddings, is a wqo
- all the standard problems are undecidable.

The conjecture is confirmed for:

WQO Dichotomy Conjecture:

For a homogeneous data domain $\mathbb{A}$, exactly one of the following conditions holds:

- age($\mathbb{A}$), ordered by embeddings, is a wqo
- all the standard problems are undecidable.

The conjecture is confirmed for:

- graphs

WQO Dichotomy Conjecture:

For a homogeneous data domain $\mathbb{A}$, exactly one of the following conditions holds:

- age($\mathbb{A}$), ordered by embeddings, is a wqo
- all the standard problems are undecidable.

The conjecture is confirmed for:

- graphs
- directed graphs

WQO Dichotomy Conjecture:

For a homogeneous data domain $\mathbb{A}$, exactly one of the following conditions holds:

- age($\mathbb{A}$), ordered by embeddings, is a wqo
- all the standard problems are undecidable.

The conjecture is confirmed for:

- graphs
- directed graphs
- structures, where all relations are equivalences

WQO Dichotomy Conjecture:

For a homogeneous data domain $\mathbb{A}$, exactly one of the following conditions holds:

- age($\mathbb{A}$), ordered by embeddings, is a wqo
- all the standard problems are undecidable.

The conjecture is confirmed for:

- graphs
- directed graphs
- structures, where all relations are equivalences
- ...

idea of the proof for graphs

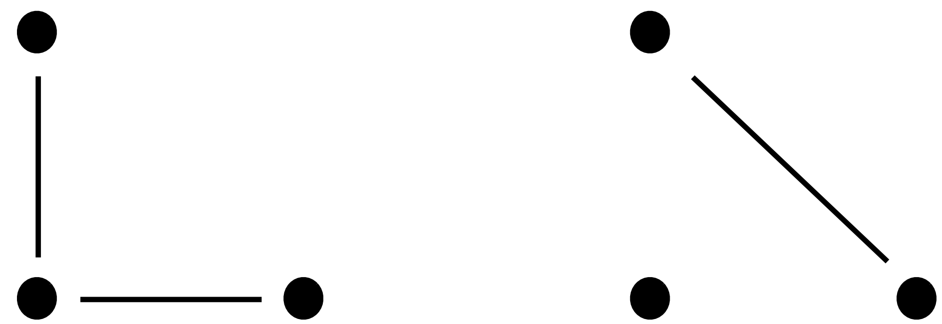
idea of the proof for graphs

Consider an amalgamation class of graphs.

idea of the proof for graphs

Consider an amalgamation class of graphs.

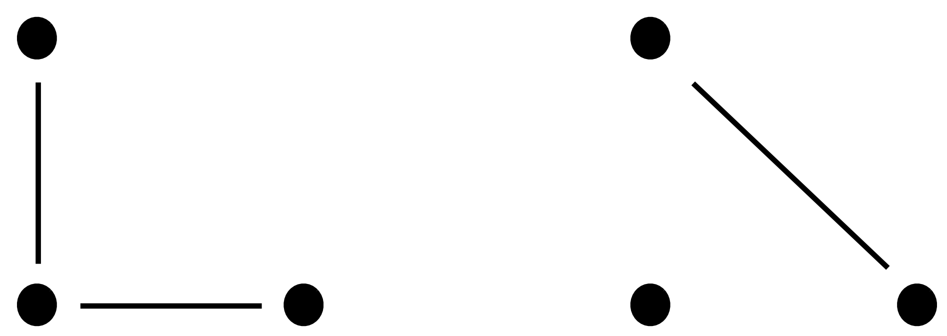
- if the class excludes any of the following graphs, wqo



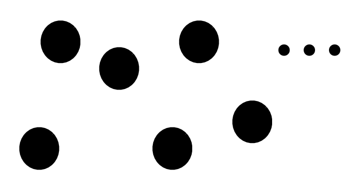
idea of the proof for graphs

Consider an amalgamation class of graphs.

- if the class excludes any of the following graphs, wqo



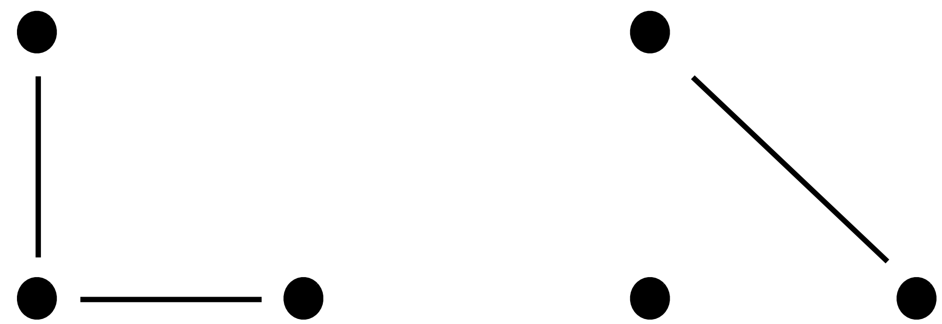
- otherwise, wlog. all co-cliques are in the class,



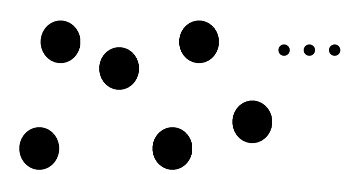
idea of the proof for graphs

Consider an amalgamation class of graphs.

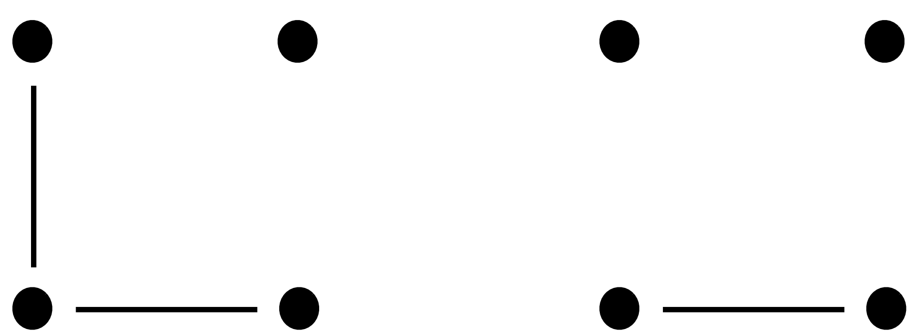
- if the class excludes any of the following graphs, wqo



- otherwise, wlog. all co-cliques are in the class,



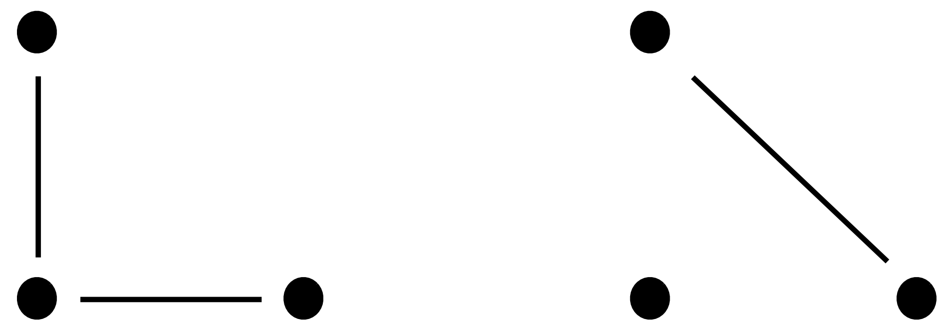
- and the following graphs are implied:



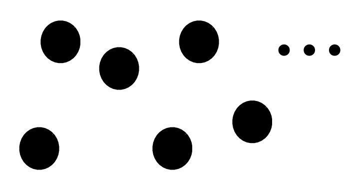
idea of the proof for graphs

Consider an amalgamation class of graphs.

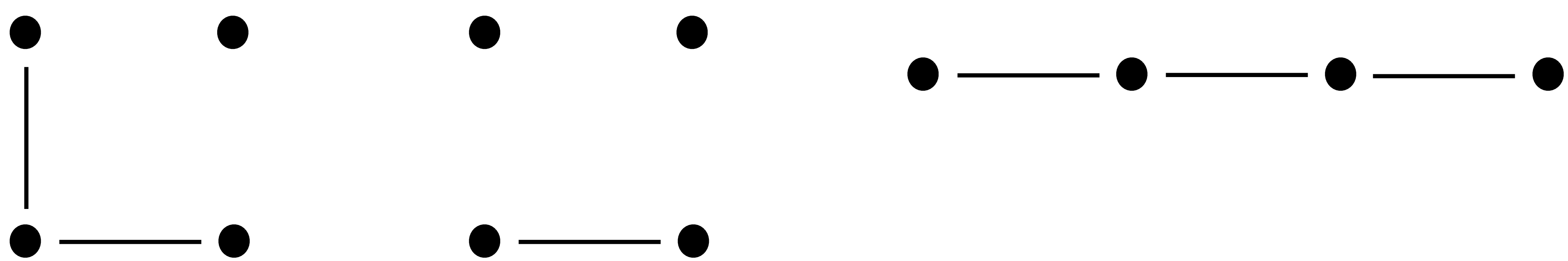
- if the class excludes any of the following graphs, wqo



- otherwise, wlog. all co-cliques are in the class,



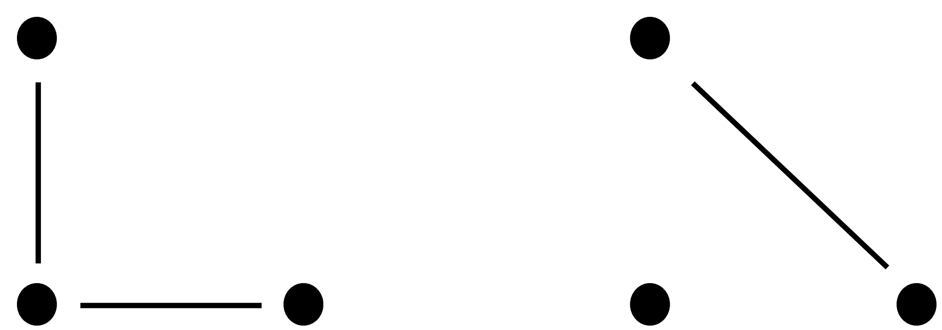
- and the following graphs are implied:



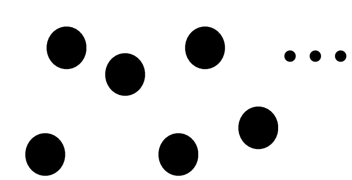
idea of the proof for graphs

Consider an amalgamation class of graphs.

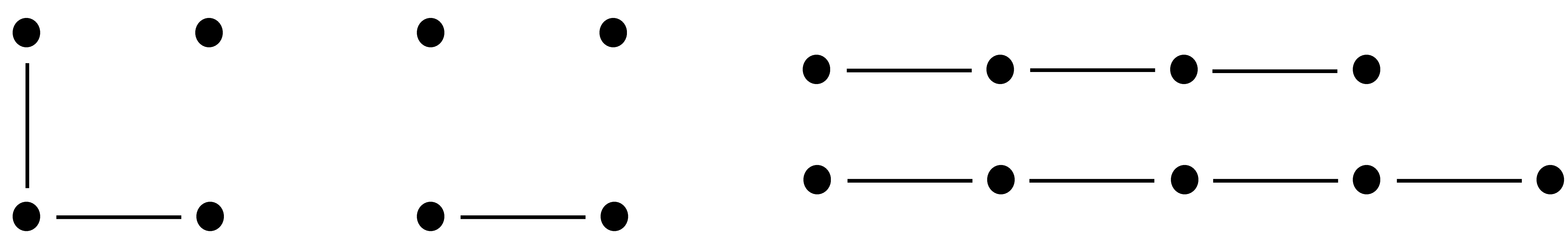
- if the class excludes any of the following graphs, wqo



- otherwise, wlog. all co-cliques are in the class,



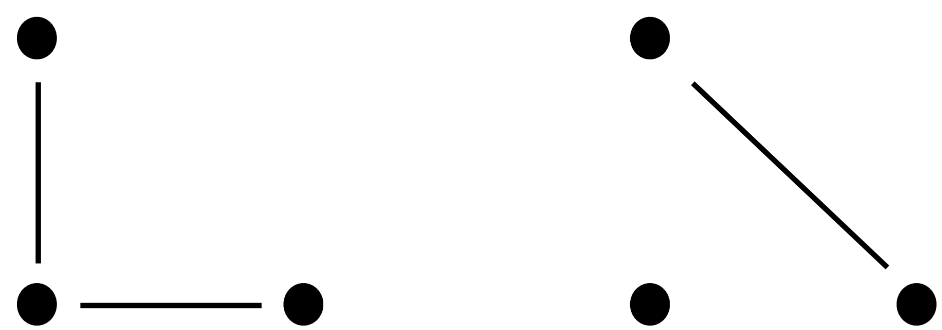
- and the following graphs are implied:



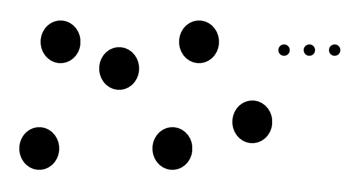
idea of the proof for graphs

Consider an amalgamation class of graphs.

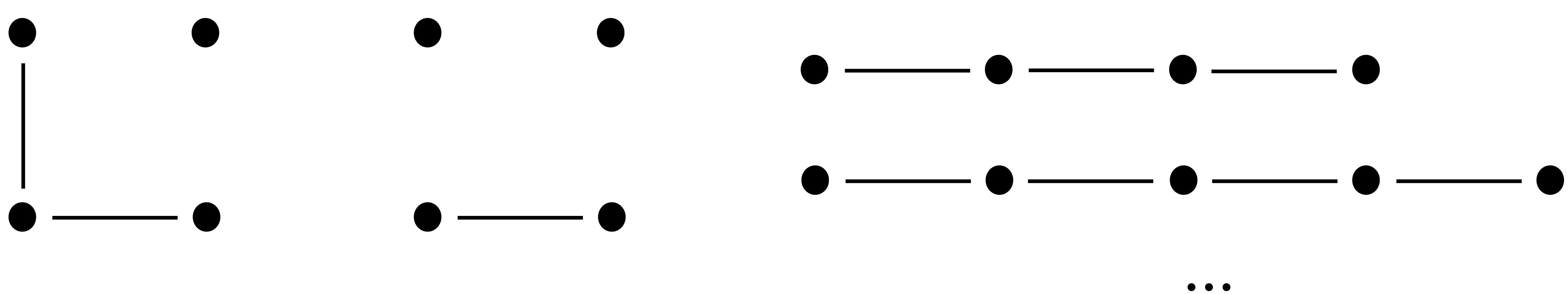
- if the class excludes any of the following graphs, wqo



- otherwise, wlog. all co-cliques are in the class,



- and the following graphs are implied:



WQO Dichotomy Conjecture:

For a homogeneous data domain $\mathbb{A}$, exactly one of the following conditions holds:

- age($\mathbb{A}$), ordered by embeddings, is a wqo
- all the standard problems are undecidable for Petri nets with data $\mathbb{A}$.

WQO Dichotomy Conjecture:

For a homogeneous data domain $\mathbb{A}$, exactly one of the following conditions holds:

- age($\mathbb{A}$), ordered by embeddings, is a wqo
- all the standard problems are undecidable for Petri nets with data $\mathbb{A}$.

Likewise for other wqo-based algorithms, for instance emptiness of alternating automata with 1-register.

WQO Dichotomy Conjecture:

For a homogeneous data domain $\mathbb{A}$, exactly one of the following conditions holds:

- age($\mathbb{A}$), ordered by embeddings, is a wqo
- all the standard problems are undecidable for Petri nets with data $\mathbb{A}$.

Likewise for other wqo-based algorithms, for instance emptiness of alternating automata with 1-register.

thank you!