The semiring of dynamical systems

Séminaire CANA

Antonio E. Porreca • aeporreca.org
LIS, Marseille • 6 October 2020

Dramatis personae

In (partial) order of appearance

Antonio E. Porreca .... 🇫🇷 Aix-Marseille Université & LIS

Luca Manzoni .... 🇮🇹 Università degli Studi di Trieste

Enrico Formenti .... 🇫🇷 Université Côte d’Azur & I3S

Valentina Dorigatti .... 🇮🇹 Università degli Studi dell’Insubria

Alberto Dennunzio .... 🇮🇹 Università degli Studi di Milano-Bicocca

Maximilien Gadouleau .... 🇬🇧 Durham University

Florian Bridoux .... 🇫🇷 Aix-Marseille Université & LIS

Caroline Gaze-Maillot .... 🇫🇷 Aix-Marseille Université & LIS

Émile Naquin-Touileb .... 🇫🇷 ENS Lyon & LIS

Other characters

Doing related work

Sara Riva .... 🇫🇷 Université Côte d’Azur & I3S

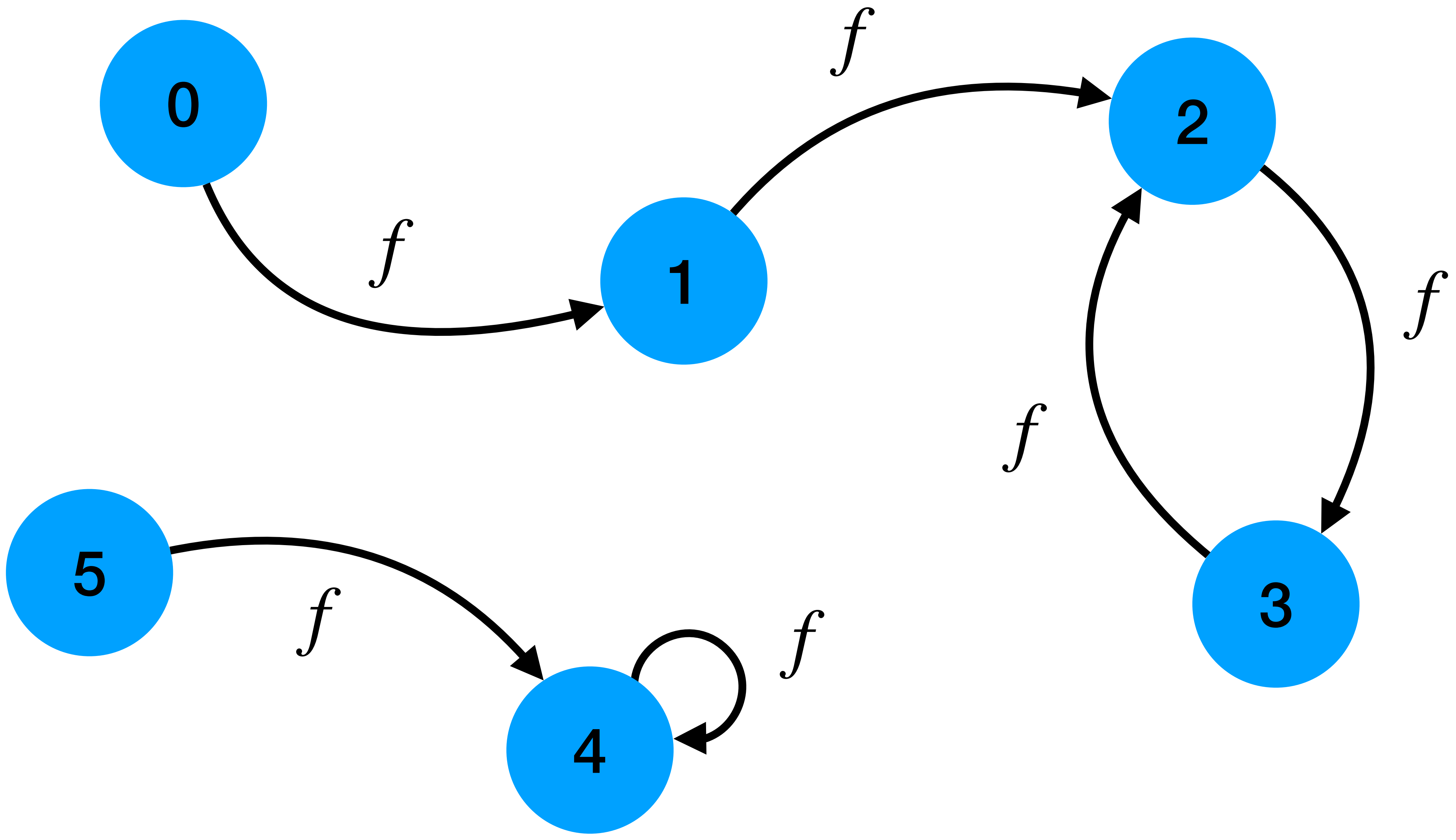
Valentin Montmirail .... 🇫🇷 Université Côte d’Azur & I3S

Luciano Margara .... 🇮🇹 Università degli Studi di Bologna

Finite, discrete-time dynamical systems

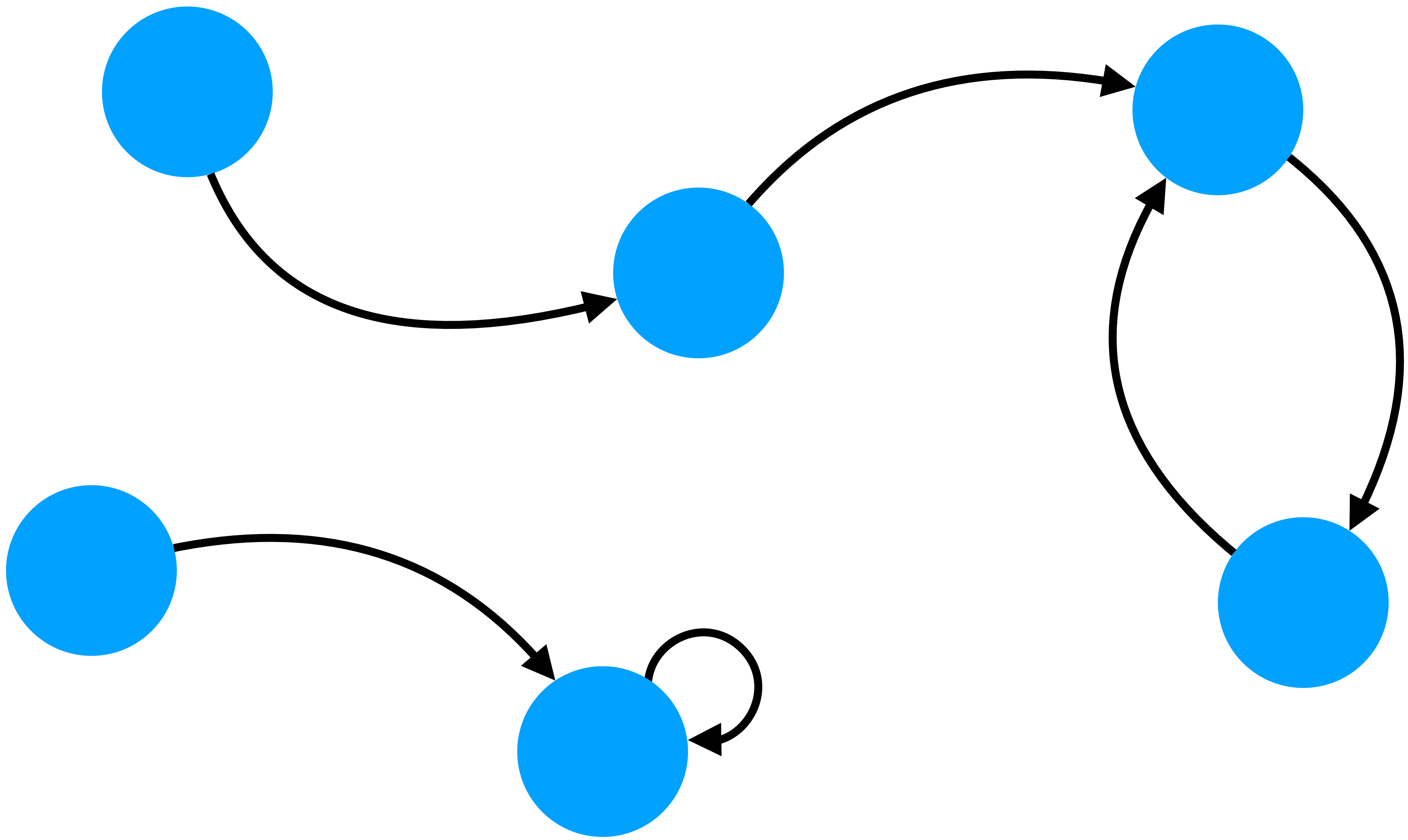
Finite, discrete-time dynamical systems

Just a finite set with a transition function (A, f)



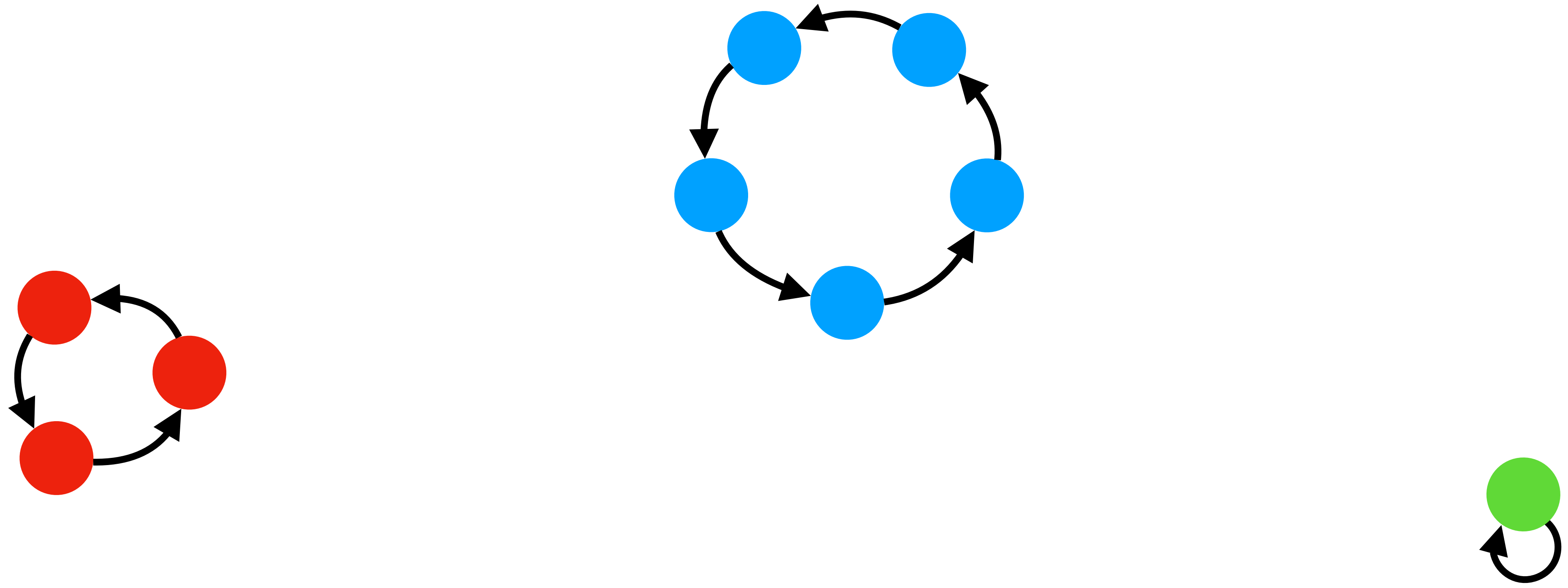
Finite, discrete-time dynamical systems

Just a finite set with a transition function (A, f) modulo isomorphism



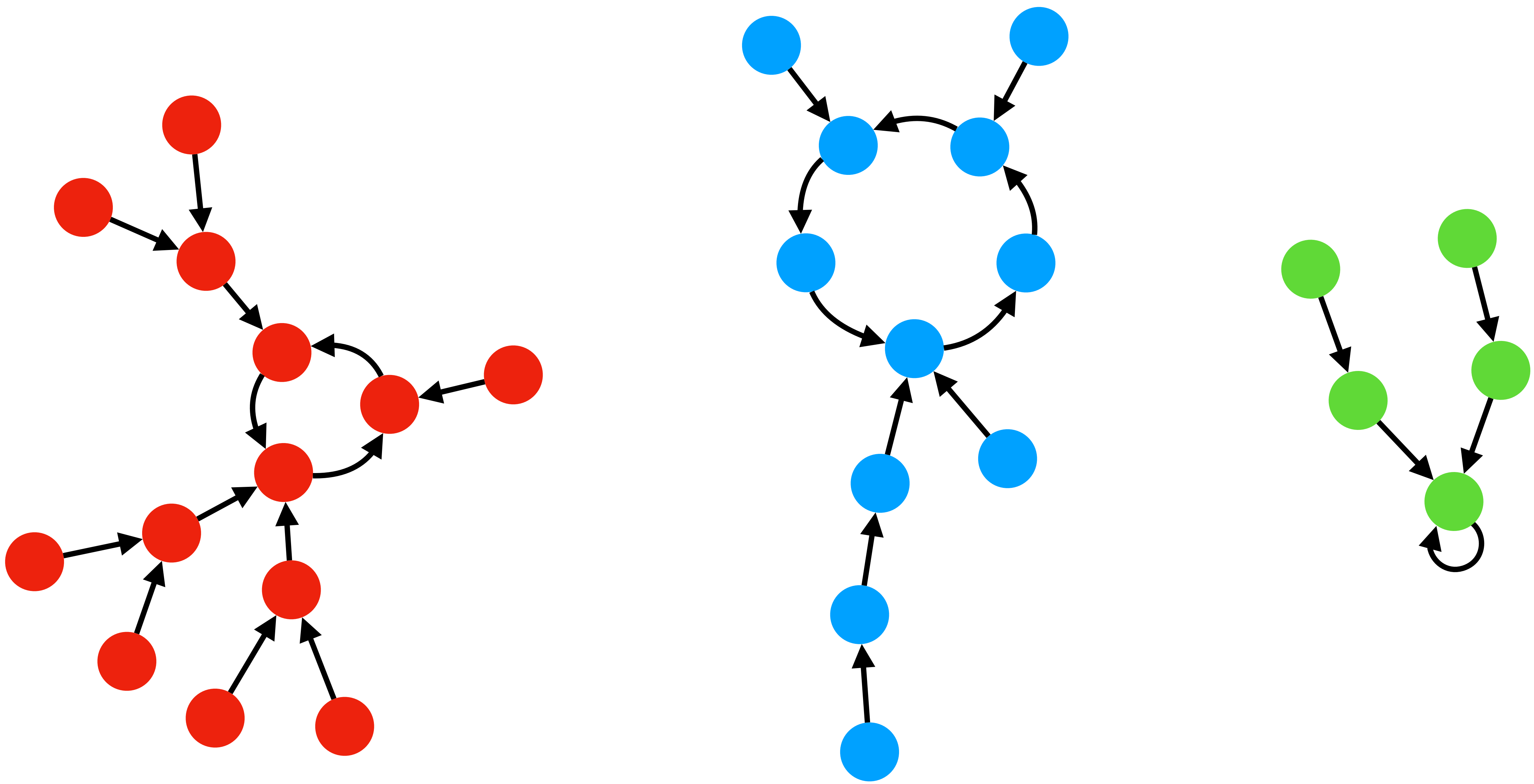
General shape of a dynamical system

A few limit cycles



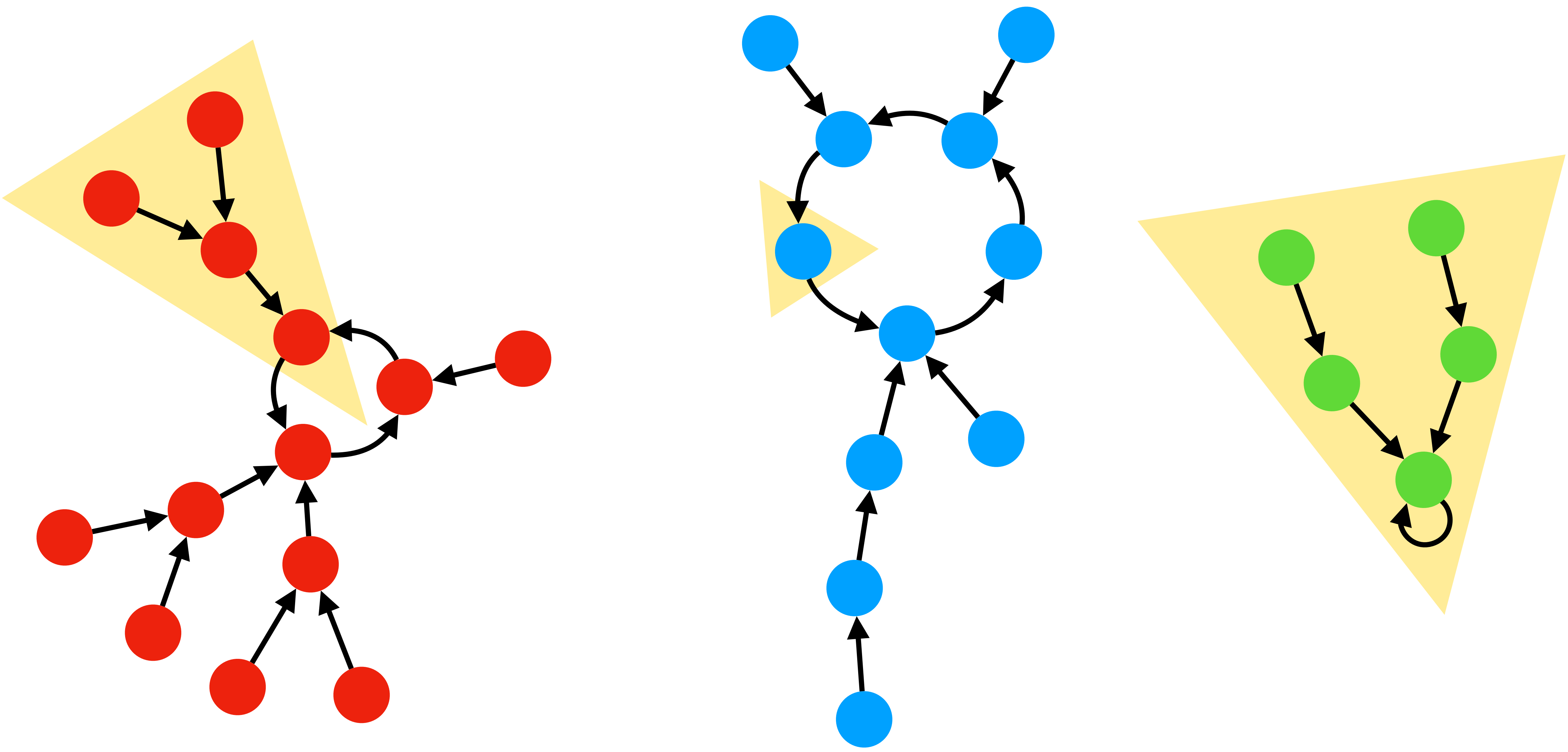
General shape of a dynamical system

A few limit cycles with trees going in



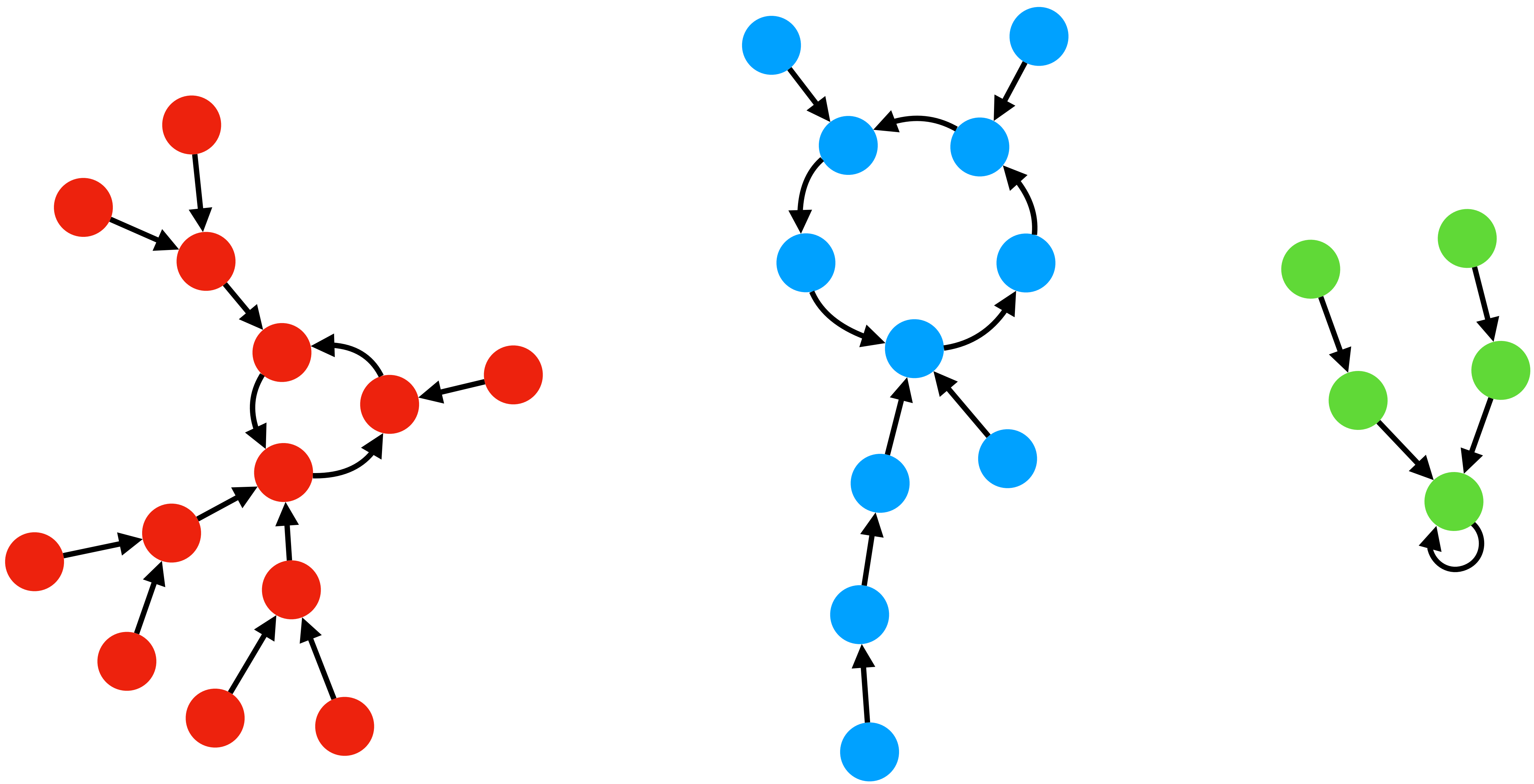
General shape of a dynamical system

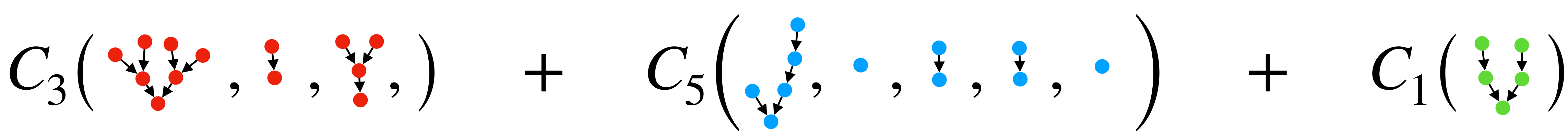
A few limit cycles with trees going in



General shape of a dynamical system

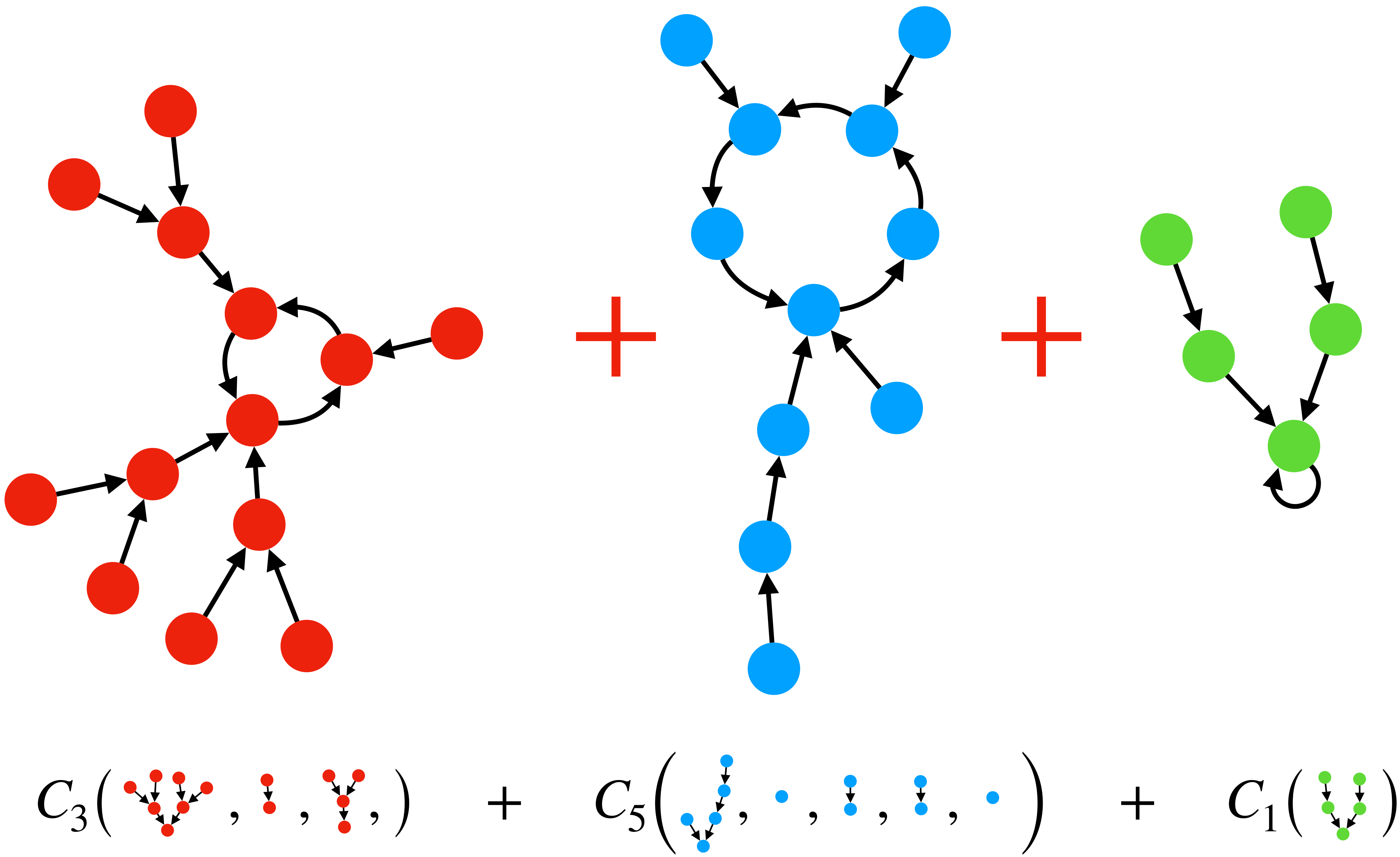
A few limit cycles **with trees going in**





General shape of a dynamical system

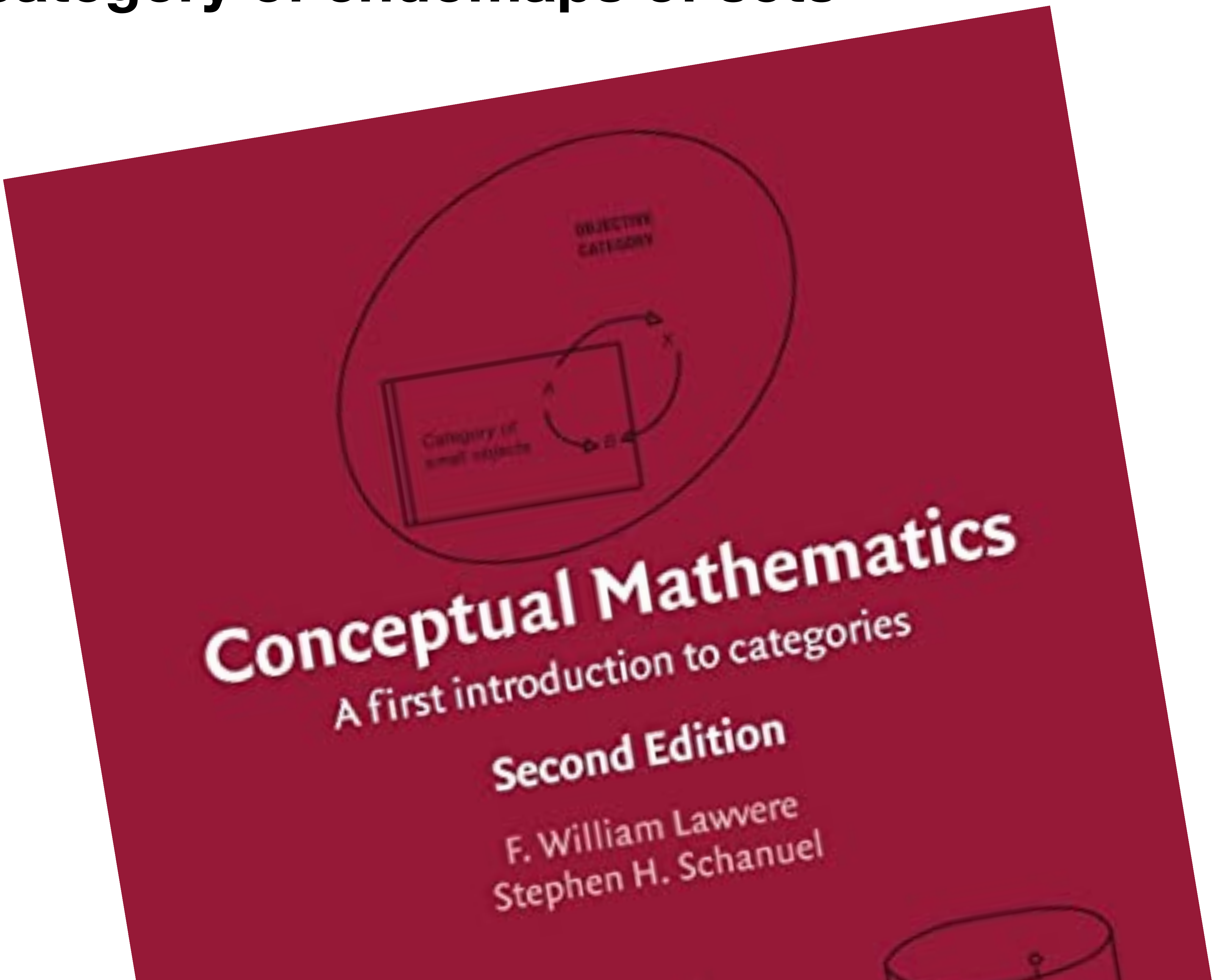
A few limit cycles **with trees going in**



The category $\mathbf{D}$ of dynamical systems

The inspiration

The category of endomaps of sets



Objects & arrows 🏹

- The objects are the dynamical systems (A, f)
- An arrow $(A, f) \xrightarrow{\varphi} (B, g)$ is a function $\varphi\colon A \to B$ which commutes with f and g

$$\begin{array}{ccc} A & \xrightarrow{f} & A \\ \varphi \downarrow & & \downarrow \varphi \\ B & \xrightarrow[g]{} & B \end{array}$$

The category $\mathbf{D}$ has sums (coproducts)

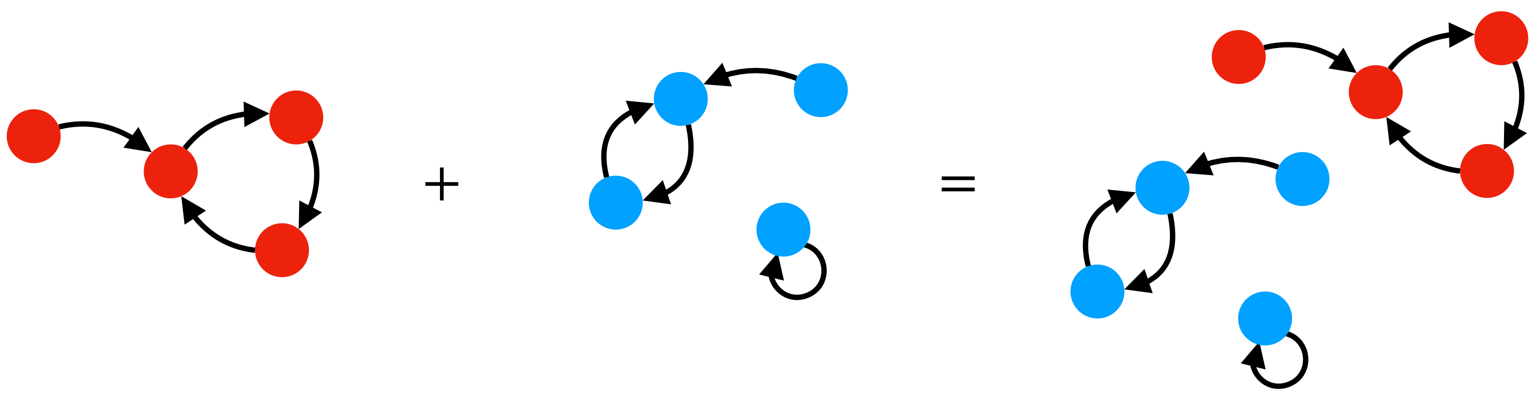
Necessary but not that interesting

- In graph-theoretic terms, it's just the disjoint union

$$(A, f) + (B, g) = (A \uplus B, f + g) \quad \text{with } (f + g)(x) = \begin{cases} f(x) & \text{if } x \in A \\ g(x) & \text{if } x \in B \end{cases}$$

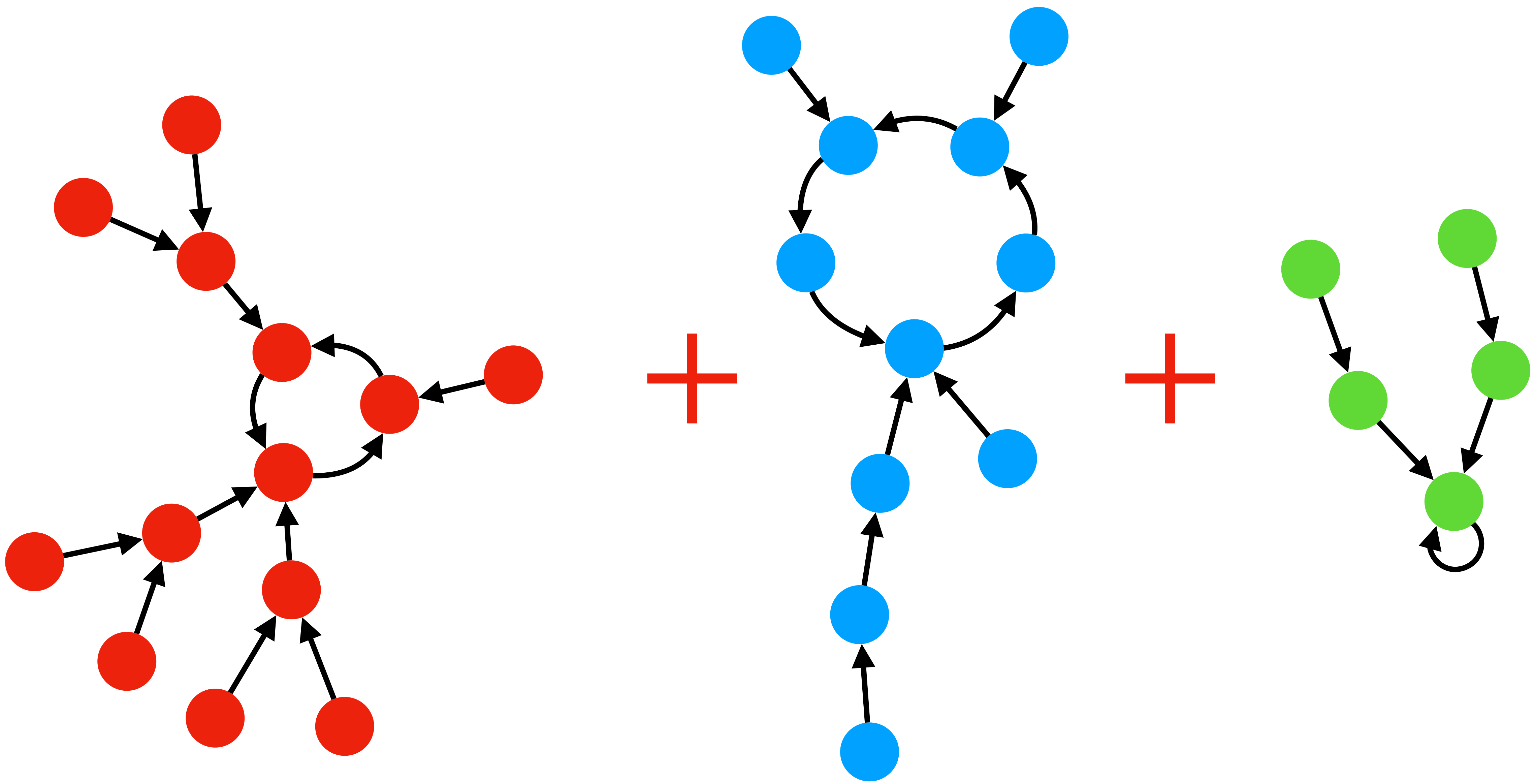
- This represents the alternative execution of A and B

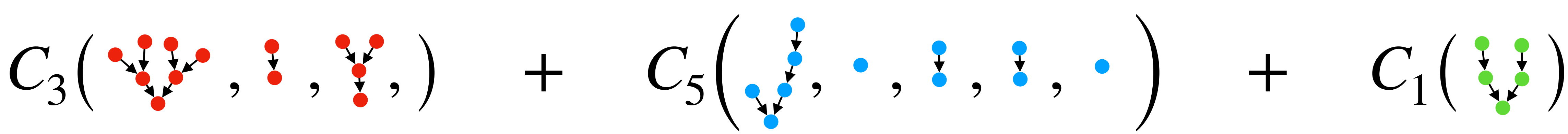
- The identity is the empty system $\mathbf{0} = (\varnothing, \varnothing)$



General shape of a dynamical system

It’s a sum of cycles with trees going in





The category $\mathbf{D}$ admits products

Now we're talking!

- In graph-theoretic terms, it's the tensor product

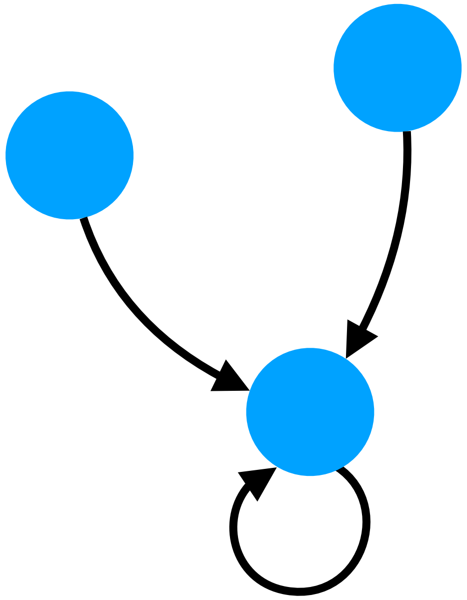
$$(A, f) \times (B, g) = (A \times B, f \times g)$$

$$\text{with } (f \times g)(a, b) = (f(a), g(b))$$

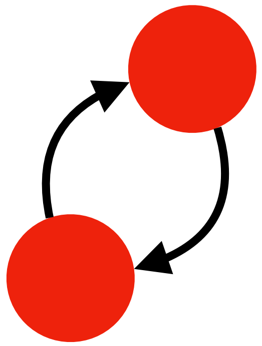
- This represents the synchronous execution of A and B
- The identity is the singleton system $\mathbf{1} = (\{0\}, \mathrm{id})$

Product in $\mathbf{D}$ is graph tensor product

Two systems modulo isomorphism



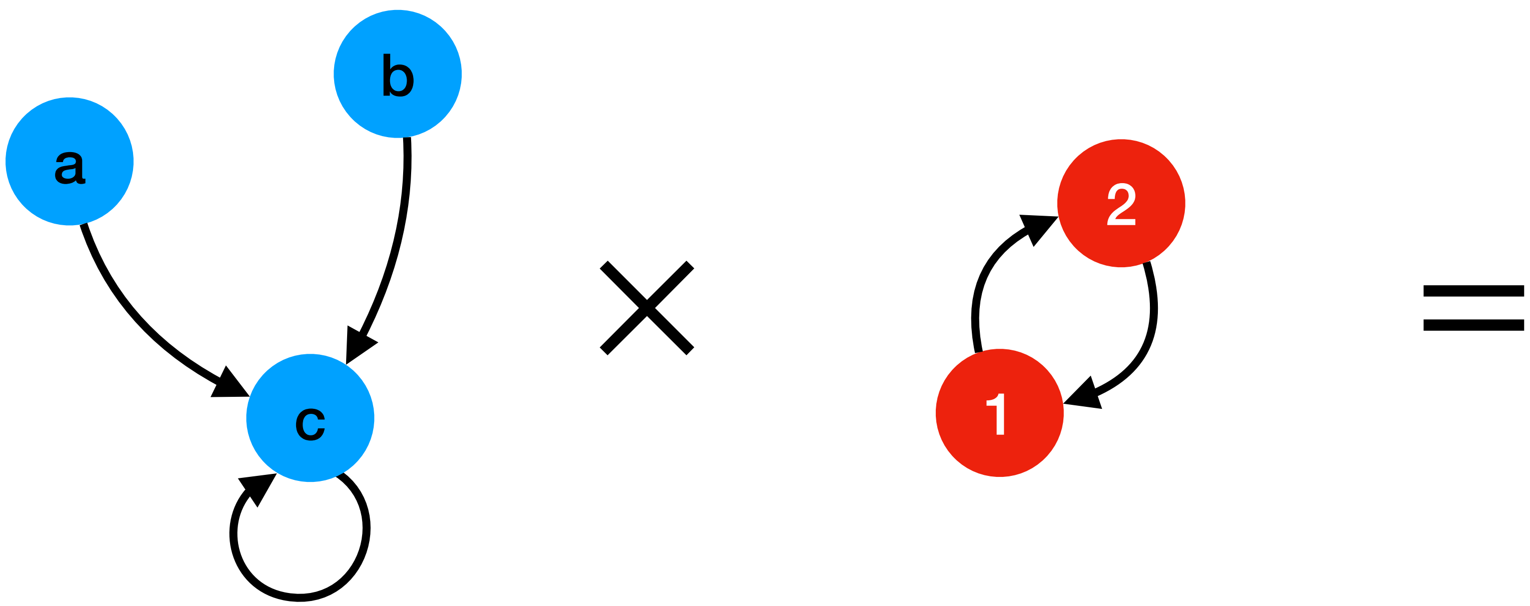
$\times$



$=$

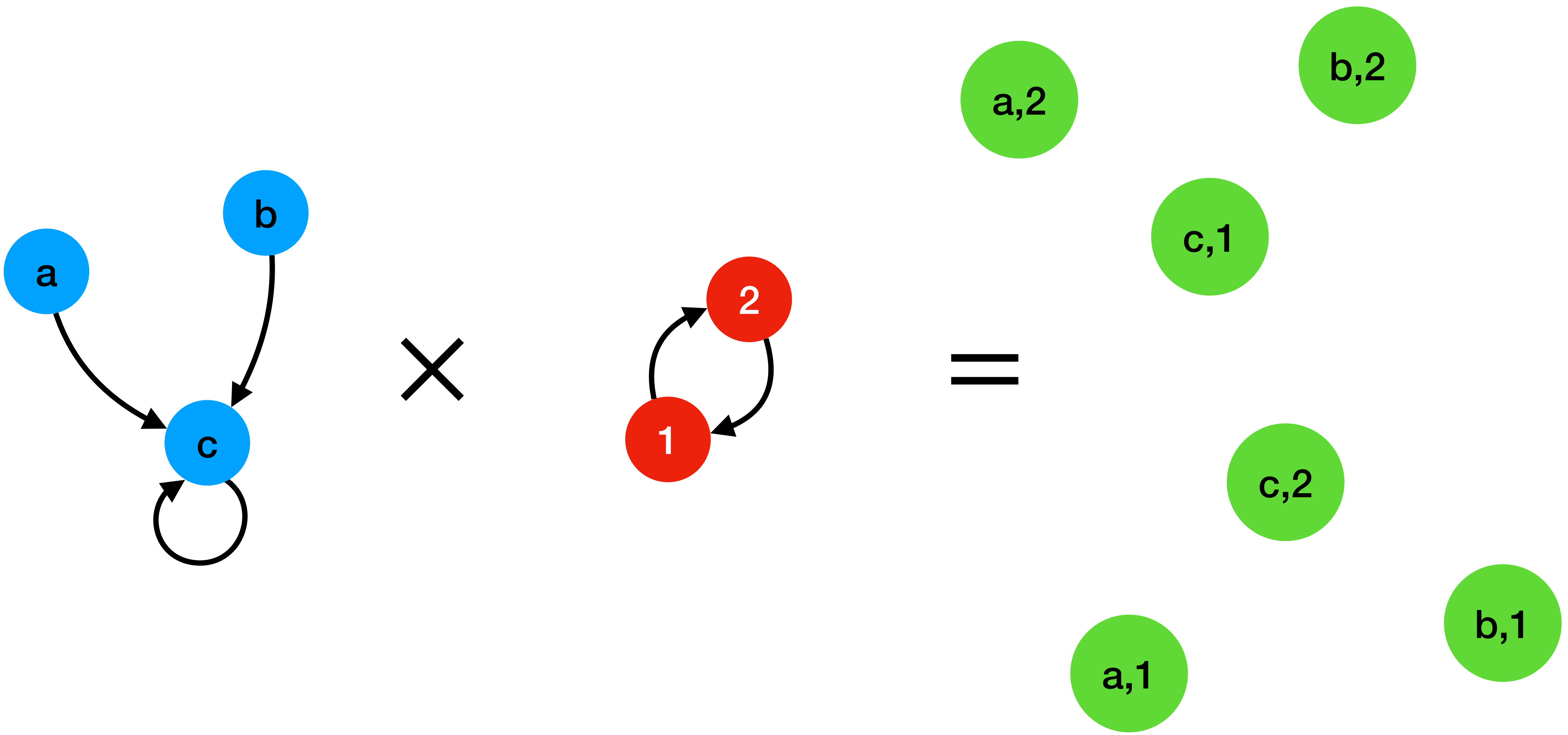
Product in $\mathbf{D}$ is graph tensor product

Temporary state names



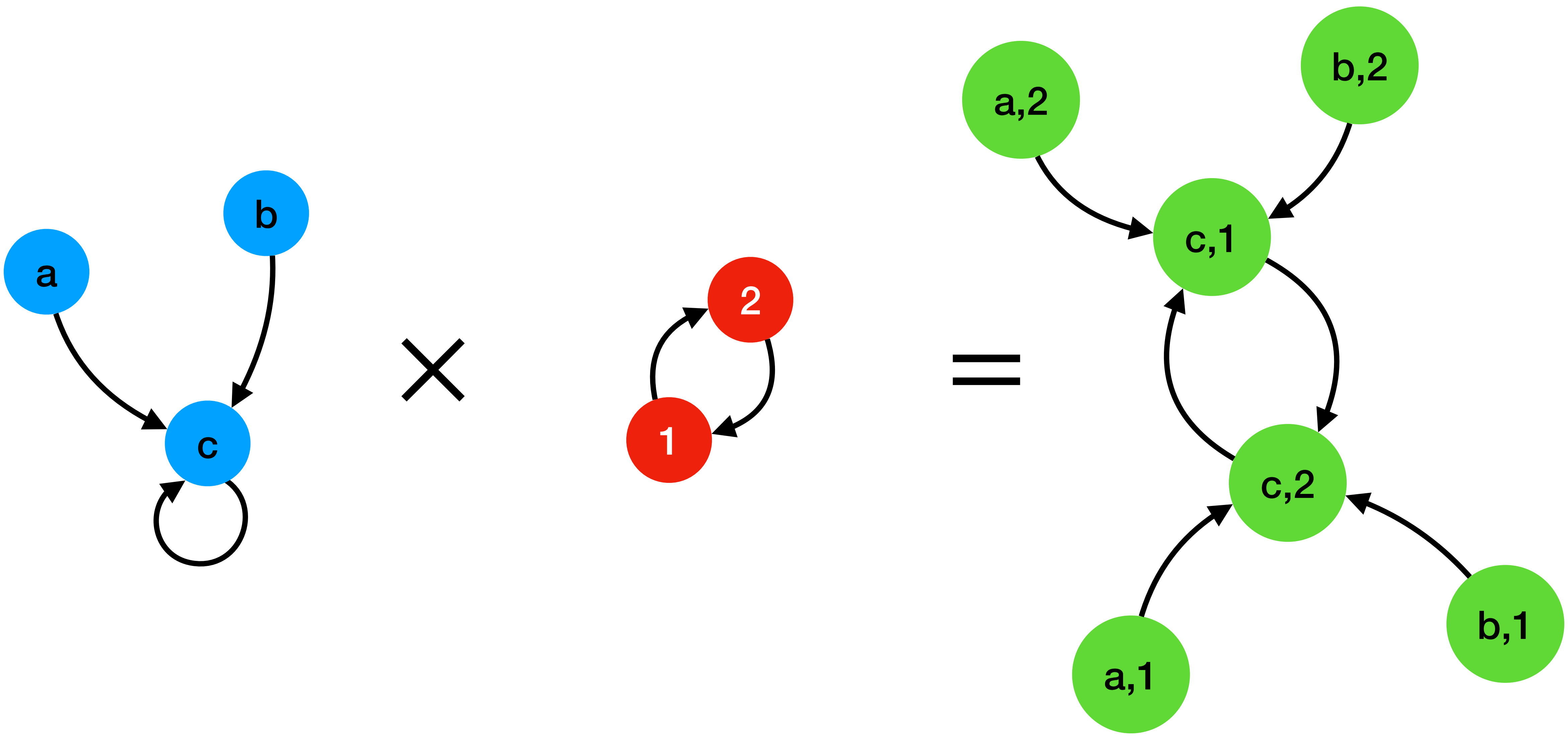
Product in $\mathbb{D}$ is graph tensor product

Cartesian product of the states



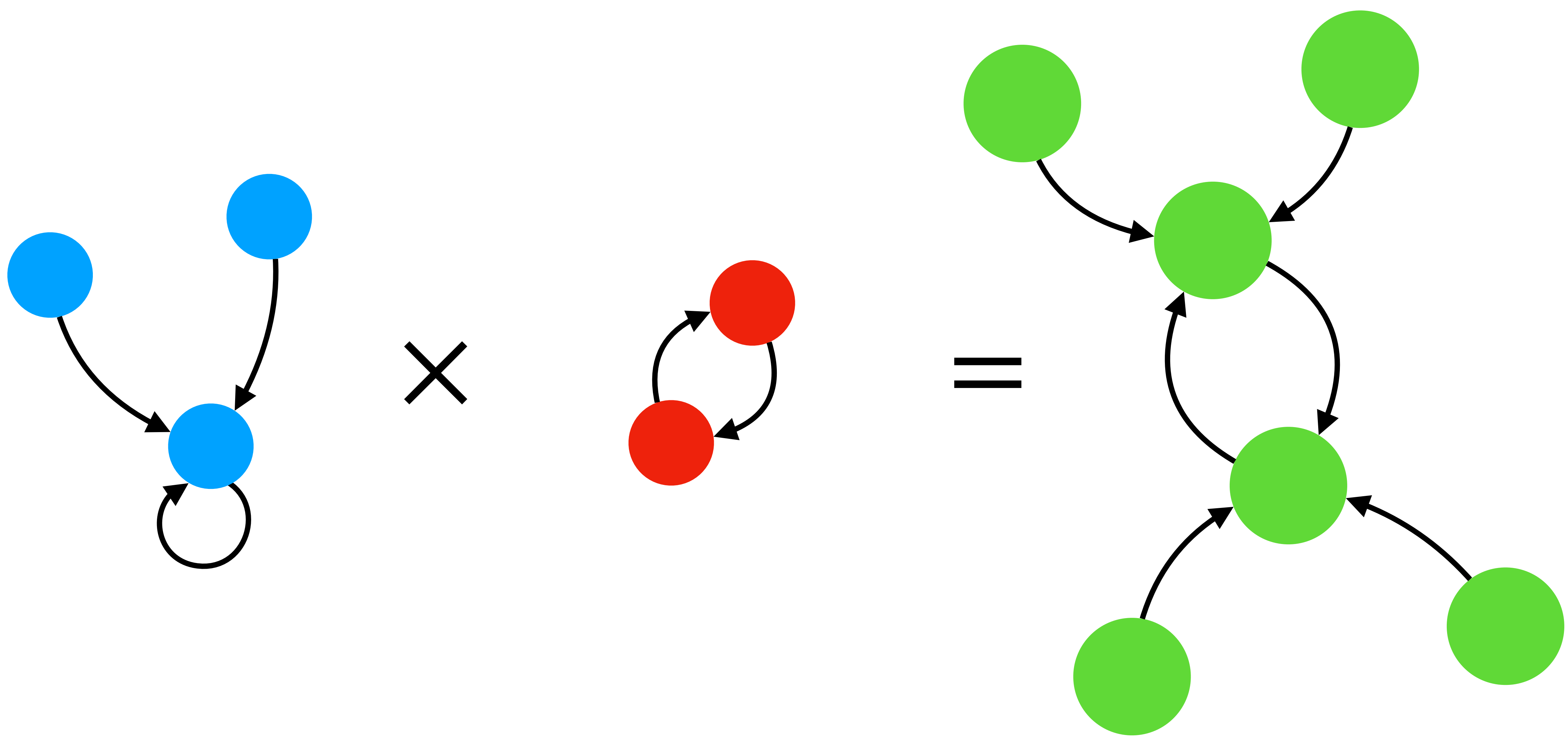
Product in $\mathbb{D}$ is graph tensor product

Arrows iff arrows between both components

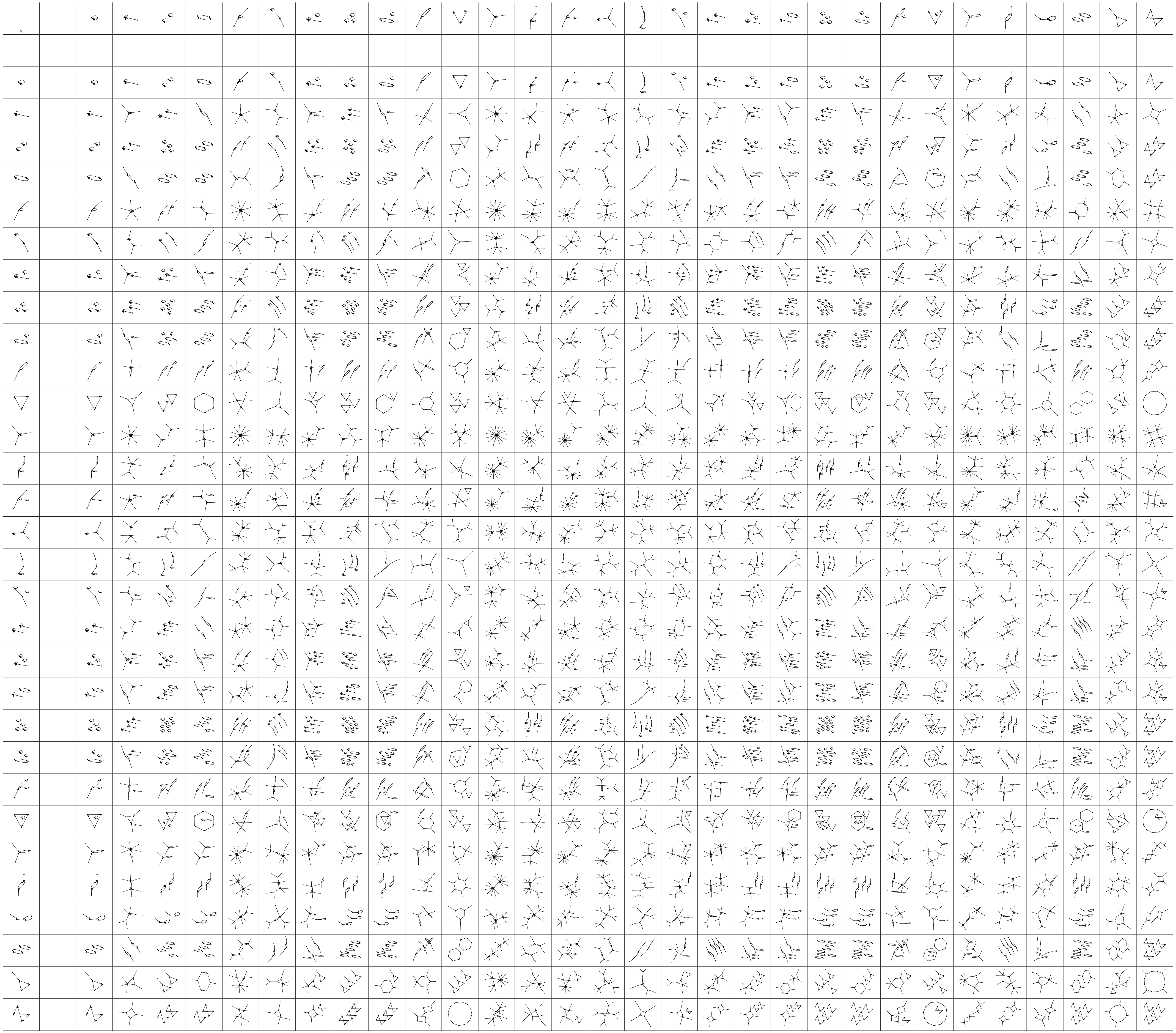


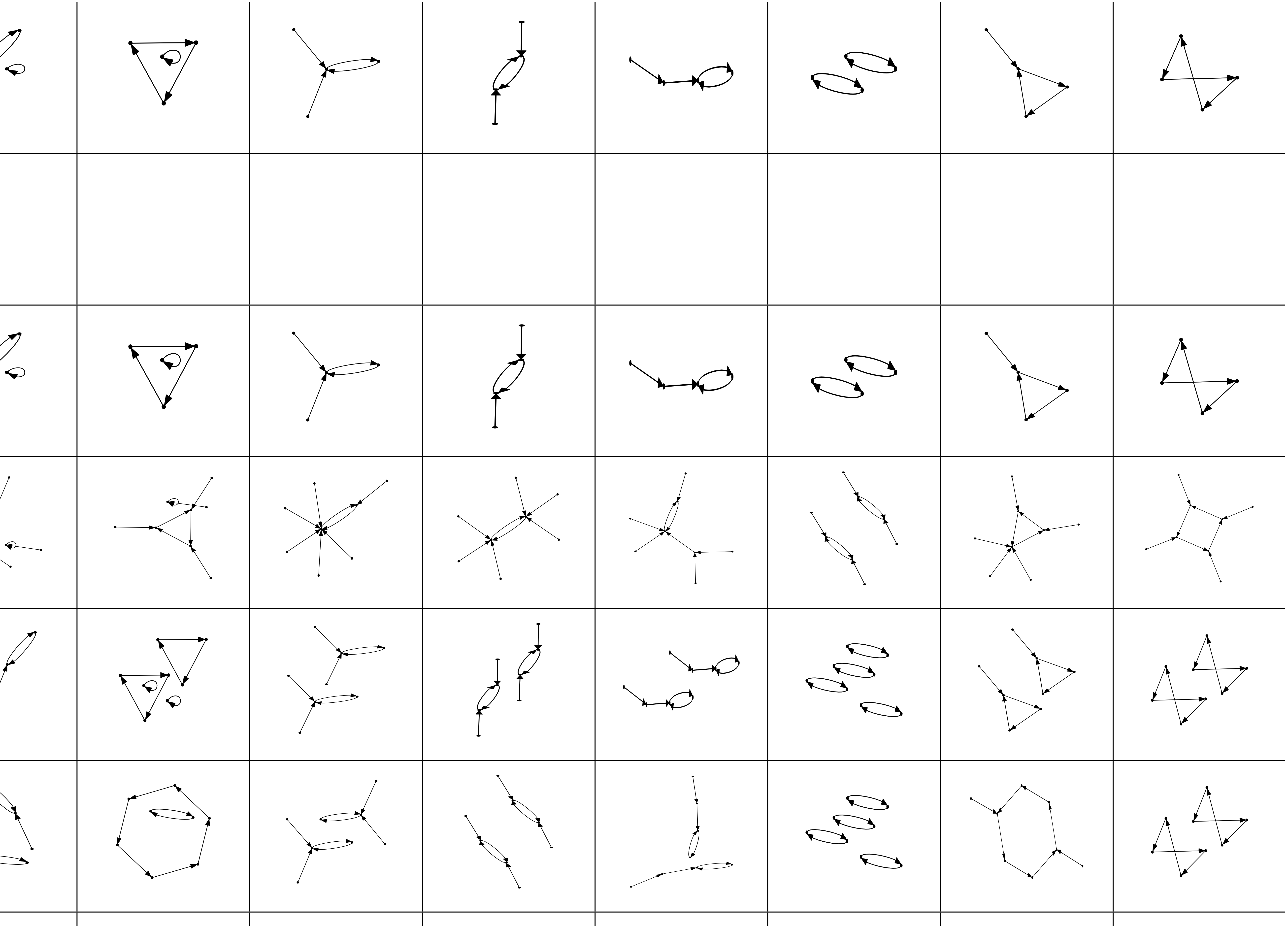
Product in $\mathbf{D}$ is graph tensor product

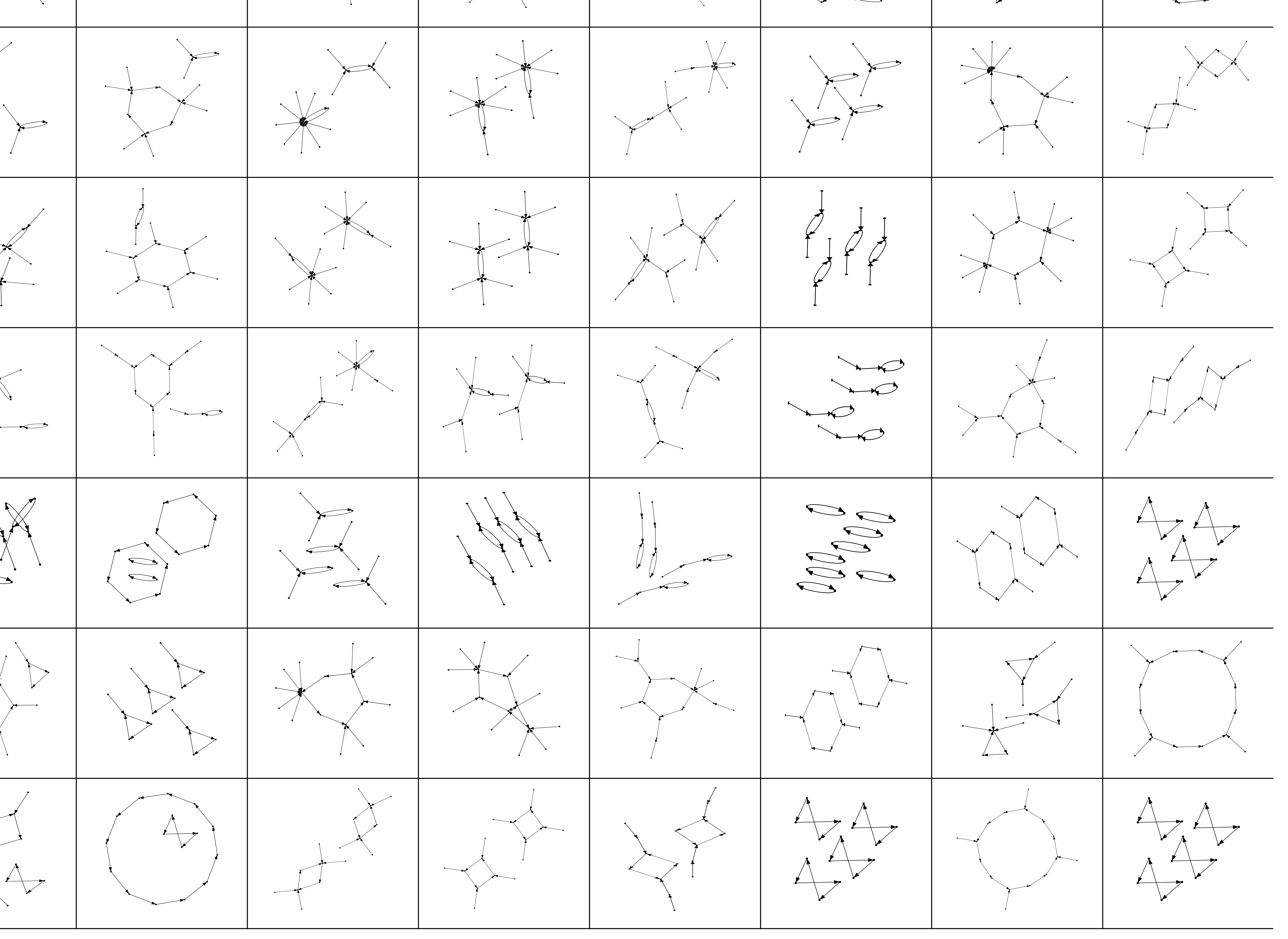
We forget the state names once again

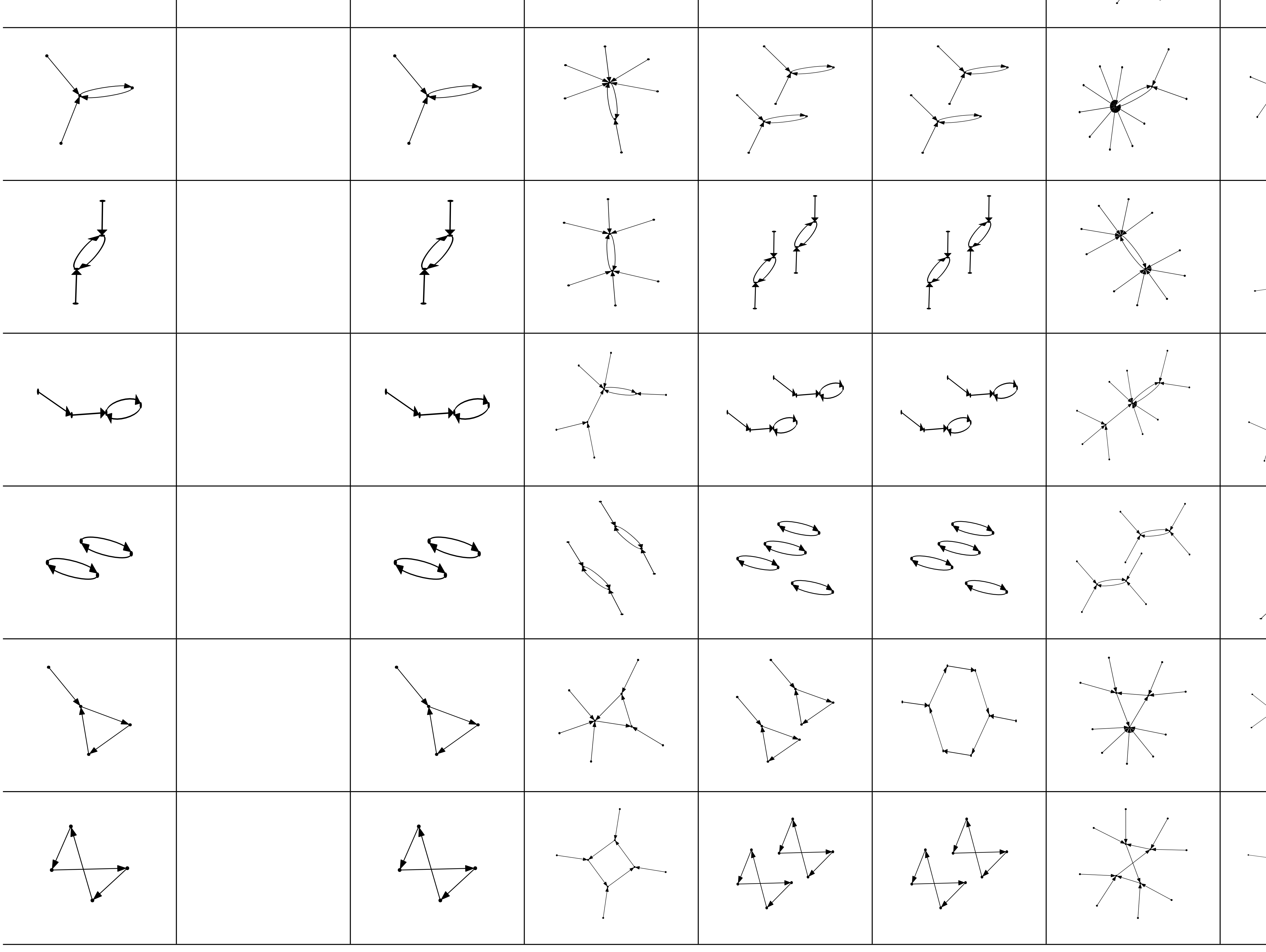


Introducing: the multiplication table, poster-size

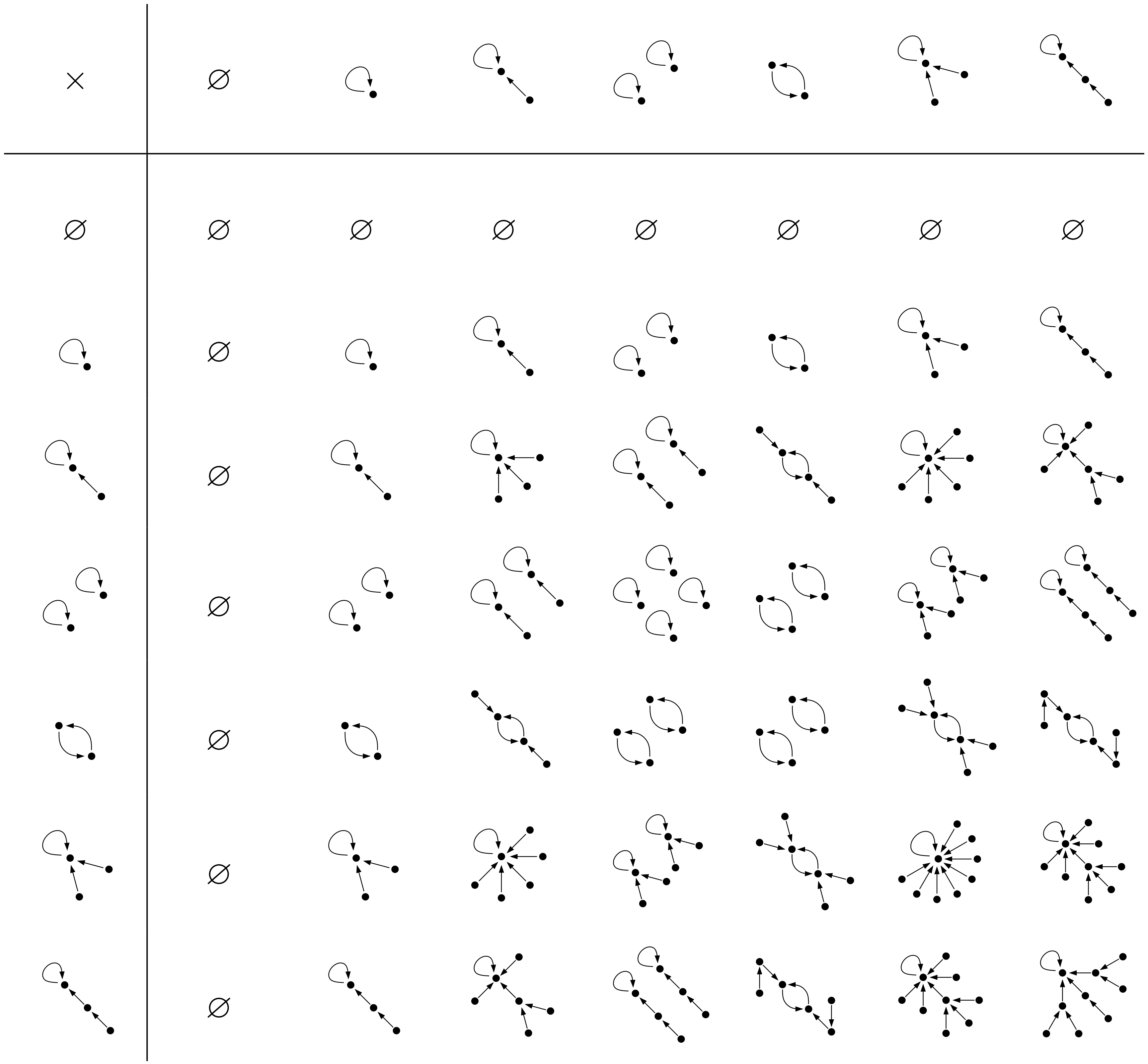








Prettier version



The semiring $\mathbf{D}$ of dynamical systems

$\mathbf{D}$ (modulo isomorphisms) is a semiring

Like a ring, without subtraction

- Product is (modulo isomorphism) commutative, associative and has identity $\mathbf{0} = (\varnothing, \varnothing)$ in any category where it exists; so, it's a commutative monoid

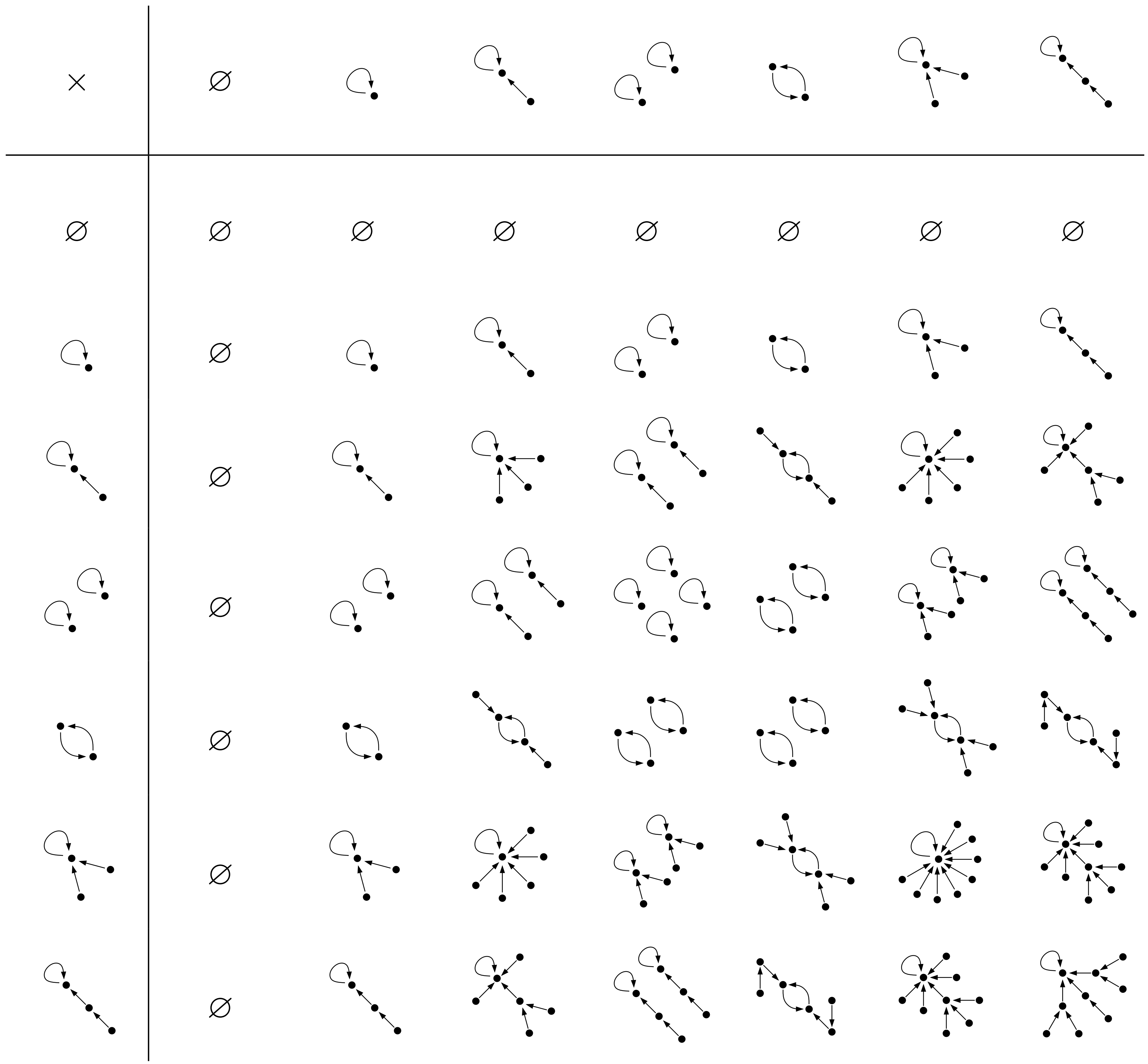
- Sum is (modulo isomorphism) commutative, associative and has identity $\mathbf{1} = (\{0\}, \mathrm{id})$ in any category where it exists; so, another commutative monoid

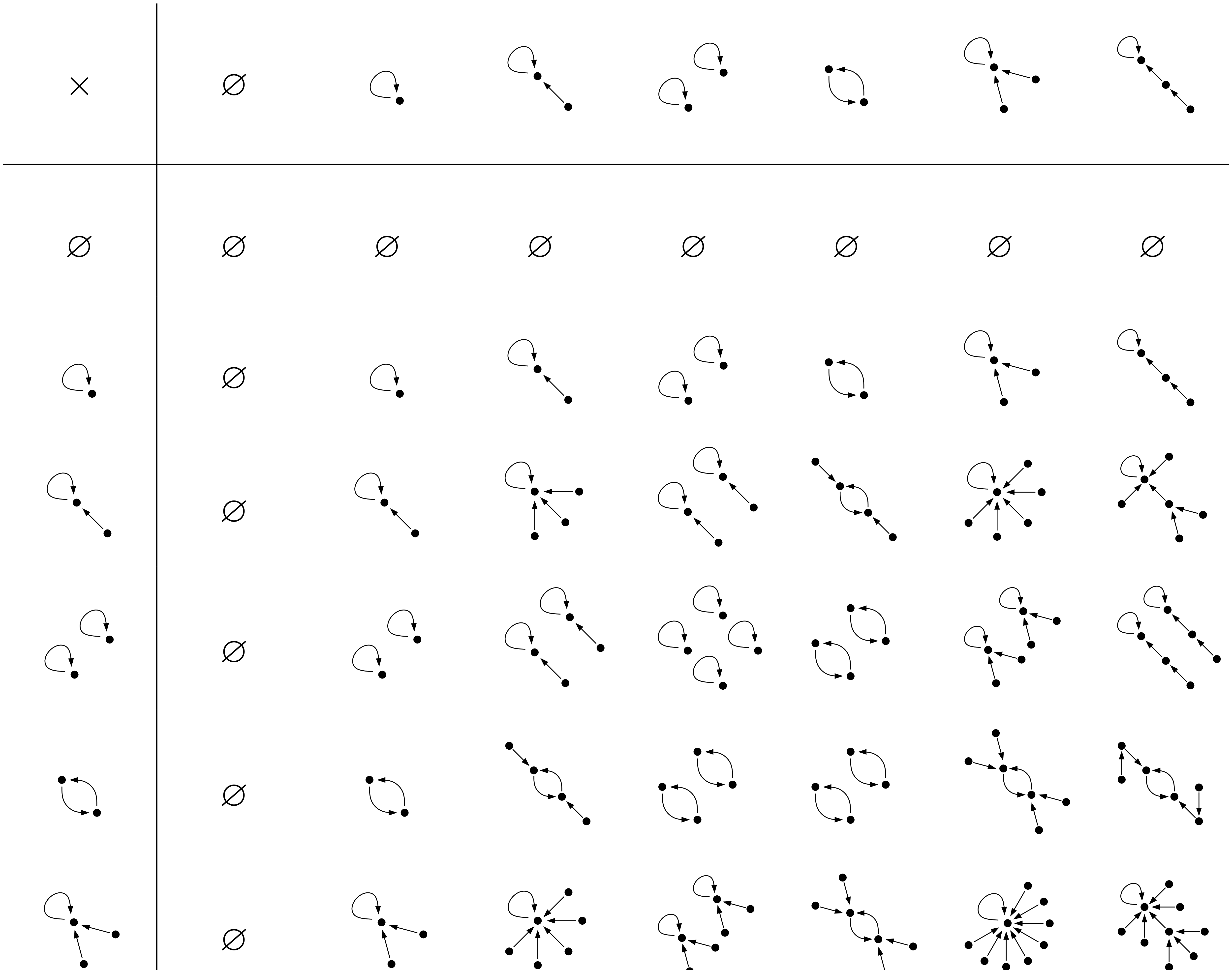
- The sum is the free commutative monoid (i.e., the multisets) over the set of connected, nonempty dynamical systems

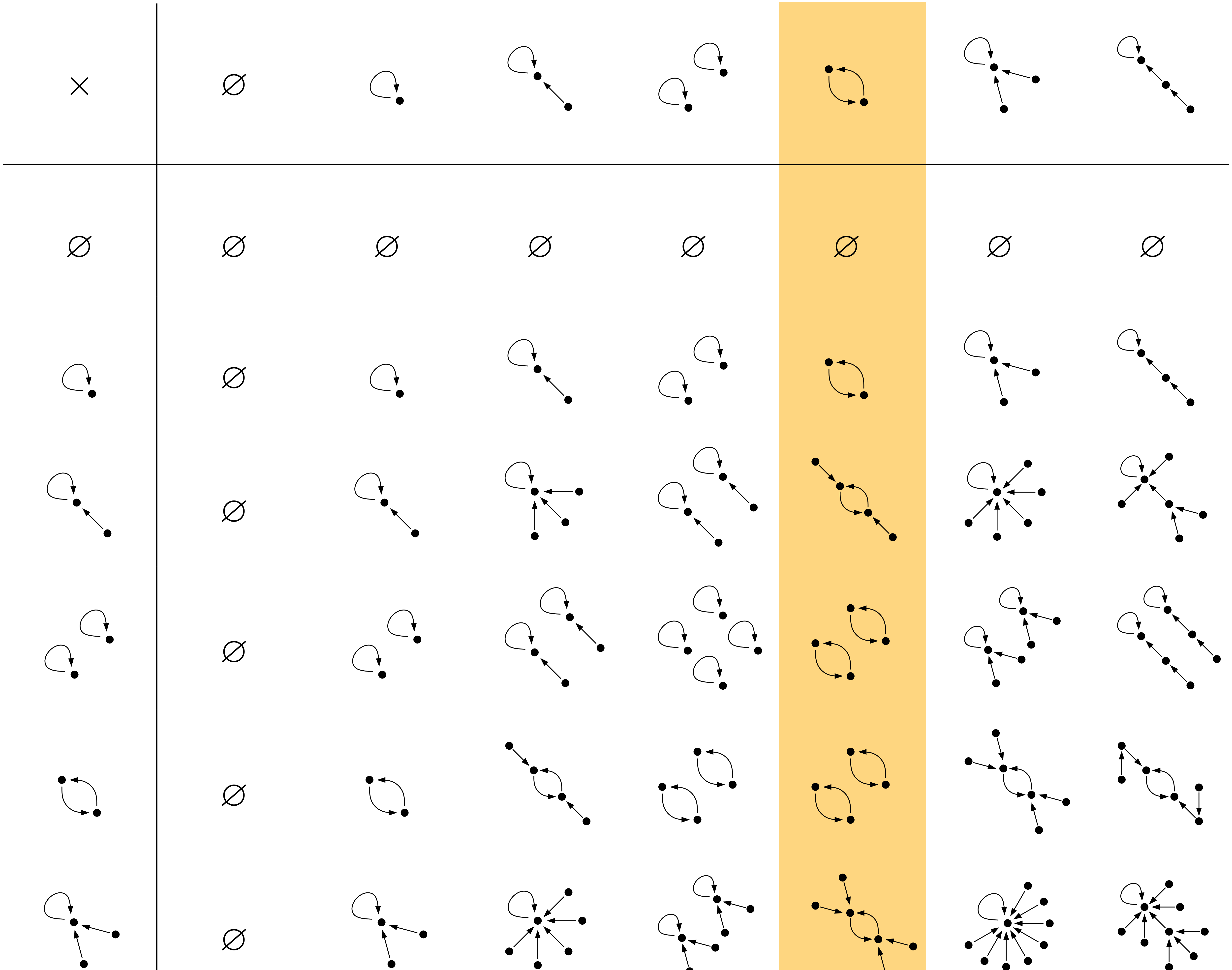
- The distributive law and the product annihilation law do not hold for arbitrary categories, but they do here

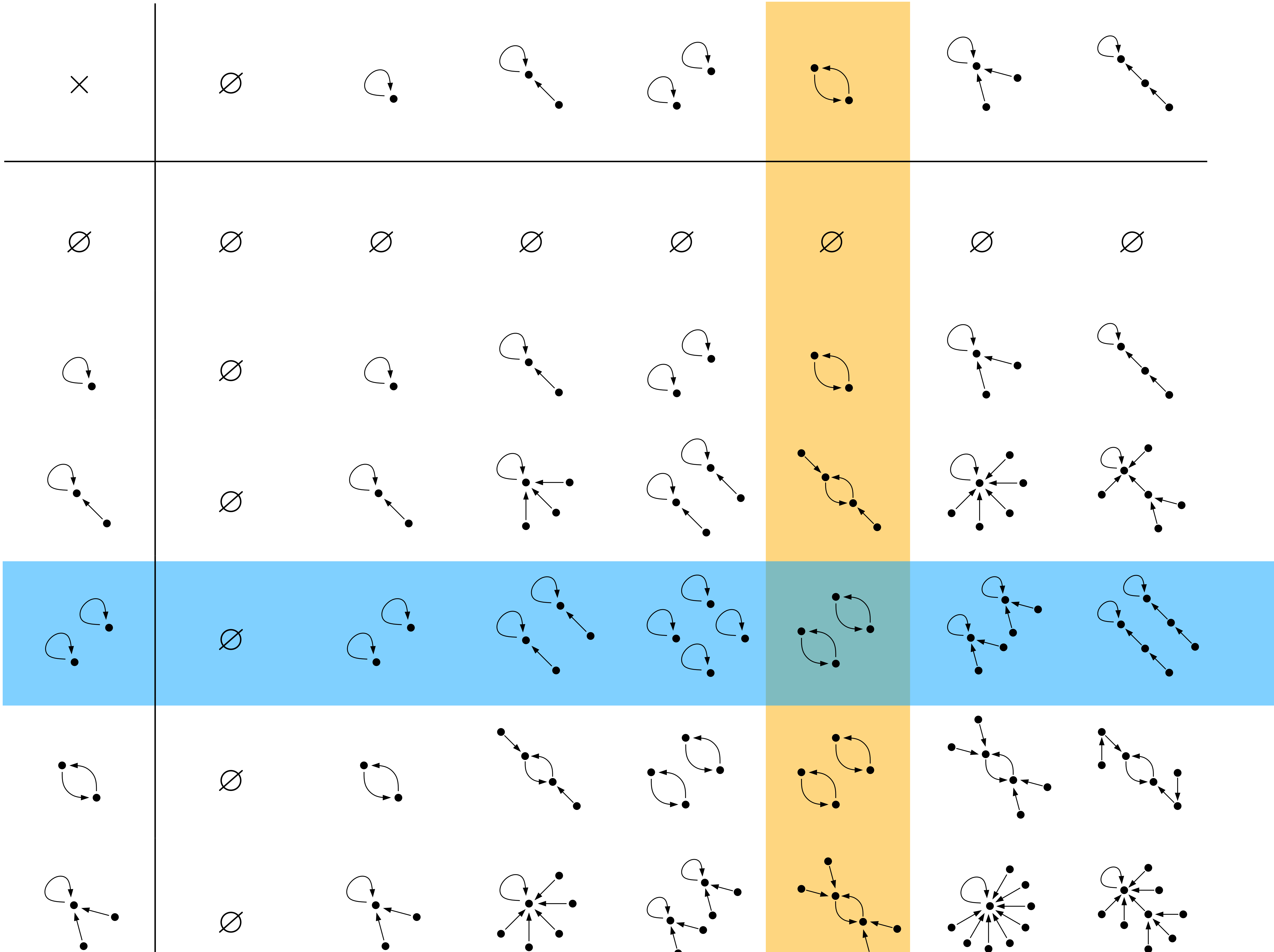
No unique factorisation!

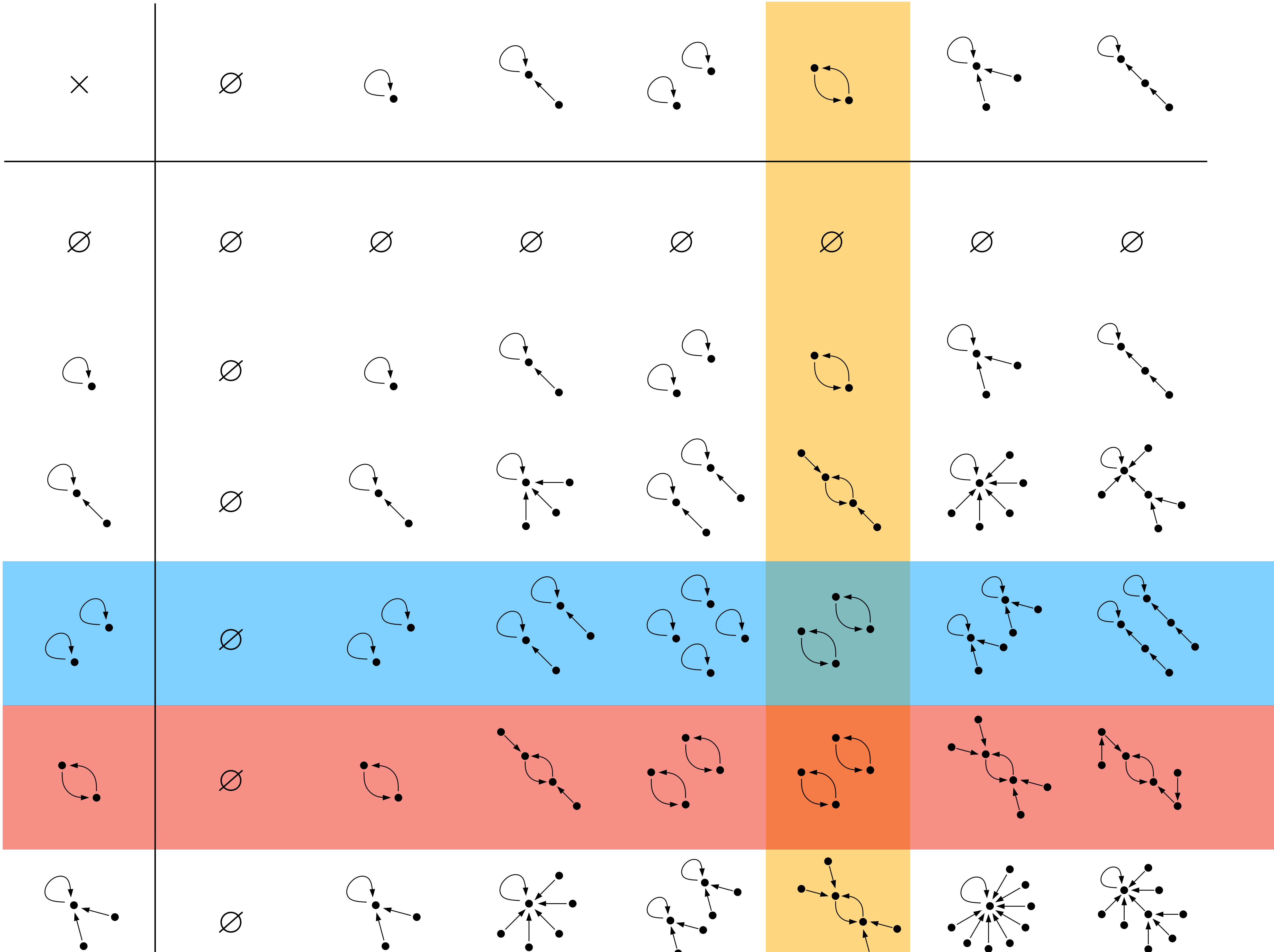
Multiplication table





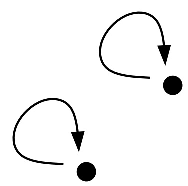
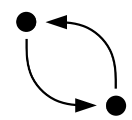
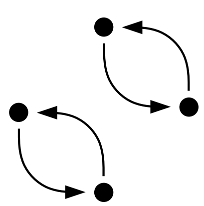


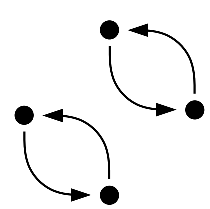
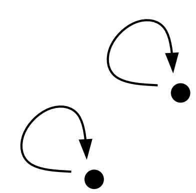
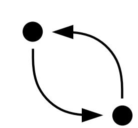


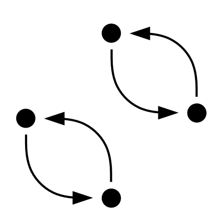
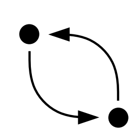
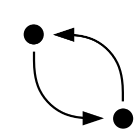


No unique factorisation

And the counterexample is minuscule

- The systems  and  are irreducible
- Any system with a prime number of states is irreducible, since the state space is a cartesian product
- So  has two distinct factorisations into irreducibles

 $=$  $\times$ 

 $=$  $\times$ 

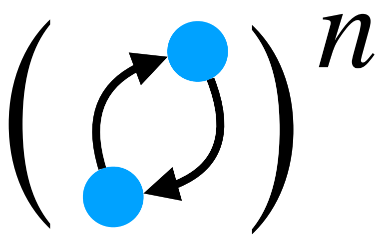
Systems with arbitrarily many factorisations

Theorem

For each n, there exist a dynamical system with at least n factorisations

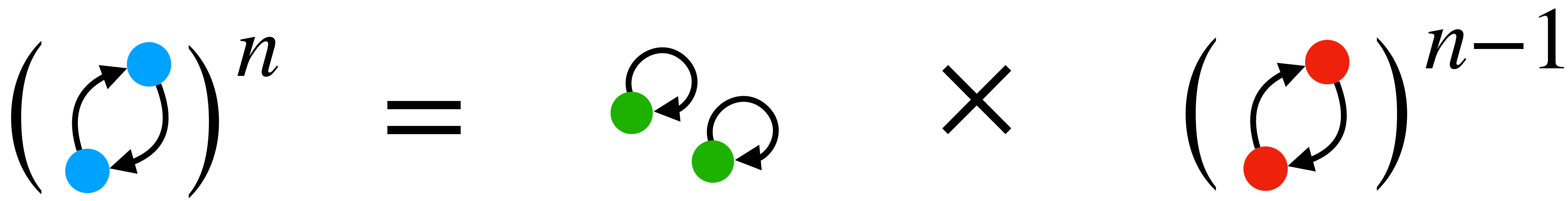
Theorem

For each n, there exist a dynamical system with at least n factorisations



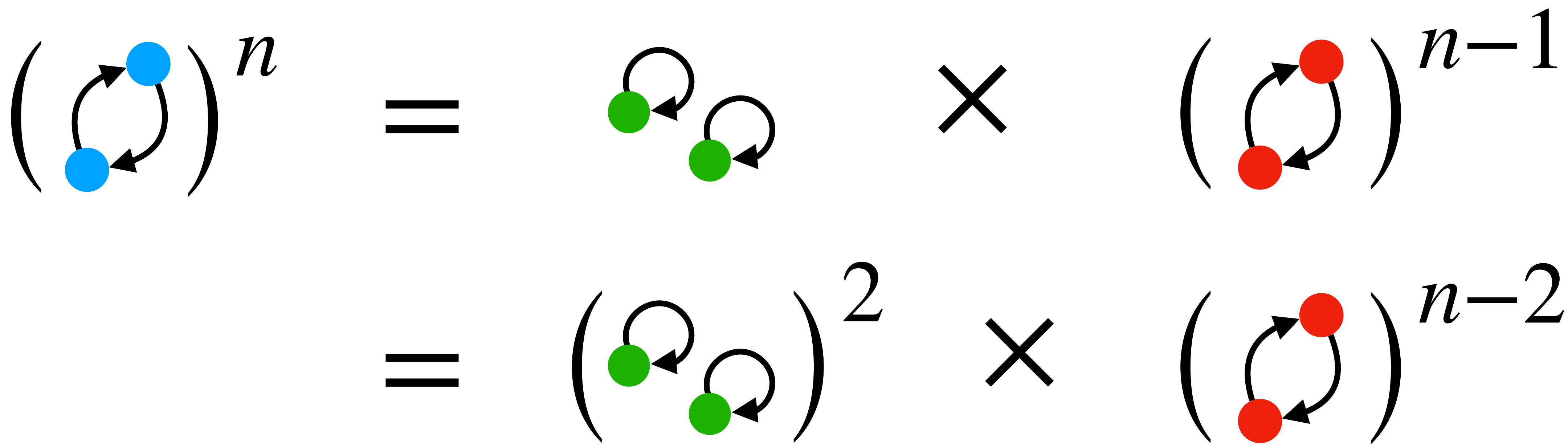
Theorem

For each n, there exist a dynamical system with at least n factorisations



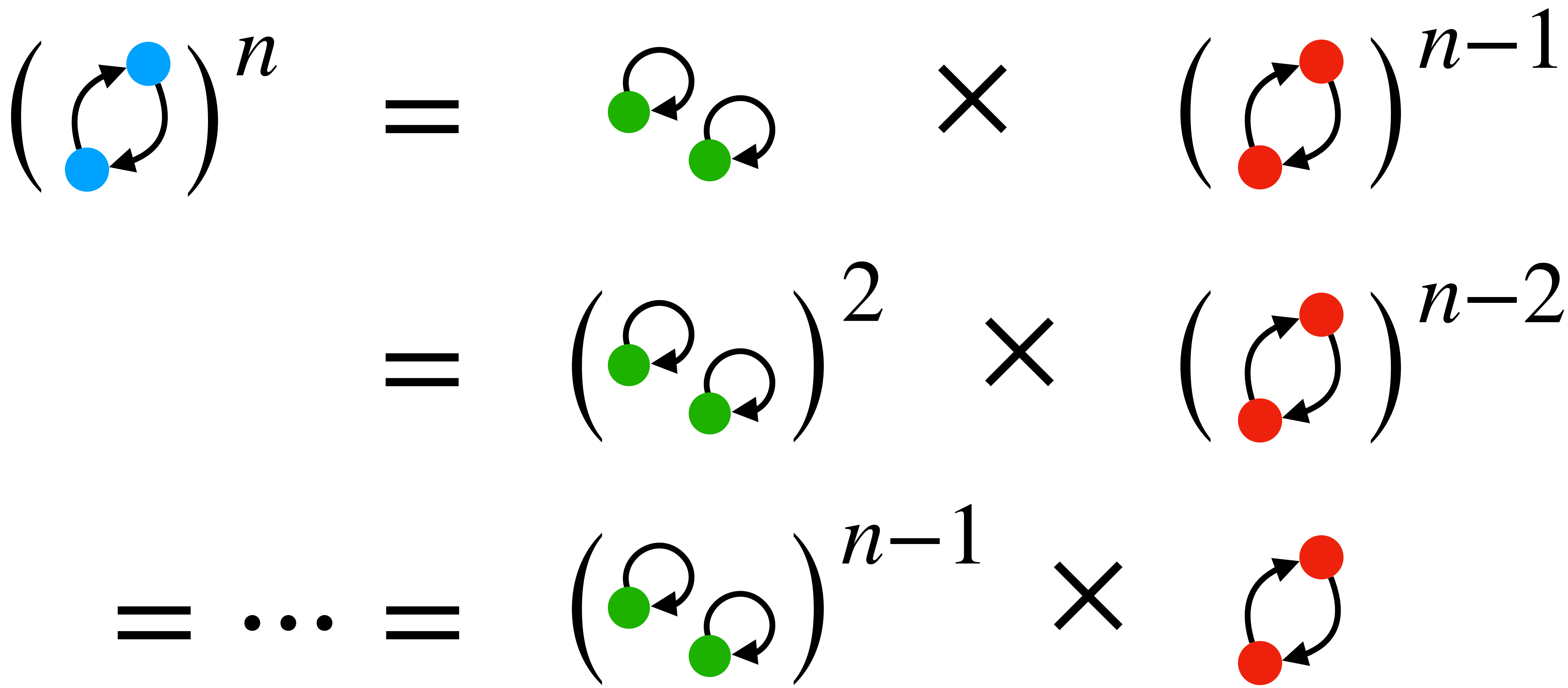
Theorem

For each n, there exist a dynamical system with at least n factorisations



Theorem

For each n, there exist a dynamical system with at least n factorisations



A notable subsemiring

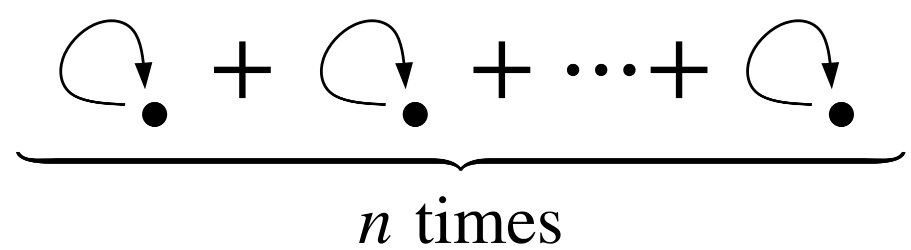
$\mathbb{N}$ is a subsemiring of $\mathbf{D}$

This means trouble

- $\mathbb{N}$ is initial in the category of semirings

- Meaning that there is only one homomorphism $\varphi : \mathbb{N} \to \mathbf{D}$

$$\varphi(n) = \underbrace{\mathbf{1} + \mathbf{1} + \cdots + \mathbf{1}}_{n \text{ times}} =$$



- In the case of $\mathbf{D}$, the homomorphism is injective, since $(\mathbf{D}, +)$ is the free monoid over connected, nonempty dynamical systems

- So $\mathbf{D}$ contains a isomorphic copy of $\mathbb{N}$

A bit more algebra, of the linear kind

$\mathbf{D}$ is a $\mathbb{N}$-semimodule

Like a vector space, but over a semiring

- Here the vectors are dynamical systems and the scalars are naturals
- Trivial because the semimodule axioms are a consequence of $\mathbb{N}$ being a subsemiring of $\mathbf{D}$:

$$n(A + B) = nA + nB \qquad (m + n)A = mA + nA$$

$$(mn)A = m(nA) \qquad 1A = A \qquad 0A = n\mathbf{0} = \mathbf{0}$$

- $\mathbf{D}$ as a semimodule has a unique, countably infinite basis consisting of all nonempty, connected dynamical systems
- The fact that $\mathbf{D}$ is a semimodule will be useful later

Irreducible systems

Most dynamical systems are irreducible

A is irreducible iff $A = BC$ implies $B = \mathbf{1}$ or $C = \mathbf{1}$

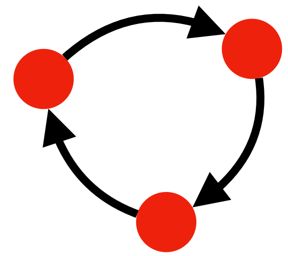
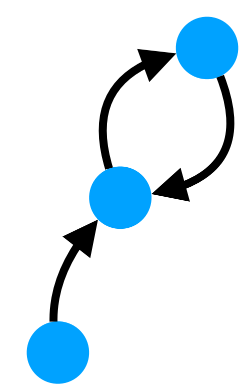
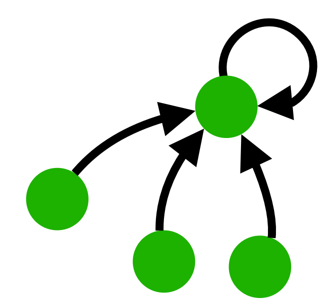
- Formally: $$\lim_{n\to\infty} \frac{\text{number of reducible systems over} \leq n \text{ states}}{\text{total number of systems over} \leq n \text{ states}} = 0$$
- The total number of systems over exactly n states is asymptotically $\eta \frac{\alpha^n}{\sqrt{n}}$, with $\eta \approx 0.443$ and $\alpha \approx 2.956$
- A reducible system over n states is the product of two systems with p and q states such that $pq = n$
- With a few summations and upper bounds, we get the result
- Notice that this is the opposite of the subsemiring $\mathbb{N}$

Polynomial equations over $\mathbf{D}[X_1, \ldots, X_m]$

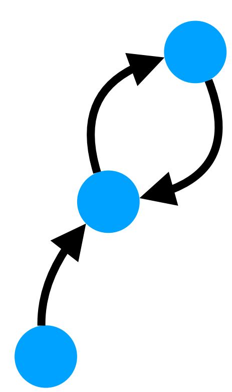
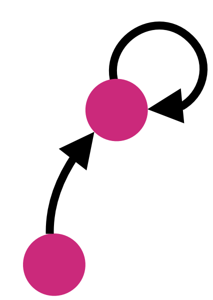
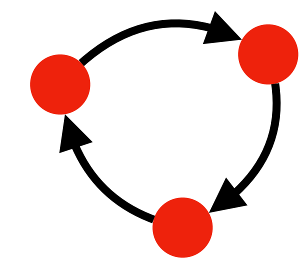
Polynomial equations over $\mathbf{D}[X_1, \ldots, X_m]$

For the analysis of complex systems

- Consider the equation

 $X + Y^2 =$  $Z +$ 

- There is least one solution

$X =$  $Y =$  $Z =$ 

Polynomial equations in semirings

As opposed to rings

- A ring has additive inverses (aka, it has subtraction)
- Each polynomial equation in a ring can be written as $p(\overrightarrow{X}) = 0$
- This is not the case for our semiring, which has no subtraction
- The general polynomial equation has the form $p(\overrightarrow{X}) = q(\overrightarrow{X})$ with two polynomials $p, q \in \mathbf{D}[\overrightarrow{X}]$

Solvability of polynomial equations over $\mathbf{D}$ is undecidable

Undecidability of polynomial equations

The spectre of Hilbert's 10th problem is haunting $\mathbf{D}$

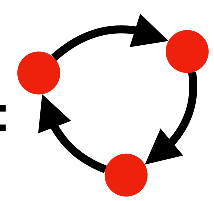
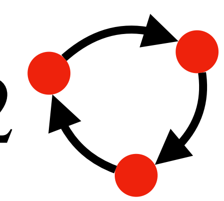
- We have showed that $\mathbb{N}$ is a subsemiring of $\mathbf{D}$
- But sometimes enlarging the solution space makes the problem actually easier: given $p, q \in \mathbb{N}[\overrightarrow{X}]$
  - Finding if $p(\overrightarrow{X}) = q(\overrightarrow{X})$ has solution in $\mathbb{N}$ is undecidable
  - Finding if $p(\overrightarrow{X}) = q(\overrightarrow{X})$ has solution in $\mathbb{R}$ is decidable
  - Finding if $p(\overrightarrow{X}) = q(\overrightarrow{X})$ has solution in $\mathbb{C}$ is trivial
- So, what about finding solutions in $\mathbf{D}$?

Natural polynomial equations

With non-natural solutions

- Let $p(X, Y) = 2X^2$ and $q(X, Y) = 3Y$ with $p, q \in \mathbb{N}[X, Y] \leq \mathbf{D}[X, Y]$

- Then $2X^2 = 3Y$ has the non-natural solution

$X =$  $Y = 2$ 

- But, of course, it also has the natural solution $X' = 3$, $Y' = 6$

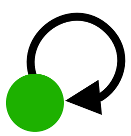
- Notice how $X' = |X|$ and $Y' = |Y|$

- This is not a coincidence!

The function “size” $|\cdot|: \mathbf{D} \to \mathbb{N}$

It’s a semiring homomorphism

- $|\varnothing| = 0$

- |  | $= 1$

- Since $+$ is the disjoint union, we have

$$|A + B| = |A| + |B|$$

- Since $\times$ is the cartesian product, we have

$$|AB| = |A| \times |B|$$

Notation for polynomials $p \in \mathbf{D}[\overrightarrow{X}]$

Of degree $\leq d$ over the variables $\overrightarrow{X} = (X_1, \ldots, X_k)$

$$p = \sum_{\vec{i} \in \{0, \ldots, d\}^k} a_{\vec{i}} \overrightarrow{X}^{\vec{i}}$$

where $\overrightarrow{X}^{\vec{i}} = \prod_{j=1}^{k} X_j^{i_j}$

Notation for polynomials $p \in \mathbf{D}[\overrightarrow{X}]$

Of degree $\leq d$ over the variables $\overrightarrow{X} = (X_1, \ldots, X_k)$

$$p = \sum_{\vec{i} \in \{0,\ldots,d\}^k} a_{\vec{i}} \overrightarrow{X}^{\vec{i}}$$

where $\overrightarrow{X}^{\vec{i}} = \prod_{j=1}^{k} X_j^{i_j}$

for instance $(X, Y, X)^{(2,4,3)} = X^2 Y^4 Z^3$

Theorem

Solvability of natural equations

- If a polynomial equation over $\mathbb{N}[X_1, \ldots, X_k]$ has a solution in $\mathbf{D}^k$, then it also has a solution in $\mathbb{N}^k$
- In the larger semiring $\mathbf{D}$ we may find extra solutions, but only if the equation is already solvable over the naturals
- Then, by reduction from Hilbert's 10th problem, we obtain the undecidability in $\mathbf{D}$ of equations over $\mathbb{N}[\overrightarrow{X}]$…
- …and thus of arbitrary equations over $\mathbf{D}[\overrightarrow{X}]$

Proof

Consider $p(\overrightarrow{X}) = q(\overrightarrow{X})$ **with** $p, q \in \mathbb{N}[\overrightarrow{X}]$

$$\sum_{i \in \{0,\ldots,d\}^k} a_{\vec{i}} \overrightarrow{X}^{\vec{i}} = \sum_{i \in \{0,\ldots,d\}^k} b_{\vec{i}} \overrightarrow{X}^{\vec{i}}$$

Proof

Suppose that $\overrightarrow{A} \in \mathbf{D}^k$ is a solution

$$\sum_{i \in \{0,\ldots,d\}^k} a_{\vec{i}} \overrightarrow{A}^{\vec{i}} = \sum_{i \in \{0,\ldots,d\}^k} b_{\vec{i}} \overrightarrow{A}^{\vec{i}}$$

Proof

Apply the size function $|\cdot|$

$$\left| \sum_{i \in \{0,\ldots,d\}^k} a_{\vec{i}} \overrightarrow{A}^{\vec{i}} \right| = \left| \sum_{i \in \{0,\ldots,d\}^k} b_{\vec{i}} \overrightarrow{A}^{\vec{i}} \right|$$

Proof

The size function $|\cdot|$ is a homomorphism

$$\sum_{\vec{i}\in\{0,\ldots,d\}^k} \left| a_{\vec{i}} \overrightarrow{A}^{\vec{i}} \right| = \sum_{\vec{i}\in\{0,\ldots,d\}^k} \left| b_{\vec{i}} \overrightarrow{A}^{\vec{i}} \right|$$

Proof

The size function $|\cdot|$ is a homomorphism

$$\sum_{i \in \{0,\ldots,d\}^k} |a_{\vec{i}}| \, |\overrightarrow{A}^{\vec{i}}| = \sum_{i \in \{0,\ldots,d\}^k} |b_{\vec{i}}| \, |\overrightarrow{A}^{\vec{i}}|$$

Proof

The coefficients are natural

$$\sum_{i \in \{0,\ldots,d\}^k} a_{\vec{i}} \, | \overrightarrow{A}^{\vec{i}} | = \sum_{i \in \{0,\ldots,d\}^k} b_{\vec{i}} \, | \overrightarrow{A}^{\vec{i}} |$$

Proof

We have $\overrightarrow{A}^{\vec{i}} = \prod_{j=1}^{k} A_j^{i_j}$

$$\sum_{i \in \{0,\ldots,d\}^k} a_{\vec{i}} \left| \prod_{j=1}^{k} A_j^{i_j} \right| = \sum_{i \in \{0,\ldots,d\}^k} b_{\vec{i}} \left| \prod_{j=1}^{k} A_j^{i_j} \right|$$

Proof

The size function $|\cdot|$ is a homomorphism

$$\sum_{i \in \{0,\ldots,d\}^k} a_{\vec{i}} \prod_{j=1}^{k} |A_j^{i_j}| = \sum_{i \in \{0,\ldots,d\}^k} b_{\vec{i}} \prod_{j=1}^{k} |A_j^{i_j}|$$

Proof

The size function $|\cdot|$ is a homomorphism

$$\sum_{i \in \{0,\ldots,d\}^k} a_{\vec{i}} \prod_{j=1}^{k} |A_j|^{i_j} = \sum_{i \in \{0,\ldots,d\}^k} b_{\vec{i}} \prod_{j=1}^{k} |A_j|^{i_j}$$

Proof

So $|\overrightarrow{A}| = (|A_1|, \ldots, |A_k|)$ **is also a solution, QED**

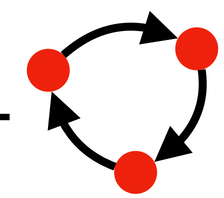
$$p(|A_1|, \ldots, |A_k|) = q(|A_1|, \ldots, |A_k|)$$

Equations with non-natural coefficients

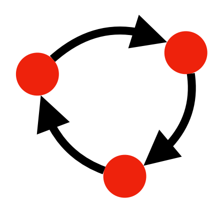
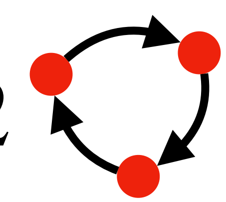
Equations without natural solutions

They do exist

- Consider, for instance

$$X^2 = Y +$$ 

- This equation has solution

$$X =$$  $$Y = 2$$ 

- But there is no natural solution, because the RHS is non-natural and cannot be made natural by adding stuff

Polynomial equations with constant RHS are decidable and in $\mathbf{NP}$

Nondeterministic algorithm

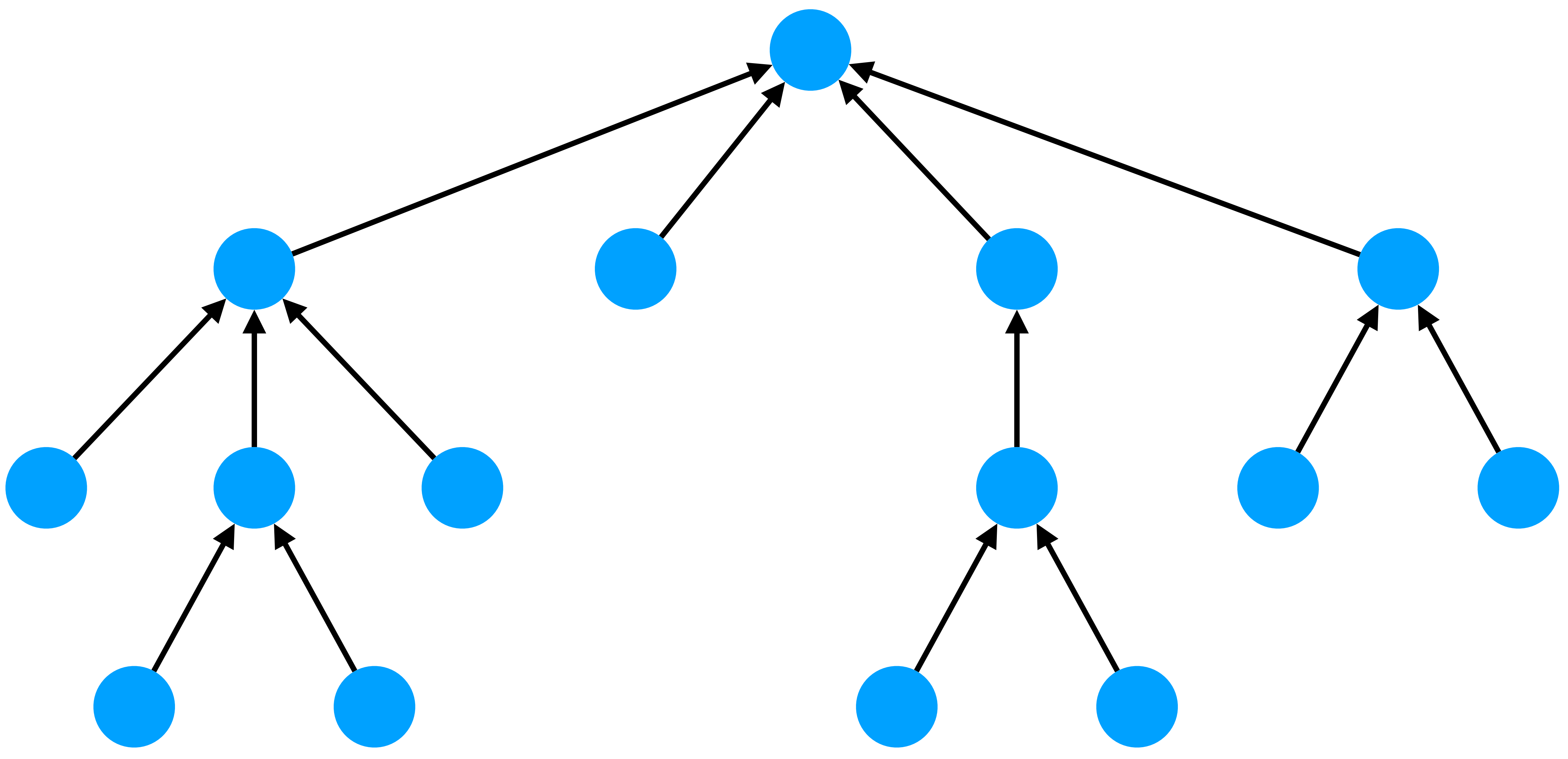
For $p(\overrightarrow{X}) = C$ with $C \in \mathbf{D}$

- Since $+$ and $\times$ are monotonic wrt the sizes of the operands, each X_i in a solution to the equation has size $\leq |C|$
- So it suffices to guess a dynamical system of size $\leq |C|$ for each variable in polynomial time, then calculate LHS
- Finally we check whether LHS and RHS are isomorphic, exploiting the fact that graph isomorphism is in $\mathbf{NP}$
- Only one caveat: if at any time during the calculations the LHS becomes larger than $|C|$, we halt and reject (otherwise the algorithm might take exponential time)

Isomorphism of dynamical systems in polynomial time

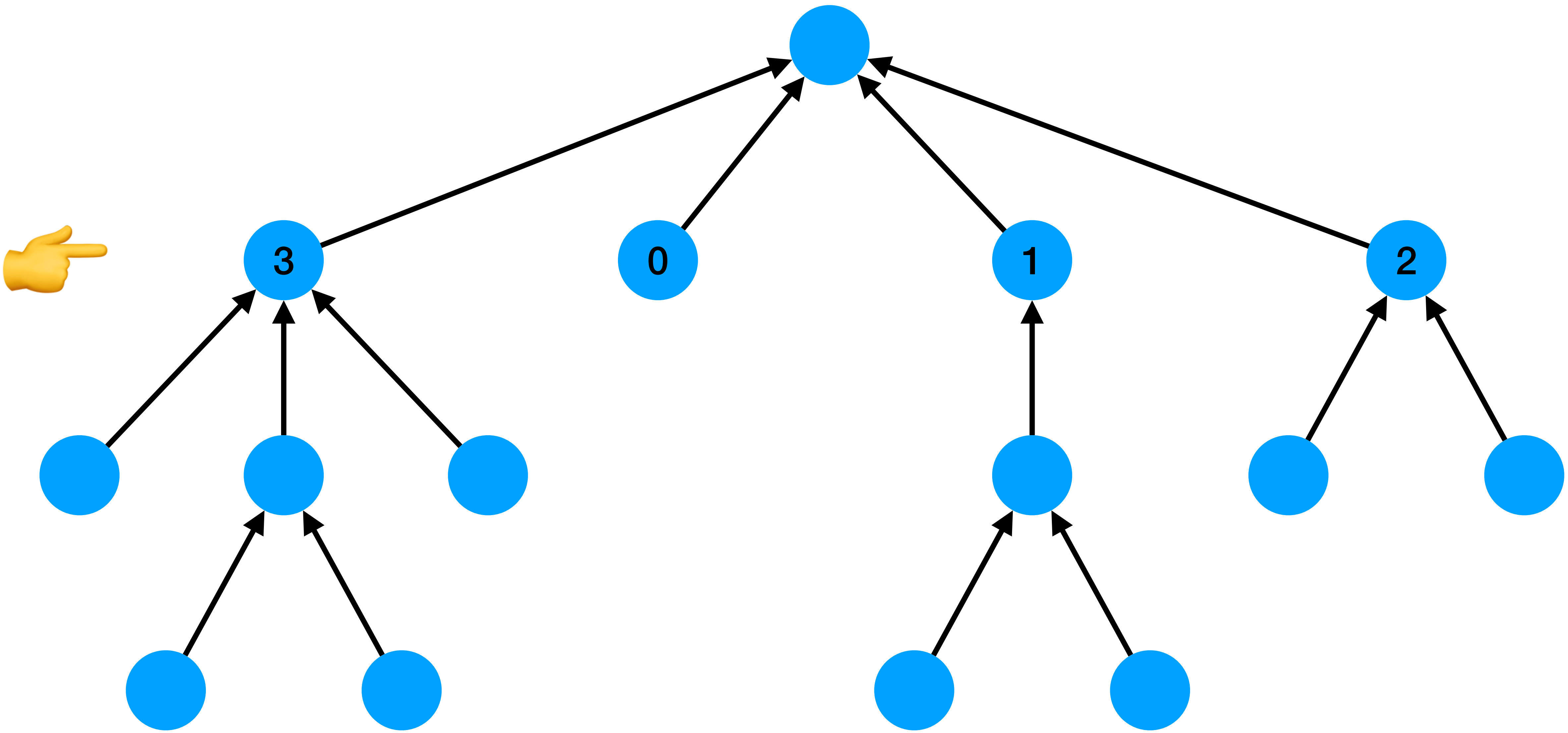
Tree canonisation

A polynomial-time algorithm



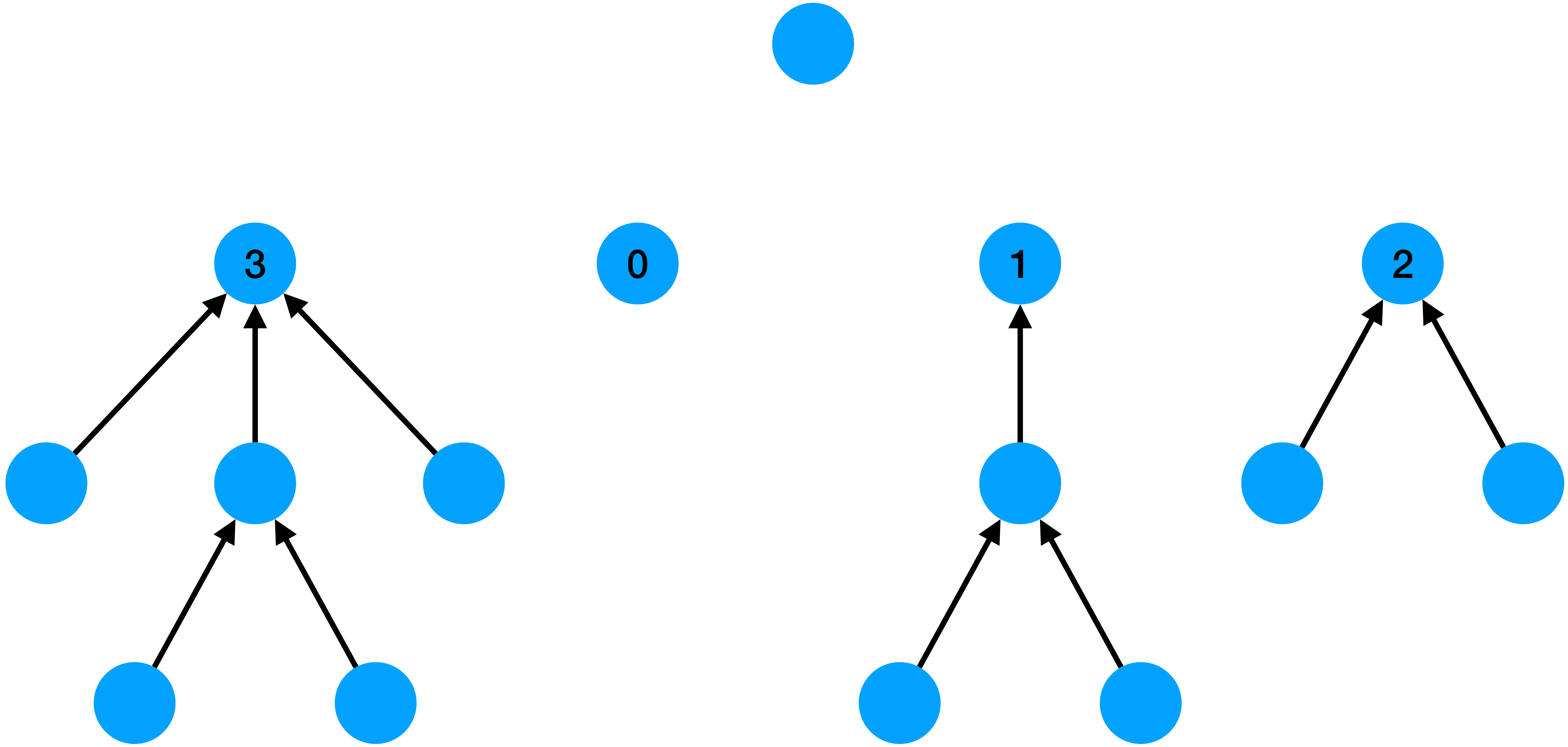
Tree canonisation

A polynomial-time algorithm



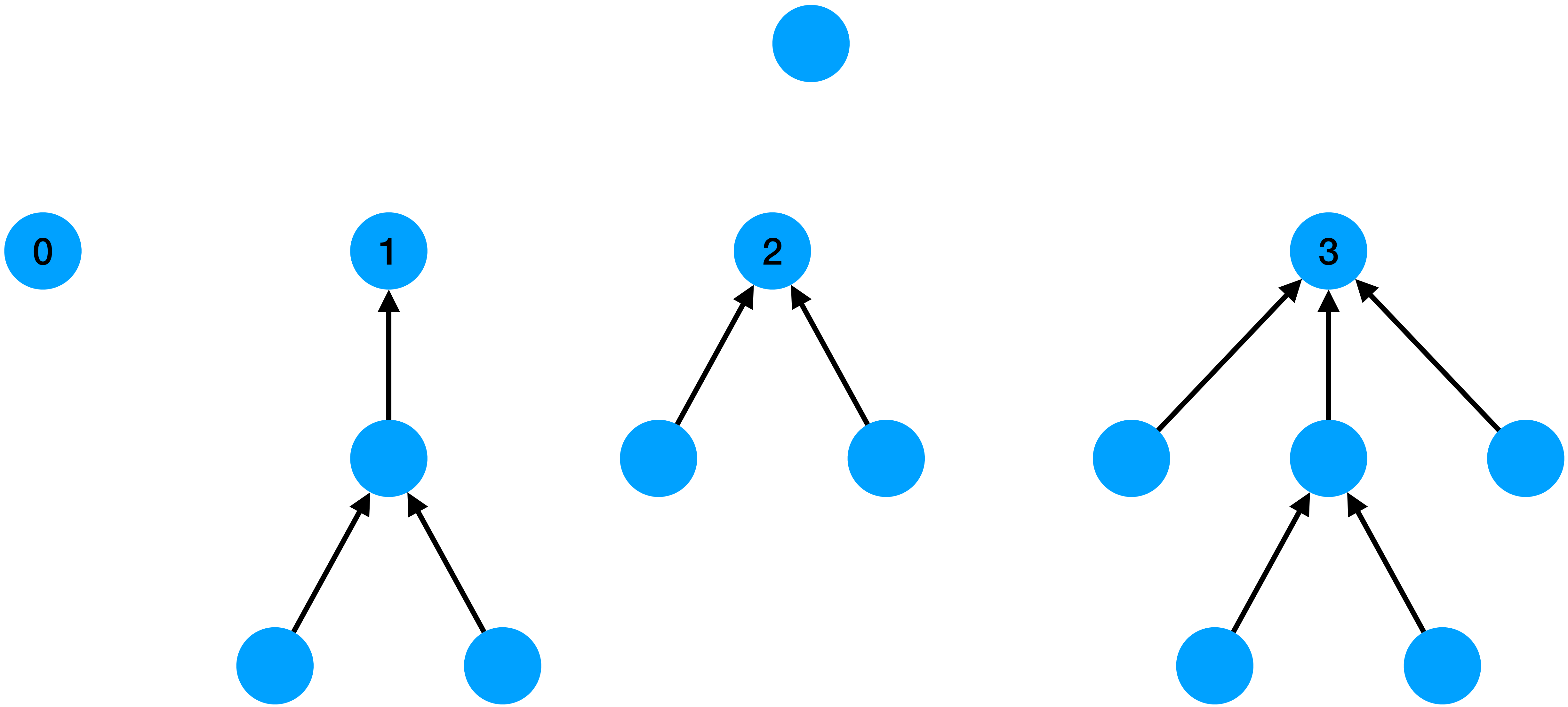
Tree canonisation

A polynomial-time algorithm



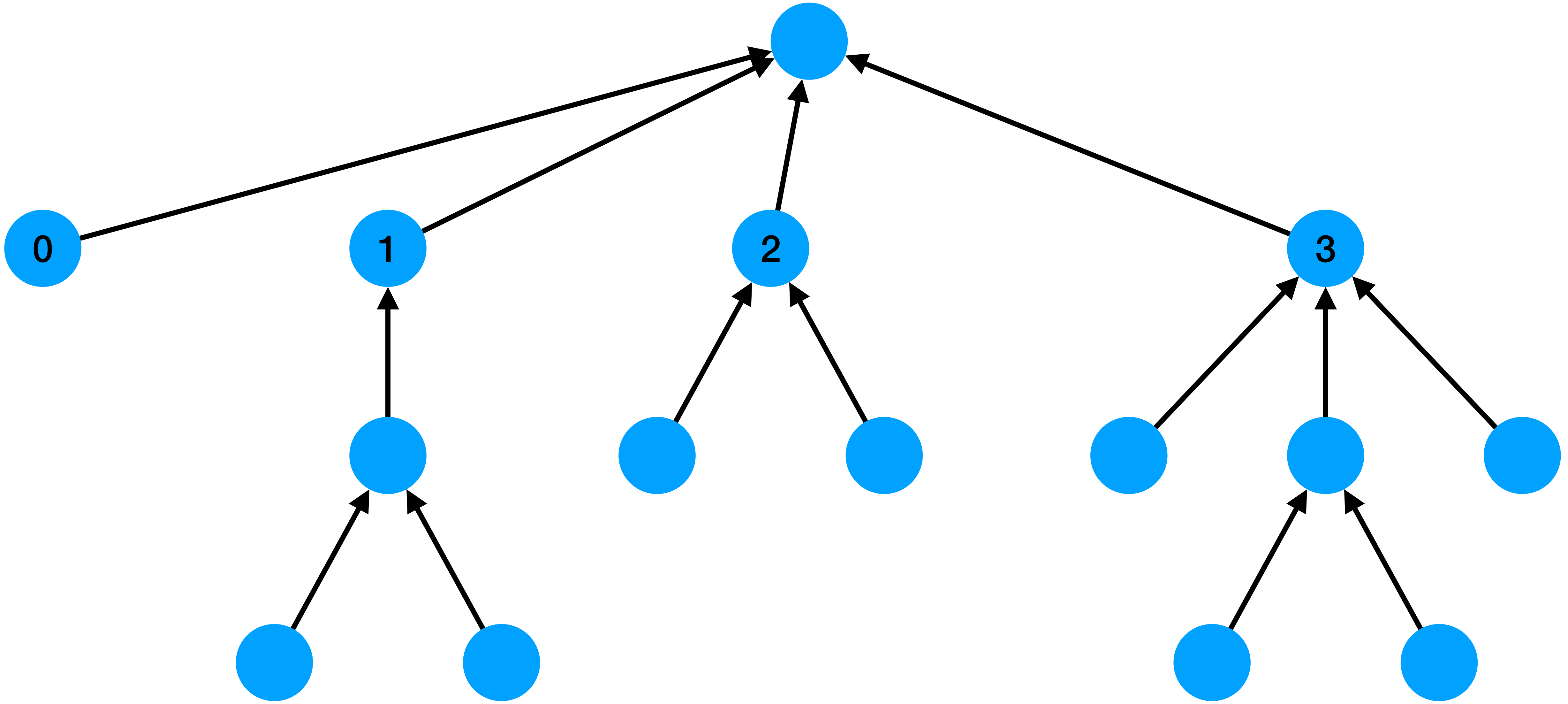
Tree canonisation

A polynomial-time algorithm



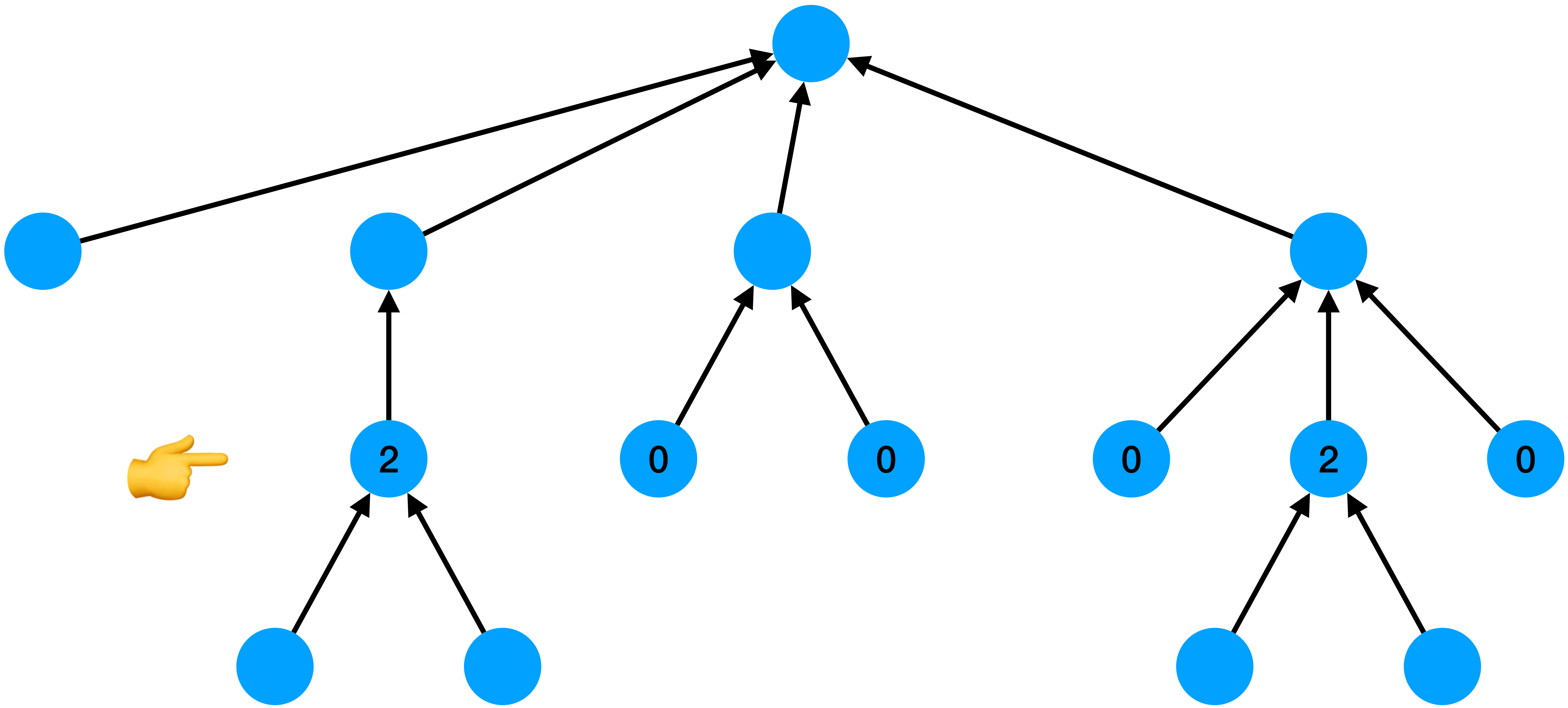
Tree canonisation

A polynomial-time algorithm



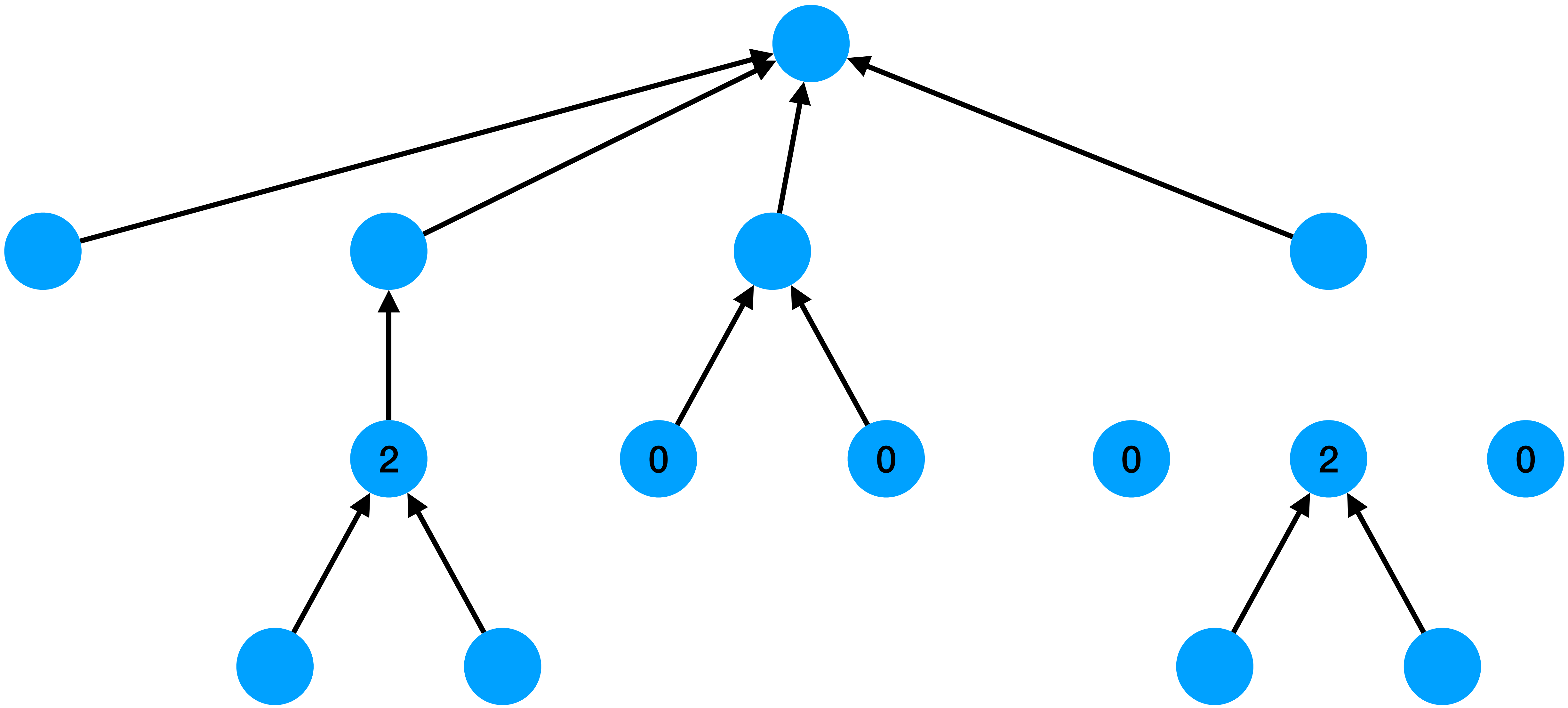
Tree canonisation

A polynomial-time algorithm



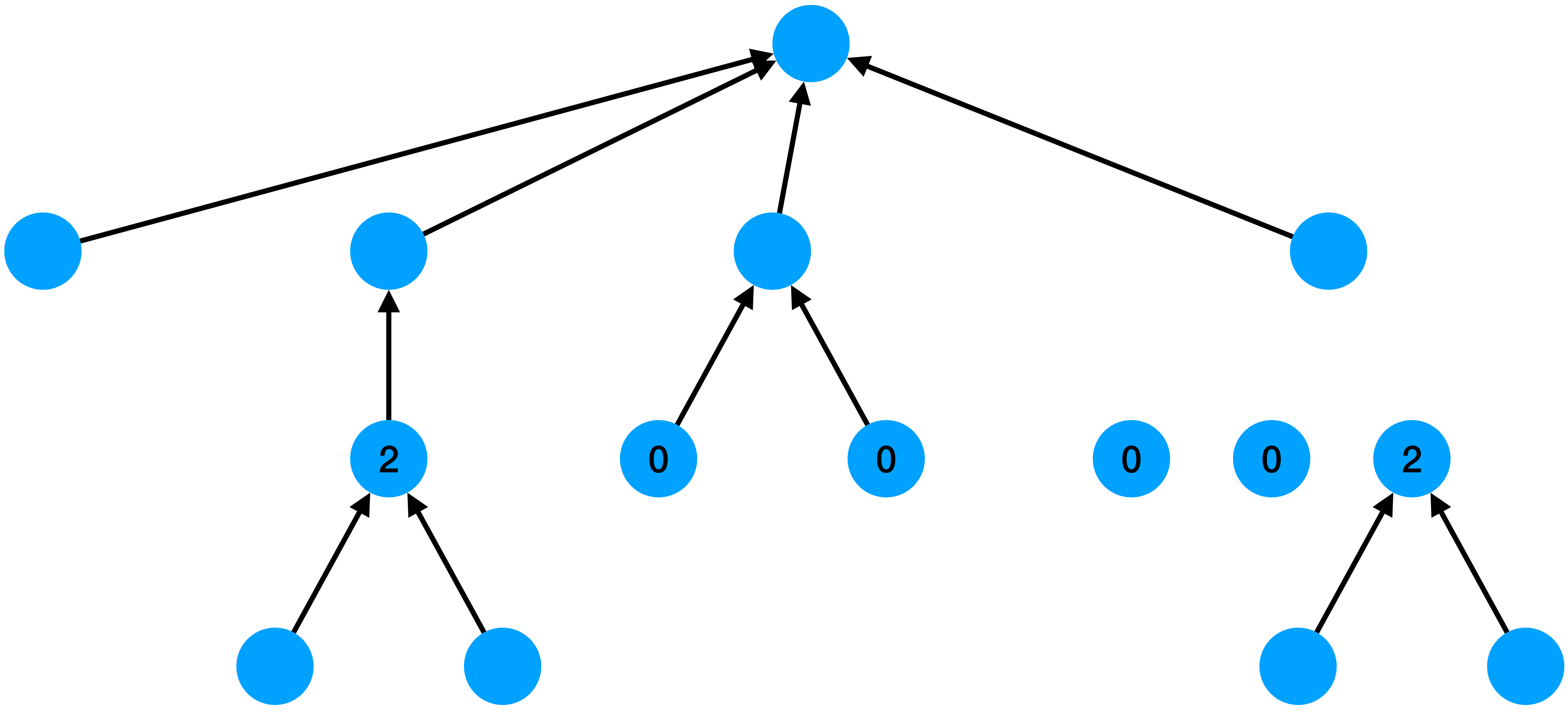
Tree canonisation

A polynomial-time algorithm



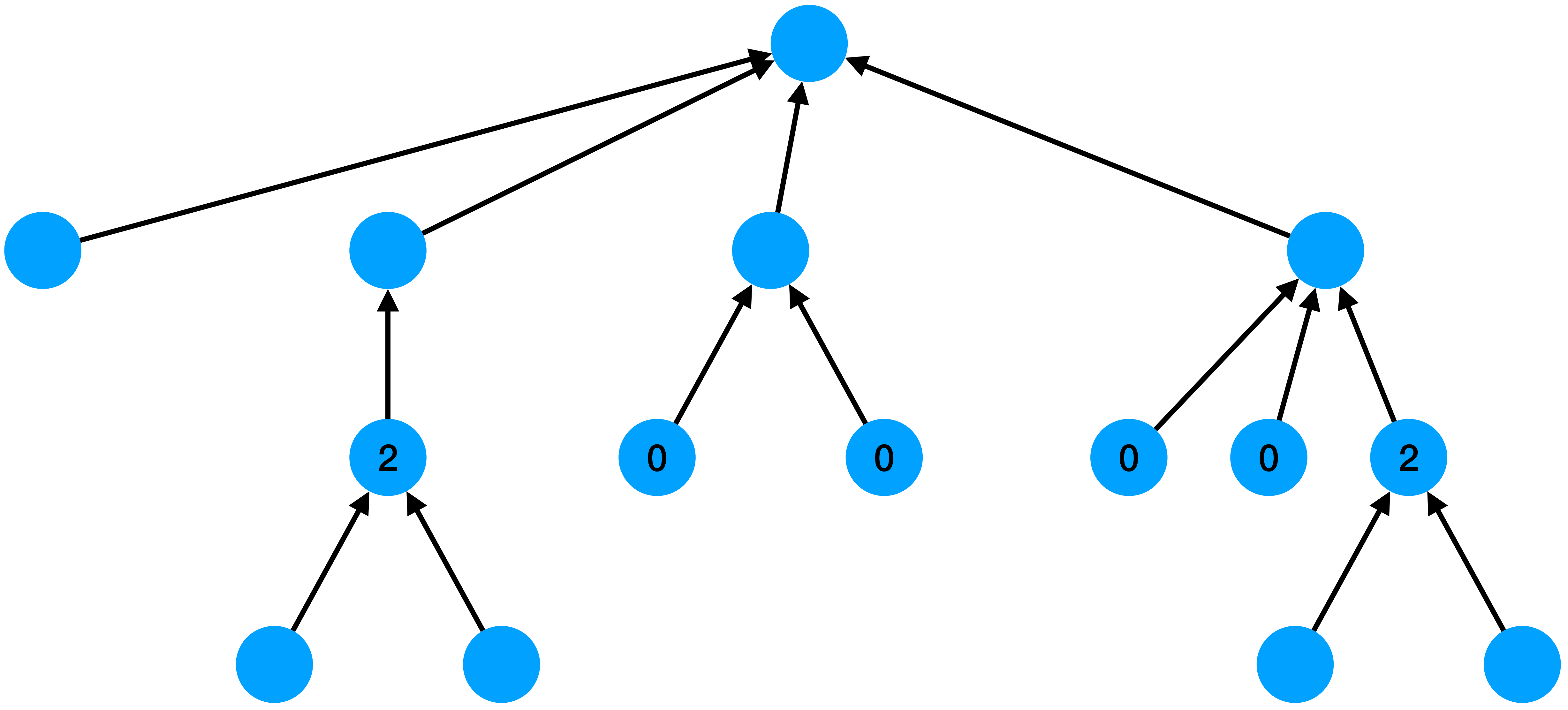
Tree canonisation

A polynomial-time algorithm



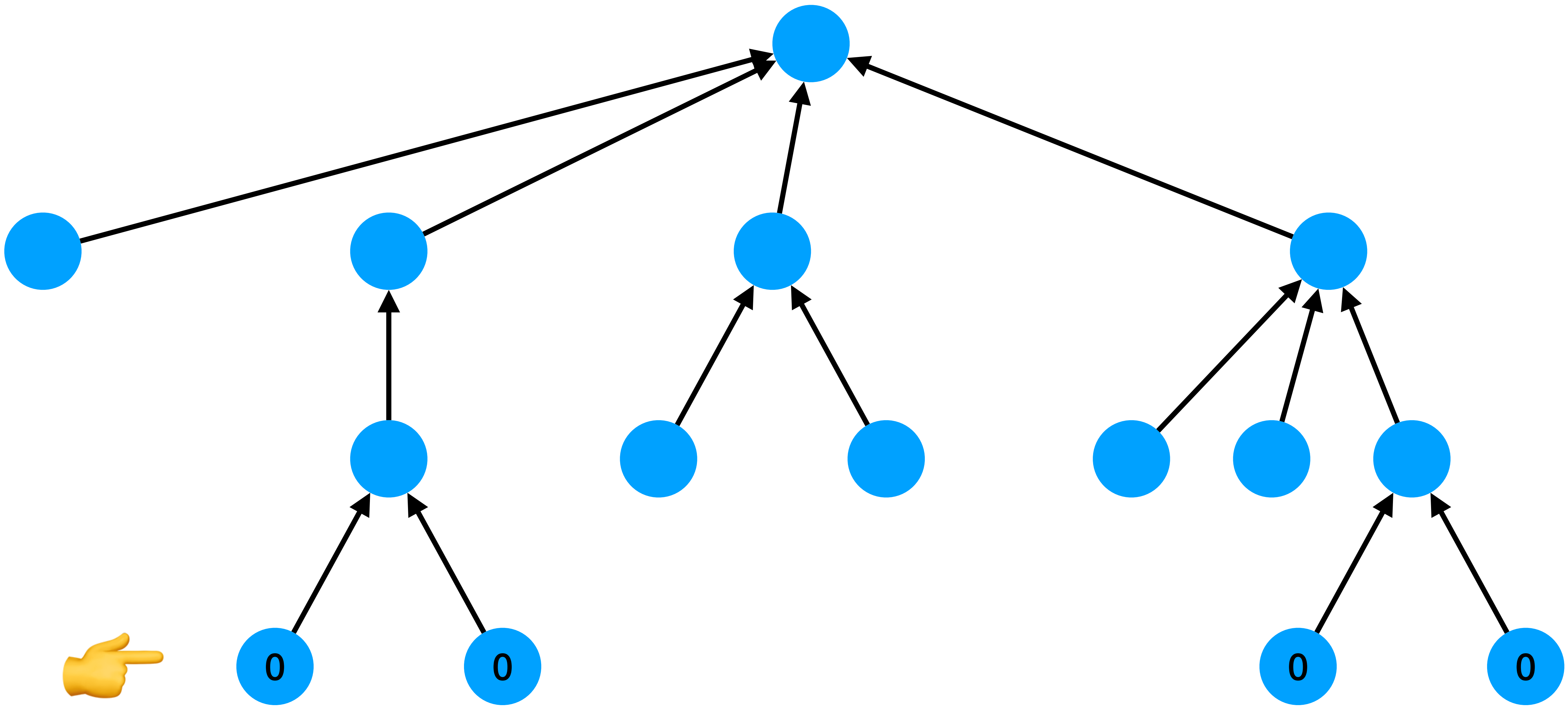
Tree canonisation

A polynomial-time algorithm



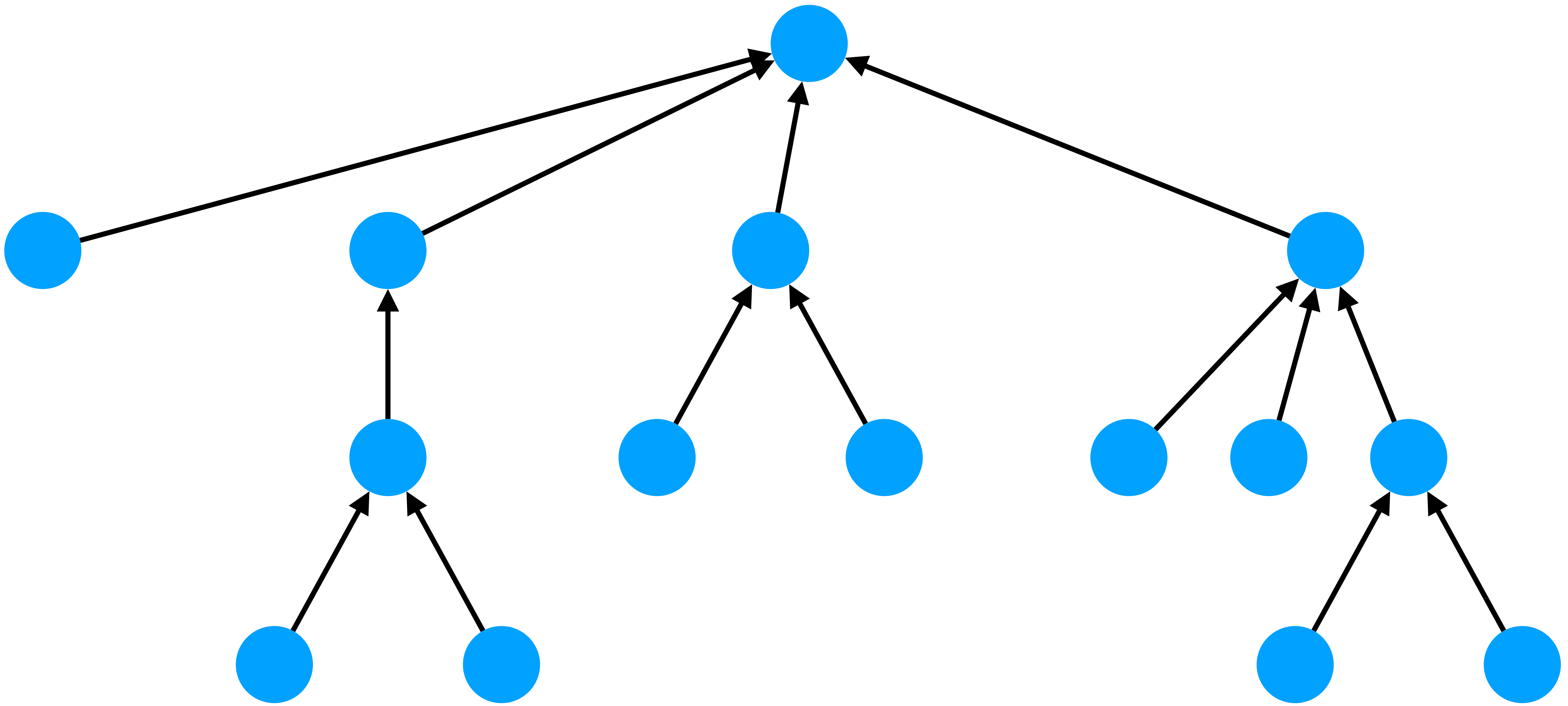
Tree canonisation

A polynomial-time algorithm



Tree canonisation

A polynomial-time algorithm



Connected dynamical system isomorphism

Another polynomial-time algorithm

- **if** the systems have cycles of different length **then return** false
- let T_A and T_B be the sequences of trees of the two systems
- **for each** rotation R of T_B **do**
    - compare R and T_A elementwise in order
    - **if** each pair of trees is isomorphic **then return** true
- **return** false

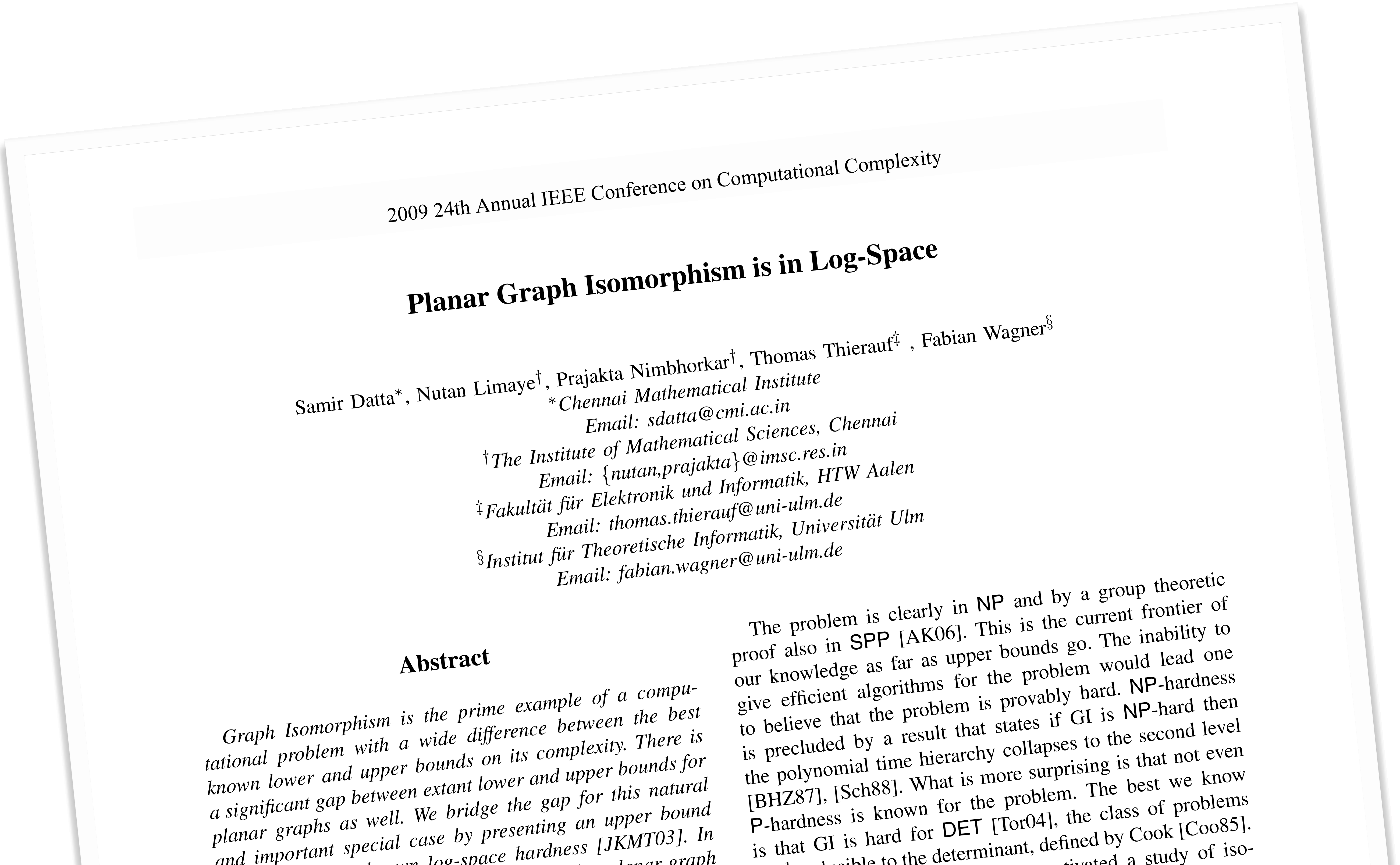
General dynamical system isomorphism

It can also be done in polynomial time

- A dynamical system is a multiset of connected dynamical systems (more about this later…)
- Checking multiset equality can be done naively with a quadratic number of element comparisons
- And we’ve seen that each comparison can be done in polynomial time
- This means that the semiring of dynamical systems is different from a more general semiring of graphs (nondeterministic dynamical systems),
where the isomorphism problem is presumably hard

Dynamical system isomorphism

Even easier than that!



2009 24th Annual IEEE Conference on Computational Complexity

Planar Graph Isomorphism is in Log-Space

Samir Datta*, Nutan Limaye†, Prajakta Nimbhorkar†, Thomas Thierauf‡ , Fabian Wagner§
*Chennai Mathematical Institute
Email: sdatta@cmi.ac.in
†The Institute of Mathematical Sciences, Chennai
Email: {nutan,prajakta}@imsc.res.in
‡Fakultät für Elektronik und Informatik, HTW Aalen
Email: thomas.thierauf@uni-ulm.de
§Institut für Theoretische Informatik, Universität Ulm
Email: fabian.wagner@uni-ulm.de

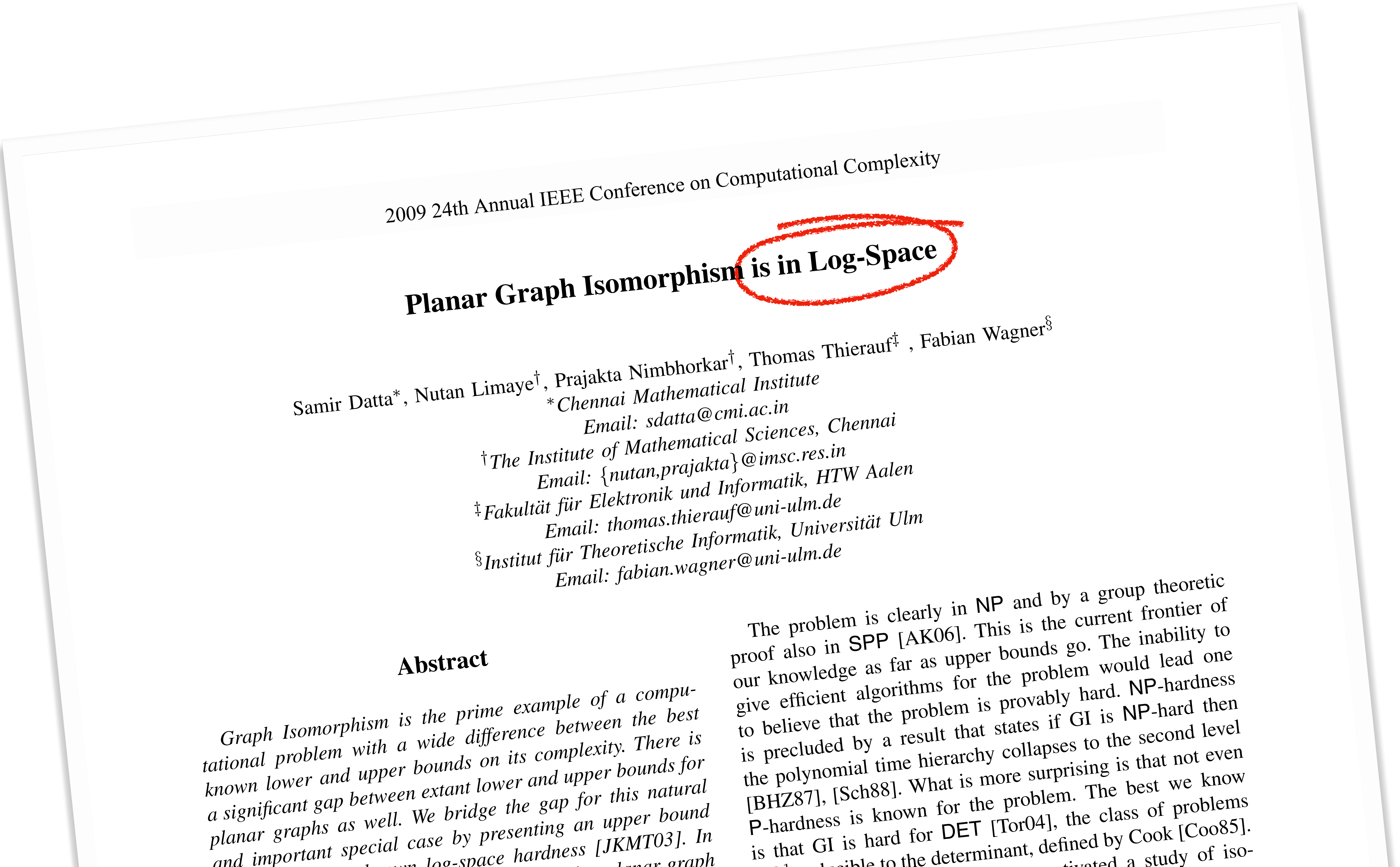
Abstract

Graph Isomorphism is the prime example of a computational problem with a wide difference between the best known lower and upper bounds on its complexity. There is a significant gap between extant lower and upper bounds for planar graphs as well. We bridge the gap for this natural and important special case by presenting an upper bound ... log-space hardness [JKMT03]. In ...

The problem is clearly in NP and by a group theoretic proof also in SPP [AK06]. This is the current frontier of our knowledge as far as upper bounds go. The inability to give efficient algorithms for the problem would lead one to believe that the problem is provably hard. NP-hardness is precluded by a result that states if GI is NP-hard then the polynomial time hierarchy collapses to the second level [BHZ87], [Sch88]. What is more surprising is that not even P-hardness is known for the problem. The best we know is that GI is hard for DET [Tor04], the class of problems ... to the determinant, defined by Cook [Coo85]. ...

Dynamical system isomorphism

Even easier than that!



2009 24th Annual IEEE Conference on Computational Complexity

Planar Graph Isomorphism is in Log-Space

Samir Datta*, Nutan Limaye†, Prajakta Nimbhorkar†, Thomas Thierauf‡, Fabian Wagner§

*Chennai Mathematical Institute
Email: sdatta@cmi.ac.in
†The Institute of Mathematical Sciences, Chennai
Email: {nutan,prajakta}@imsc.res.in
‡Fakultät für Elektronik und Informatik, HTW Aalen
Email: thomas.thierauf@uni-ulm.de
§Institut für Theoretische Informatik, Universität Ulm
Email: fabian.wagner@uni-ulm.de

Abstract

Graph Isomorphism is the prime example of a computational problem with a wide difference between the best known lower and upper bounds on its complexity. There is a significant gap between extant lower and upper bounds for planar graphs as well. We bridge the gap for this natural and important special case by presenting an upper bound ... log-space hardness [JKMT03]. In ...

The problem is clearly in NP and by a group theoretic proof also in SPP [AK06]. This is the current frontier of our knowledge as far as upper bounds go. The inability to give efficient algorithms for the problem would lead one to believe that the problem is provably hard. NP-hardness is precluded by a result that states if GI is NP-hard then the polynomial time hierarchy collapses to the second level [BHZ87], [Sch88]. What is more surprising is that not even P-hardness is known for the problem. The best we know is that GI is hard for DET [Tor04], the class of problems ... to the determinant, defined by Cook [Coo85]. ... a study of iso-

Systems of linear equations with constant RHS are $\mathbf{NP}$-complete

NP-hardness of linear systems

By reduction from One-in-three-3SAT

- Given a 3CNF Boolean formula φ, is there a satisfying assignment such that exactly one literal per clause is true?
- For each variable x of φ we have one equation $X + X' = 1$, forcing one between X and X' to be 1, and the other to be 0
- For each clause, for instance $(x \vee \neg y \vee z)$, we have one equation $X + Y' + Z = 1$, which forces exactly one variable to 1
- These are all linear, constant-RHS equations over $\mathbf{D}[\overrightarrow{X}]$ (actually $\mathbb{N}[\overrightarrow{X}]$), and its solutions are the same as the satisfying assignments of φ with one true literal per clause

A single linear, constant-RHS equation is $\mathbf{NP}$-complete

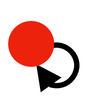
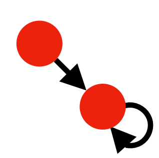
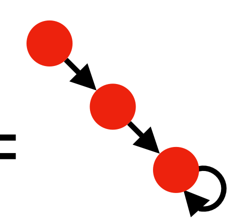
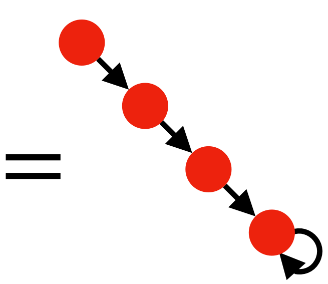
Reducing the system of equations to one

Several $\mathbb{N}[\overrightarrow{X}]$ linear equations to one $\mathbf{D}[\overrightarrow{X}]$ equation

- Let $p_1(\overrightarrow{X}) = 1, \ldots, p_n(\overrightarrow{X}) = 1$ be the previous system of equations, with $p_i \in \mathbb{N}[\overrightarrow{X}]$

- Recall that $\mathbf{D}$ is a $\mathbb{N}$-semimodule with basis all connected systems

- Take any n easy-to-compute, linearly independent systems $e_1, \ldots e_n \in \mathbf{D}$, for instance

$e_1 =$  $e_2 =$  $e_3 =$  $e_4 =$  $\cdots$

- Then the equation $e_1 p_1(\overrightarrow{X}) + \cdots + e_n p_n(\overrightarrow{X}) = e_1 + \cdots + e_n$ is a linear equation over $\mathbf{D}[\overrightarrow{X}]$ having the same solutions as the original system

A more abstract view

Abstracting away from some details

In the hope of making equations easier

- Since the complexity of solving equations over dynamical systems is too high, we want to try finding a suitable algebraic abstraction

- For instance, another semiring R with a surjective homomorphism $\mathbf{D} \to R$ that does not erase too much information

- Hoping that polynomial equations over $R[\overrightarrow{X}]$ might be easier

Profiles of dynamical systems

Definition

Profile of a dynamical system

- Given a dynamical system (A, f) define the infinite sequence

$$\text{prof}(A) = (|A|, |f(A)|, |f^2(A)|, \ldots) = (|f^n(A)| : n \in \mathbb{N})$$

- Clearly, the sequence is decreasing and ultimately constant for finite systems, since sooner or later $f^n(A) = f^{n+1}(A)$

- So we can halt the sequence as soon as it stops decreasing

- Her $f^n(A)$ is the set of periodic states, and the minimum n is the distance of the state farthest away from a limit cycle

The semiring $\mathbb{P}$ of profiles

Let (A, f) and (B, g) be dynamical systems

- We have $\mathrm{prof}(A + B) = (\,|\,(f + g)^n(A \uplus B)\,|\, : n \in \mathbb{N})$
- But $(f + g)(A \uplus B) = f(A) \uplus g(B)$, so $\mathrm{prof}(A + B) = \mathrm{prof}(A) + \mathrm{prof}(B)$ elementwise
- We have $\mathrm{prof}(A \times B) = (\,|\,(f \times g)^n(A \times B)\,|\, : n \in \mathbb{N})$
- But $(f \times g)(A \times B) = f(A) \times g(B)$, so $\mathrm{prof}(A \times B) = \mathrm{prof}(A) \times \mathrm{prof}(B)$ elementwise
- Then the set of profiles inherits a semiring structure from $\mathbb{N}$

Profiles of dynamical systems

Algebraic, computability and complexity questions

- Most algebraic properties remain the same: multiple factorisations, most elements are irreducible
- The equations are, in general, algorithmically unsolvable
- They become solvable with a constant RHS
- But they remain **NP**-complete, even for a single linear equation

Open problems

Open problems

Algebraic ones

- Are there prime elements P, that is, whenever P divides AB it divides either A or B? What do they represent?
  - We know exactly zero prime elements 🤷‍♂️
- Does it make any sense to adjoin the additive inverses in order to obtain a ring?
  - Think about imaginary numbers, using them in intermediary computation steps, but discarding any imaginary solutions
- Is it useful to find nondeterministic dynamical system (i.e., arbitrary graph) solutions to equations?
- Semirings of infinite discrete-time dynamical systems

Open problems

Computability and complexity

- Find larger classes of solvable equations, e.g., by number of variables or degree of the polynomials
  - Do we obtain the same results as for natural numbers?
- The semiring of computably infinite dynamical systems
- Discover classes of equations solvable efficiently
  - Hard for systems in succinct form
- Find out if there exist decidable equations harder than $\mathbf{NP}$
  - It would feel strange to jump from $\mathbf{NP}$ to undecidable

Open problems

Complexity of succinct representations

- Investigate the complexity of problems where a succinct representation of dynamical system is given as input
- Let (A, f) be a dynamical system, and suppose that $A \subseteq \{0,1\}^n$
- A circuit encoding for (A, f) is a pair of circuits (C_A, C_f) where
  - $C_A \colon \{0,1\}^n \to \{0,1\}$ is the characteristic function of A
  - $C_f \colon \{0,1\}^n \to \{0,1\}^n$ is such that $C_f(x) = f(x)$ if $x \in A$
- Easy to construct (even uniformly) circuits for $A + B$ and $A \times B$

Bibliography 📖

Something to read before bed

- A. Dennunzio, V. Dorigatti, E. Formenti, L. Manzoni, A.E. Porreca, Polynomial equations over finite, discrete-time dynamical systems, 13th International Conference on Cellular Automata for Research and Industry, ACRI 2018, https://doi.org/10.1007/978-3-319-99813-8_27

- C. Gaze-Maillot, A.E. Porreca, Profiles of dynamical systems and their algebra, arXiv e-prints 2020, https://arxiv.org/abs/2008.00843

- A. Dennunzio, E. Formenti, L. Margara, V. Montmirail, S. Riva, Solving equations on discrete dynamical systems (extended version), 16th International Conference on Computational Intelligence methods for Bioinformatics and Biostatistics, CIBB 2019, https://arxiv.org/abs/1904.13115

Thanks for your attention!

Merci de votre attention !

Any questions?