The algebra of alternation and synchronisation of finite dynamical systems

Antonio E. Porreca • aeporreca.org
Aix-Marseille Université & LIS

The team

Florian Bridoux

Johan Couturier

Alberto Dennunzio

François Doré

Valentina Dorigatti

Enrico Formenti

Caroline Gaze-Maillot

Luca Manzoni

Émile Naquin

Kévin Perrot

Antonio E. Porreca

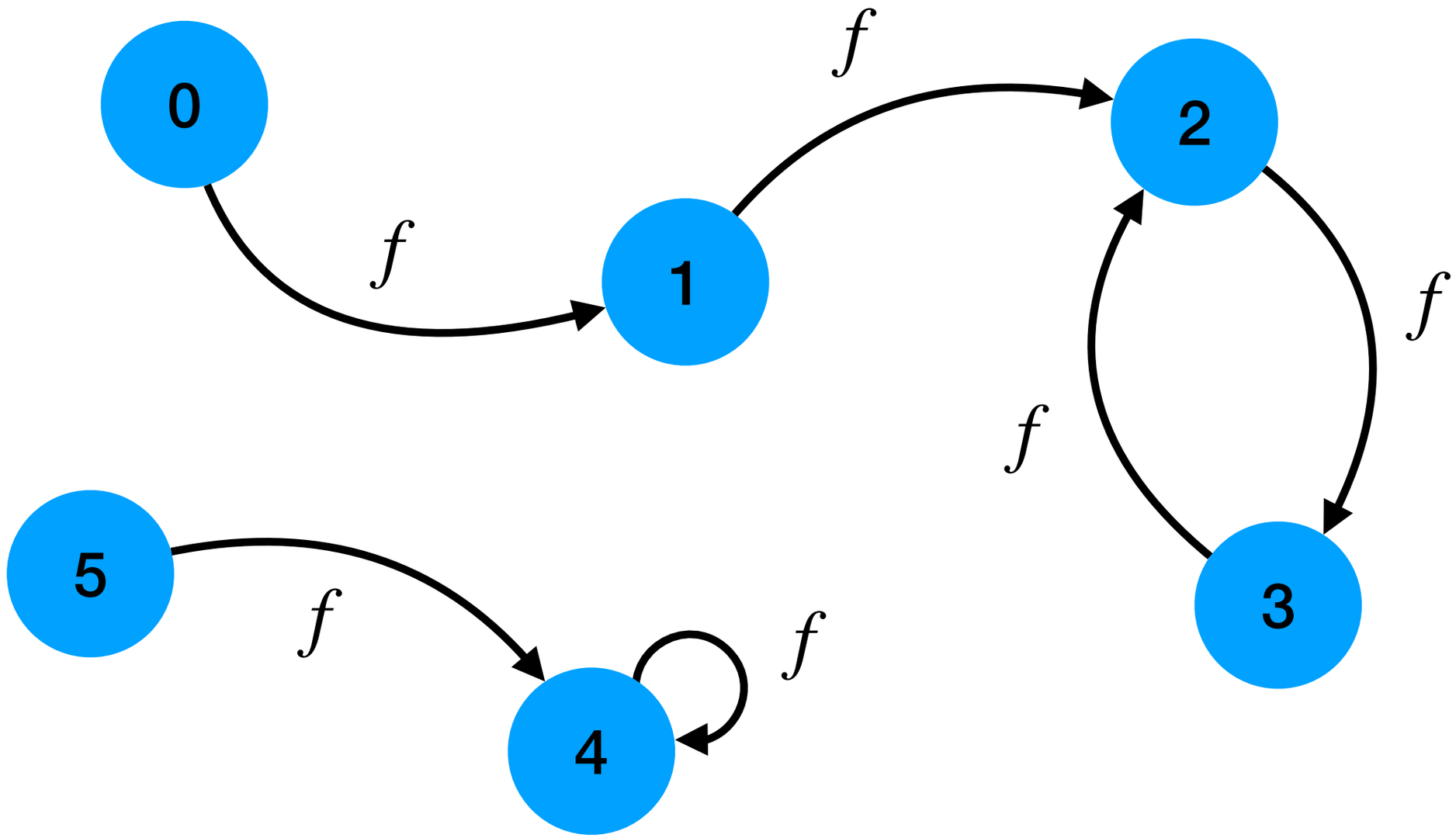
Sara Riva

Marius Rolland

Ekaterina Timofeeva

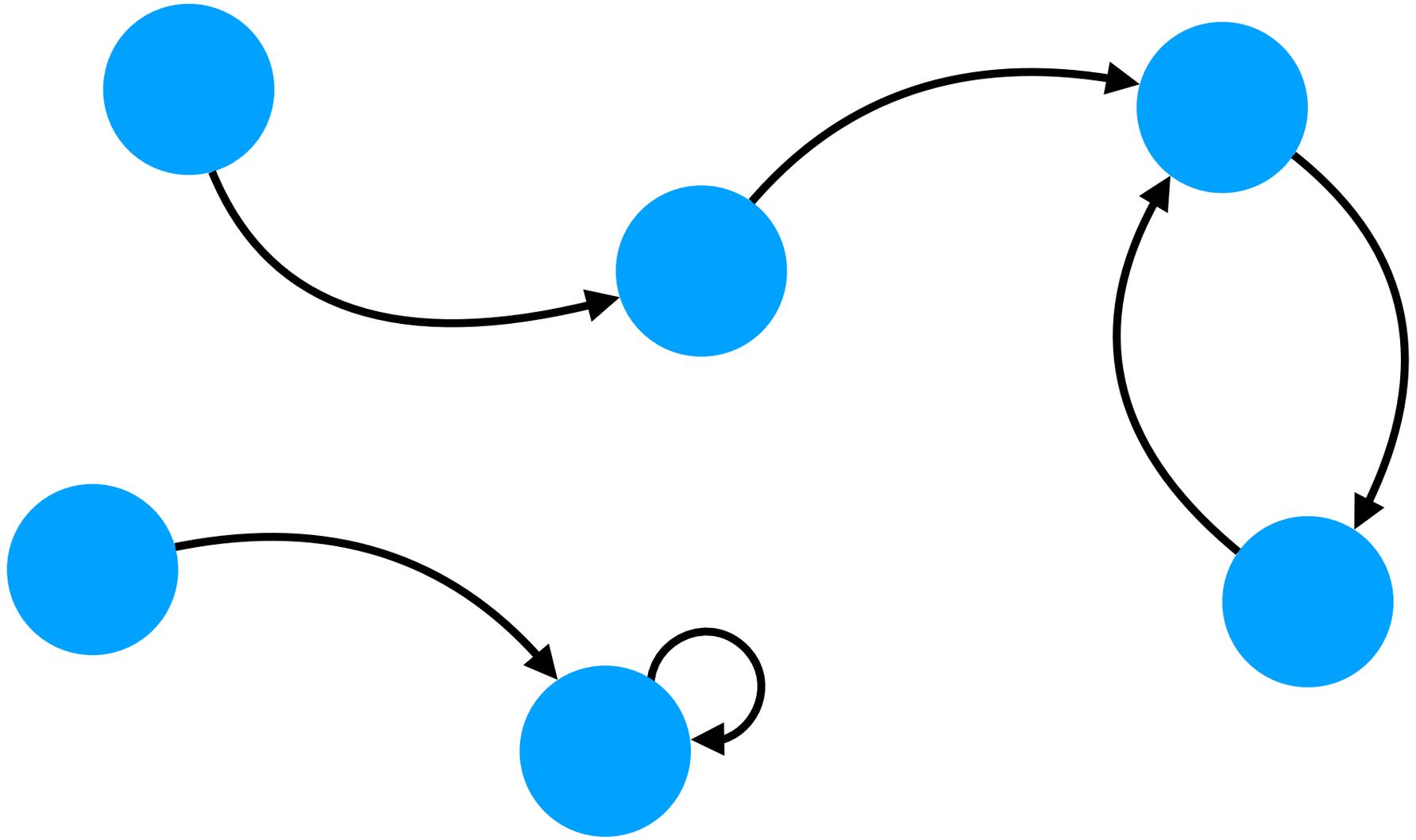
Finite, discrete-time dynamical systems

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



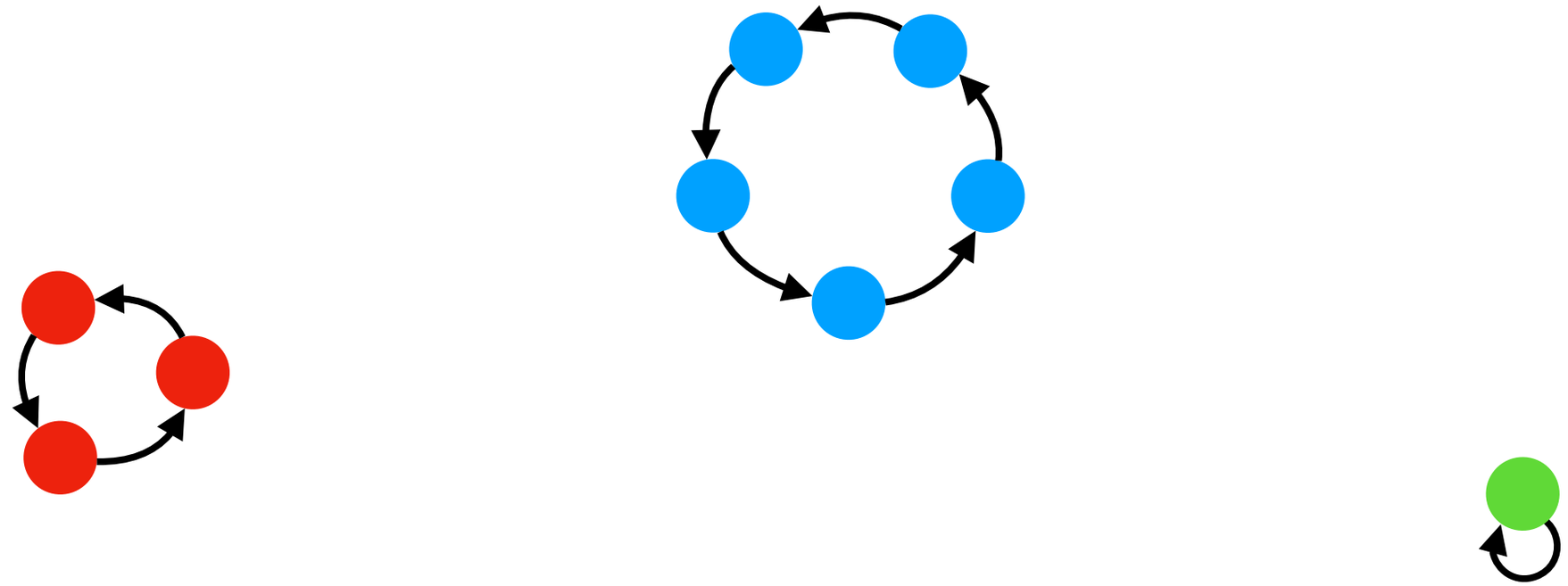
Finite, discrete-time dynamical systems

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



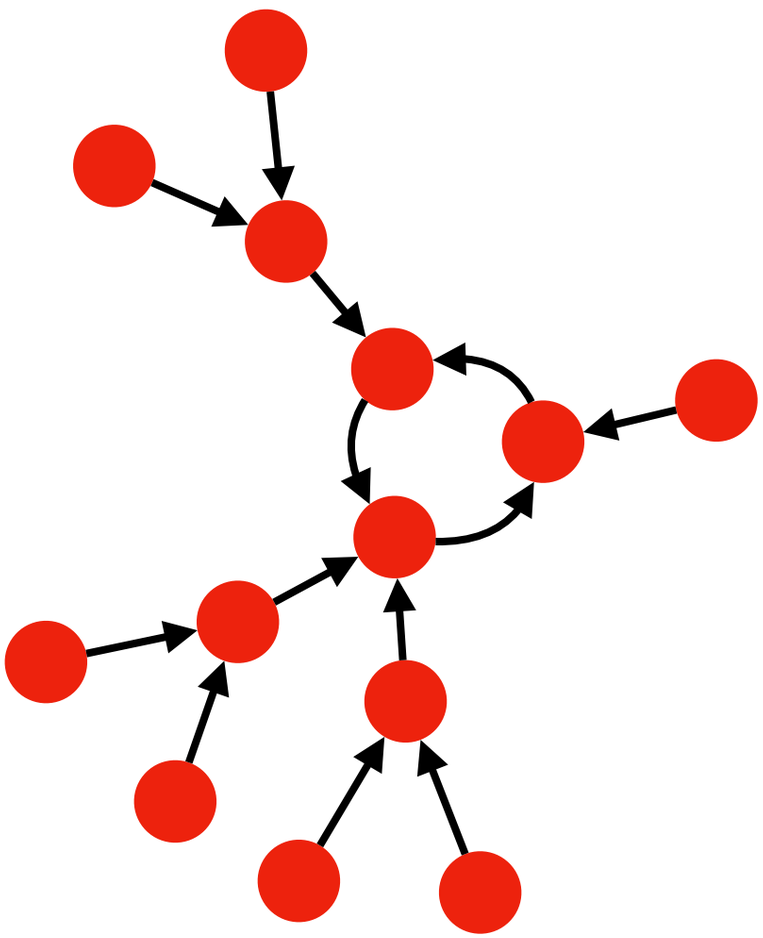
General shape of a dynamical system

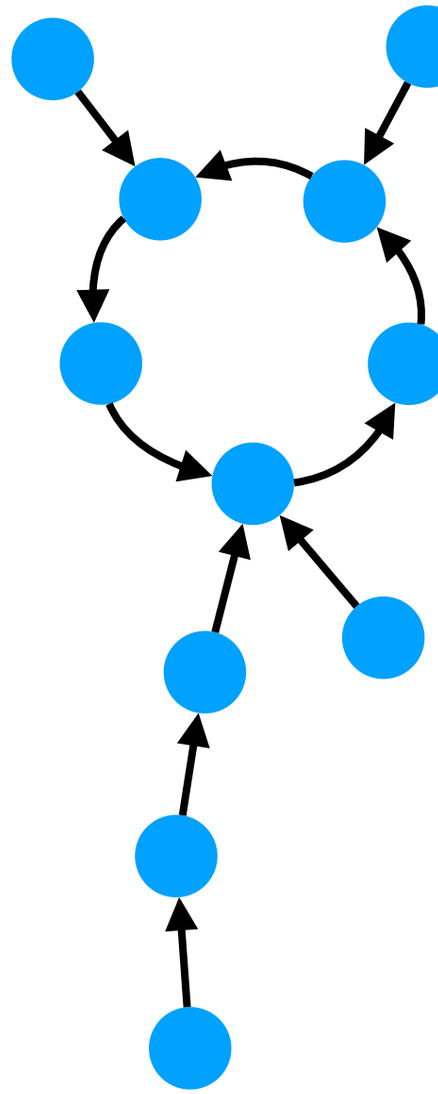
A few limit cycles

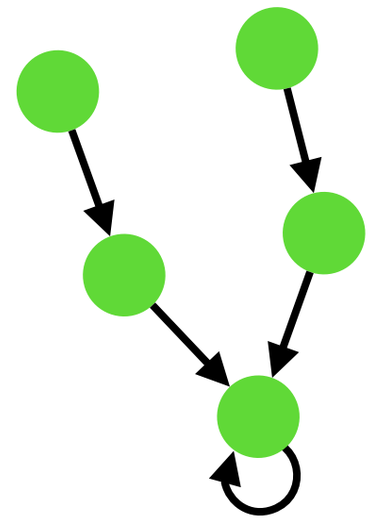


General shape of a dynamical system

A few limit cycles with trees going in

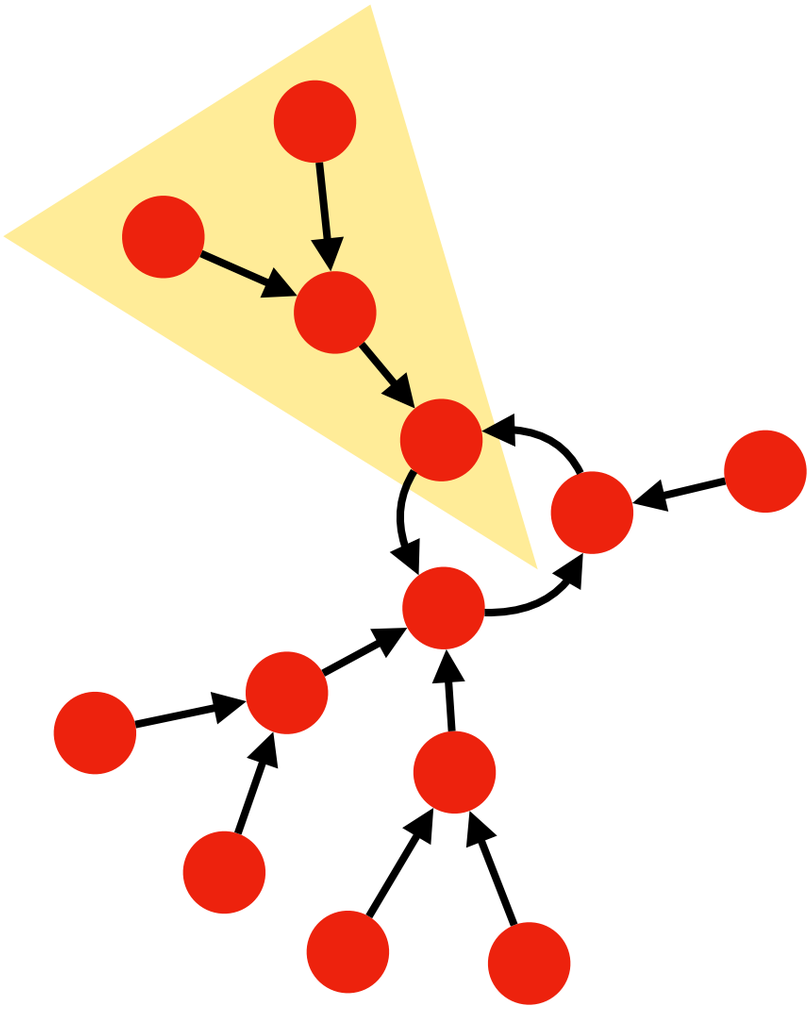


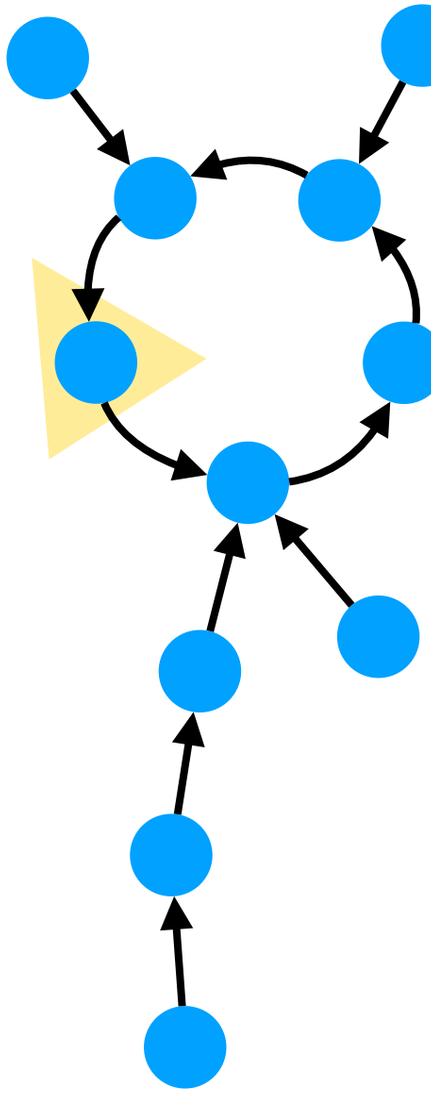


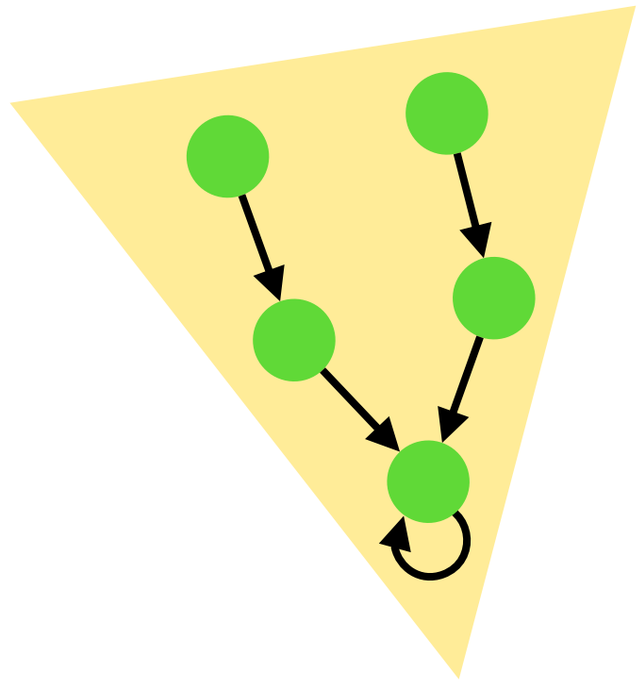


General shape of a dynamical system

A few limit cycles with trees going in

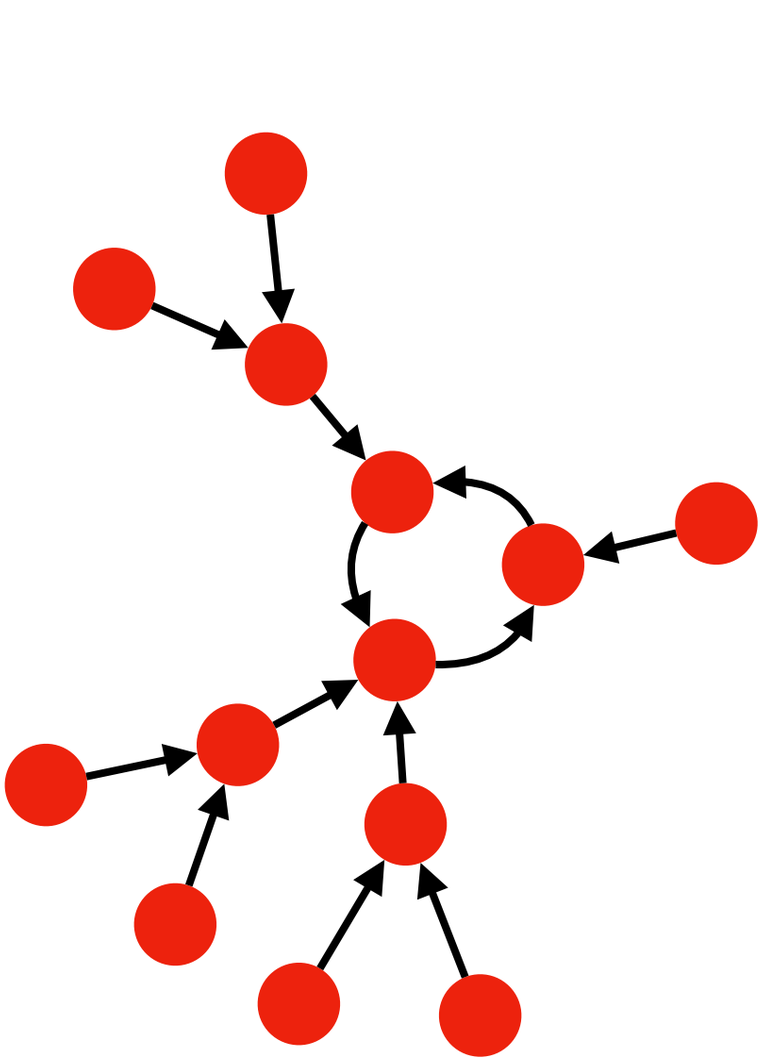


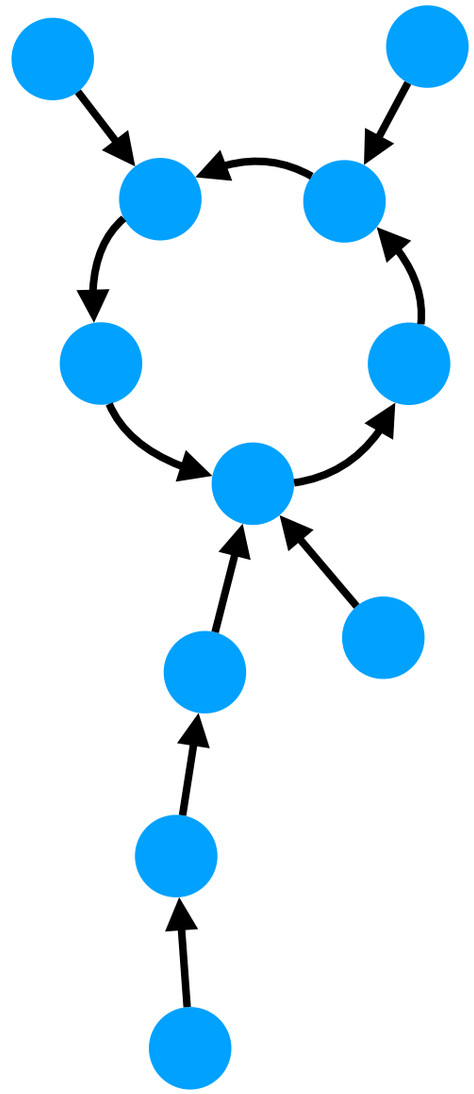


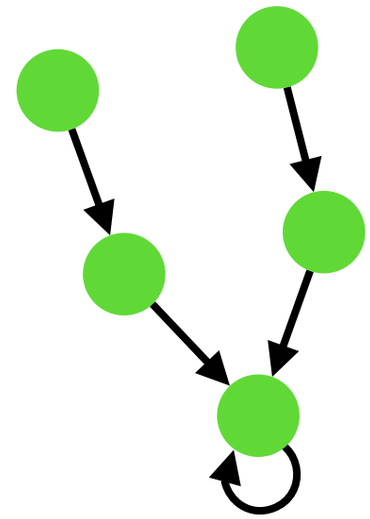


General shape of a dynamical system

A few limit cycles with trees going in



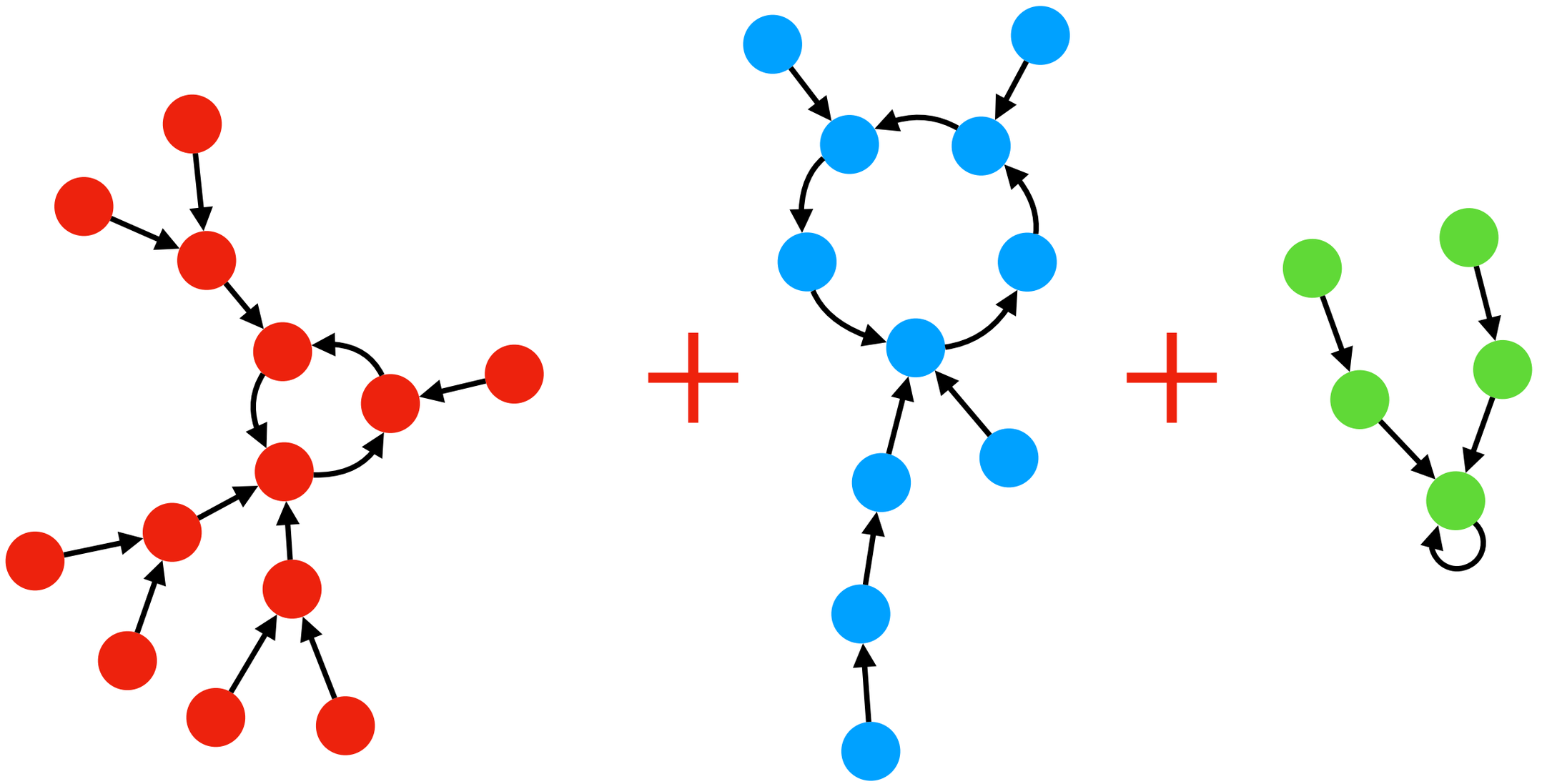




$$C_3\left(\ ,\ ,\ ,\right) \quad + \quad C_5\left(\ ,\ ,\ ,\ ,\ \right) \quad + \quad C_1\left(\ \right)$$

General shape of a dynamical system

A few limit cycles with trees going in



$$C_3\left(\ ,\ ,\ ,\ \right) + C_5\left(\ ,\ ,\ ,\ ,\ ,\ \right) + C_1\left(\ \right)$$

Isomorphism of dynamical systems is easy

2009 24th Annual IEEE Conference on Computational Complexity

Planar Graph Isomorphism is in Log-Space

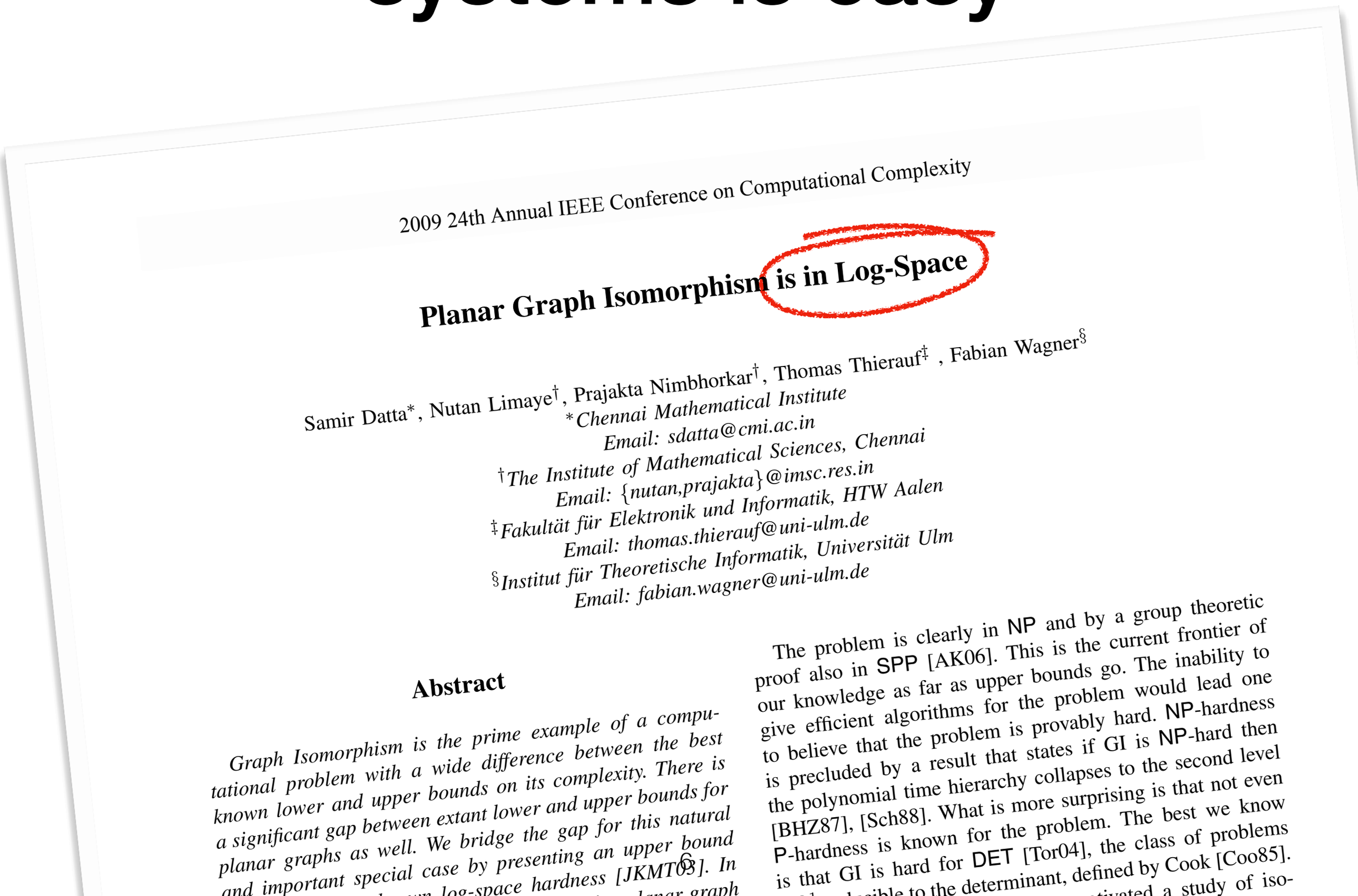
Samir Datta*, Nutan Limaye†, Prajakta Nimbhorkar†, Thomas Thierauf‡ , Fabian Wagner§
*Chennai Mathematical Institute
Email: sdatta@cmi.ac.in
†The Institute of Mathematical Sciences, Chennai
Email: {nutan,prajakta}@imsc.res.in
‡Fakultät für Elektronik und Informatik, HTW Aalen
Email: thomas.thierauf@uni-ulm.de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

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

Isomorphism of dynamical systems is easy



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

INTERMISSION

How to efficiently generate dynamical systems, aka functional digraphs

How to efficiently generate dynamical systems, aka functional digraphs

- In theory: Antonio E. Porreca, Ekaterina Timofeeva, Polynomial-delay generation of functional digraphs up to isomorphism, arXiv:2302.13832

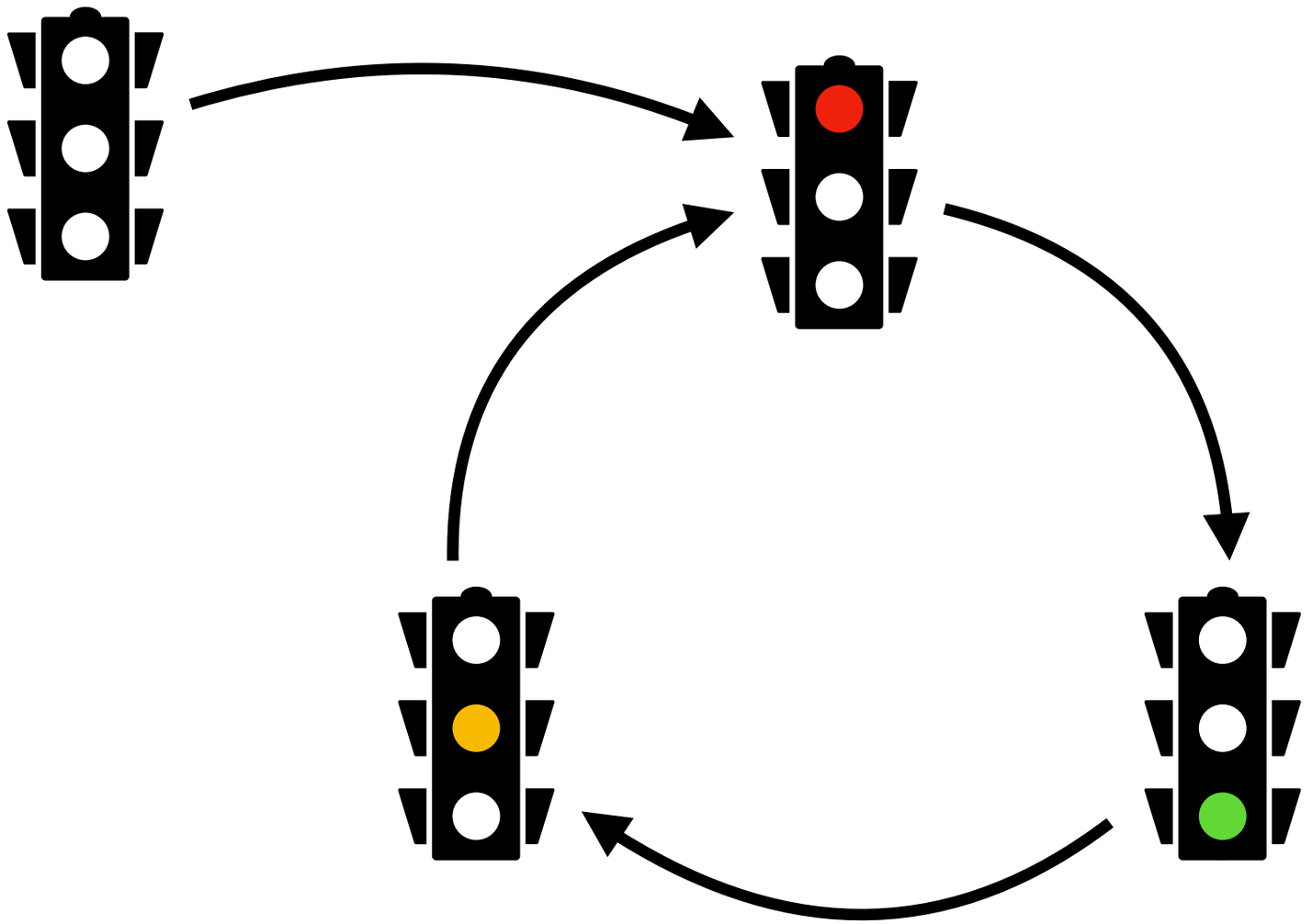
How to efficiently generate dynamical systems, aka functional digraphs

- In theory: Antonio E. Porreca, Ekaterina Timofeeva, Polynomial-delay generation of functional digraphs up to isomorphism, arXiv:2302.13832

- In practice: `funkdigen2`, a fast implementation of the above, github.com/aeporreca/funkdigen2

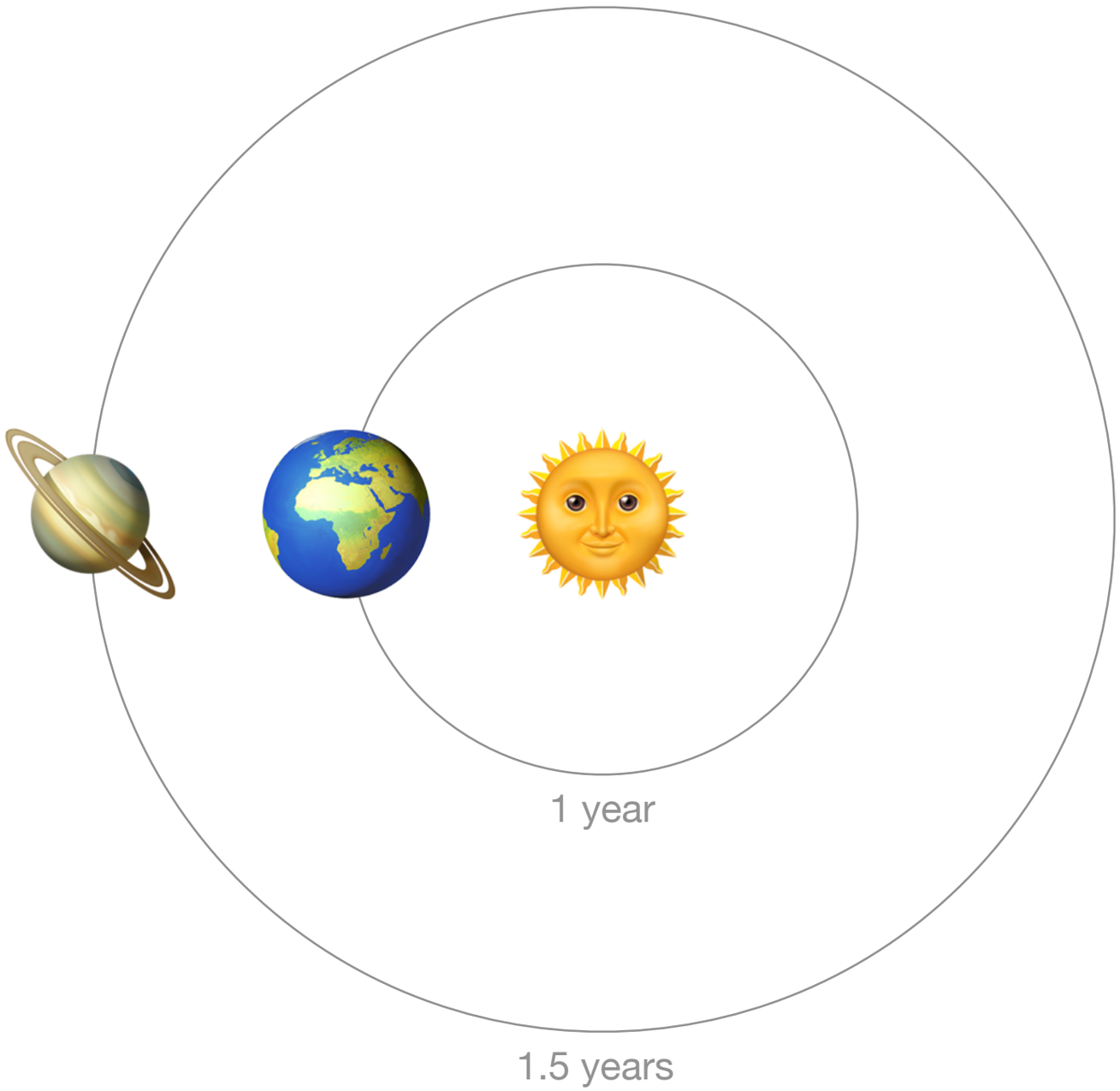
A toy example from engineering

Traffic lights

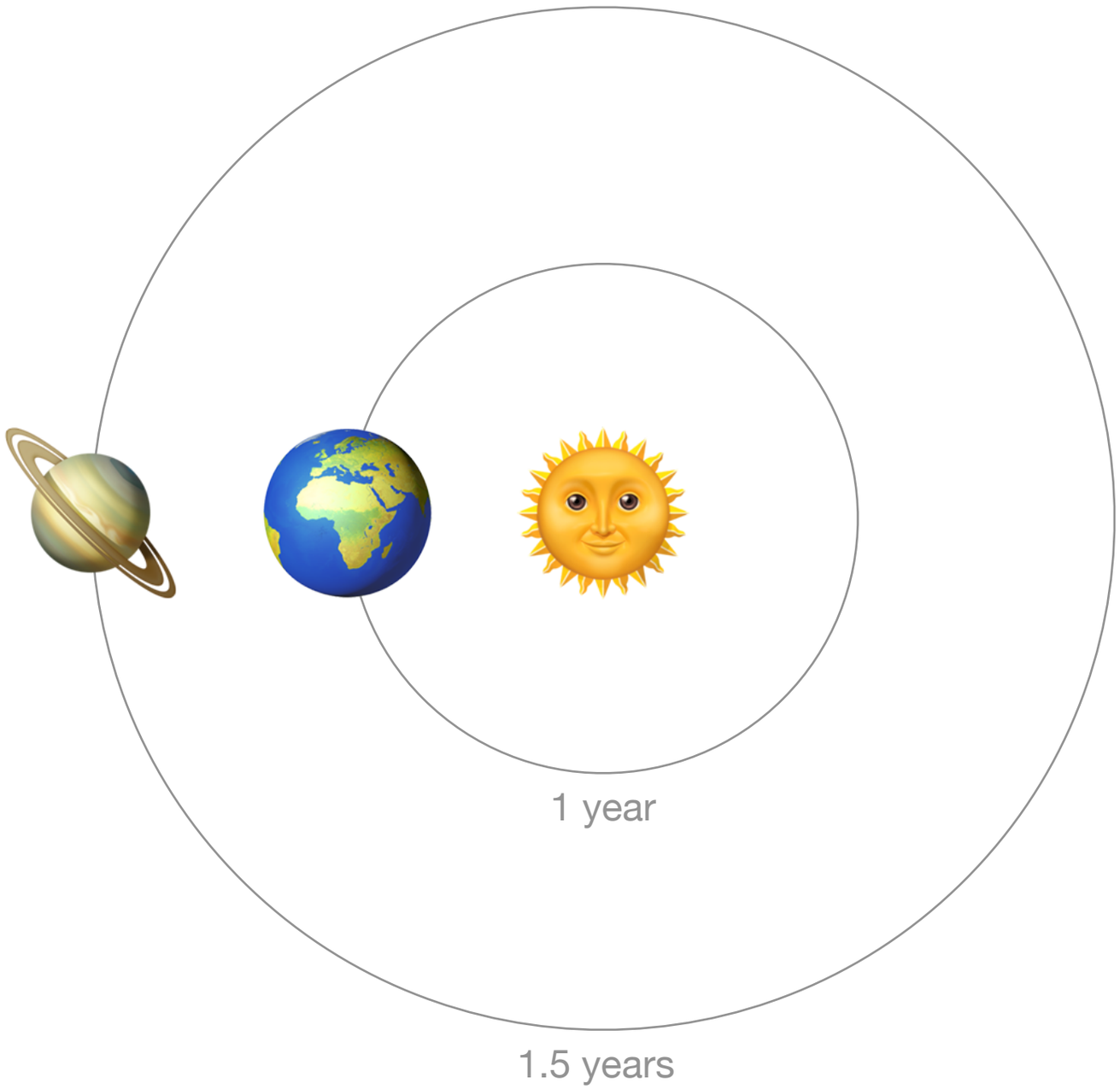


A toy example from science

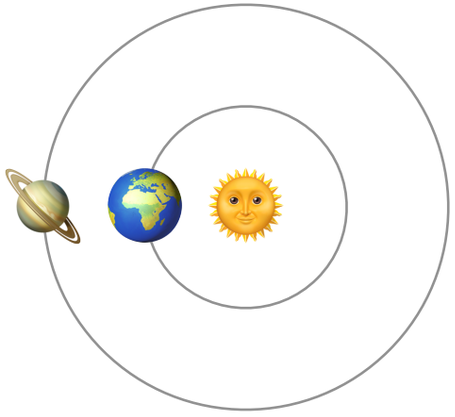
A planetary system



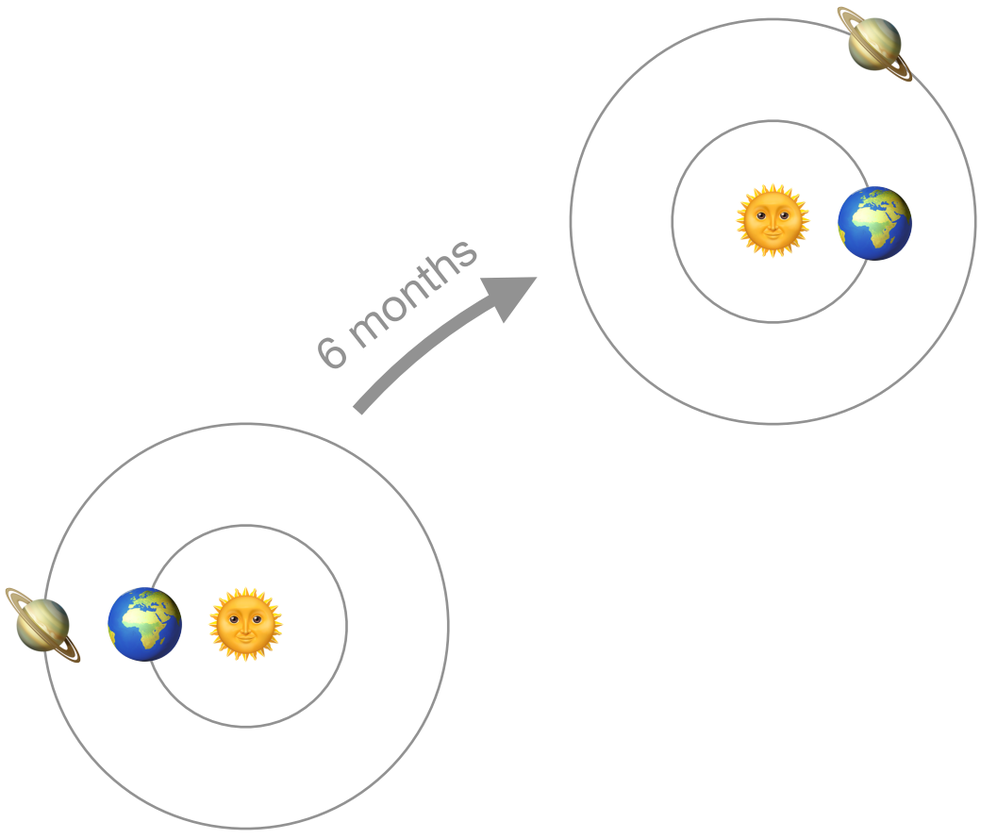
A planetary system



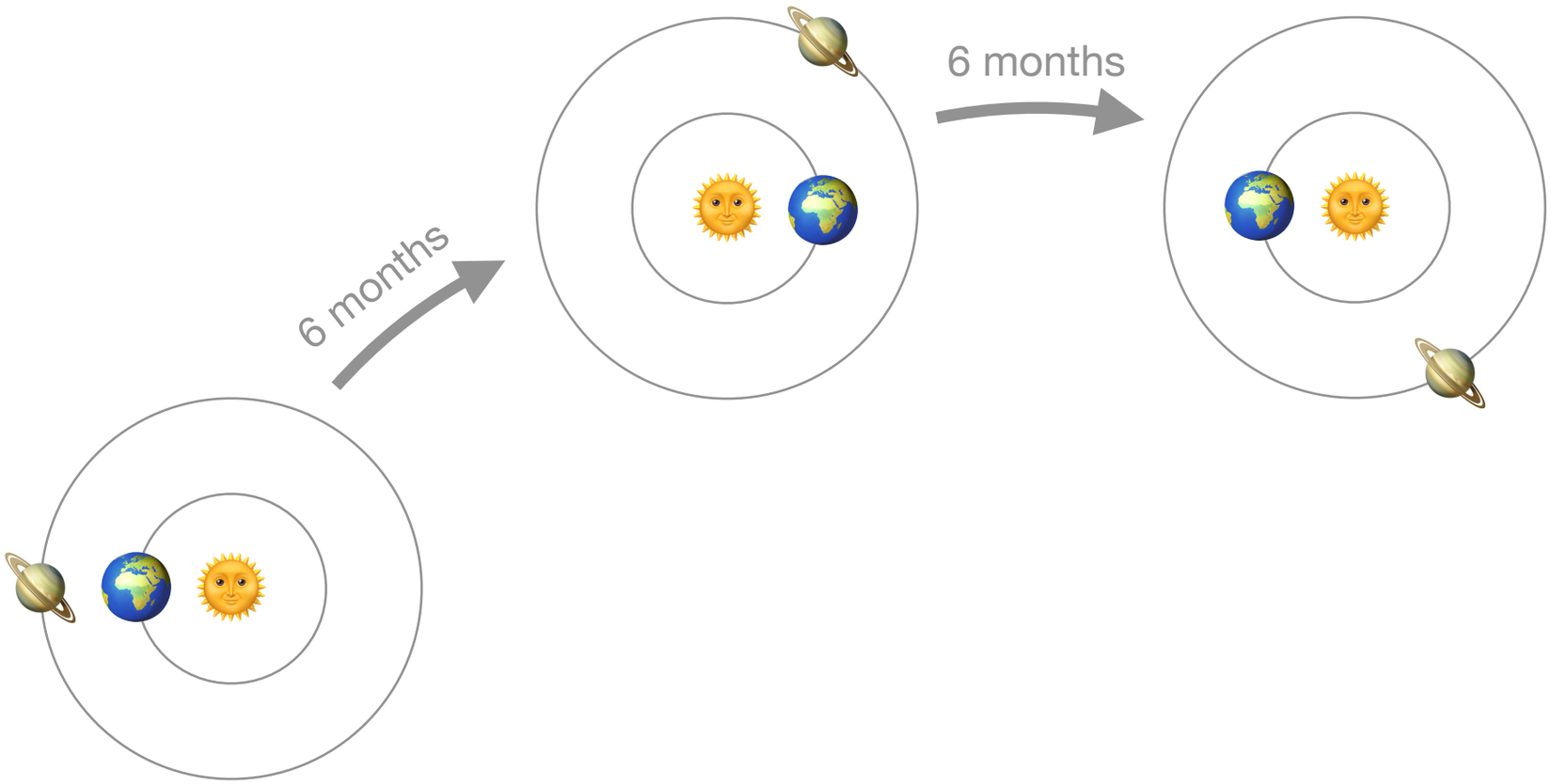
Evolution in time



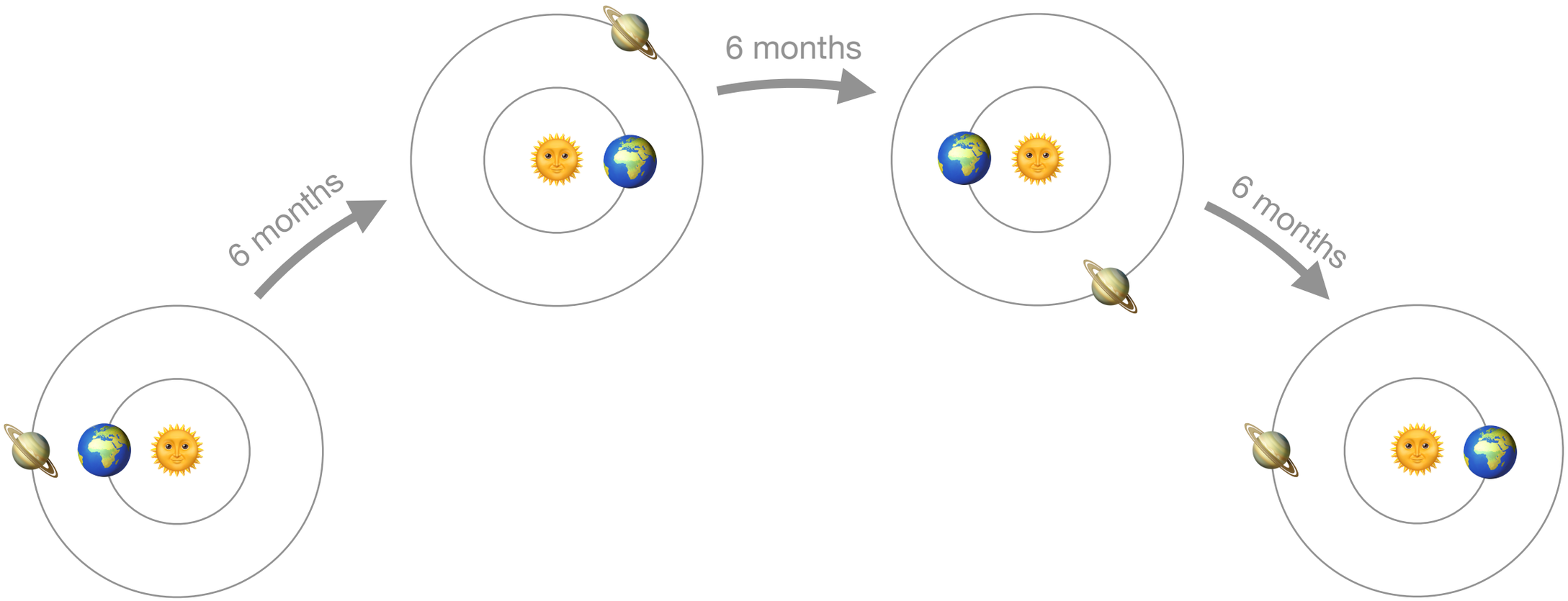
Evolution in time



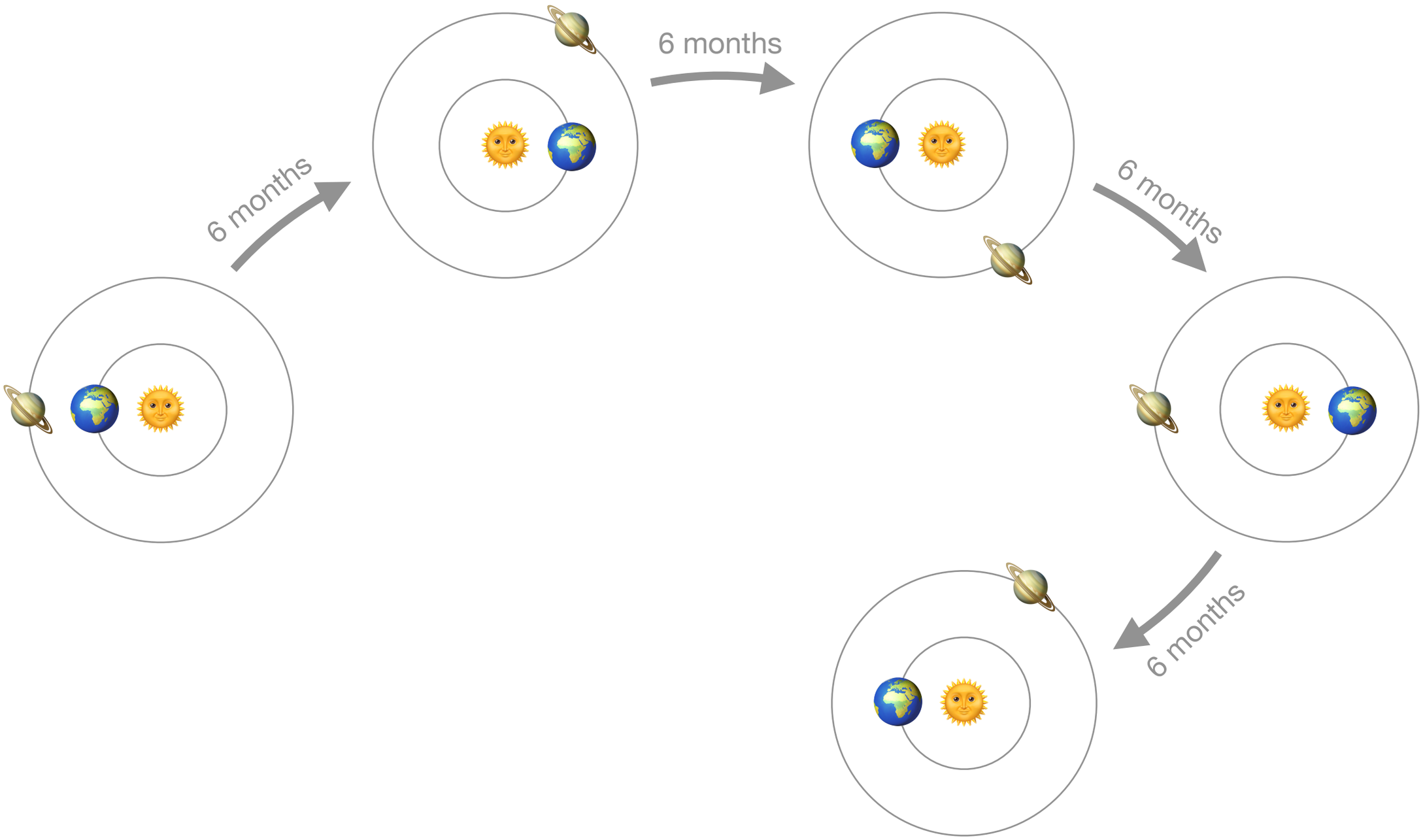
Evolution in time



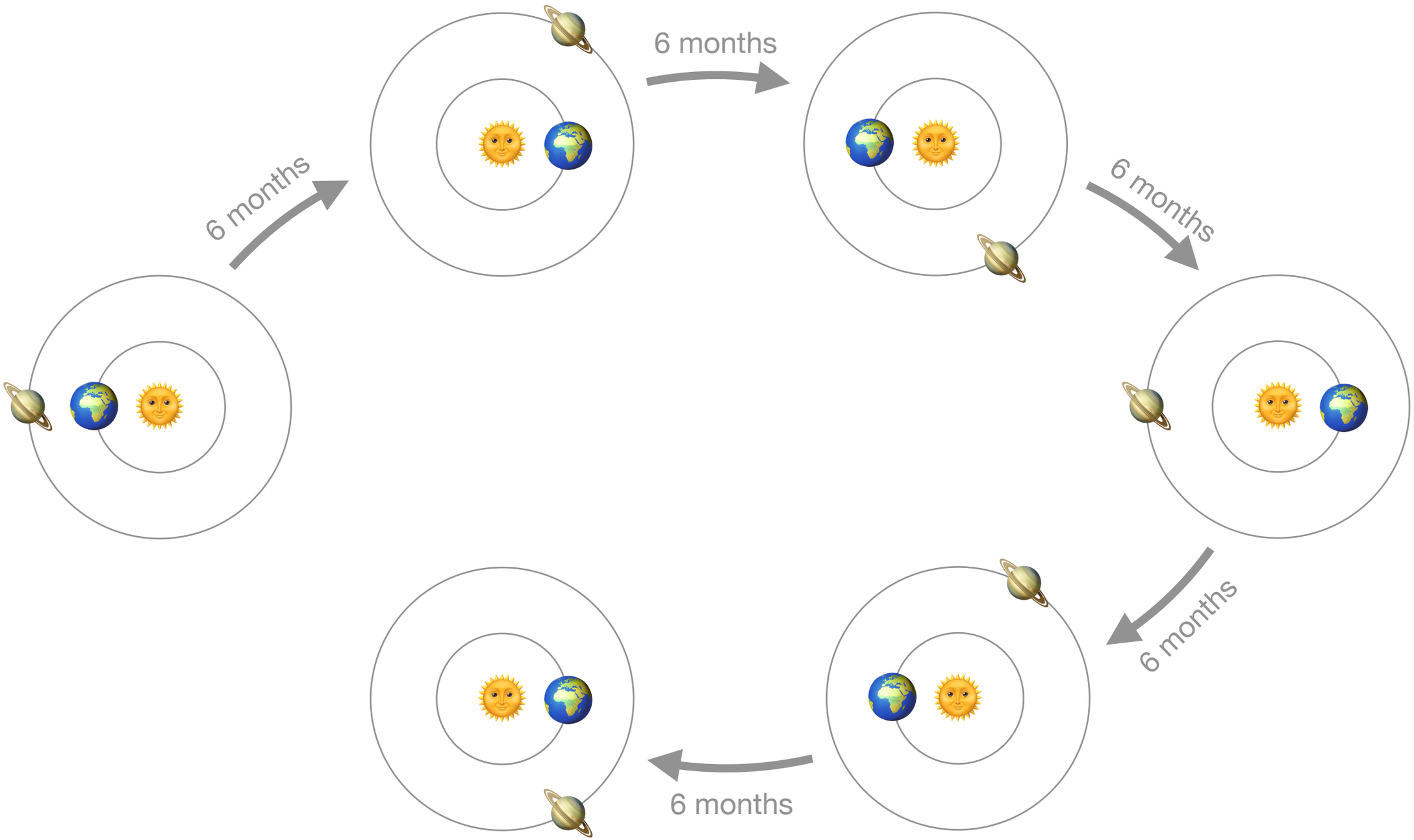
Evolution in time



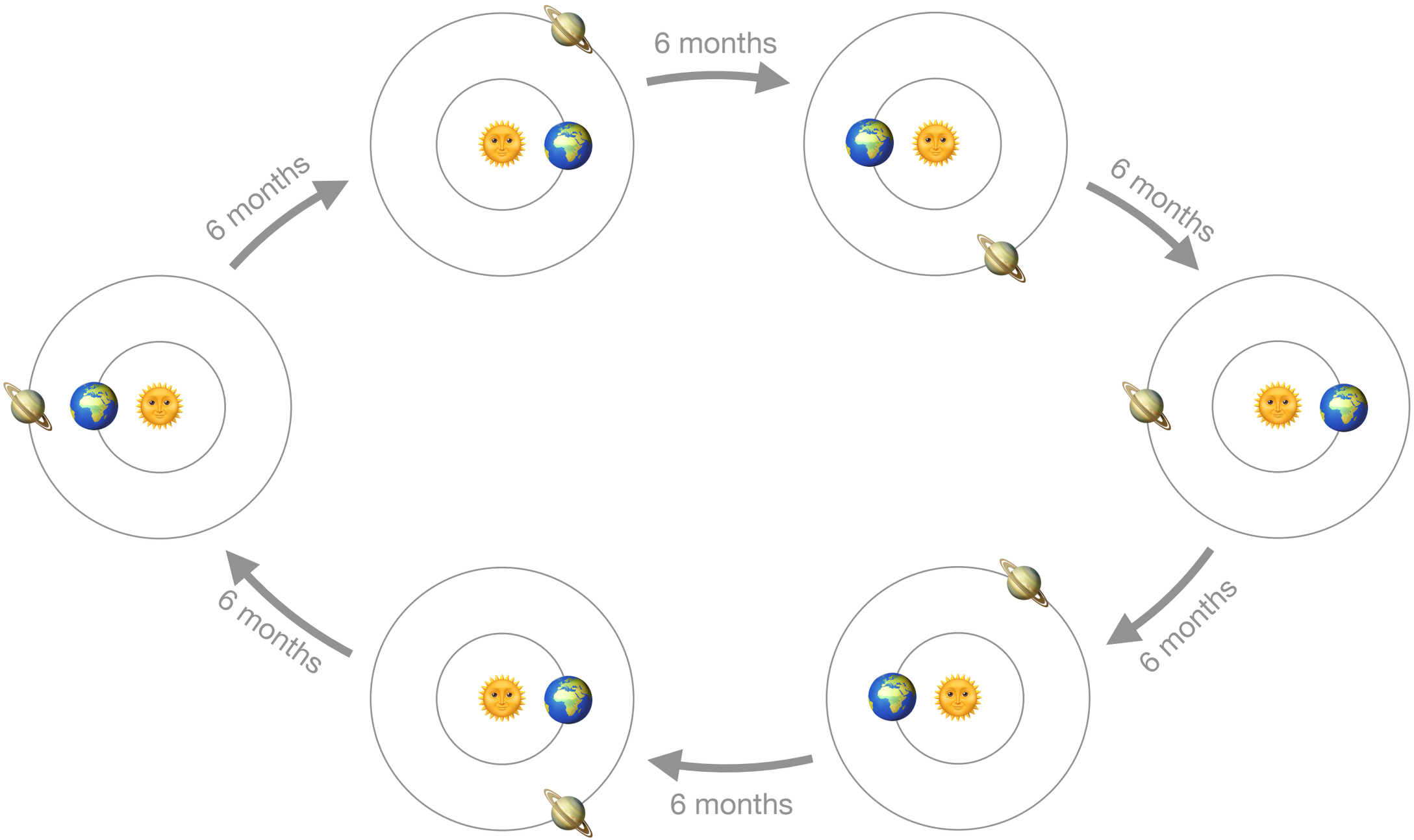
Evolution in time



Evolution in time

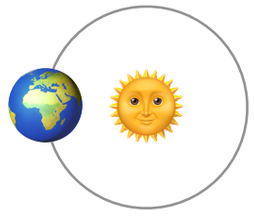


Evolution in time

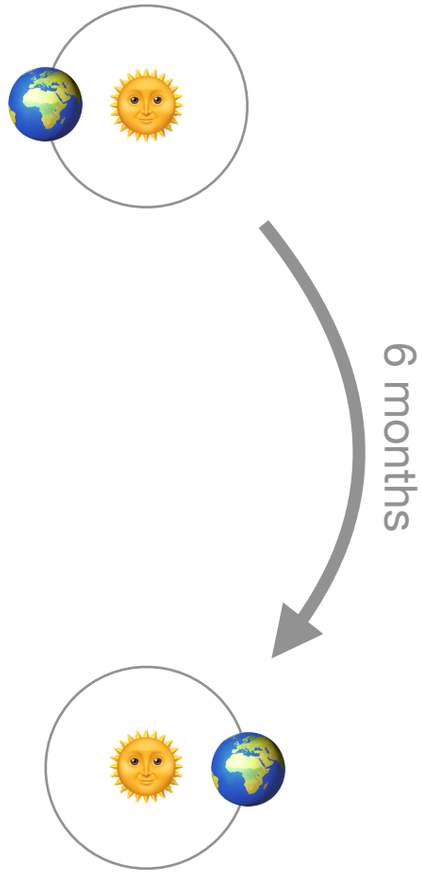


Decomposing the system

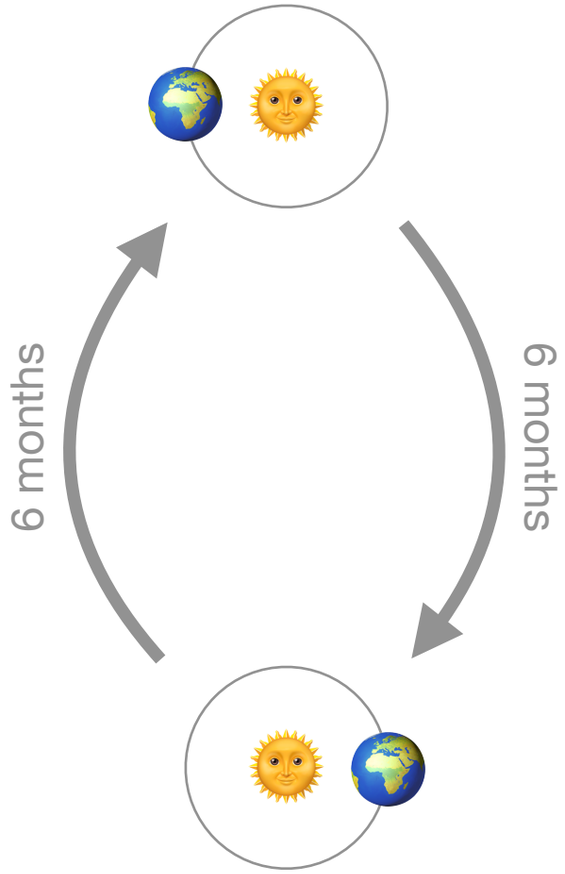
Decomposing the system



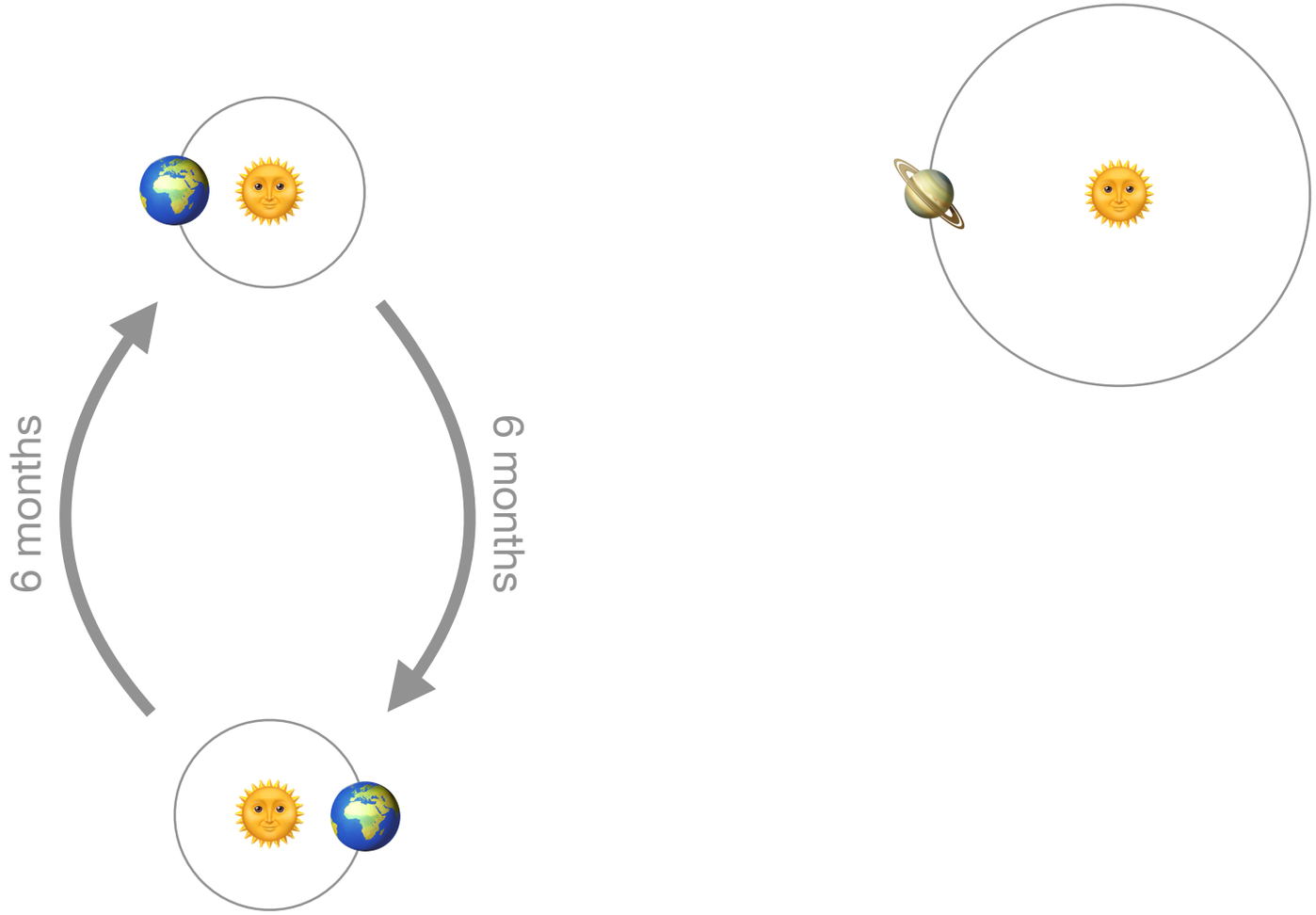
Decomposing the system



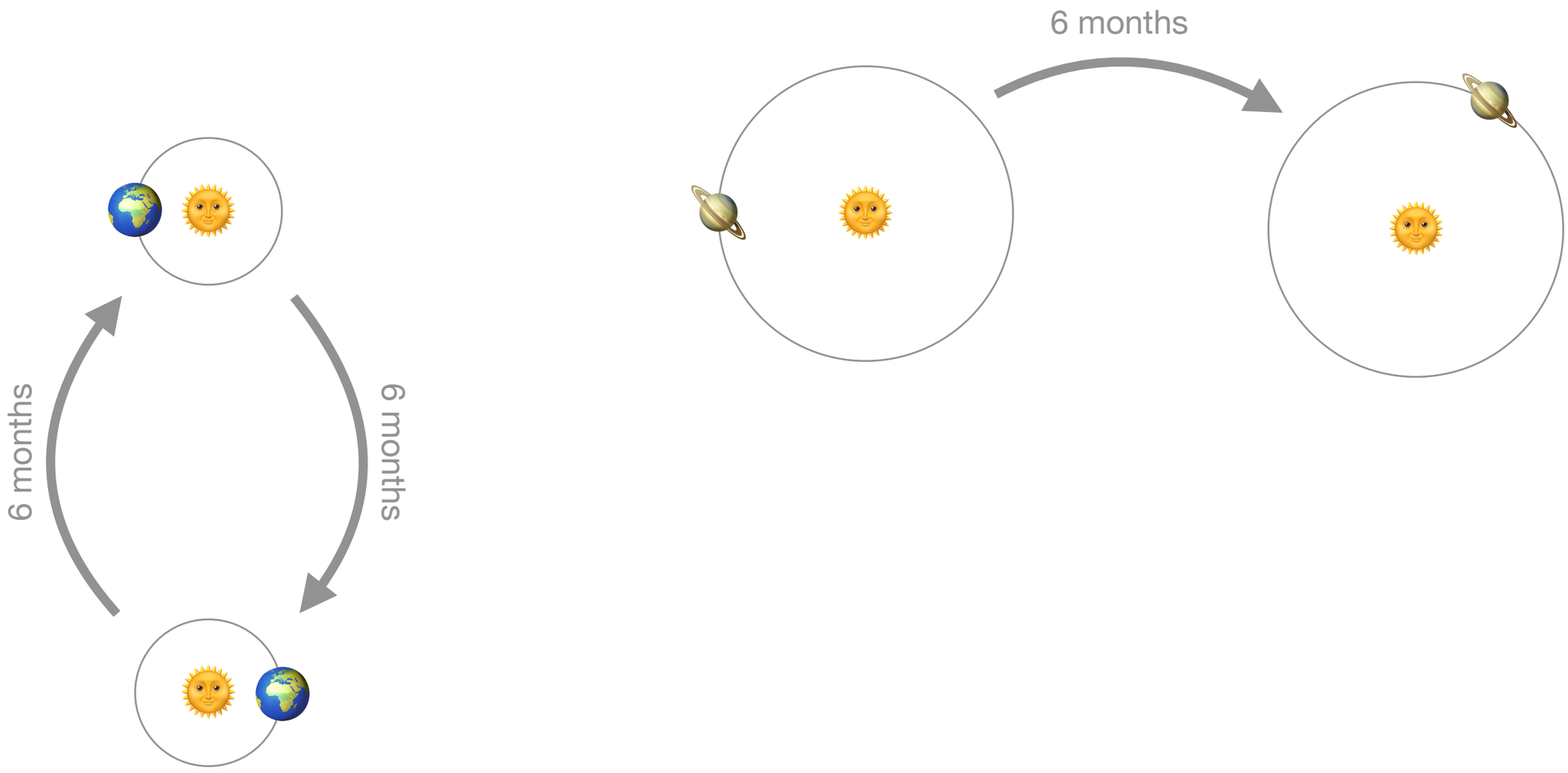
Decomposing the system



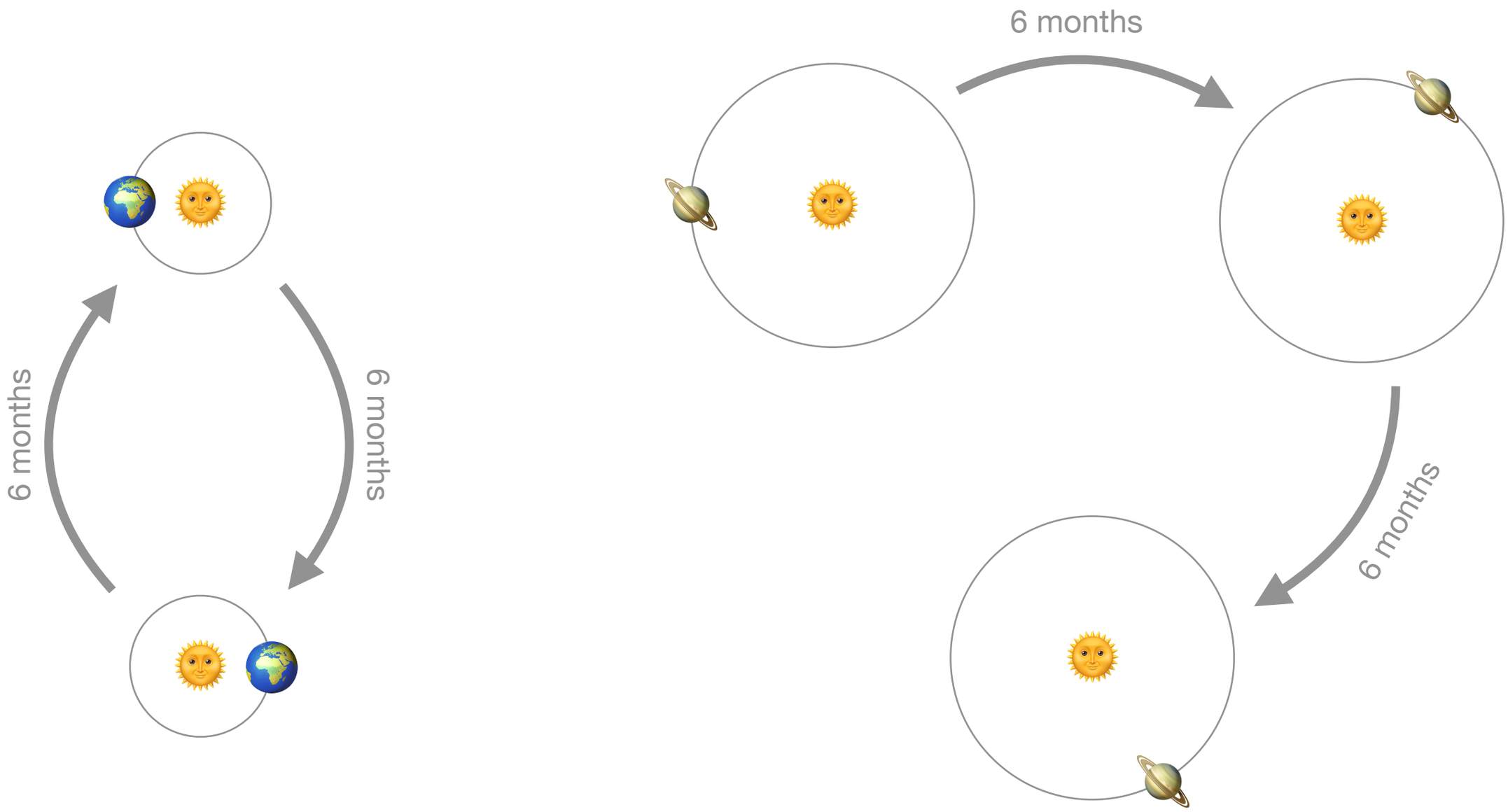
Decomposing the system



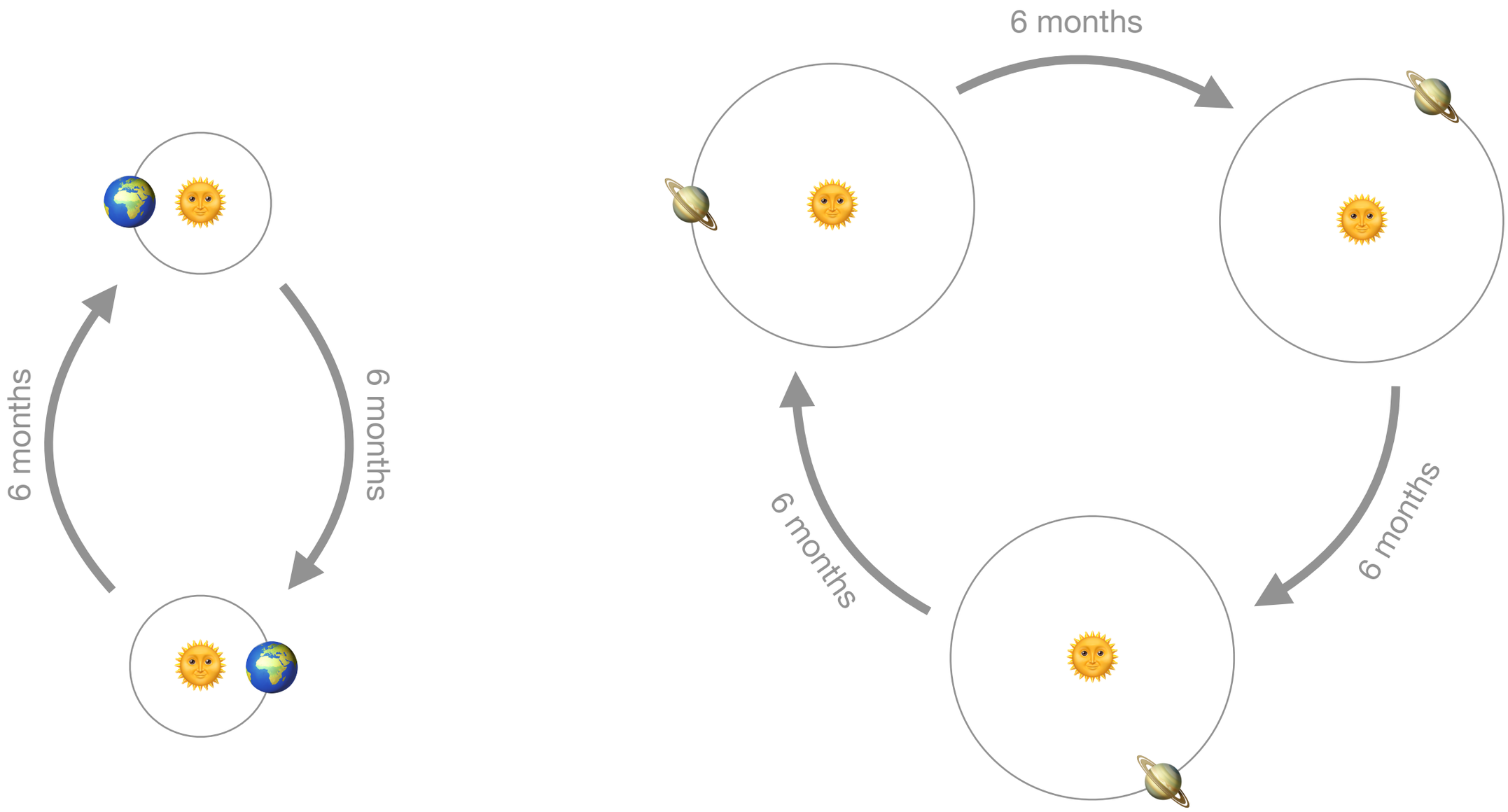
Decomposing the system



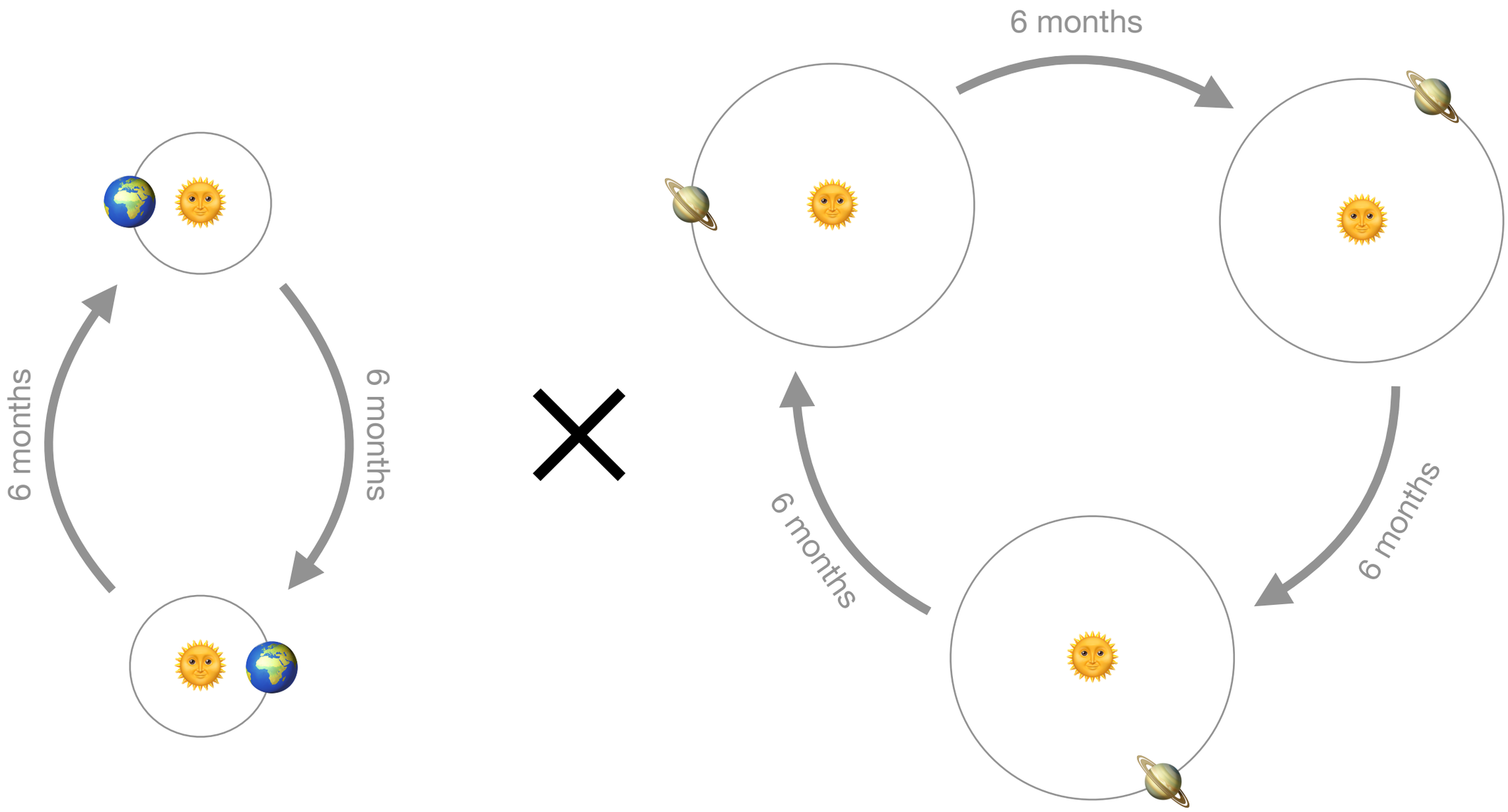
Decomposing the system



Decomposing the system

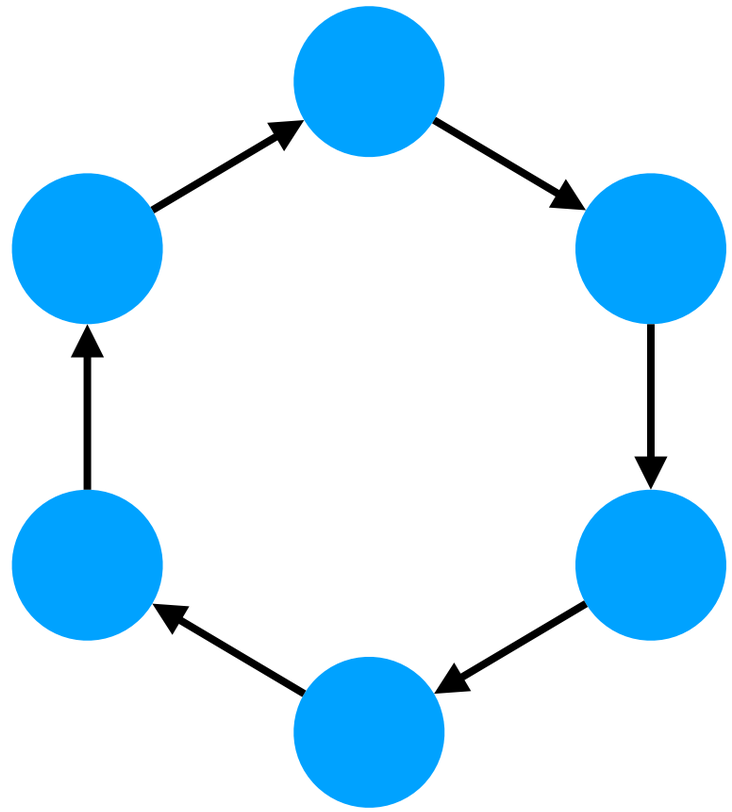


Decomposing the system

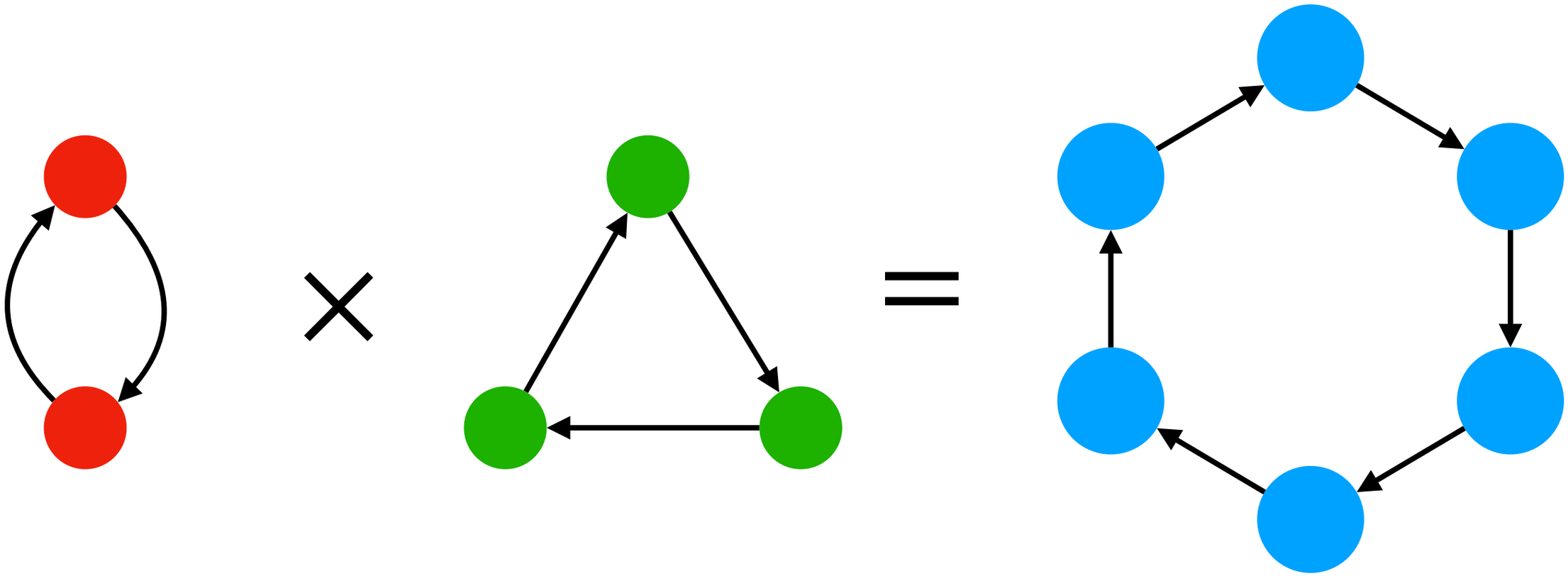


What if our instruments are less sophisticated?

Abstract evolution of the system

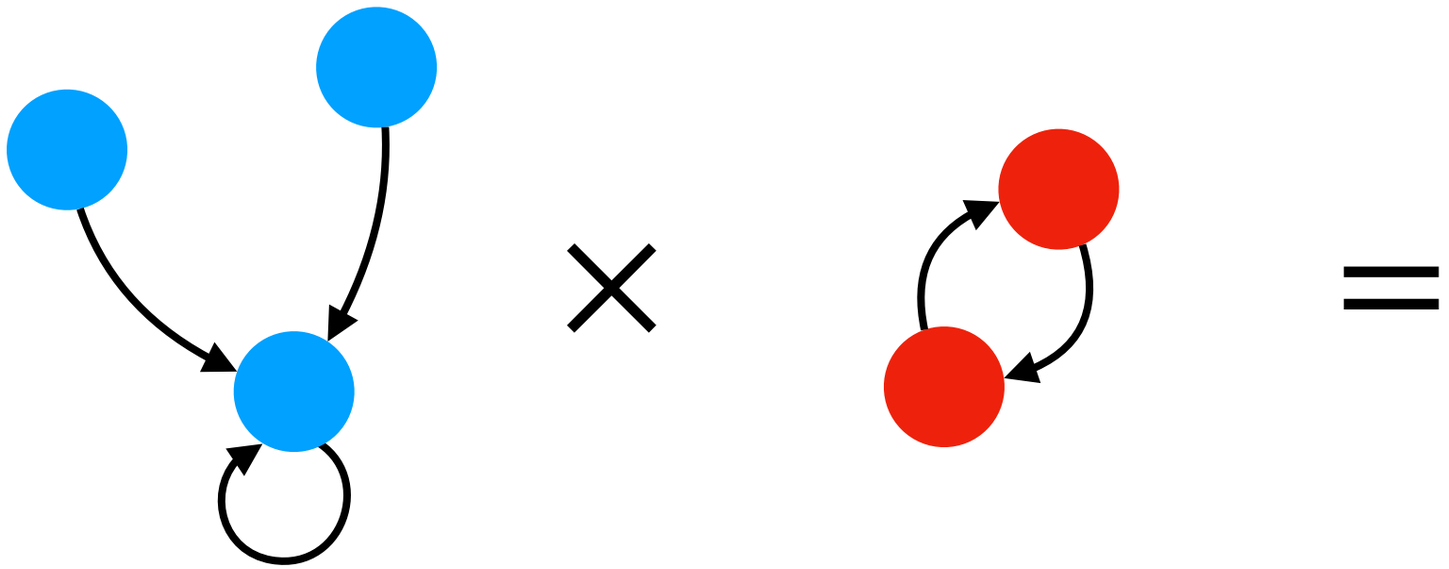


Abstract evolution of the system

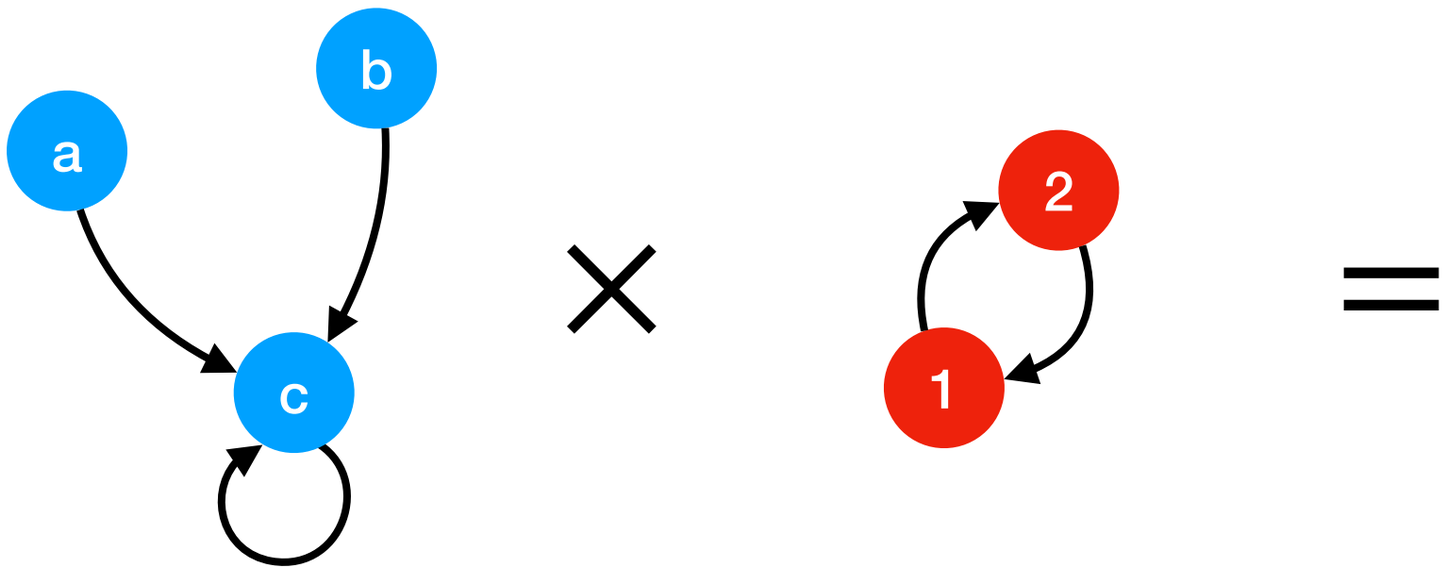


Product of dynamical systems

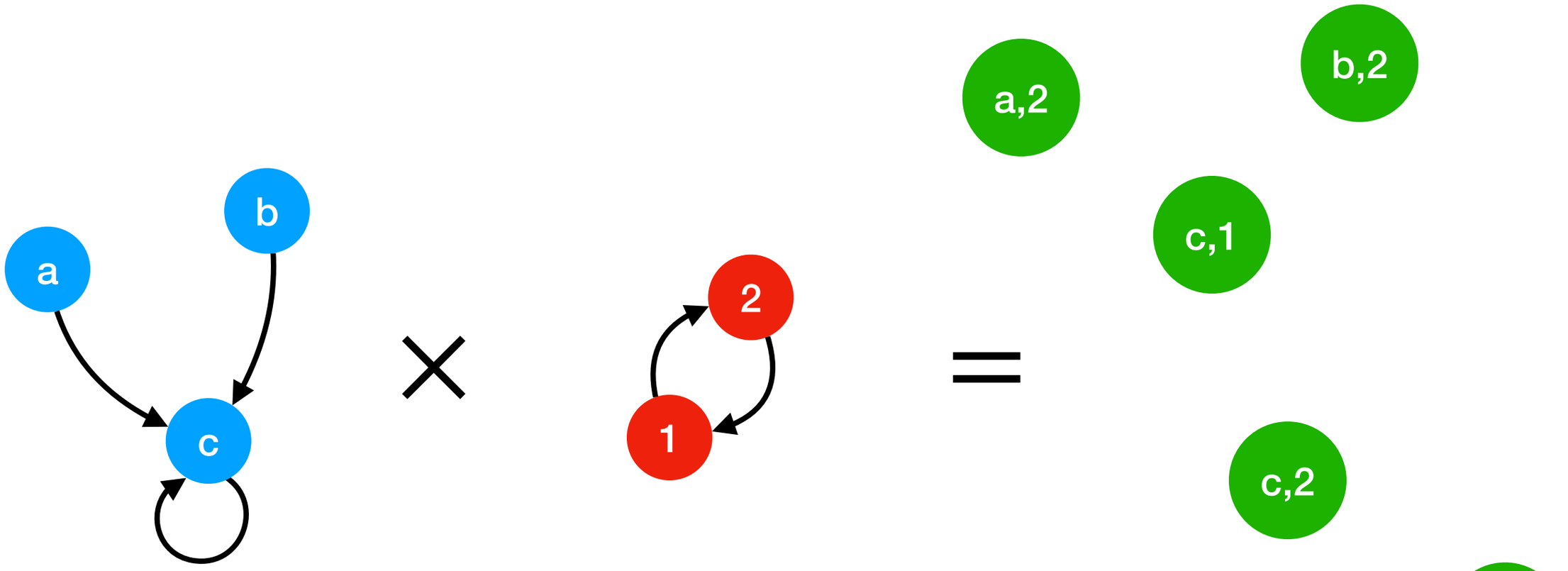
Product of systems



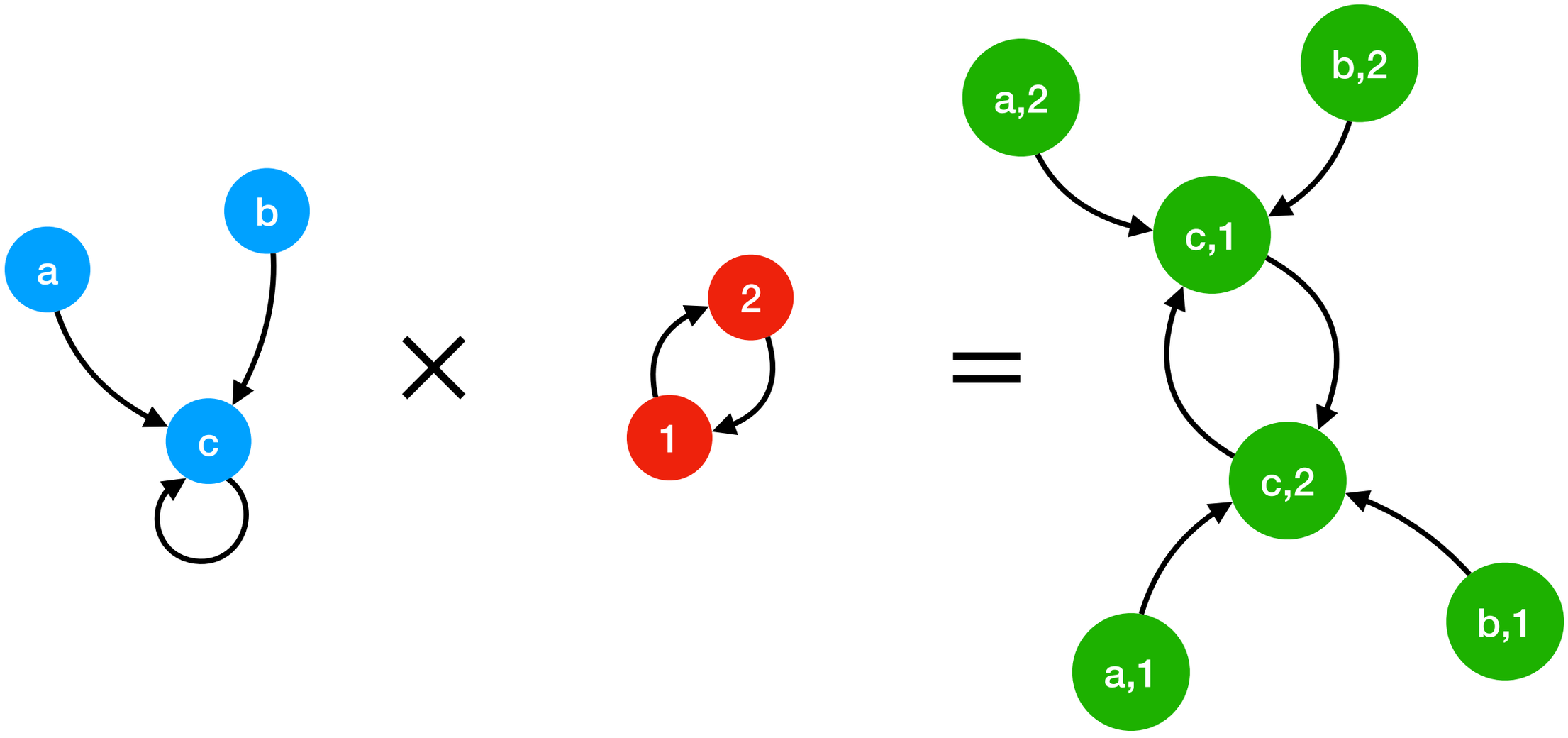
Give temporary names to the states



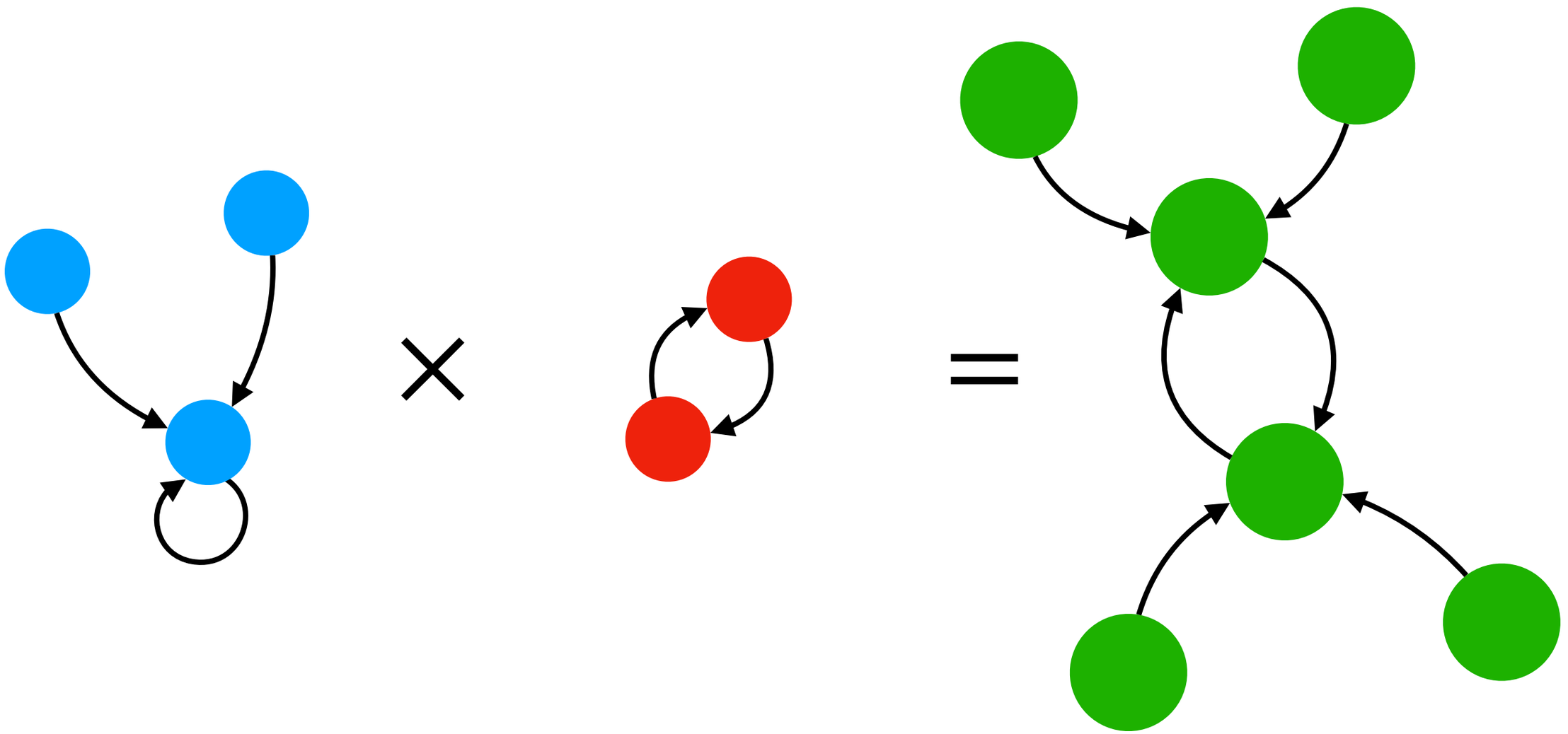
Compute the Cartesian product



Add the arcs between states

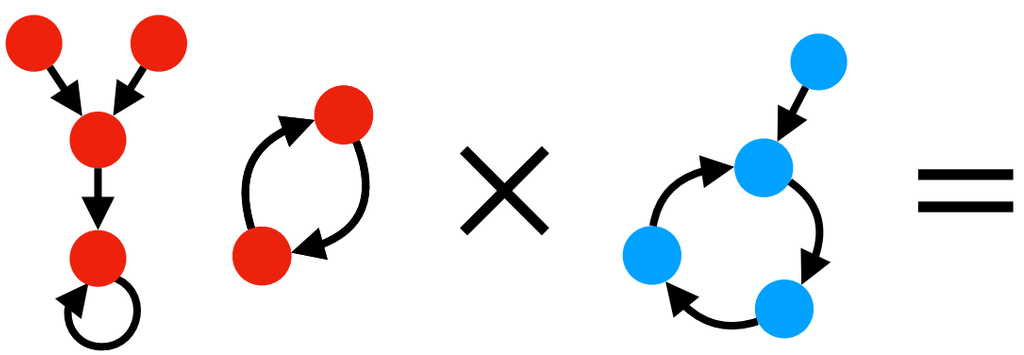


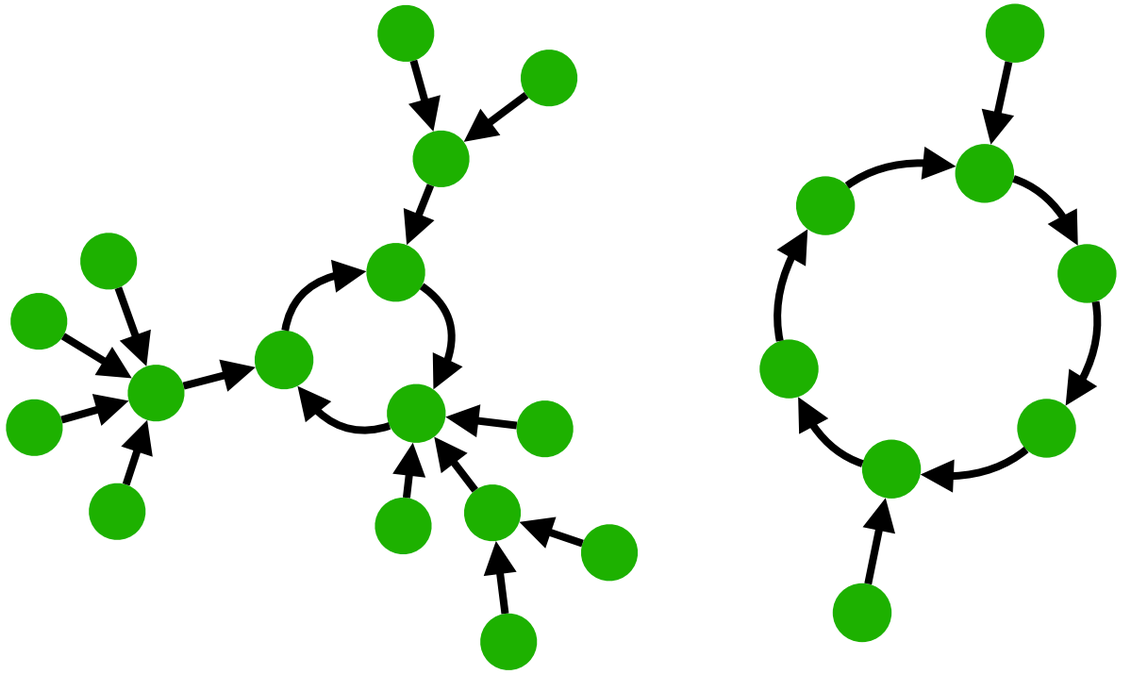
Forget the names once again



Products “preserve” behaviours

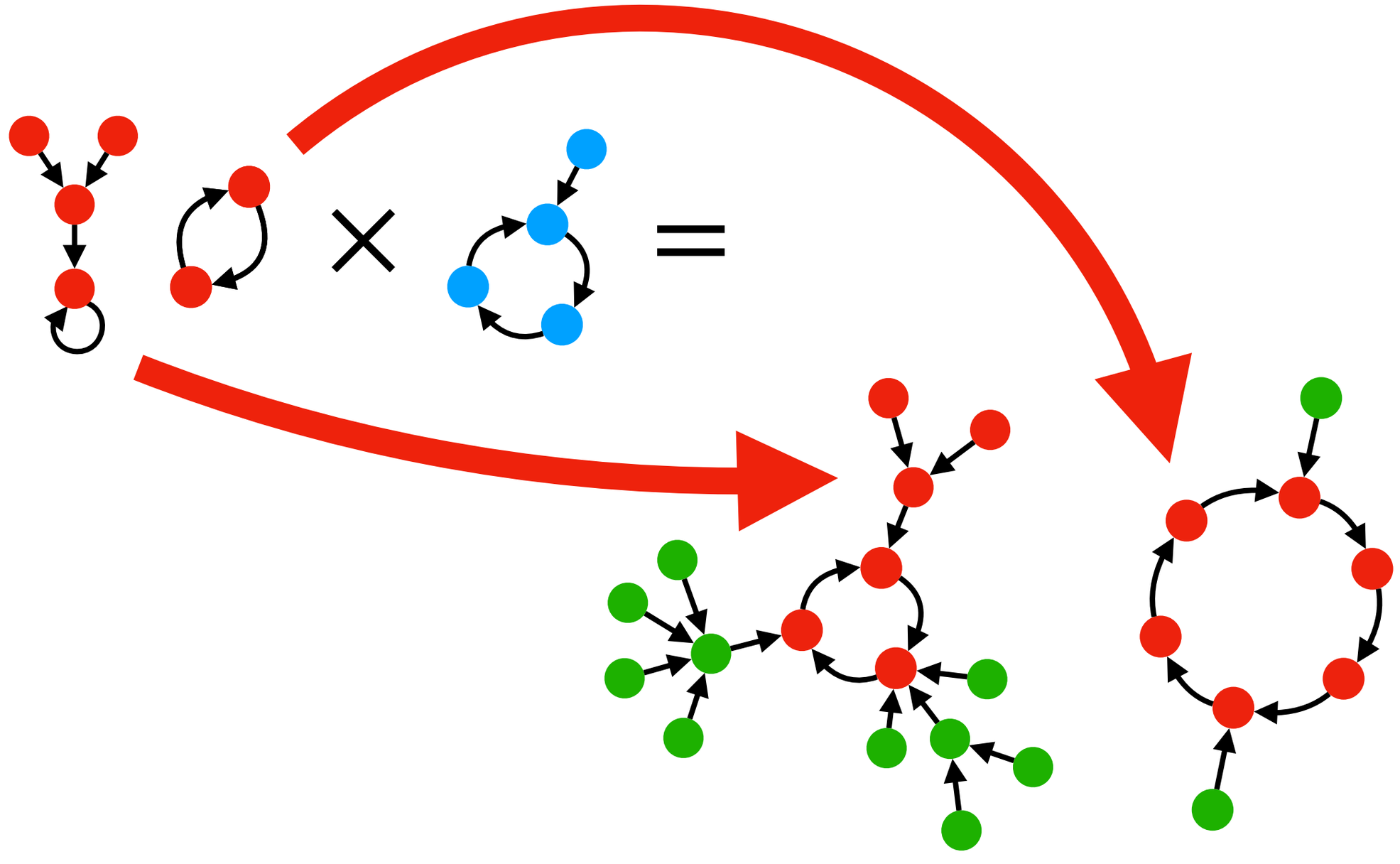
A is a minor of $A \times B$ for $B \neq \emptyset$





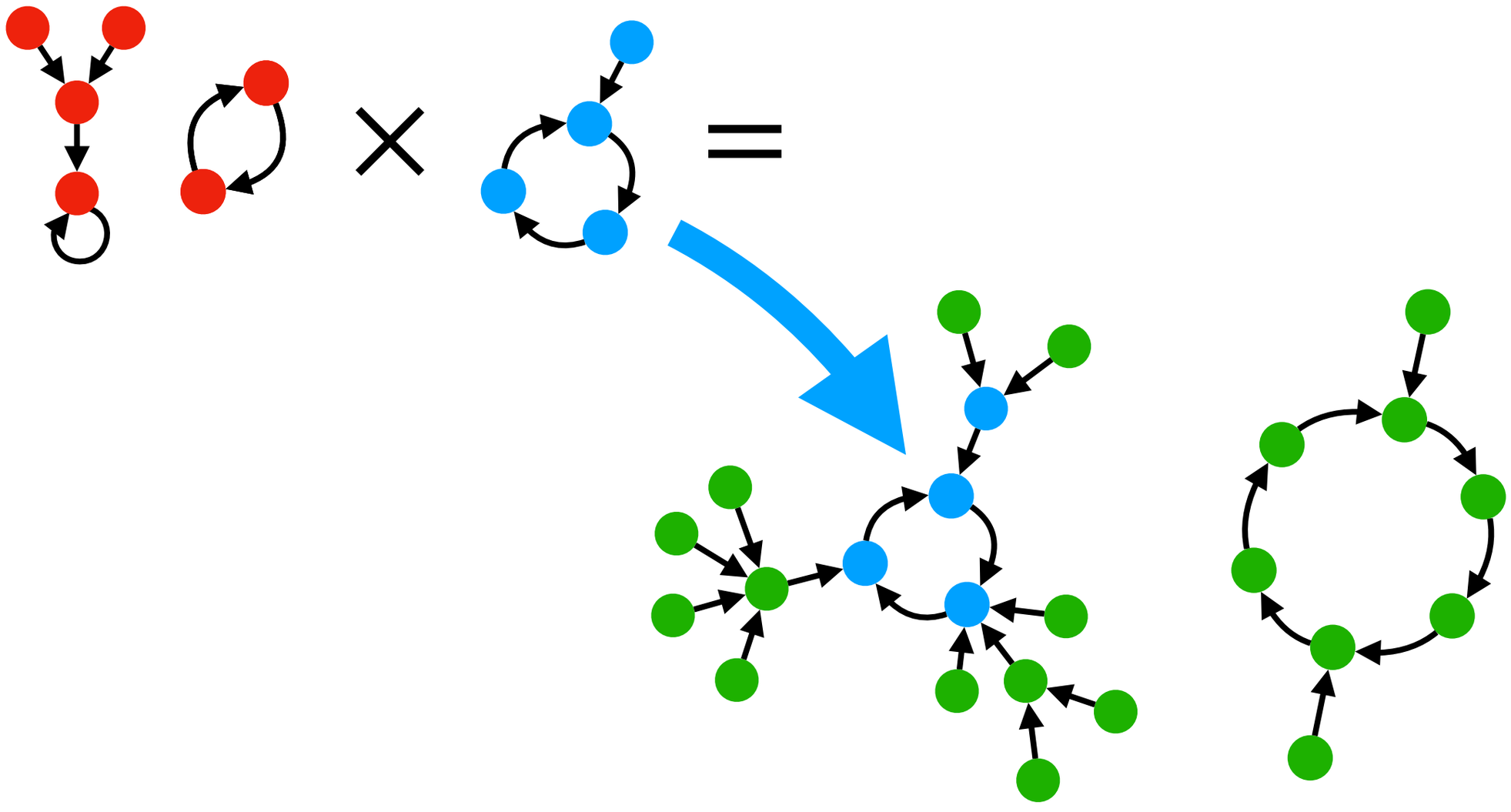
Products “preserve” behaviours

A is a minor of $A \times B$ for $B \neq \emptyset$



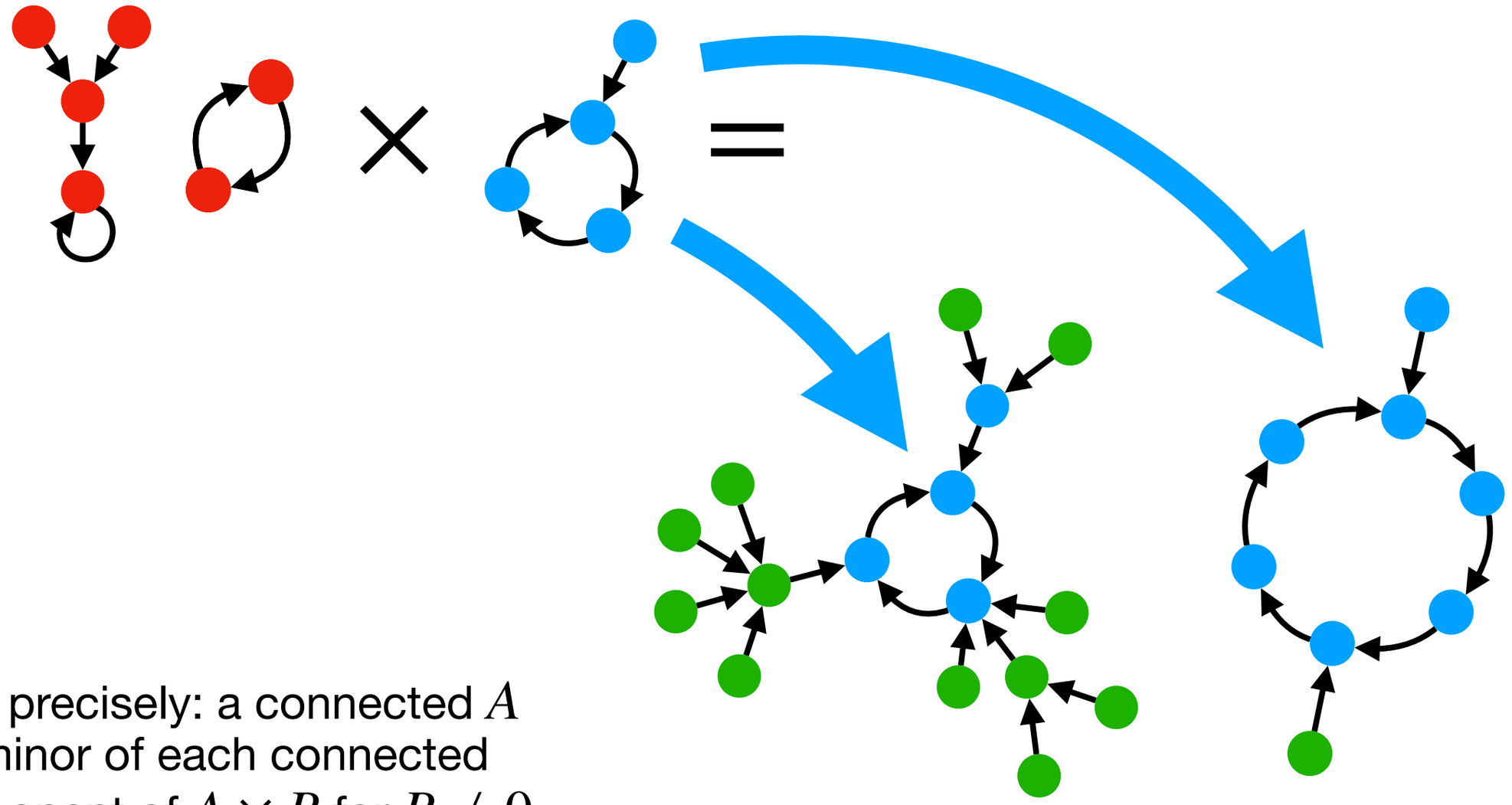
Products “preserve” behaviours

A is a minor of $A \times B$ for $B \neq \emptyset$



Products “preserve” behaviours

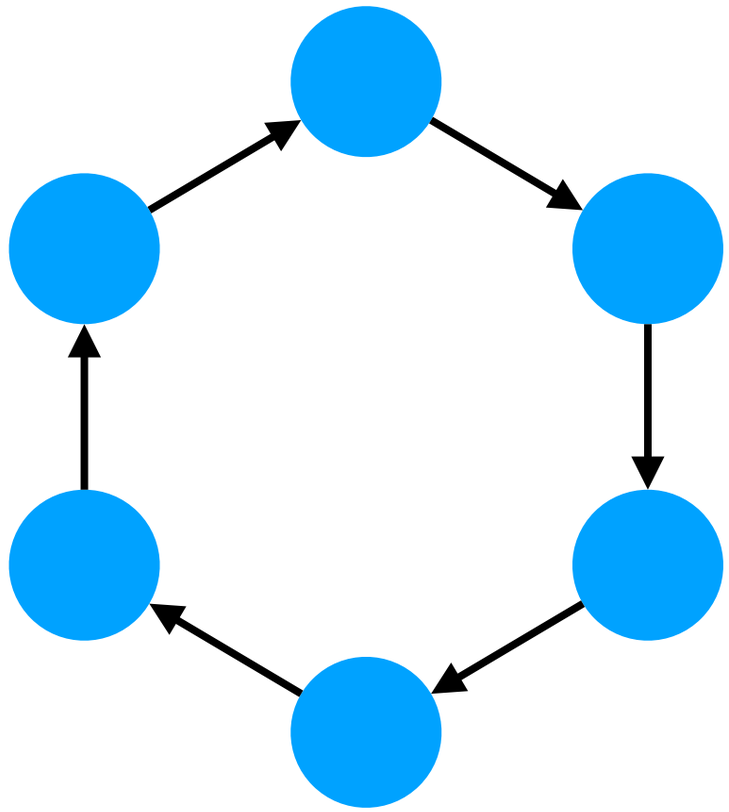
A is a minor of $A \times B$ for $B \neq \emptyset$



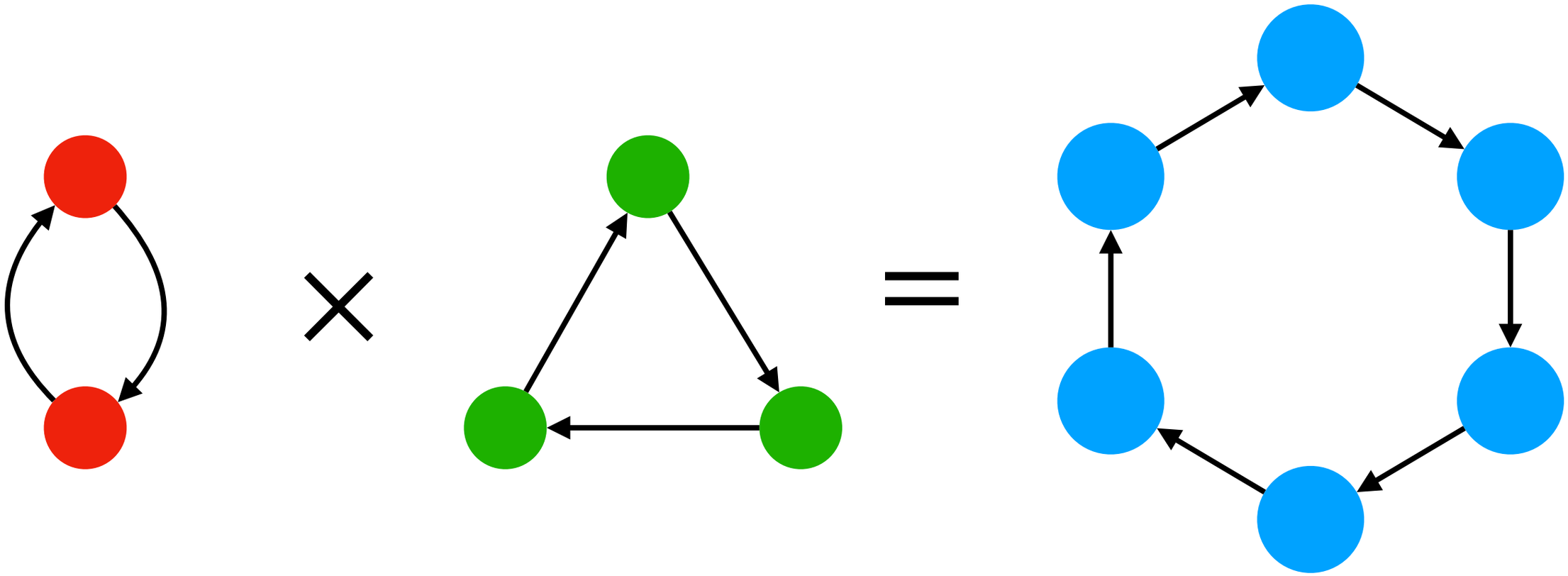
more precisely: a connected A is a minor of each connected component of $A \times B$ for $B \neq 0$

Back to our planetary system

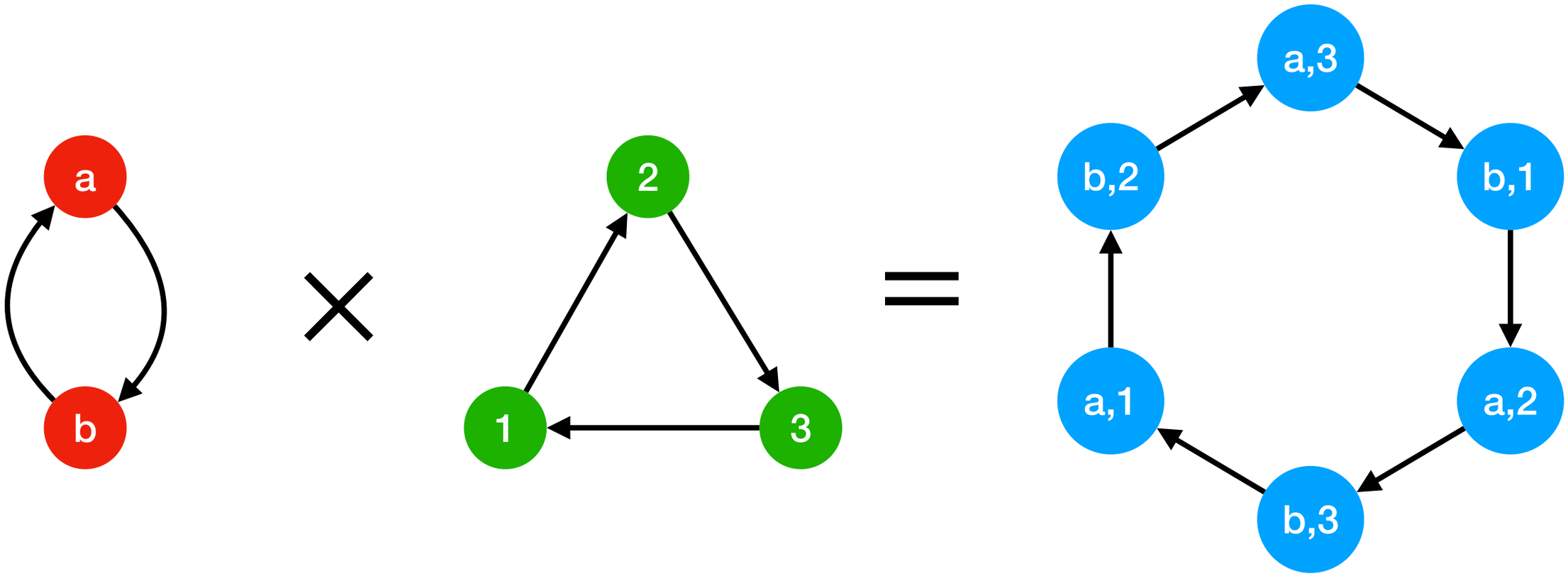
Decomposition



Decomposition



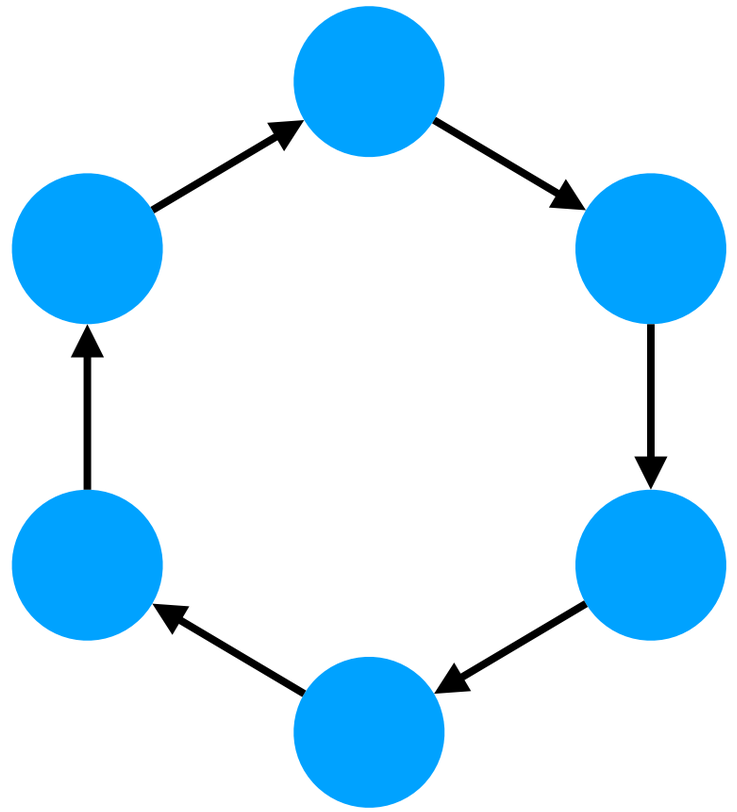
Decomposition



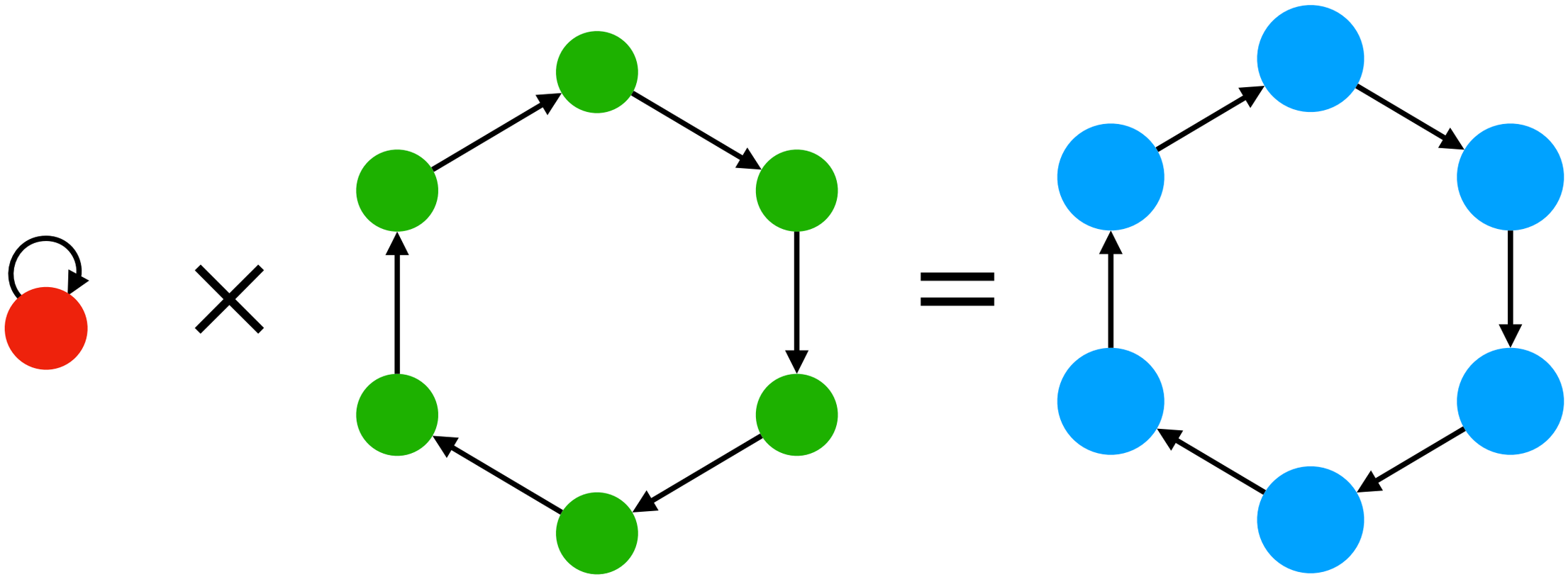
Any other decomposition?

Another decomposition

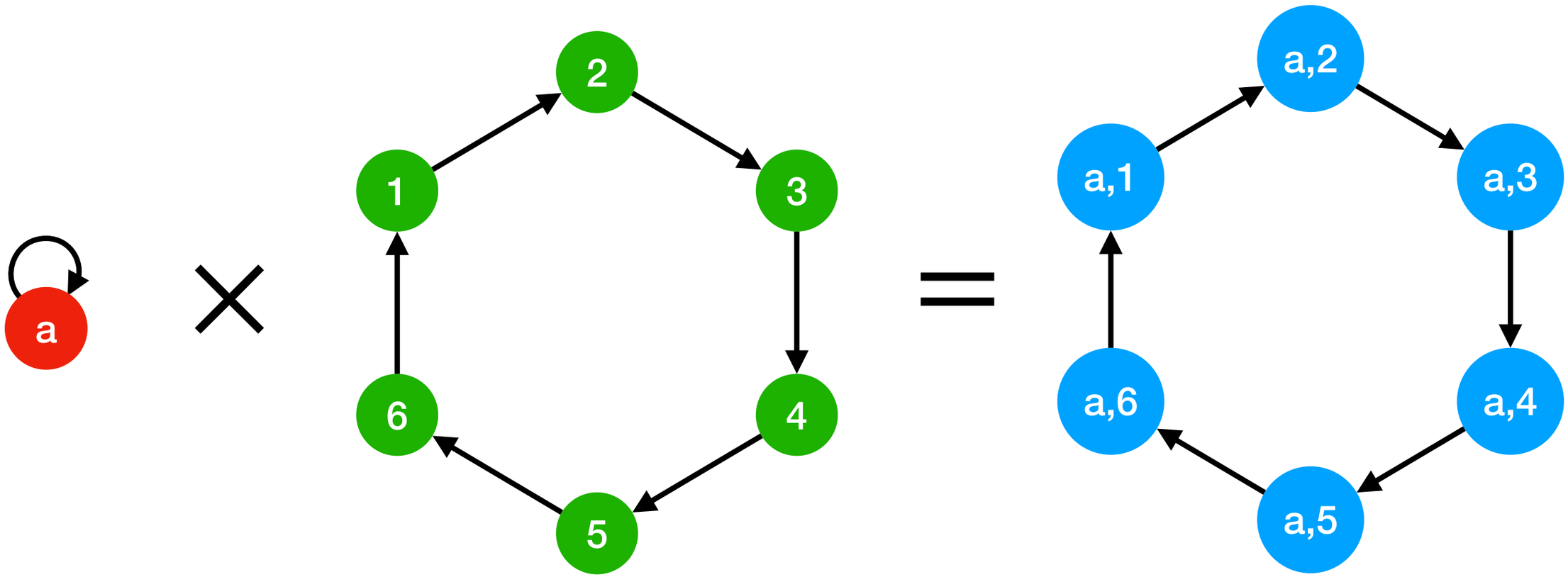
=



Another decomposition

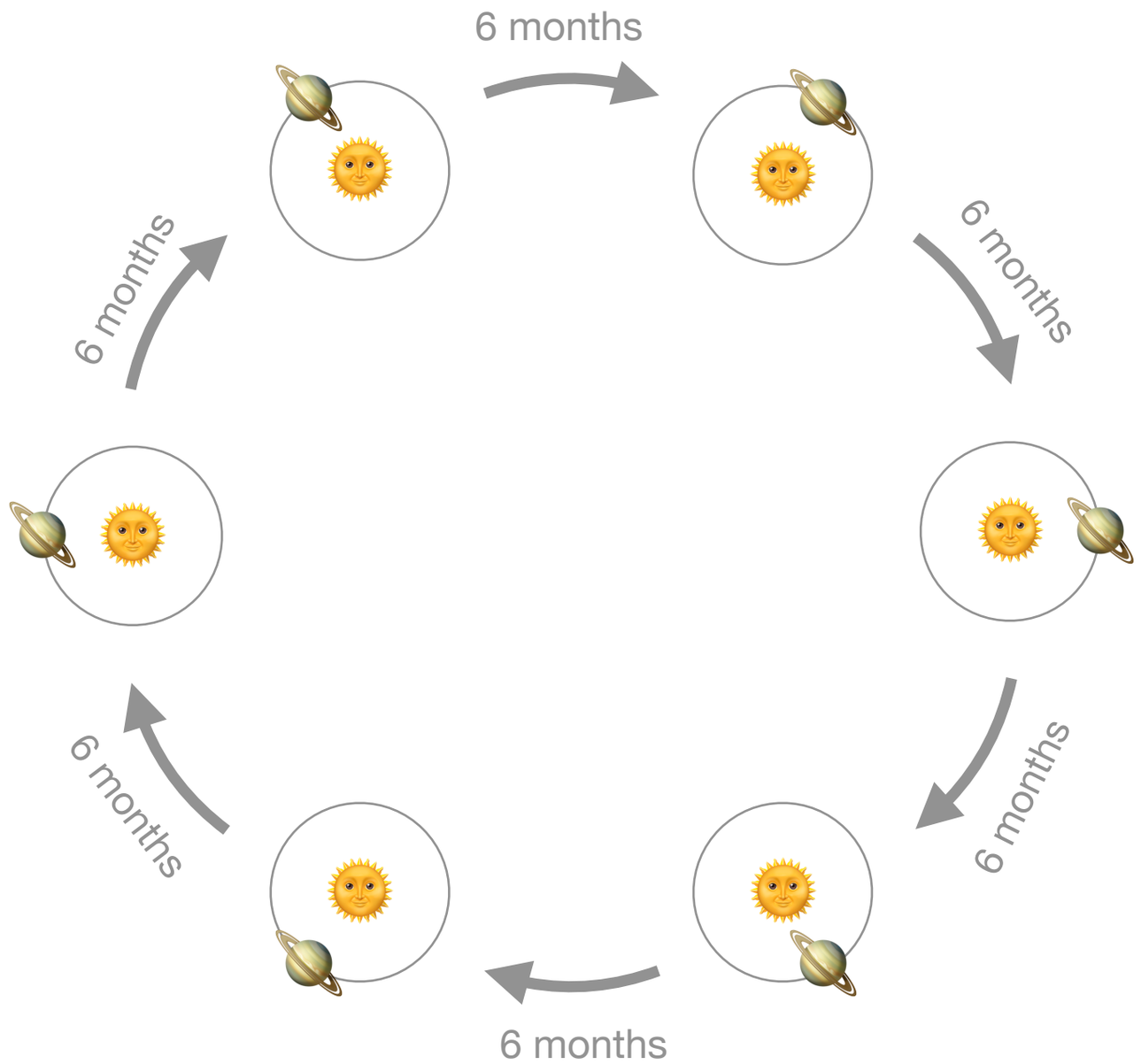


Another decomposition

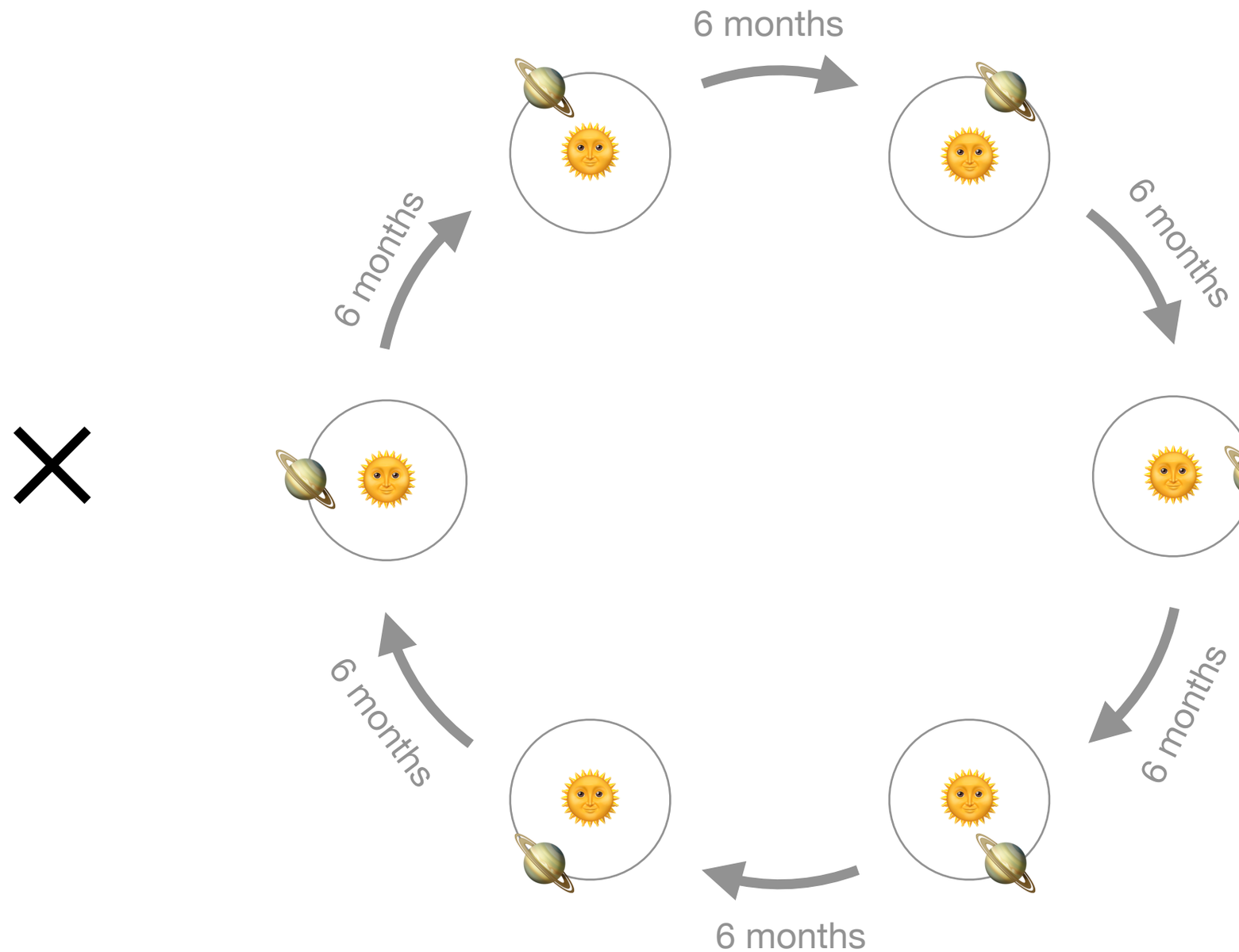


More concretely…

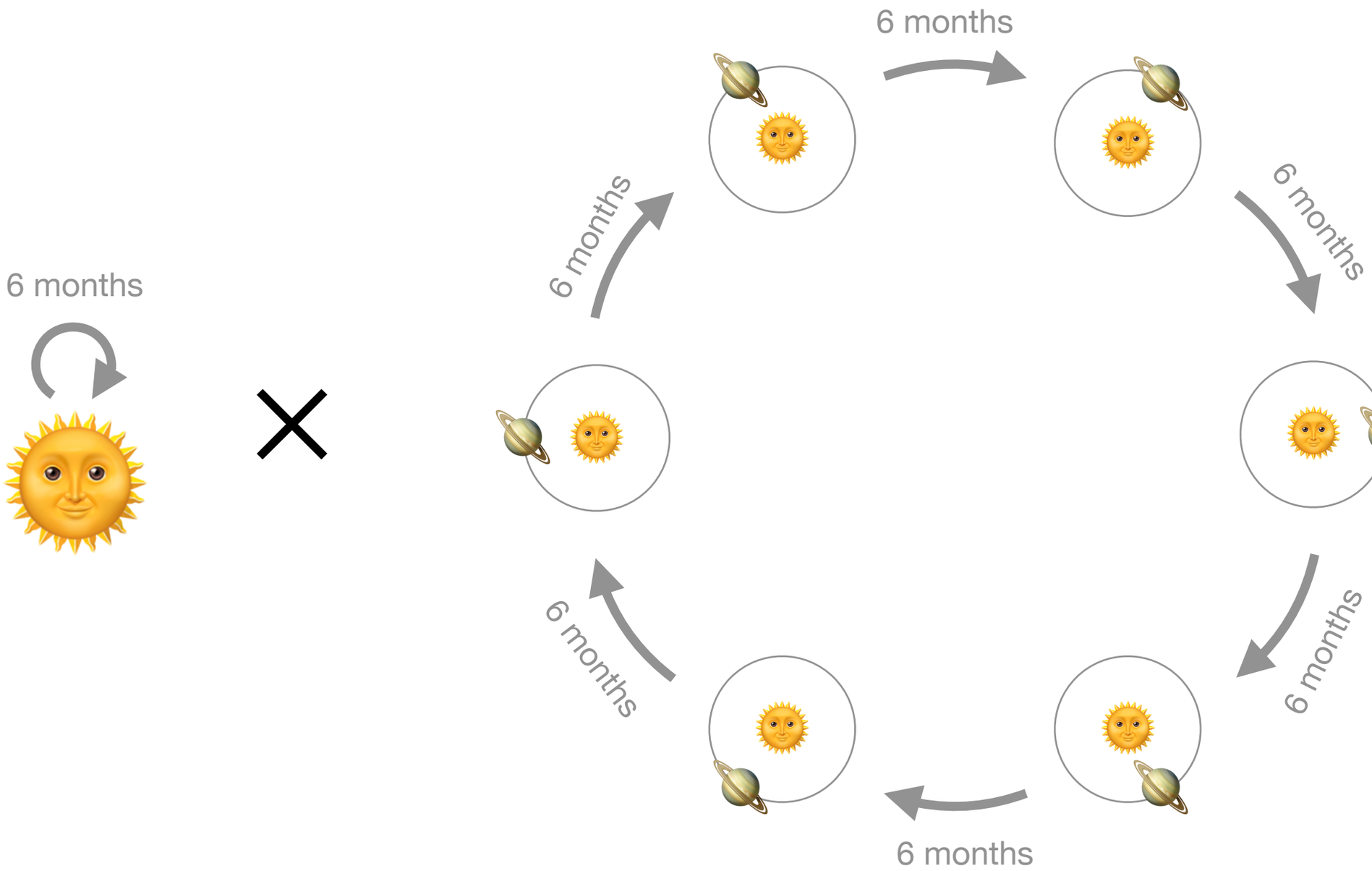
More concretely…



More concretely…

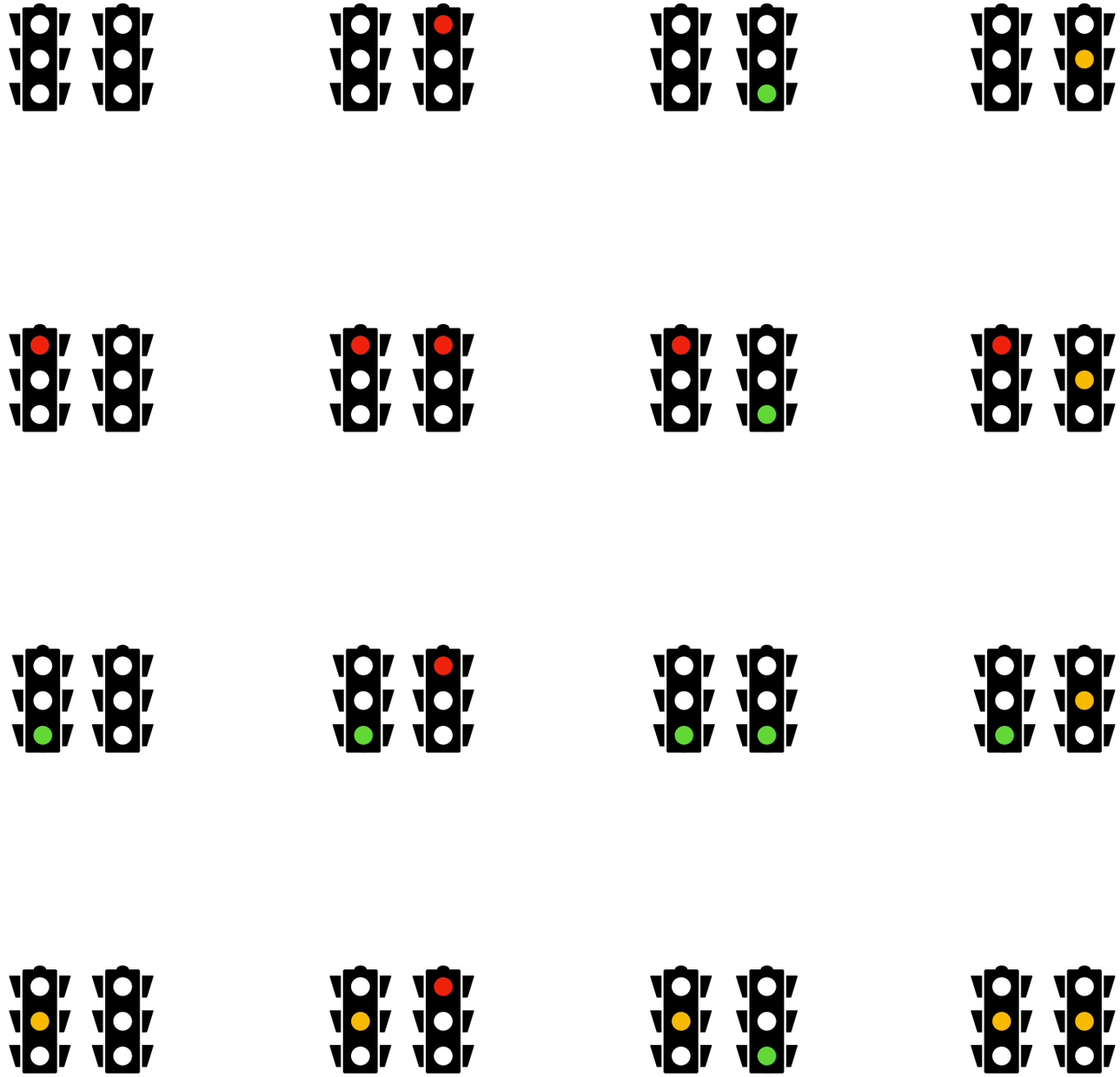


More concretely…

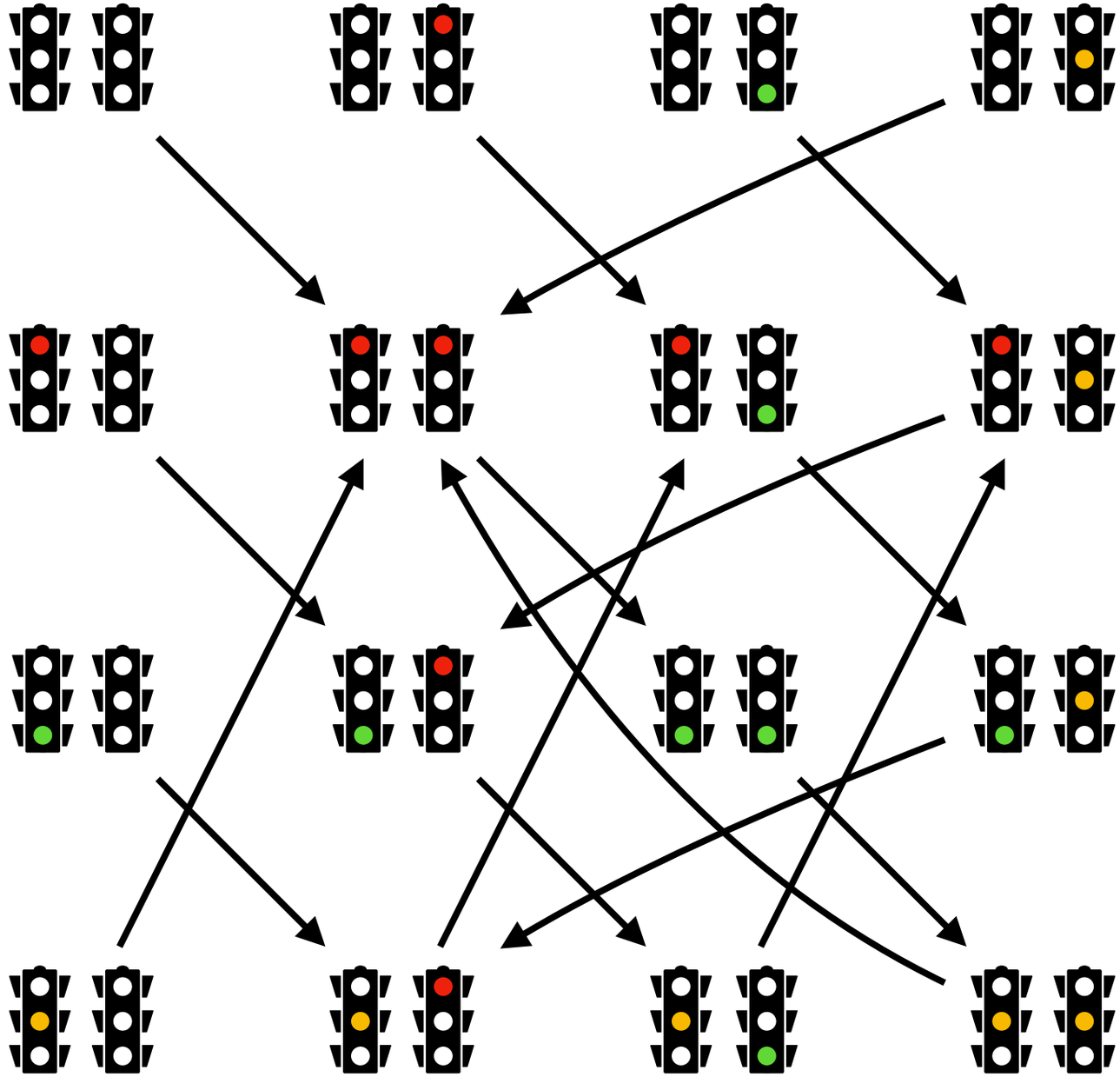


Untangling complex systems

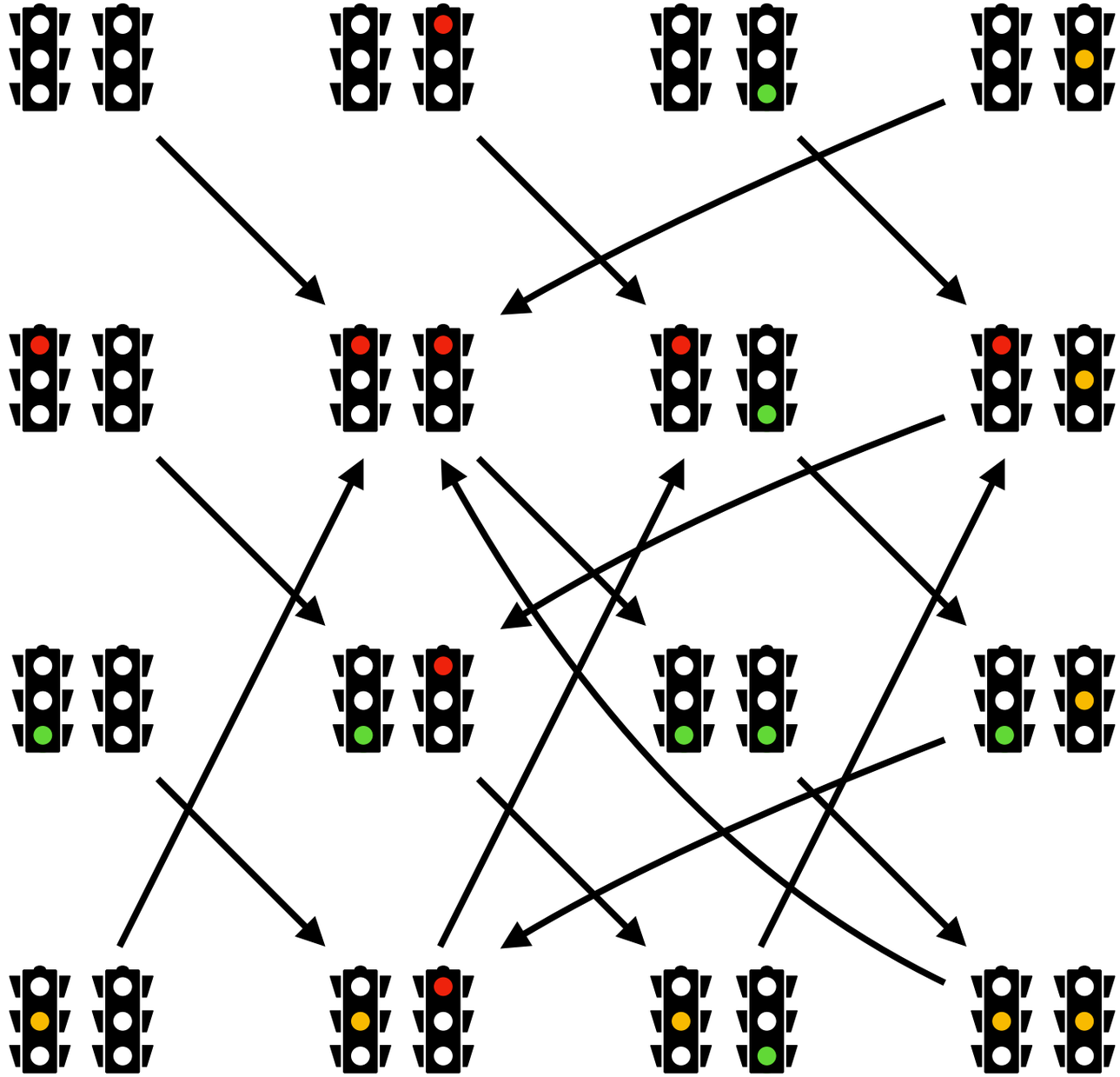
Traffic lights at a crossroads



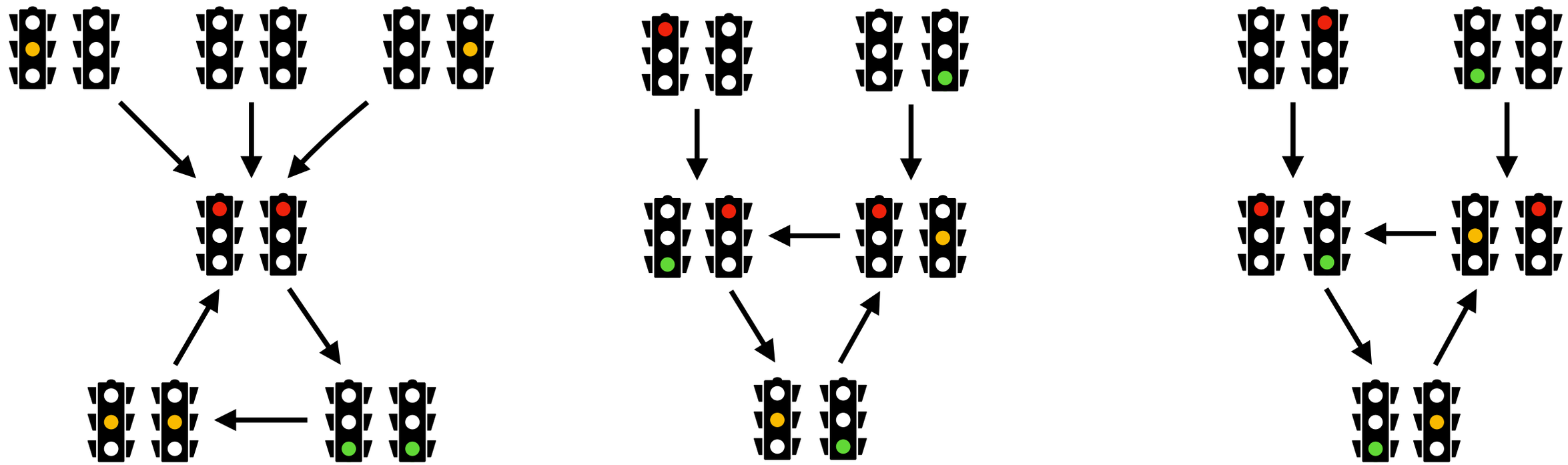
Traffic lights at a crossroads



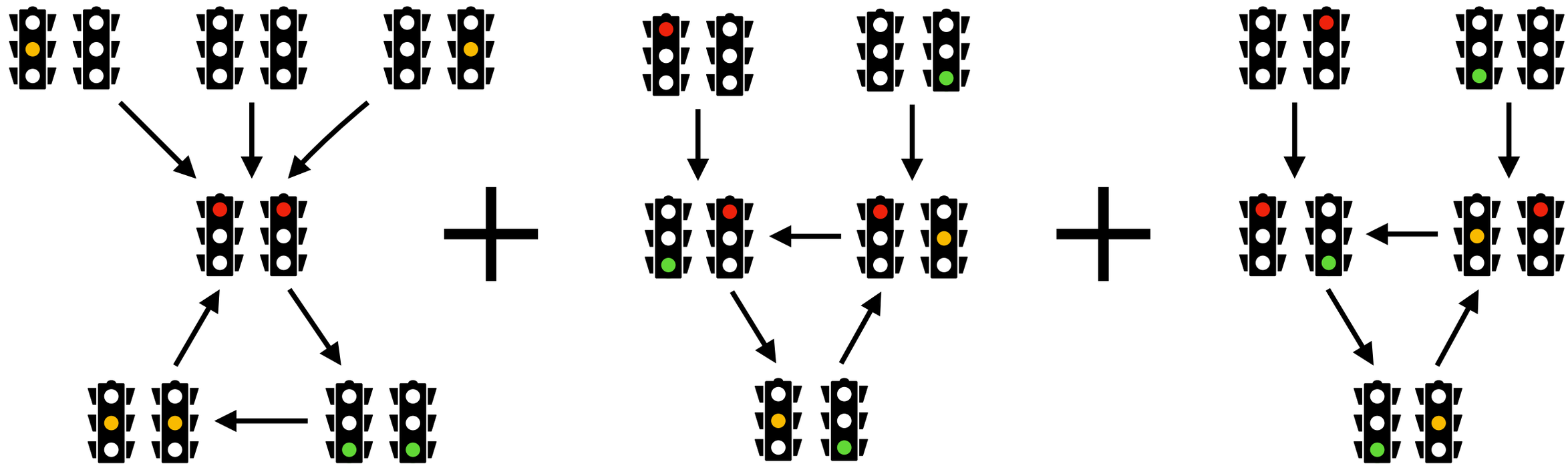
Traffic lights at a crossroads



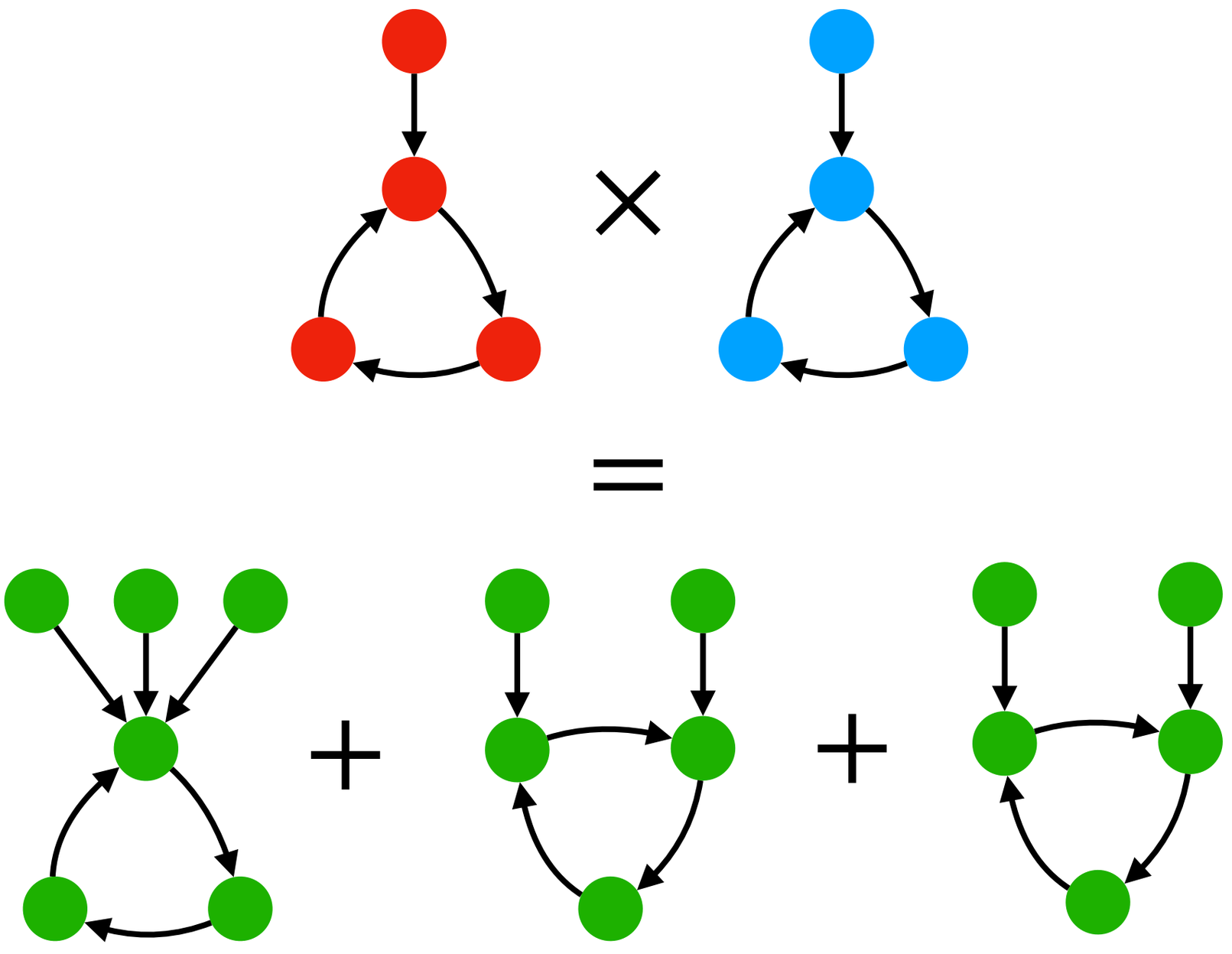
Traffic lights at a crossroads



Traffic lights at a crossroads



More abstractly…



The operations $+$ and $\times$ are a commutative semiring

The operations $+$ and $\times$ are a commutative semiring

- Commutative: $X + Y = Y + X$ and $X \times Y = Y \times X$

The operations $+$ and $\times$ are a commutative semiring

- Commutative: $X + Y = Y + X$ and $X \times Y = Y \times X$
- Associative: $X + (Y + Z) = (Y + X) + Z$ and $X \times (Y \times Z) = (Y \times X) \times Z$

The operations $+$ and $\times$ are a commutative semiring

- Commutative: $X + Y = Y + X$ and $X \times Y = Y \times X$
- Associative: $X + (Y + Z) = (Y + X) + Z$ and $X \times (Y \times Z) = (Y \times X) \times Z$
- Neutral elements: $\varnothing + X = X$ and  $\times X = X$

The operations $+$ and $\times$ are a commutative semiring

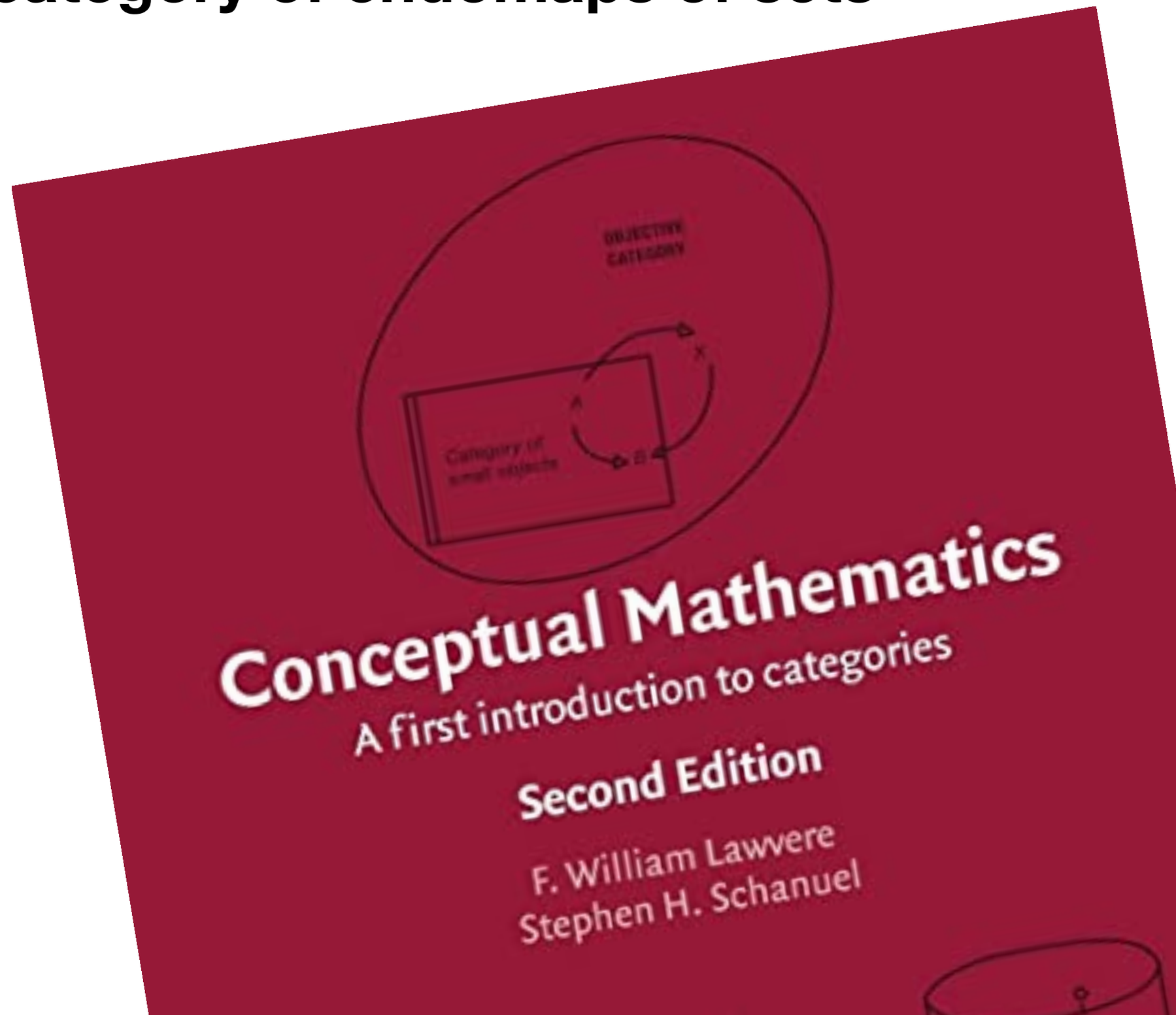
- Commutative: $X + Y = Y + X$ and $X \times Y = Y \times X$
- Associative: $X + (Y + Z) = (Y + X) + Z$ and $X \times (Y \times Z) = (Y \times X) \times Z$
- Neutral elements: $\varnothing + X = X$ and  $\times X = X$
- Distributive: $X \times (Y + Z) = X \times Y + X \times Z$

The operations $+$ and $\times$ are a commutative semiring

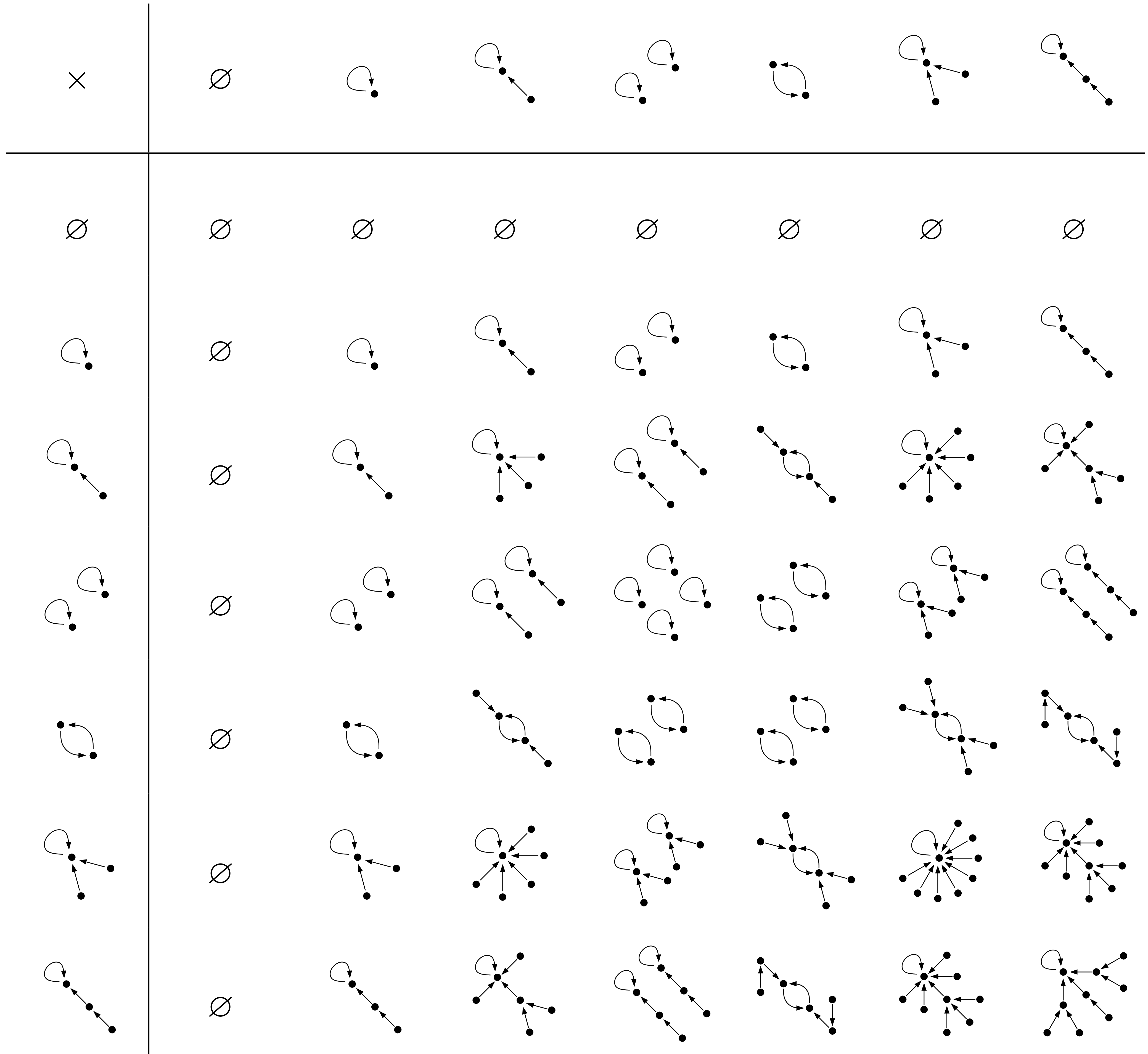
- Commutative: $X + Y = Y + X$ and $X \times Y = Y \times X$
- Associative: $X + (Y + Z) = (Y + X) + Z$ and $X \times (Y \times Z) = (Y \times X) \times Z$
- Neutral elements: $\varnothing + X = X$ and $\text{🎧} \times X = X$
- Distributive: $X \times (Y + Z) = X \times Y + X \times Z$
- Multiplication by zero: $\varnothing \times X = \varnothing$

The inspiration

The category of endomaps of sets

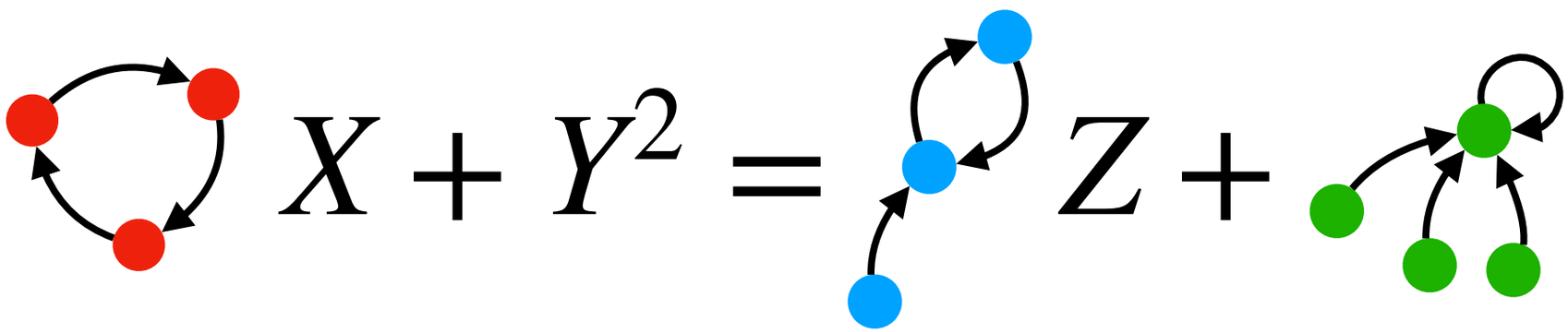


Multiplication table

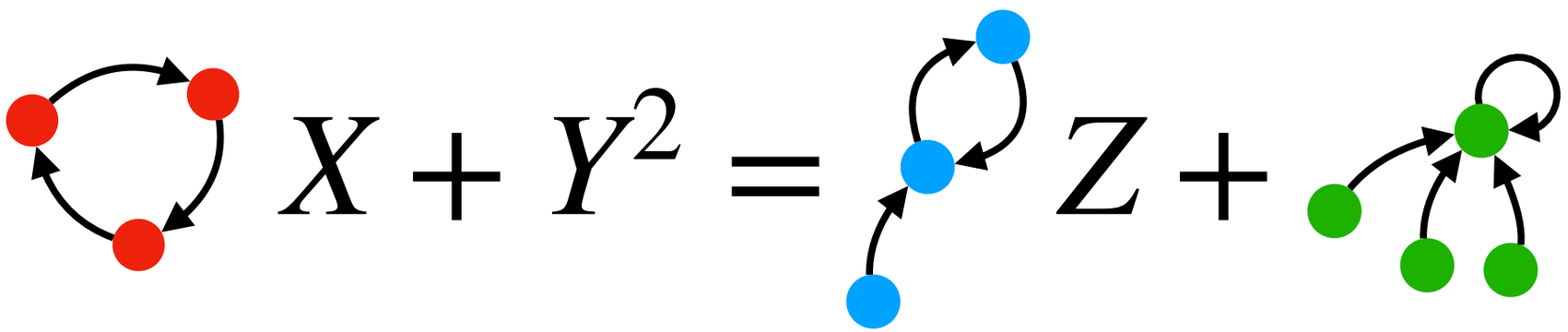


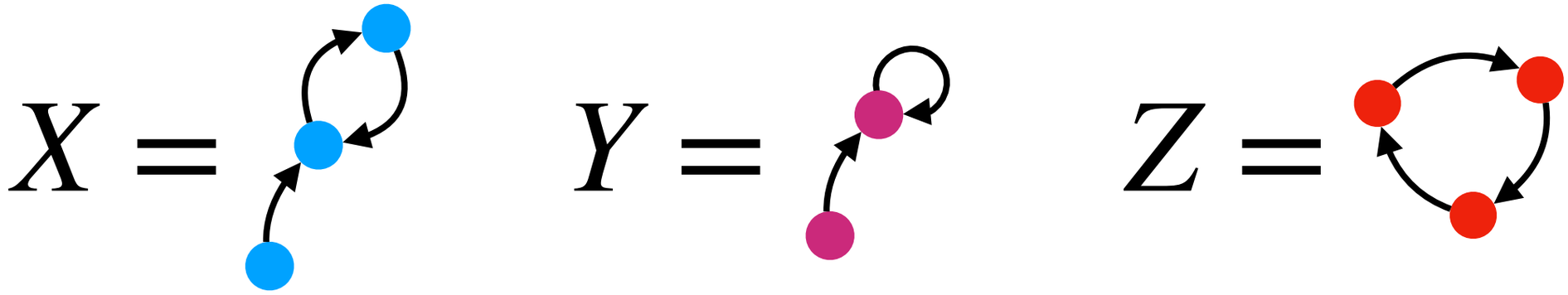
Equations for decomposing systems

Eqns over dynamical systems



Eqns over dynamical systems





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

- There is an injective homomorphism $\varphi : \mathbb{N} \to \mathbf{D}$

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

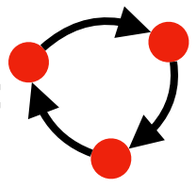
- n fixed points behave exactly as the integer n

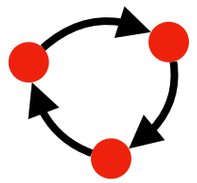
- So $\mathbf{D}$ contains a isomorphic copy of $\mathbb{N}$

Natural polynomial equations

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

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 $\mathbb{N}[\vec{X}]$…

- …and thus of arbitrary equations over $\mathbf{D}[\vec{X}]$

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

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

Proof

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

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

Proof

The coefficients are natural

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

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

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

Proof

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

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

Proof

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

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

⚠️ Bad news about solving equations ⚠️

⚠️ Bad news about solving equations ⚠️

- There is no algorithm at all for solving general equations

⚠️ Bad news about solving equations ⚠️

- There is no algorithm at all for solving general equations
- Equations without variables on one side admit an algorithm, but even linear ones of this form are **NP**-complete:

$$A_1X_1 + A_2X_2 + \cdots + A_nX_n = B$$

⚠️ Bad news about solving equations ⚠️

- There is no algorithm at all for solving general equations
- Equations without variables on one side admit an algorithm, but even linear ones of this form are **NP**-complete:

$$A_1X_1 + A_2X_2 + \cdots + A_nX_n = B$$

- We are still unsure if equations in one single variable, like $AX = B$, can be solved efficiently (conjecture: no)

Reducibility of dynamical systems

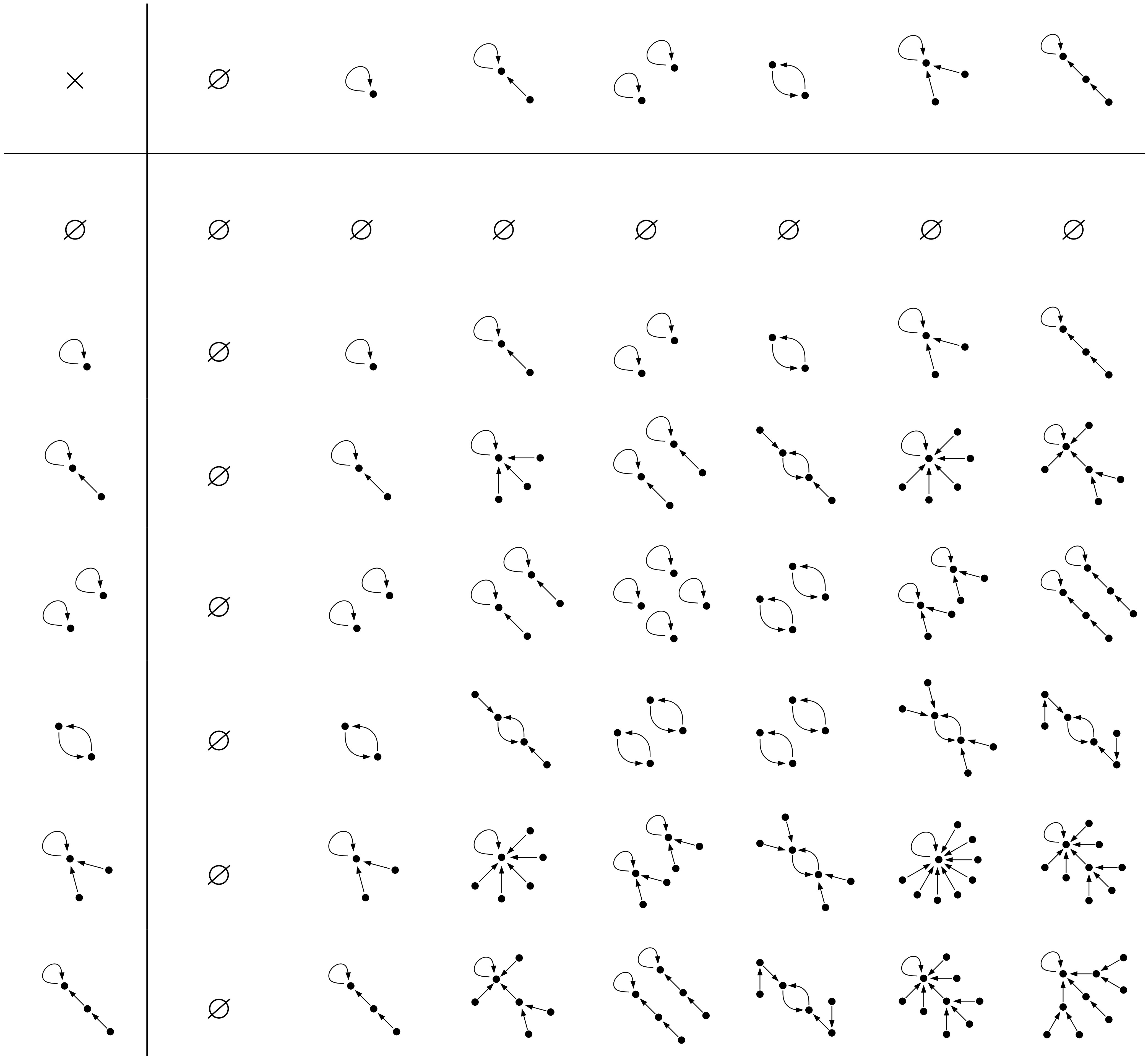
Most dynamical systems are irreducible

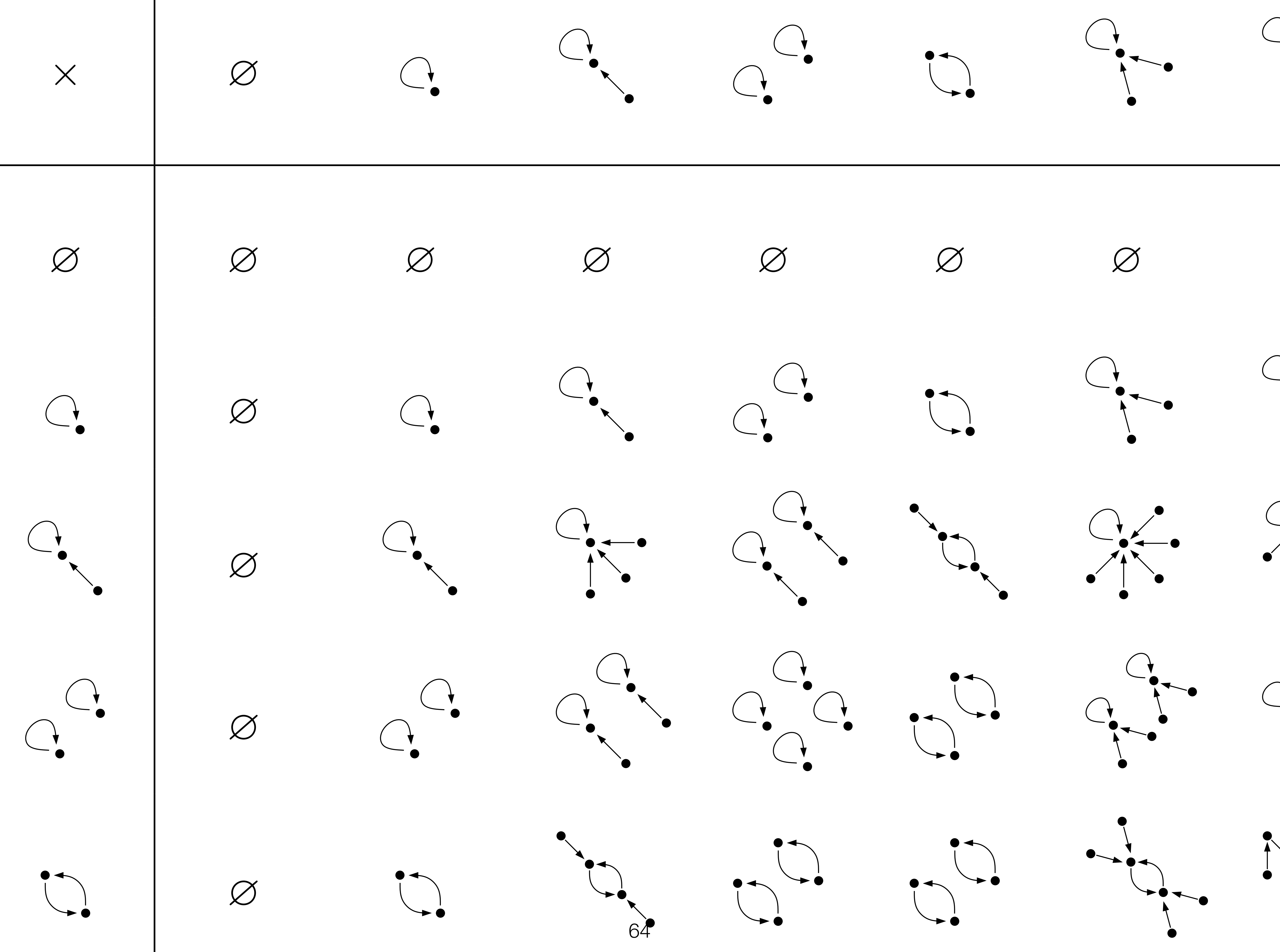
- Formally:

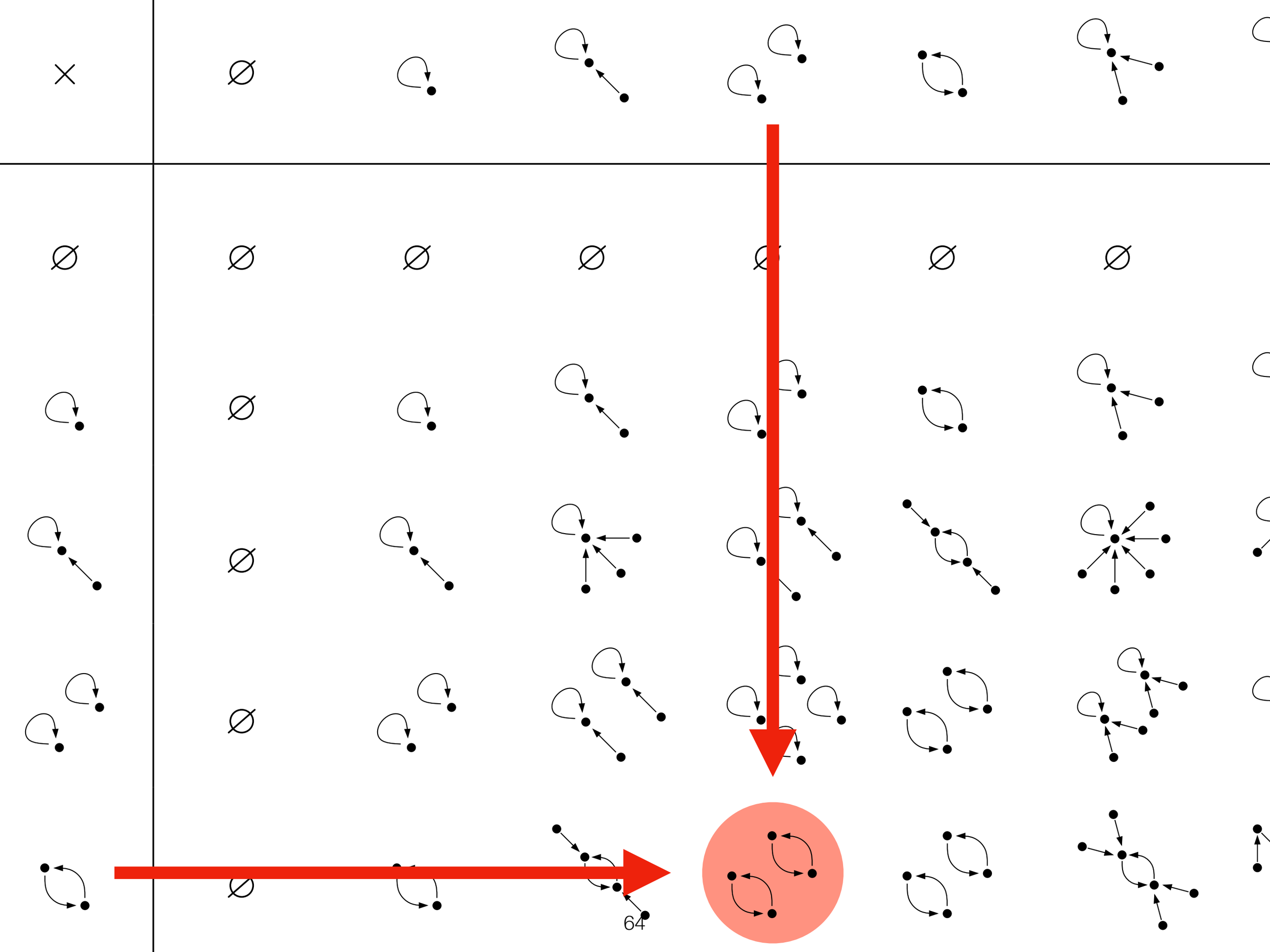
$$\lim_{n\to\infty} \frac{\text{number of reducible systems over} \leq n \text{ states}}{\text{total number of systems over} \leq n \text{ states}} = 0$$

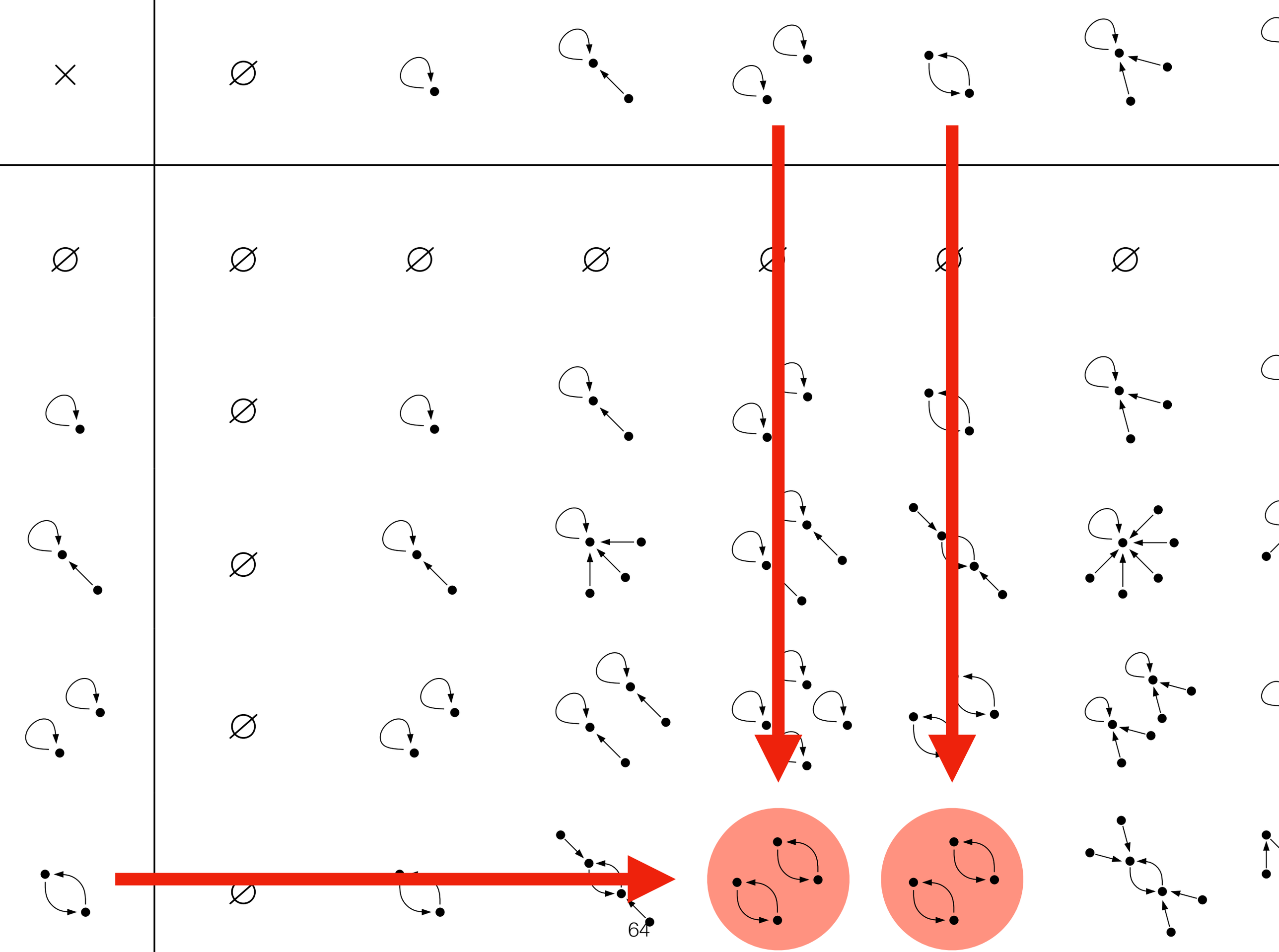
- Notice that this is the opposite of $\mathbb{N}$, where irreducible (aka prime) integers are scarce

No unique factorisation into irreducibles!

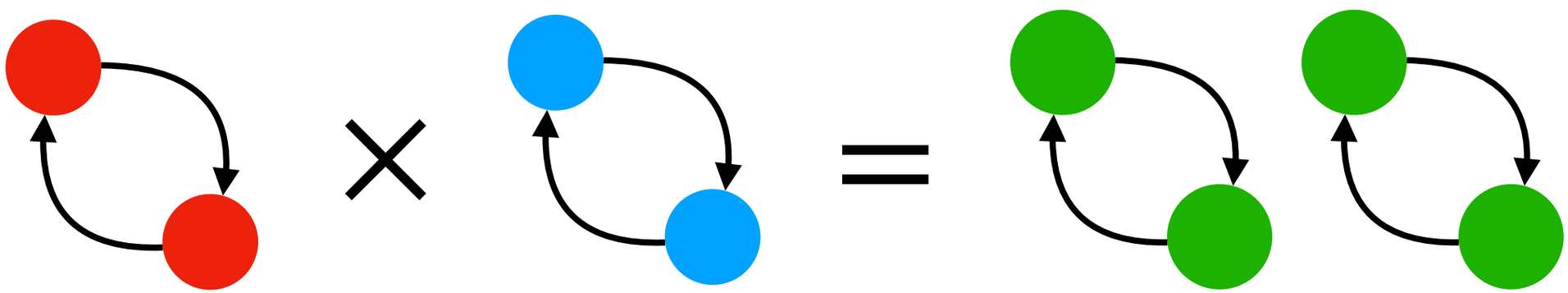


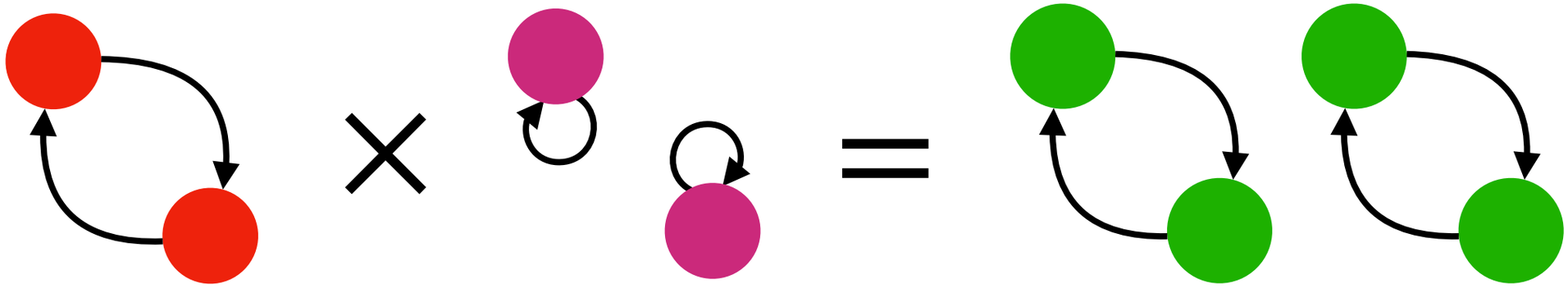




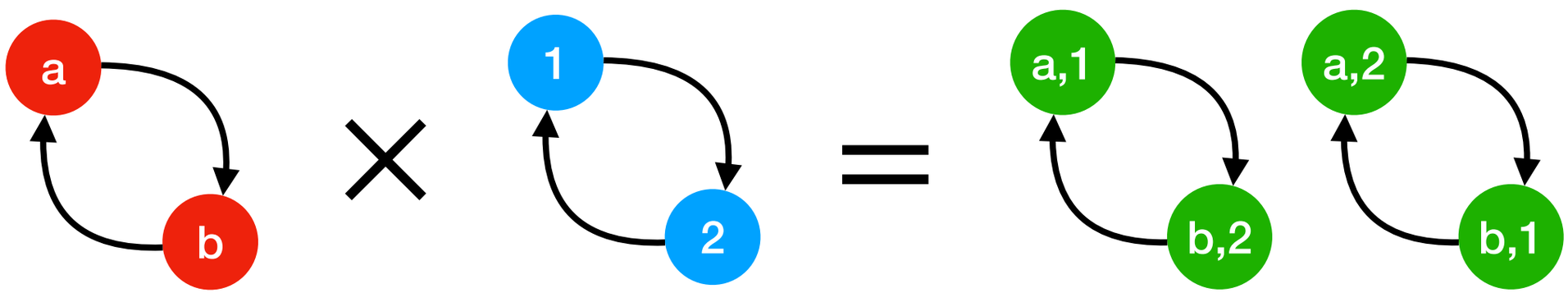


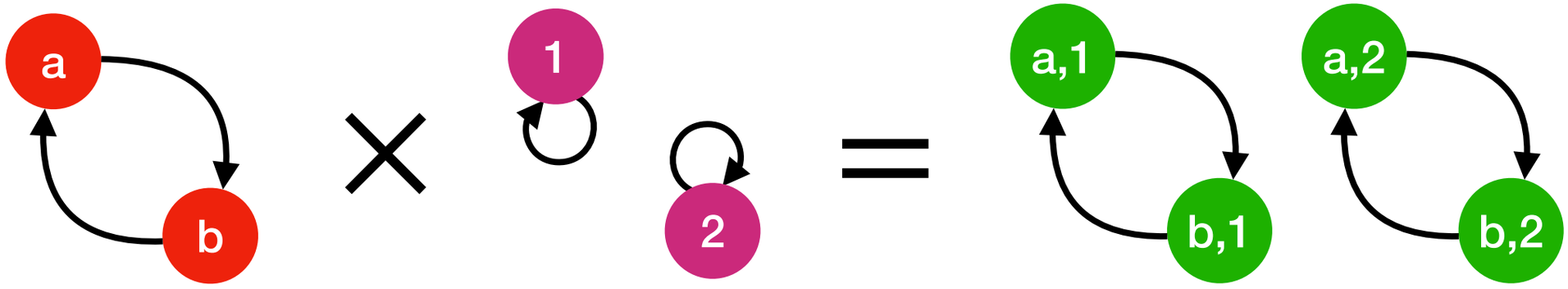
Multiple factorisations



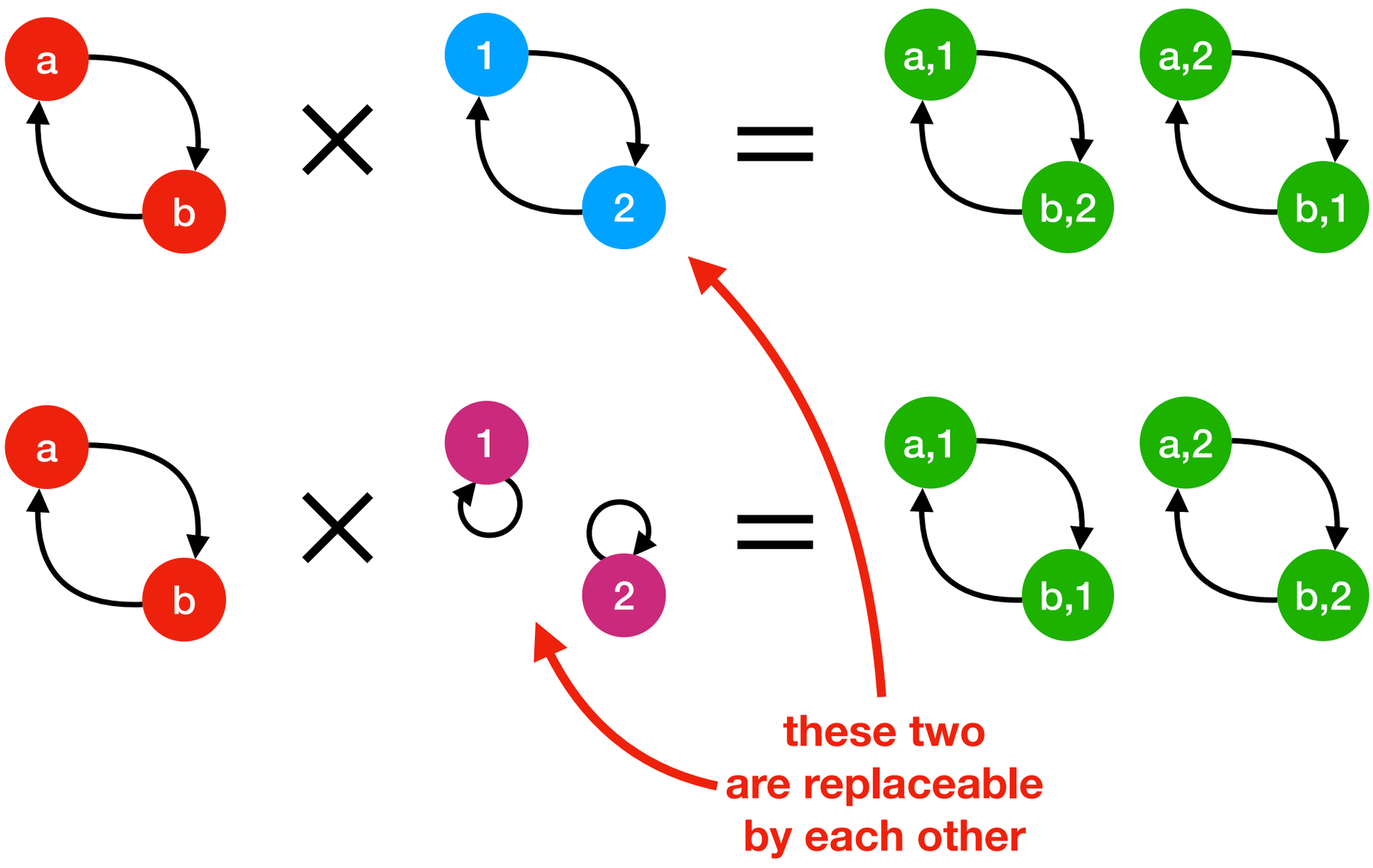


Multiple factorisations





Multiple factorisations



Prime systems

Prime systems

- A prime is a system P such that, whenever it appears in a factorisation into irreducibles of $A \times B$,
  it appears in the factorisation of either A or B

Prime systems

- A prime is a system P such that, whenever it appears in a factorisation into irreducibles of $A \times B$, it appears in the factorisation of either A or B

- In other words, if P divides $A \times B$ then it divides A or B

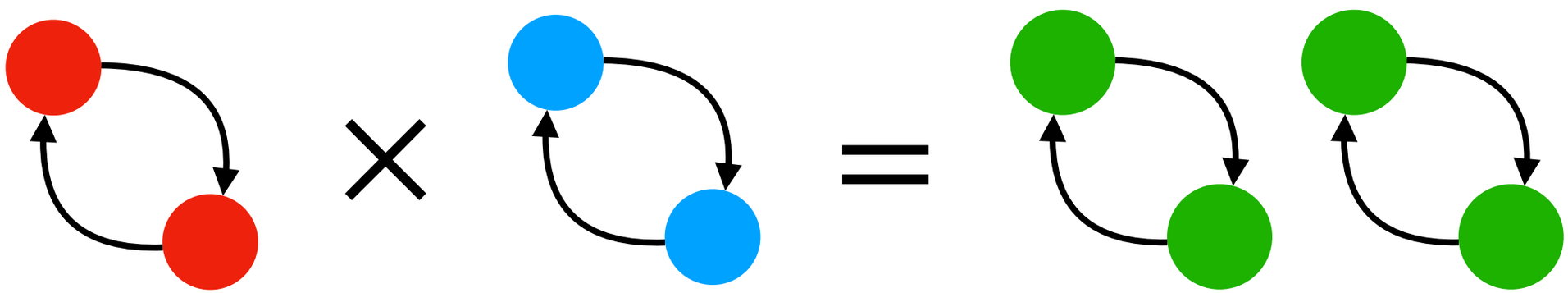
Prime systems

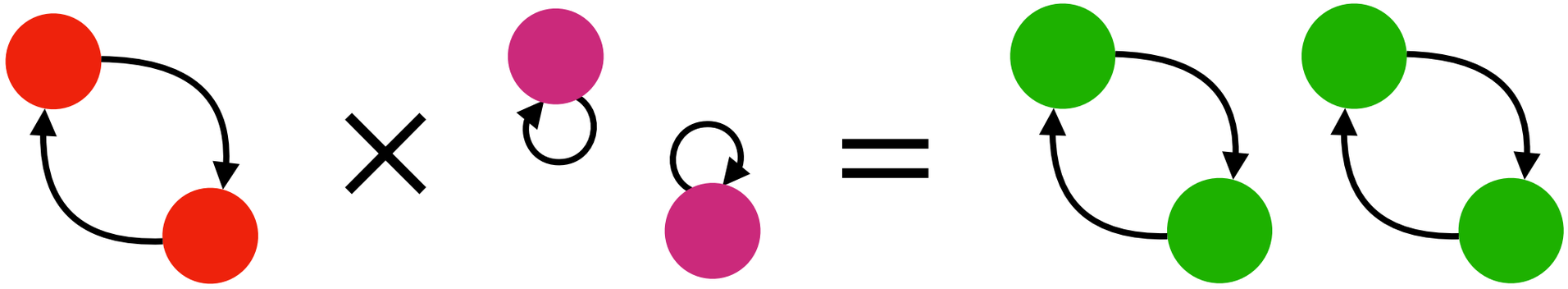
- A prime is a system P such that, whenever it appears in a factorisation into irreducibles of $A \times B$, it appears in the factorisation of either A or B

- In other words, if P divides $A \times B$ then it divides A or B

- If a prime appears in one factorisation of a system, then it appears in all the others as well (it is irreplaceable)

An example of nonprime





Finding primes

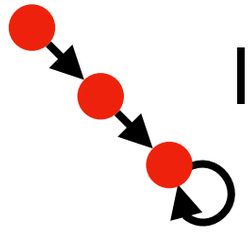
Finding primes

- We haven’t been able to find even a single prime yet!

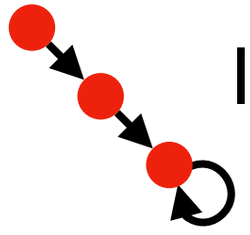
Finding primes

- We haven’t been able to find even a single prime yet!
- We have found infinitely many non-primes though

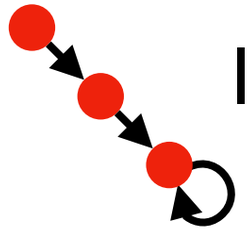
Finding primes

- We haven’t been able to find even a single prime yet!
- We have found infinitely many non-primes though
- This guy here?  It took us two years to find out that it is not prime

Finding primes

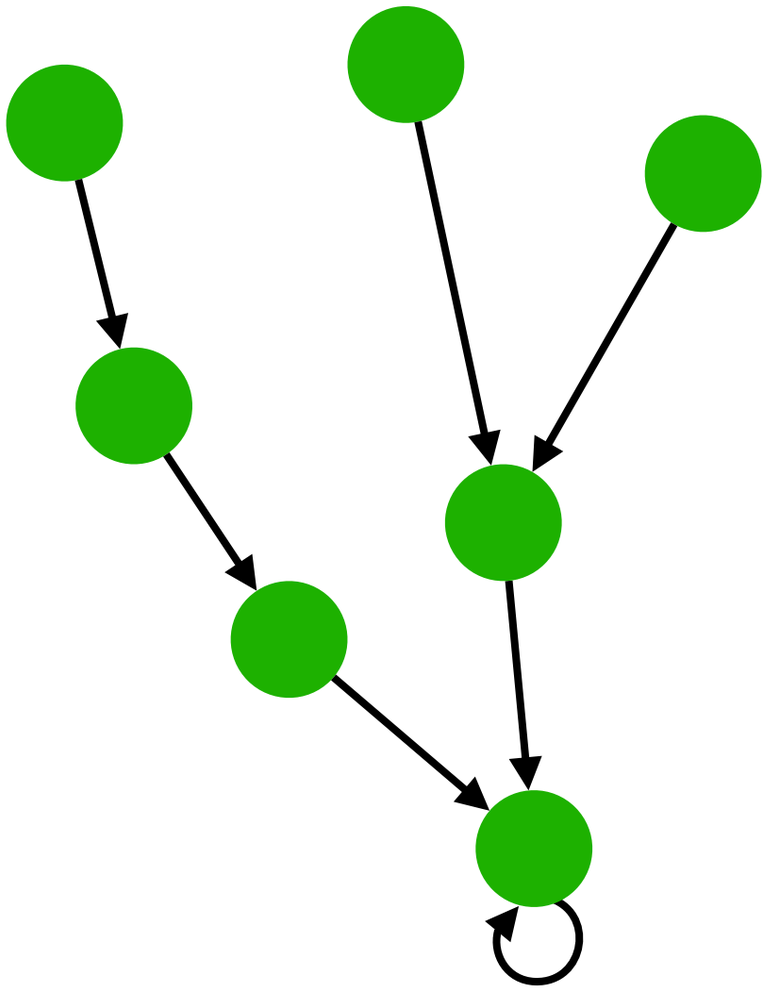
- We haven’t been able to find even a single prime yet!
- We have found infinitely many non-primes though
- This guy here?  It took us two years to find out that it is not prime
- A counterexample to the primality of P is two systems A, B such that P divides $A \times B$ but neither A nor B

Finding primes

- We haven’t been able to find even a single prime yet!
- We have found infinitely many non-primes though
- This guy here?  It took us two years to find out that it is not prime
- A counterexample to the primality of P is two systems A, B such that P divides $A \times B$ but neither A nor B
- Those A and B can be bigger than P, but we don’t know how much, so no algorithm yet…

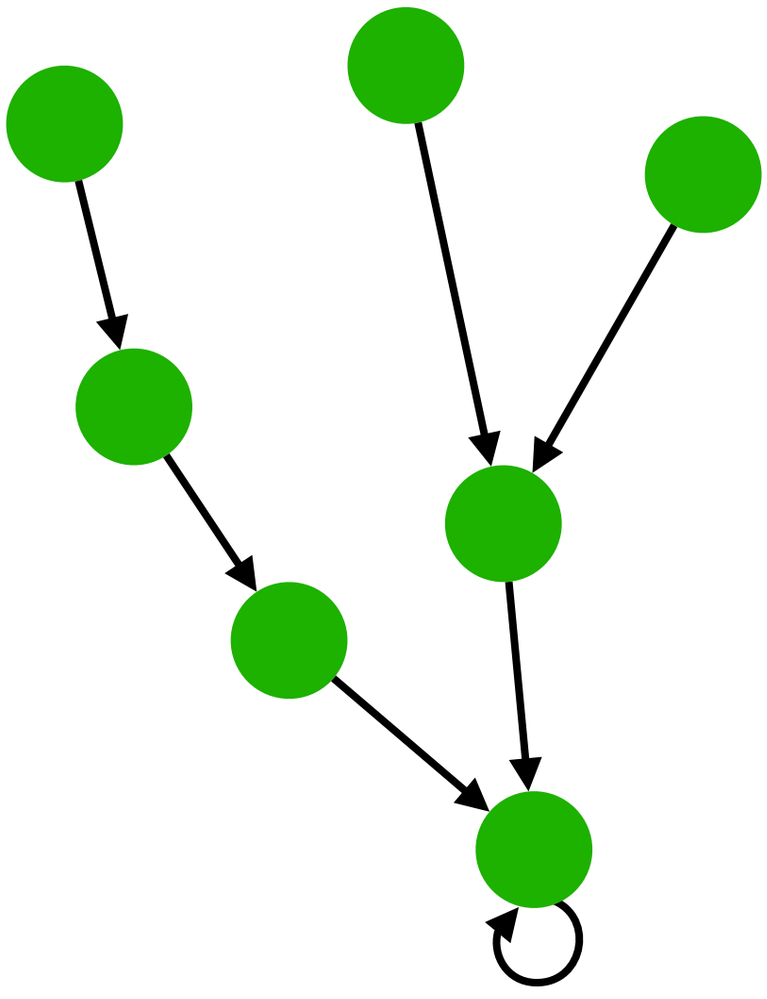
What do primes look like, if they exist at all?

What do primes look like, if they exist at all?



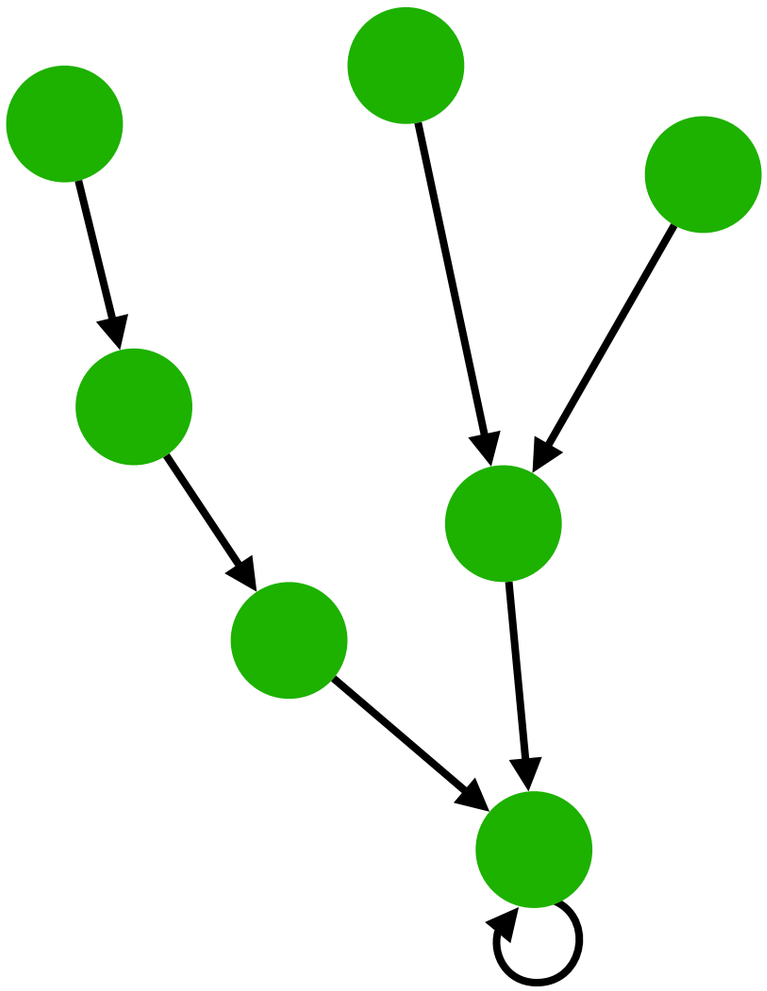
What do primes look like, if they exist at all?

- Connected



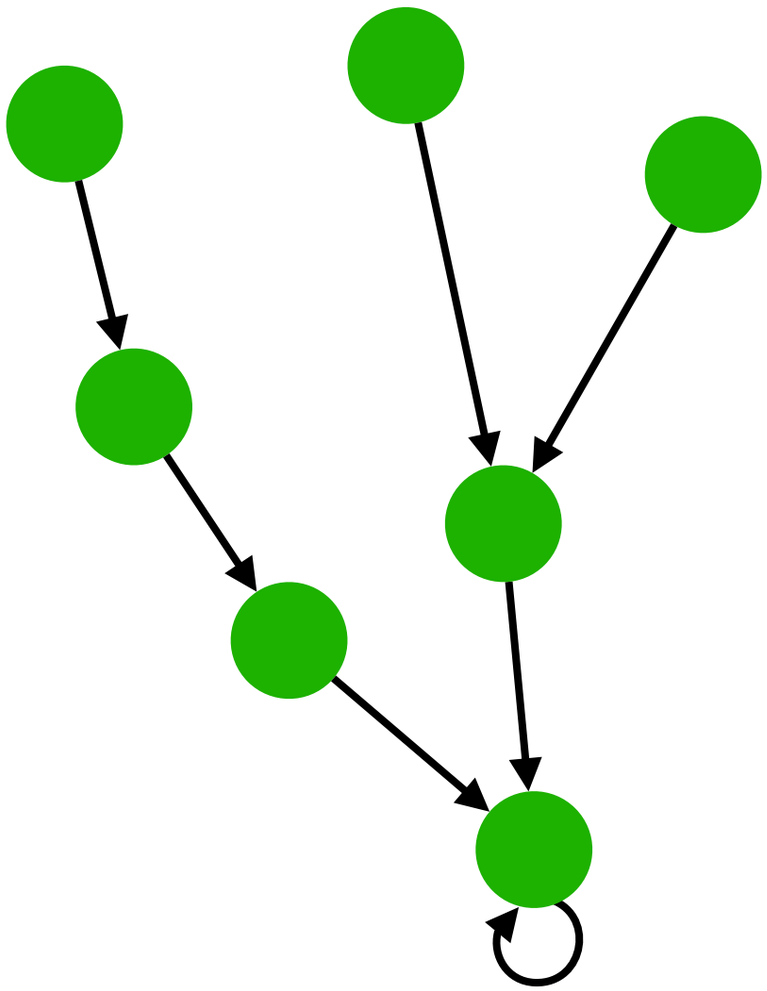
What do primes look like, if they exist at all?

- Connected
- Fixed point (no cycles of length > 1)



What do primes look like, if they exist at all?

- Connected
- Fixed point (no cycles of length > 1)
- gcd of the number of predecessors across all states must be 1



Future developments

Future developments

- Find more solvable equations, and at least one class of equations that is solvable efficiently

Future developments

- Find more solvable equations, and at least one class of equations that is solvable efficiently
- How many solutions for a given equation?
  E.g., $AX = B$ has at most one if X is connected (recent result by É. Naquin and M. Gadouleau)

Future developments

- Find more solvable equations, and at least one class of equations that is solvable efficiently

- How many solutions for a given equation?
  E.g., $AX = B$ has at most one if X is connected
  (recent result by É. Naquin and M. Gadouleau)

- Find out if prime systems exist, or at least find a primality algorithm!

Bibliography

- A. Dennunzio, V. Dorigatti, E. Formenti, L. Manzoni, A.E. Porreca, Polynomial equations over finite, discrete-time dynamical systems

- F. Doré, E. Formenti, A.E. Porreca, S. Riva, Algorithmic reconstruction of discrete dynamics (and its bibliography)

- É. Naquin, M. Gadouleau, Factorisation in the semiring of finite dynamical systems

- A.E. Porreca, E. Timofeeva, Polynomial-delay generation of functional digraphs up to isomorphism

Thanks for your attention!

Merci de votre attention !

Any questions?