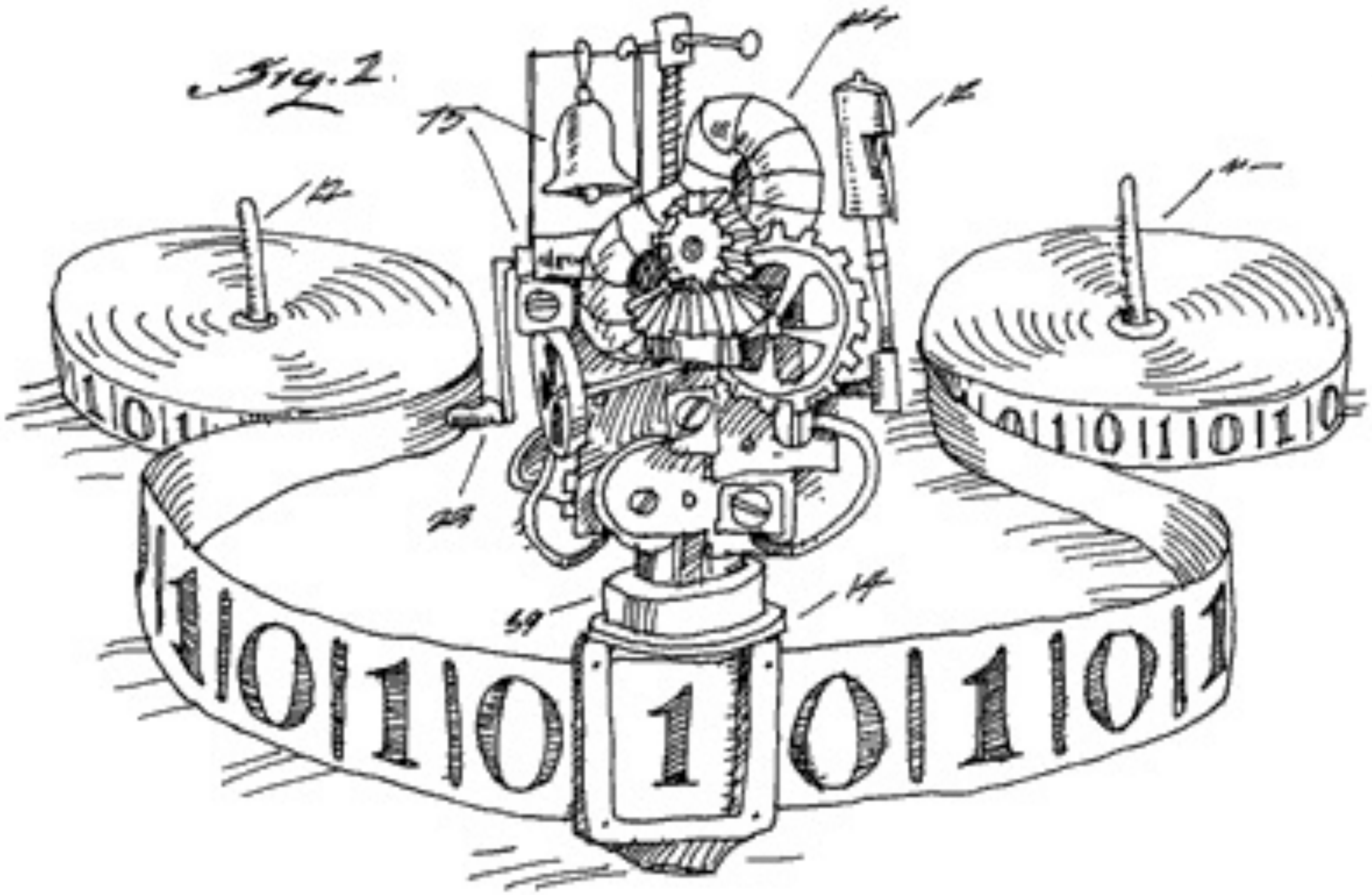
42 Mo dèles de
20 Ca lcul
11 Na turel

AntoNio E. PorReca
aeporreca.org/mocana

Modèles de calcul traditionnels

Machines de Turing



Alan Turing (1912–1954)

Machine de Turing = calculateur humain avec papier et crayon

Calculateurs humains

NACA (Comité consultatif national
pour l’aéronautique), USA, 1950s

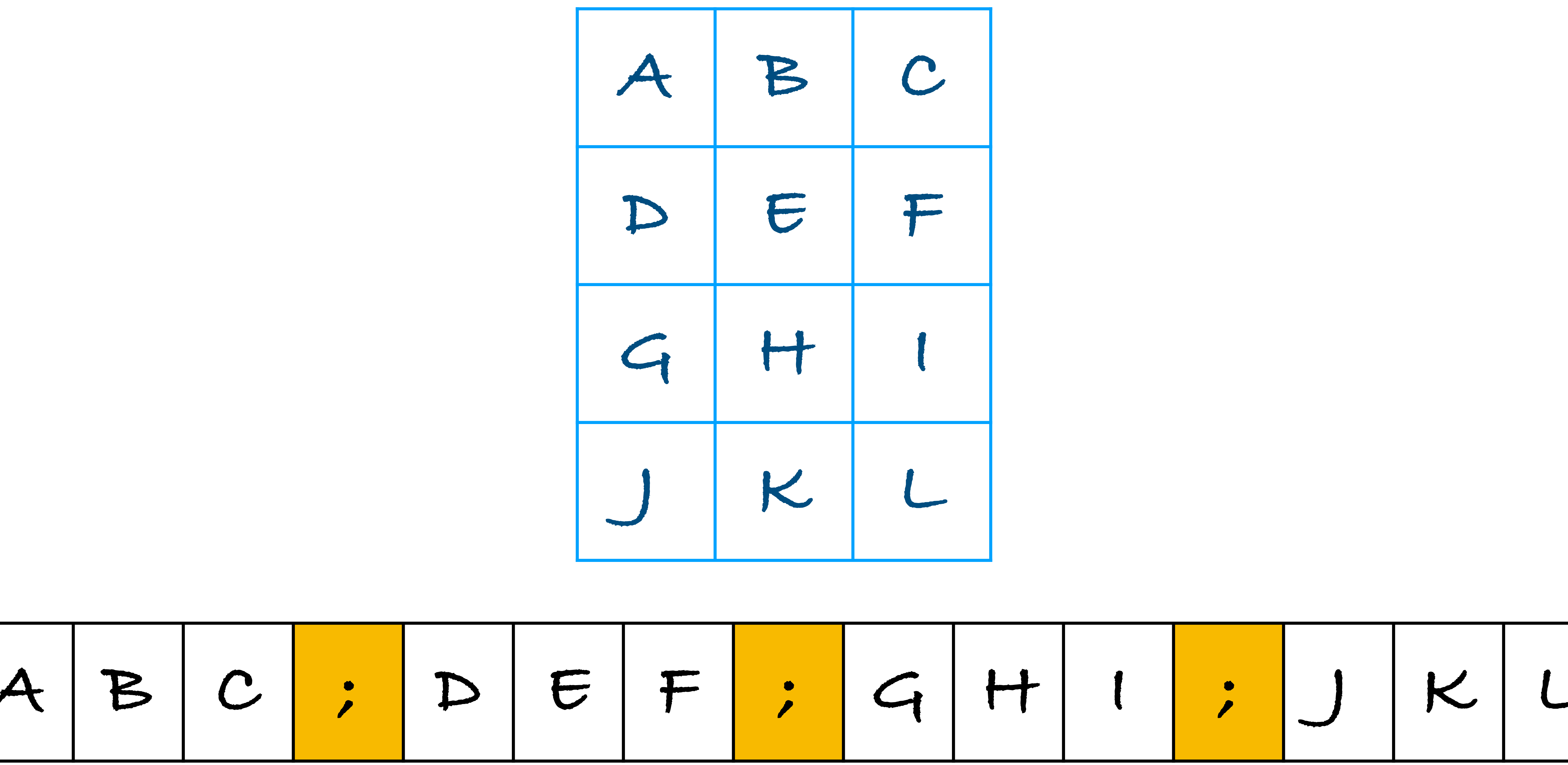
« Normalement on calcule en écrivant certains symboles sur le papier. […] Je considère qu'on effectue le calcul sur un papier unidimensionnel, c'est-à-dire, sur un ruban divisé en carrés. »

– Alan M. Turing, *On computable numbers*

Papier 2D vs ruban 1D

A	B	C
D	E	F
G	H	I
J	K	L

Papier 2D vs ruban 1D



Papier 2D vs ruban 1D

A	B	C
D	E	F
G	H	I
J	K	L

M	N	O
P	Q	R
S	T	U
V	W	X

E	F	;	G	H	I	;	J	K	L	:	M	N	O	;	P	Q	R	;	S

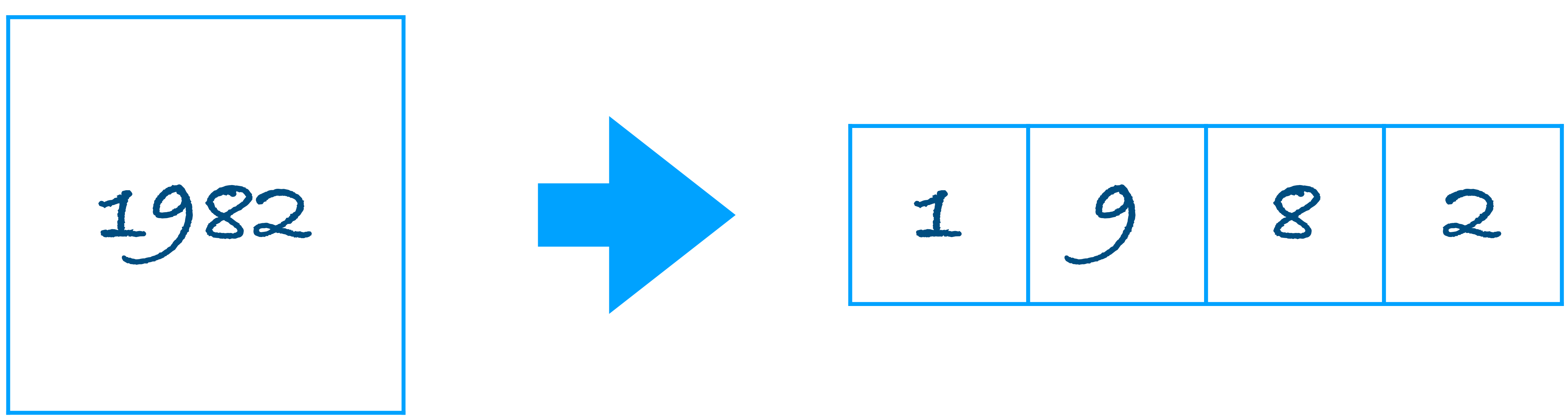
« Je suppose aussi que le nombre de symboles qu’on peut écrire soit fini. Si on permettait une infinité de symboles, il y aurait des symboles qui diffèrent dans une mesure arbitrairement faible [...] On peut toujours utiliser une séquence de symboles au lieu d’un symbole simple. »

– Alan M. Turing, *On computable numbers*

Symboles atomiques vs composites

1982

Symboles atomiques vs composites



« La différence, de notre point de vue, entre les symboles simples et composites est qu'on ne peut pas observer les symboles composites en un coup d'œil, s'ils sont trop longs. Cela est conforme à l'expérience. On ne peut pas établir en un coup d'œil si 9999999999999999 et 999999999999999 sont égales. »

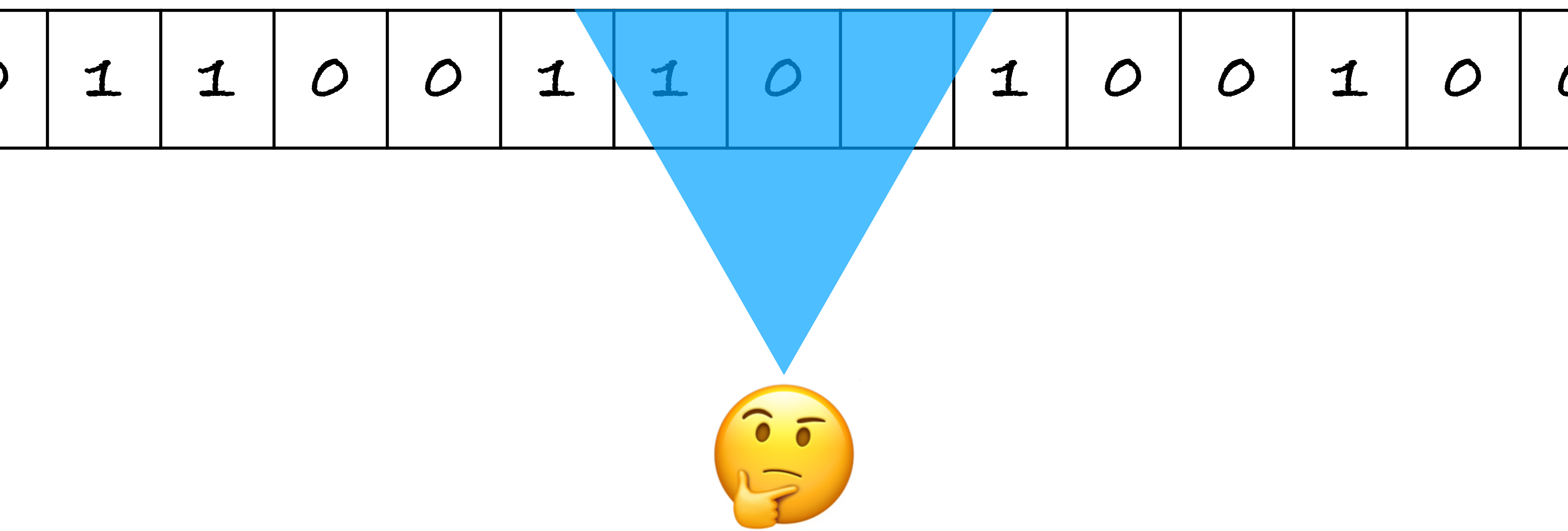
– Alan M. Turing, *On computable numbers*

« Champ visuel »

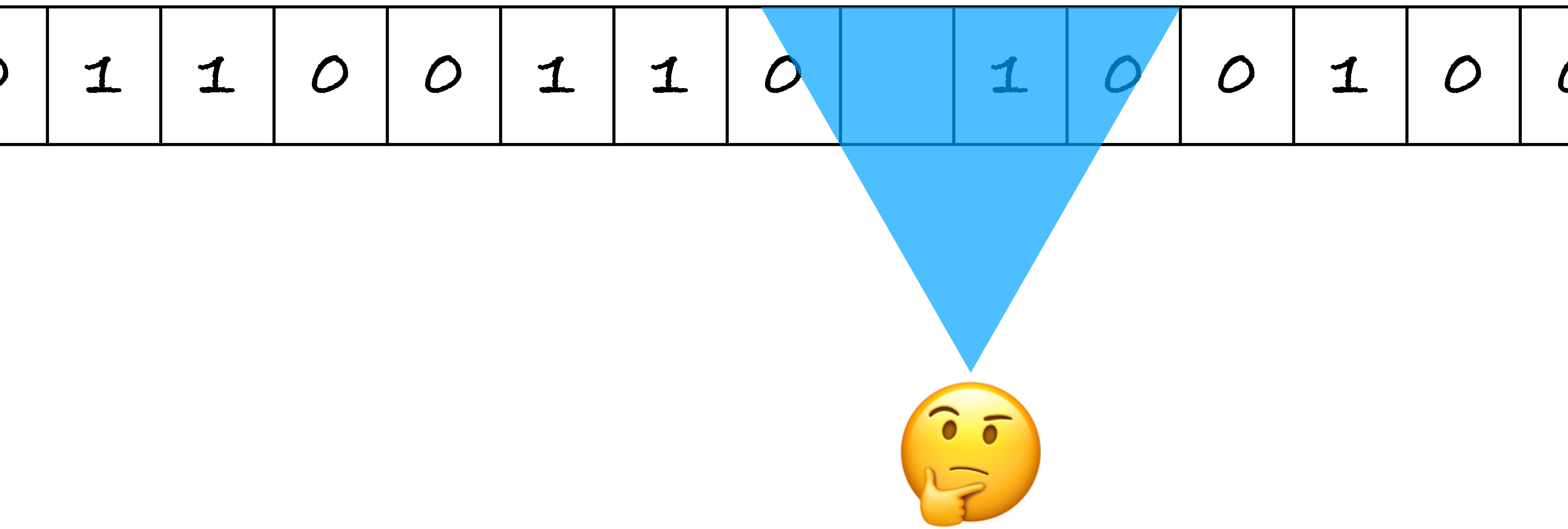
0	1	1	0	0	1	1	0		1	0	0	1	0	0

🤔

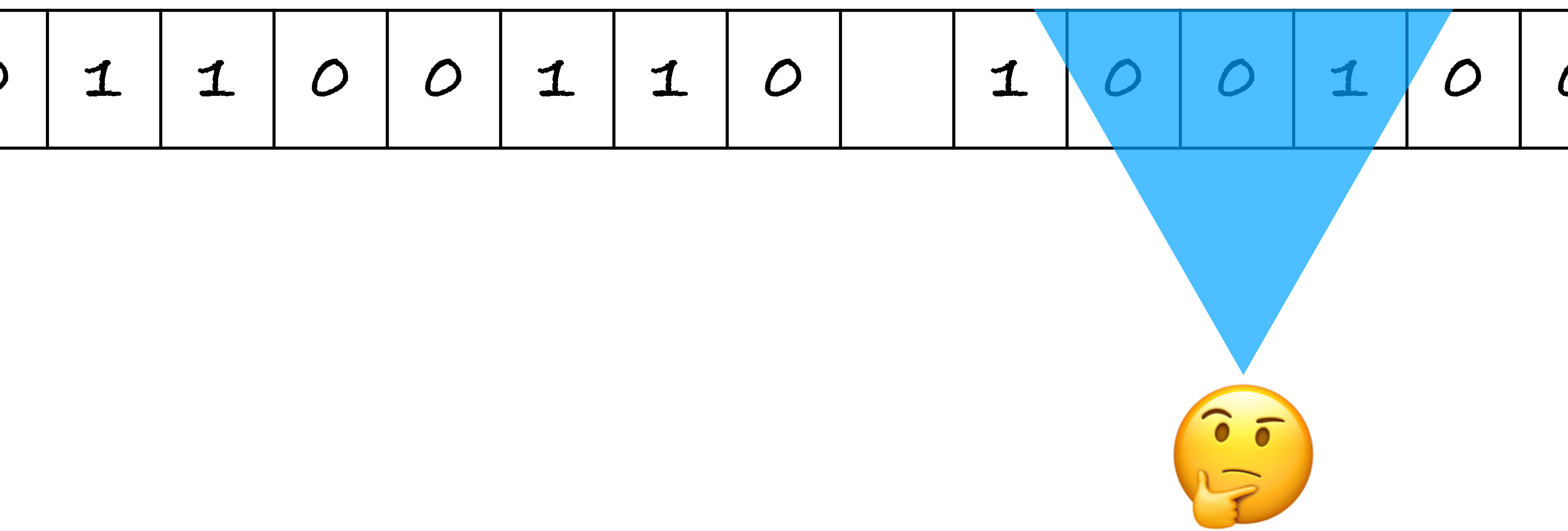
« Champ visuel »



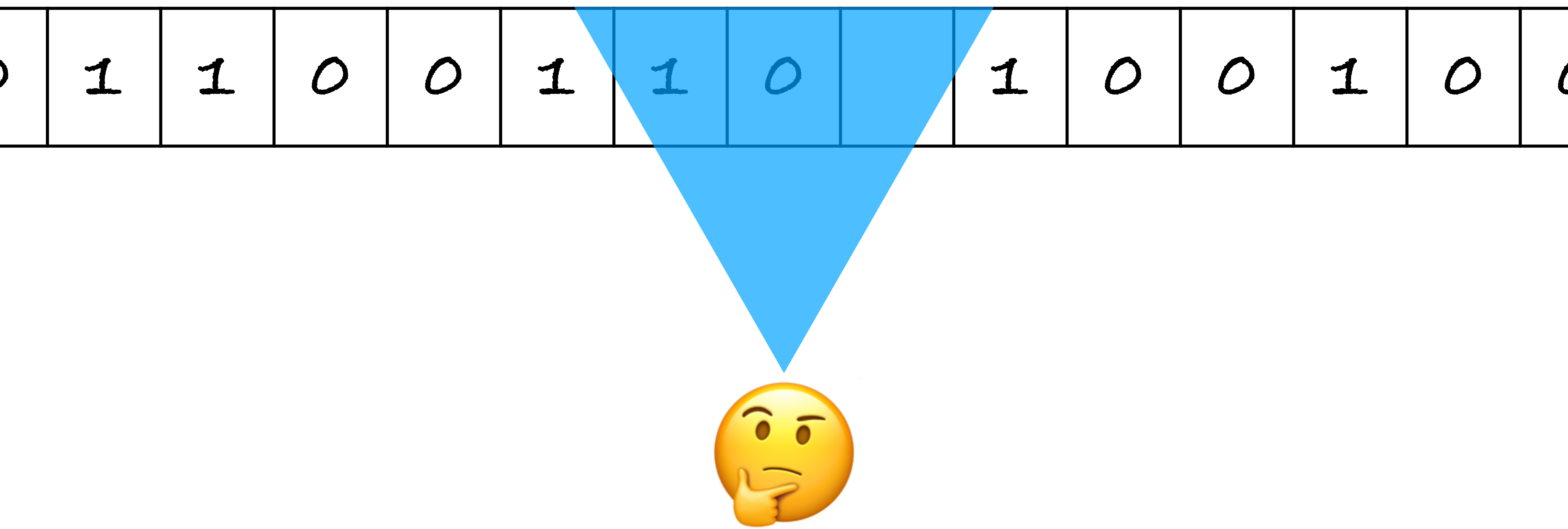
« Champ visuel »



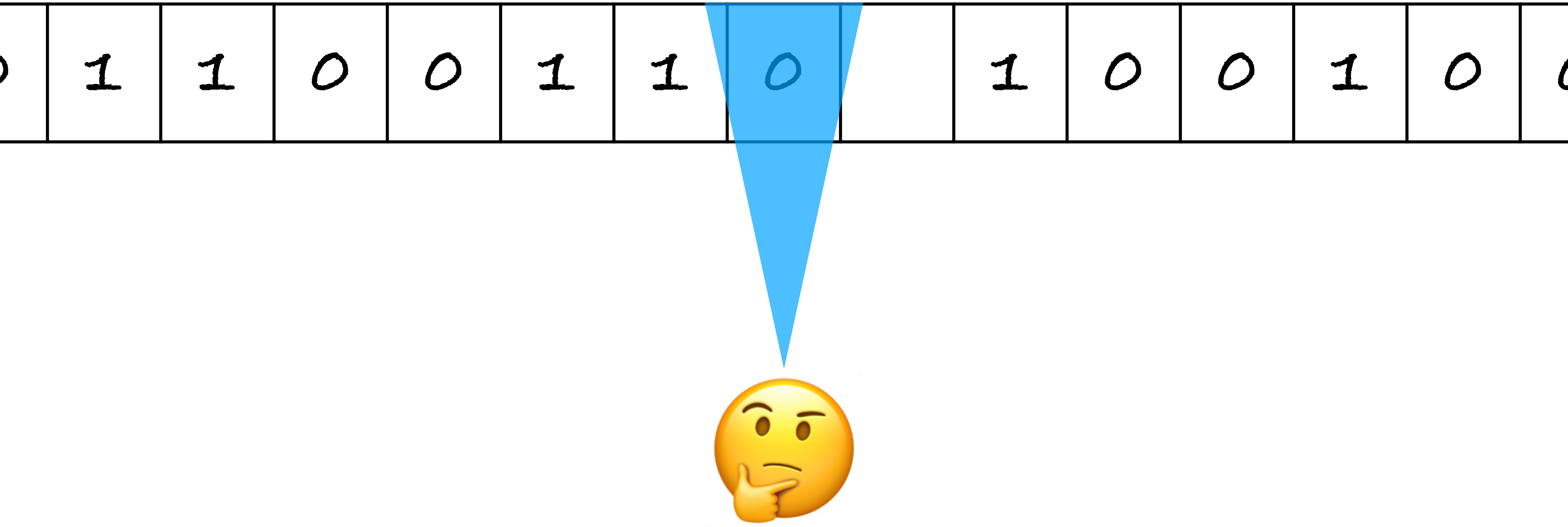
« Champ visuel »



« Champ visuel »



« Champ visuel »



« Le comportement du calculateur à chaque moment est déterminé par le symbole qu’il observe et son “état d’esprit” à ce moment. »

– Alan M. Turing, *On computable numbers*

« États d'esprit »

$$\begin{array}{r} 12932\ + \\ 19\ = \\ \hline \end{array}$$

« États d’esprit »

12932 +

19 =

J’ai lu le chiffre 2 et le chiffre 9

« États d'esprit »

$$
\begin{array}{r}
12932\ + \\
19\ = \\
\hline
\end{array}
$$

Il faut que j'écrive 1 et que je garde 1 comme retenue

« États d’esprit »

$$\begin{array}{r} 12932\ + \\ 19\ = \\ \hline 1 \end{array}$$

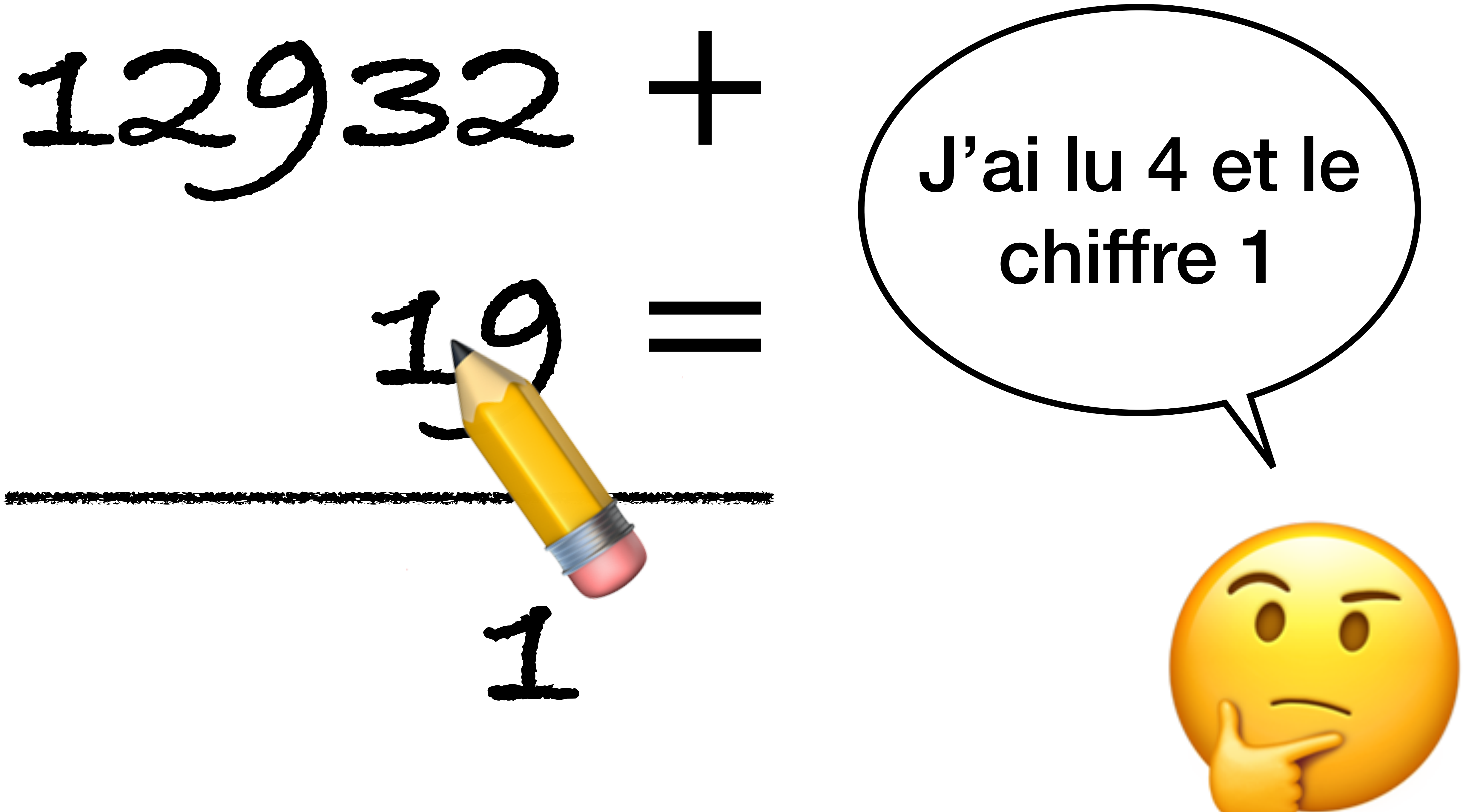
Il faut que je me déplace à gauche ; la retenue est 1

« États d'esprit »

$$\begin{array}{r} 12932\ + \\ 19\ = \\ \hline 1 \end{array}$$

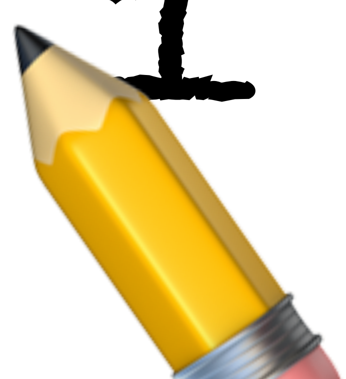
J'ai lu le chiffre 3 ; avec la retenue de 1 ça fait 4

« États d’esprit »



« États d’esprit »

$$\begin{array}{r} 12932\ + \\ 19\ = \\ \hline 1 \end{array}$$



Il faut que j’écrive 5 ; pas de retenue

« États d’esprit »

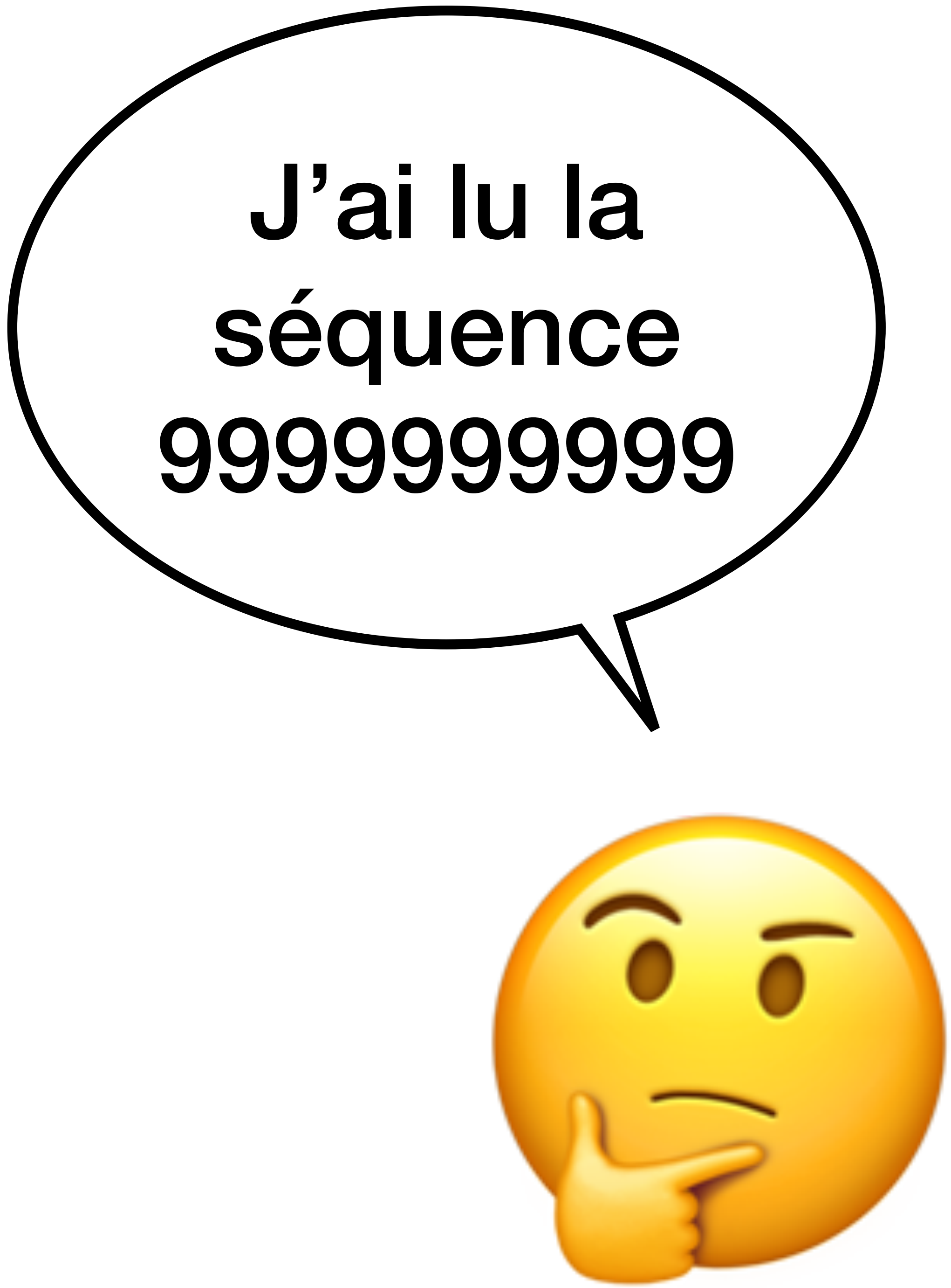
$$\begin{array}{r} 12932\ + \\ 19\ = \\ \hline 51 \end{array}$$

Il faut que je me déplace à gauche

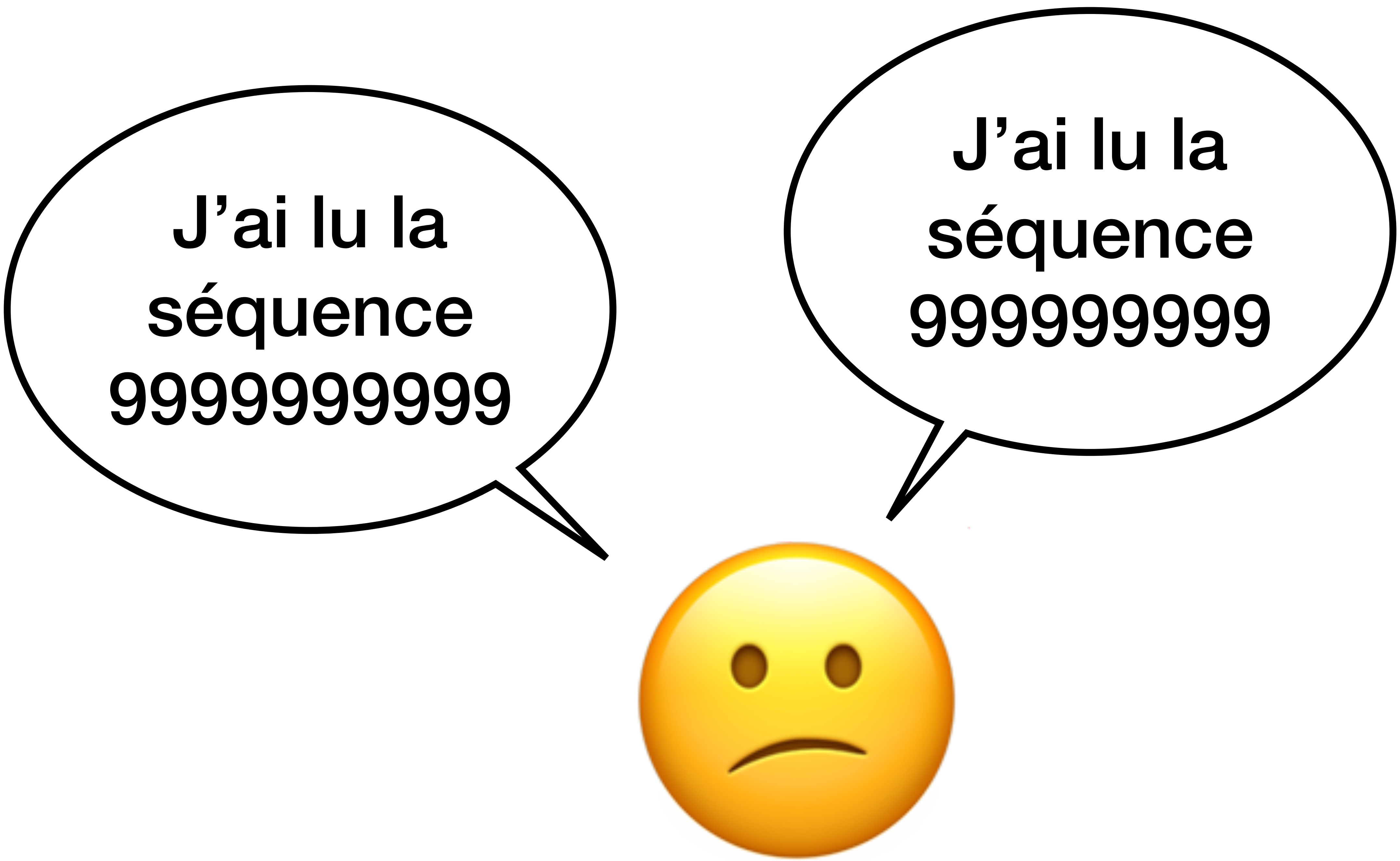
« On suppose également que le nombre d’états d’esprit qu’on doit prendre en compte soit fini. Les raisons pour cela sont de la même nature que celles qui restreignent le nombre de symboles. »

– Alan M. Turing, *On computable numbers*

États d’esprit trop proches



États d’esprit trop proches

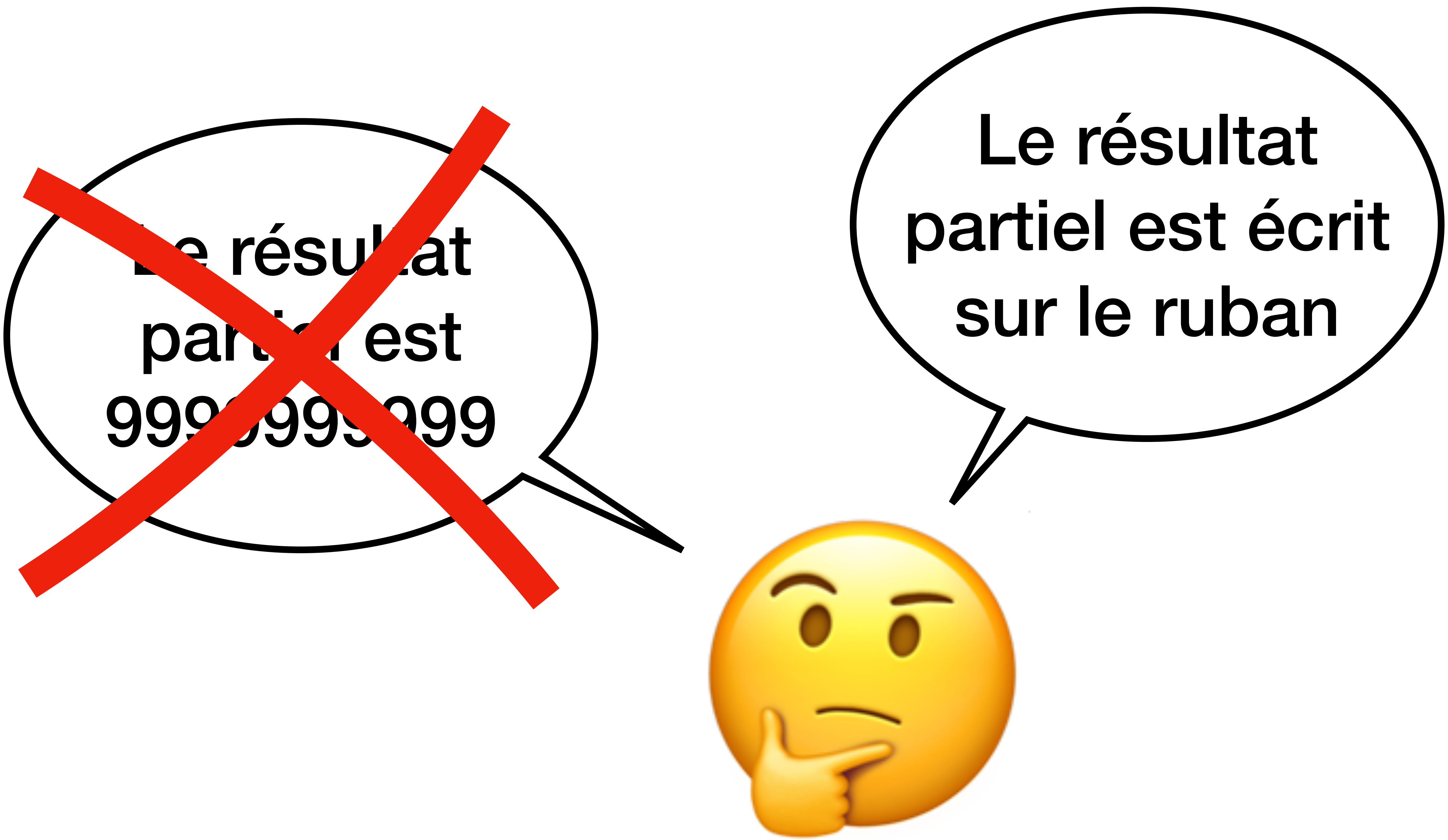


« On peut éviter l’utilisation d’états d’esprit plus compliqués en écrivant plus de symboles sur le ruban. »

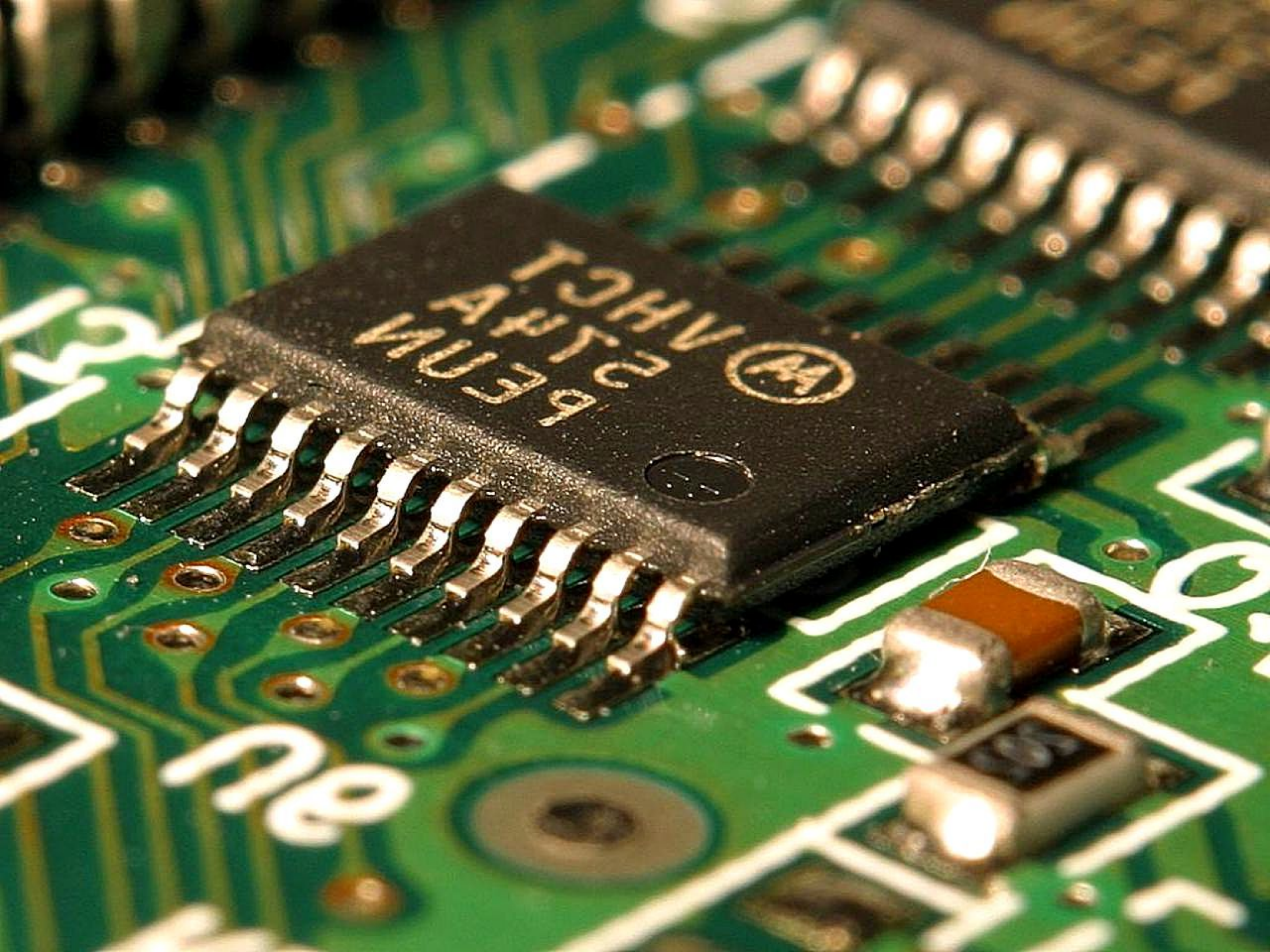
– Alan M. Turing, *On computable numbers*

Prendre note sur le ruban

Prendre note sur le ruban



Calculateurs électroniques



Équations de Maxwell

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \left(\mathbf{J} + \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right)$$

Calculateurs mécaniques

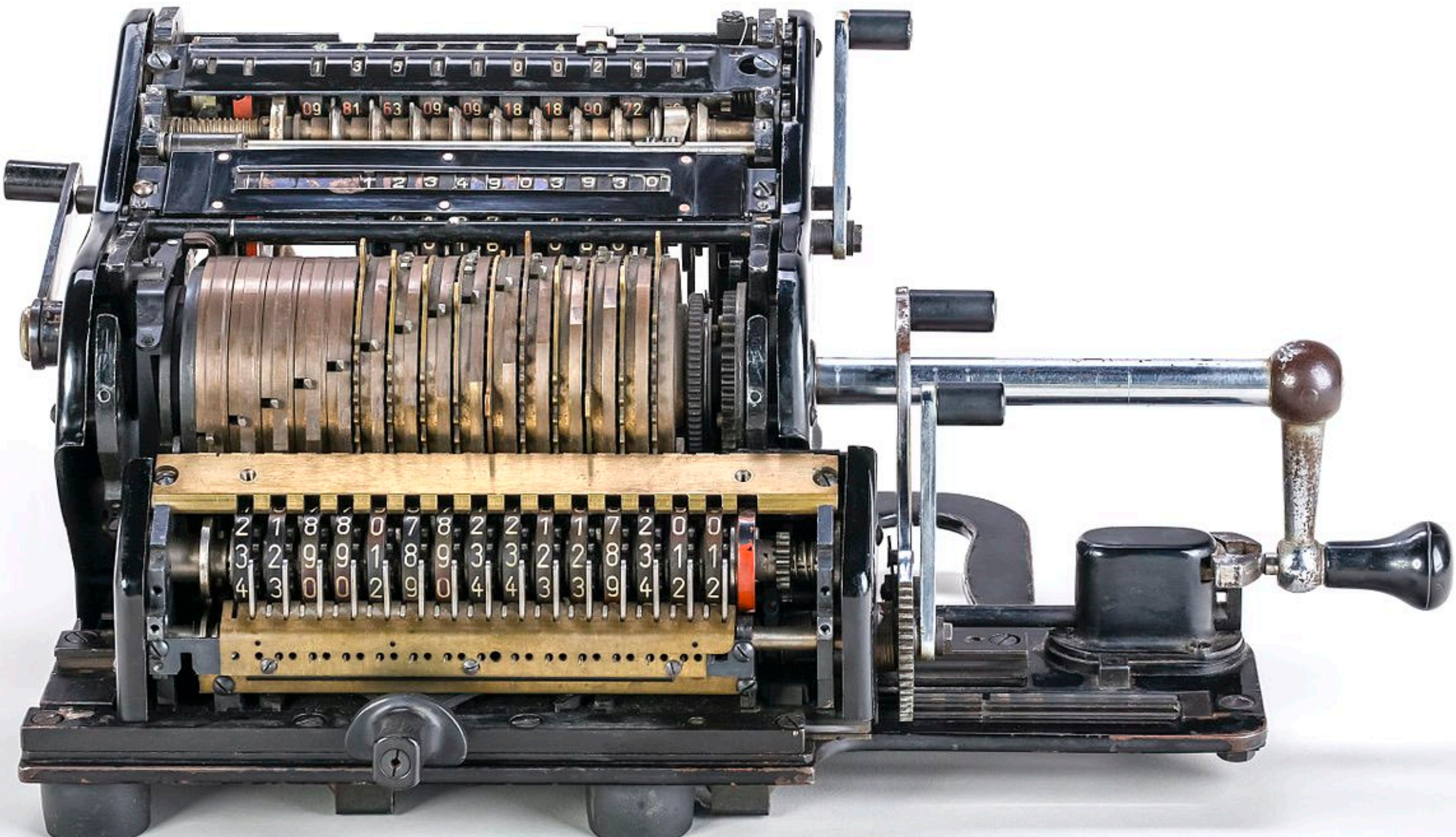
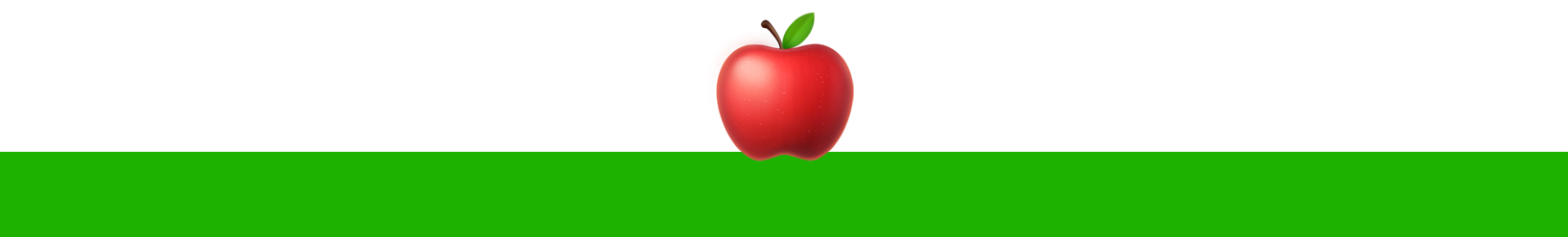


Photo by CEphoto, Uwe Aranas https://commons.wikimedia.org/wiki/File:Mechanical-calculator-Brunsviga-15-01.jpg

Calculateurs gravitationnels

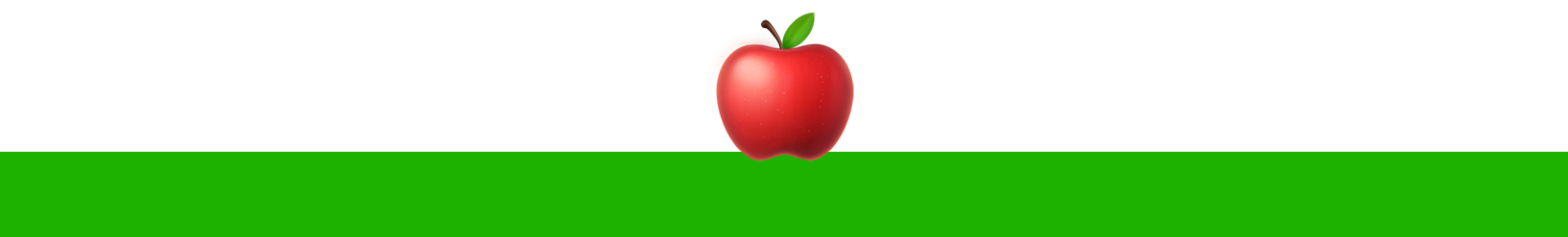
Calculer avec la gravité

Calculer avec la gravité

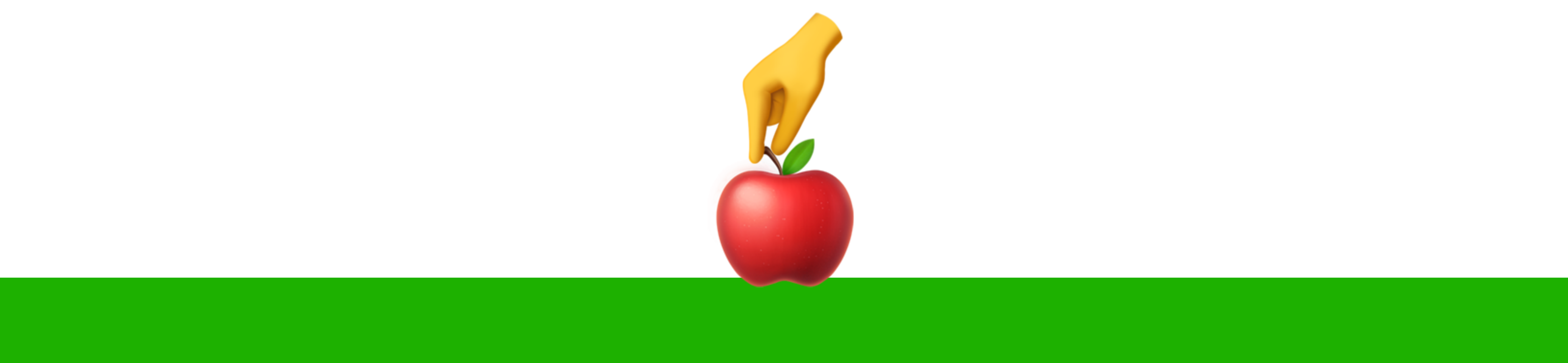


Calculer avec la gravité

$$⏱ = \sqrt{\frac{2h}{g}}$$



Calculer avec la gravité



Calculer avec la gravité

$$h = \frac{xg}{2}$$

Calculer avec la gravité

$$⏱ = \sqrt{\frac{2h}{g}}$$

$$h = \frac{xg}{2}$$

🍎

Calculer avec la gravité

$$⏱ = \sqrt{\frac{2h}{g}}$$

$$= \sqrt{\frac{2(xg/2)}{g}}$$

$$h = \frac{xg}{2}$$

🍎

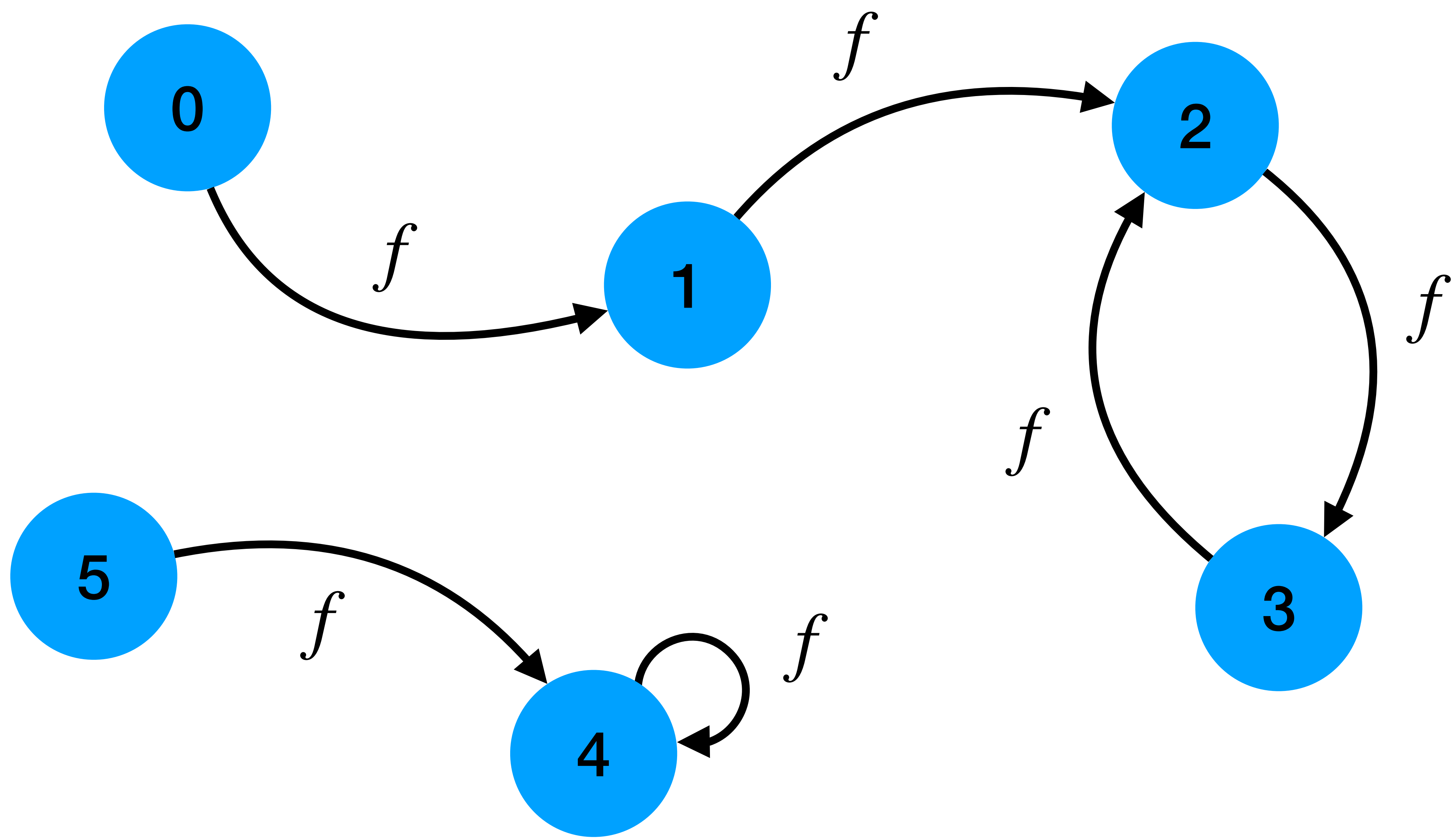
Calculer avec la gravité

$$h = \frac{xg}{2} \left\{ \begin{aligned} \text{⏱} &= \sqrt{\frac{2h}{g}} \\ &= \sqrt{\frac{2(xg/2)}{g}} \\ &= \color{red}{\sqrt{x}} \end{aligned} \right.$$

Dynamical systems and their algebra

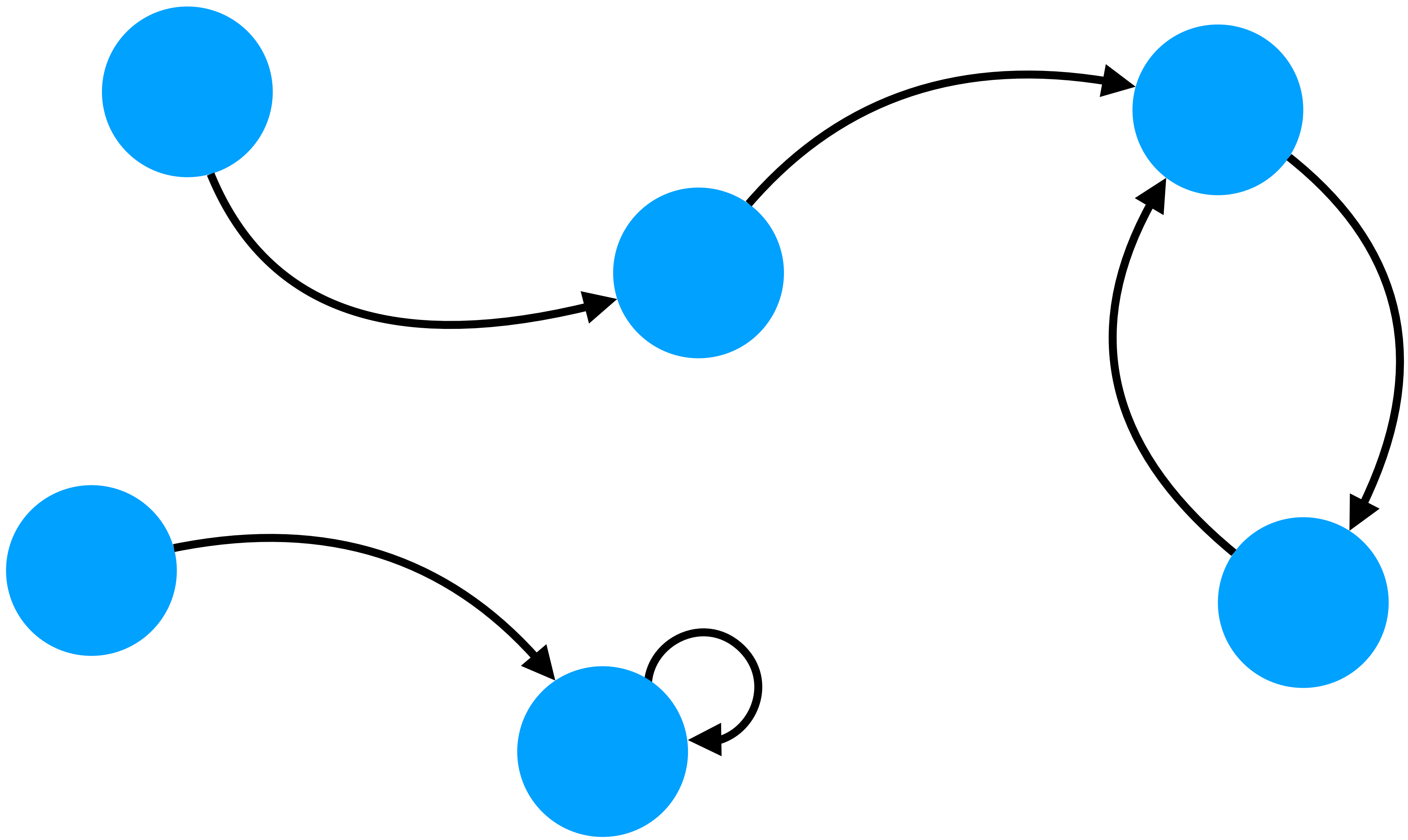
Finite, discrete-time dynamical systems

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



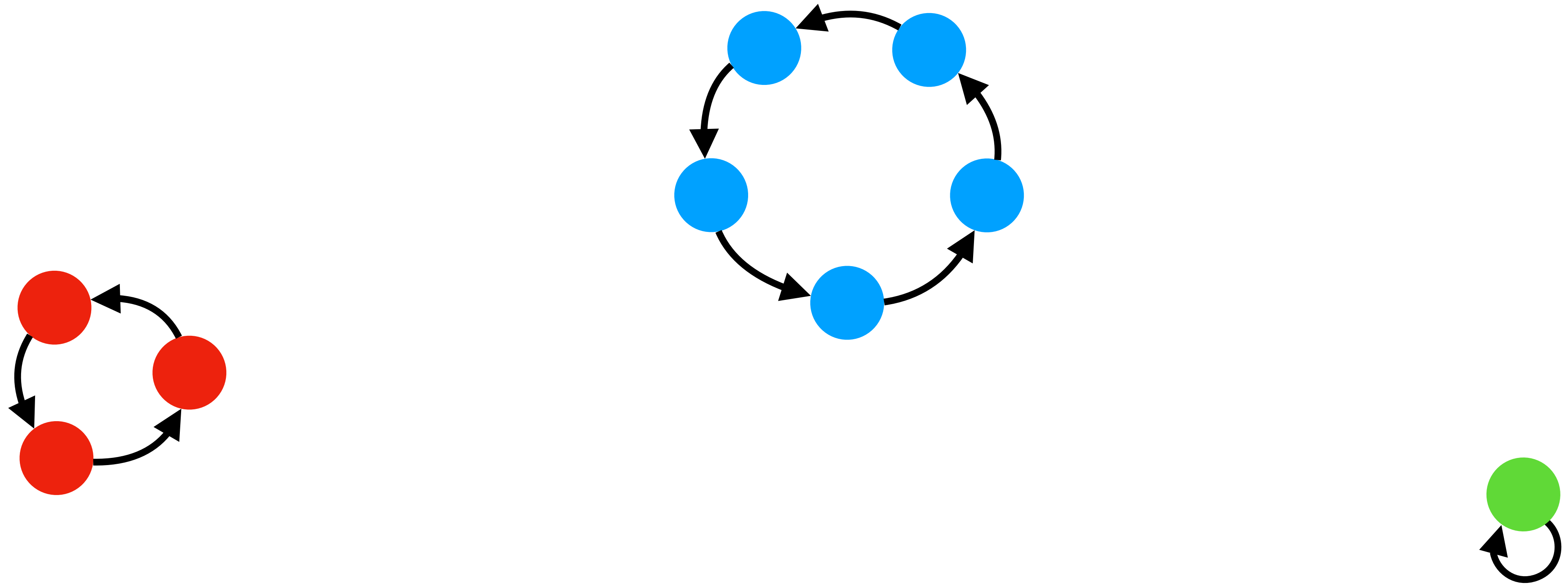
Finite, discrete-time dynamical systems

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



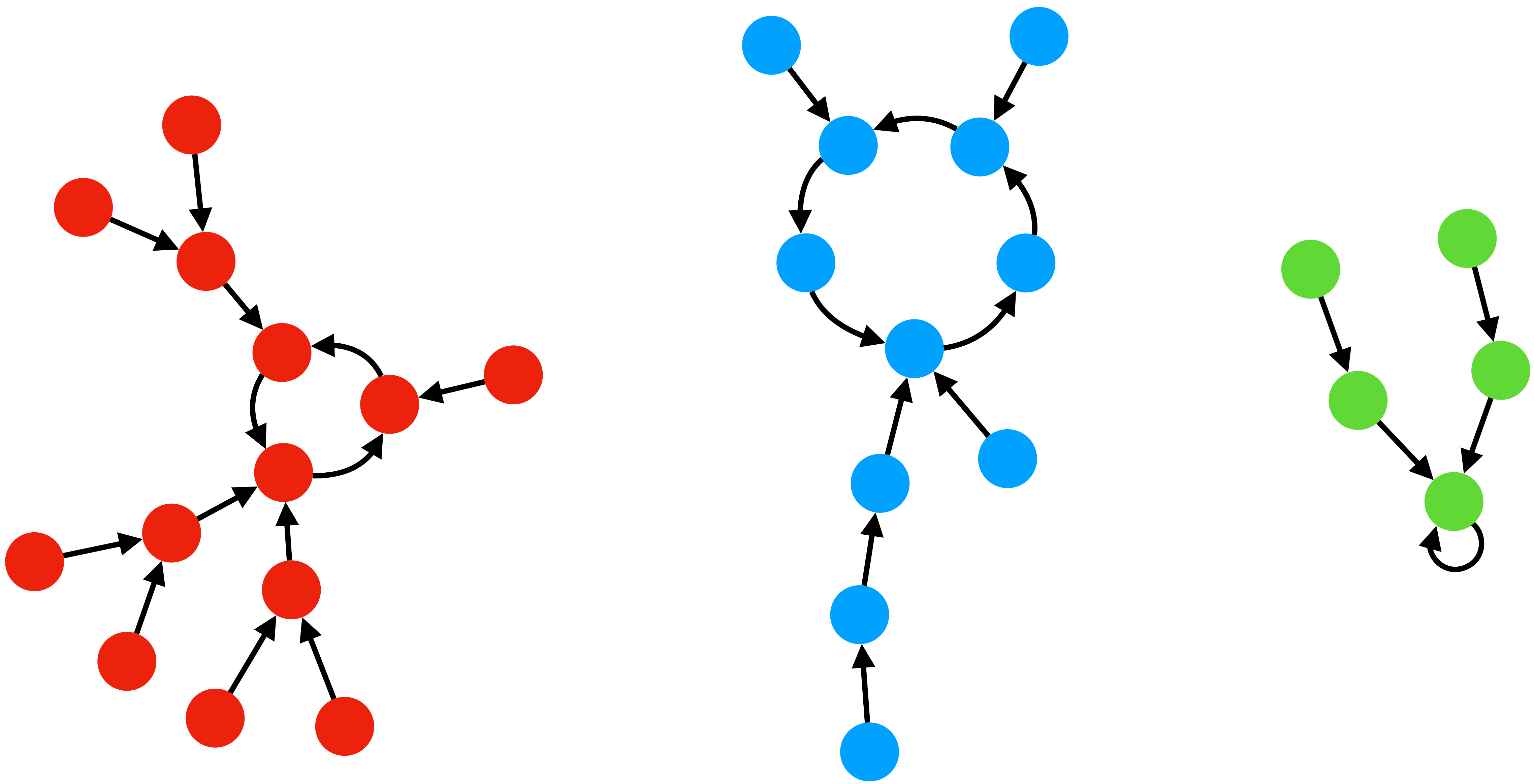
General shape of a dynamical system

A few limit cycles



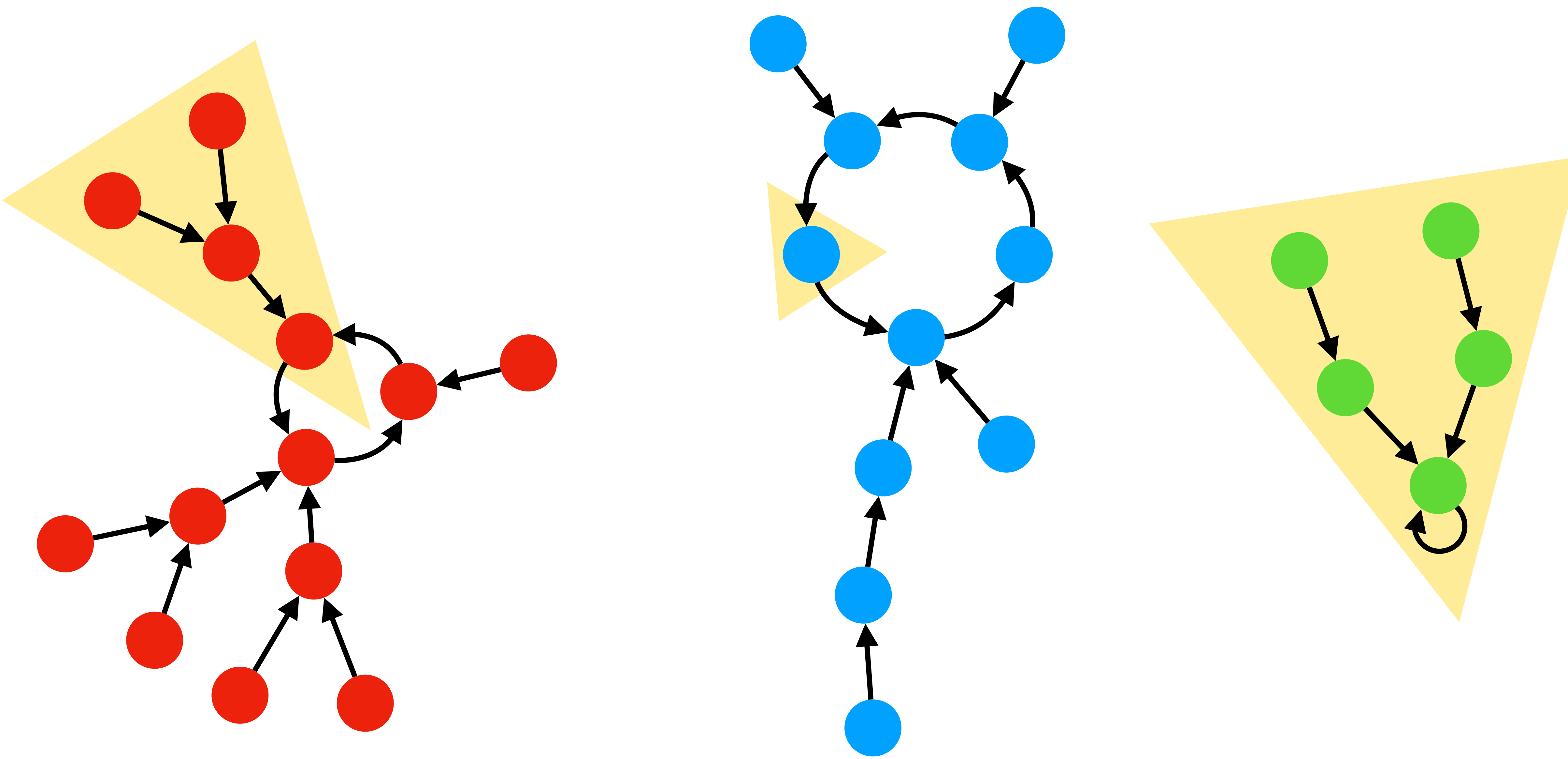
General shape of a dynamical system

A few limit cycles with trees going in



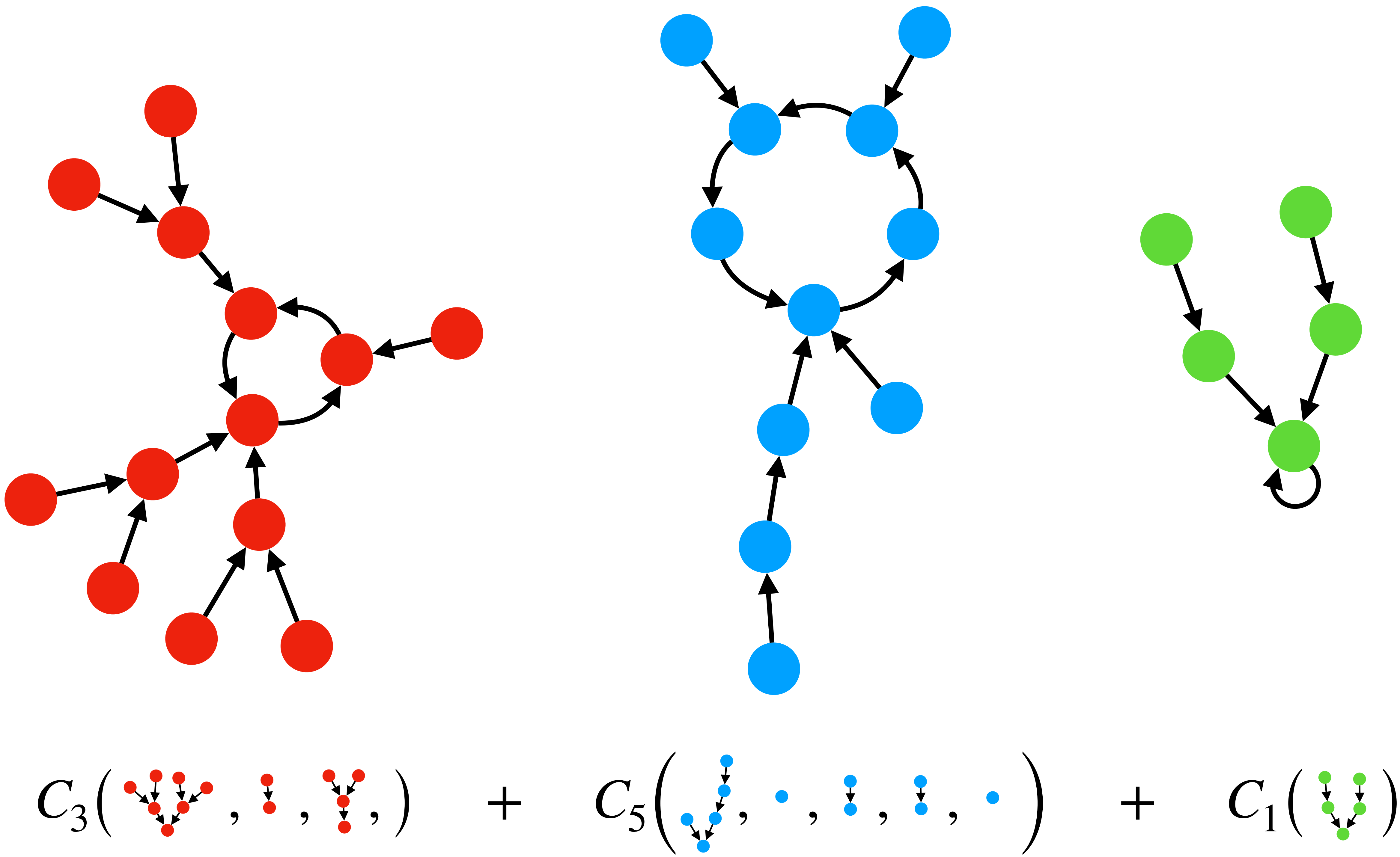
General shape of a dynamical system

A few limit cycles with trees going in



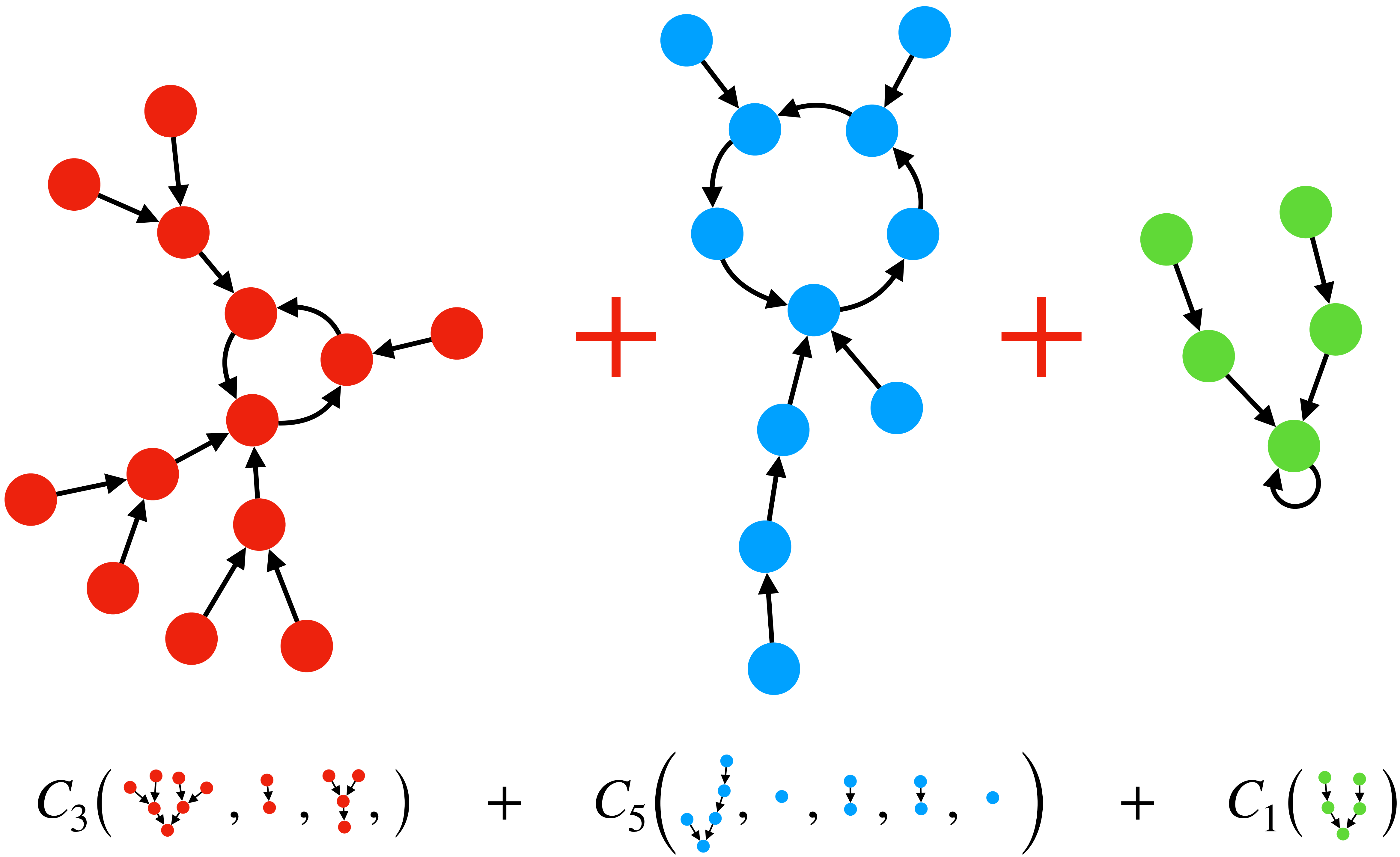
General shape of a dynamical system

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

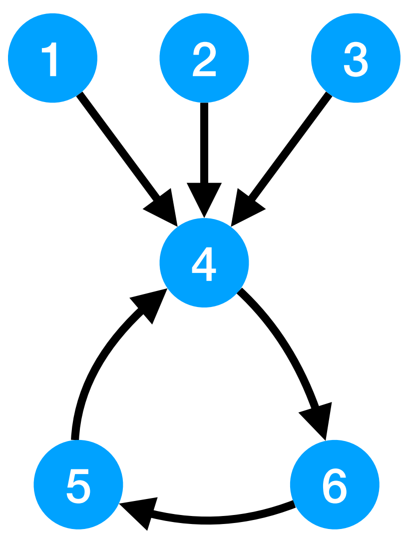


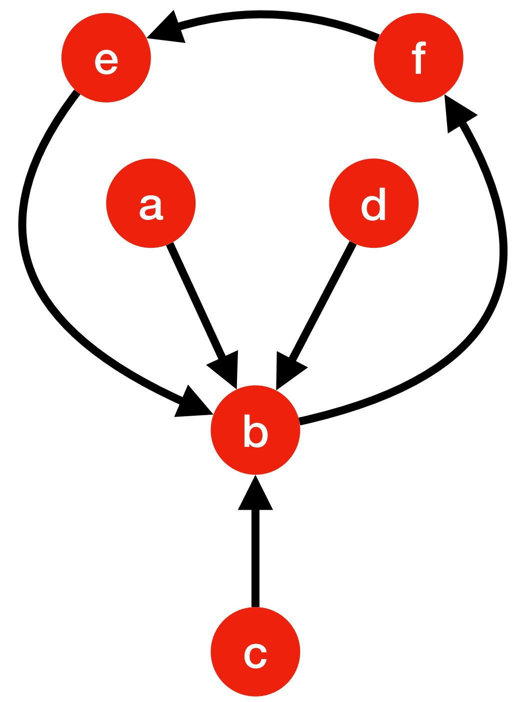
General shape of a dynamical system

A few limit cycles with trees going in

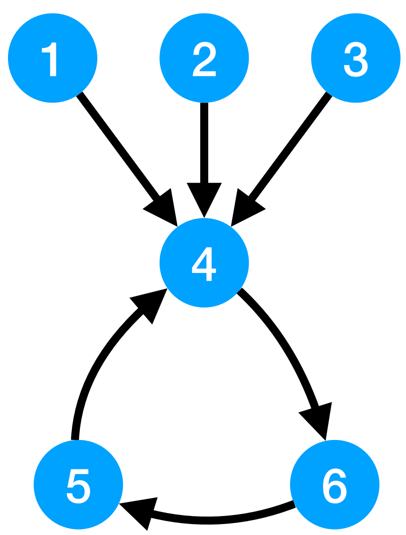


Discrete (finite, deterministic) dynamical systems up to isomorphisms

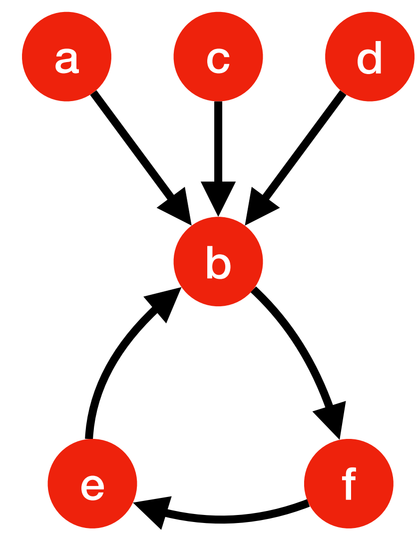




Discrete (finite, deterministic) dynamical systems up to isomorphisms

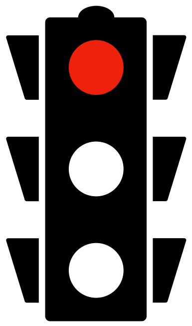


=

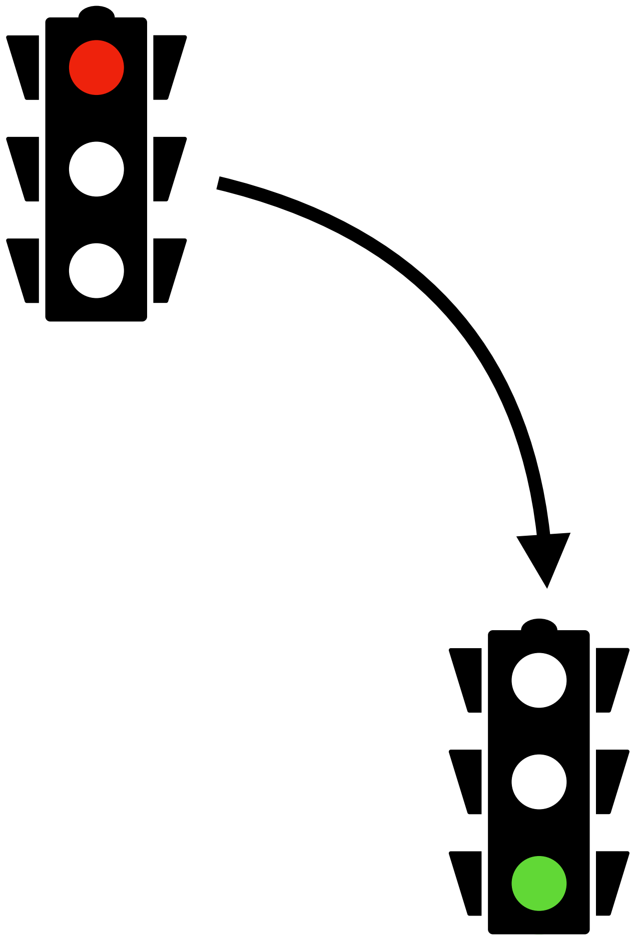


An example from engineering

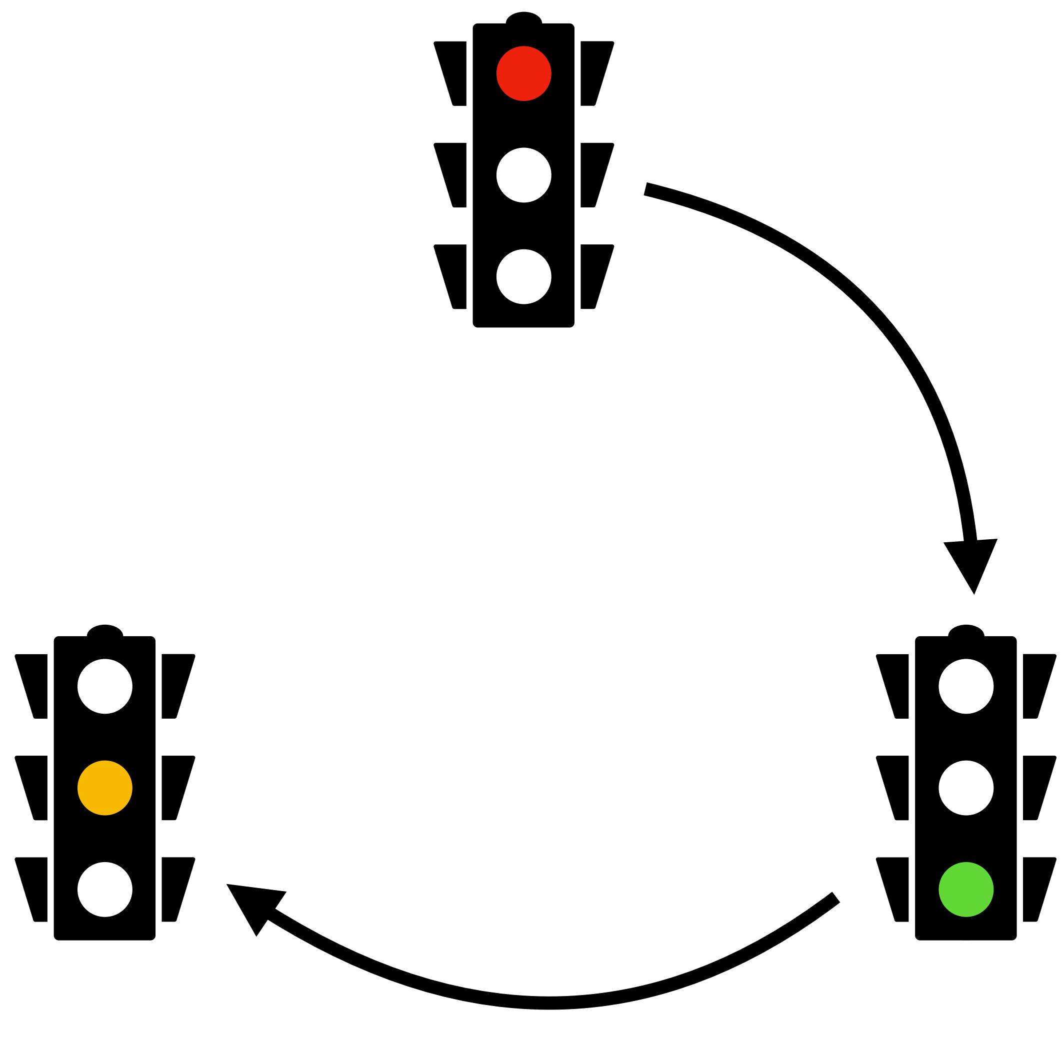
Traffic lights



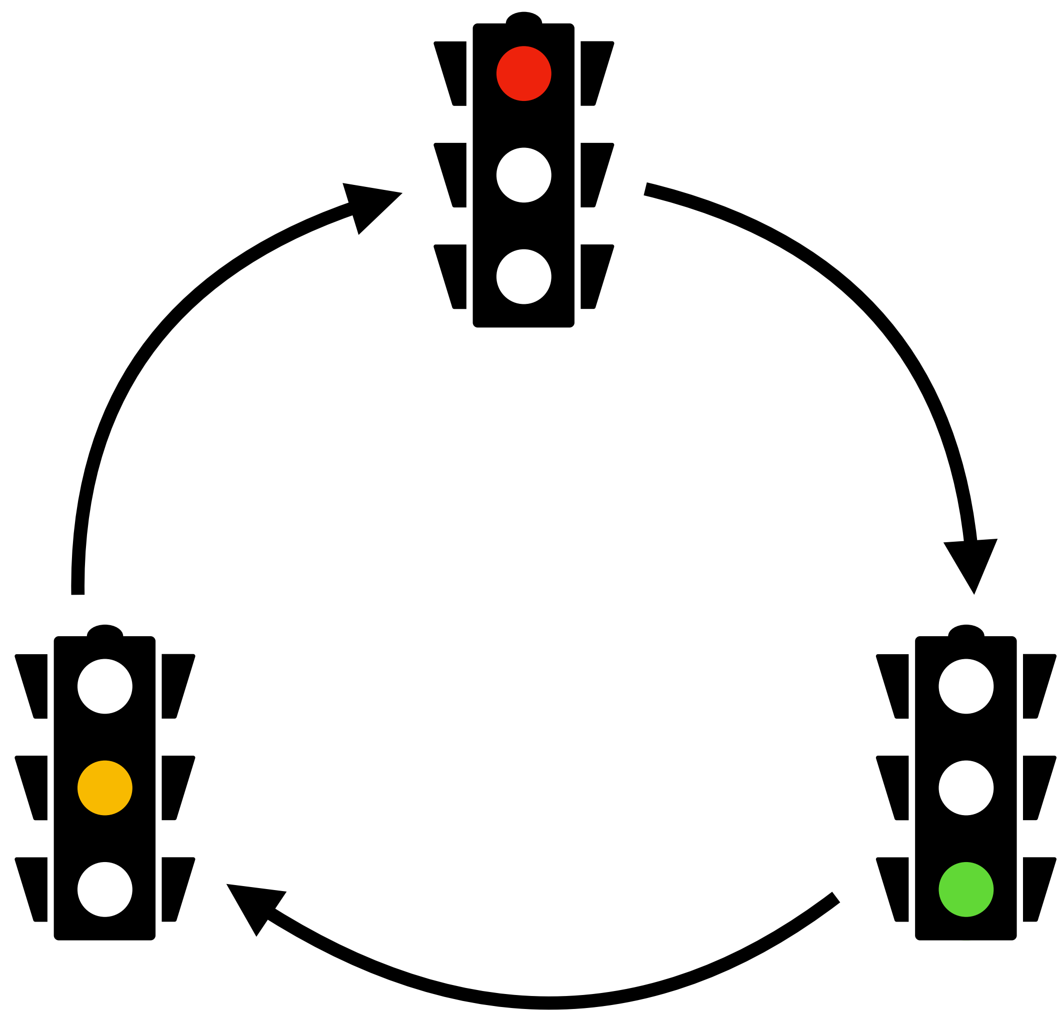
Traffic lights



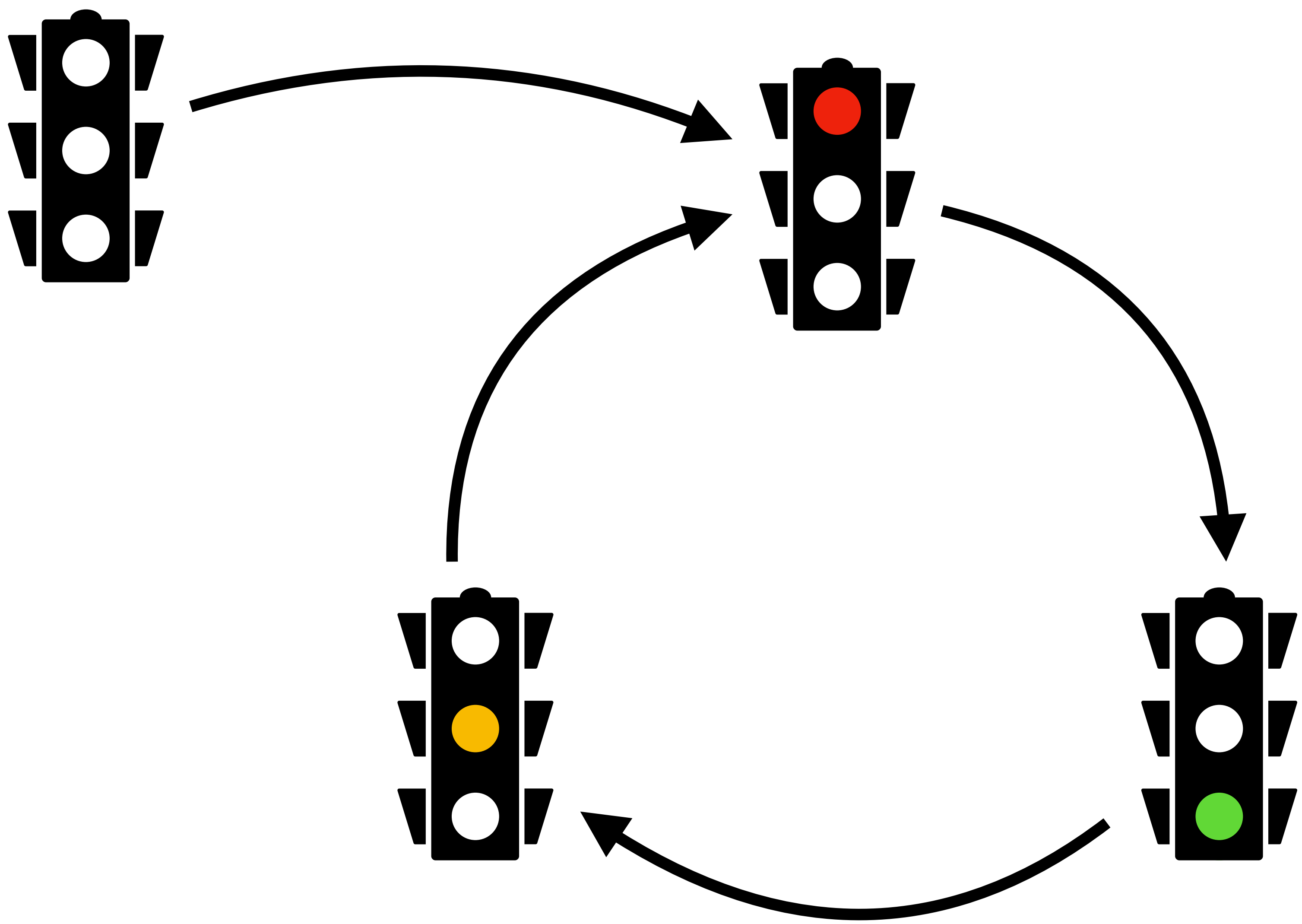
Traffic lights



Traffic lights

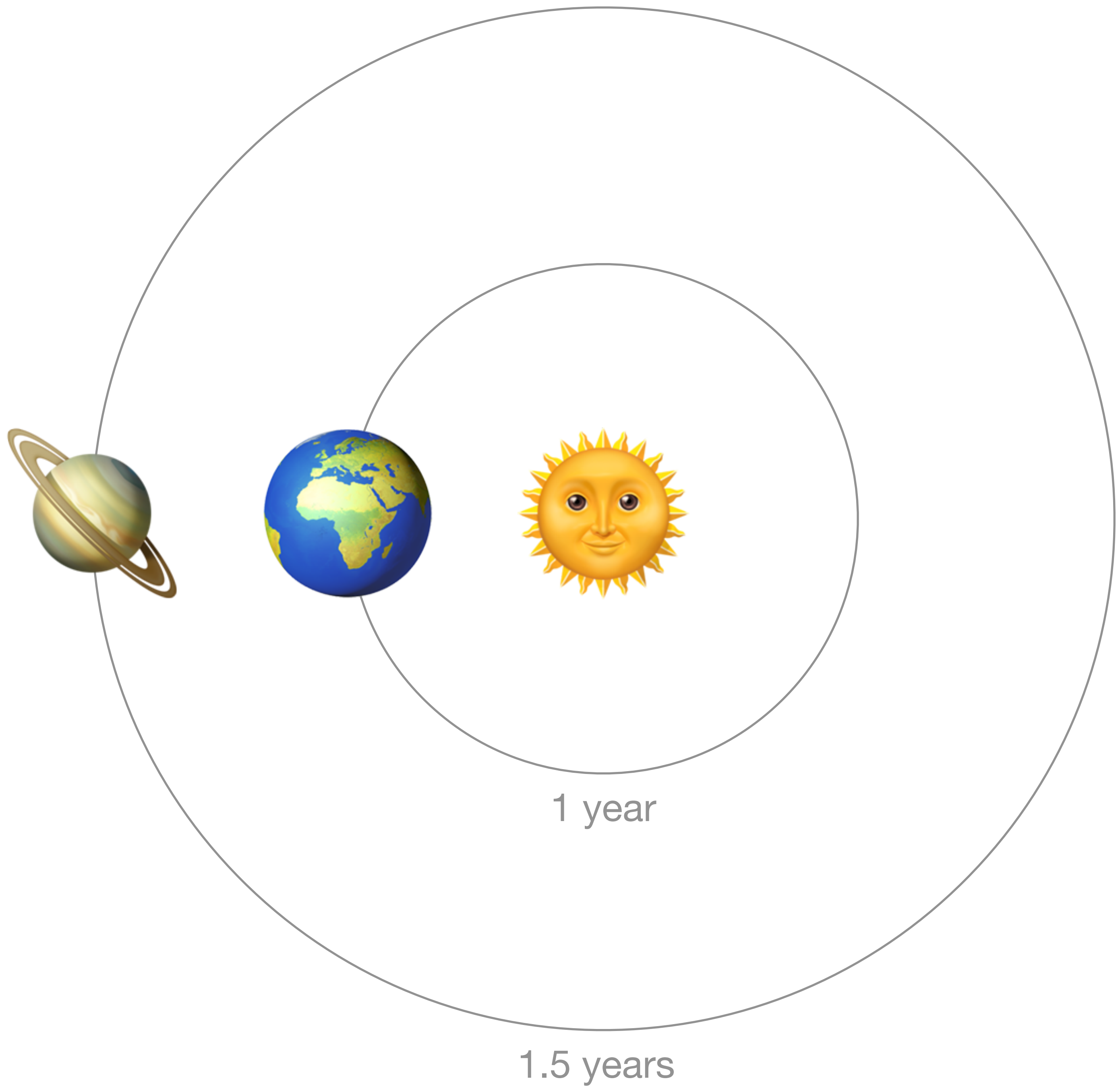


Traffic lights

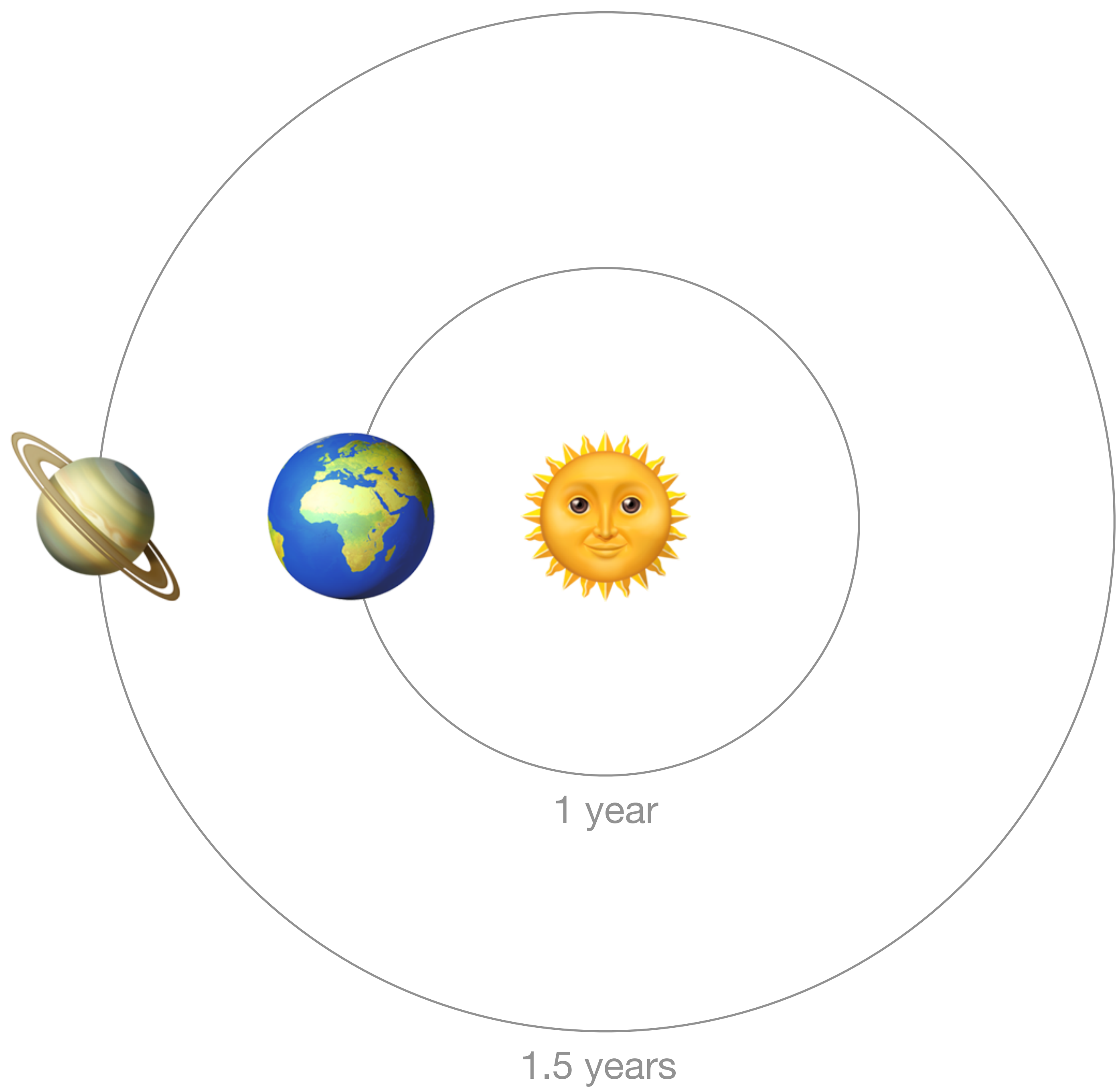


An example from science

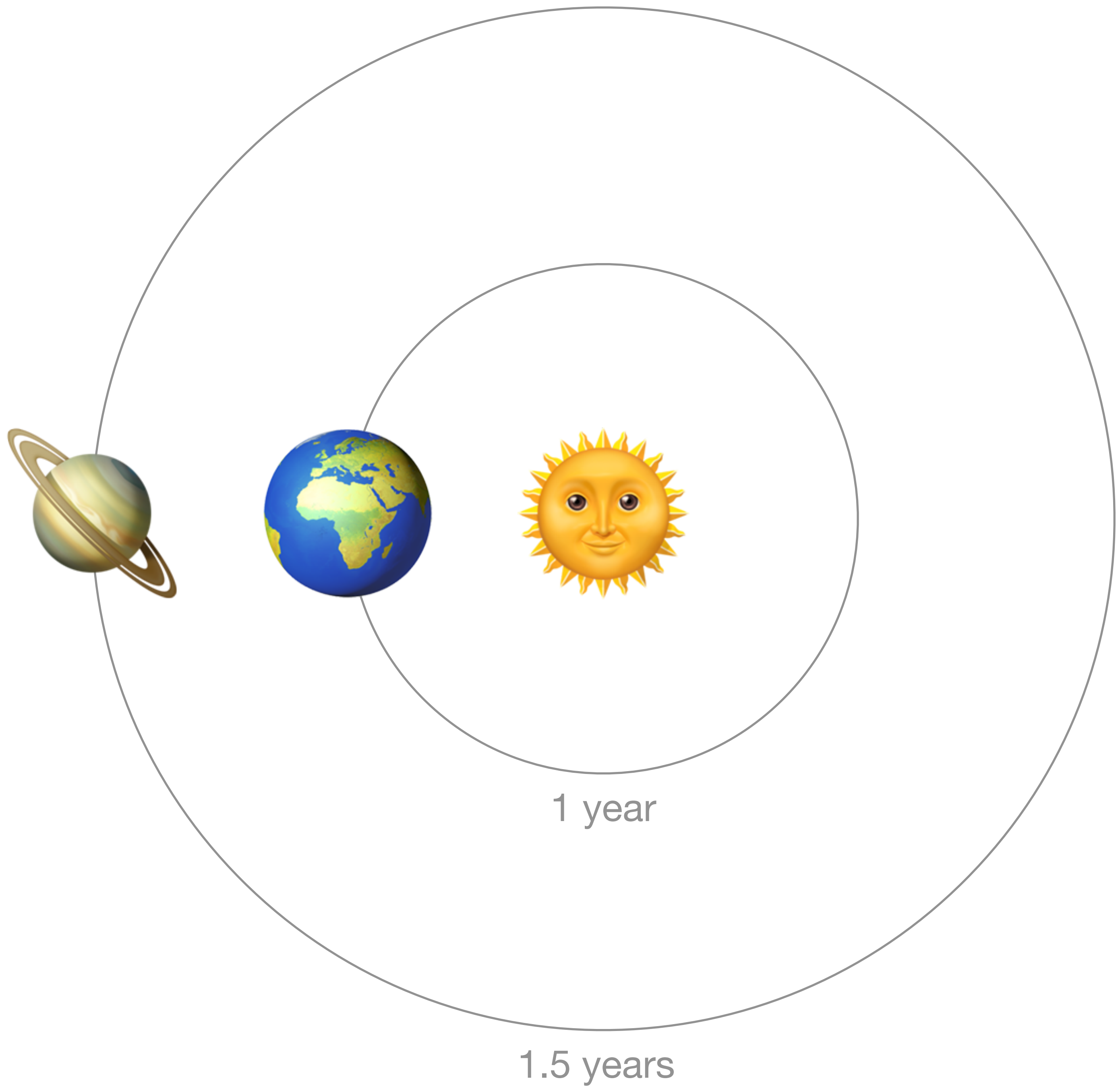
A planetary system



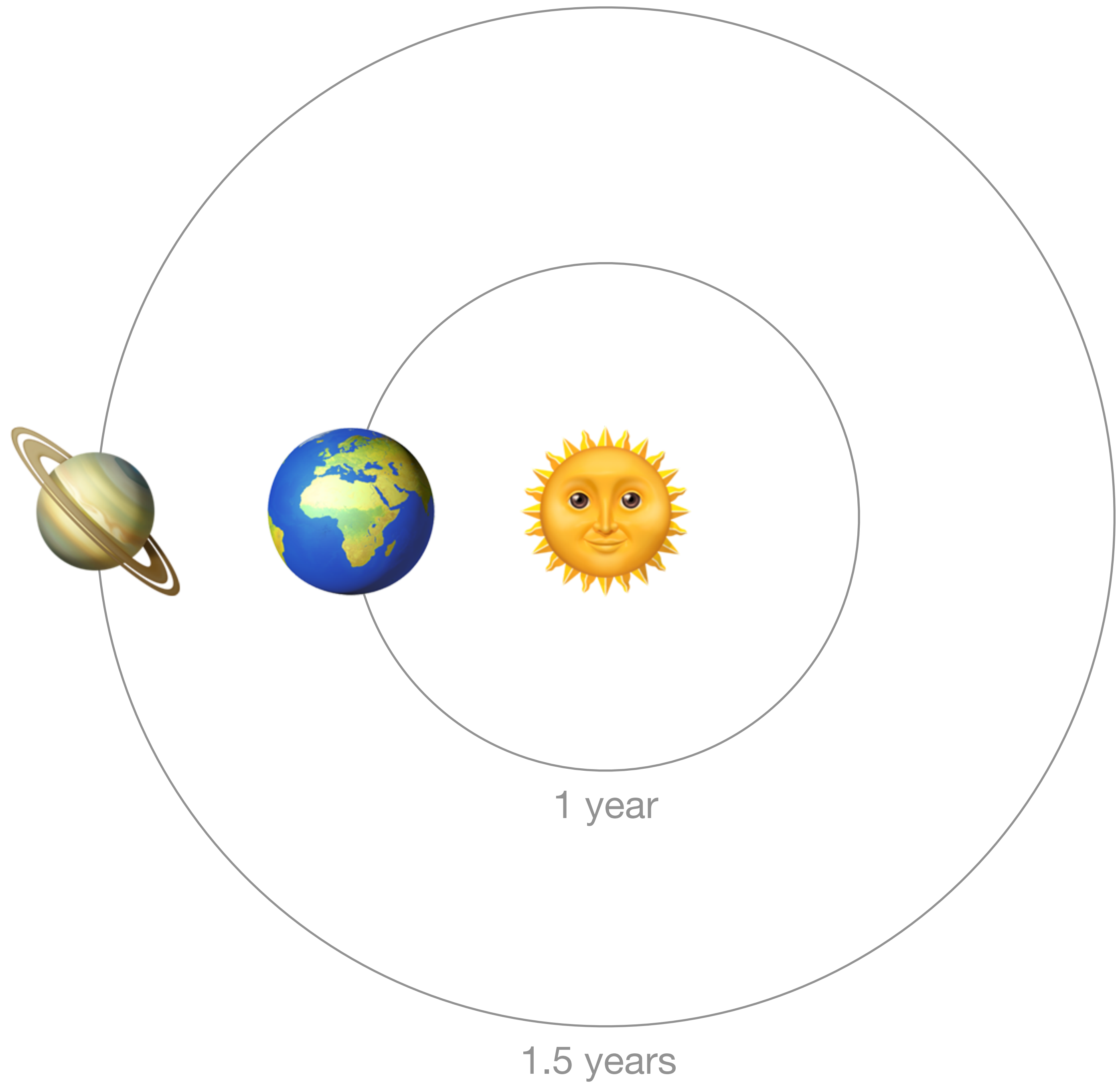
A planetary system



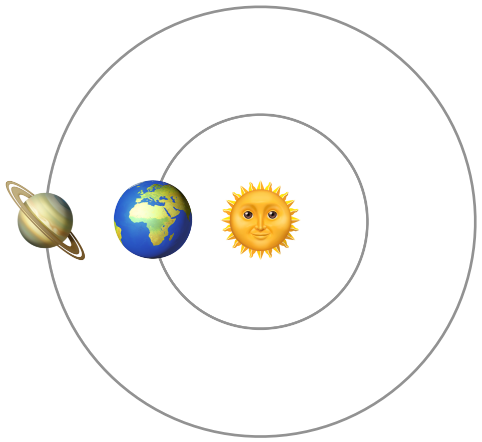
A planetary system



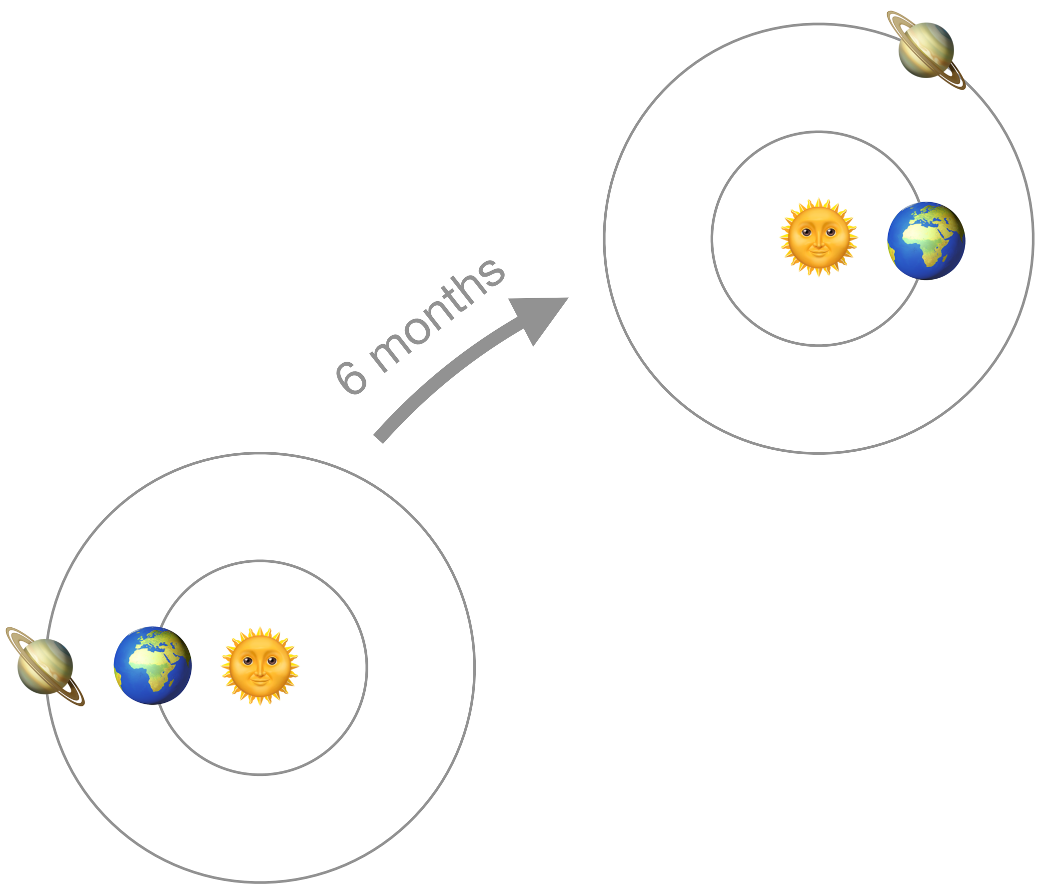
A planetary system



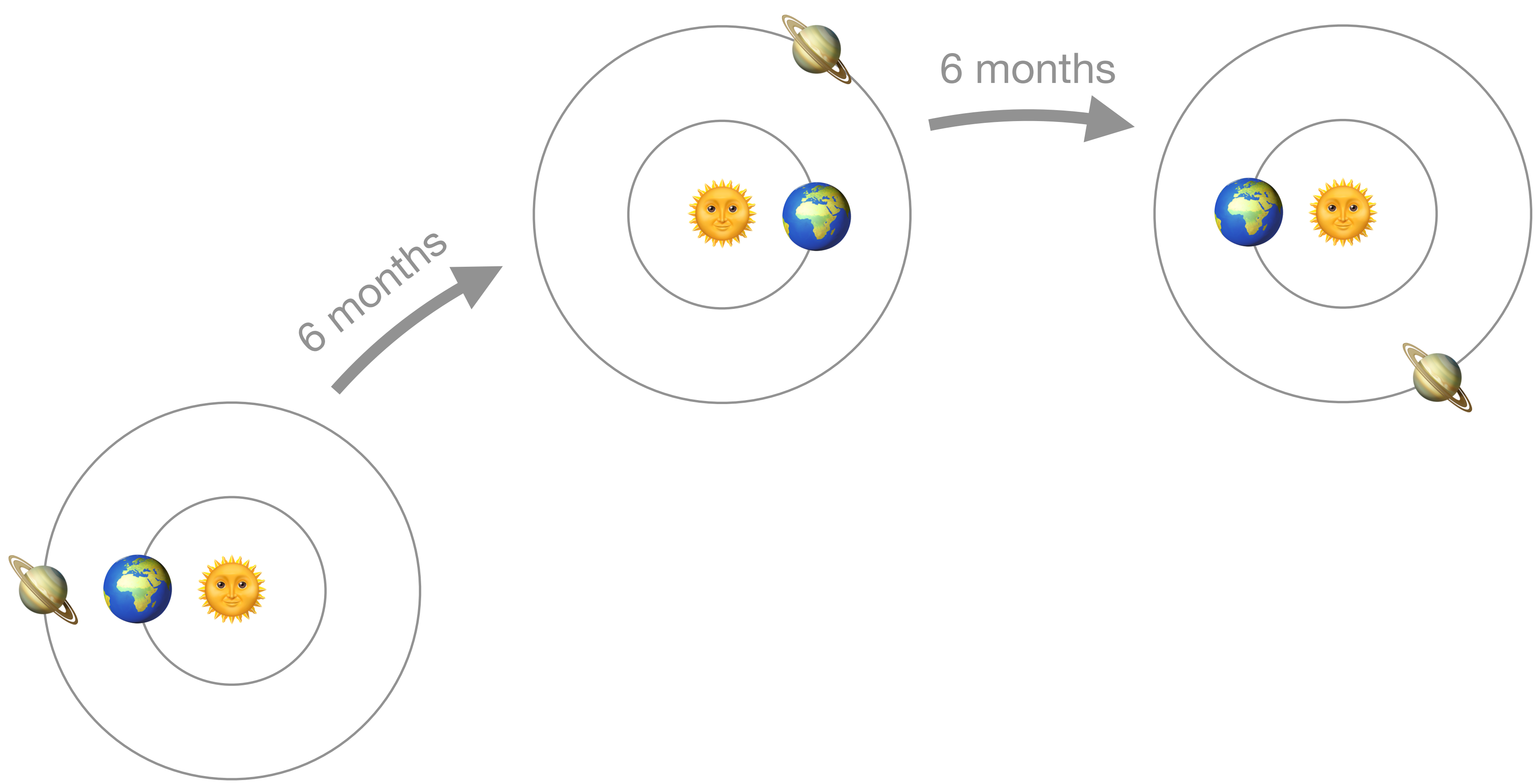
Evolution in time



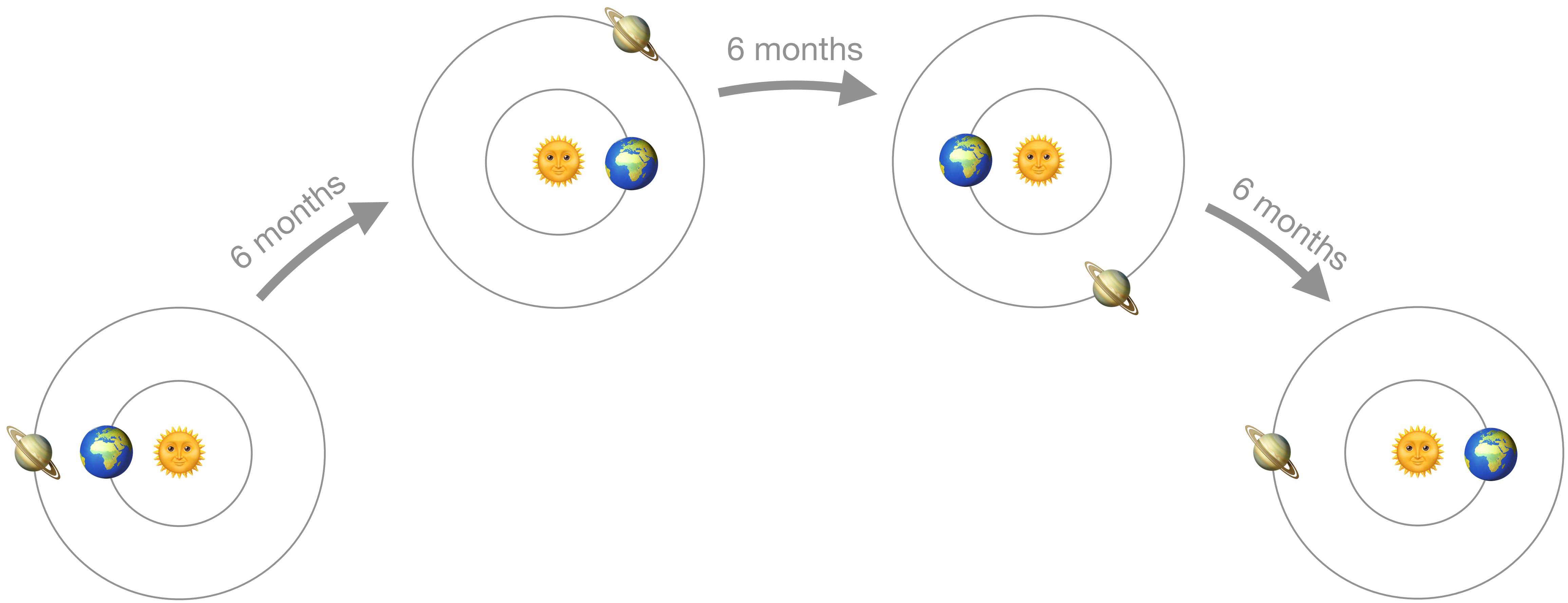
Evolution in time



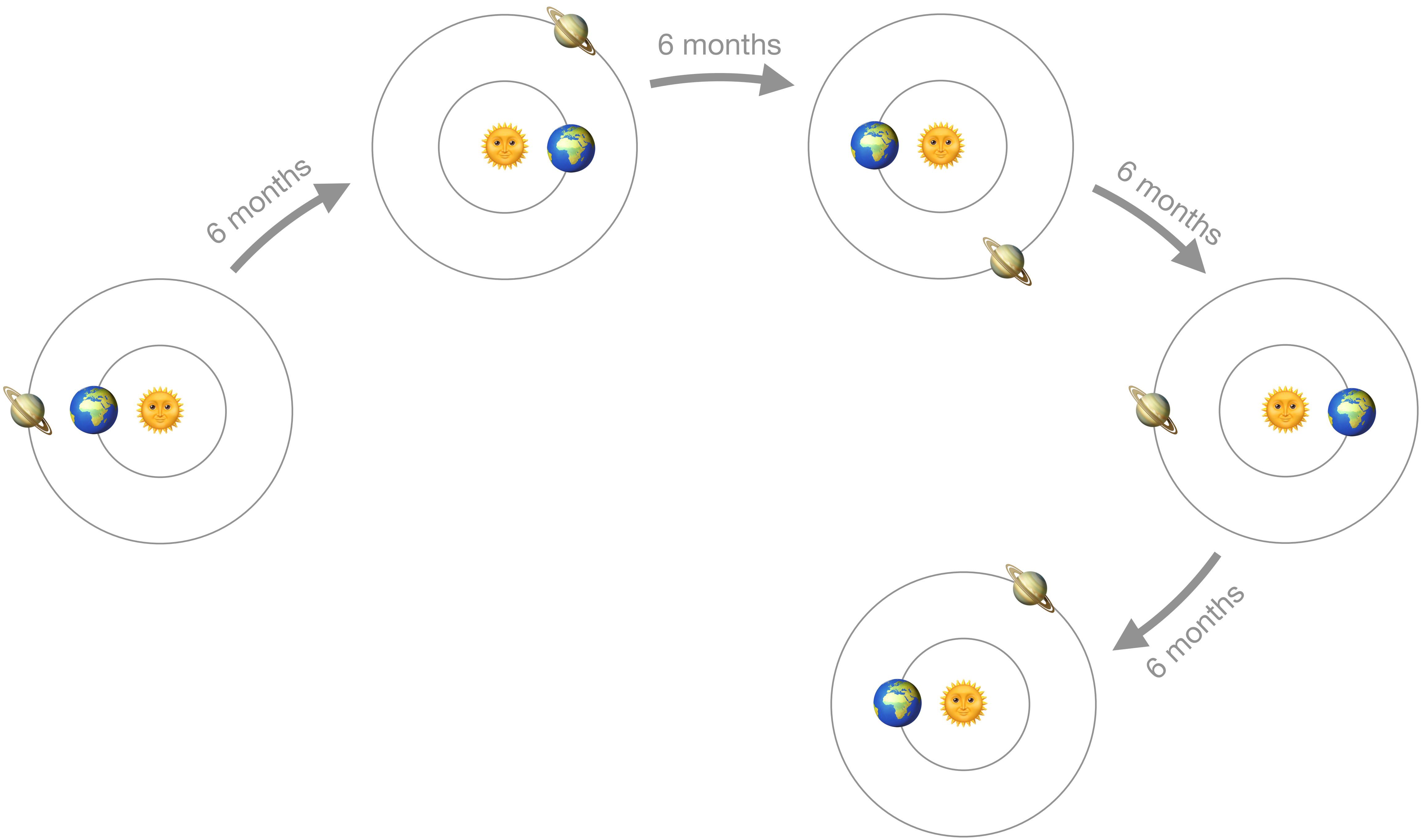
Evolution in time



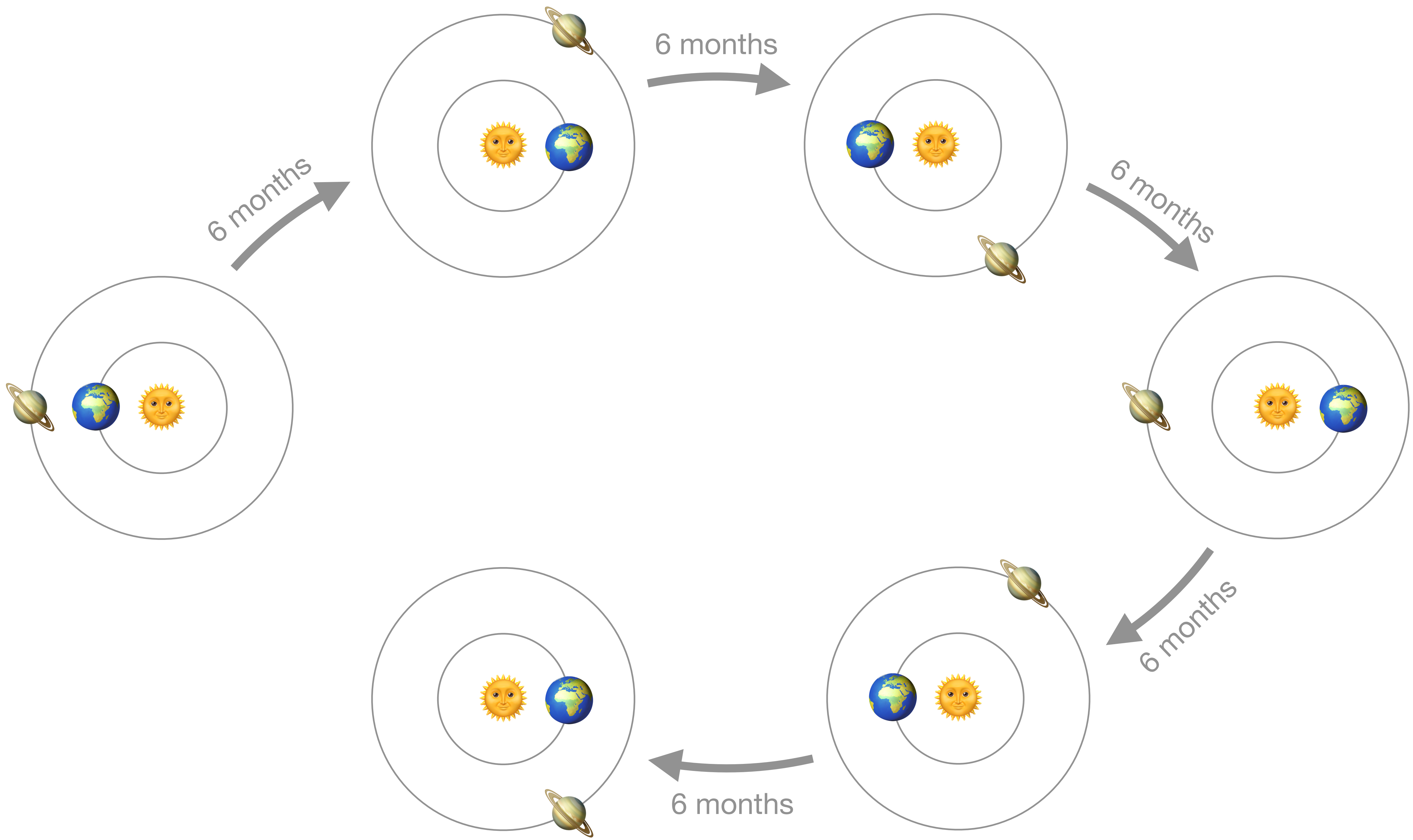
Evolution in time



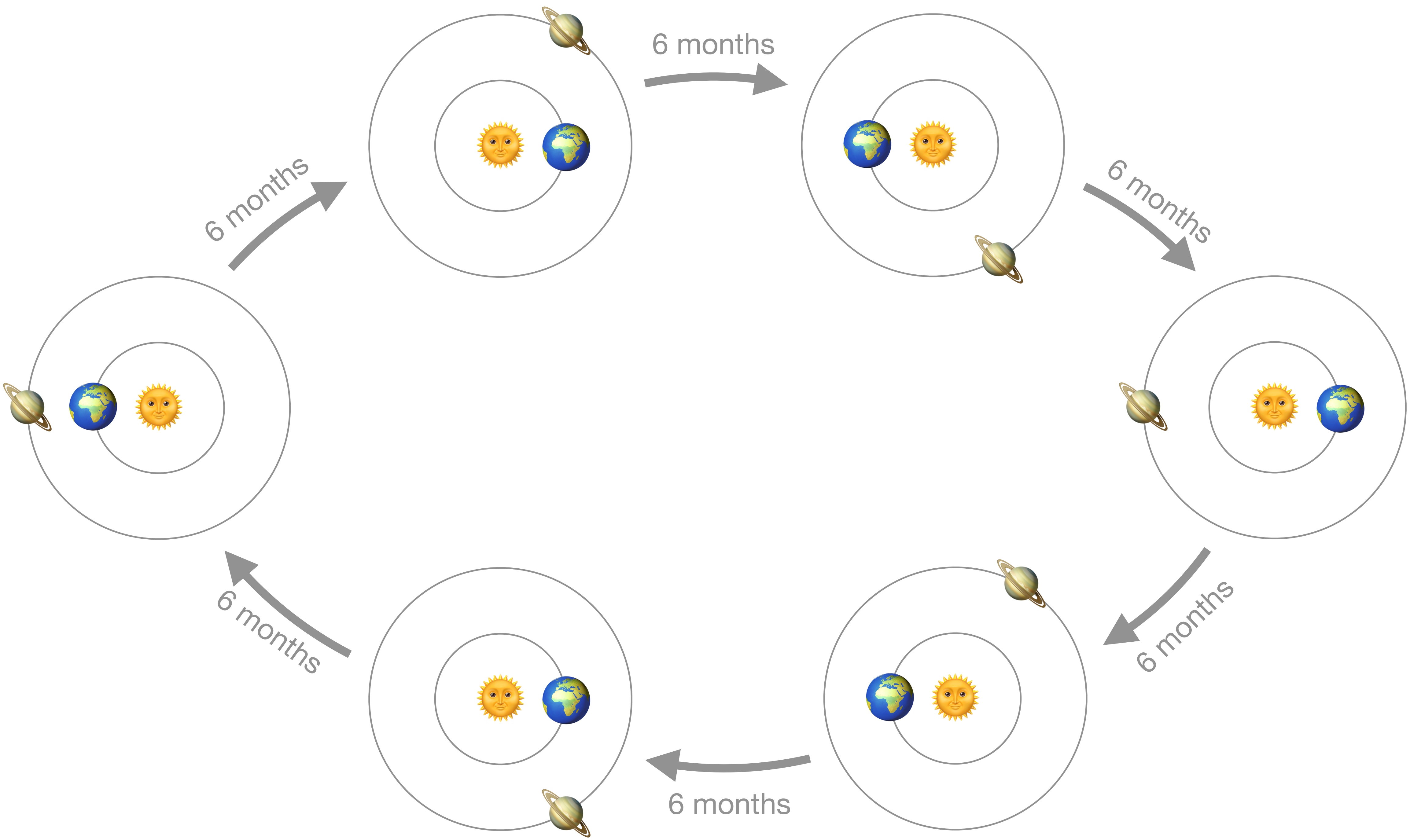
Evolution in time



Evolution in time

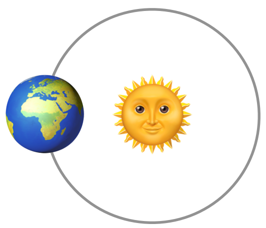


Evolution in time

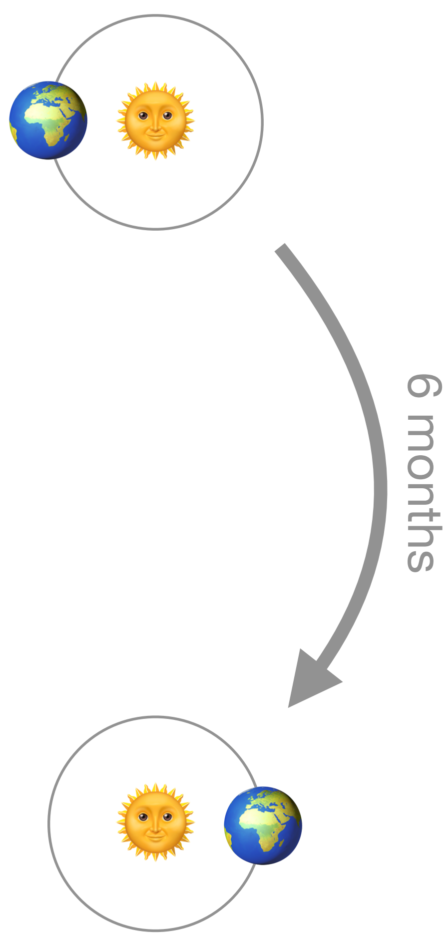


Decomposing the system

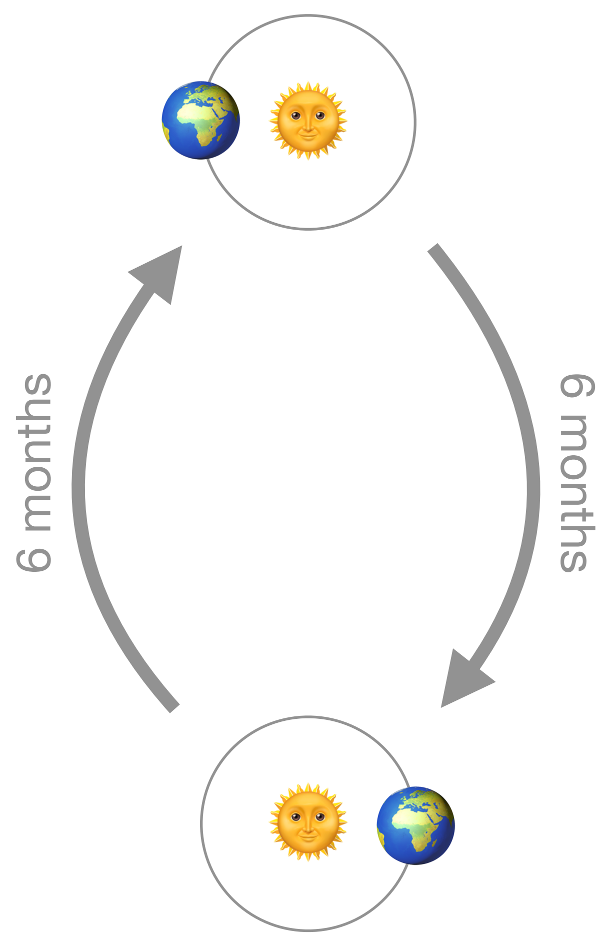
Decomposing the system



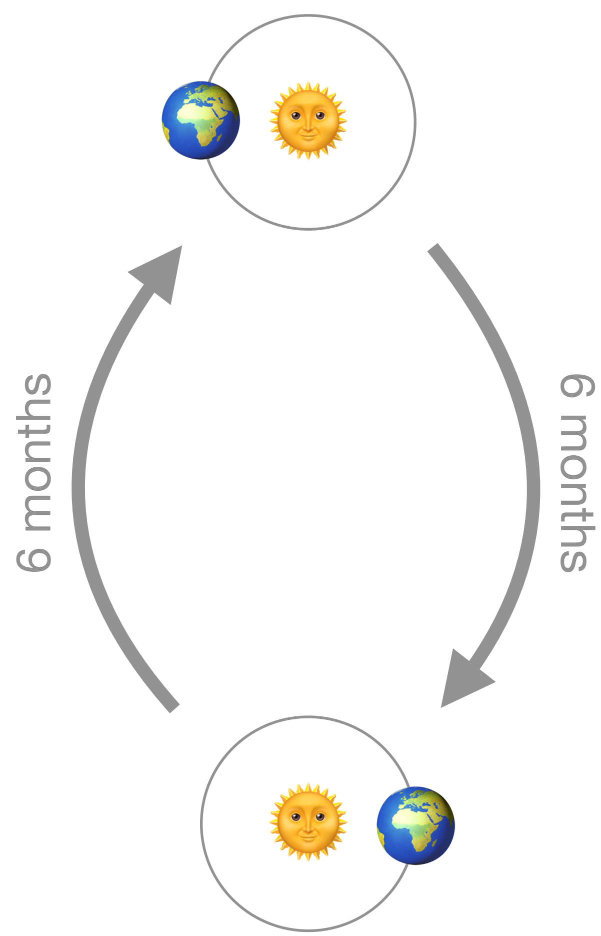
Decomposing the system

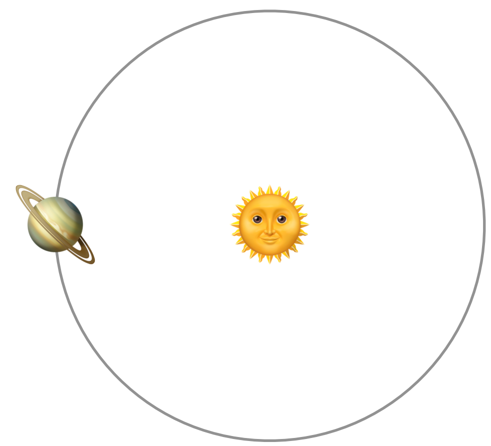


Decomposing the system

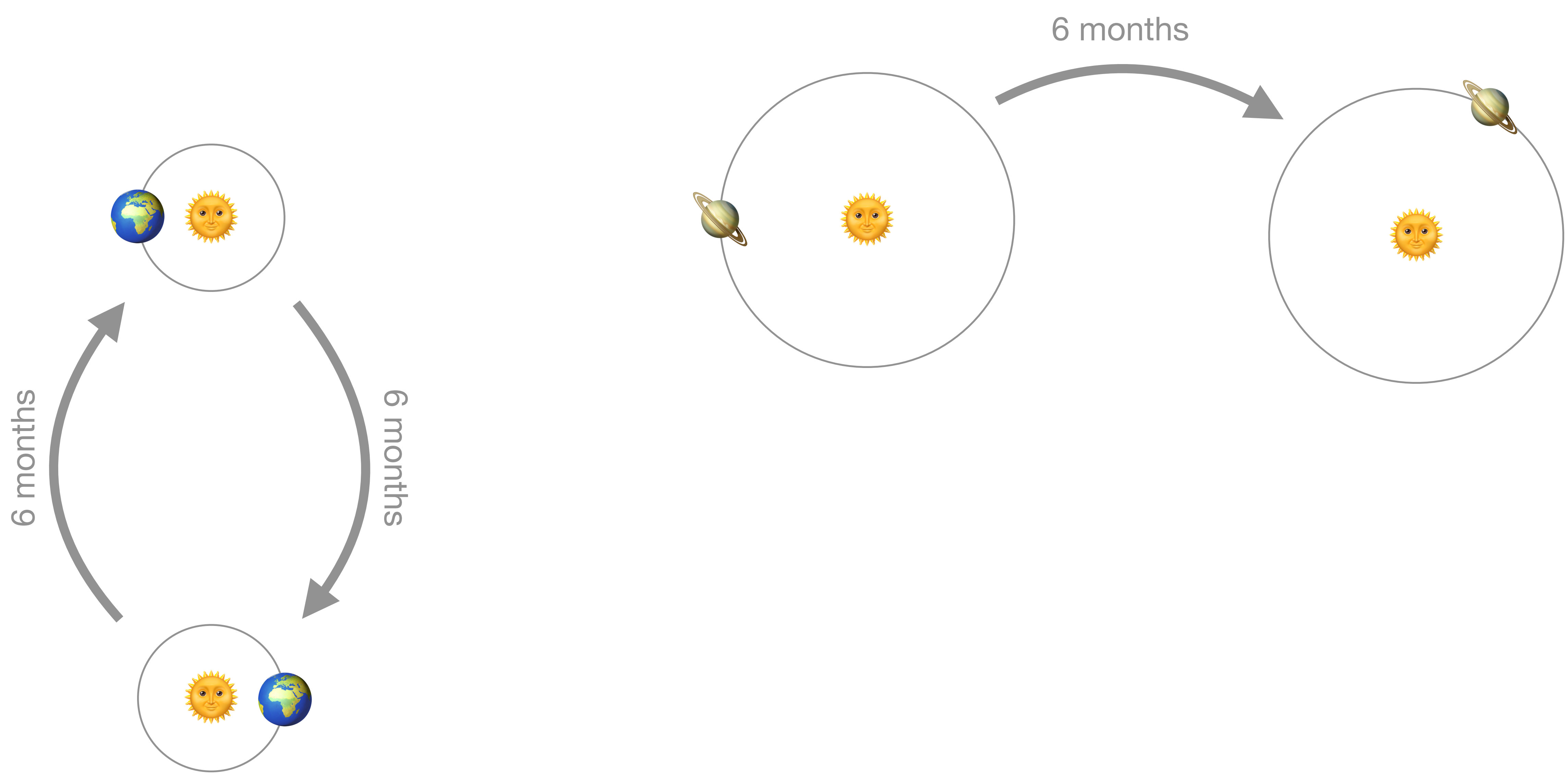


Decomposing the system

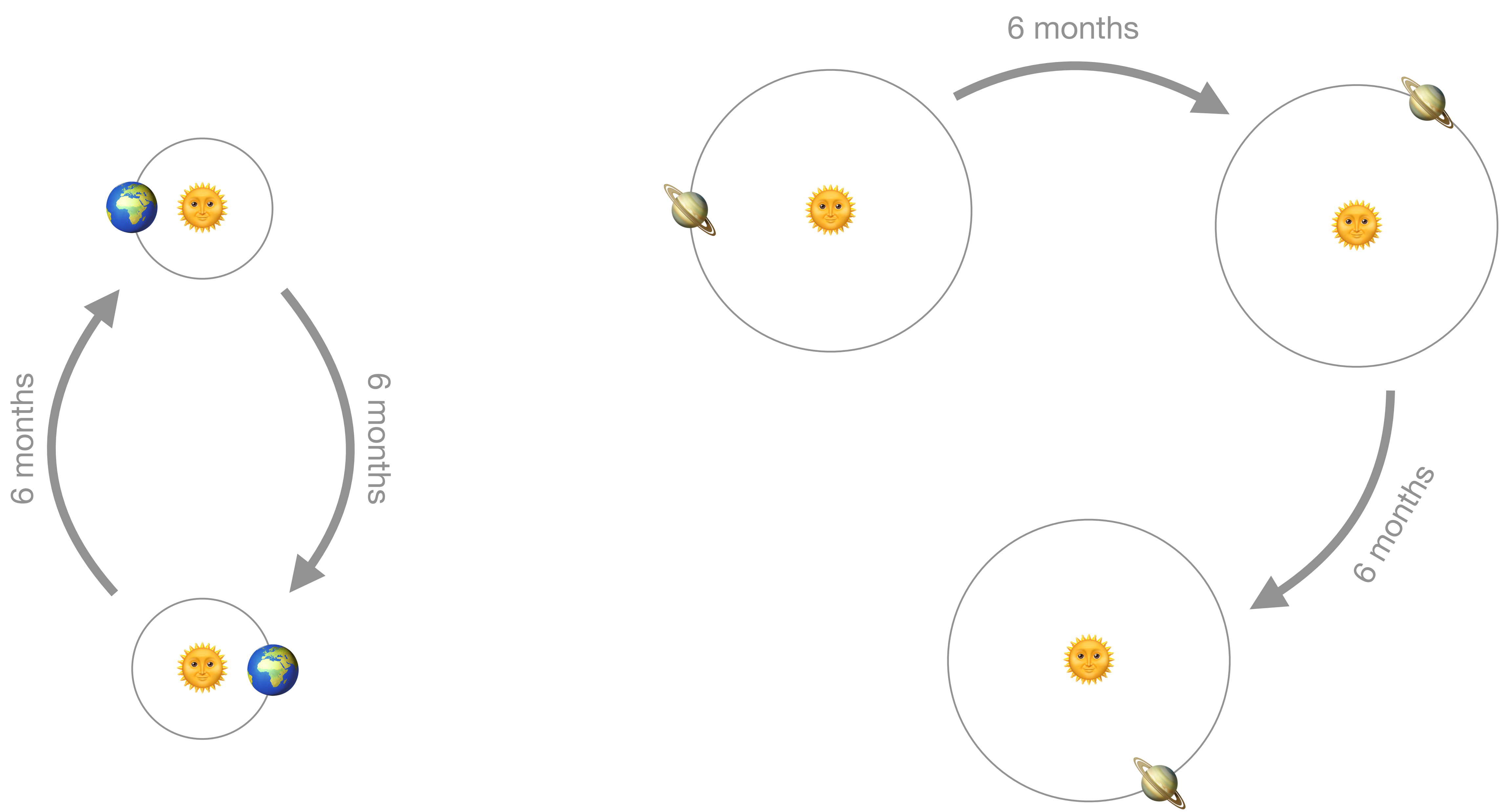




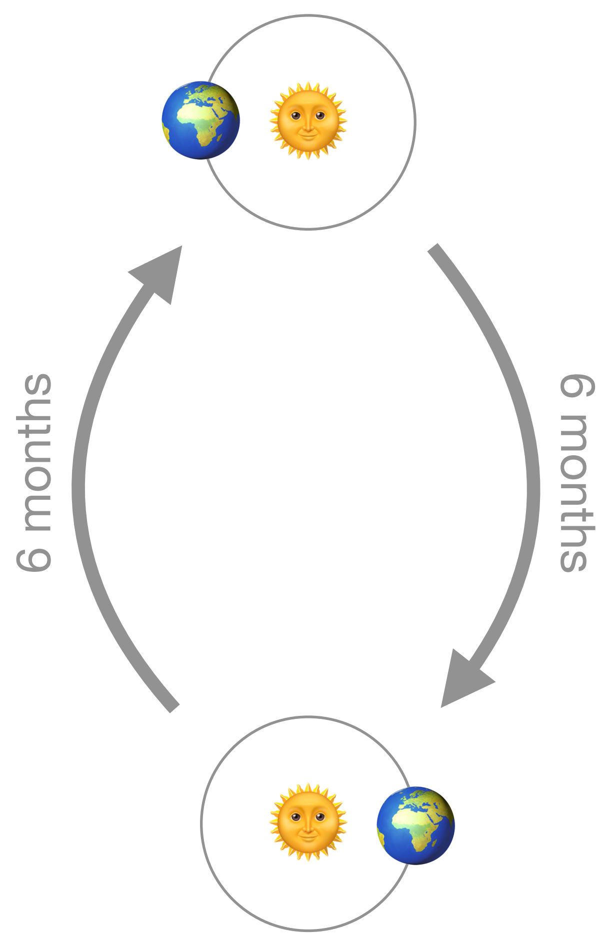
Decomposing the system

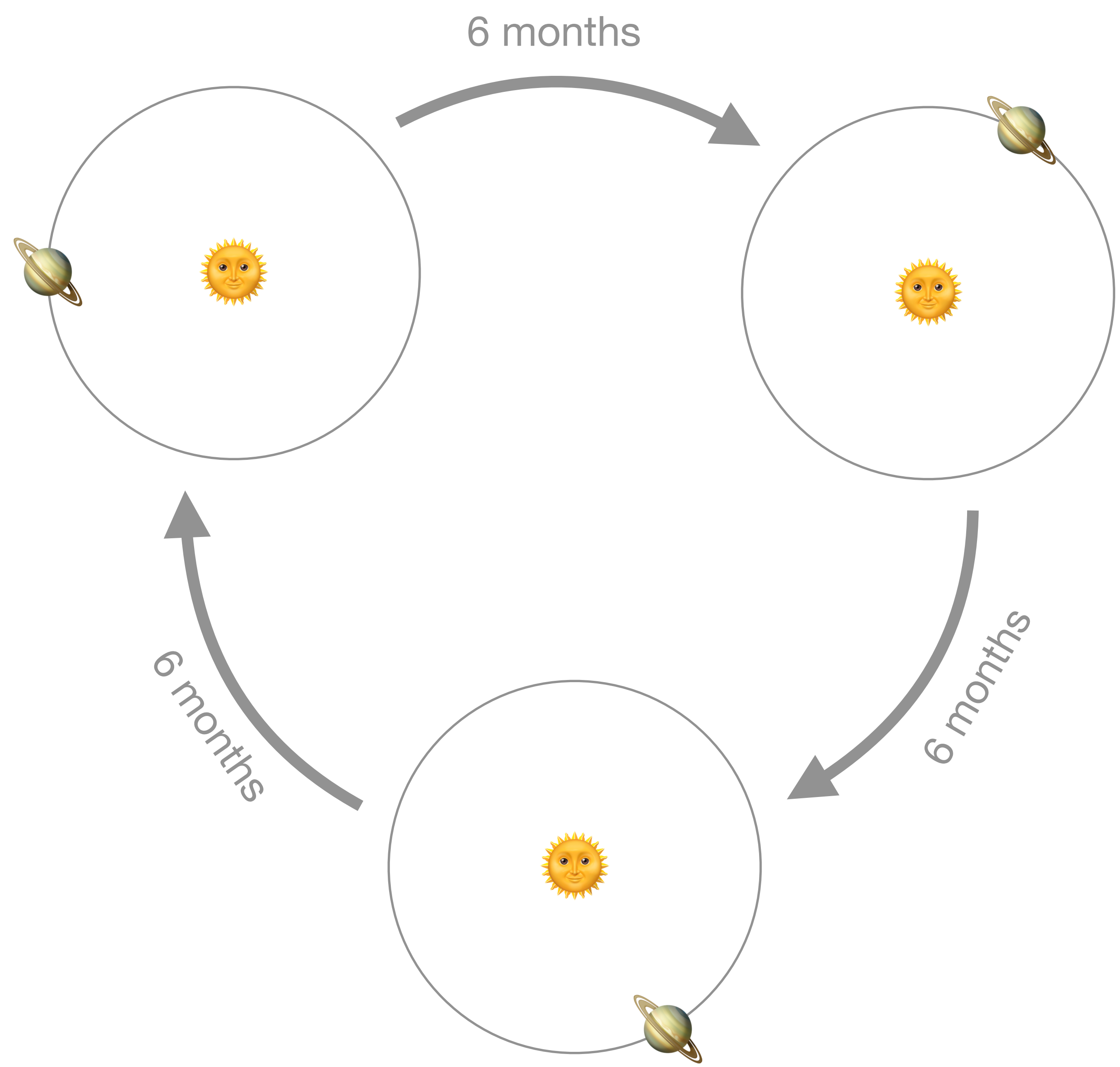


Decomposing the system

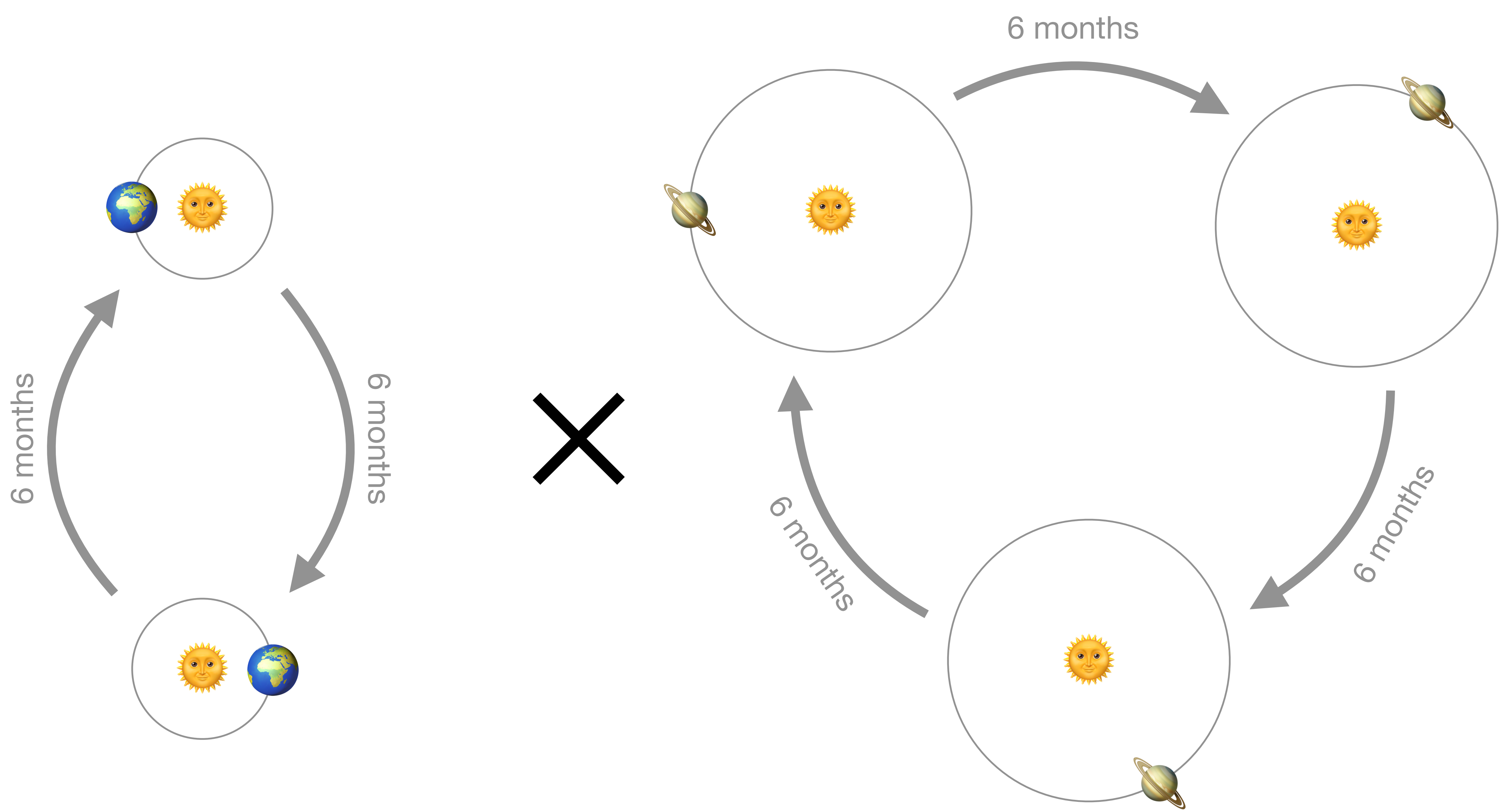


Decomposing the system



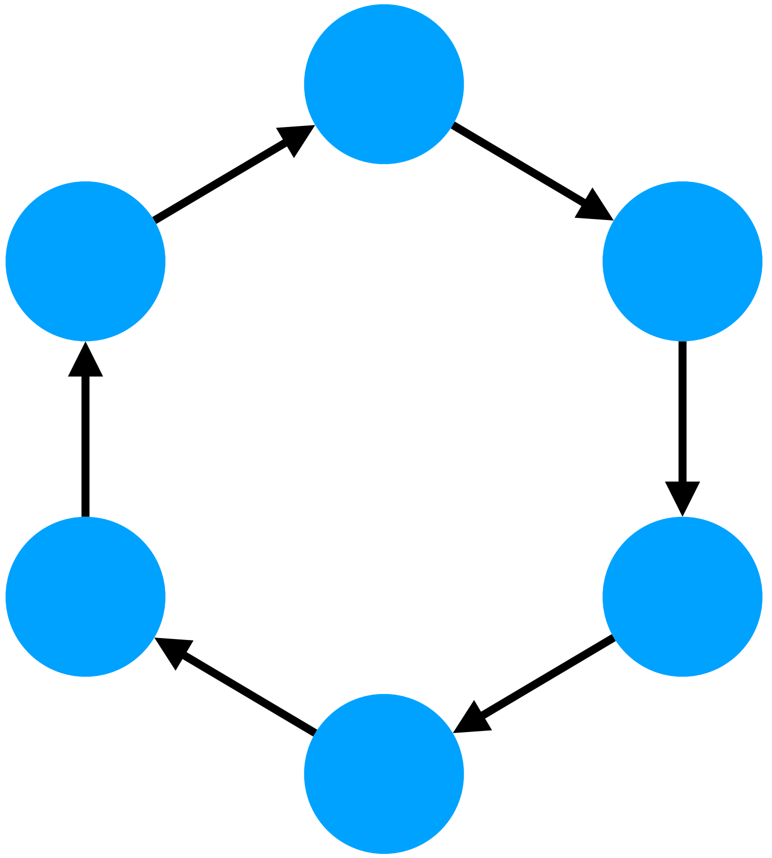


Decomposing the system

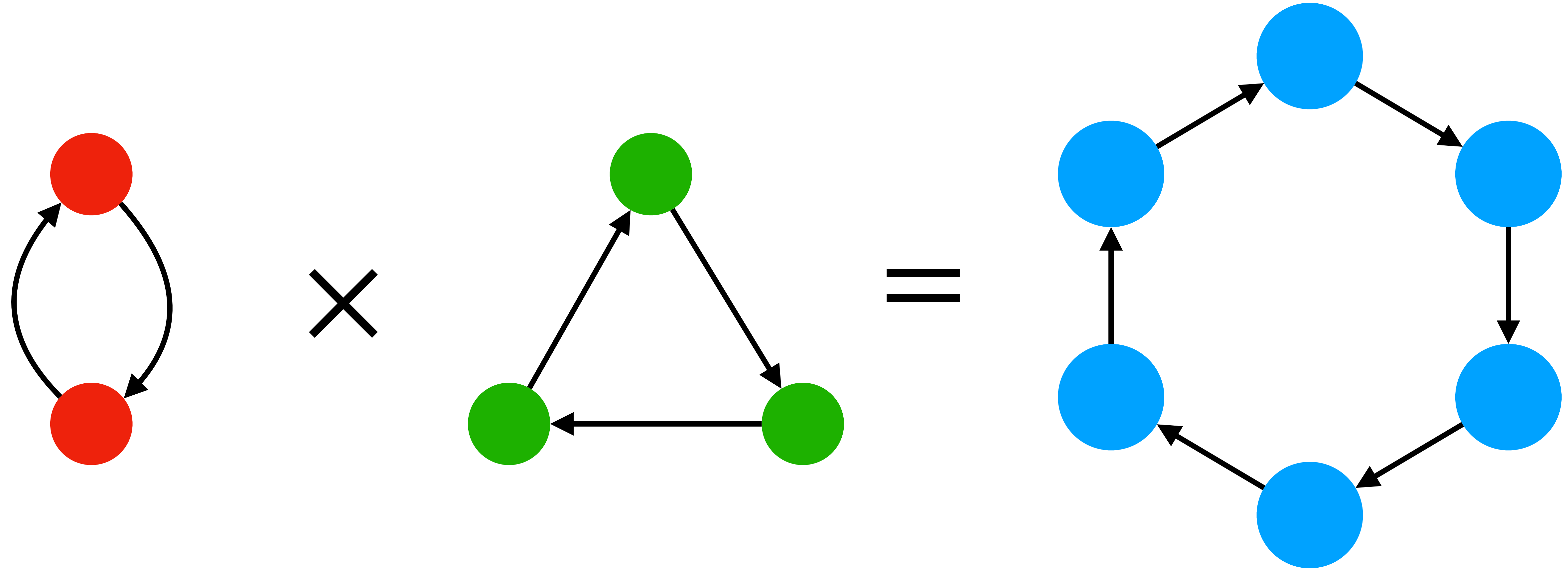


What if our instruments are less sophisticated?

Abstract evolution of the system

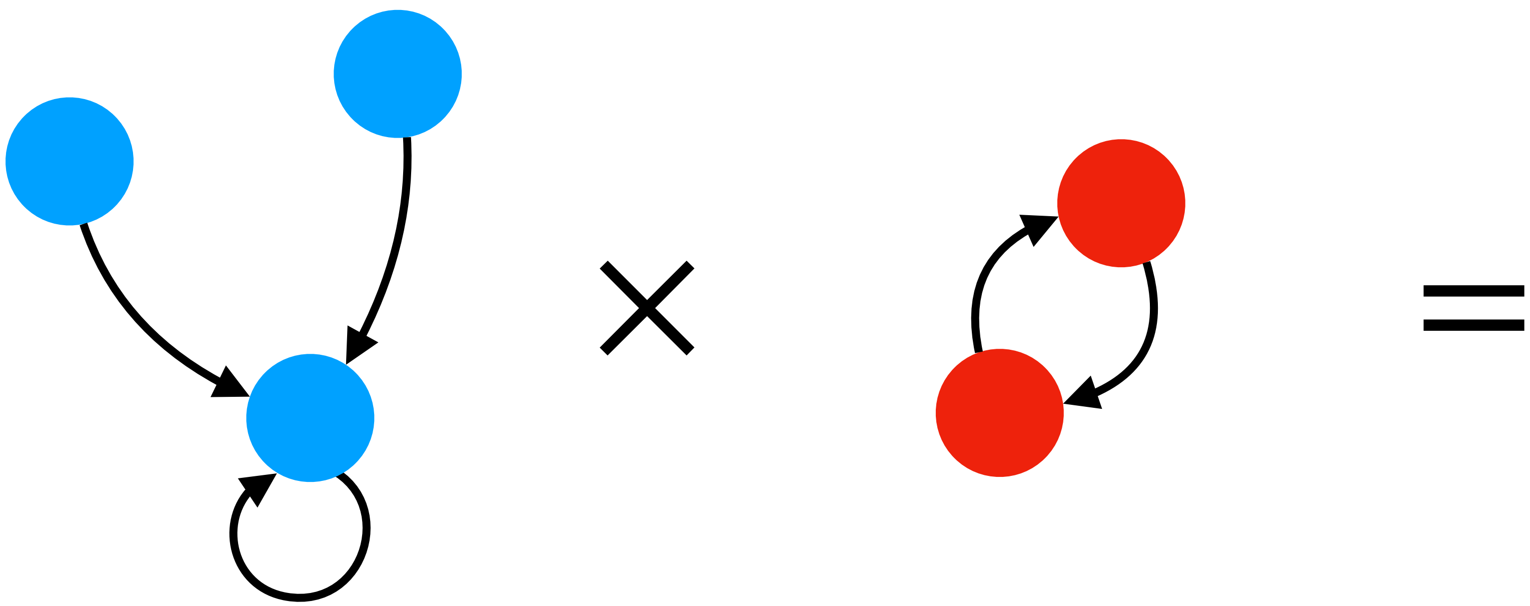


Abstract evolution of the system

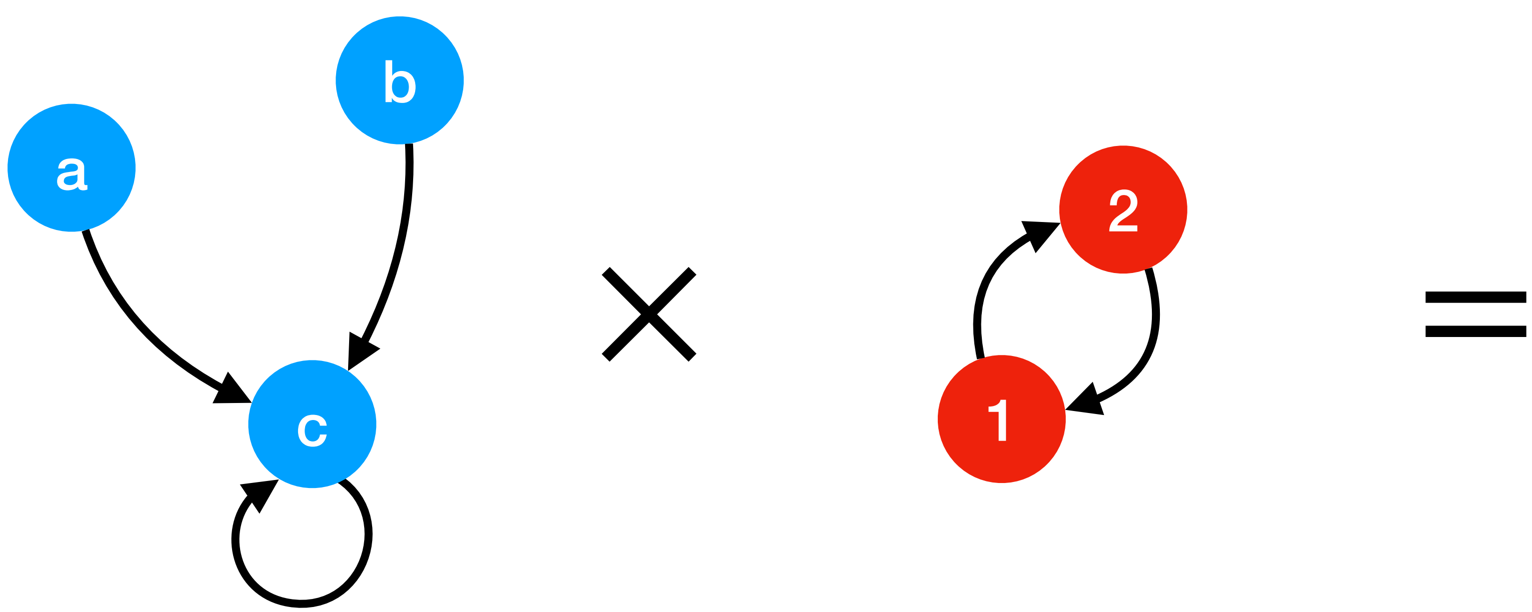


Product of dynamical systems

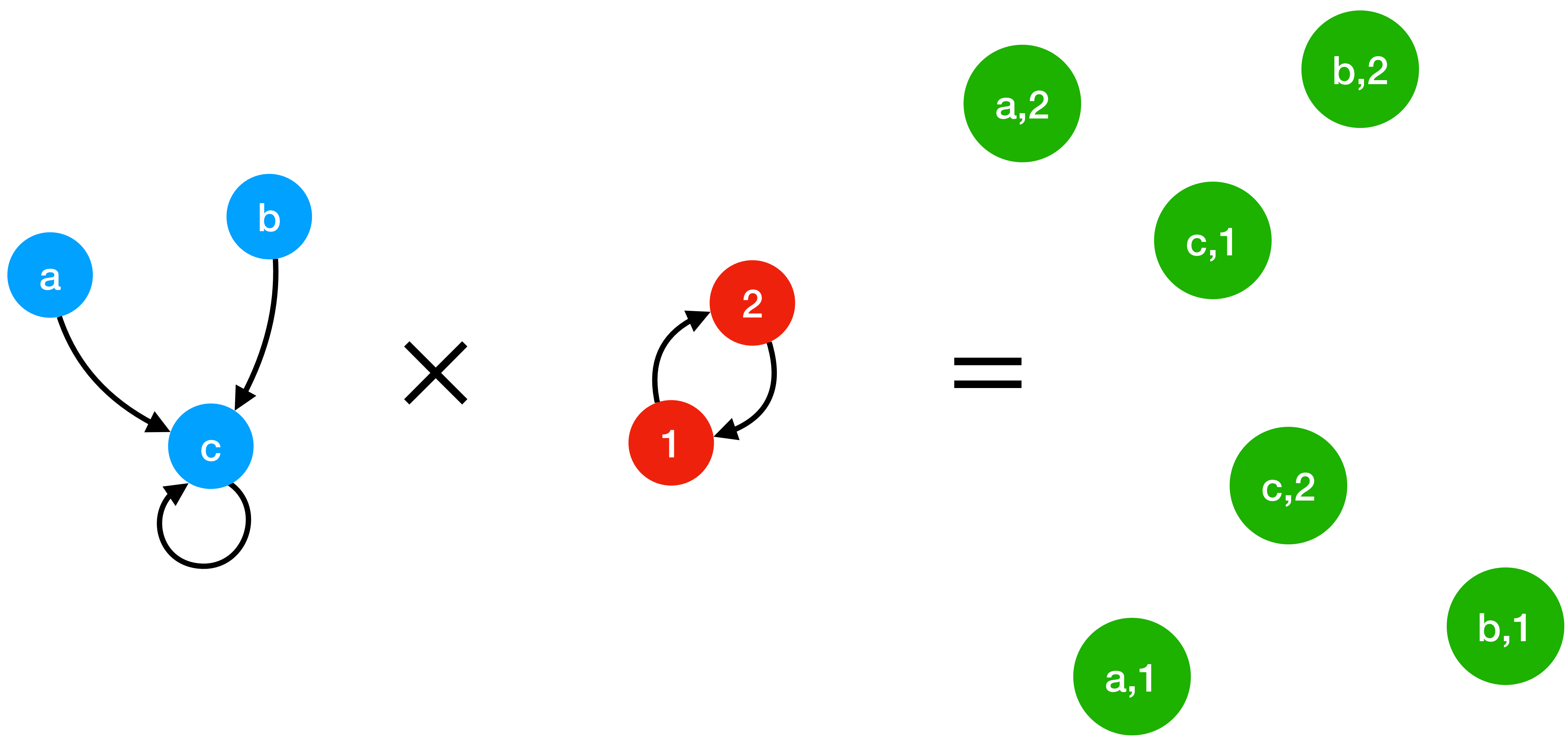
Product of systems



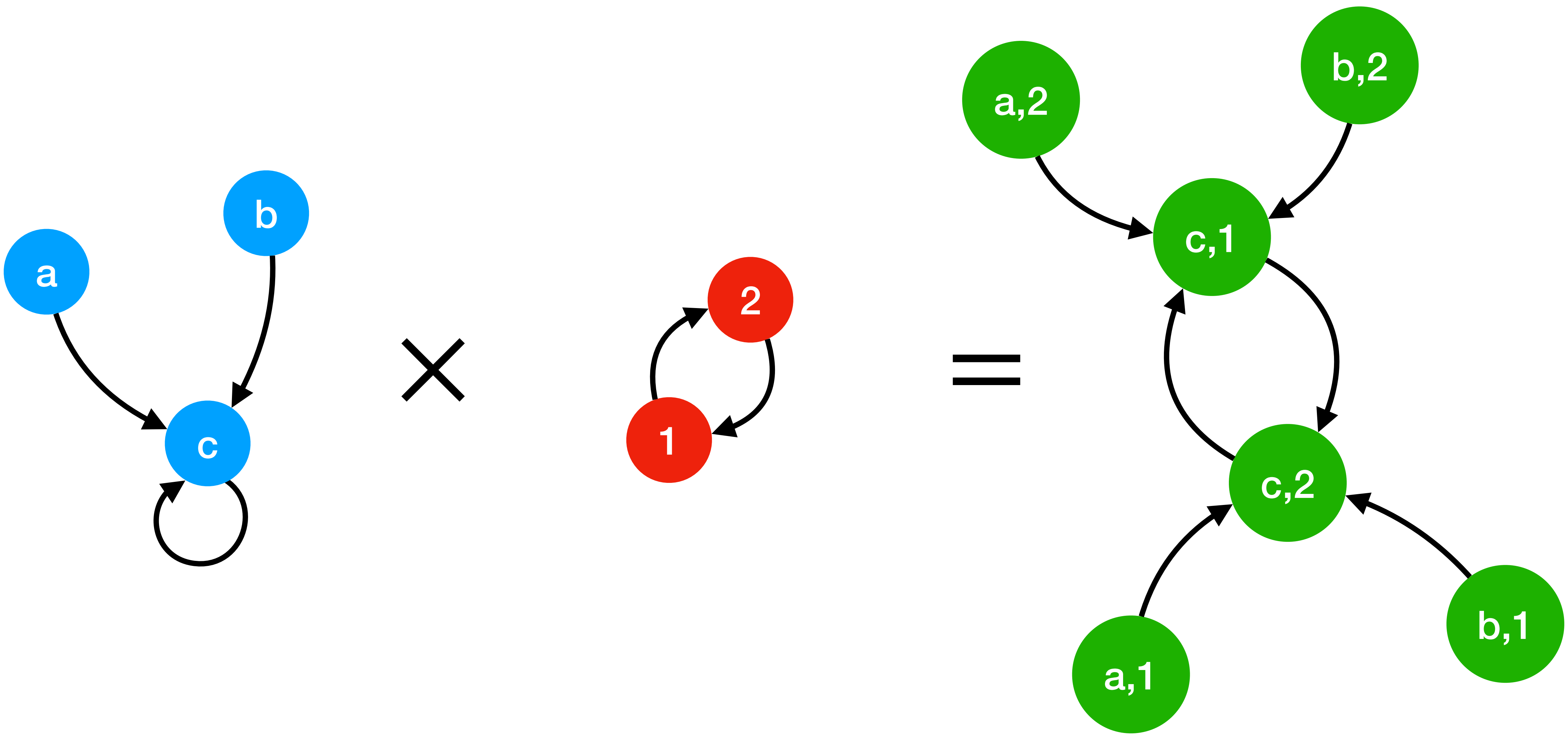
Give temporary names to the states



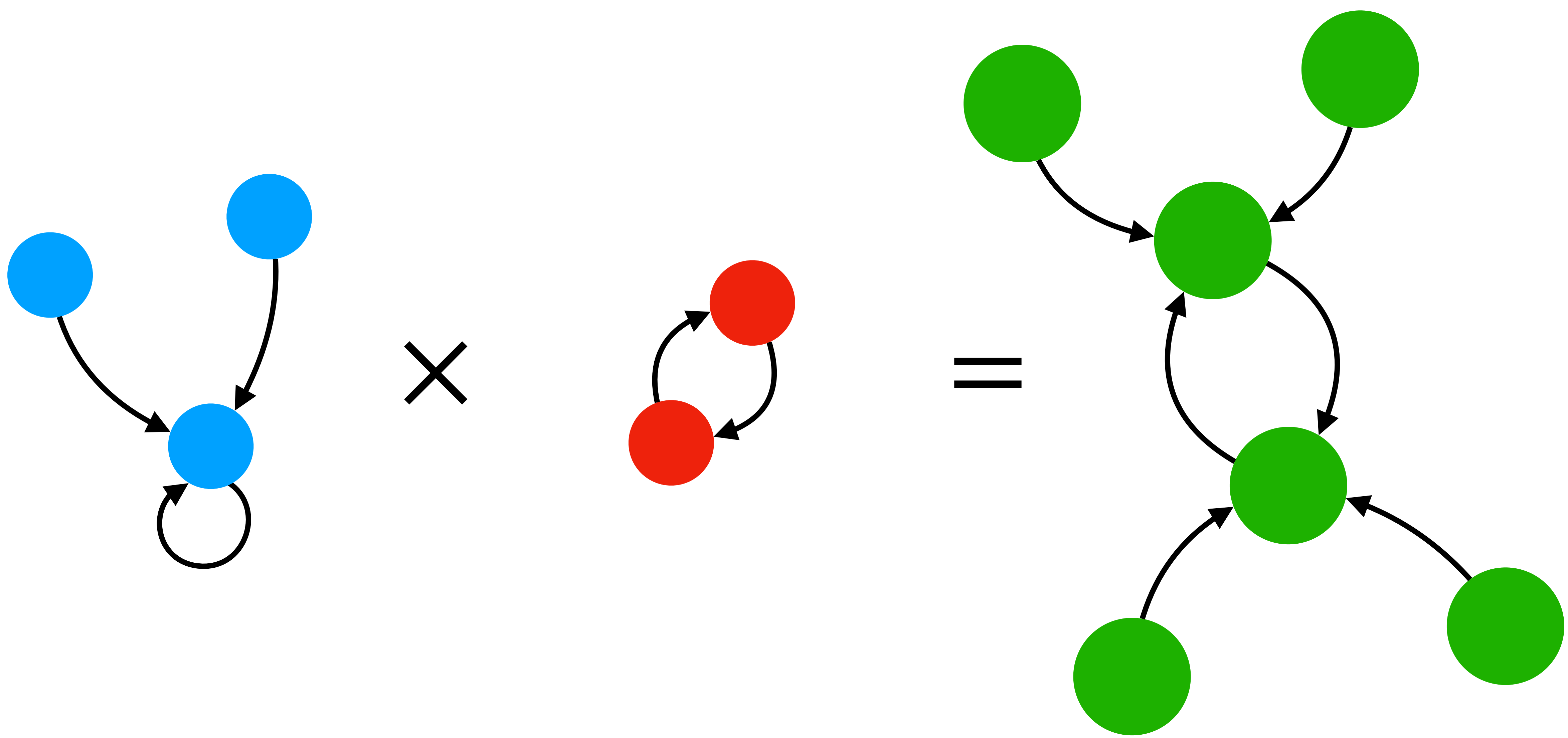
Compute the Cartesian product



Add the arcs between states

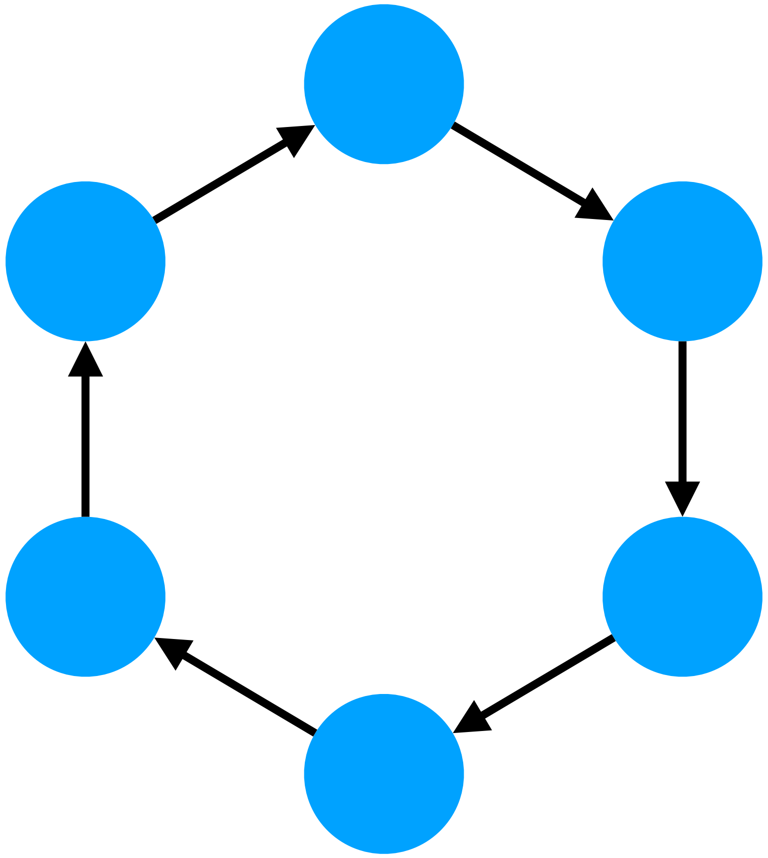


Forget the names once again

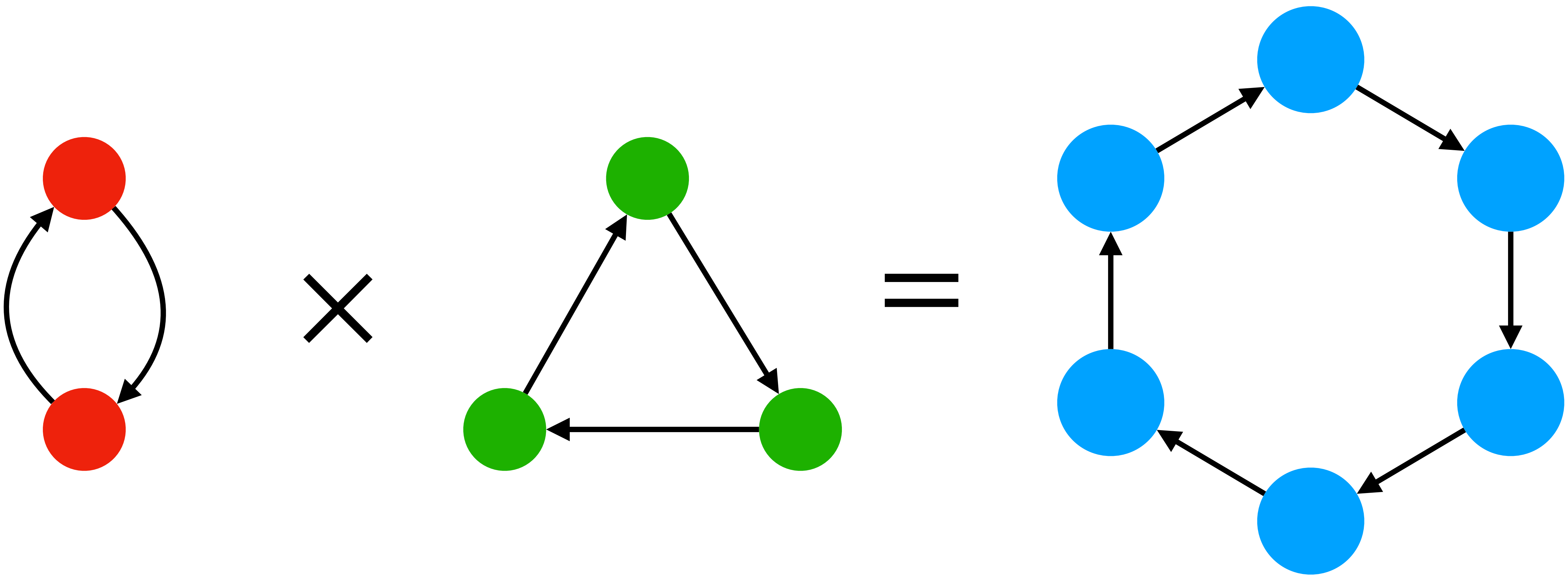


Back to our planetary system

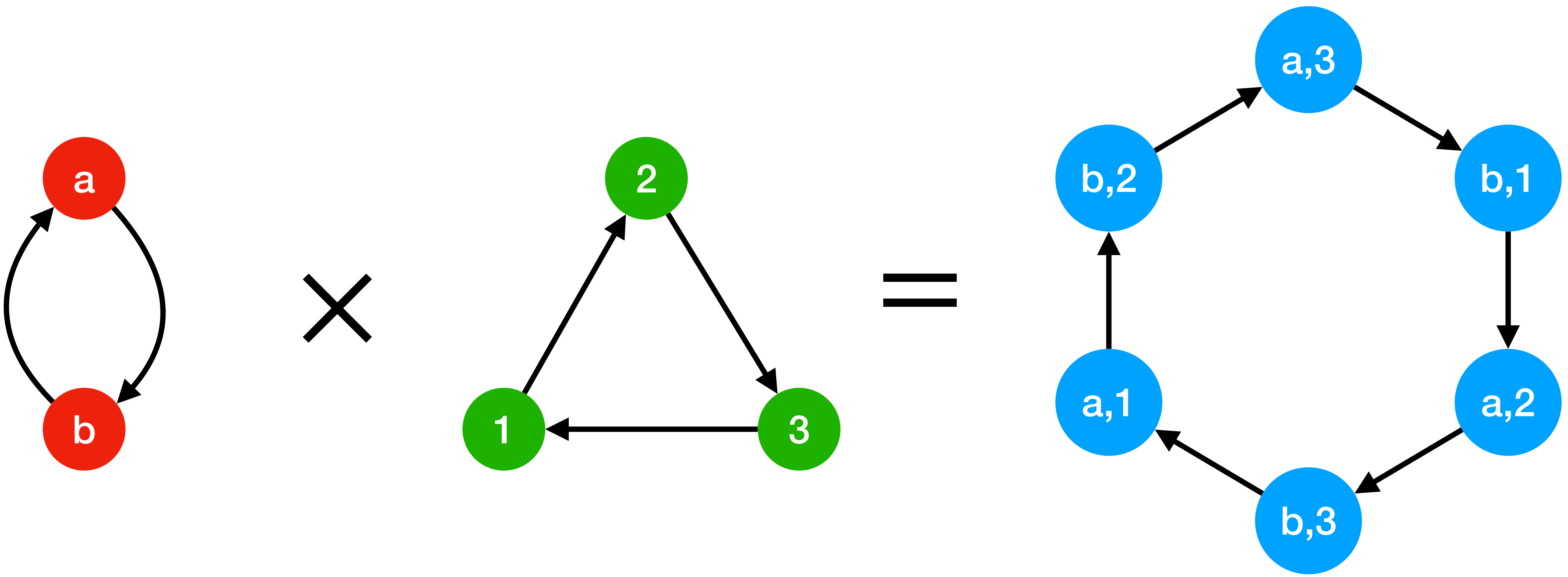
Decomposition



Decomposition



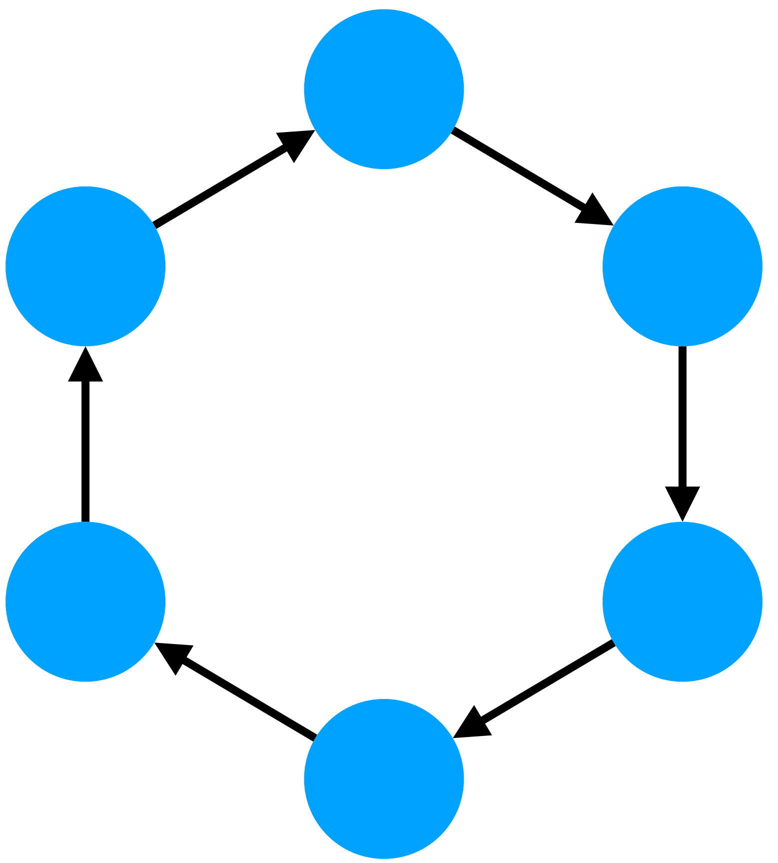
Decomposition



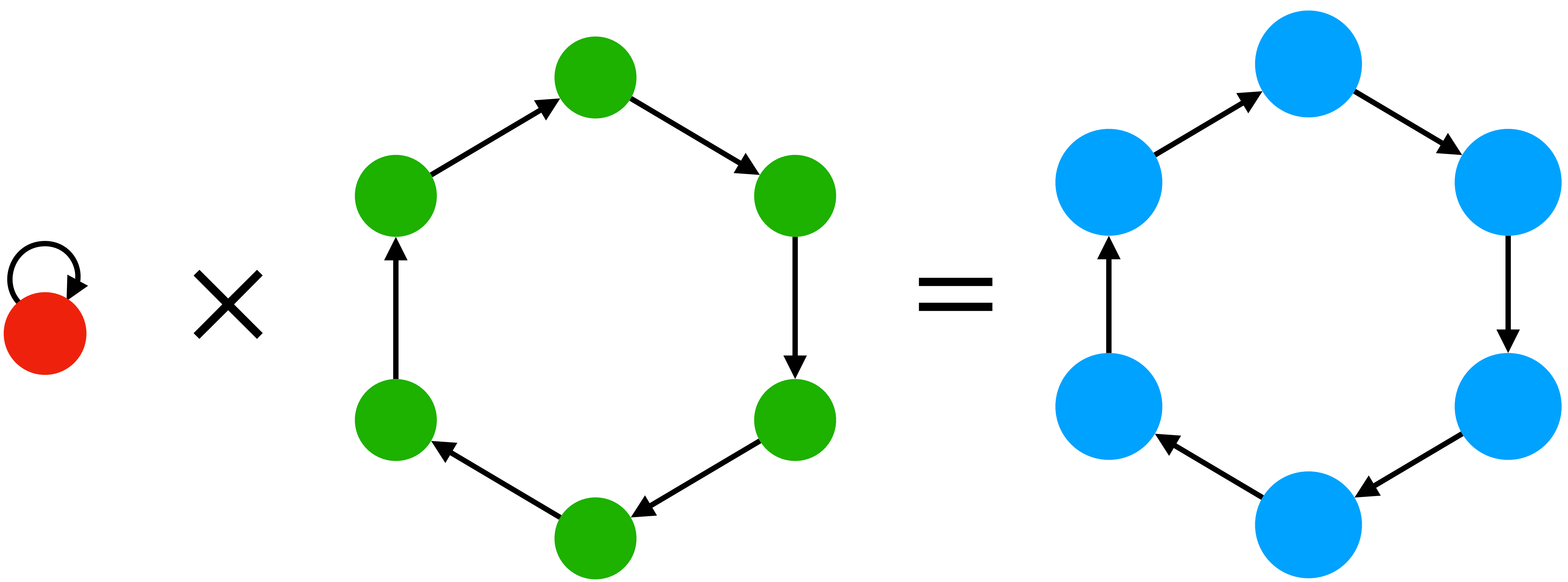
Any other decomposition?

Another decomposition

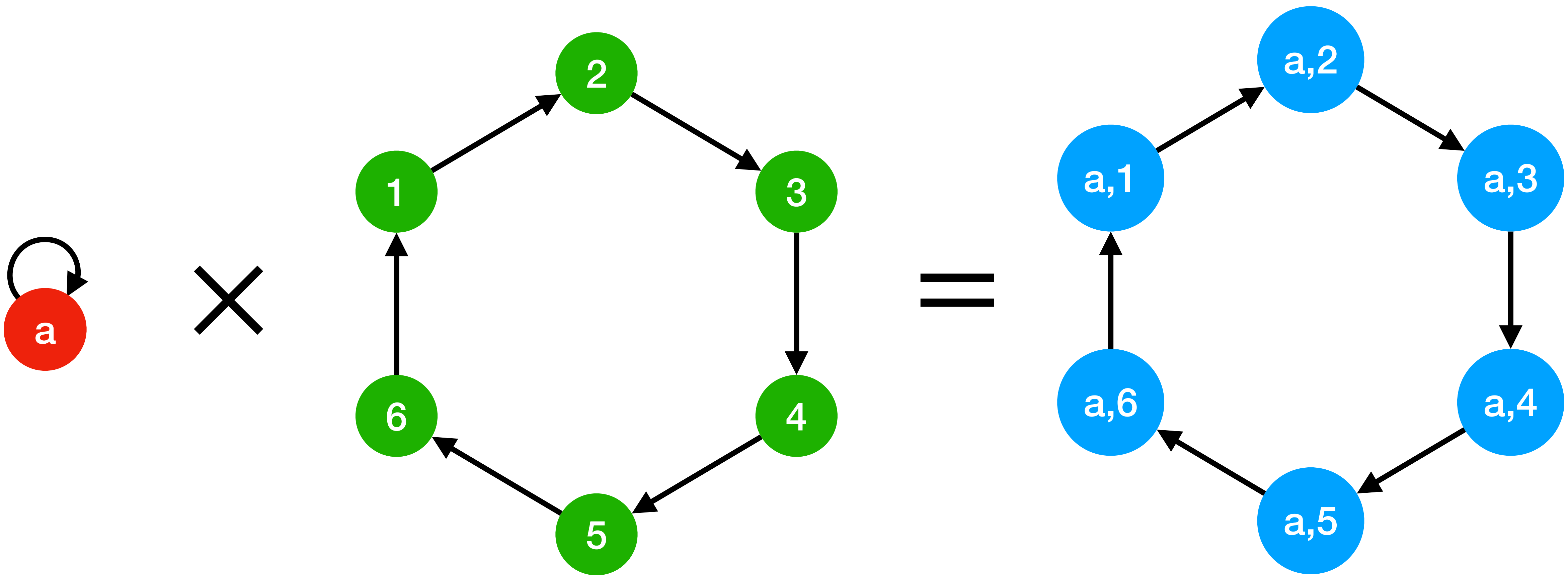
=



Another decomposition

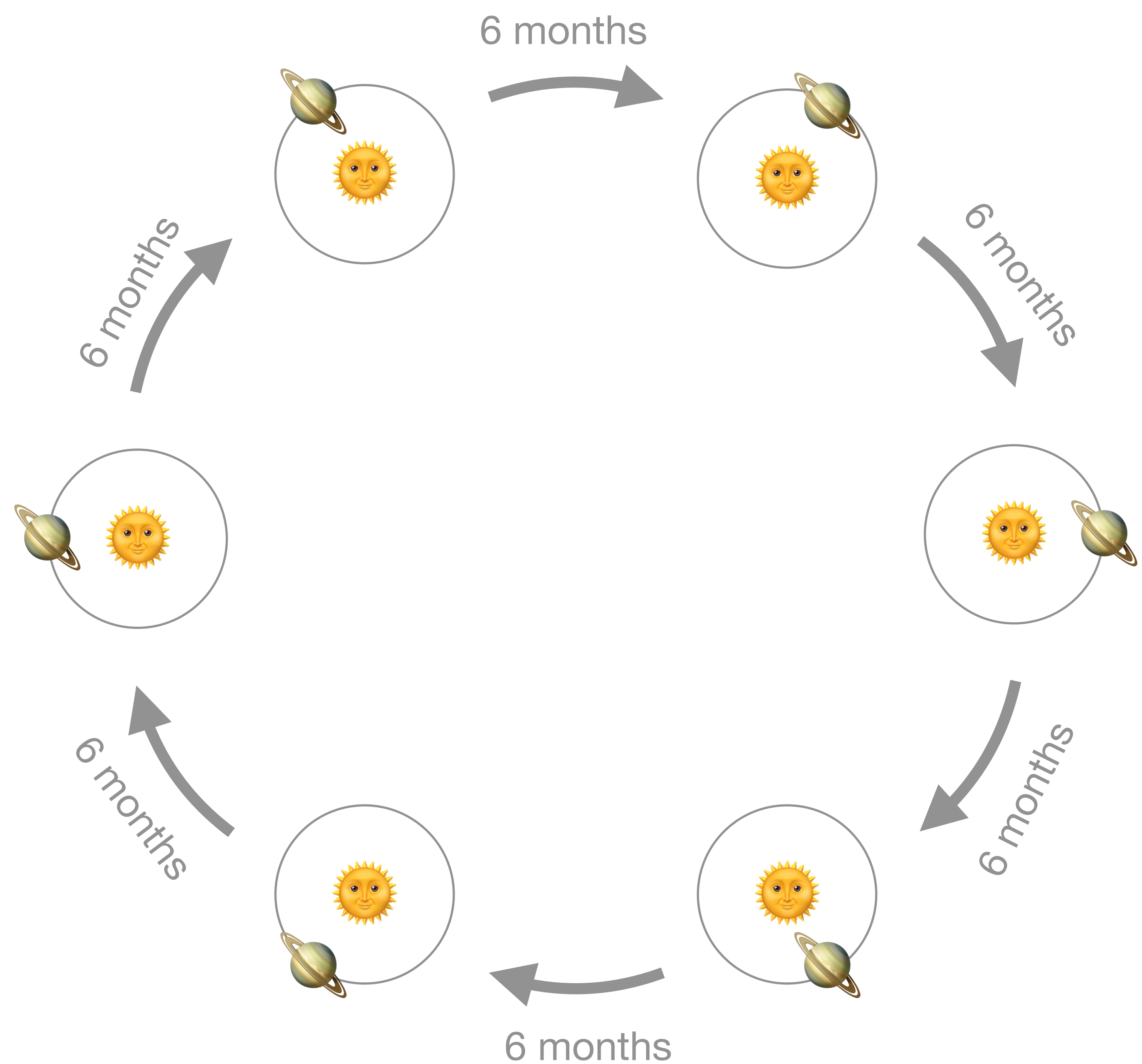


Another decomposition

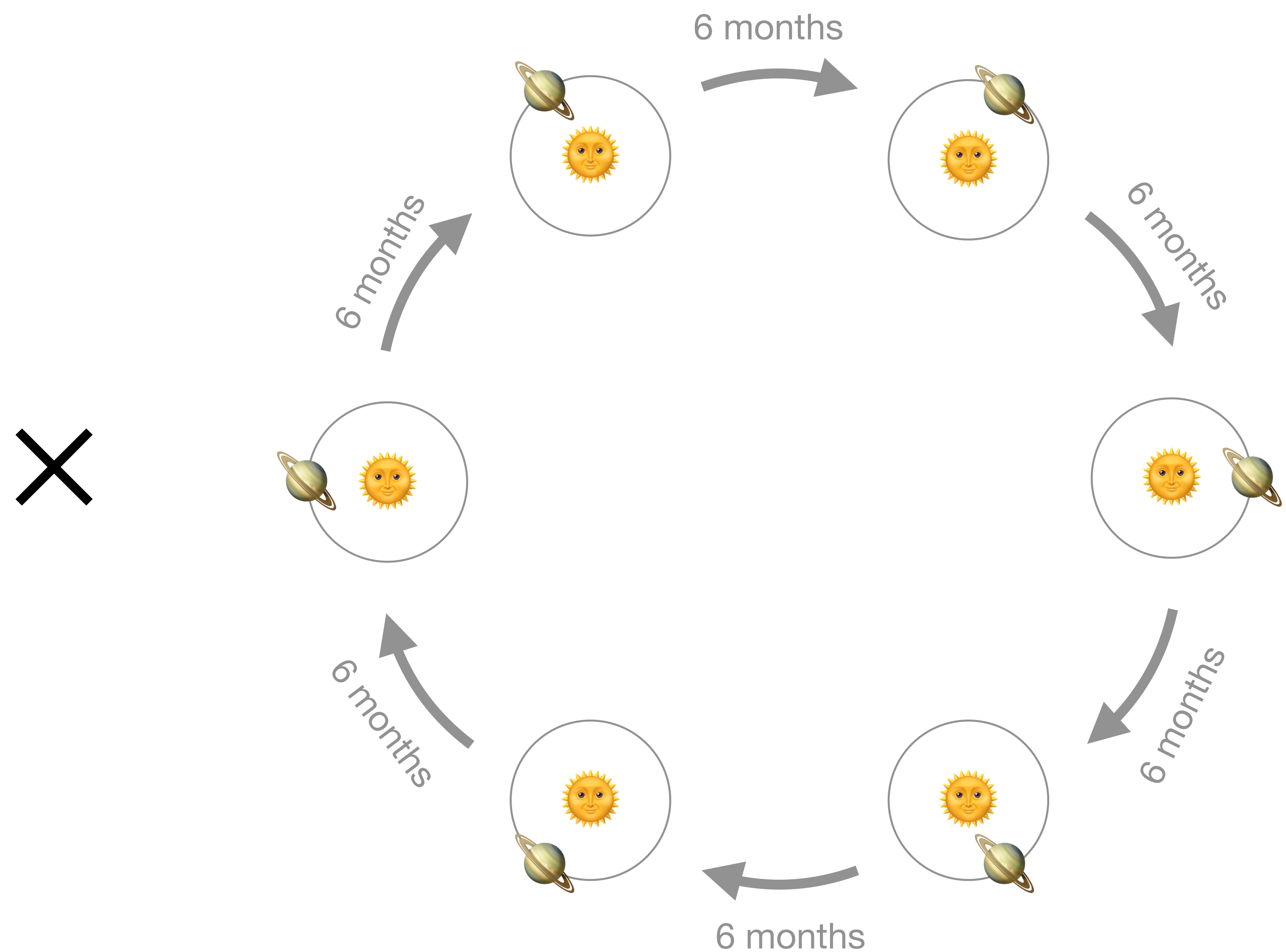


More concretely…

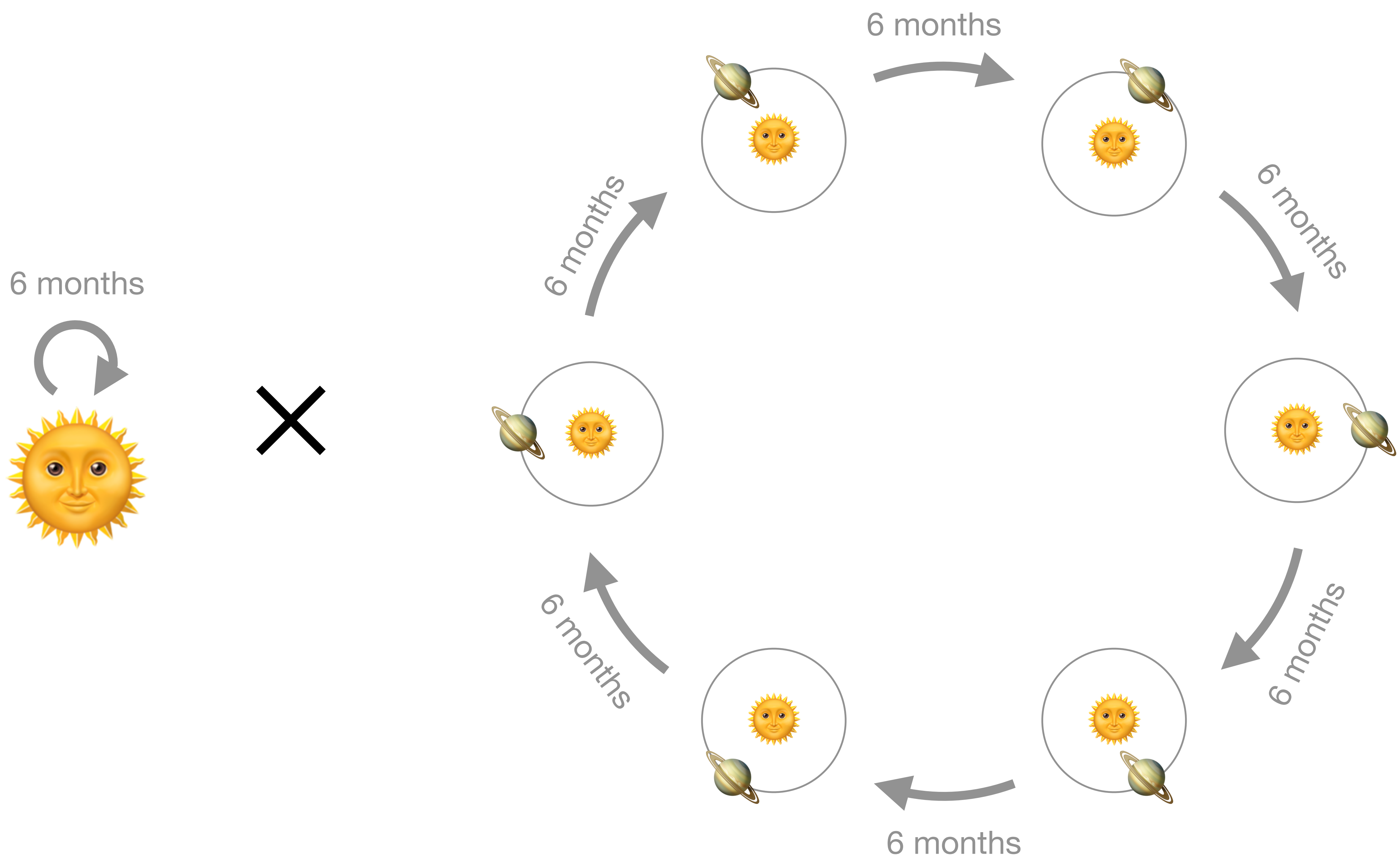
More concretely…



More concretely…

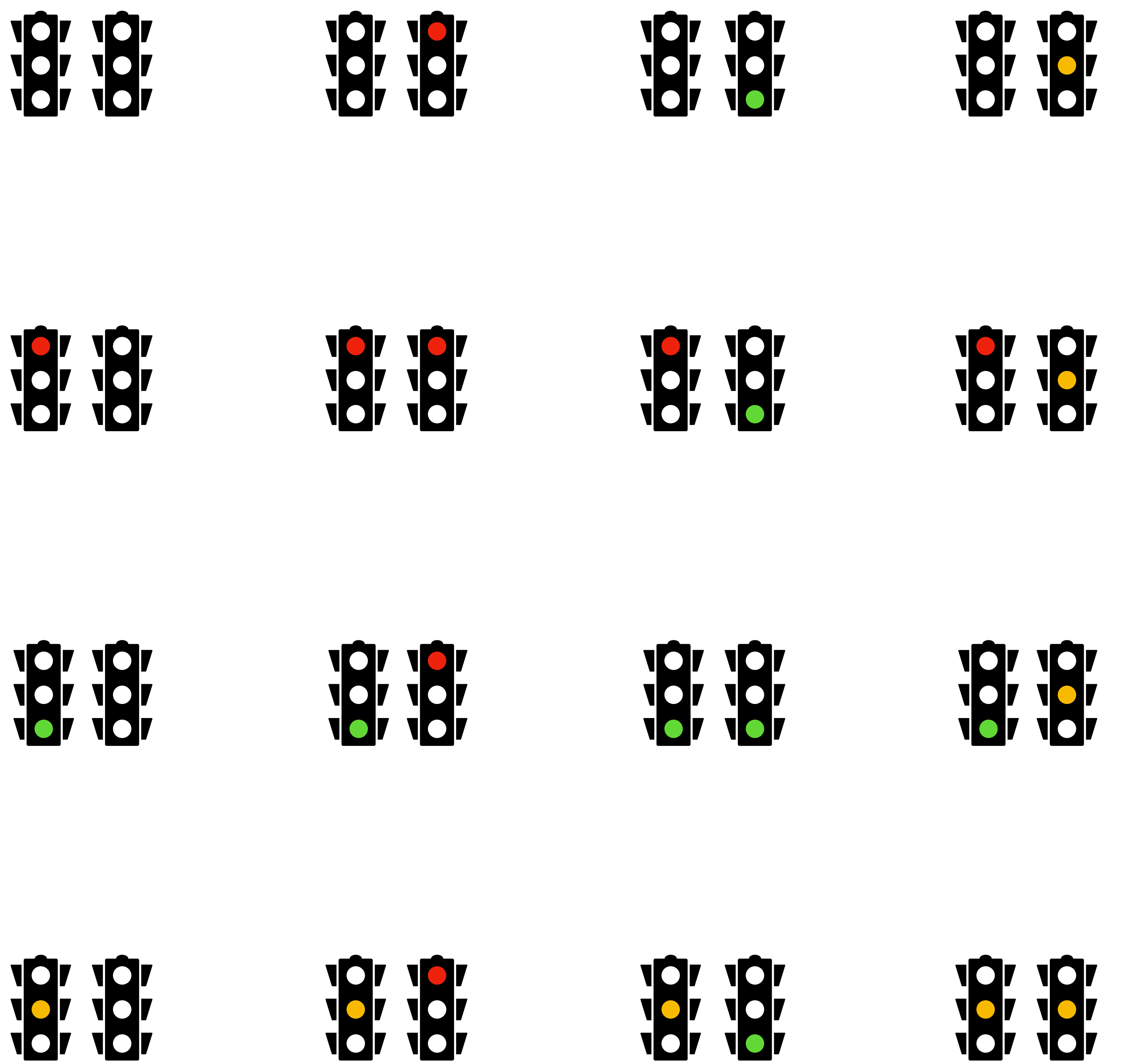


More concretely…

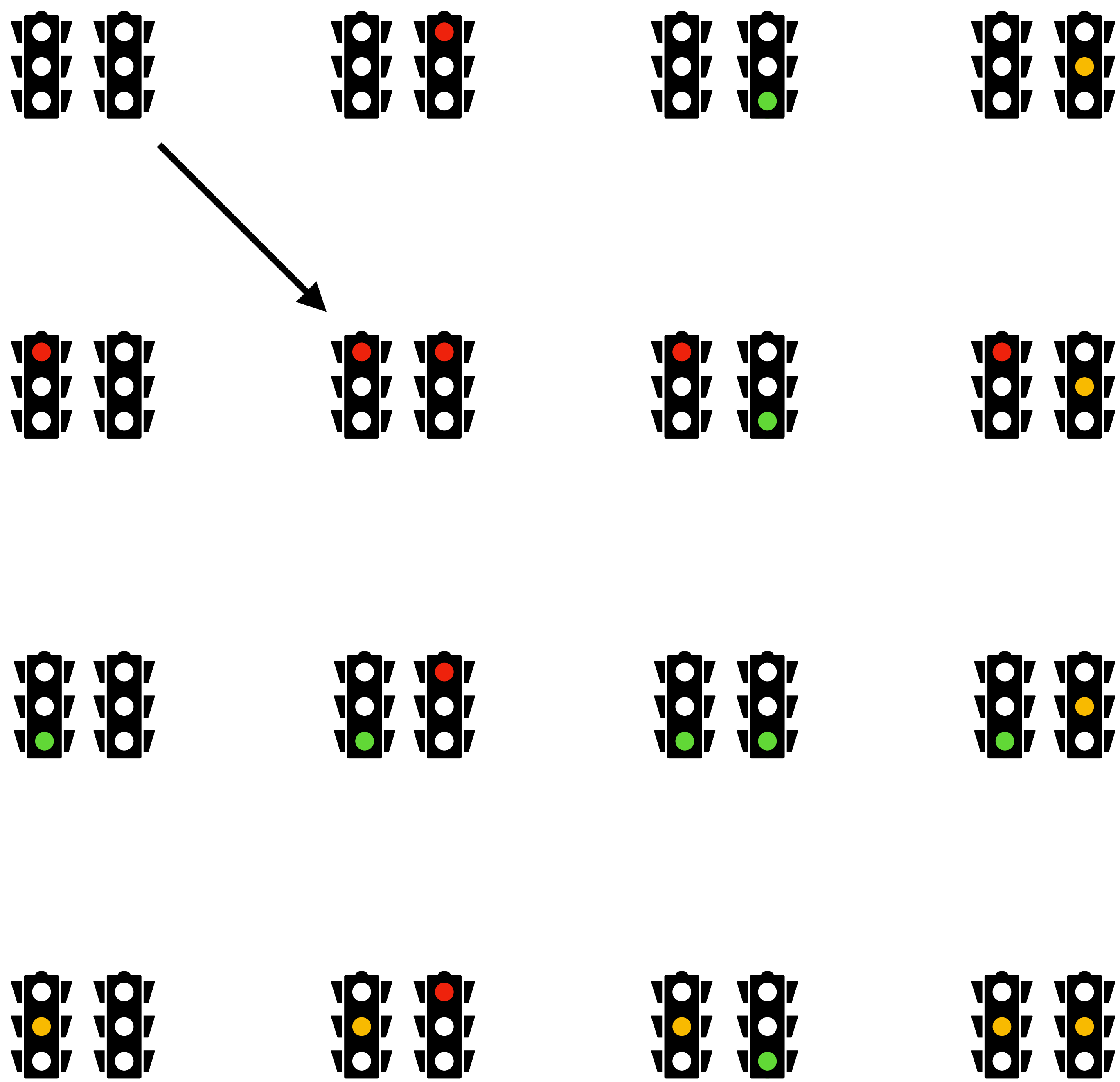


Untangling complex systems

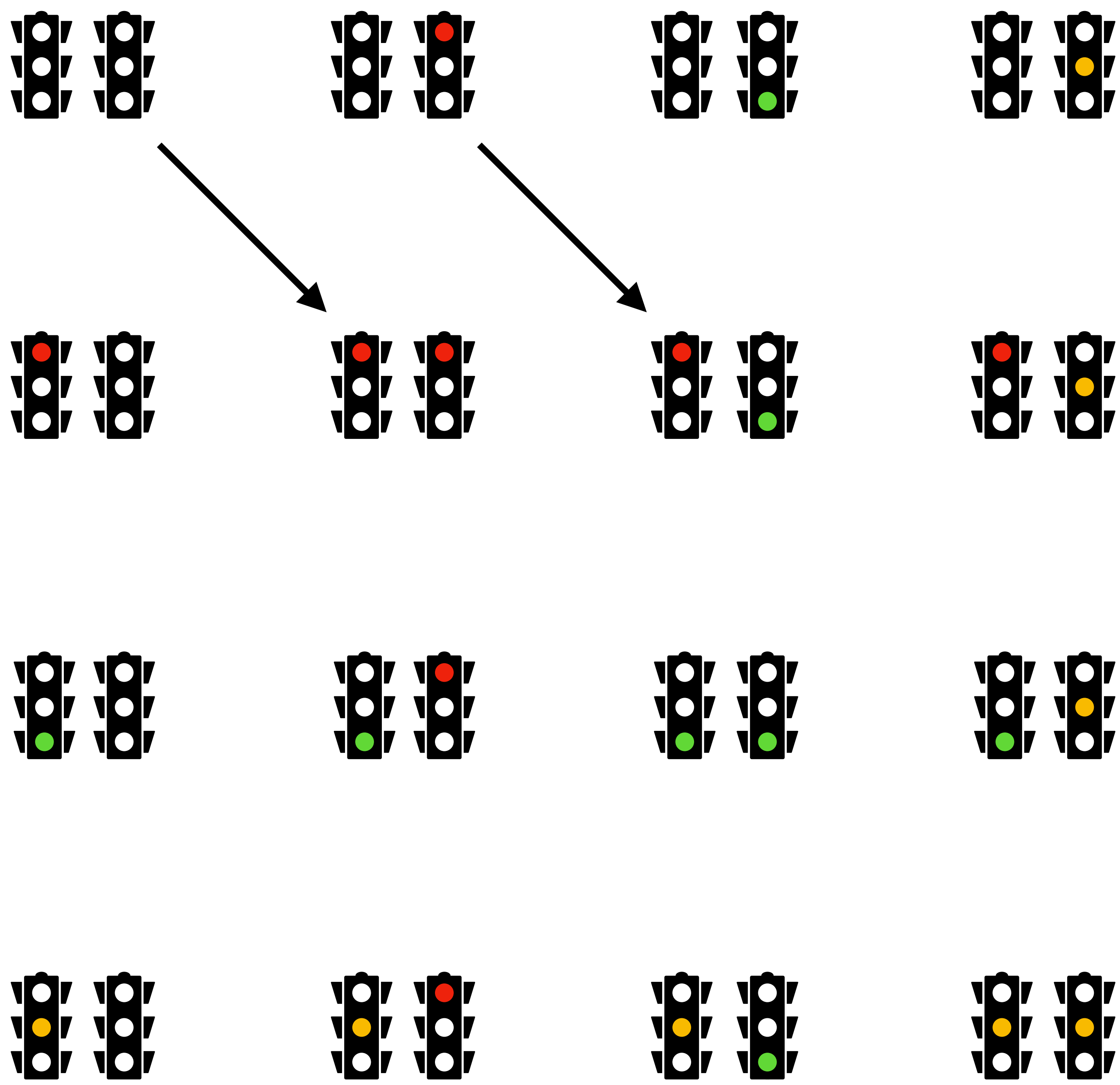
Traffic lights at a crossroads



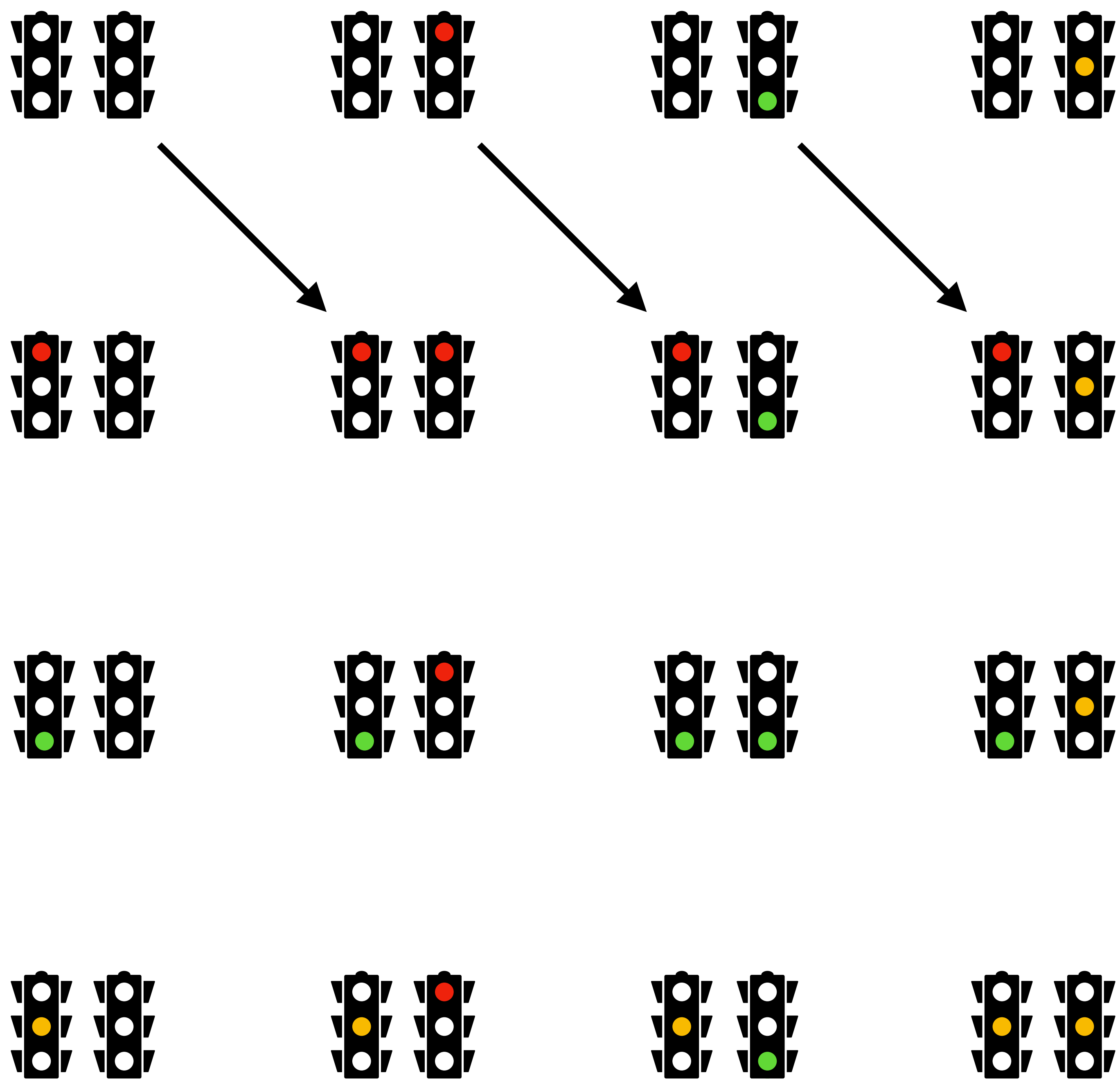
Traffic lights at a crossroads



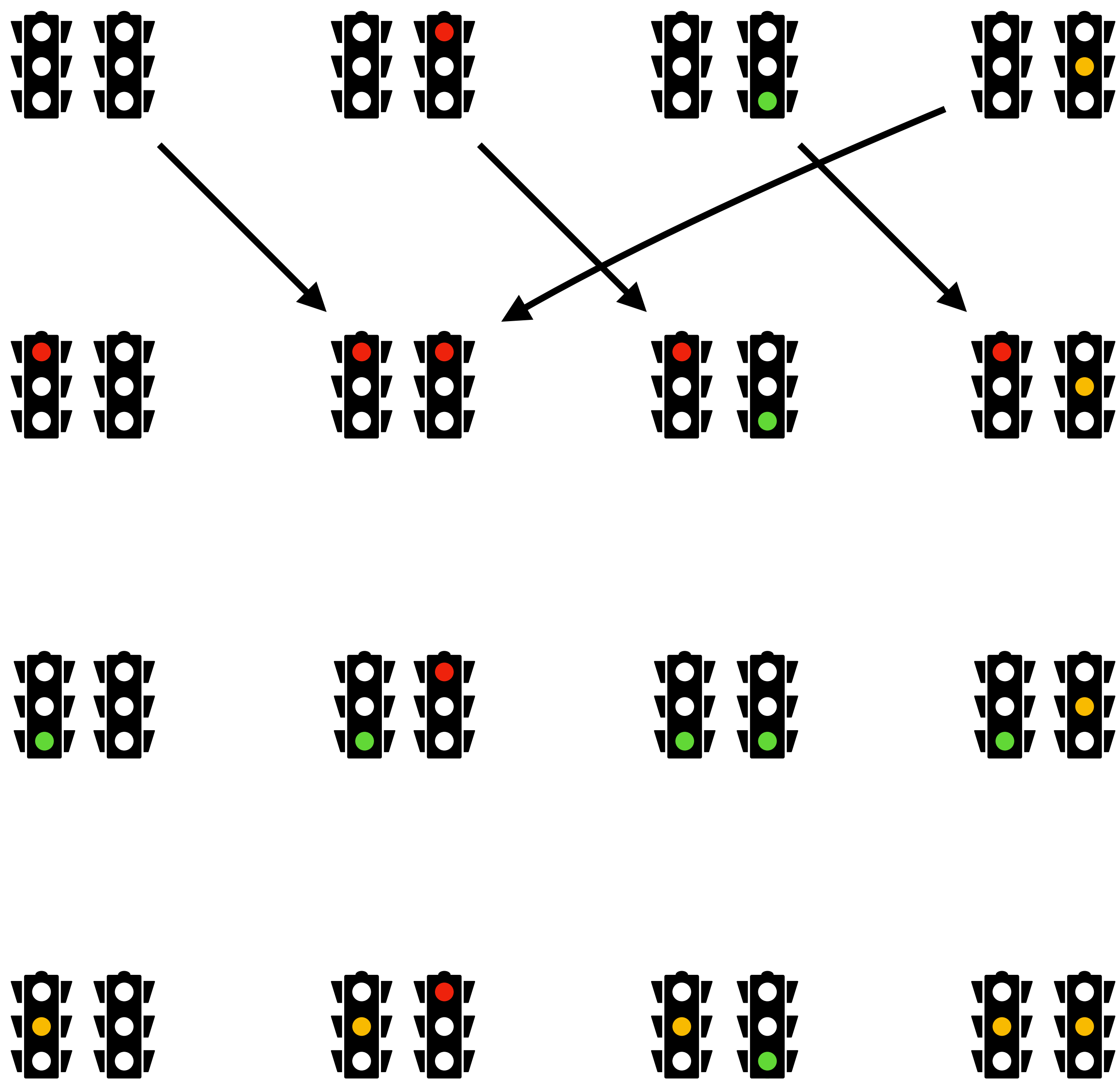
Traffic lights at a crossroads



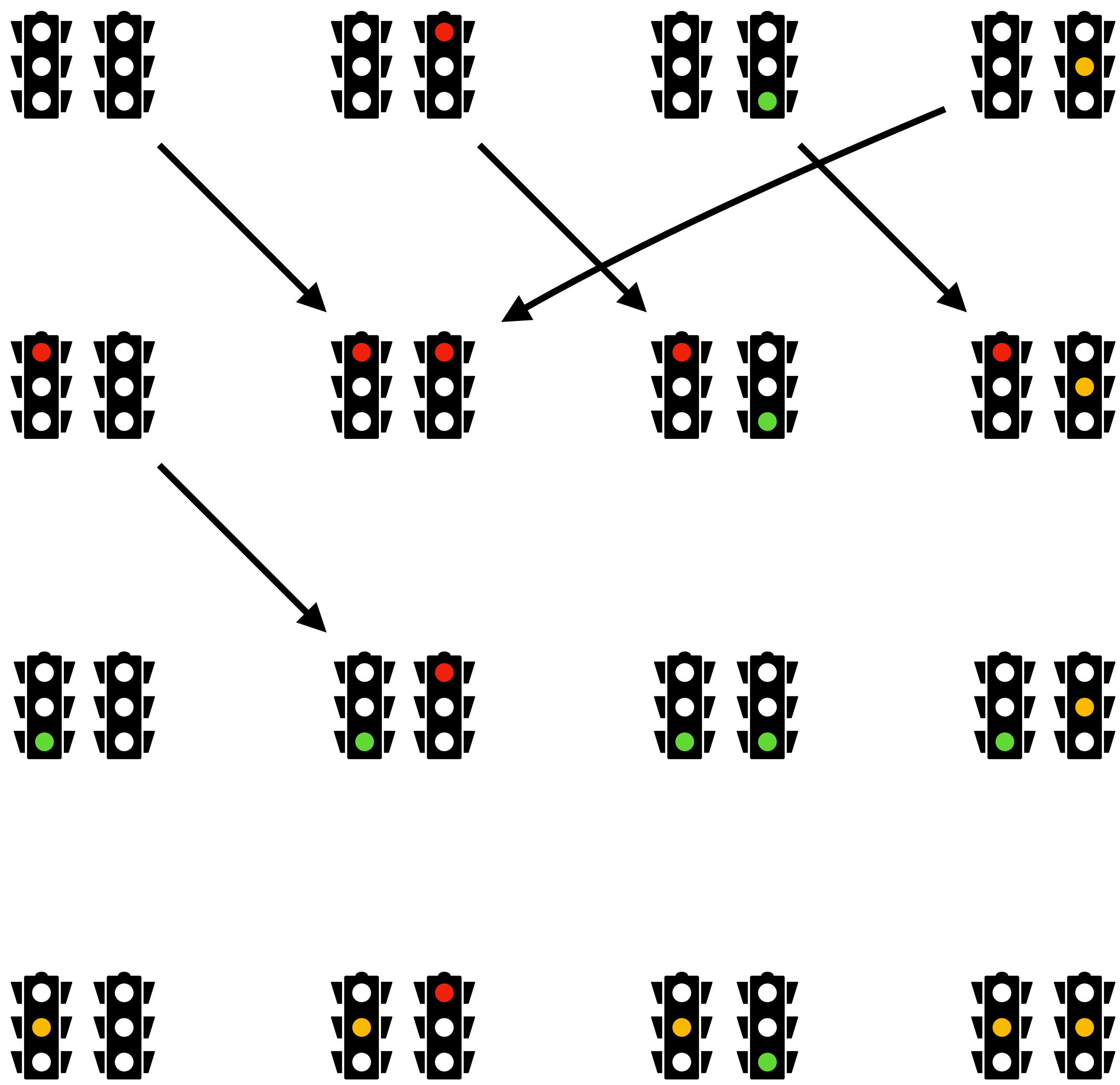
Traffic lights at a crossroads



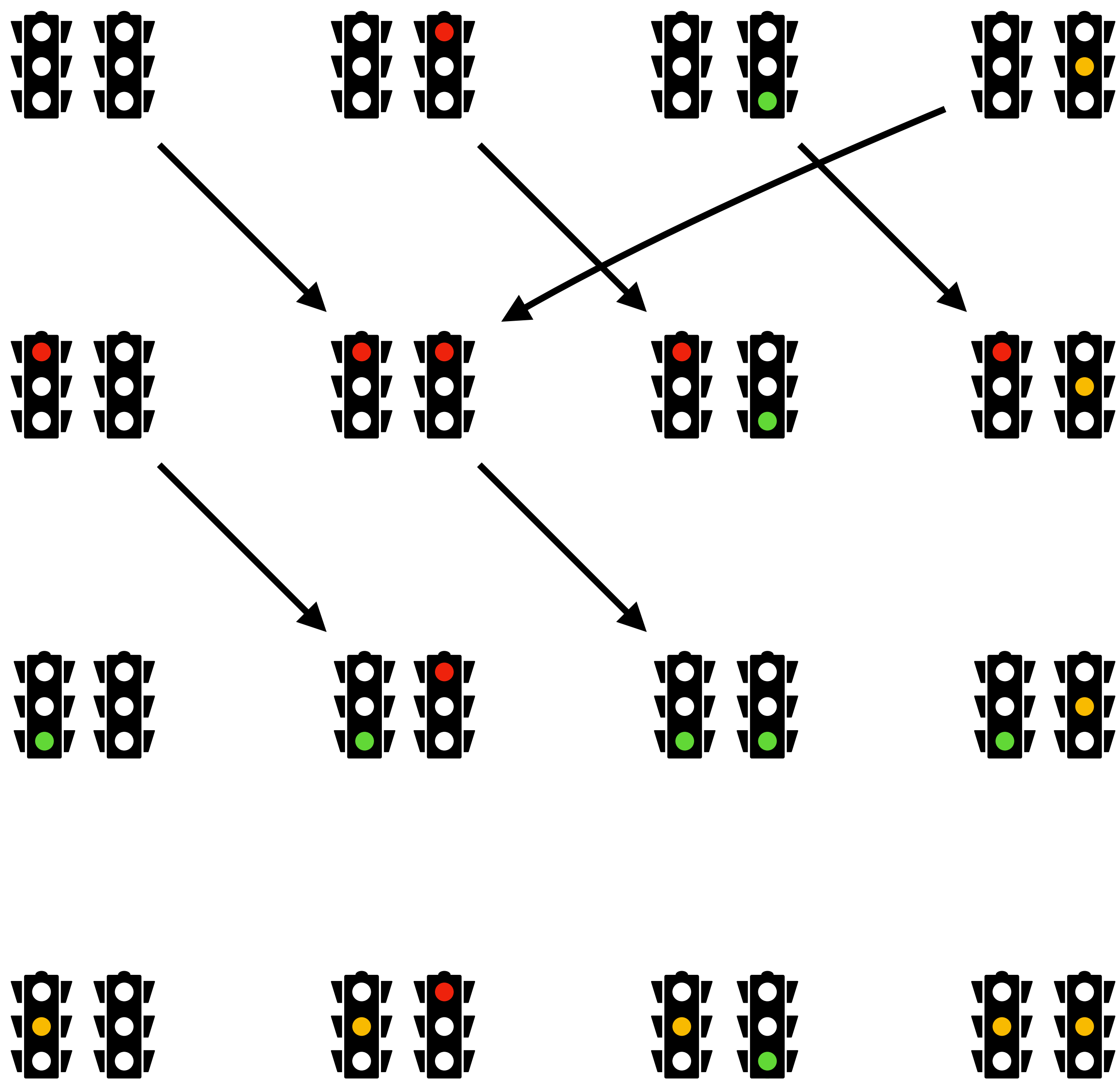
Traffic lights at a crossroads



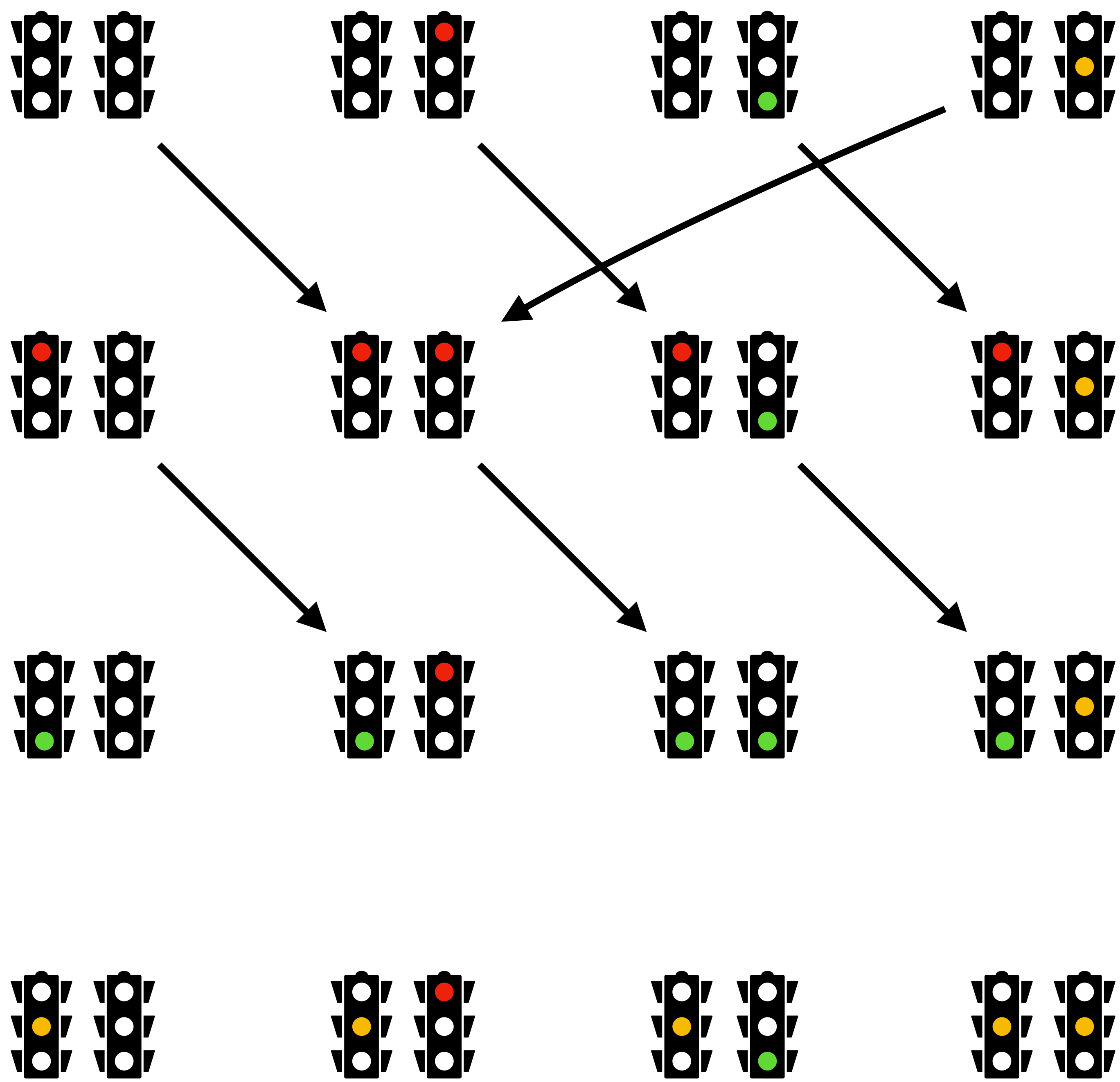
Traffic lights at a crossroads



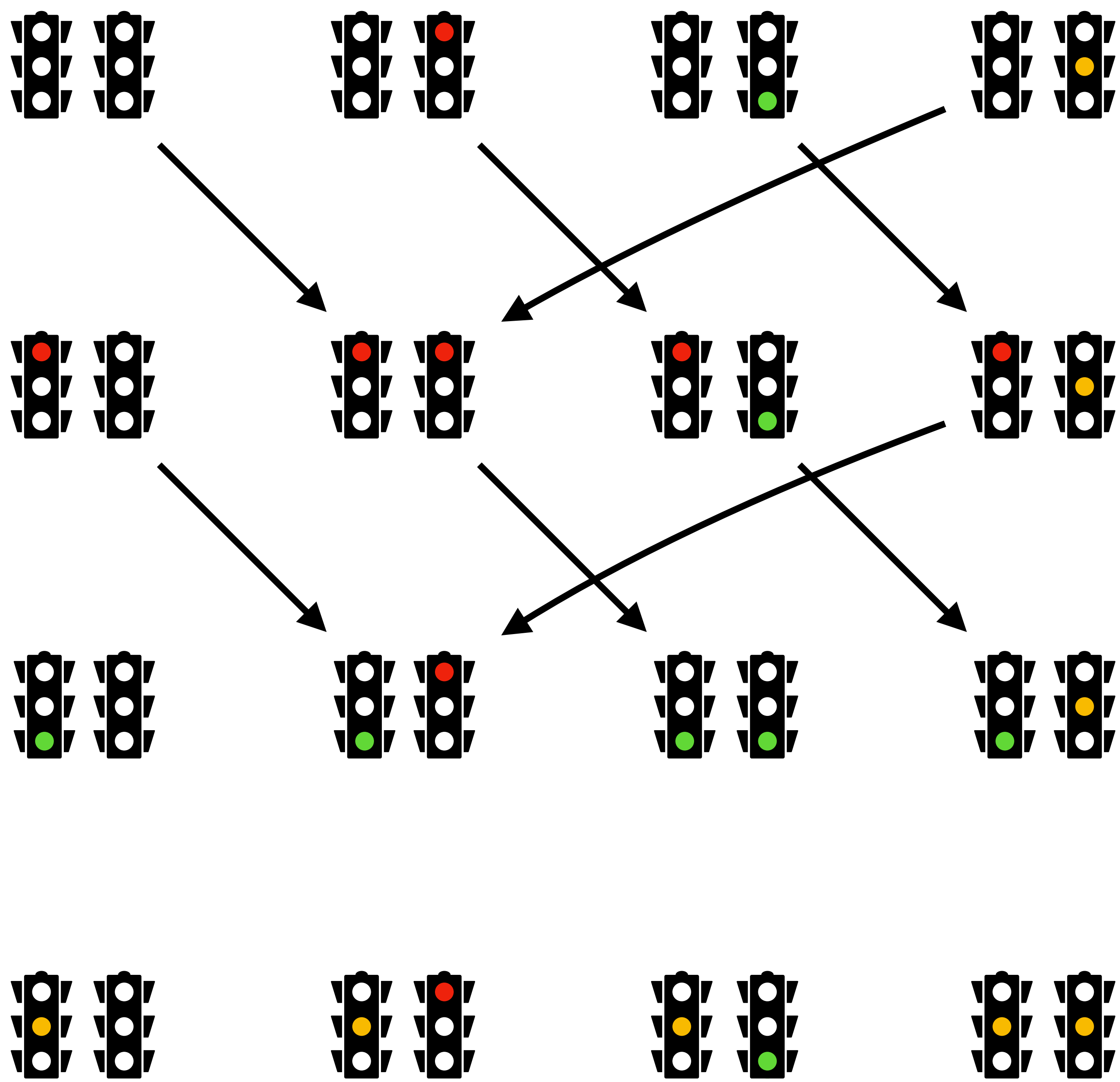
Traffic lights at a crossroads



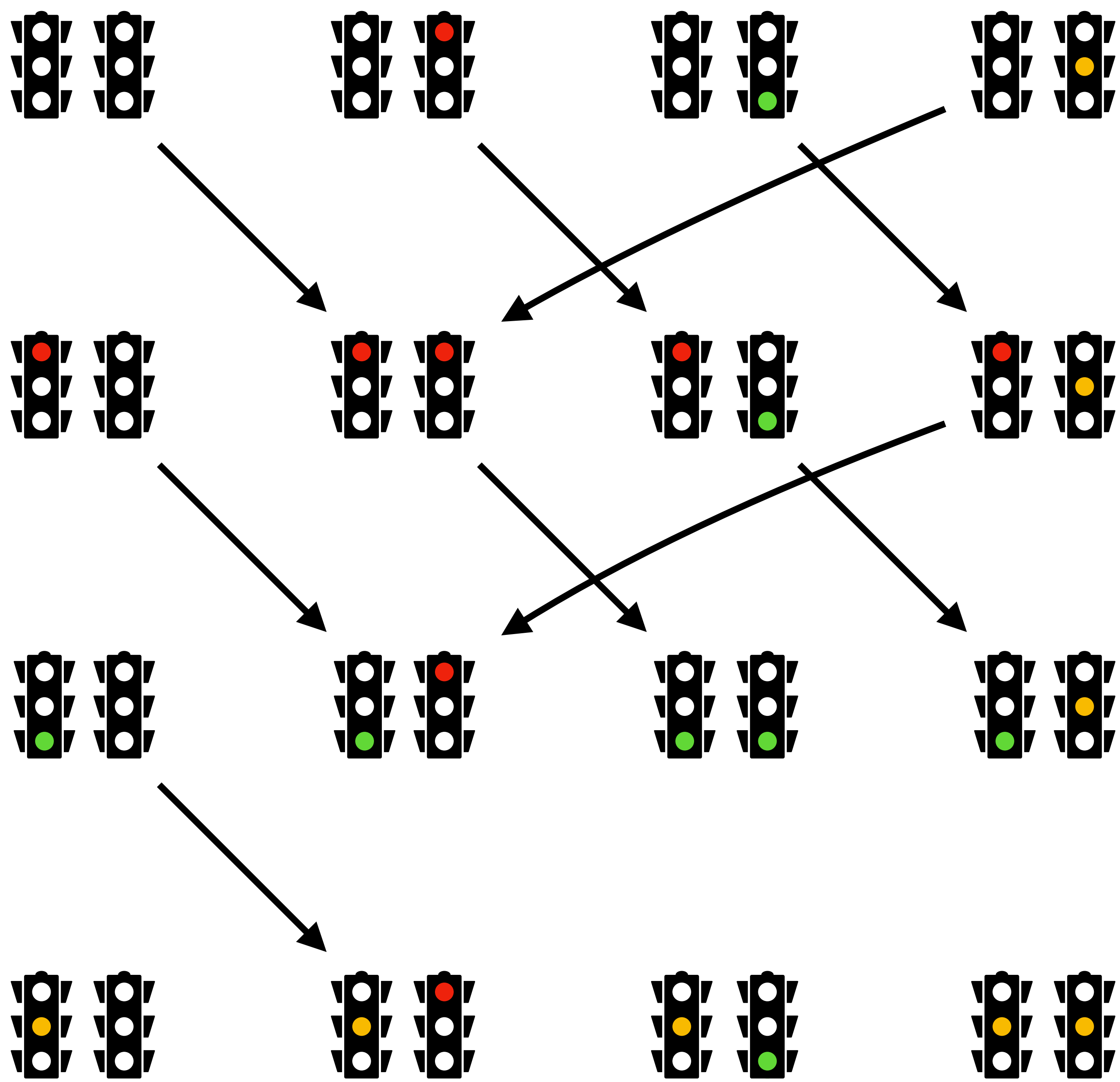
Traffic lights at a crossroads



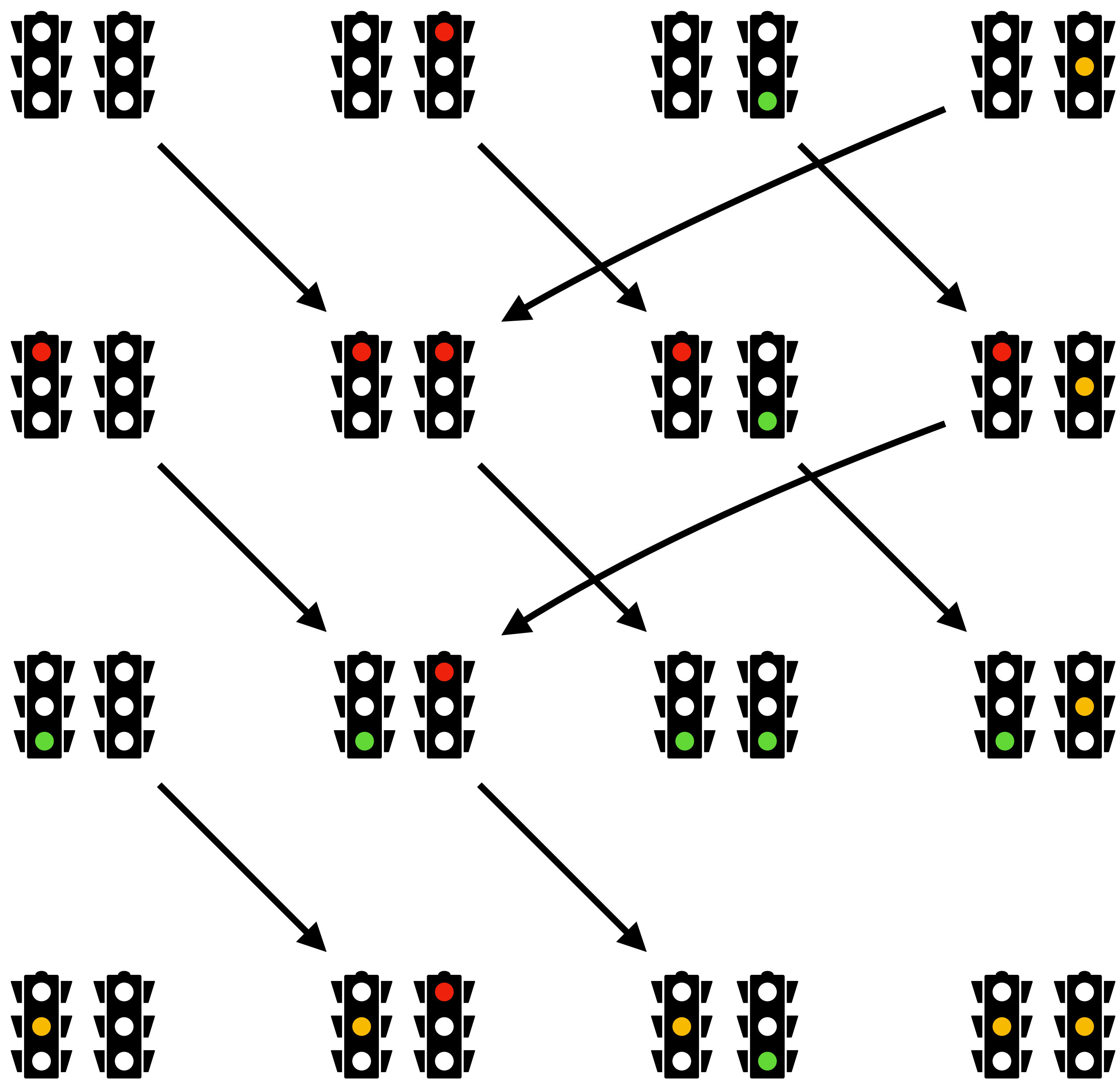
Traffic lights at a crossroads



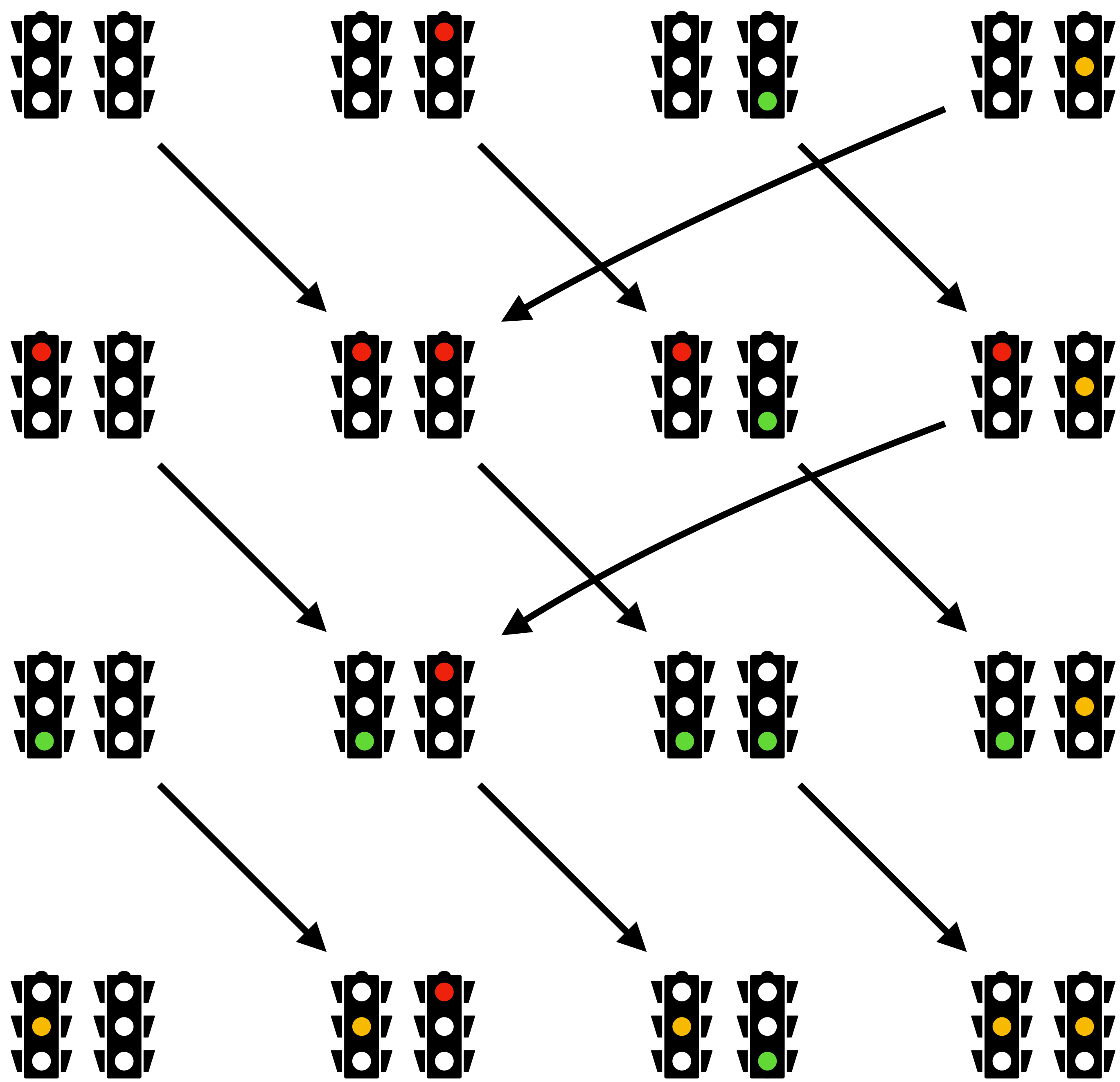
Traffic lights at a crossroads



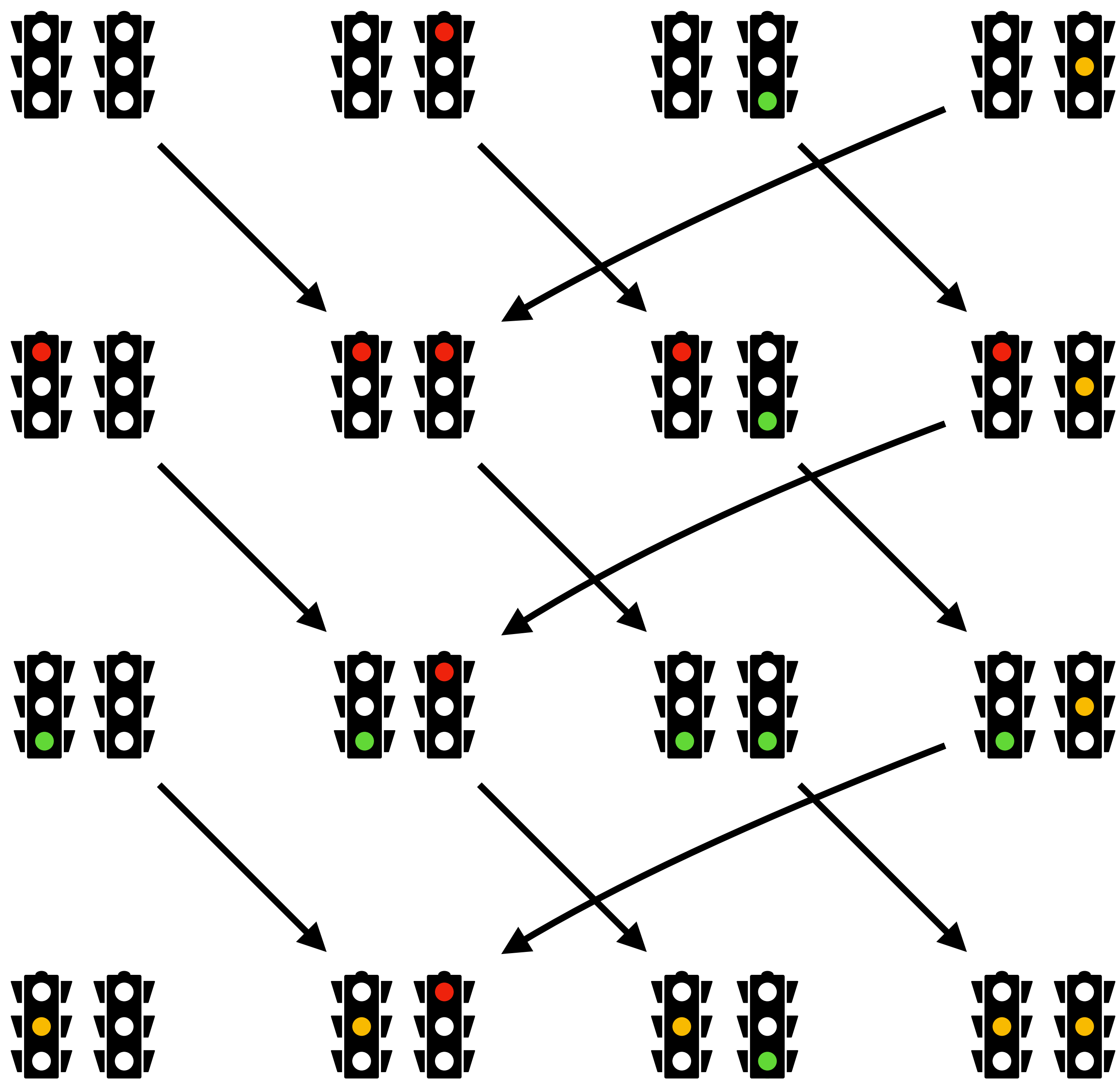
Traffic lights at a crossroads



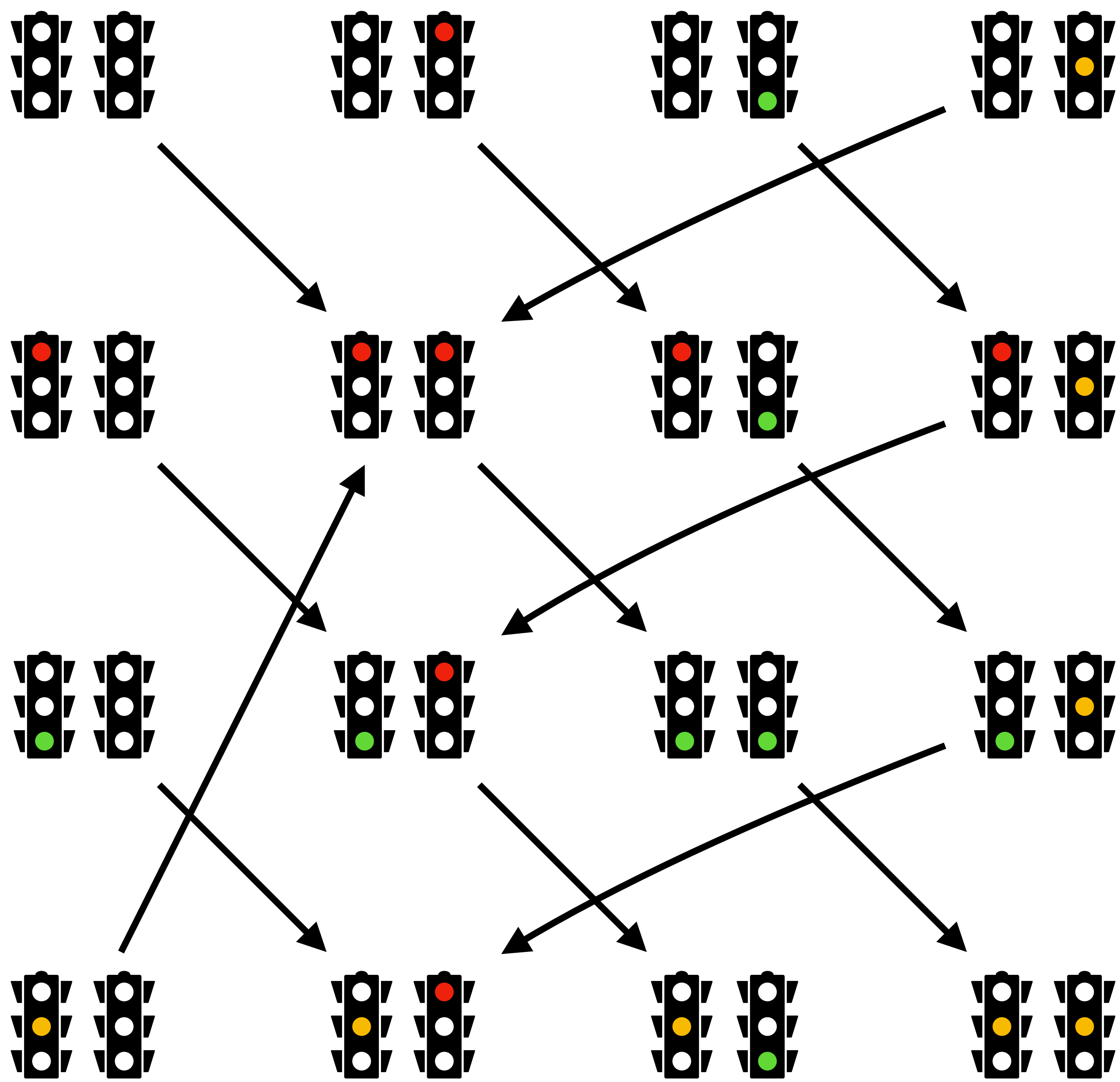
Traffic lights at a crossroads



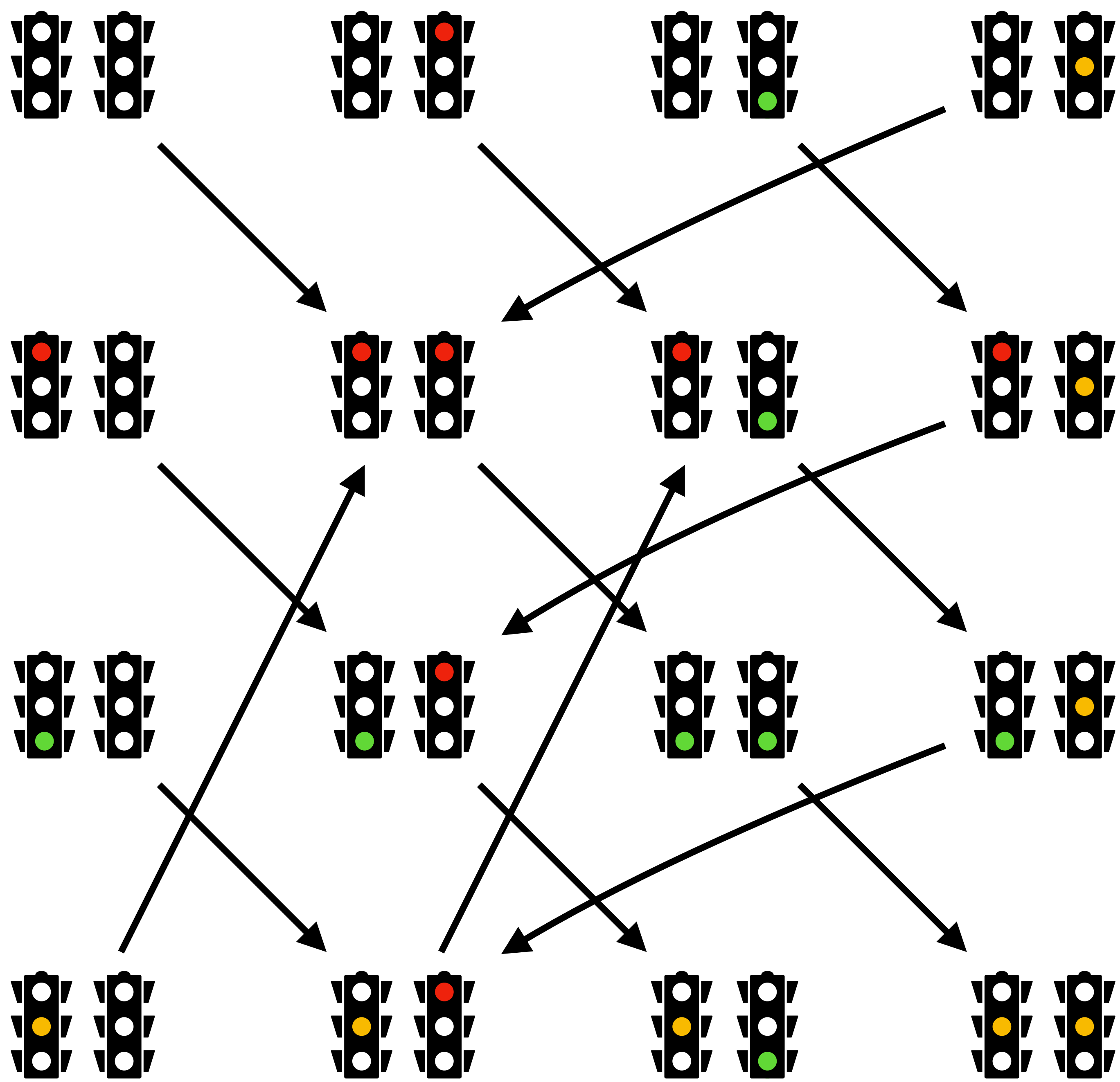
Traffic lights at a crossroads



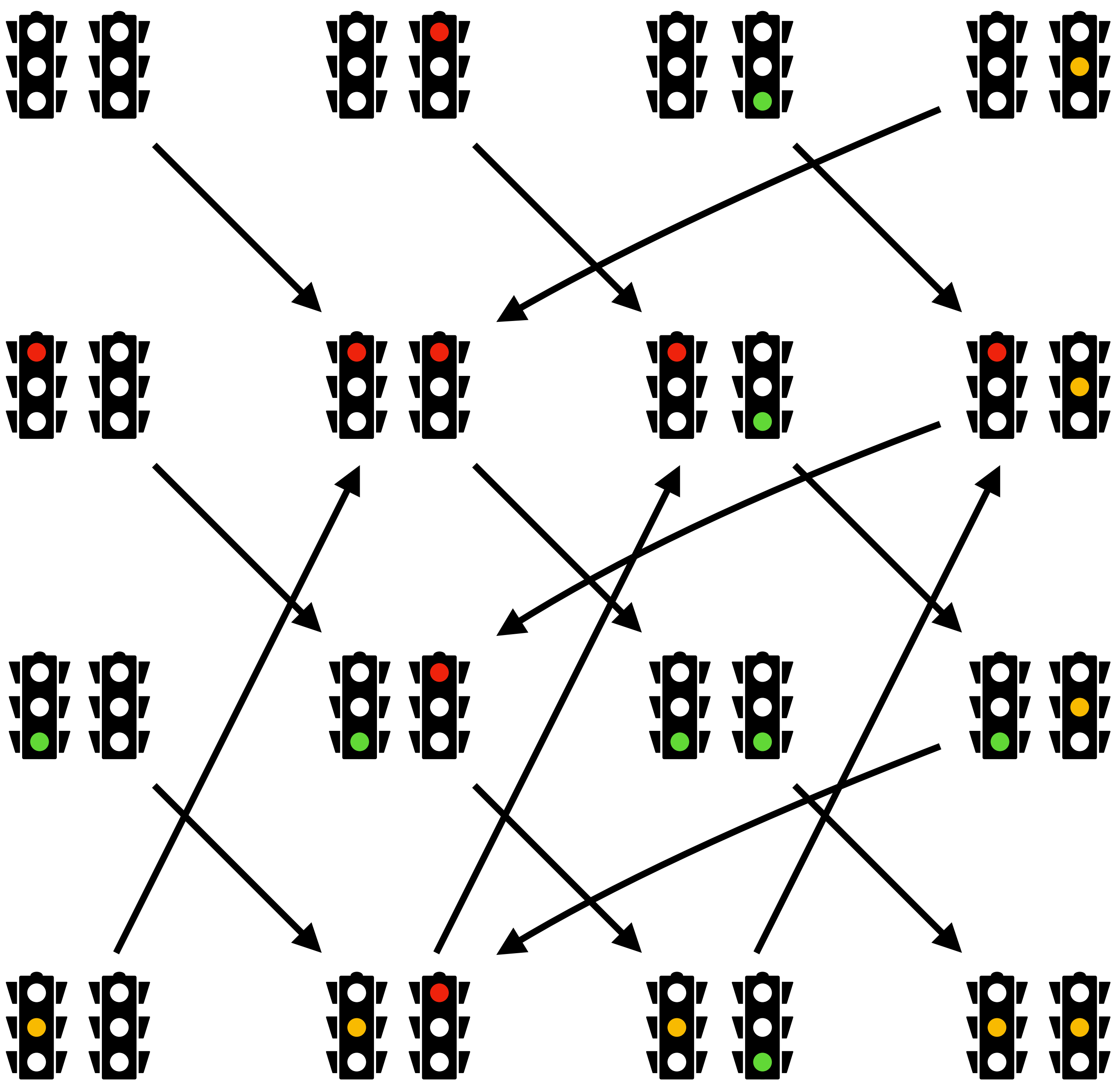
Traffic lights at a crossroads



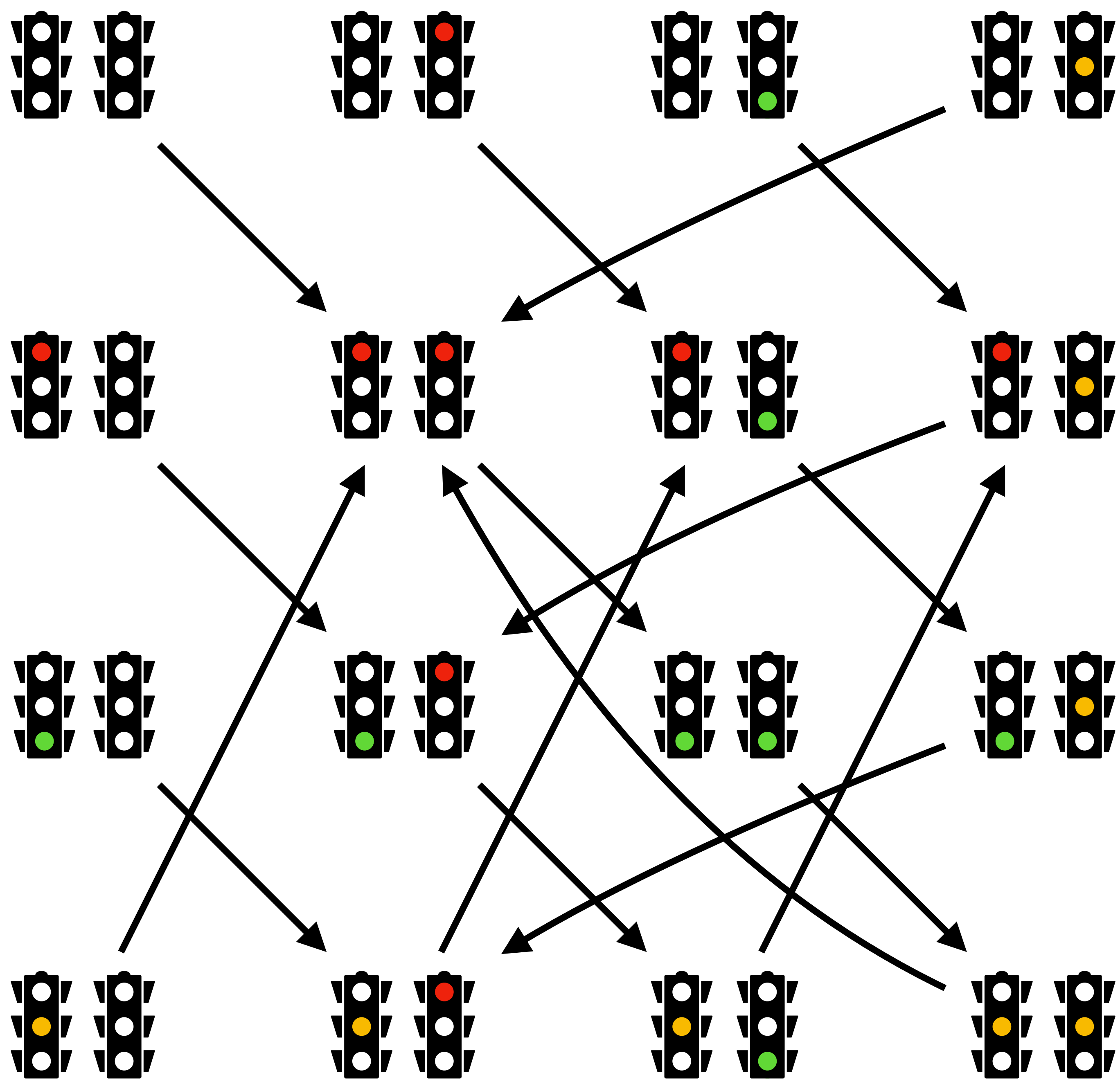
Traffic lights at a crossroads



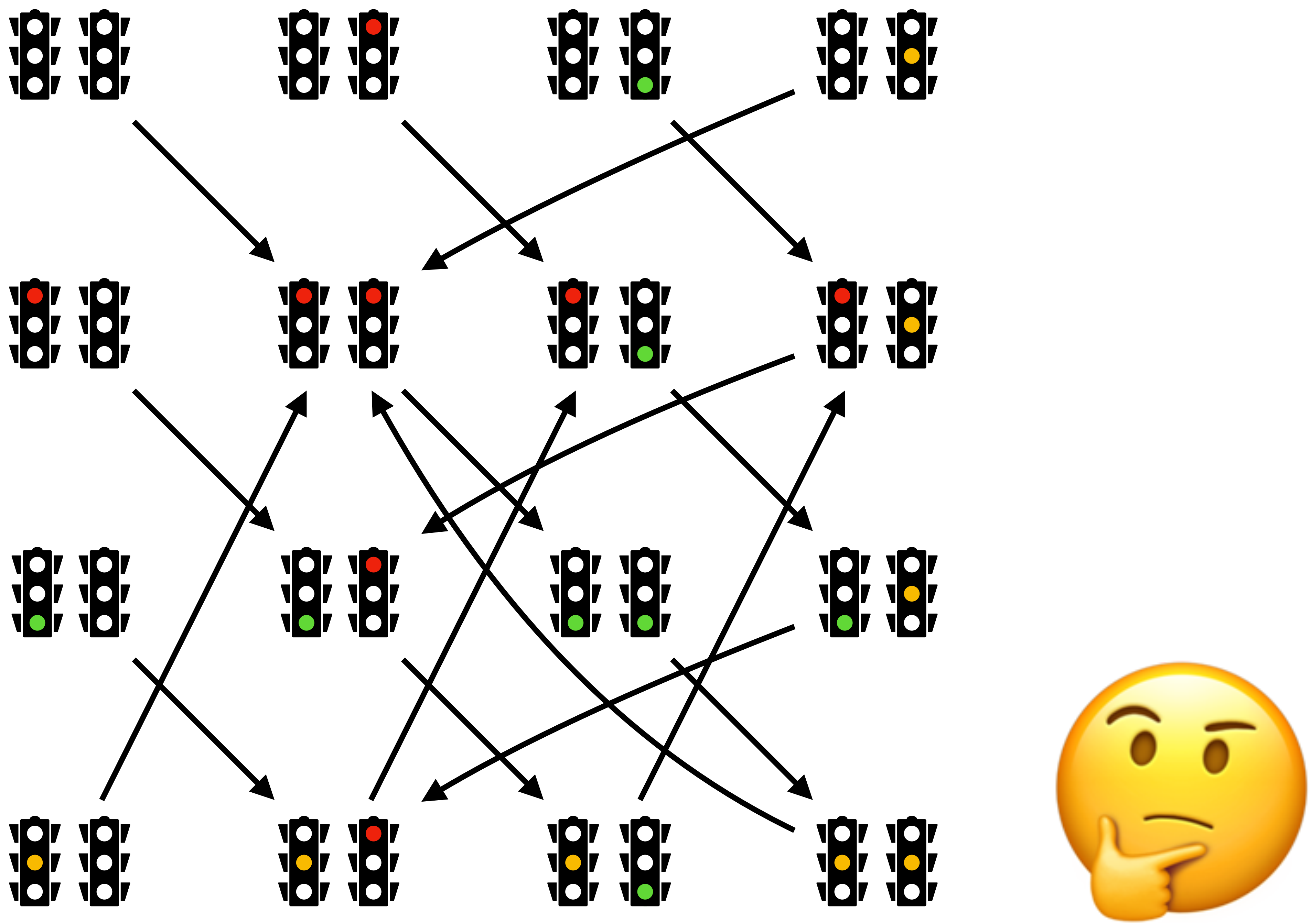
Traffic lights at a crossroads



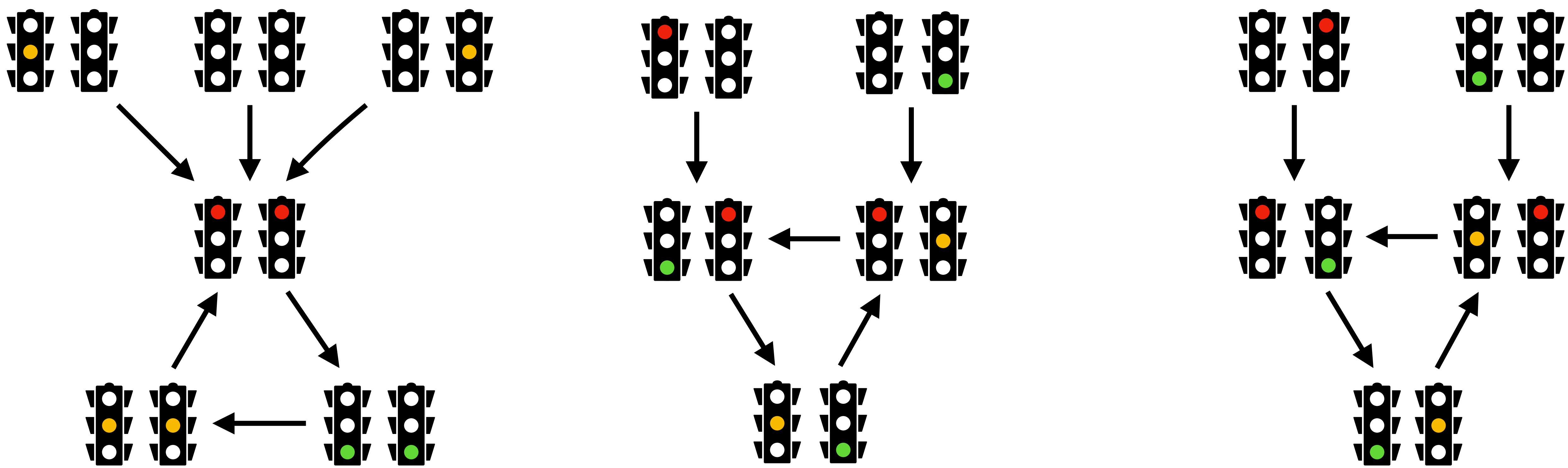
Traffic lights at a crossroads



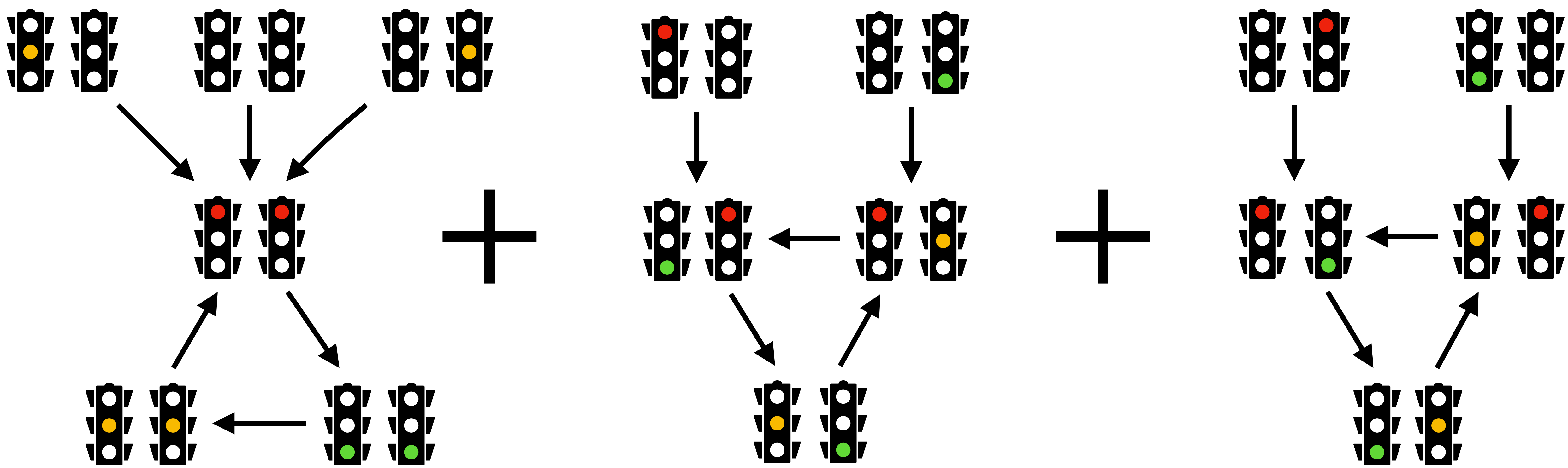
Traffic lights at a crossroads



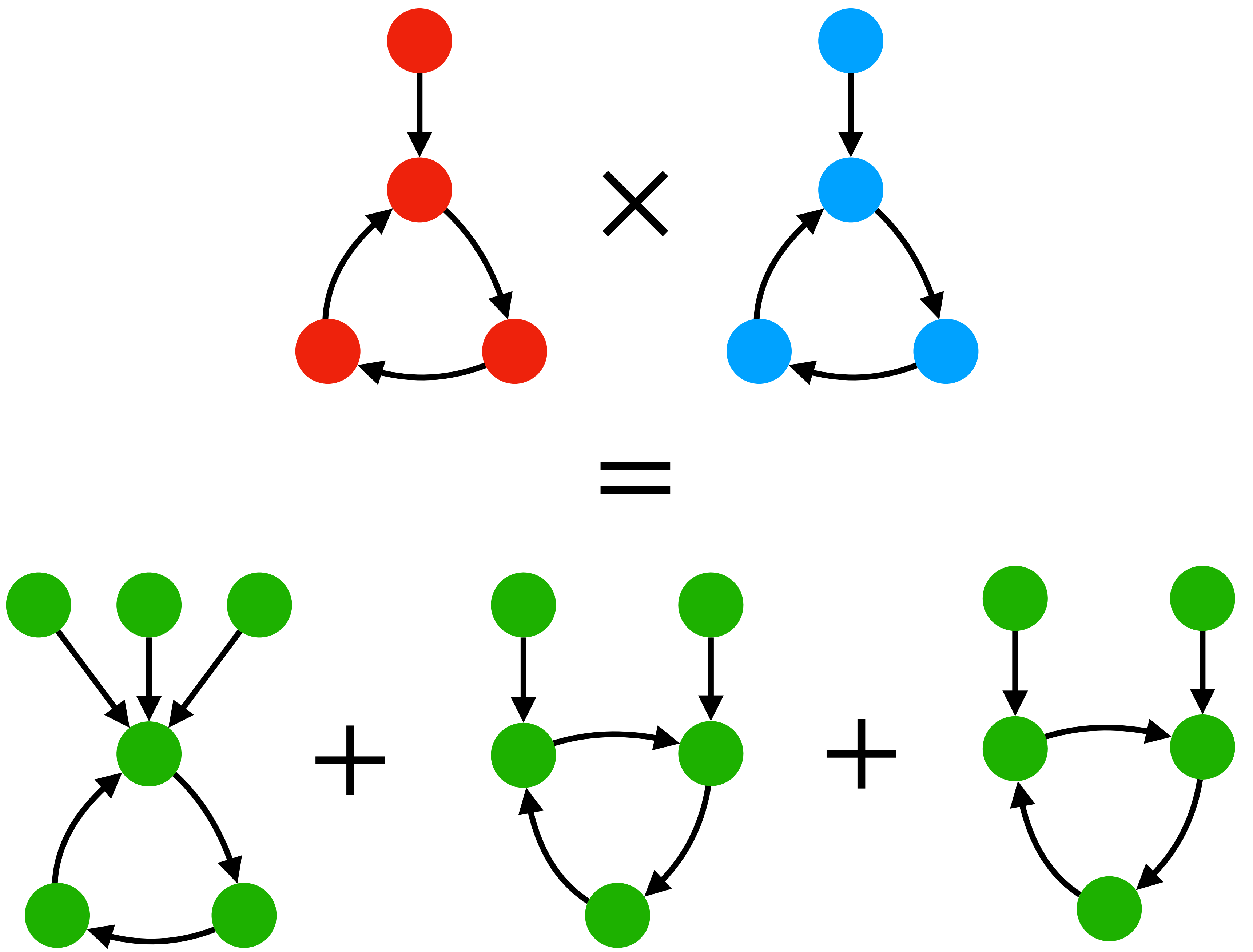
Traffic lights at a crossroads



Traffic lights at a crossroads



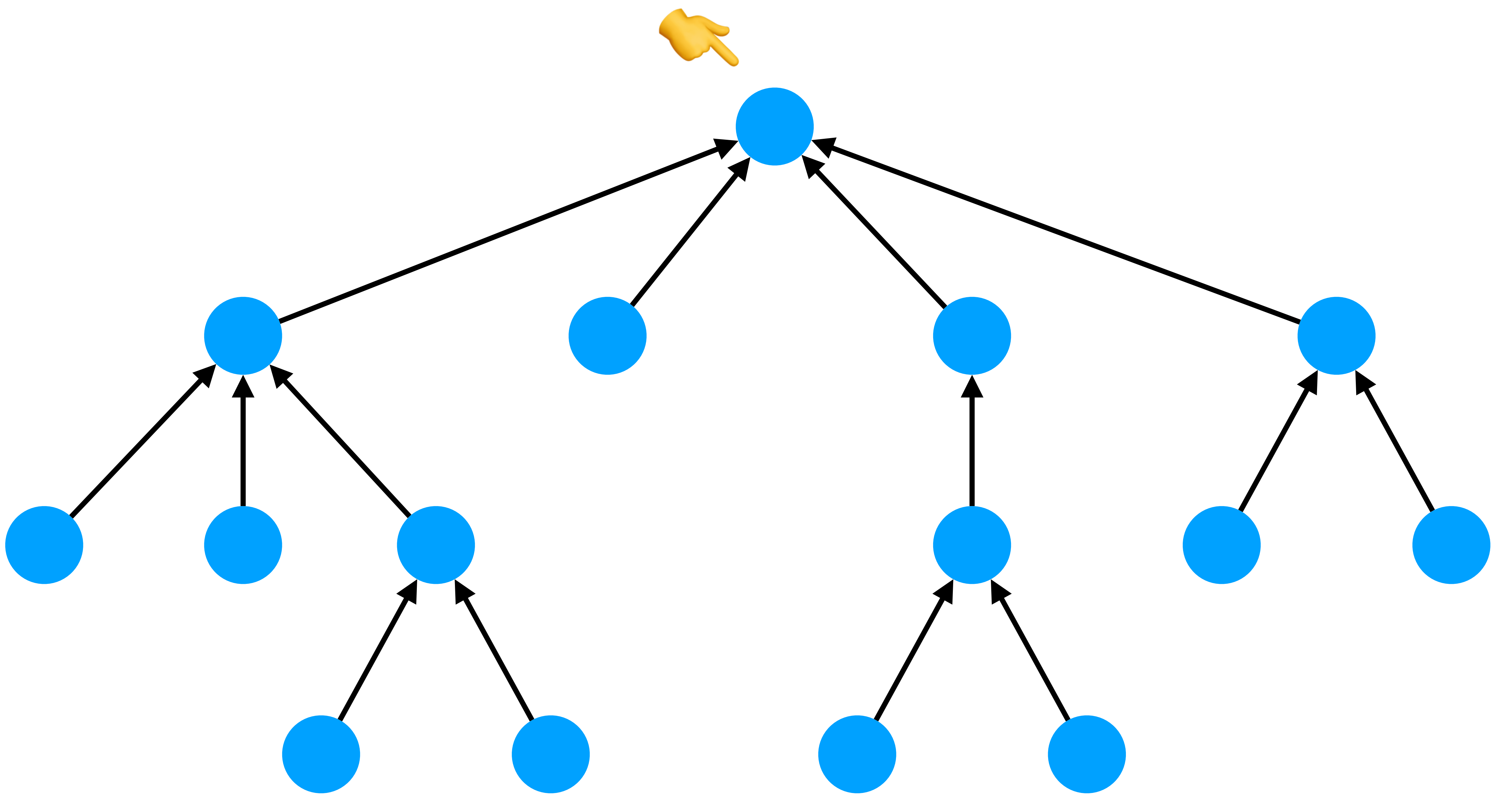
More abstractly…



Isomorphism
of dynamical systems
in polynomial time

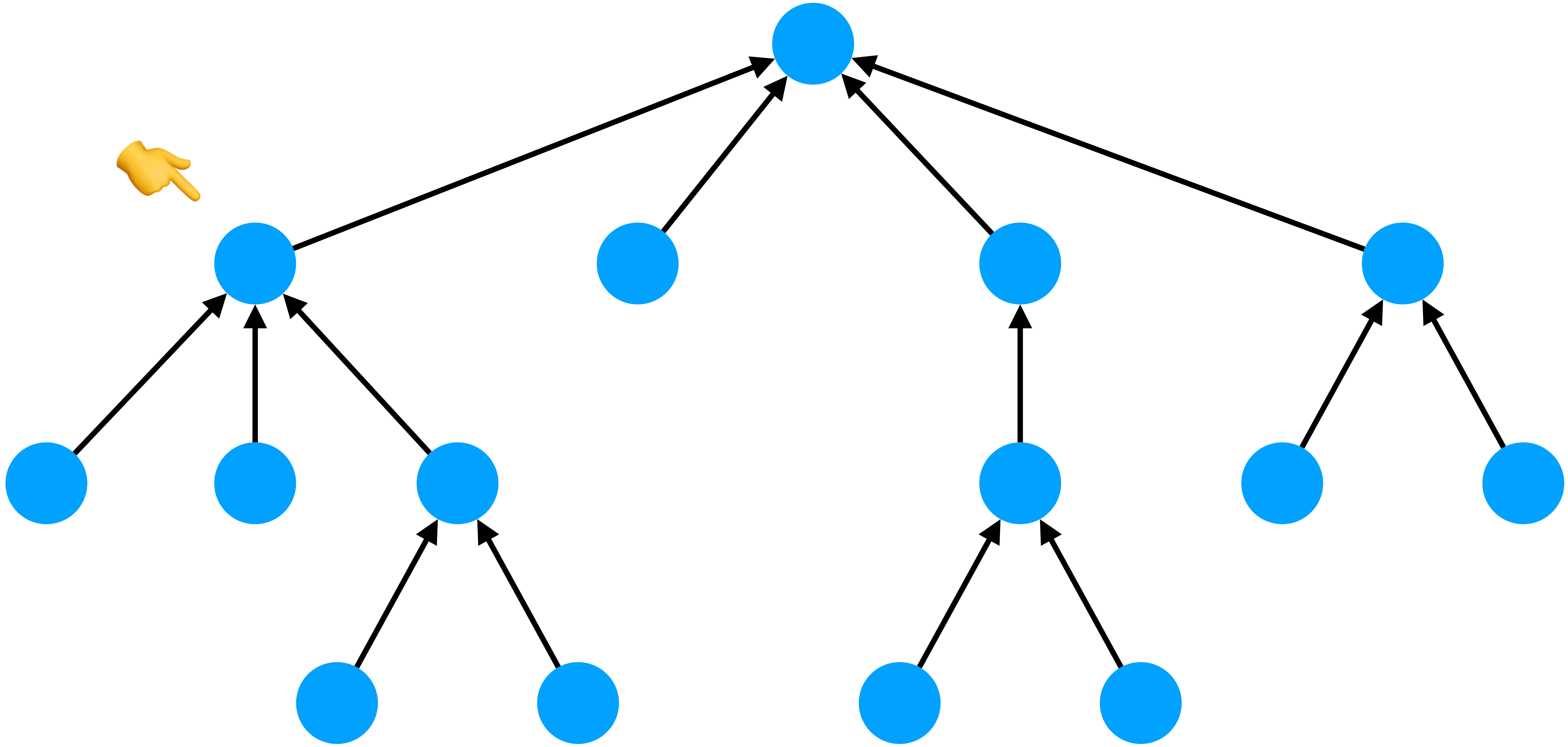
Tree canonisation

A polynomial-time algorithm



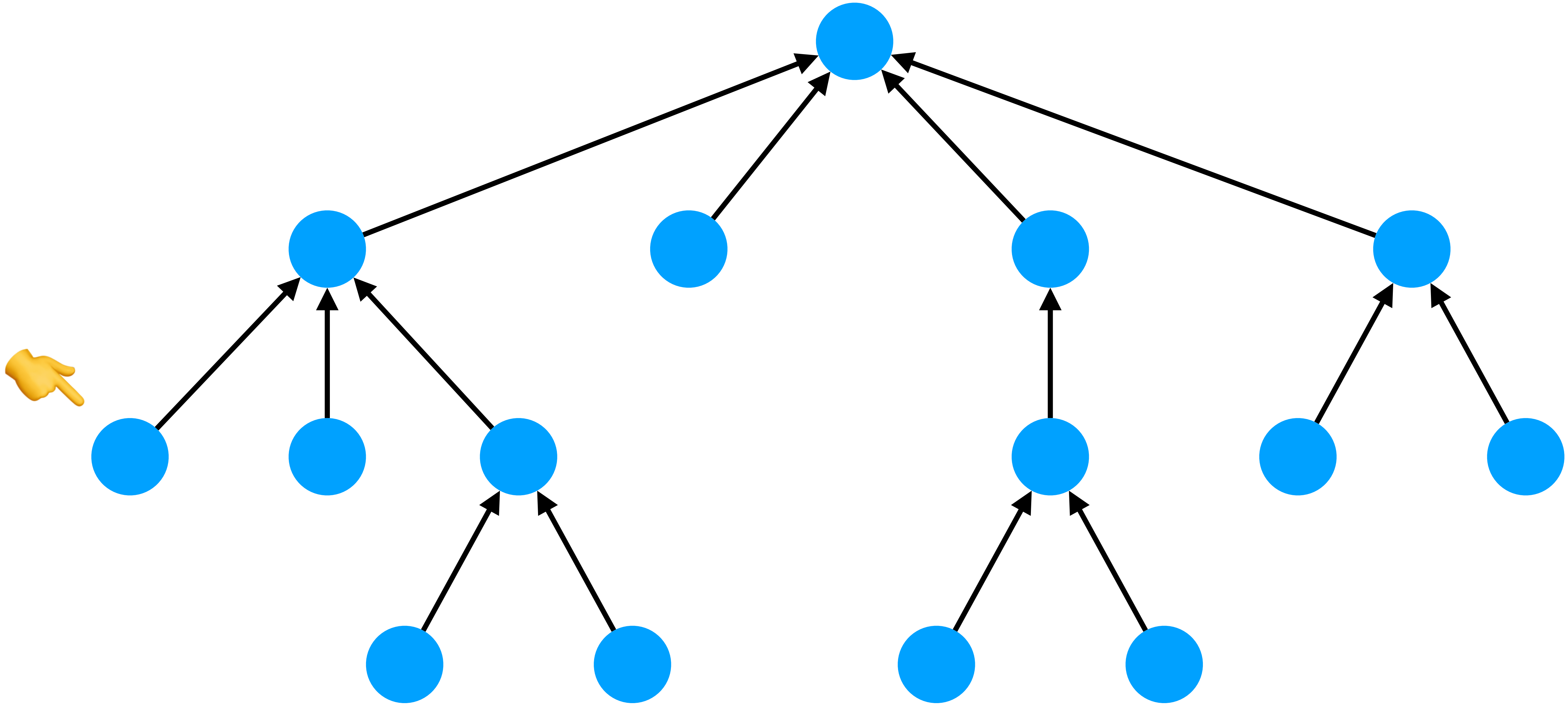
Tree canonisation

A polynomial-time algorithm



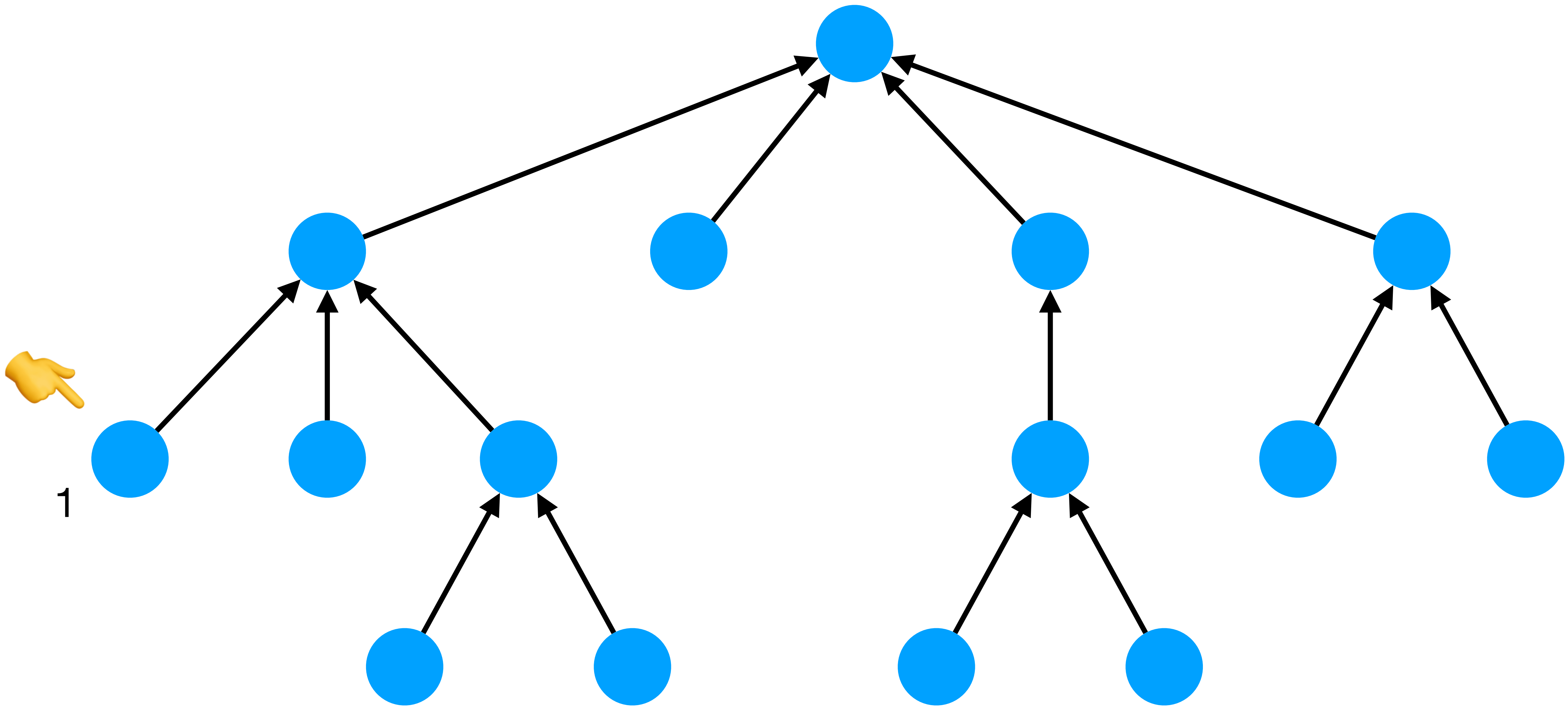
Tree canonisation

A polynomial-time algorithm



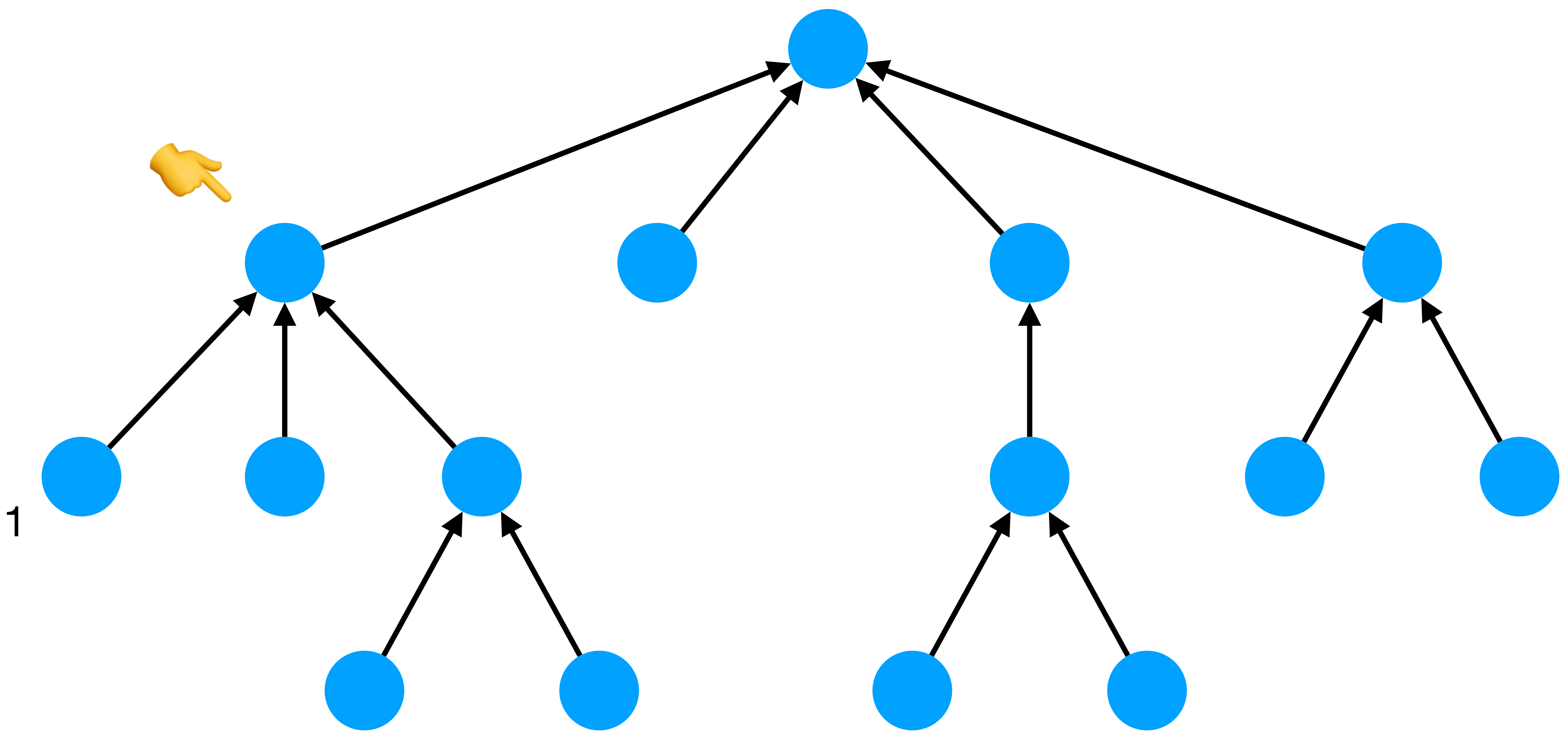
Tree canonisation

A polynomial-time algorithm



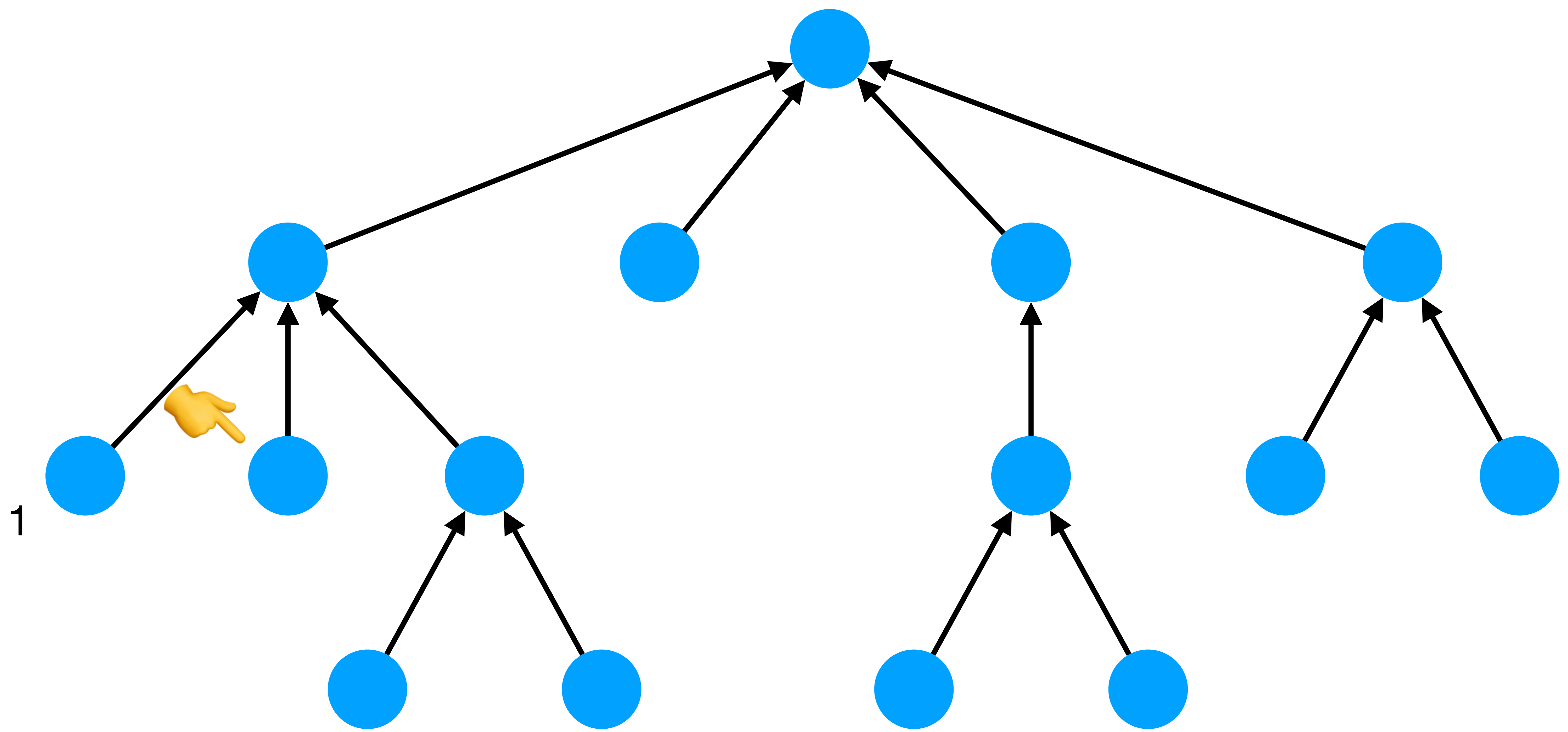
Tree canonisation

A polynomial-time algorithm



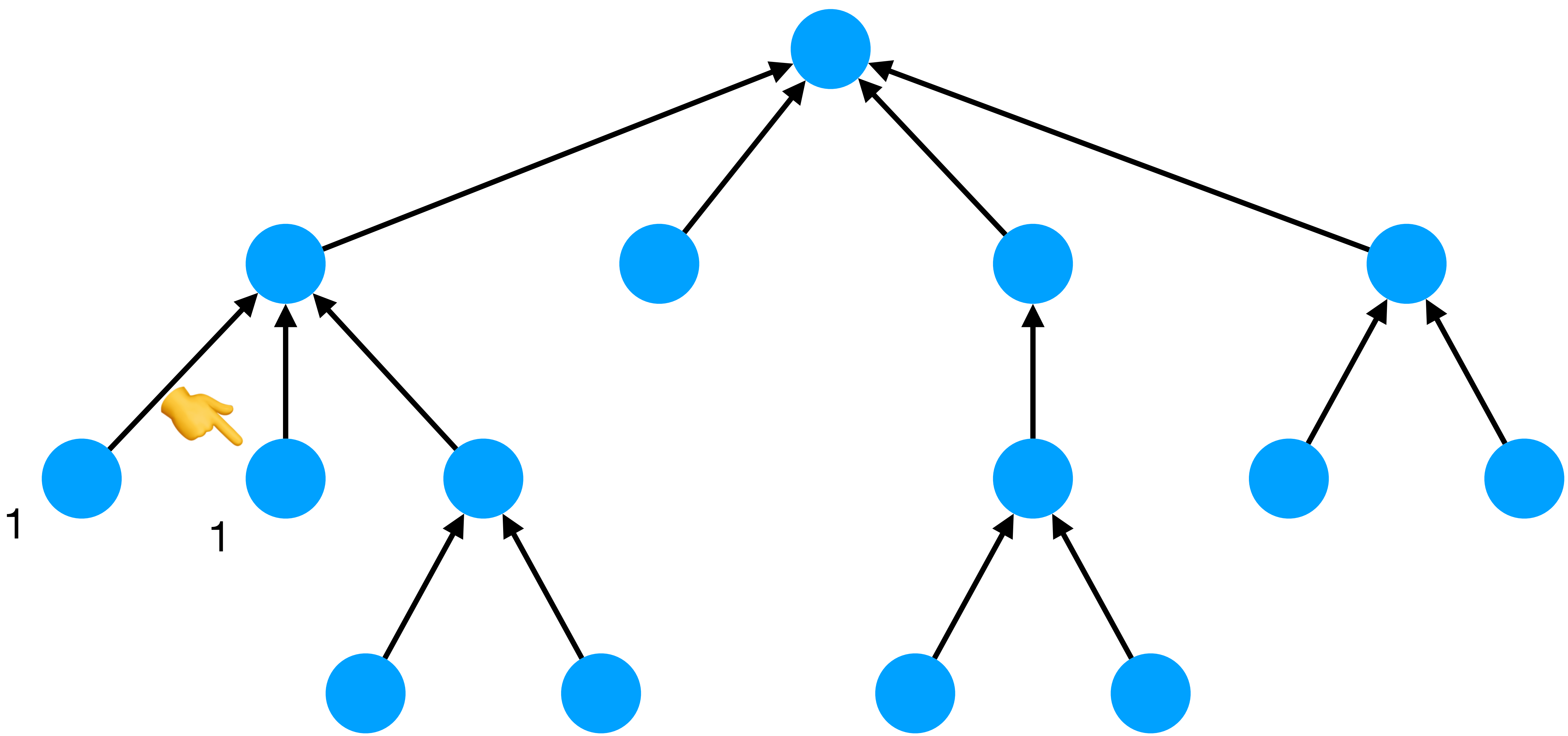
Tree canonisation

A polynomial-time algorithm



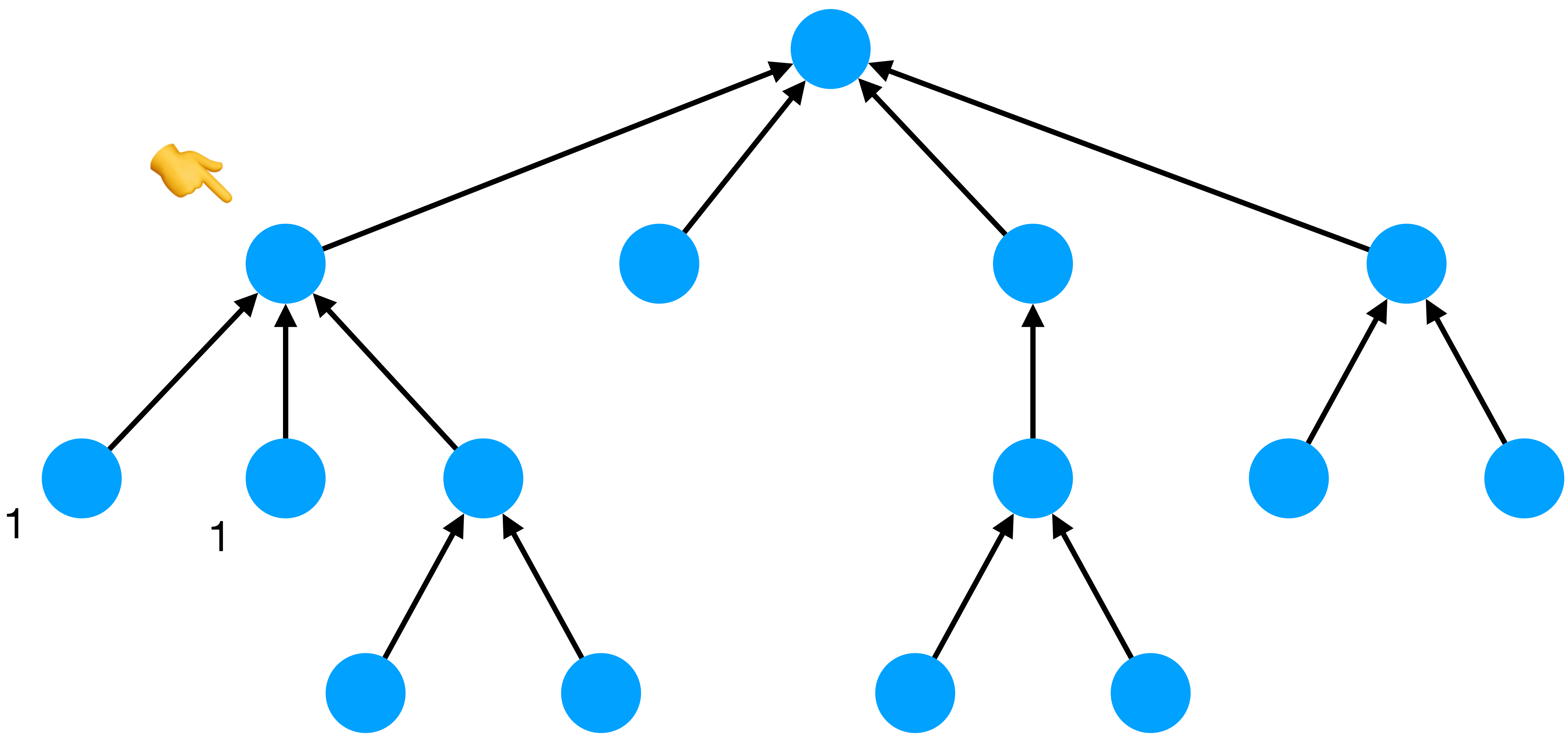
Tree canonisation

A polynomial-time algorithm



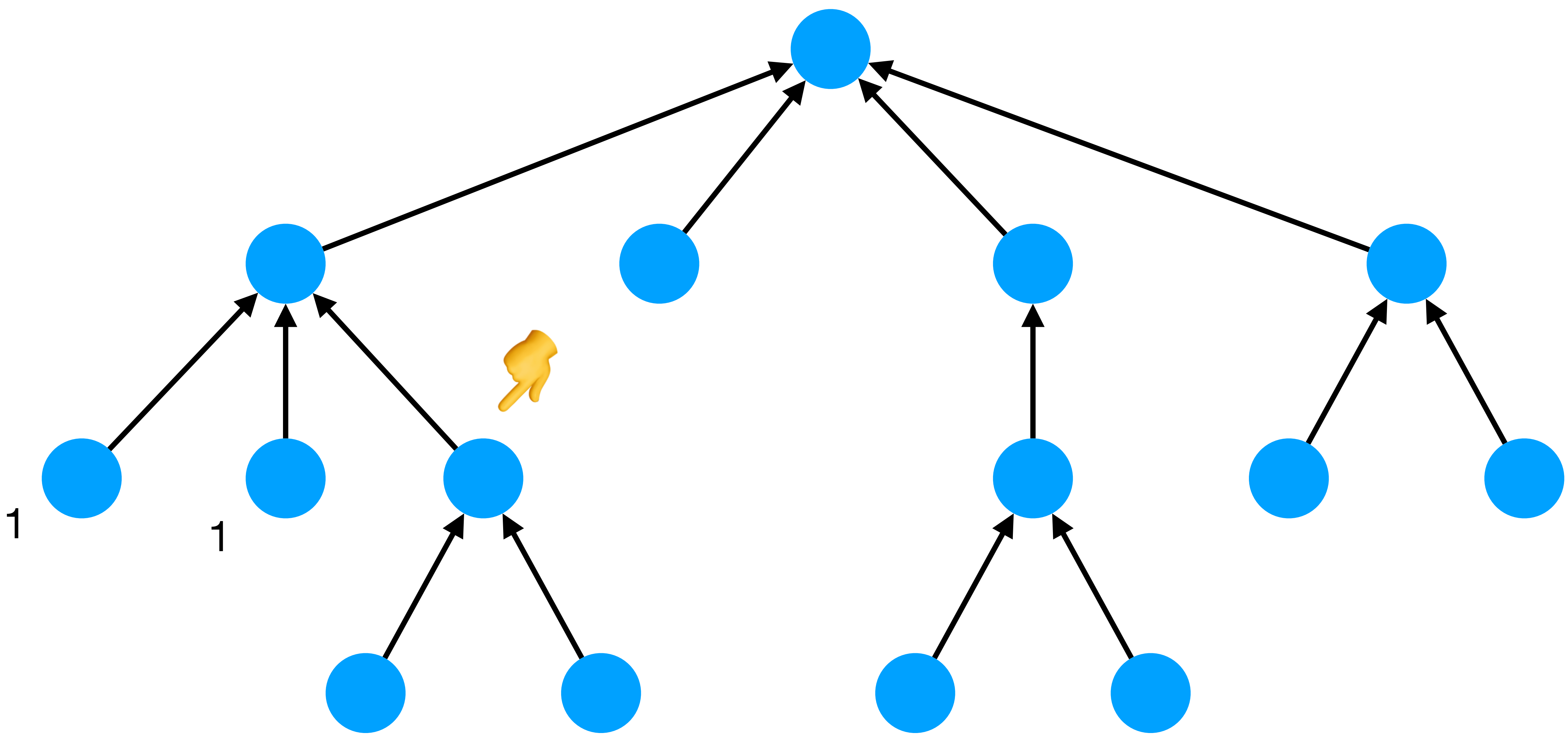
Tree canonisation

A polynomial-time algorithm



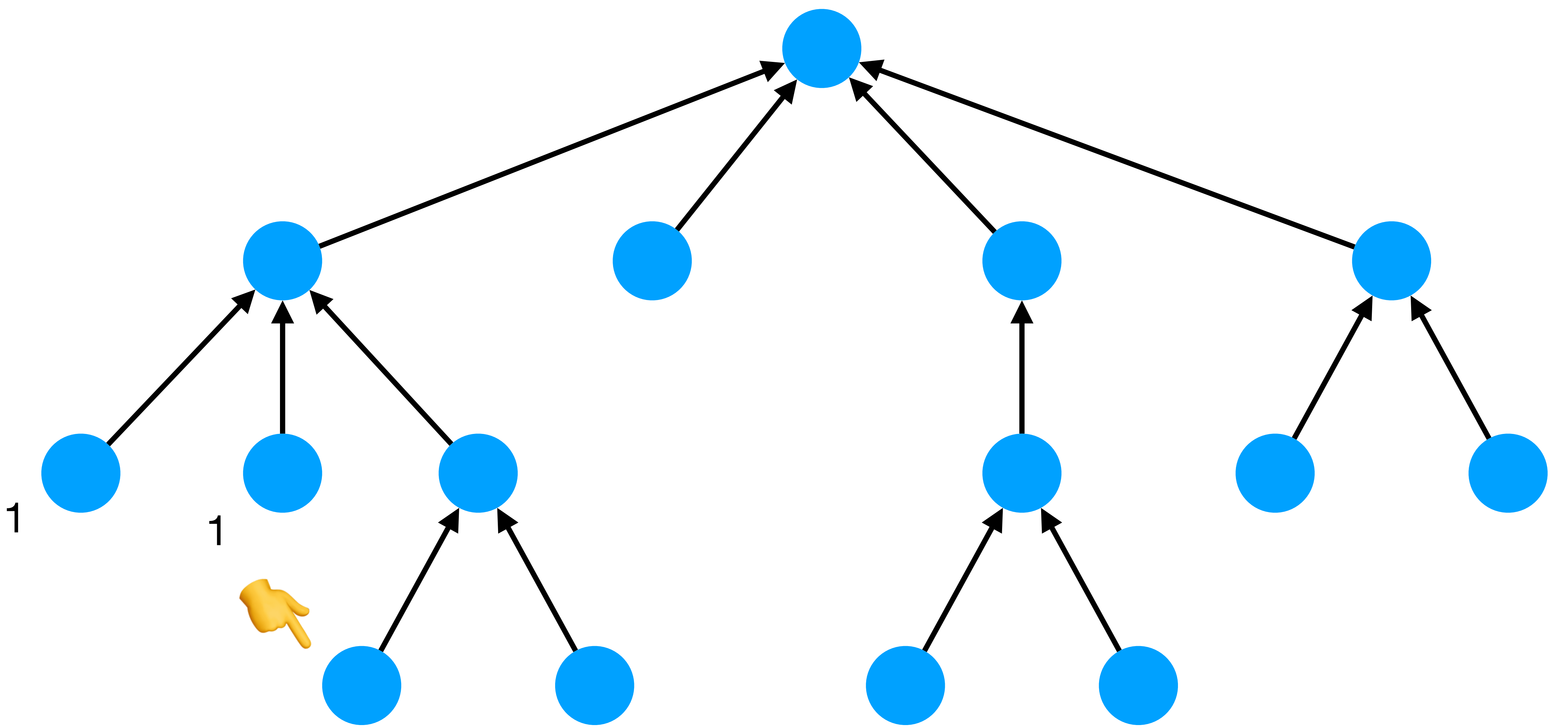
Tree canonisation

A polynomial-time algorithm



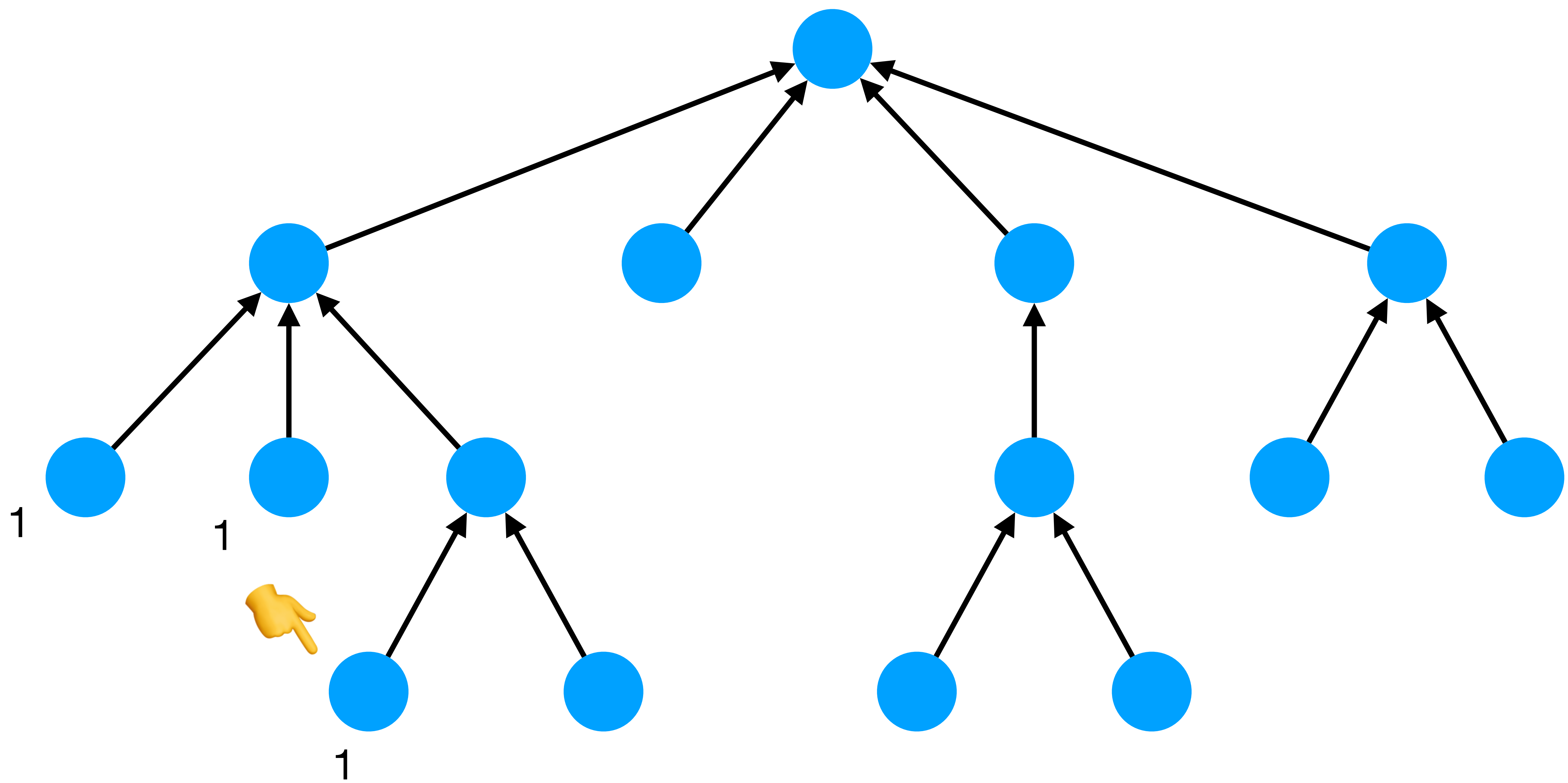
Tree canonisation

A polynomial-time algorithm



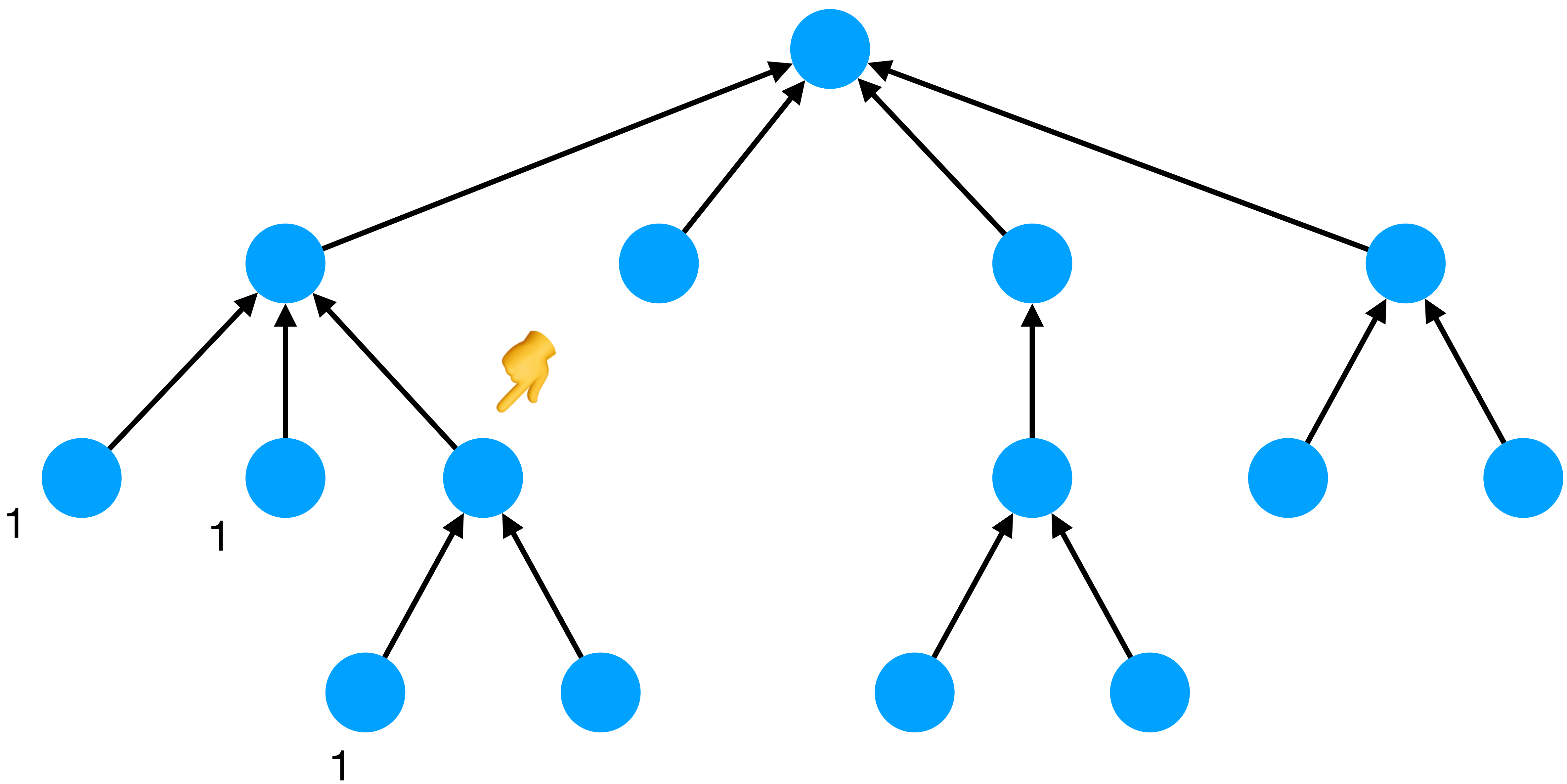
Tree canonisation

A polynomial-time algorithm



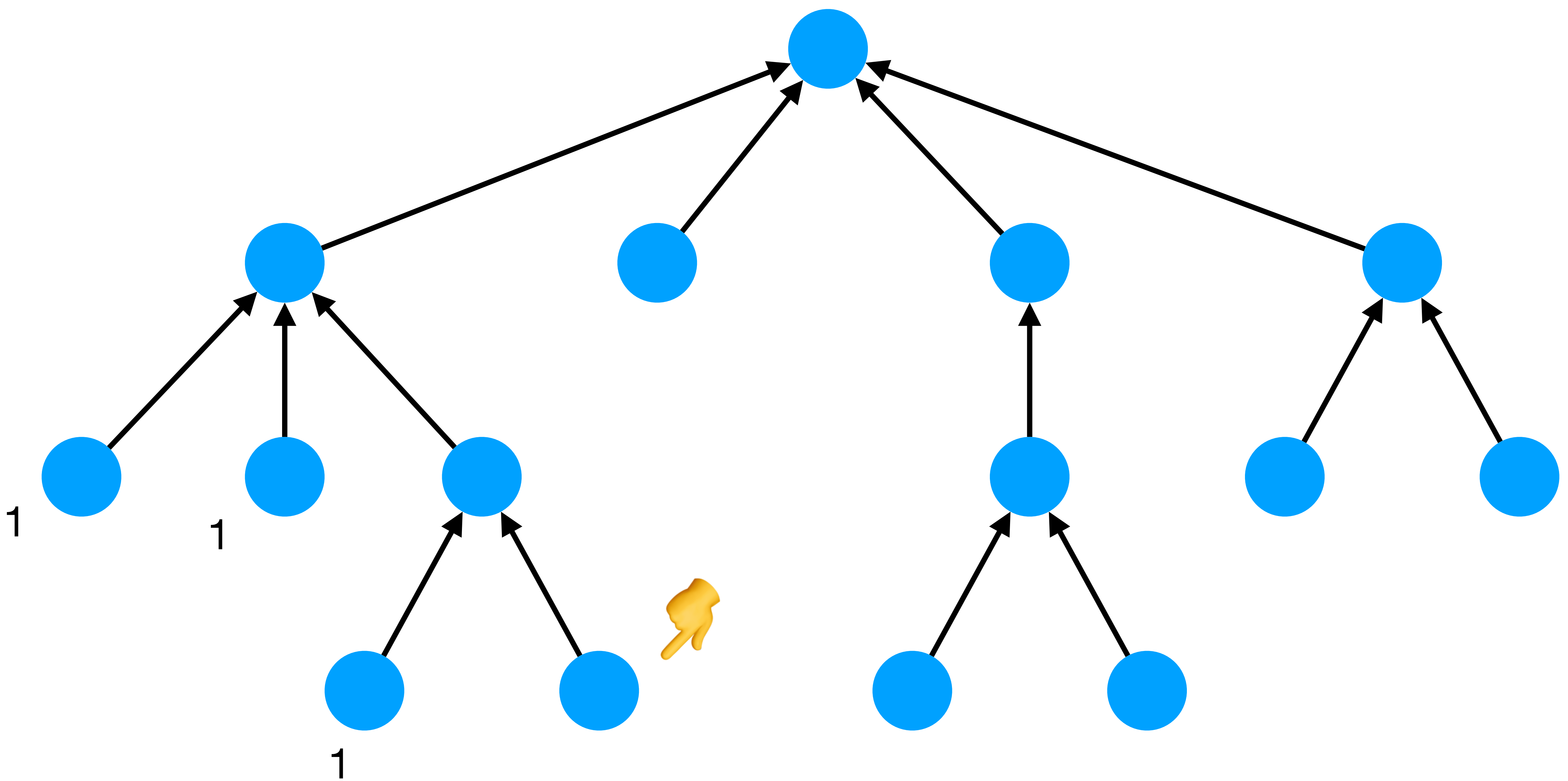
Tree canonisation

A polynomial-time algorithm



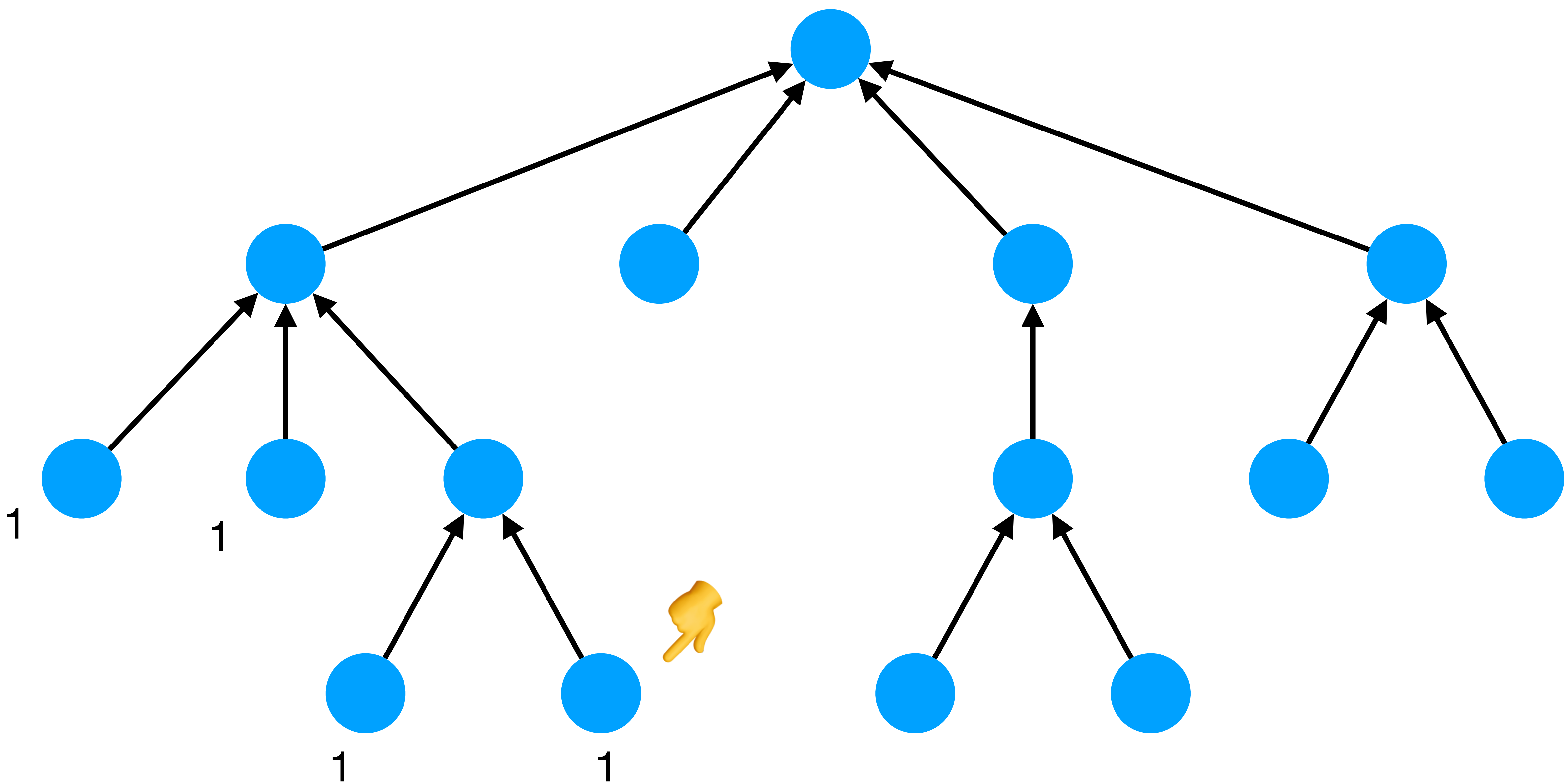
Tree canonisation

A polynomial-time algorithm



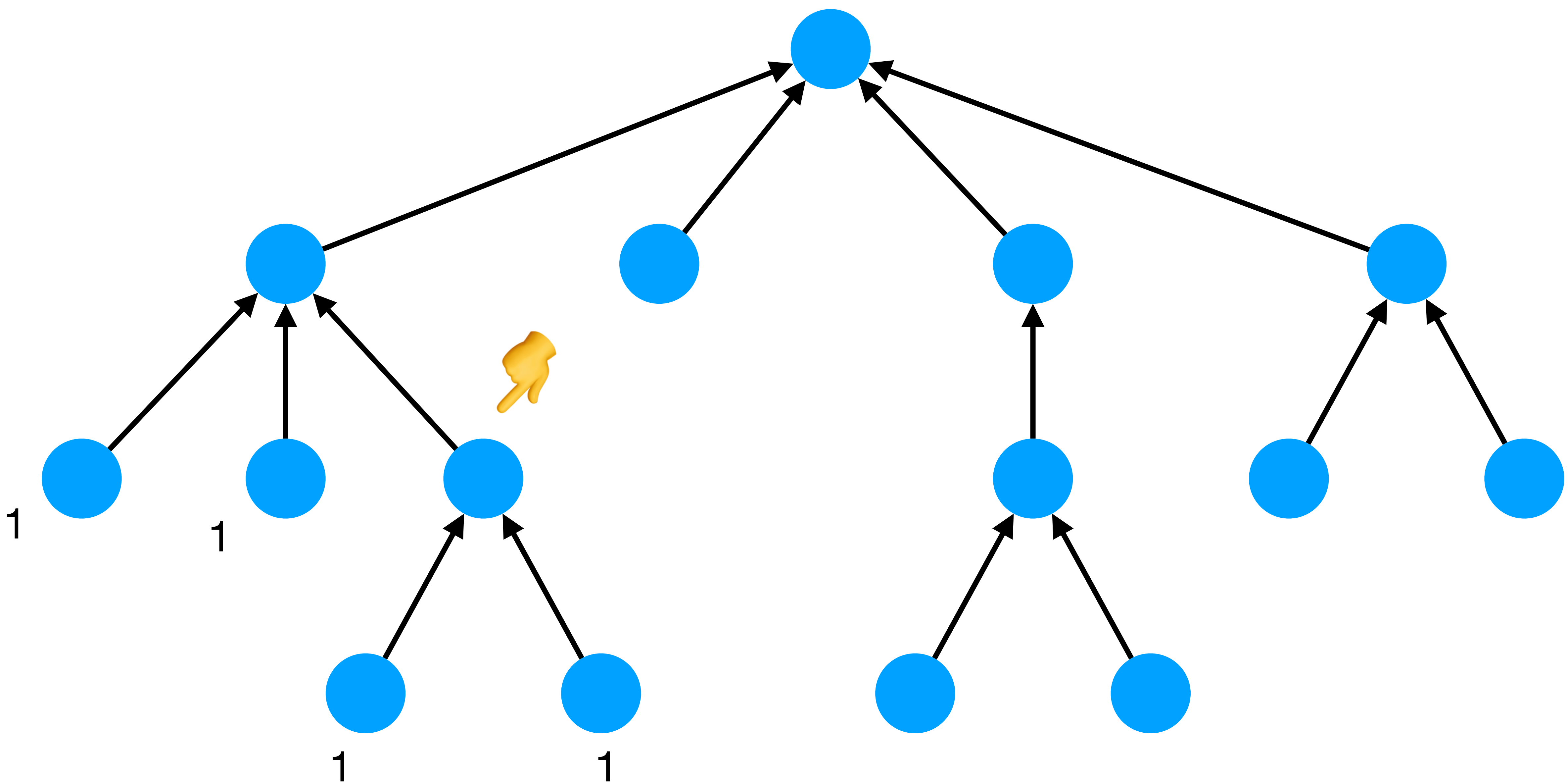
Tree canonisation

A polynomial-time algorithm



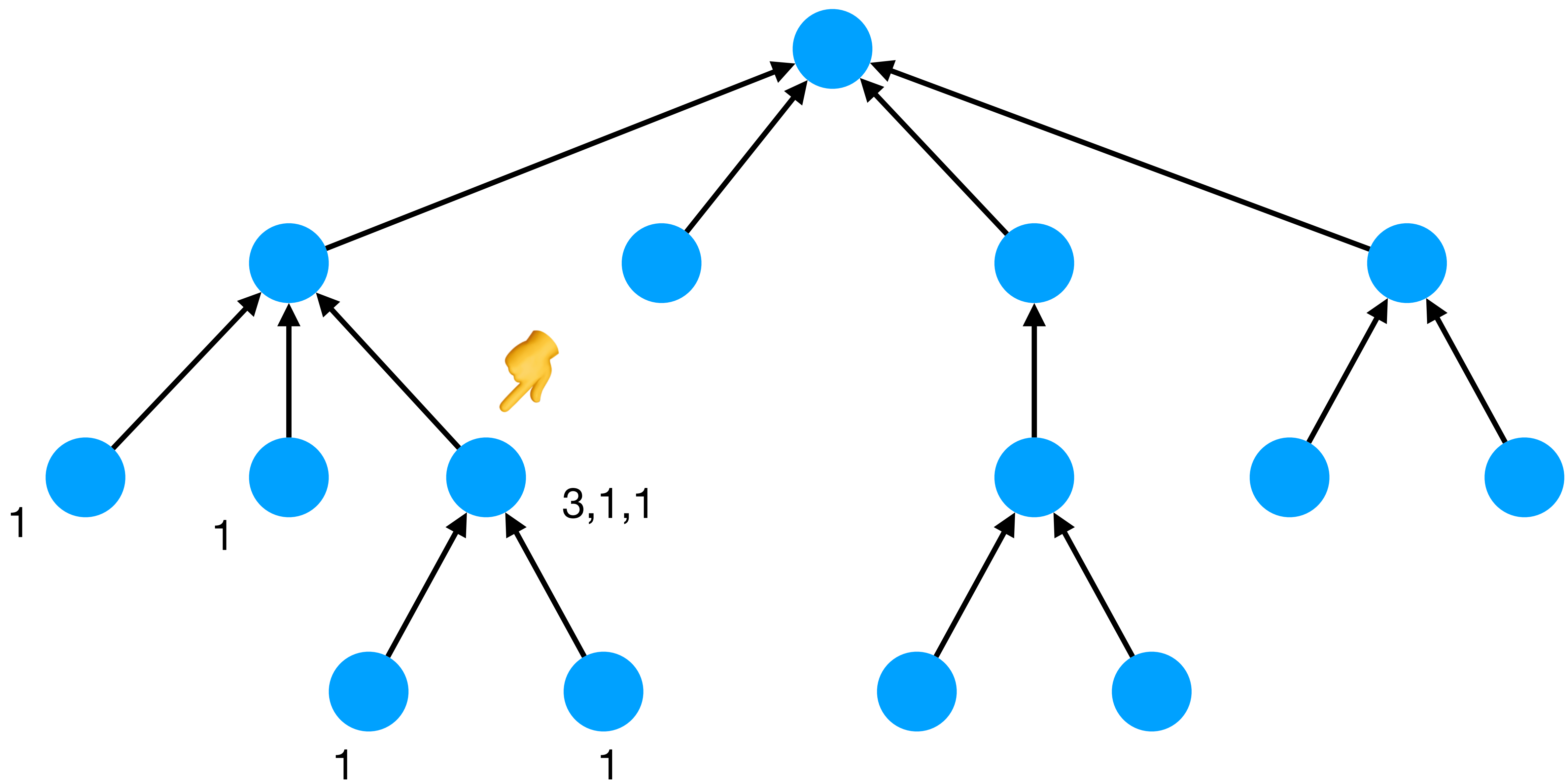
Tree canonisation

A polynomial-time algorithm



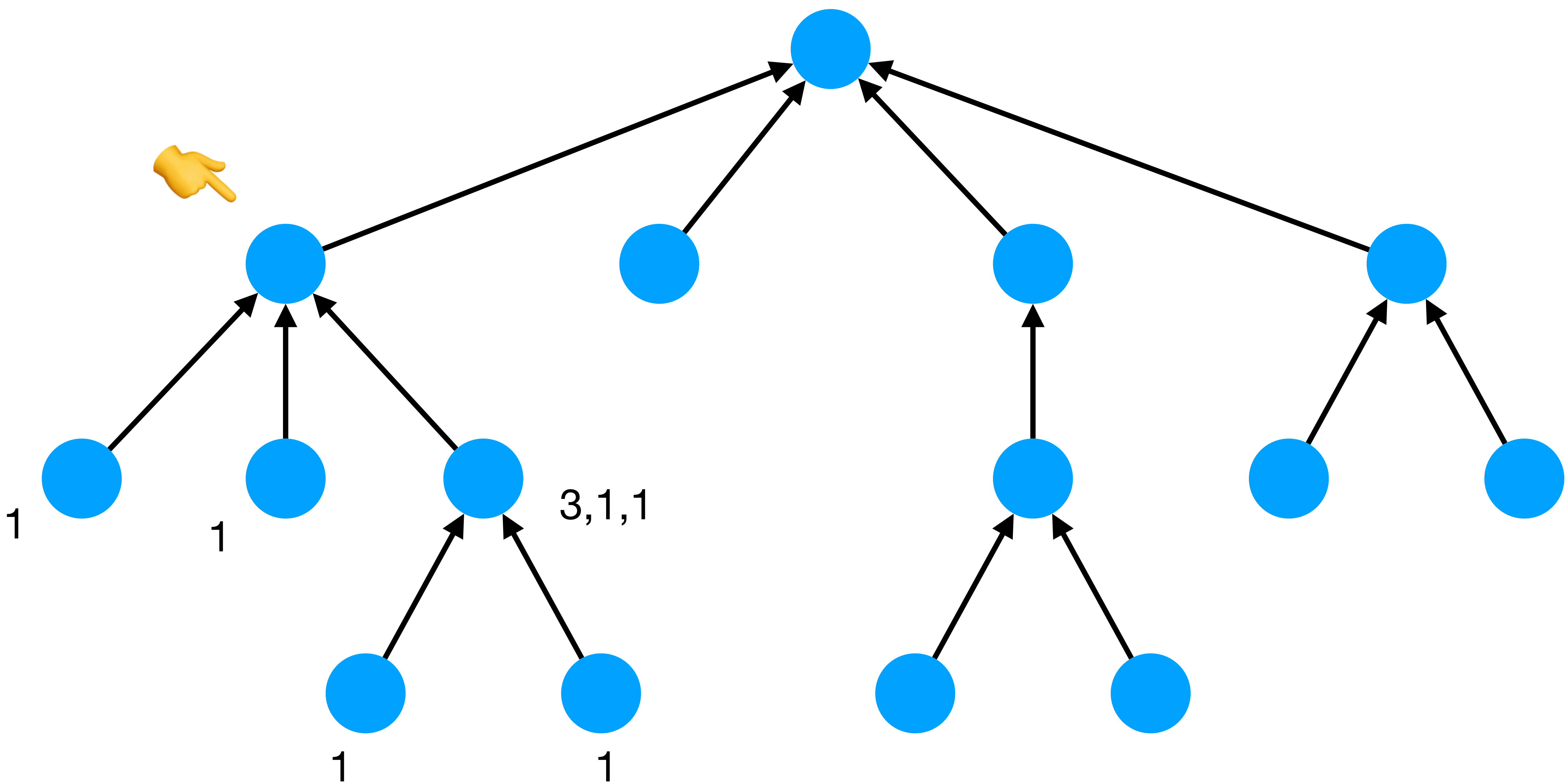
Tree canonisation

A polynomial-time algorithm



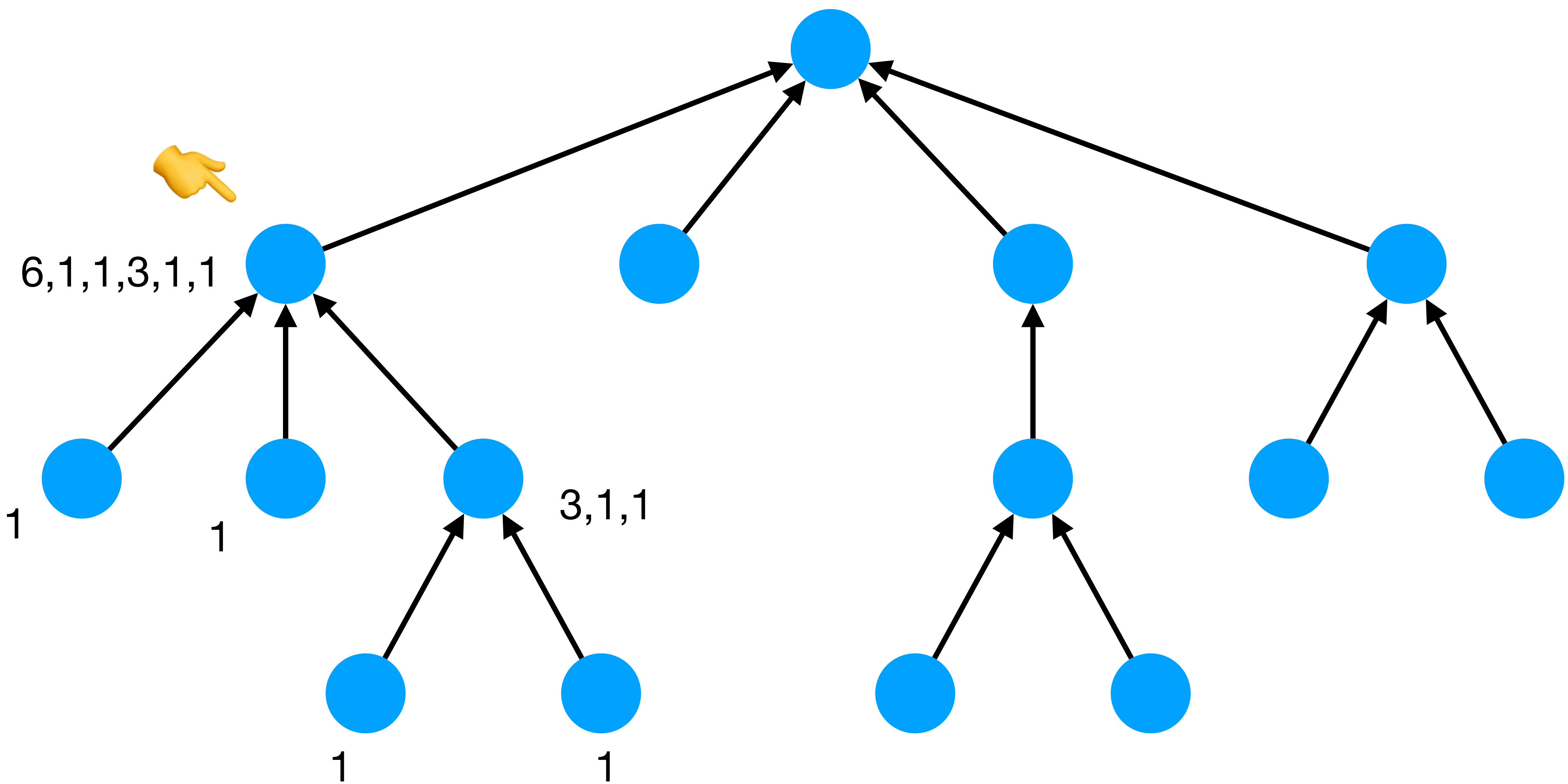
Tree canonisation

A polynomial-time algorithm



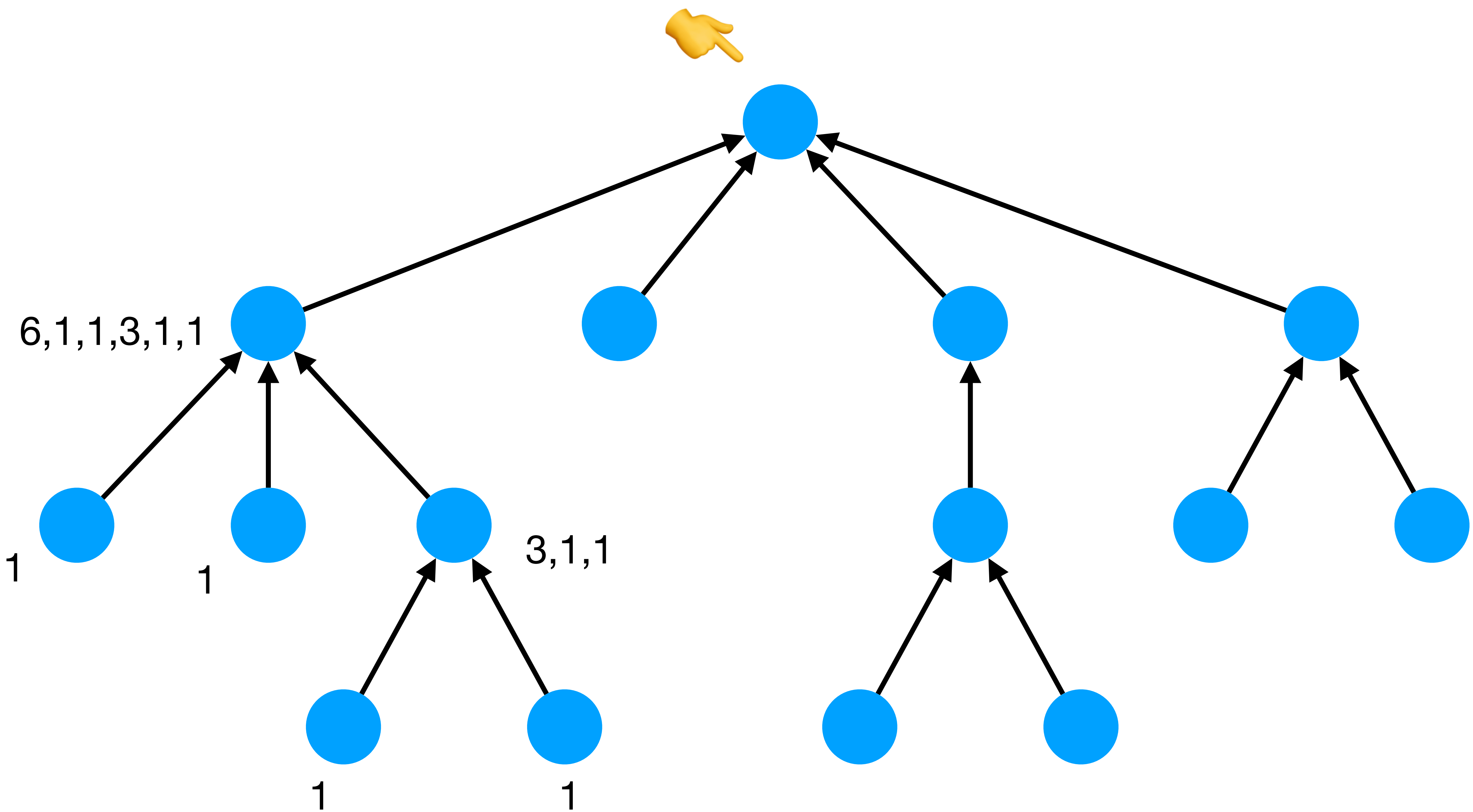
Tree canonisation

A polynomial-time algorithm



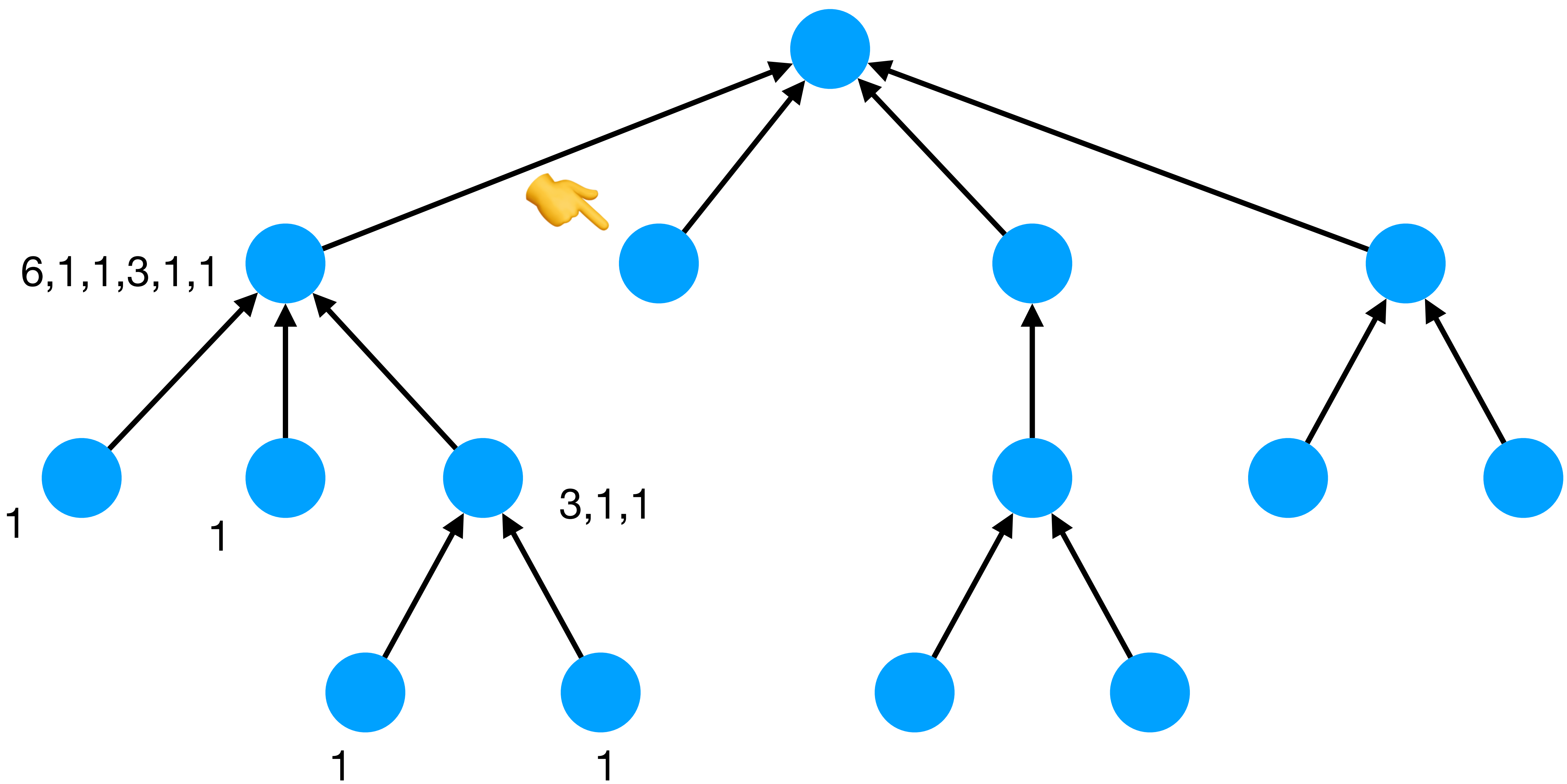
Tree canonisation

A polynomial-time algorithm



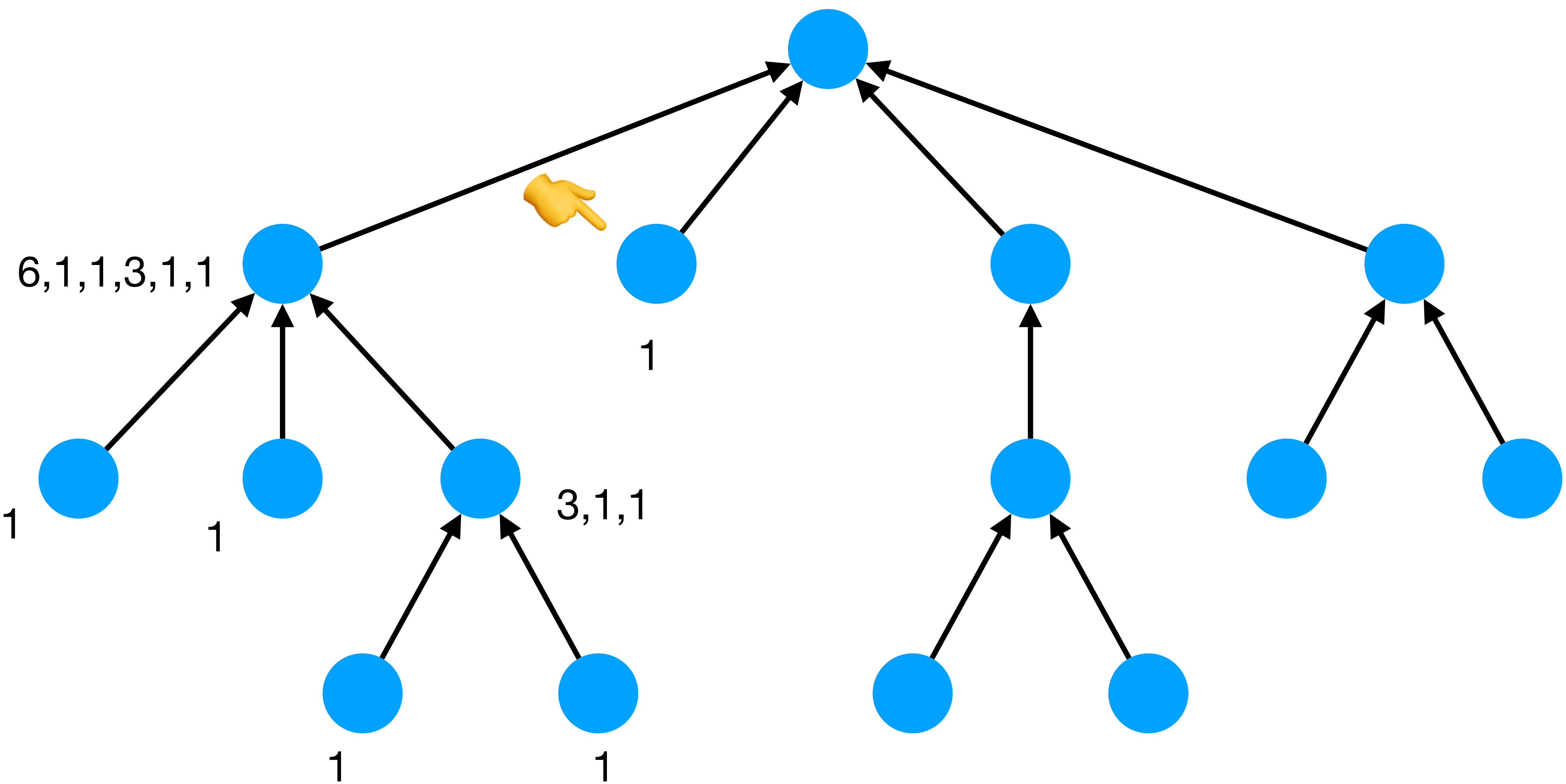
Tree canonisation

A polynomial-time algorithm



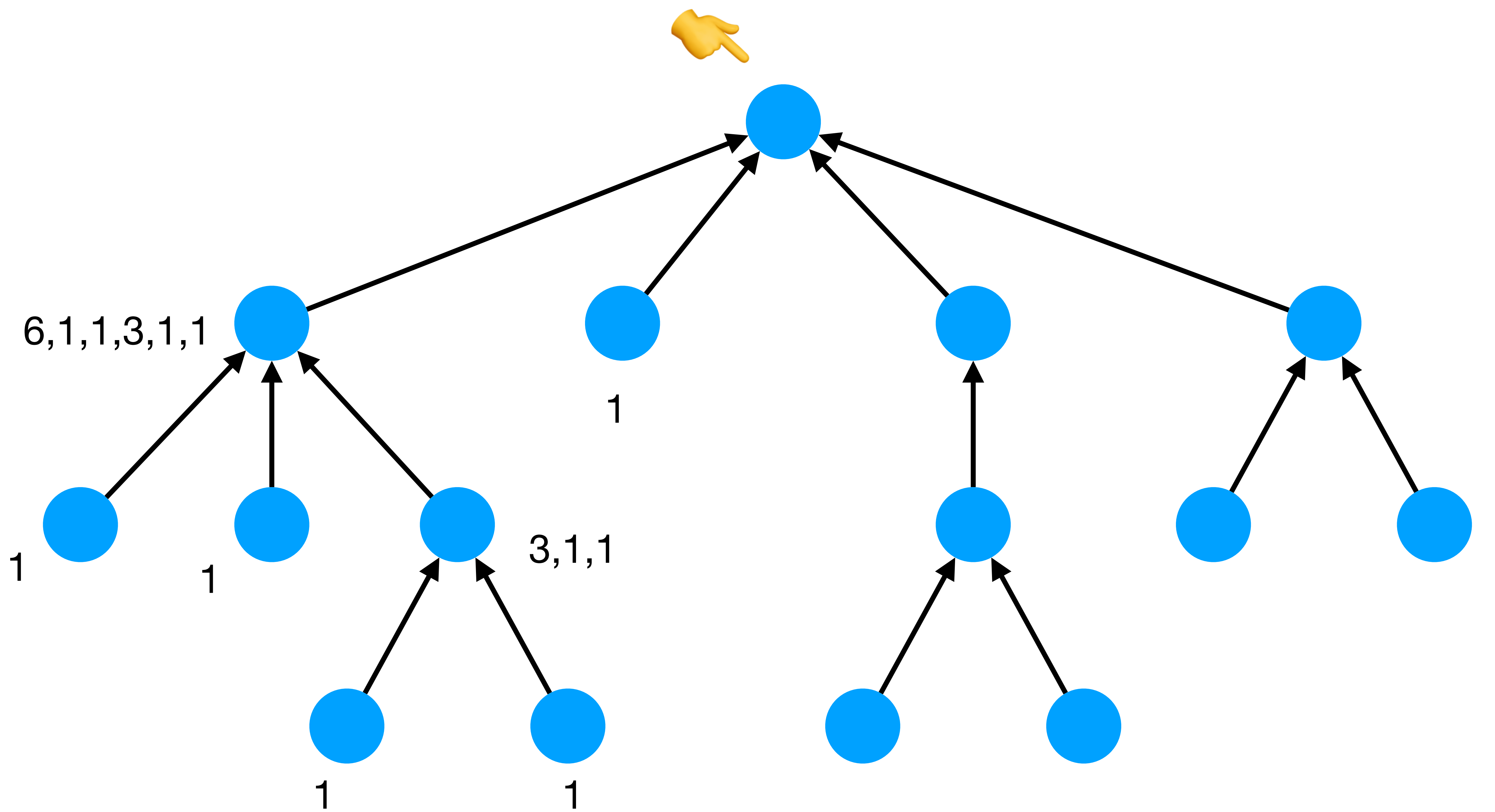
Tree canonisation

A polynomial-time algorithm



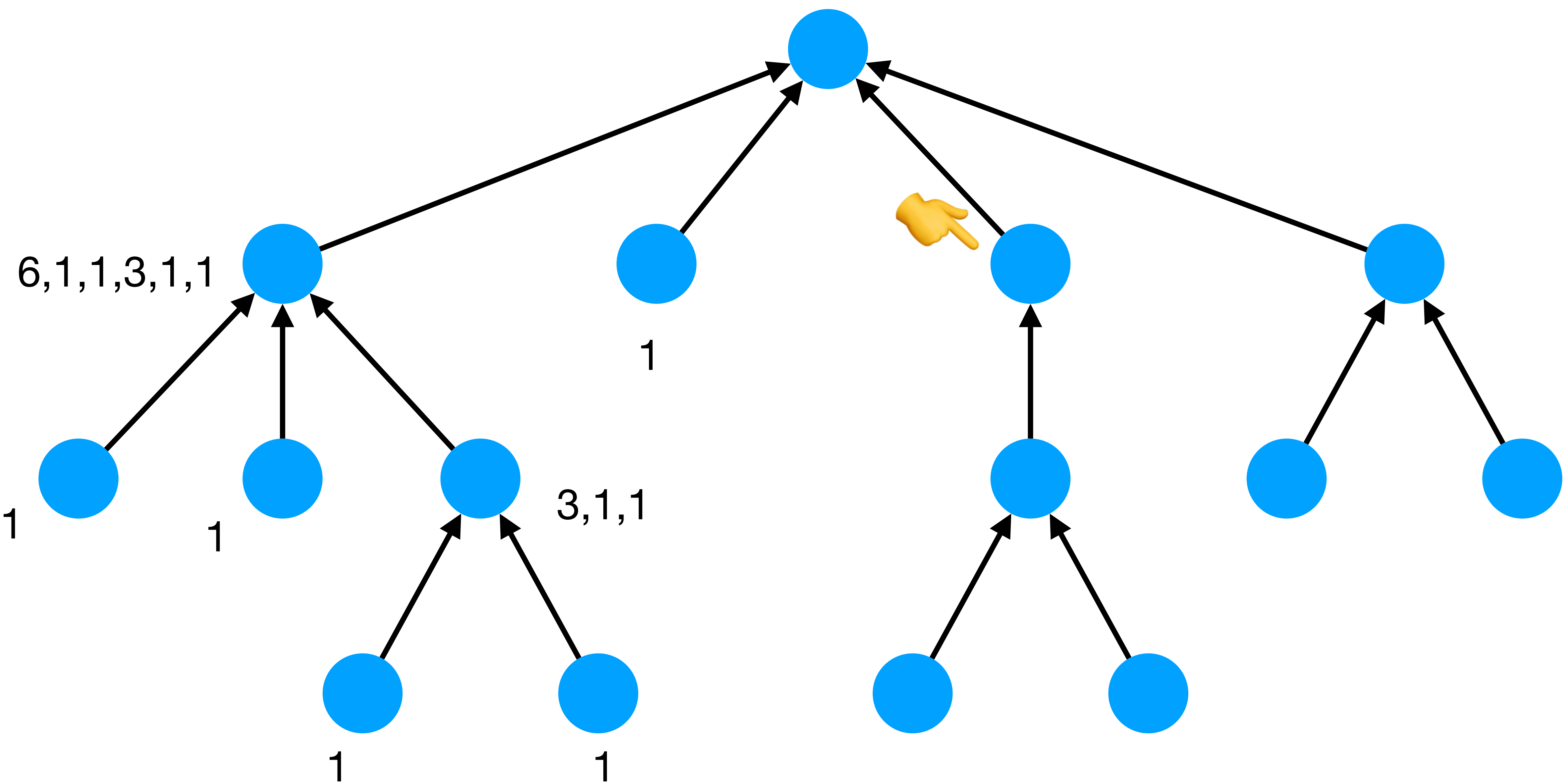
Tree canonisation

A polynomial-time algorithm



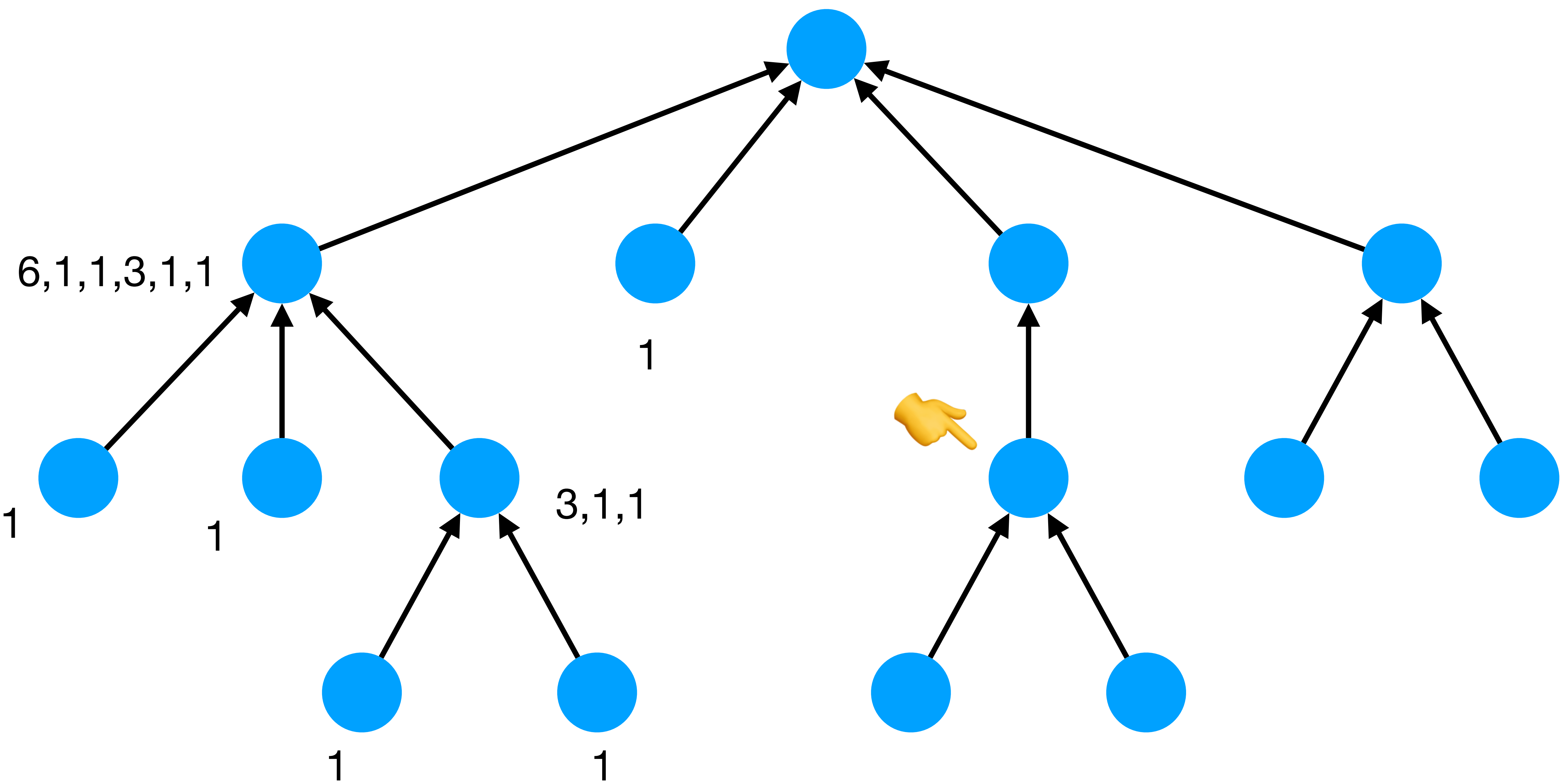
Tree canonisation

A polynomial-time algorithm



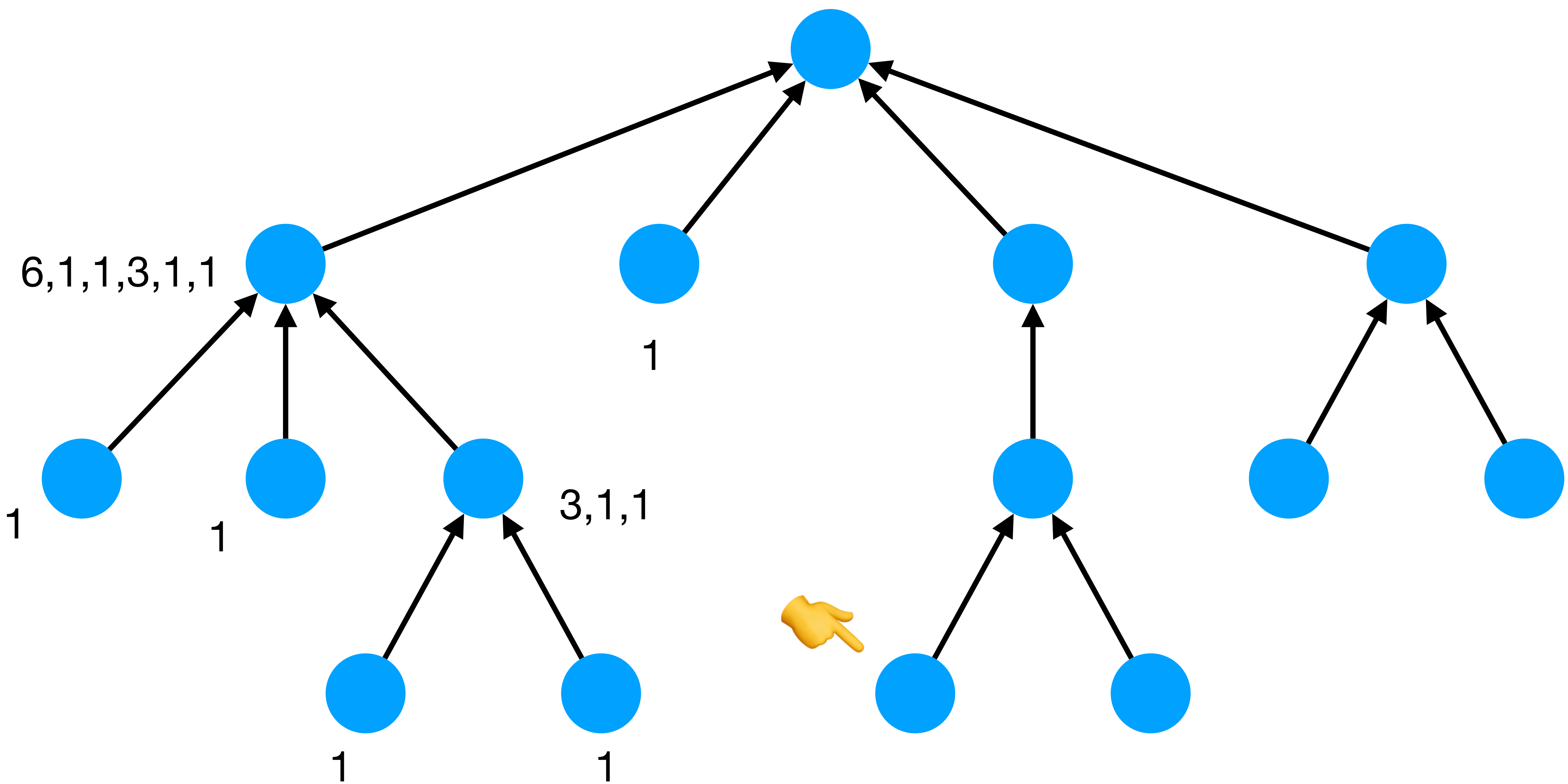
Tree canonisation

A polynomial-time algorithm



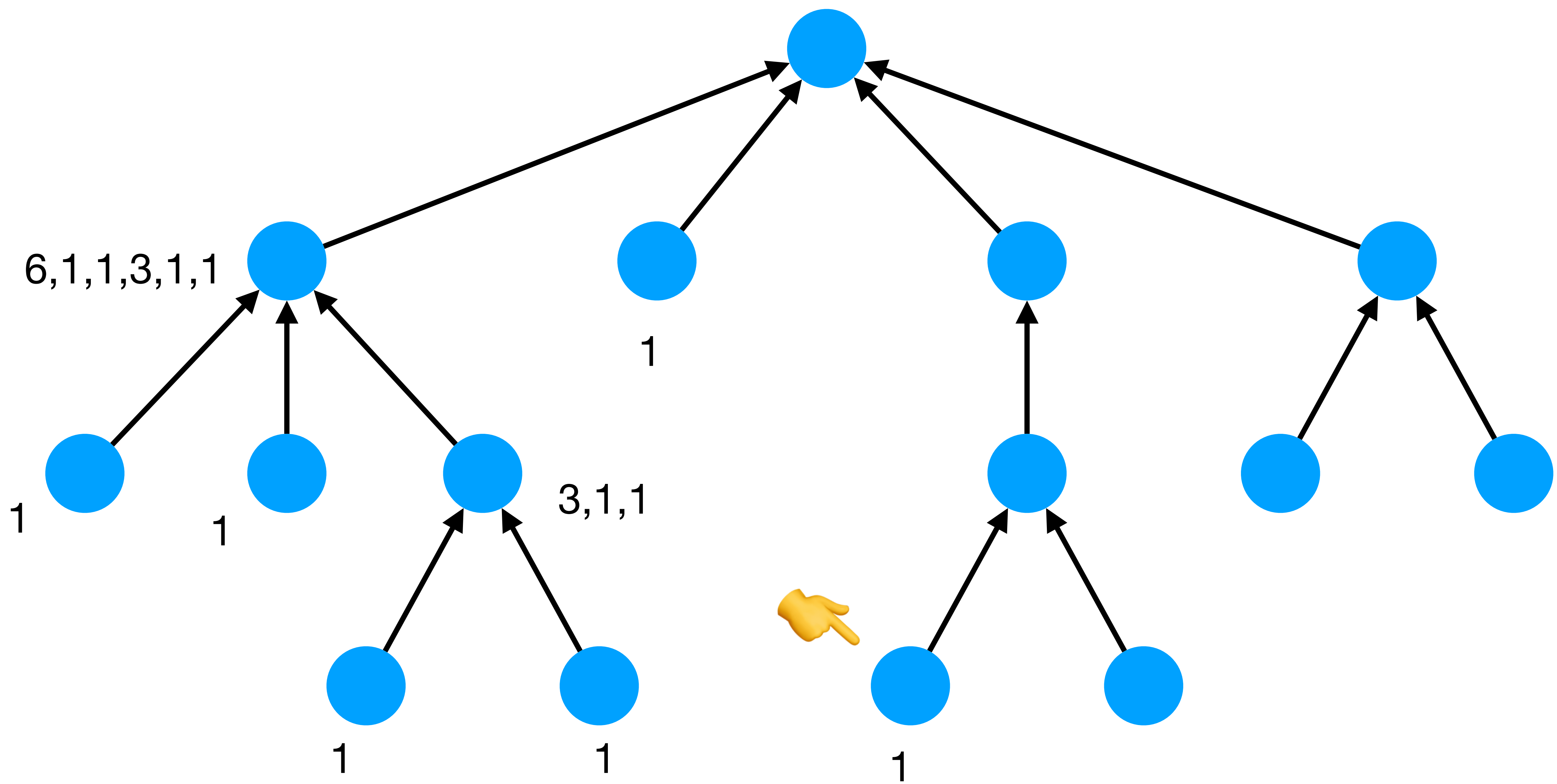
Tree canonisation

A polynomial-time algorithm



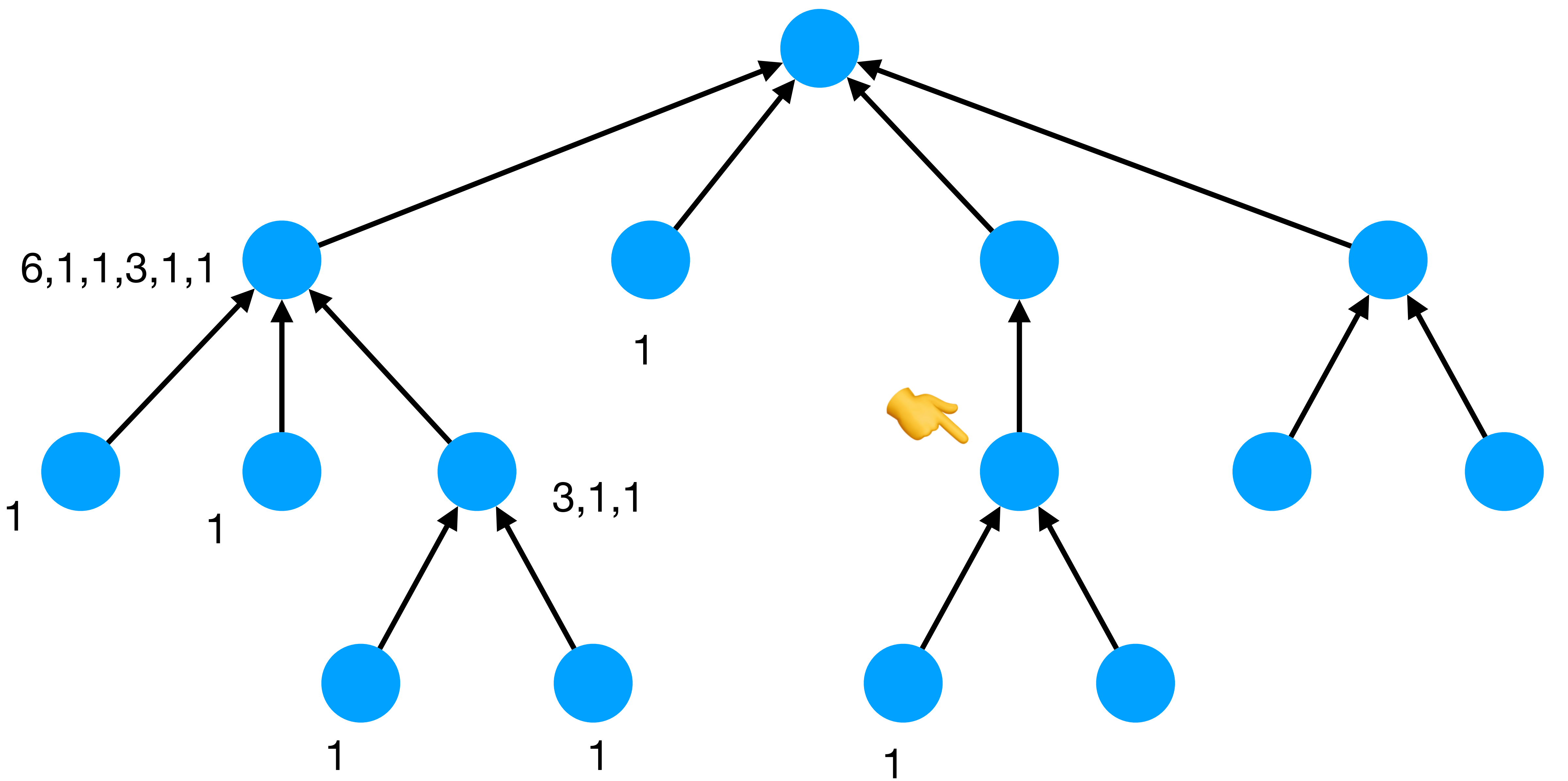
Tree canonisation

A polynomial-time algorithm



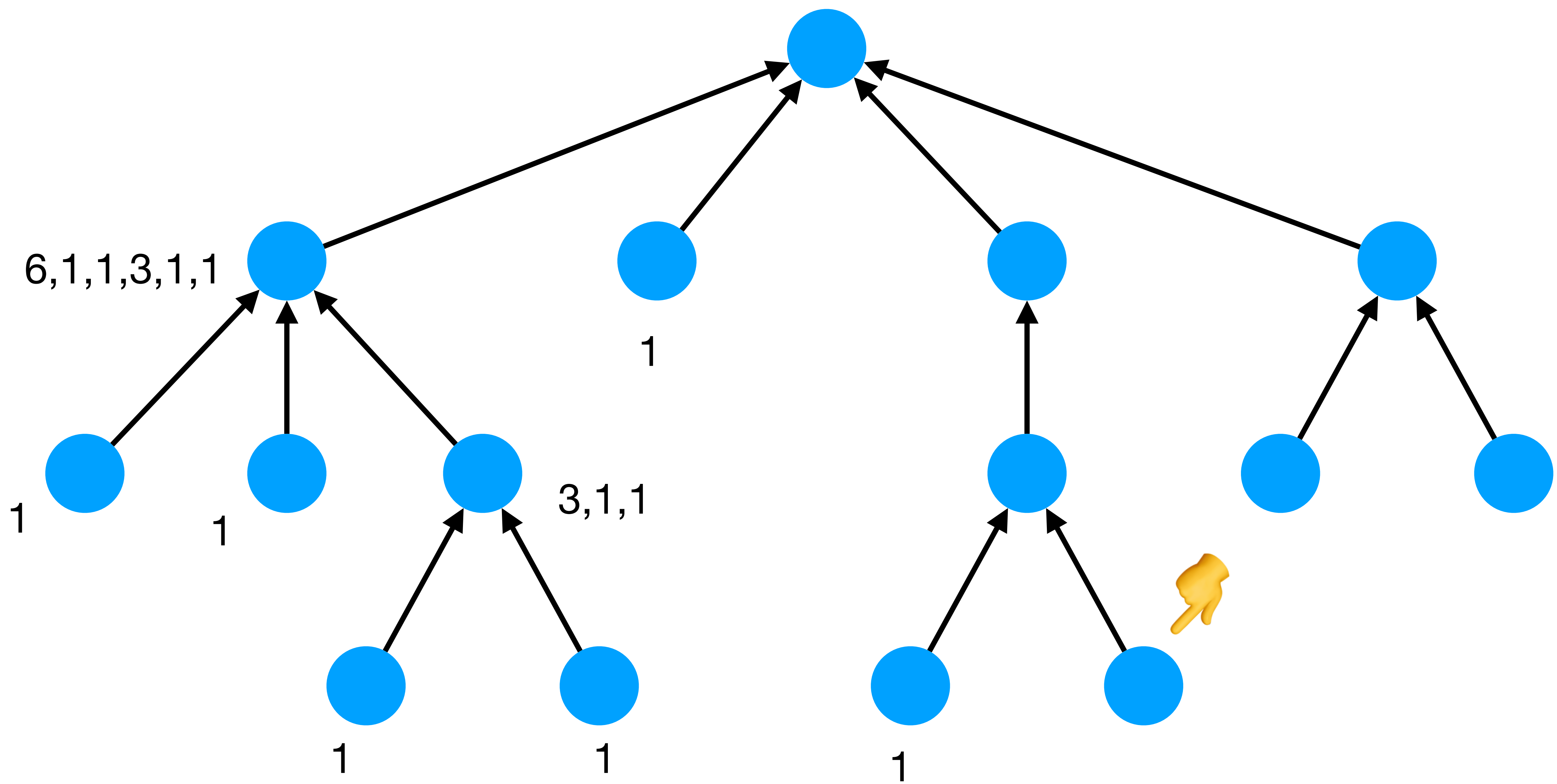
Tree canonisation

A polynomial-time algorithm



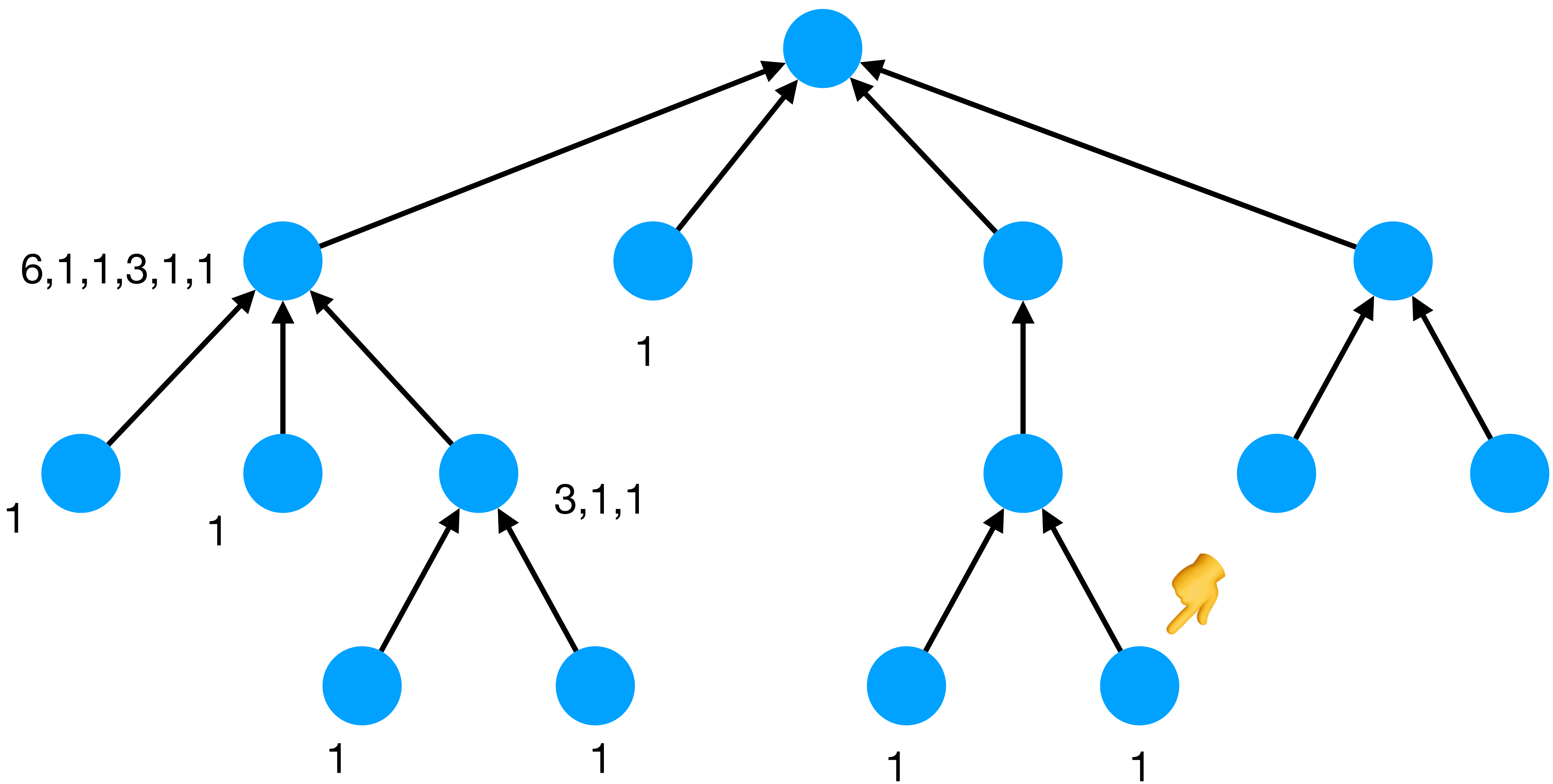
Tree canonisation

A polynomial-time algorithm



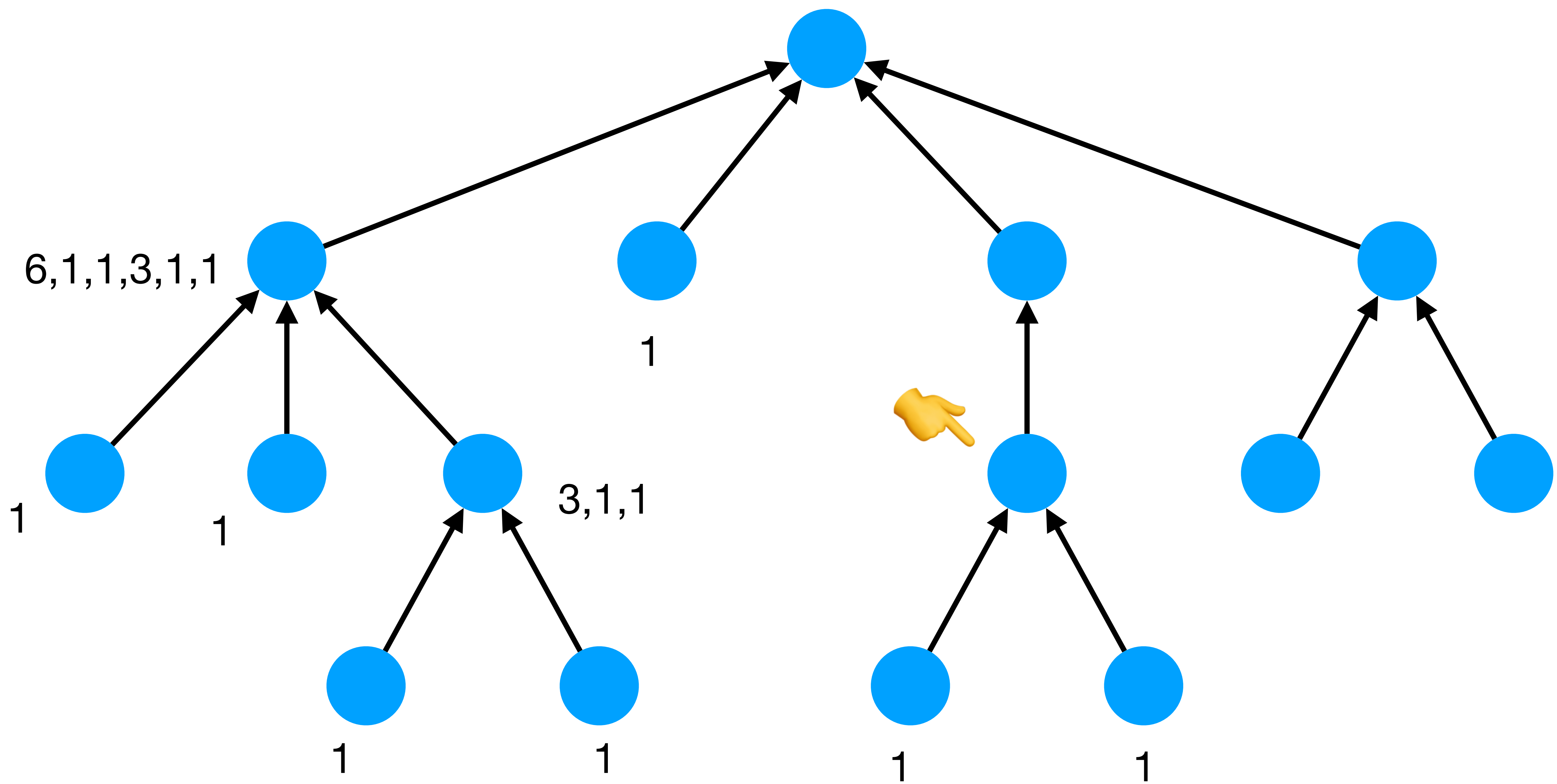
Tree canonisation

A polynomial-time algorithm



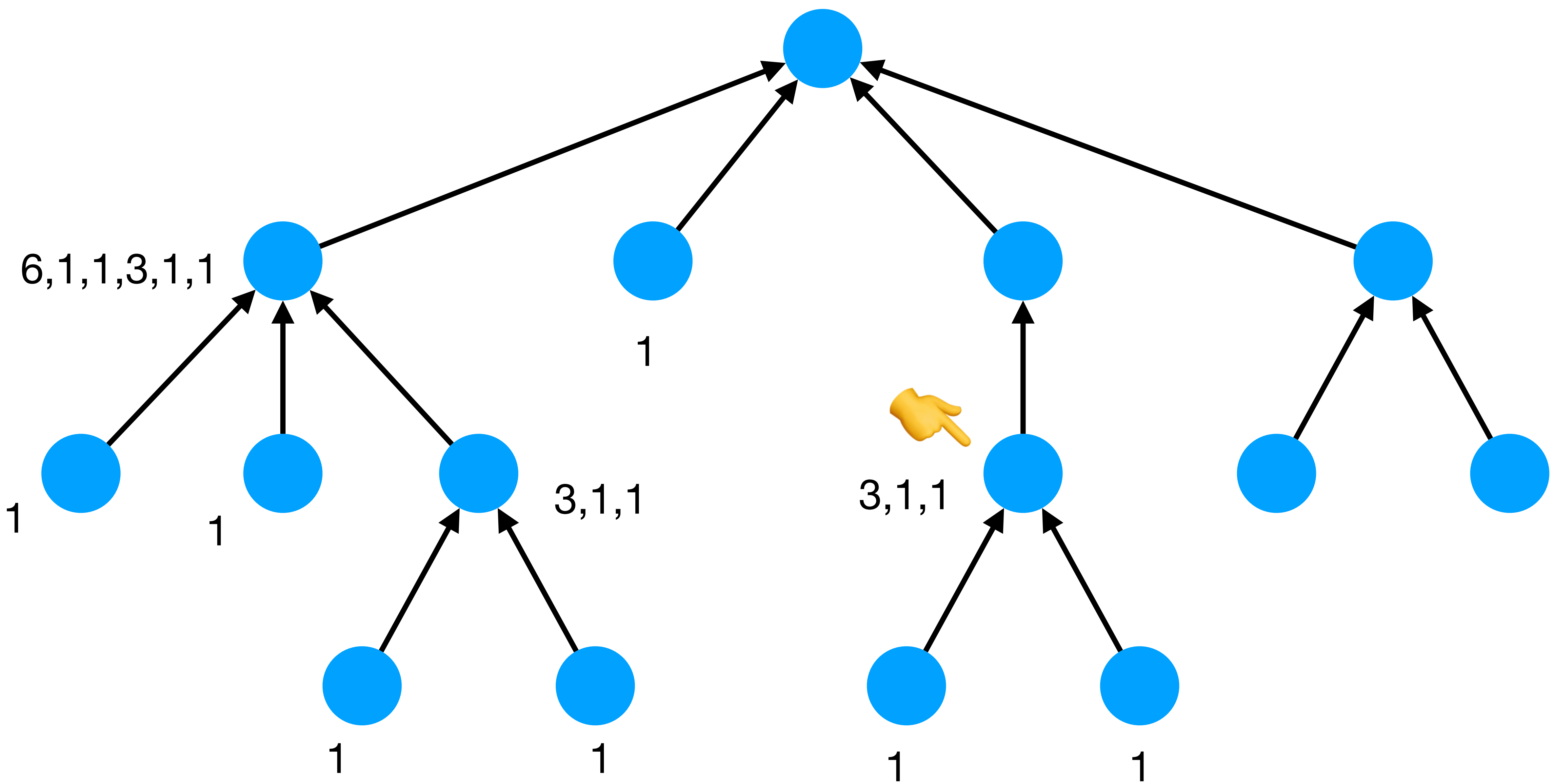
Tree canonisation

A polynomial-time algorithm



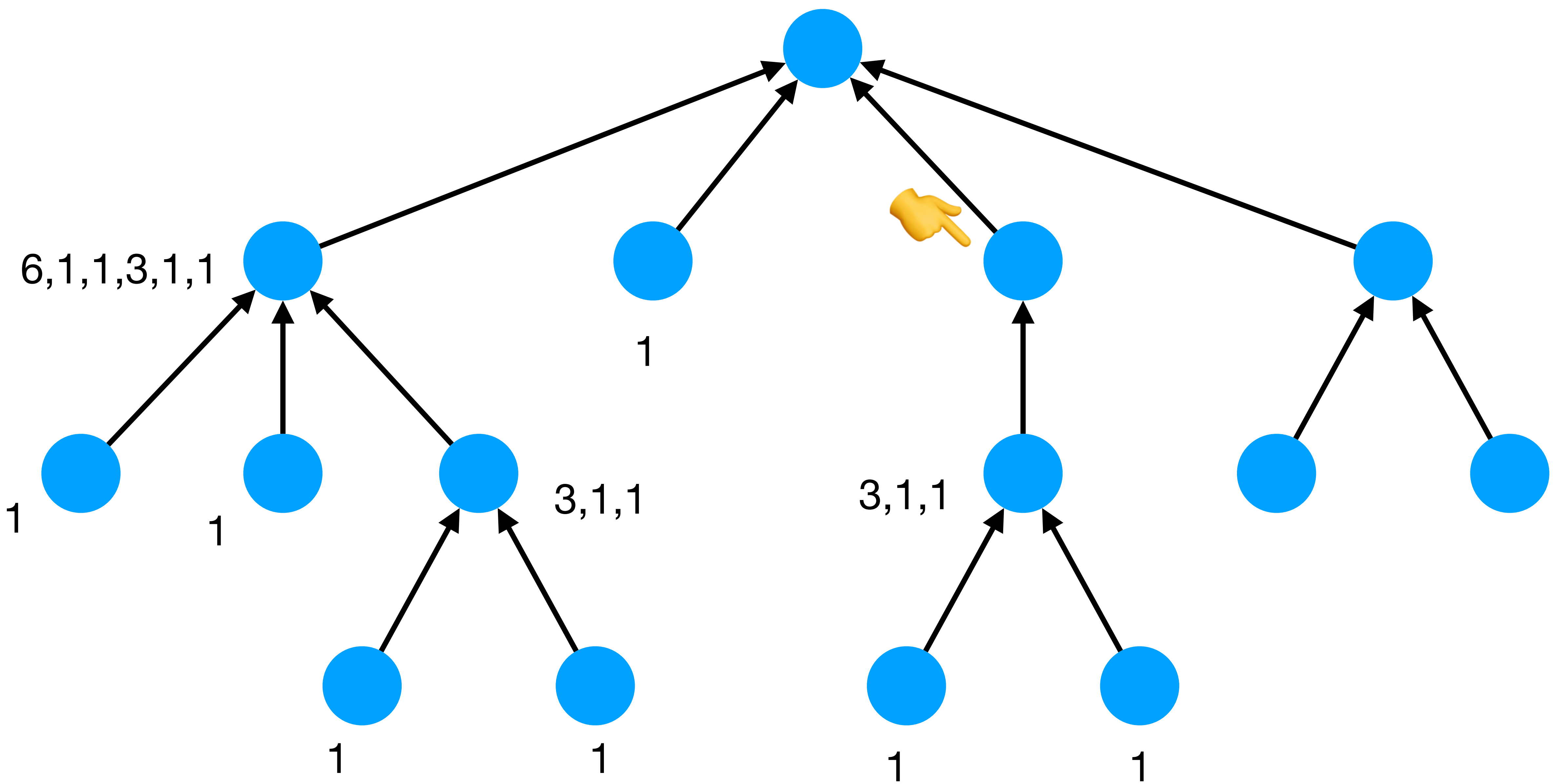
Tree canonisation

A polynomial-time algorithm



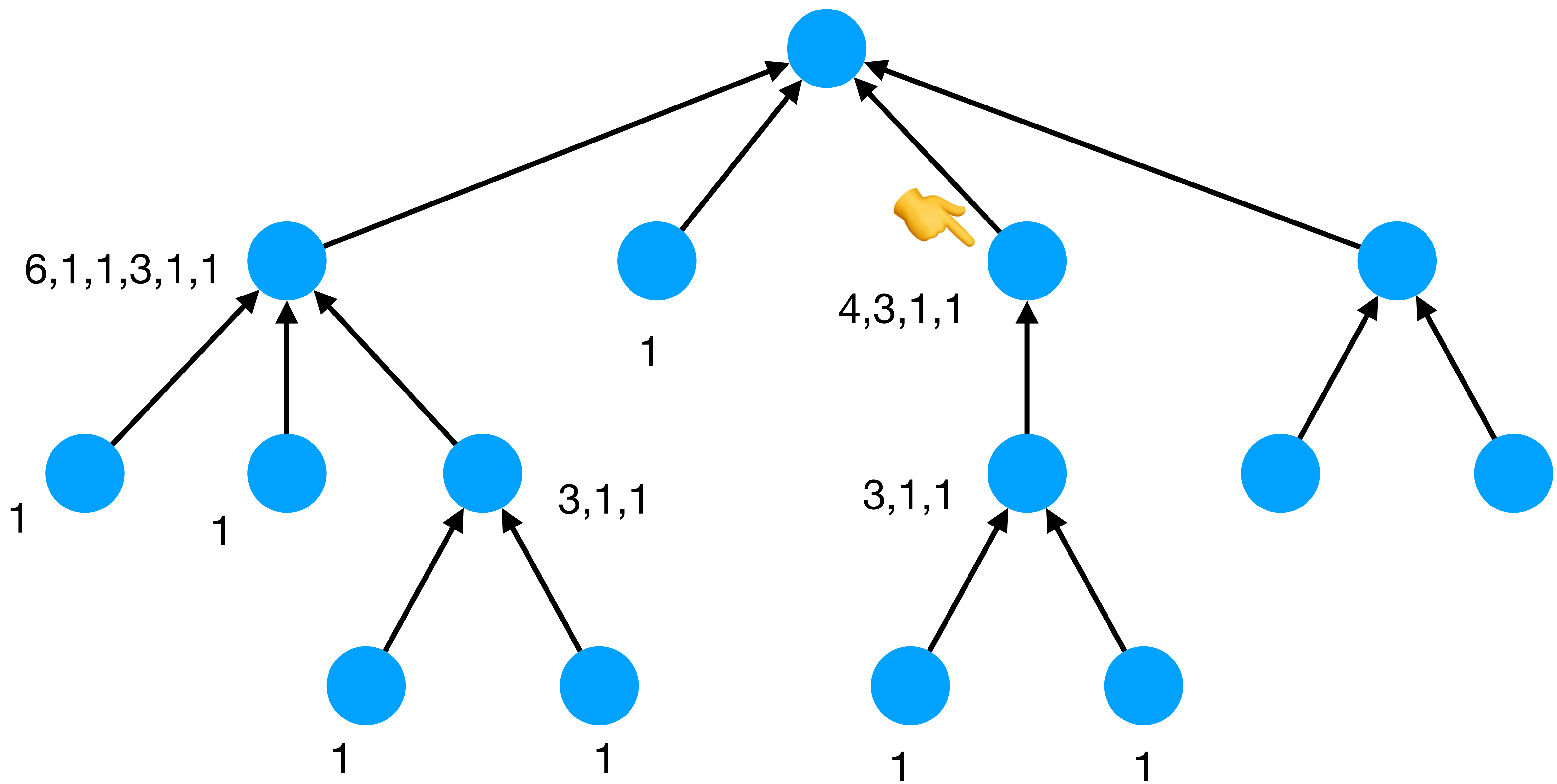
Tree canonisation

A polynomial-time algorithm



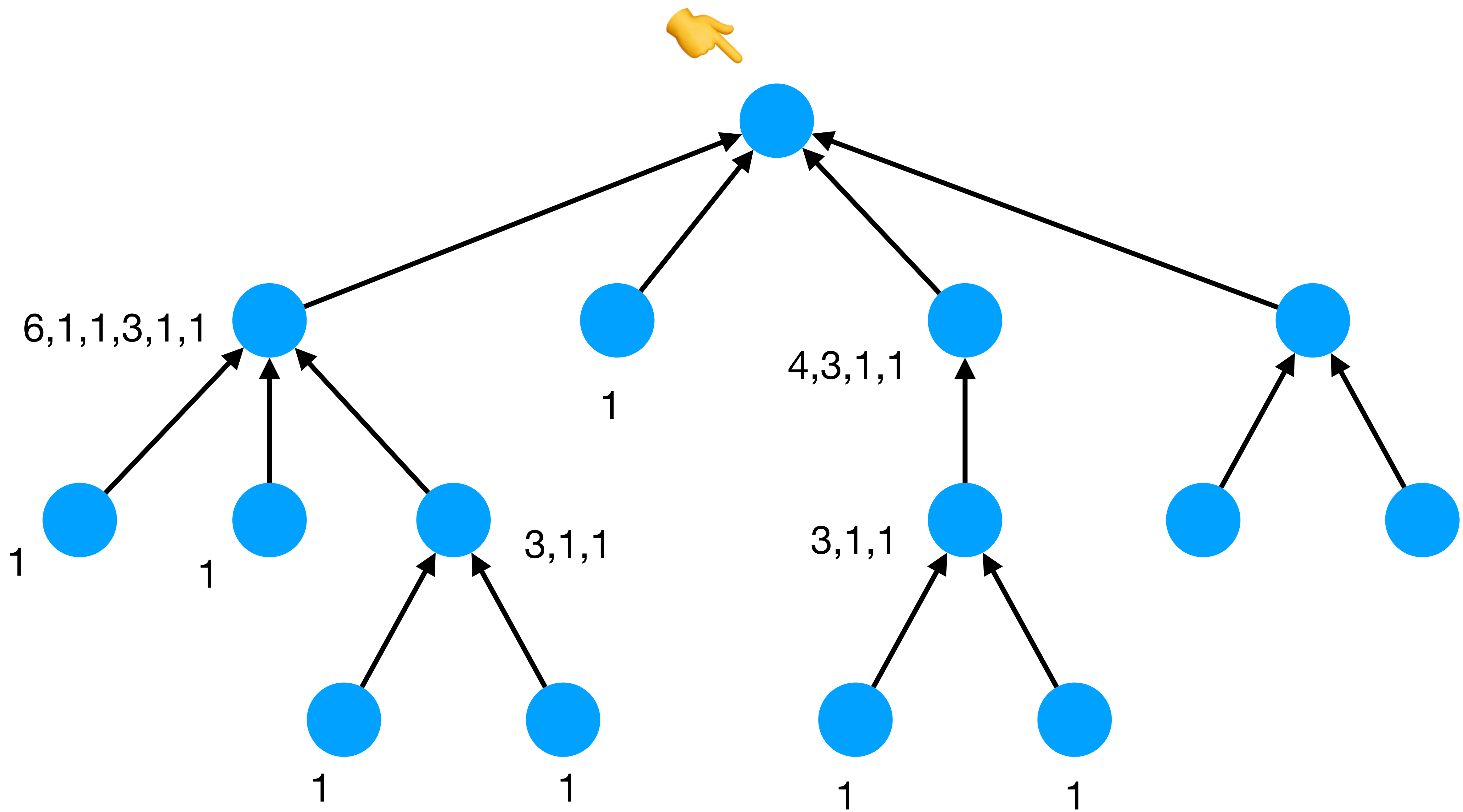
Tree canonisation

A polynomial-time algorithm



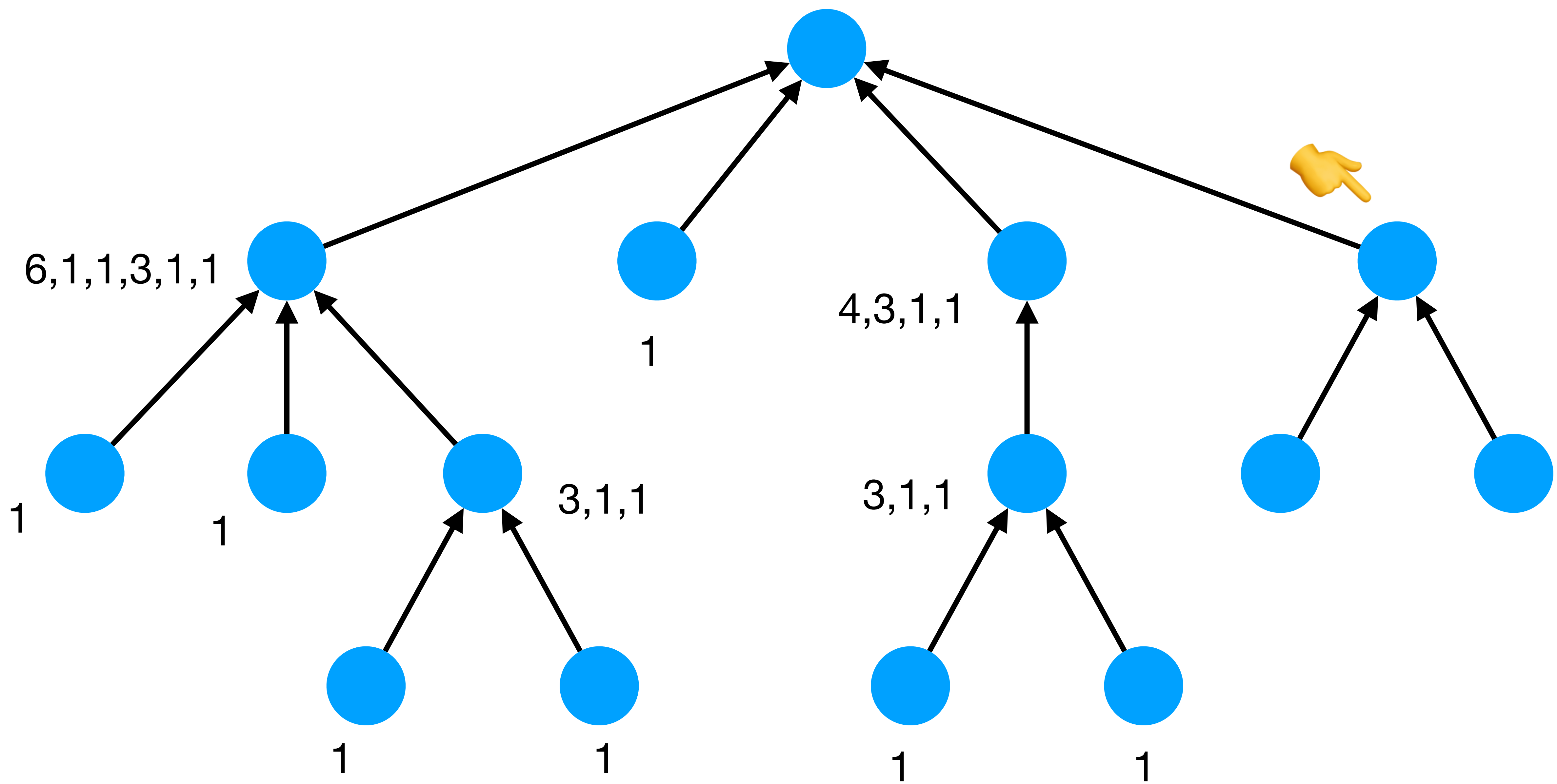
Tree canonisation

A polynomial-time algorithm



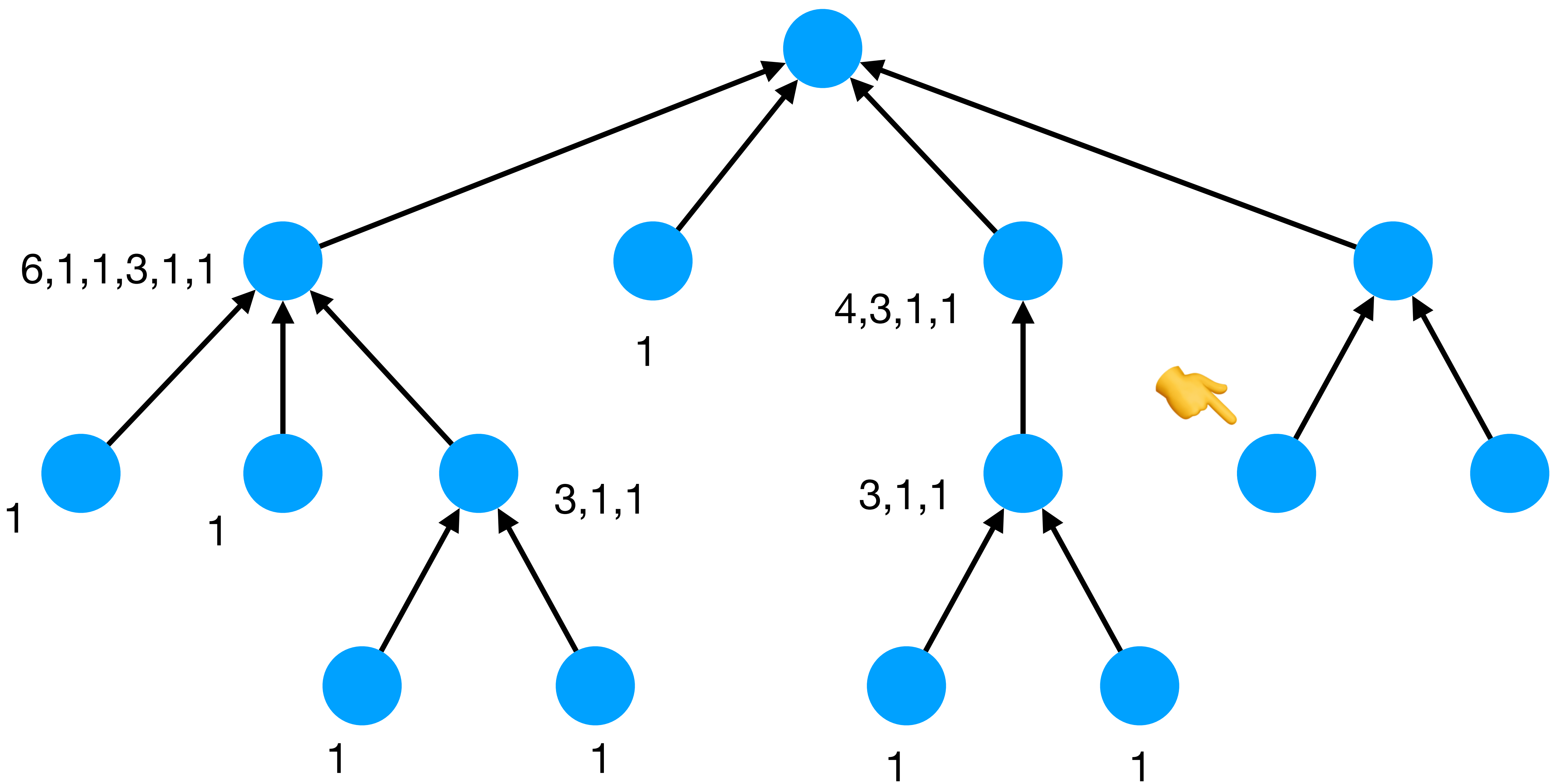
Tree canonisation

A polynomial-time algorithm



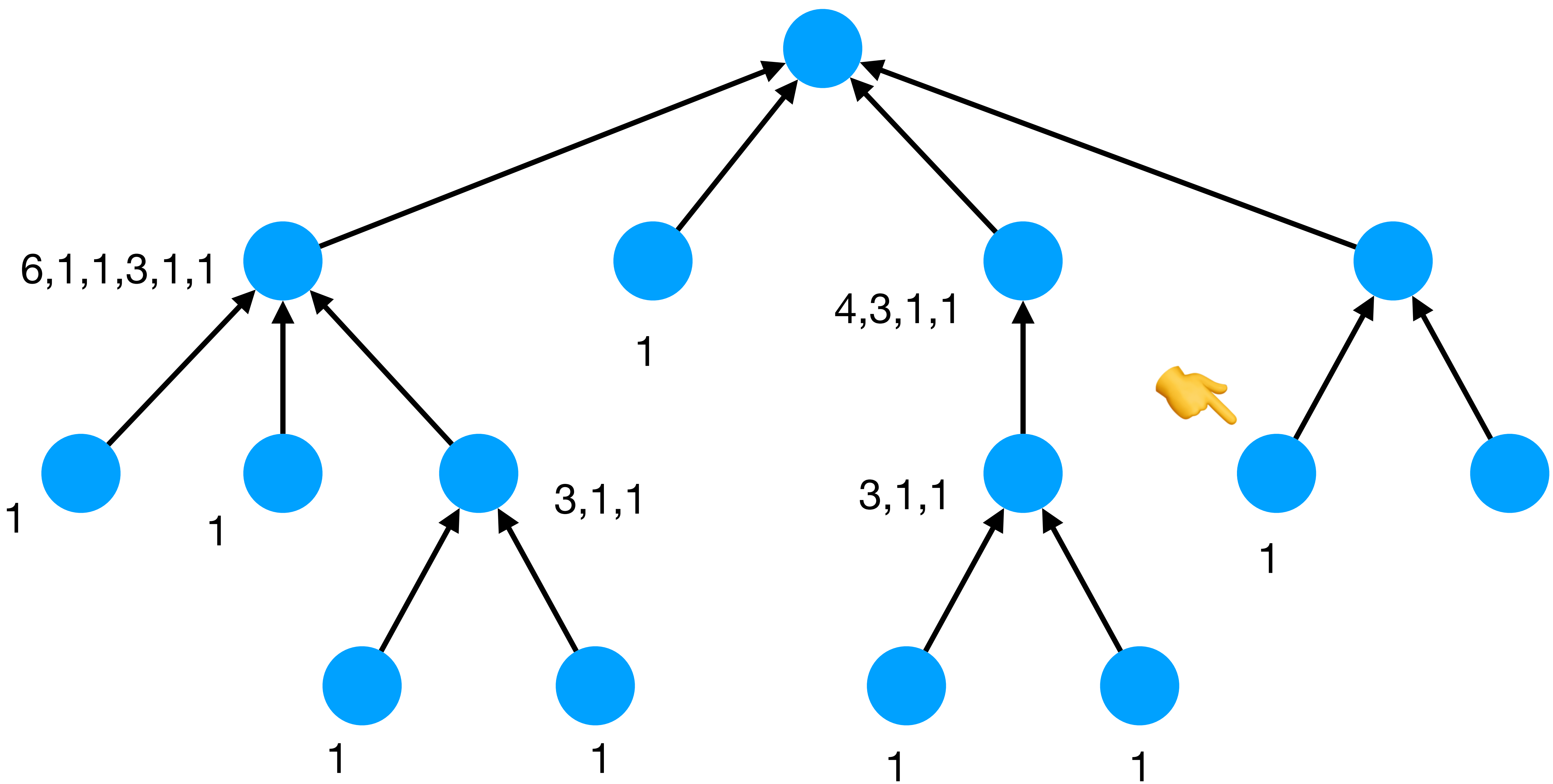
Tree canonisation

A polynomial-time algorithm



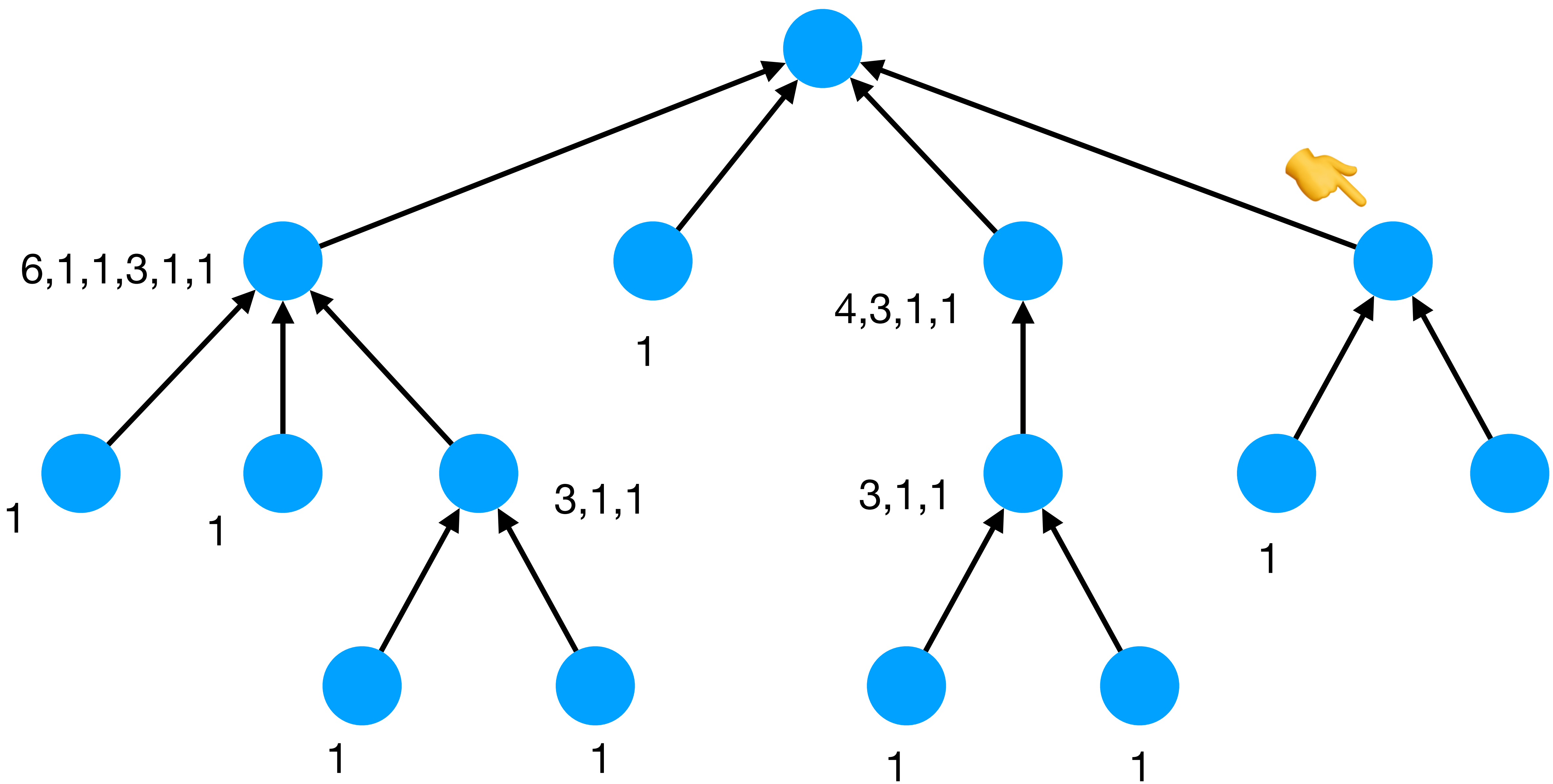
Tree canonisation

A polynomial-time algorithm



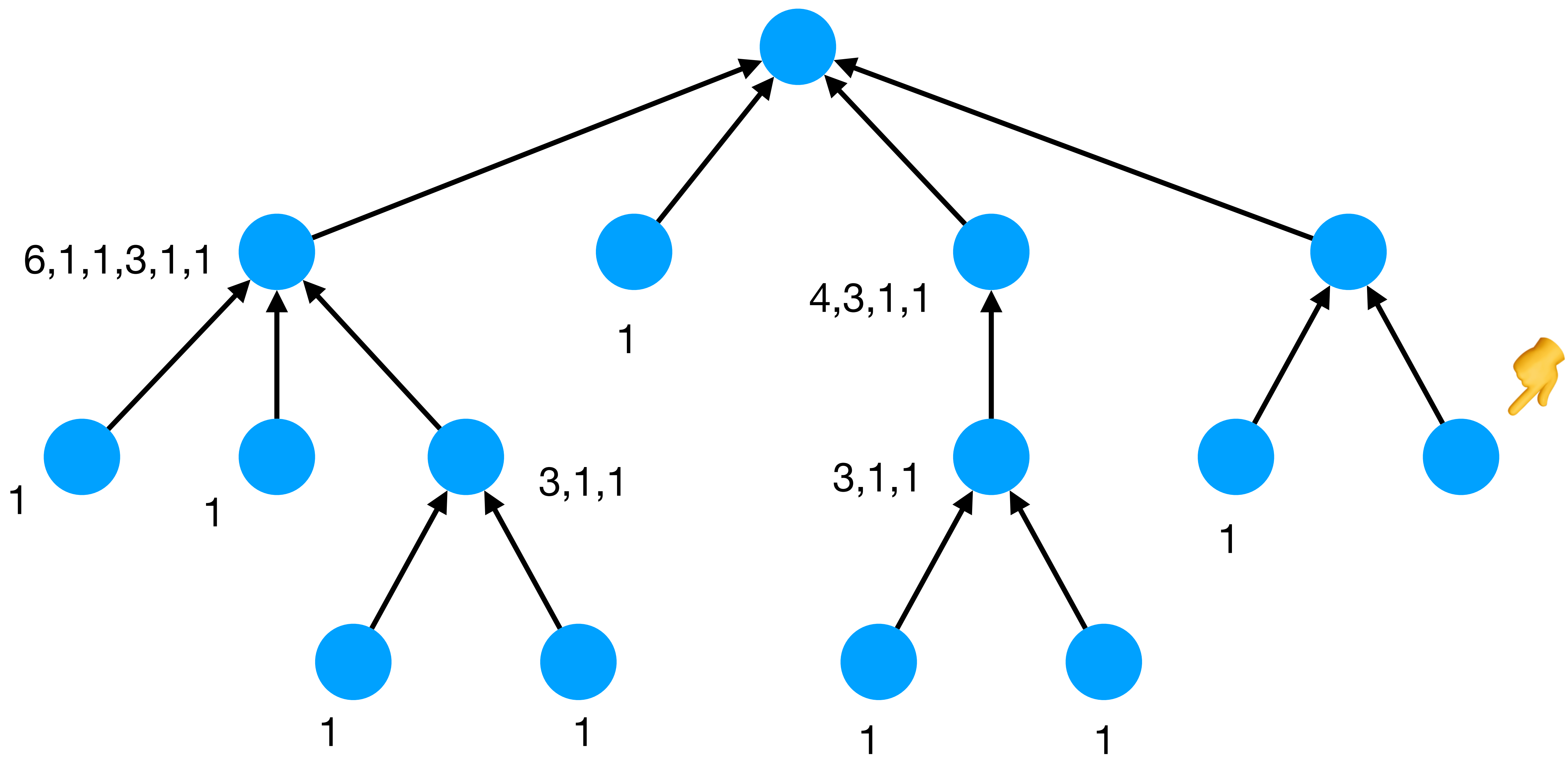
Tree canonisation

A polynomial-time algorithm



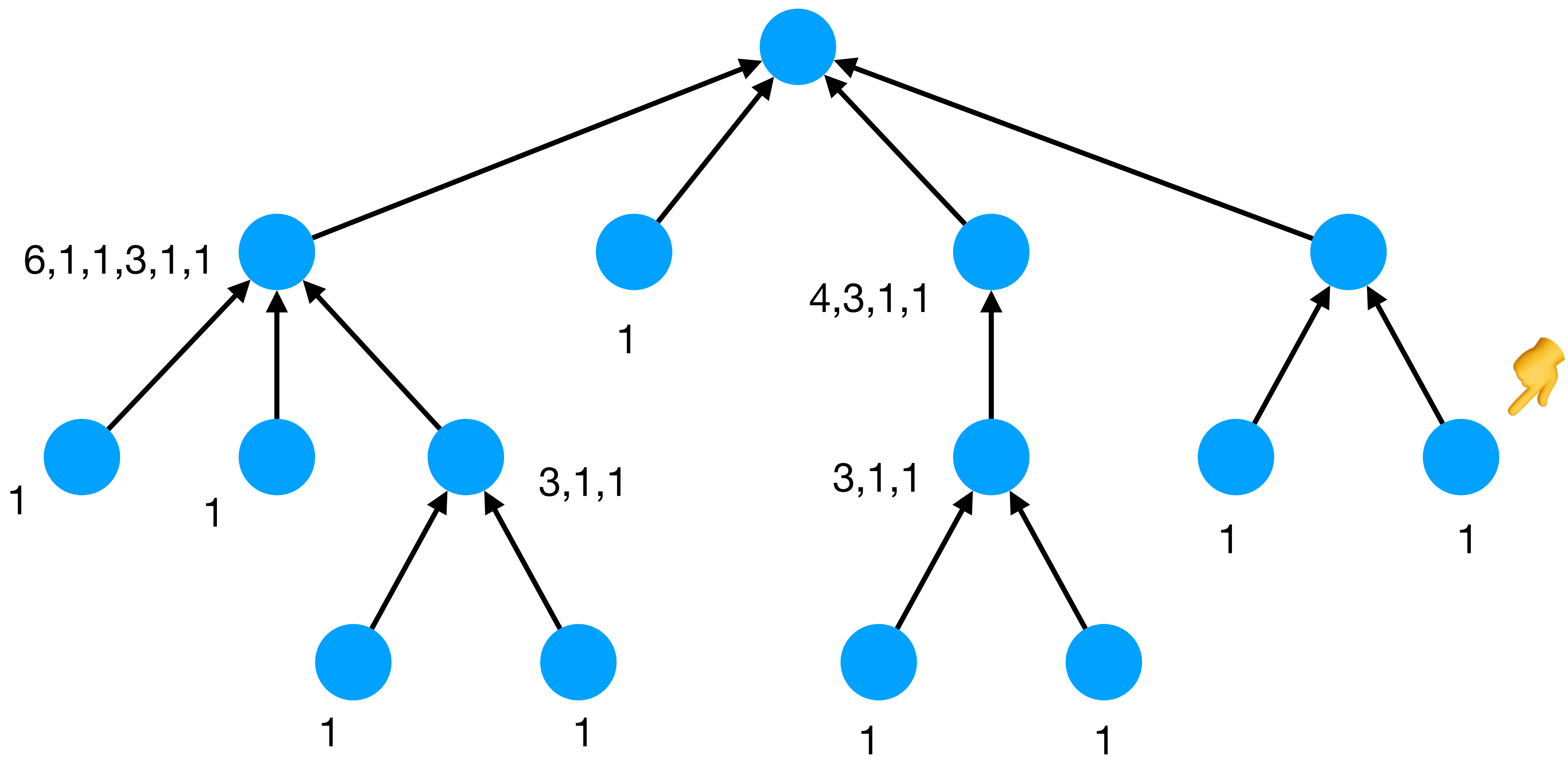
Tree canonisation

A polynomial-time algorithm



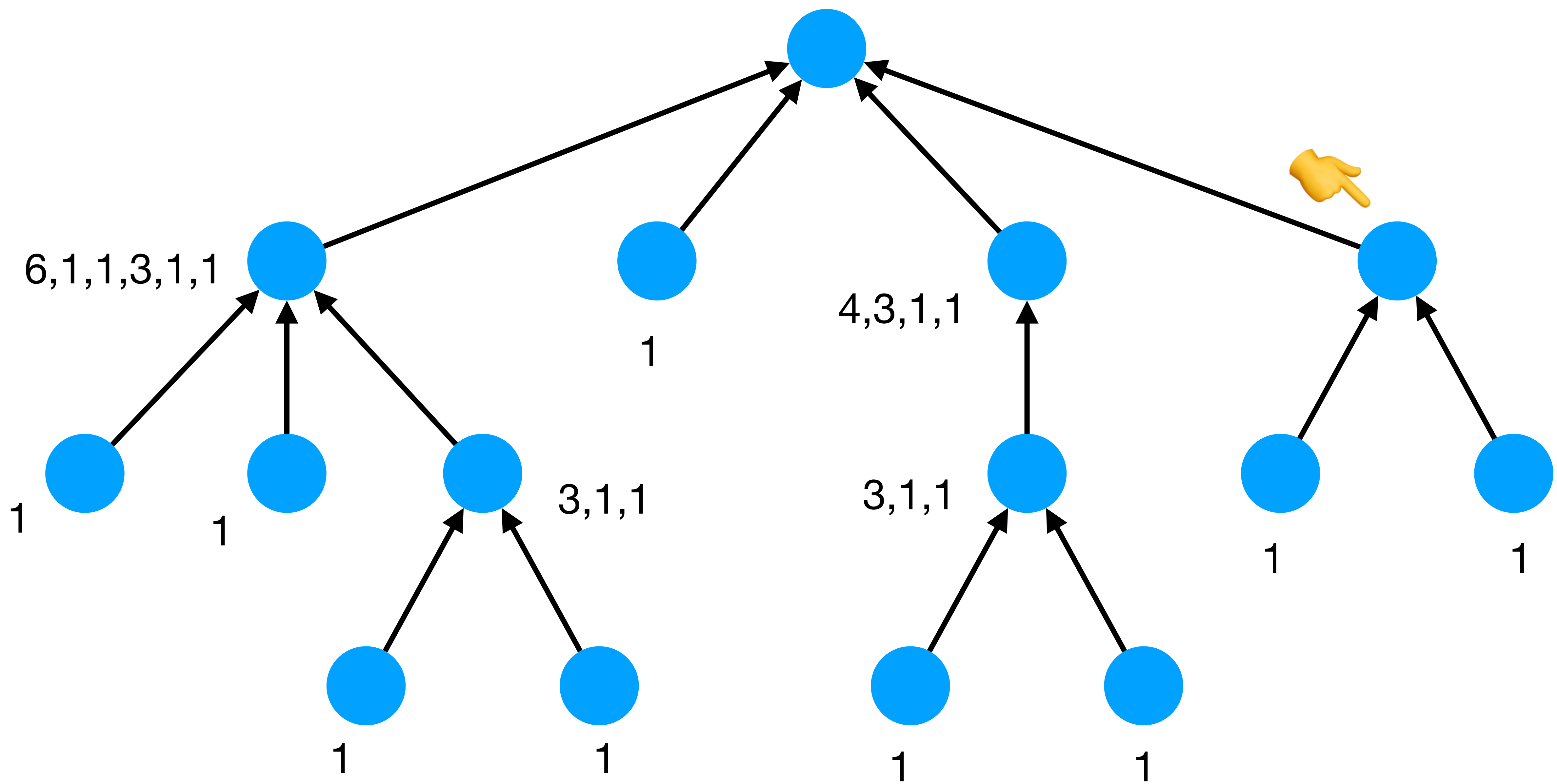
Tree canonisation

A polynomial-time algorithm



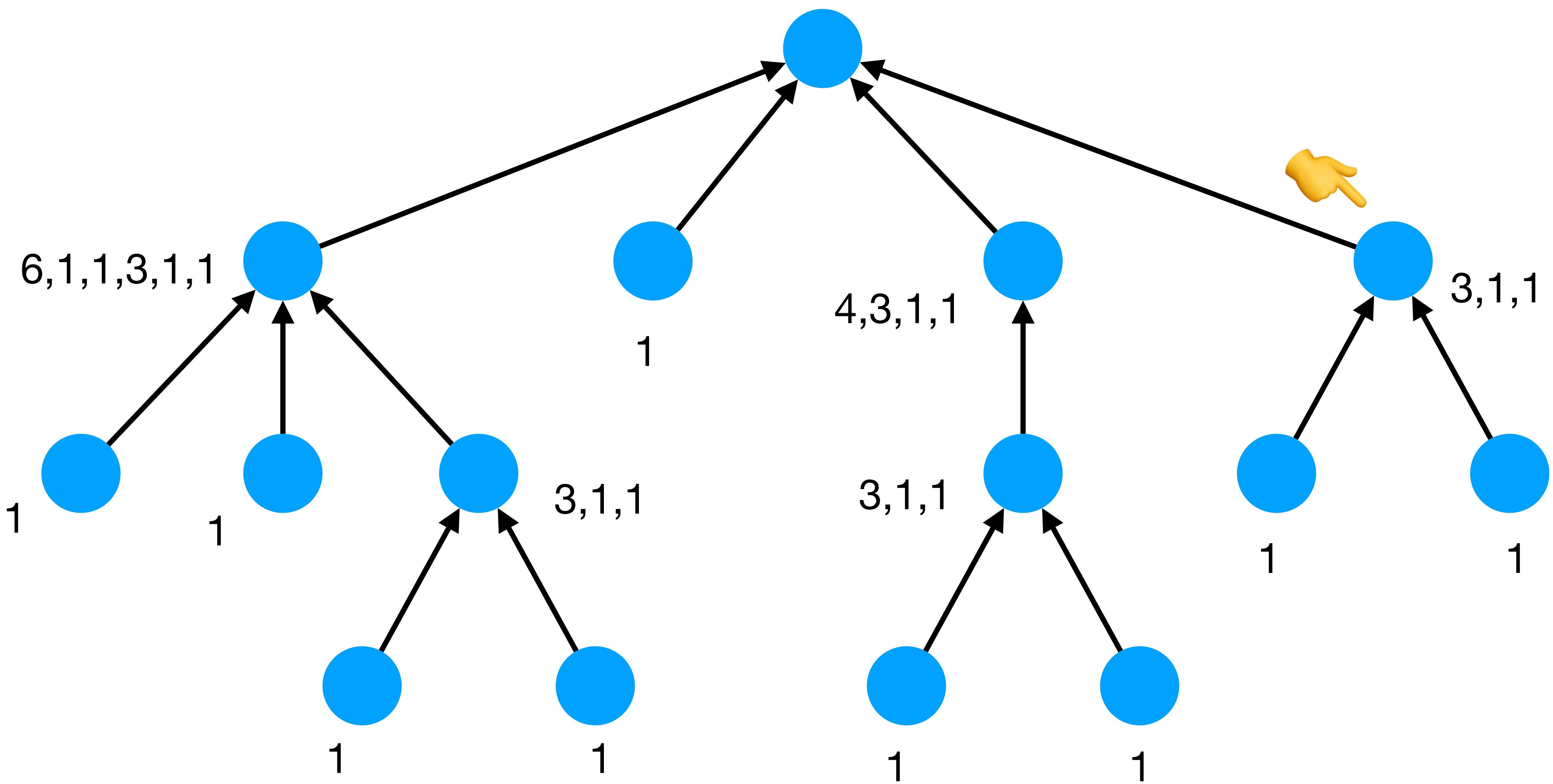
Tree canonisation

A polynomial-time algorithm



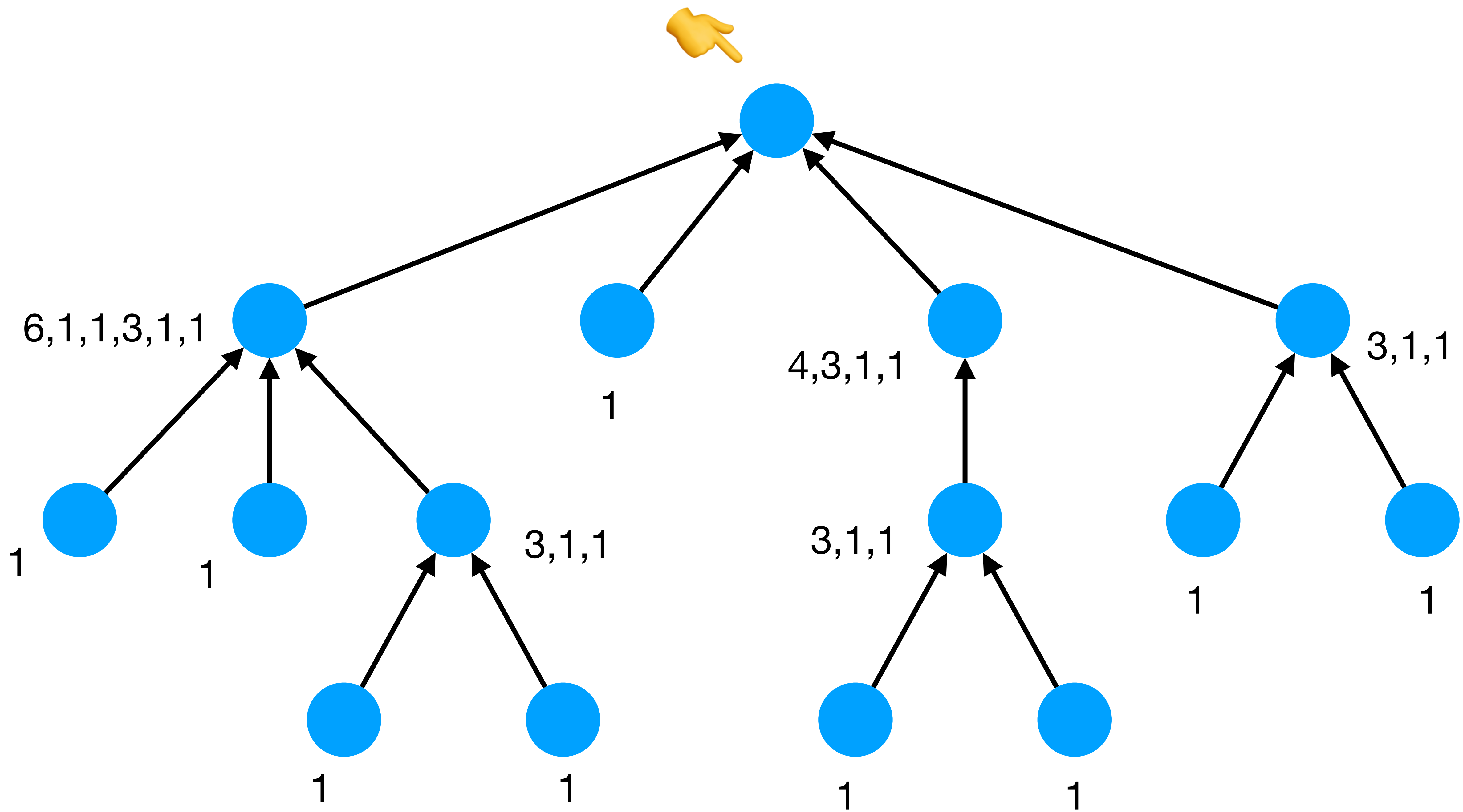
Tree canonisation

A polynomial-time algorithm



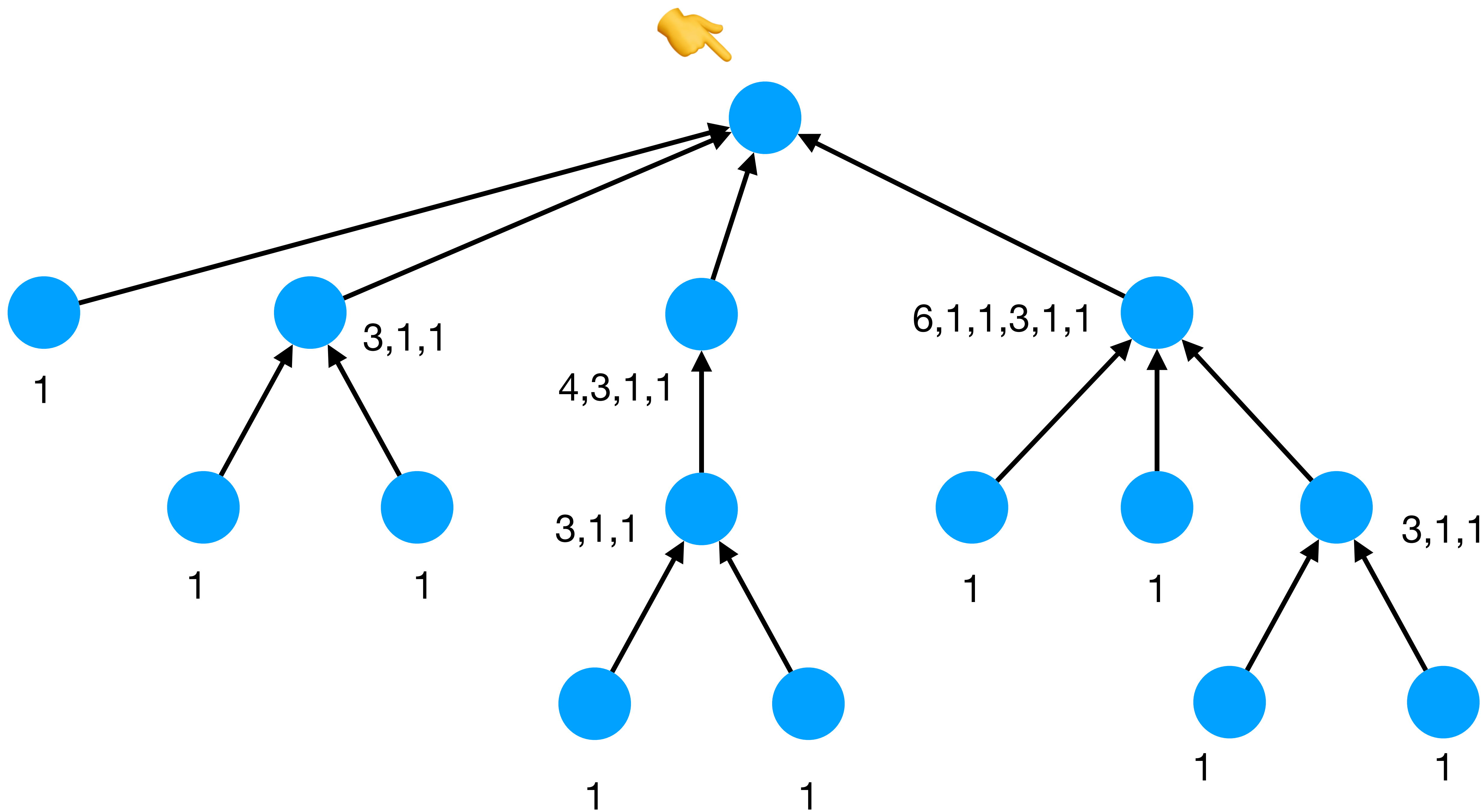
Tree canonisation

A polynomial-time algorithm



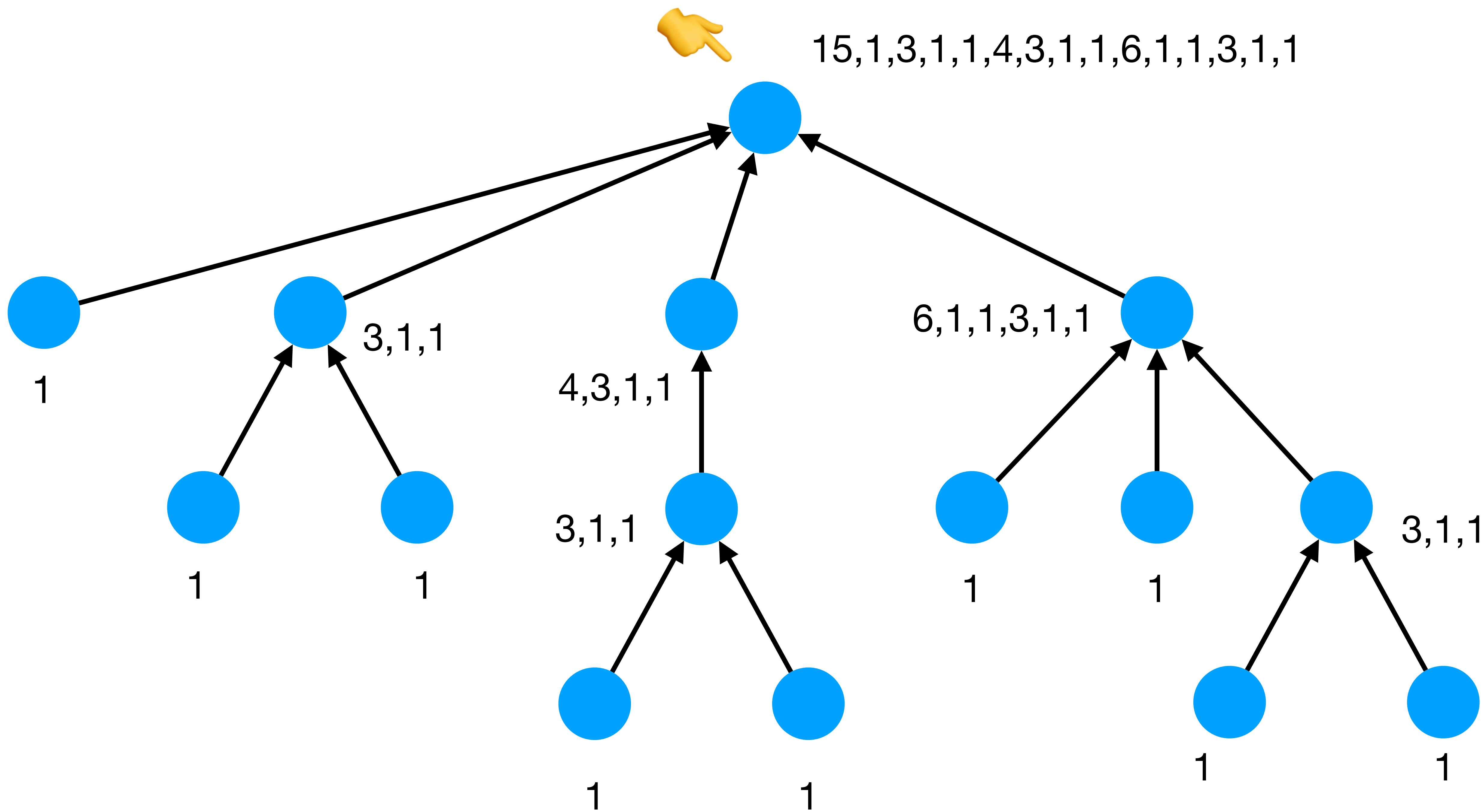
Tree canonisation

A polynomial-time algorithm



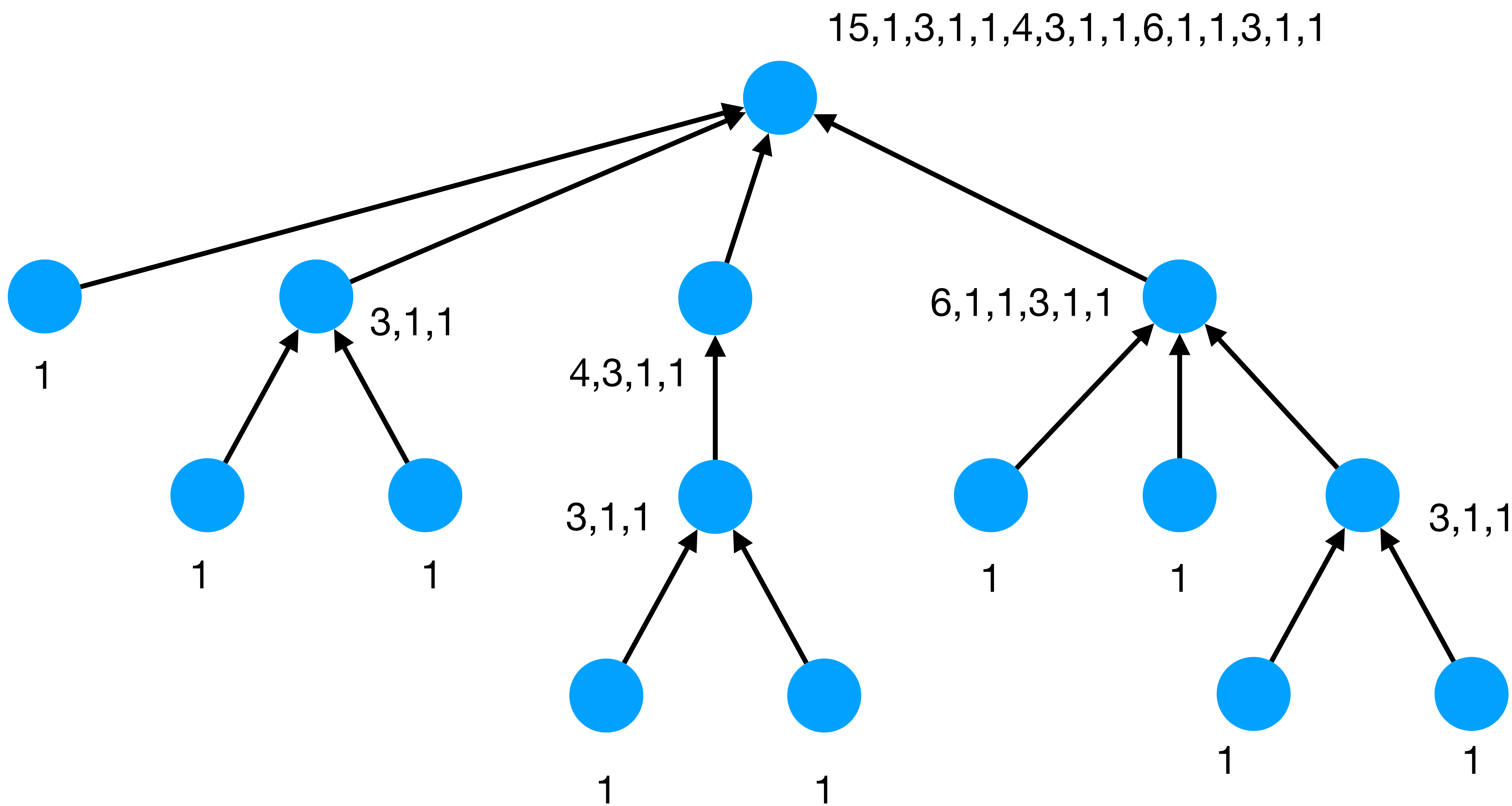
Tree canonisation

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**

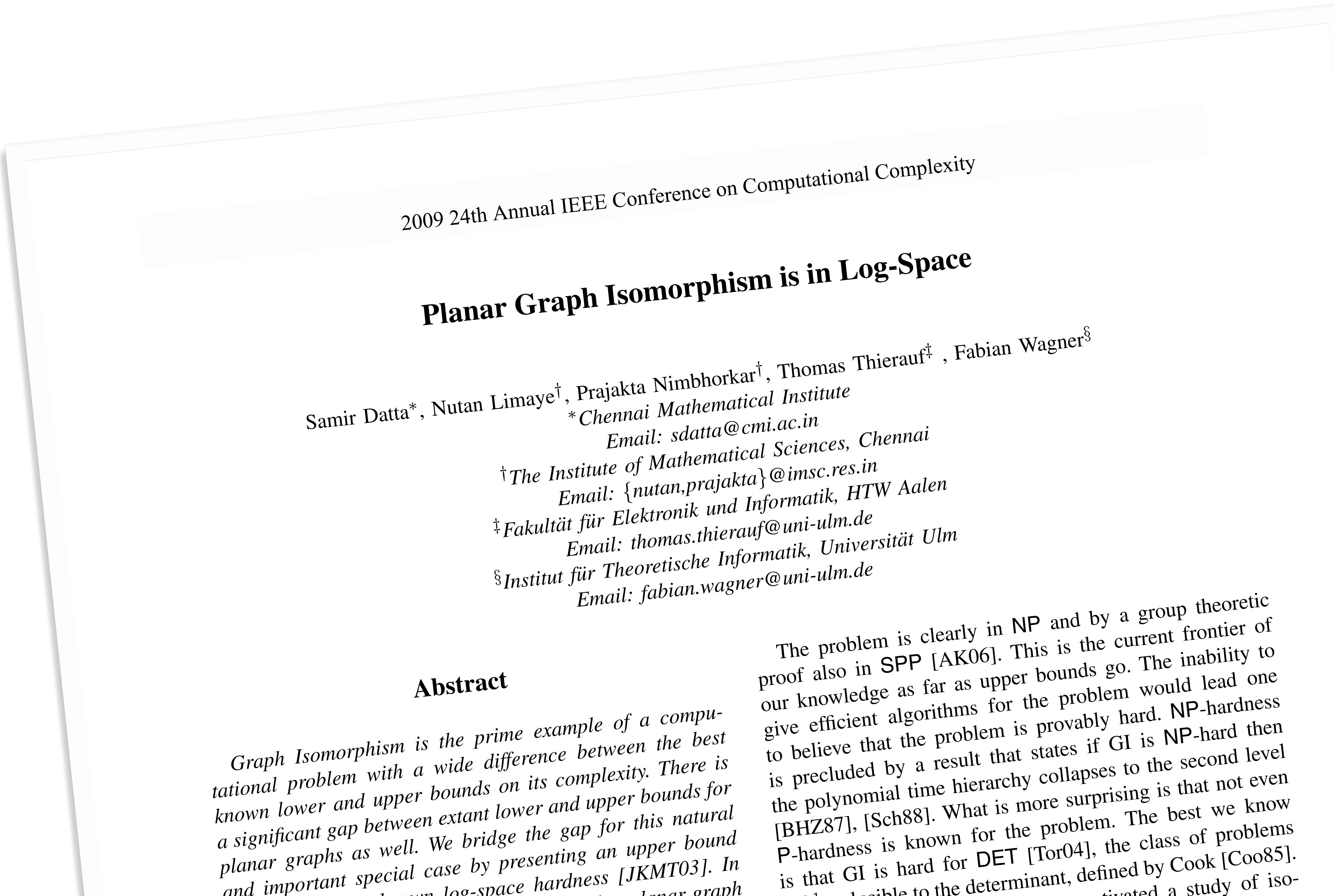
General dynamical system isomorphism

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 set of dynamical systems is different from a more general set of graphs (nondeterministic dynamical systems), where the isomorphism problem is presumably hard

Isomorphism of dynamical systems

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

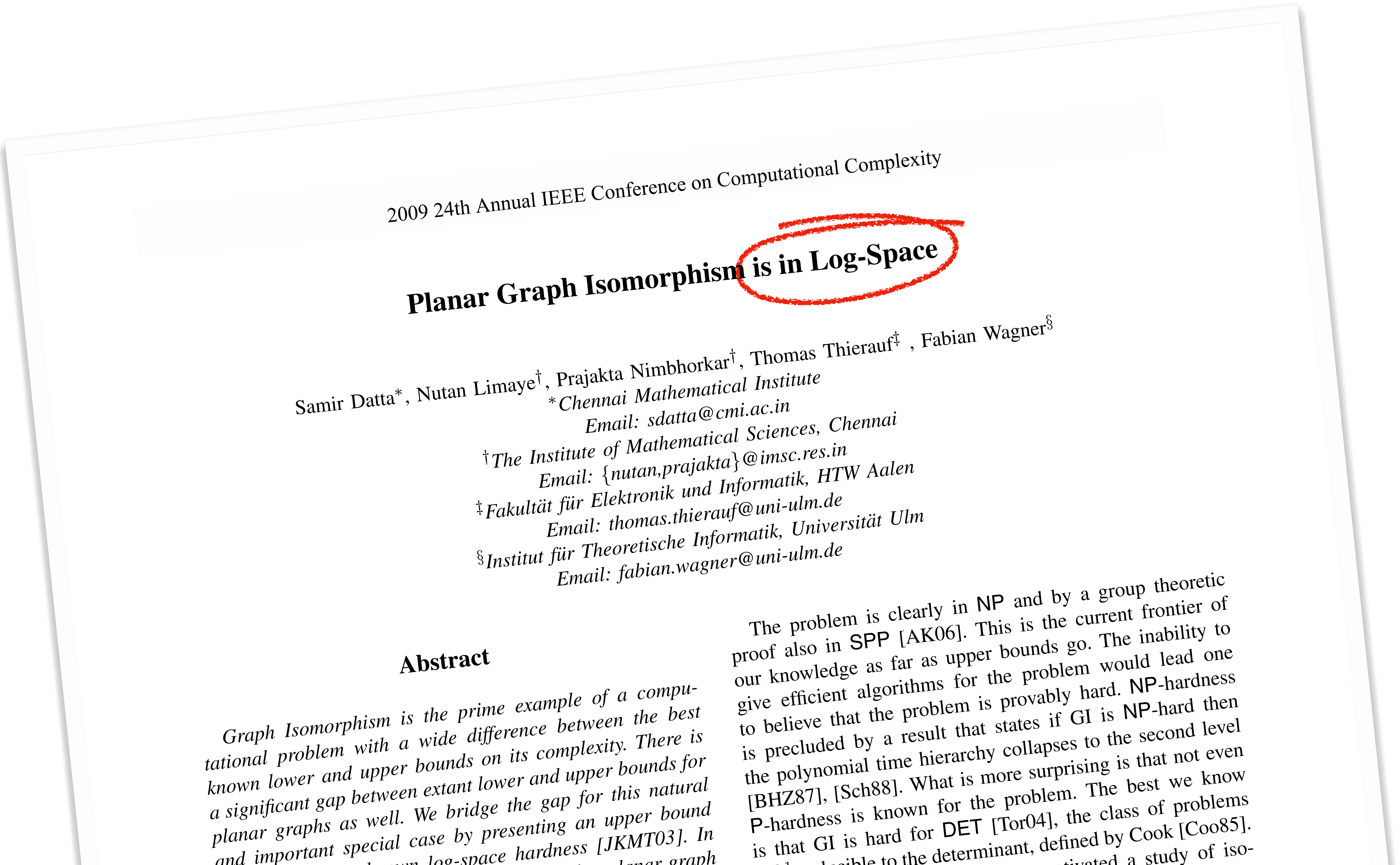
Abstract

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

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

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

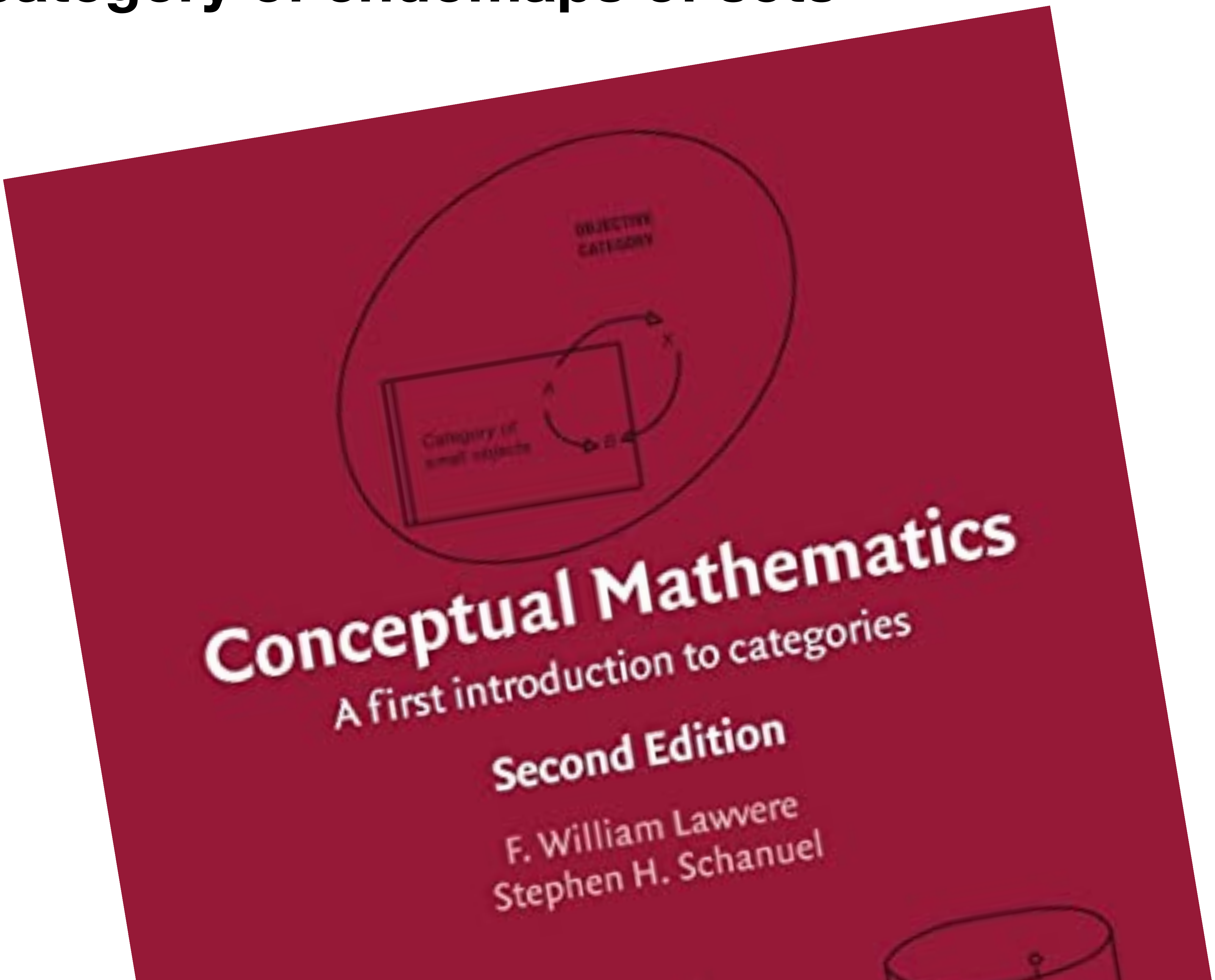
Graph Isomorphism is the prime example of a computational problem with a wide difference between the best known lower and upper bounds on its complexity. There is a significant gap between extant lower and upper bounds for planar graphs as well. We bridge the gap for this natural and important special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

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 … defined by Cook [Coo85].

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 $\varphi : A \to B$ which commutes with f and g

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\varphi"'] & A \arrow[d, "\varphi"] \\ B \arrow[r, "g"'] & B \end{tikzcd}$$

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

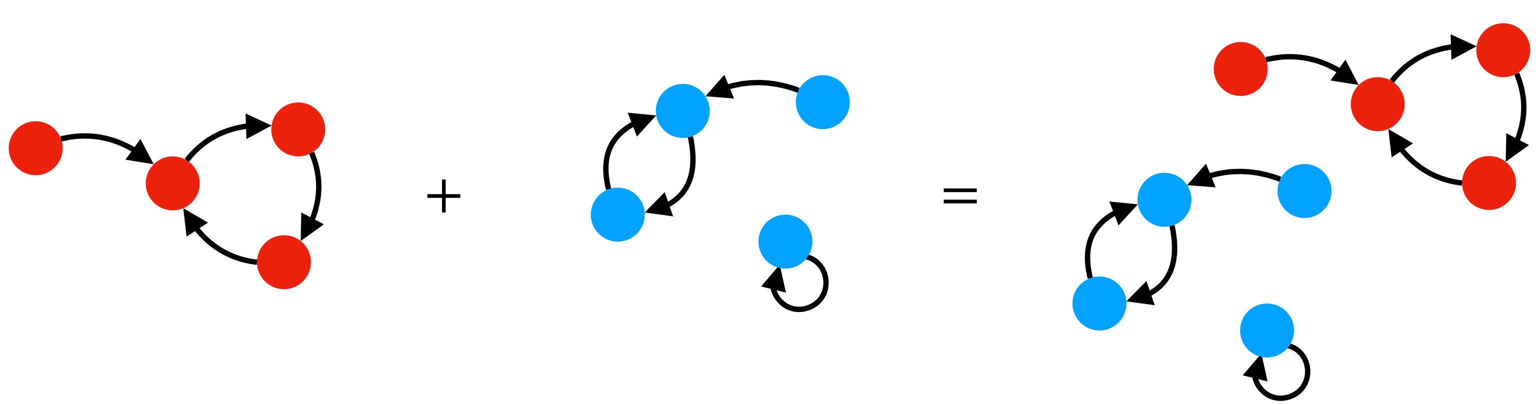
Necessary but not that interesting

- 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 $\mathbf{0} = (\varnothing, \varnothing)$



The category $\mathbf{D}$ admits products

Now we're talking!

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

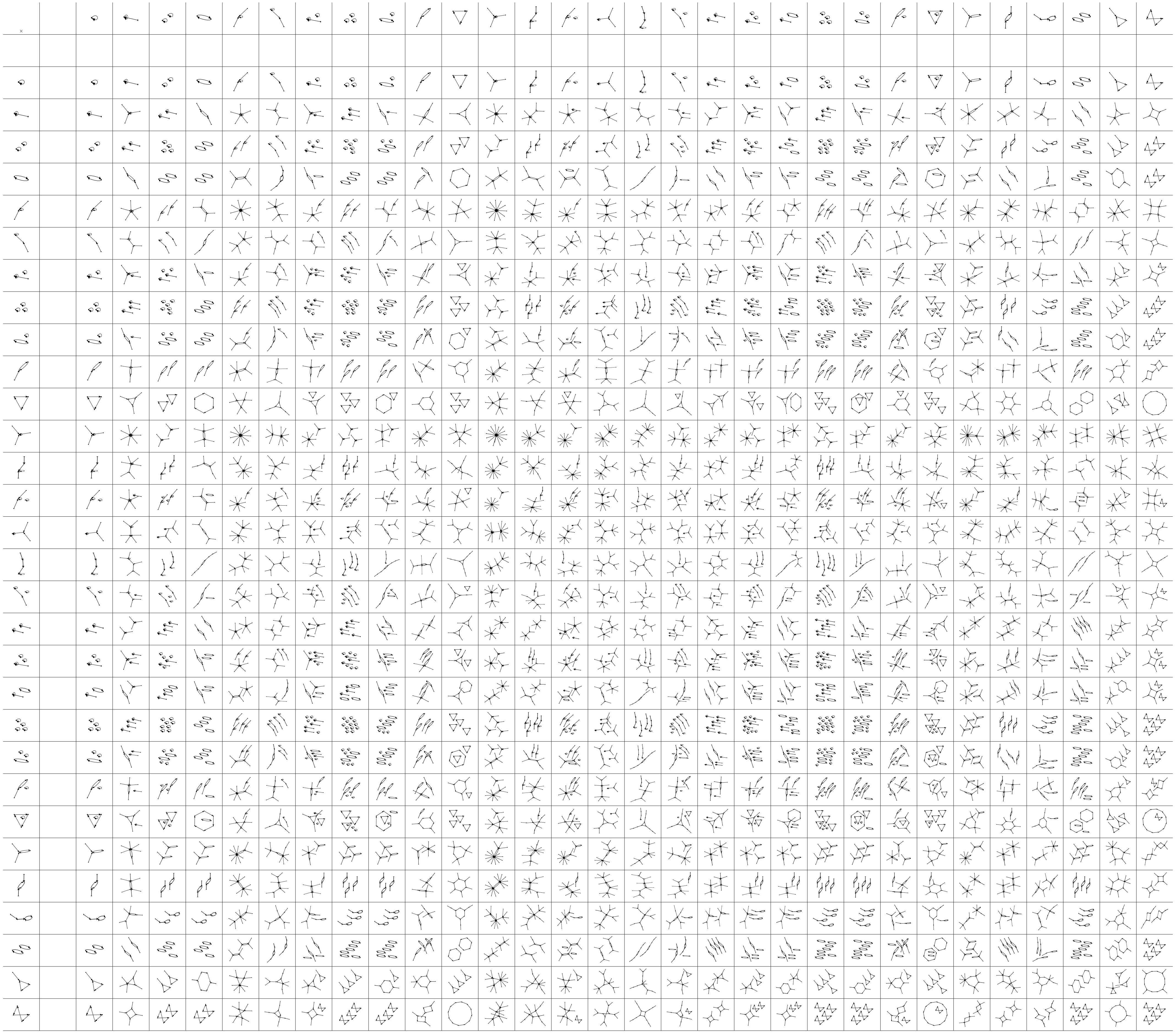
$$(A, f) \times (B, g) = (A \times B, f \times g)$$

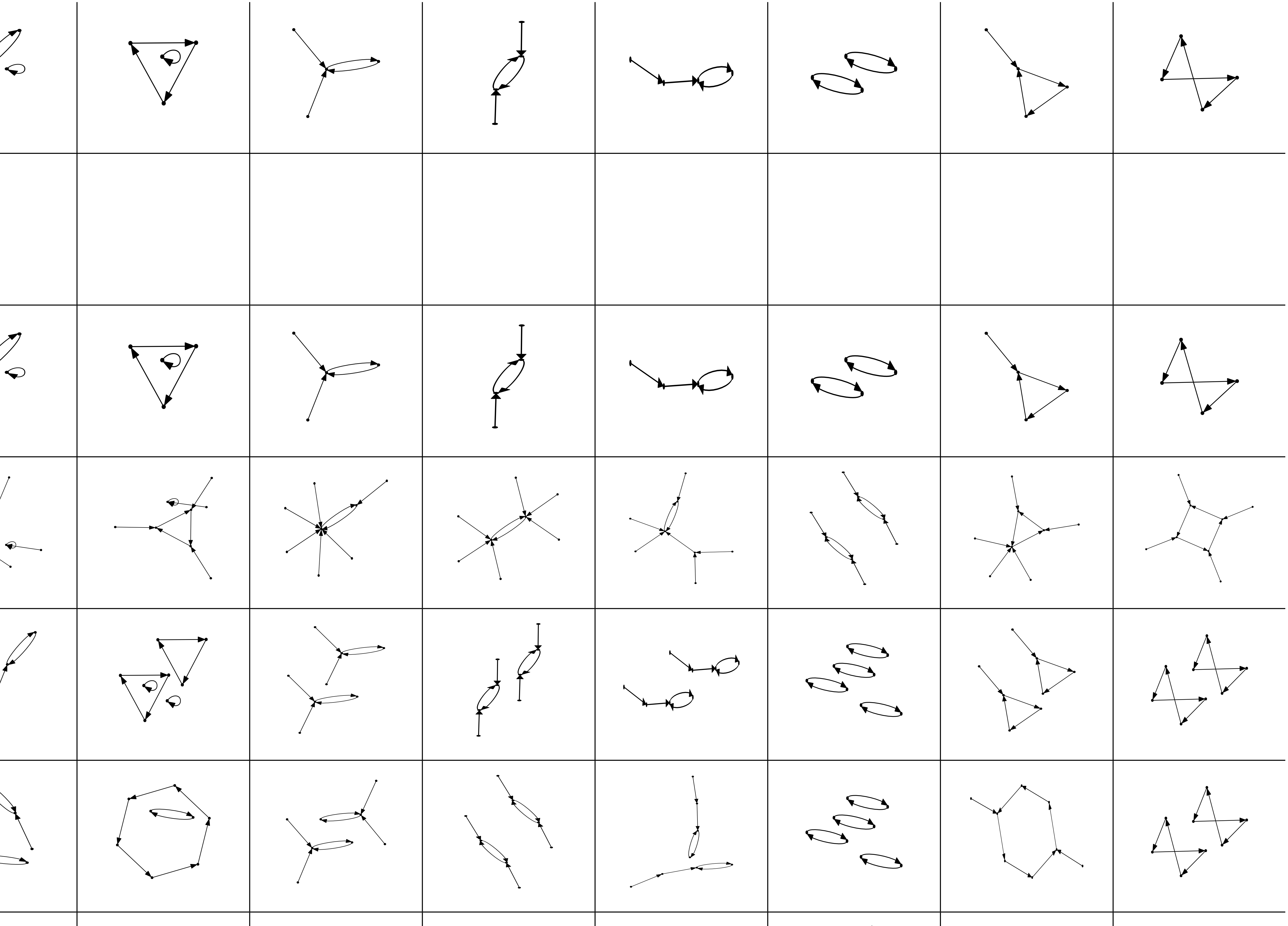
$$\text{with } (f \times g)(a, b) = (f(a), g(b))$$

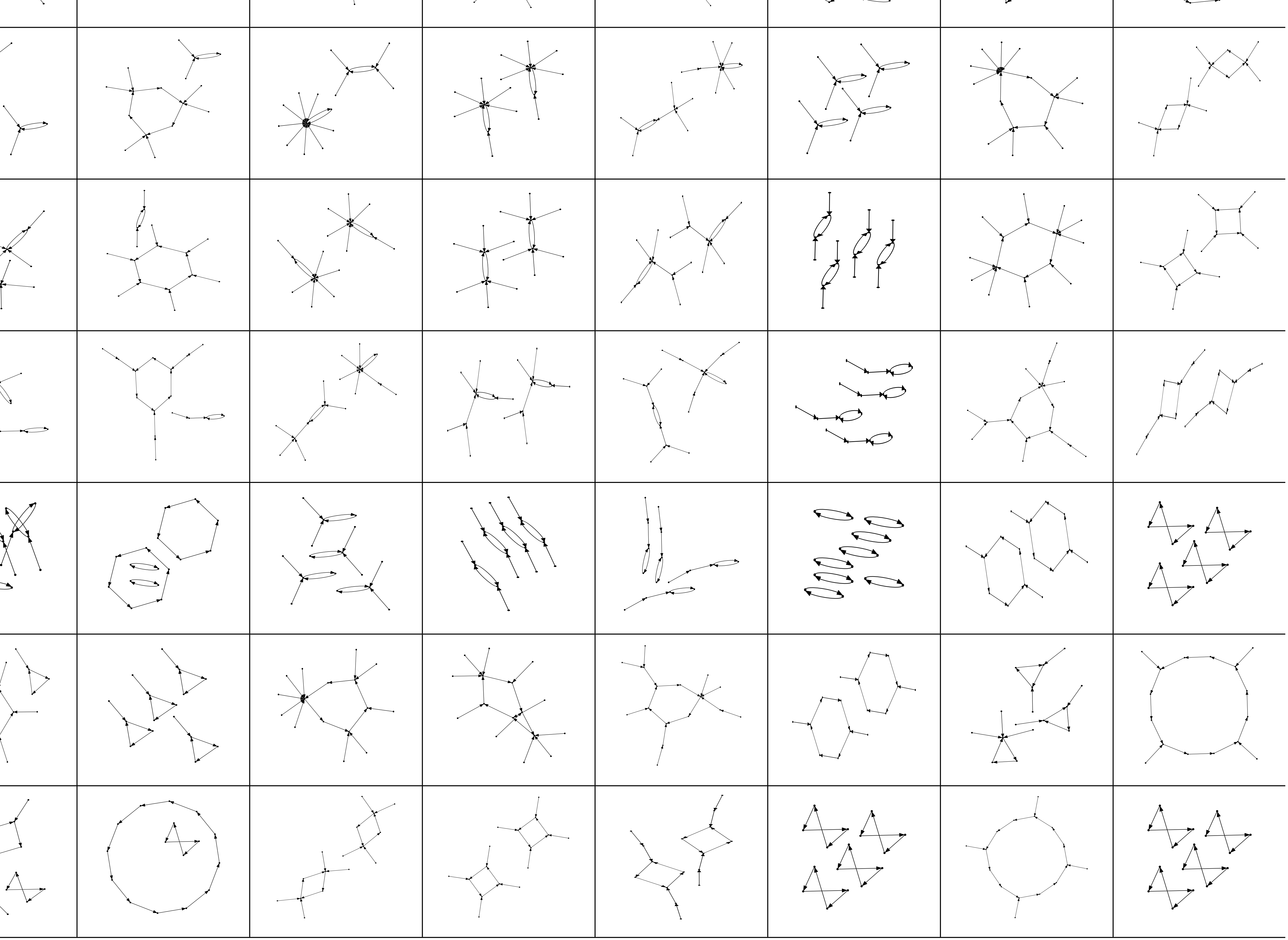
- This represents the synchronous execution of A and B

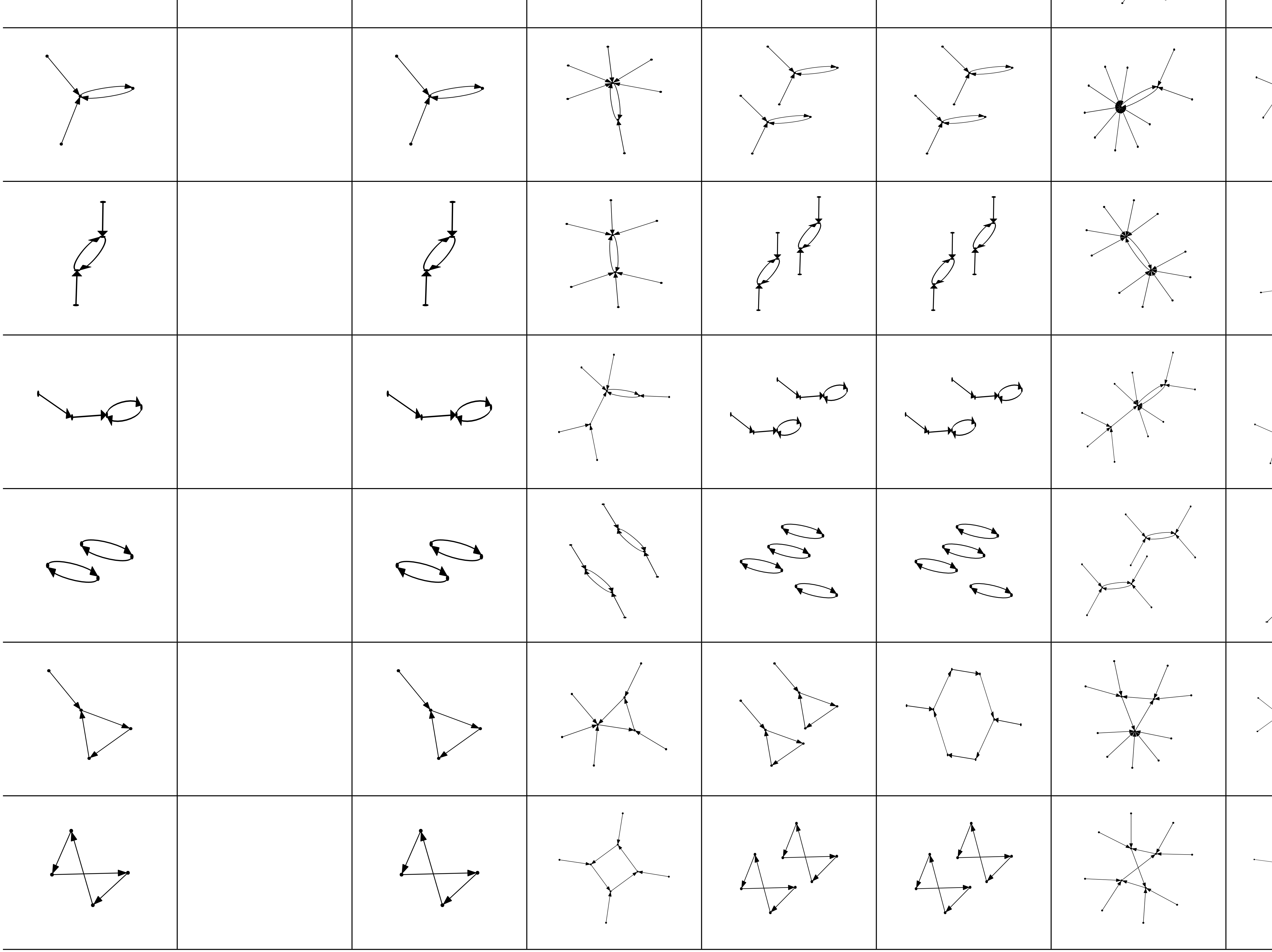
- The identity is the singleton system $\mathbf{1} = (\{0\}, \mathrm{id})$

Introducing: the multiplication table, poster-size

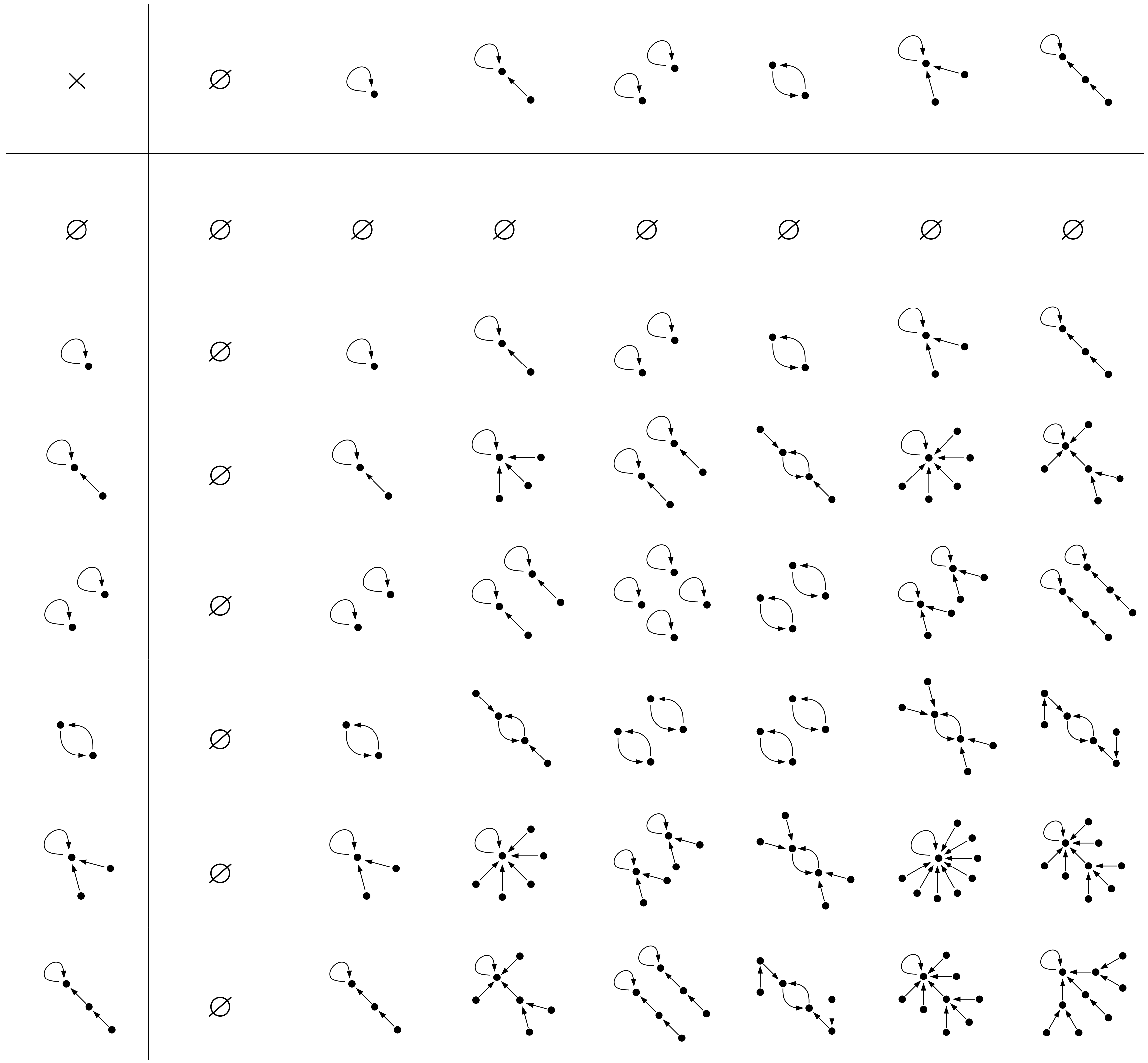






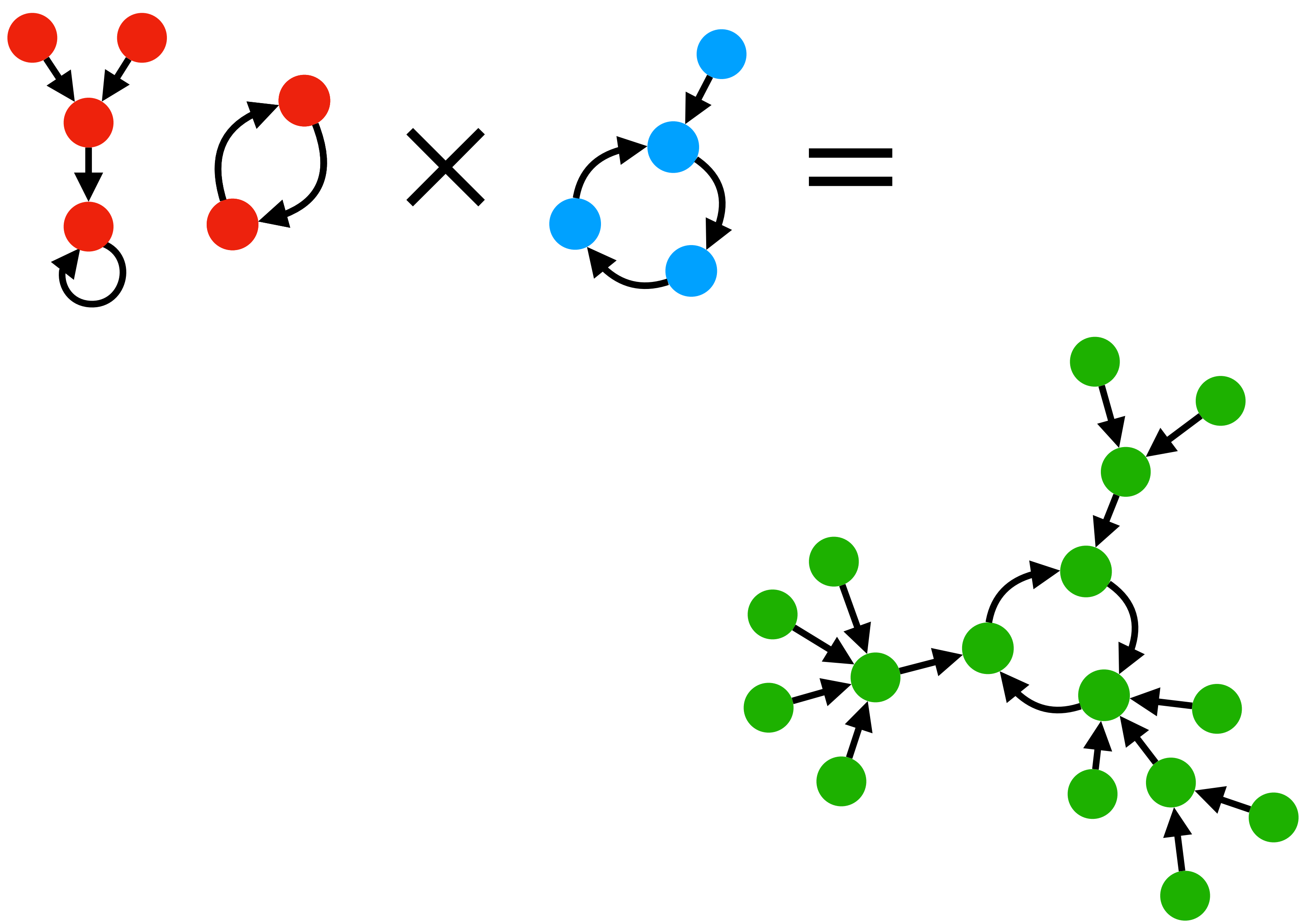


Prettier version



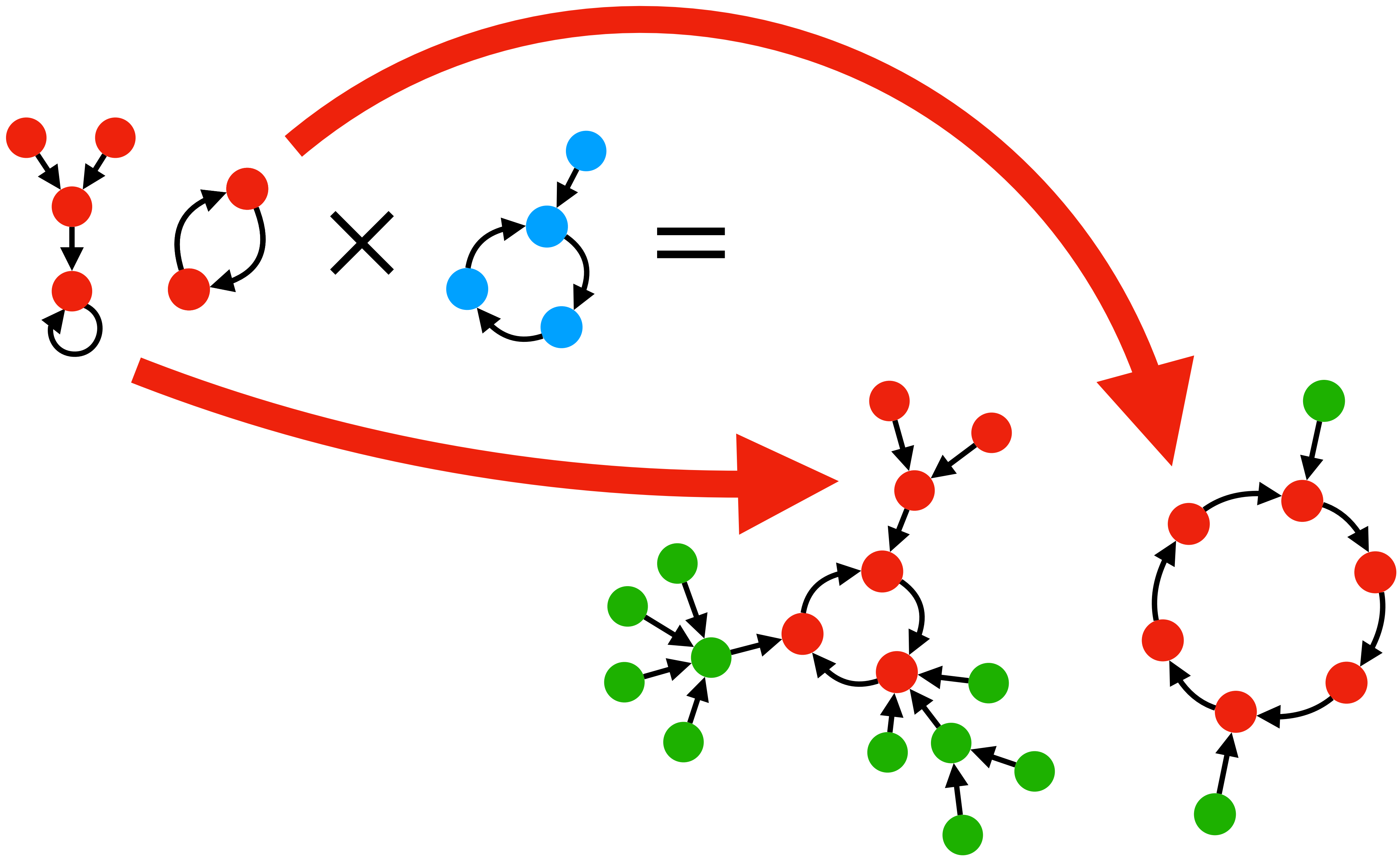
Products “preserve” behaviours

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



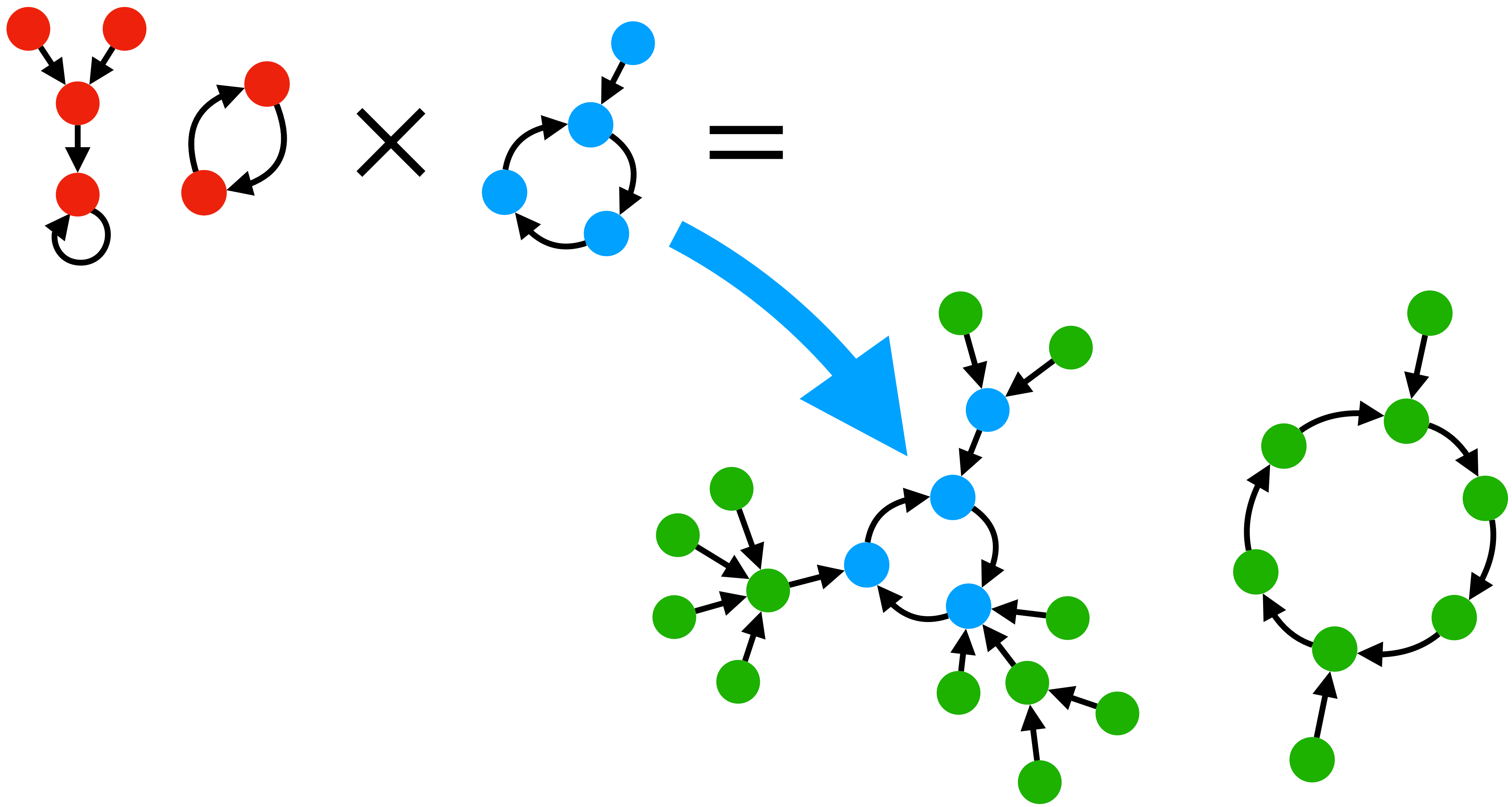
Products “preserve” behaviours

A **is a minor of** $A \times B$ **for** $B \neq \varnothing$



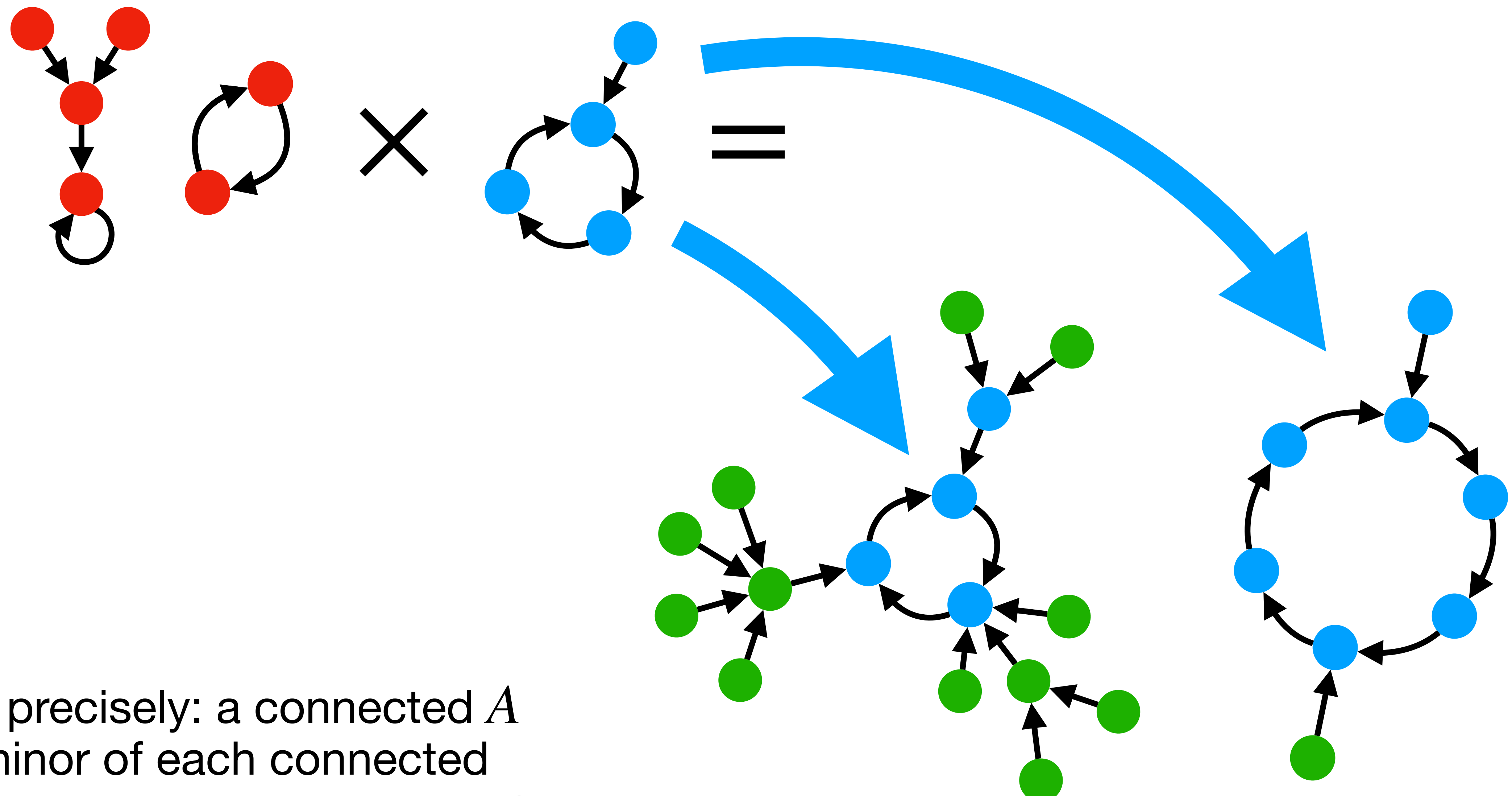
Products "preserve" behaviours

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



Products “preserve” behaviours

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



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

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 $\mathbf{1} = (\{0\}, \mathrm{id})$ in any category where it exists; so, it's a commutative monoid

- Sum is (modulo isomorphism) commutative, associative and has identity $\mathbf{0} = (\varnothing, \varnothing)$ 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

$+$ and $\times$ behave as with nonnegative integers (a commutative semiring)

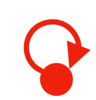
$+$ and $\times$ behave as with nonnegative integers (a commutative semiring)

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

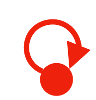
$+$ and $\times$ behave as with nonnegative integers (a commutative semiring)

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

$+$ and $\times$ behave as with nonnegative integers (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  $\times X = X$

$+$ and $\times$ behave as with nonnegative integers (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: $X \times (Y + Z) = X \times Y + X \times Z$

$+$ and $\times$ behave as with nonnegative integers (a commutative semiring)

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

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

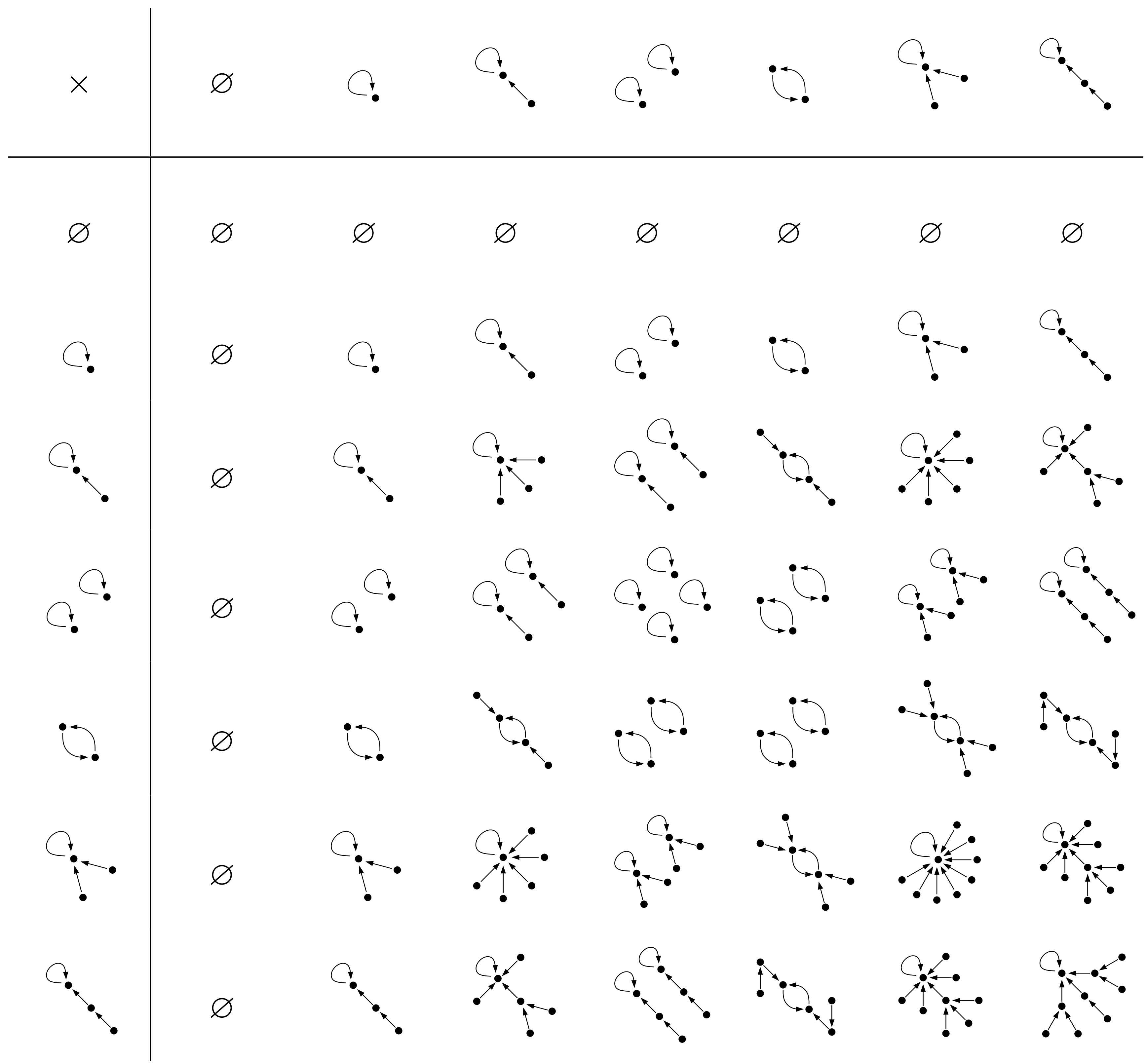
- Neutral elements: $\varnothing + X = X$ and $\times X = X$

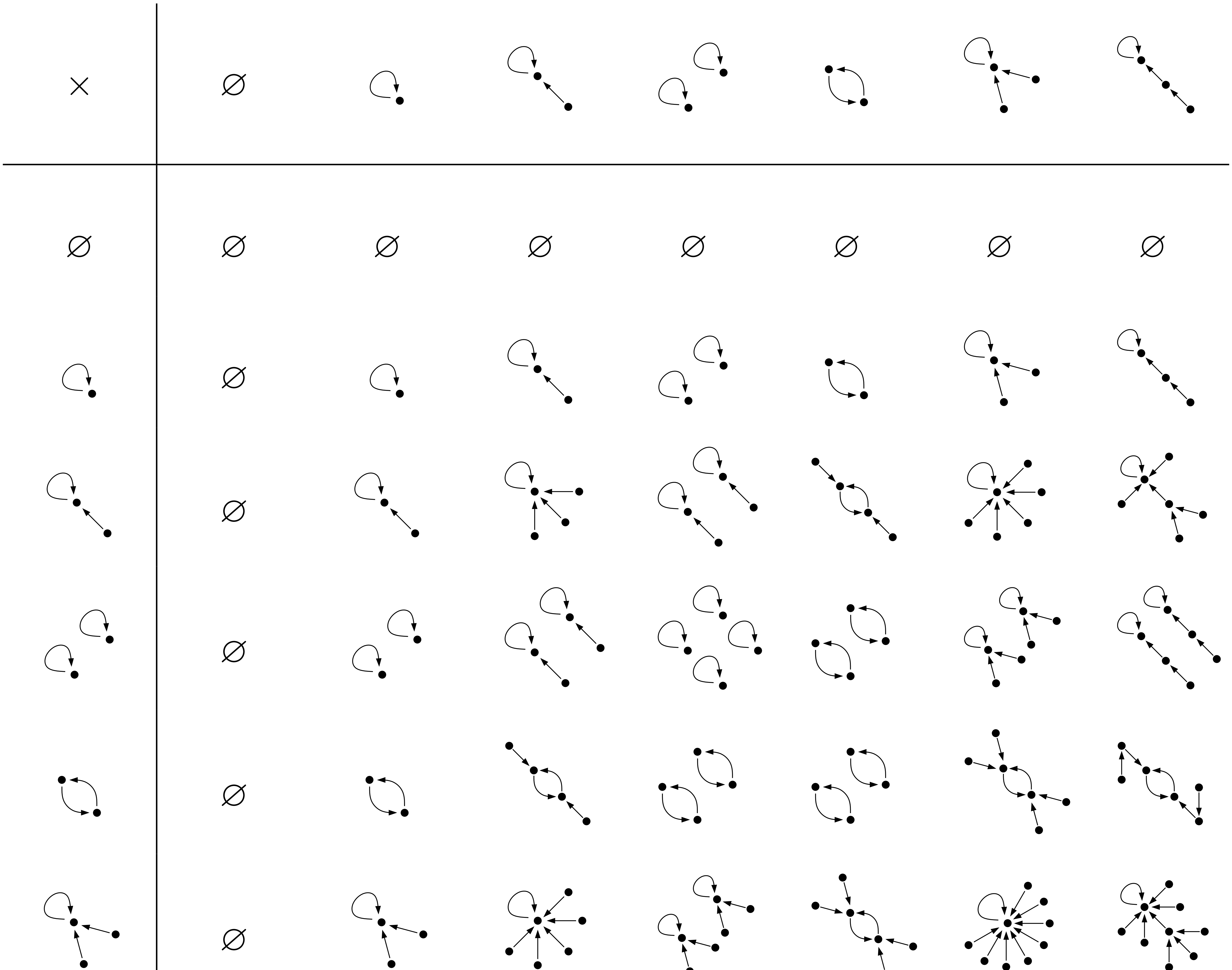
- Distributive: $X \times (Y + Z) = X \times Y + X \times Z$

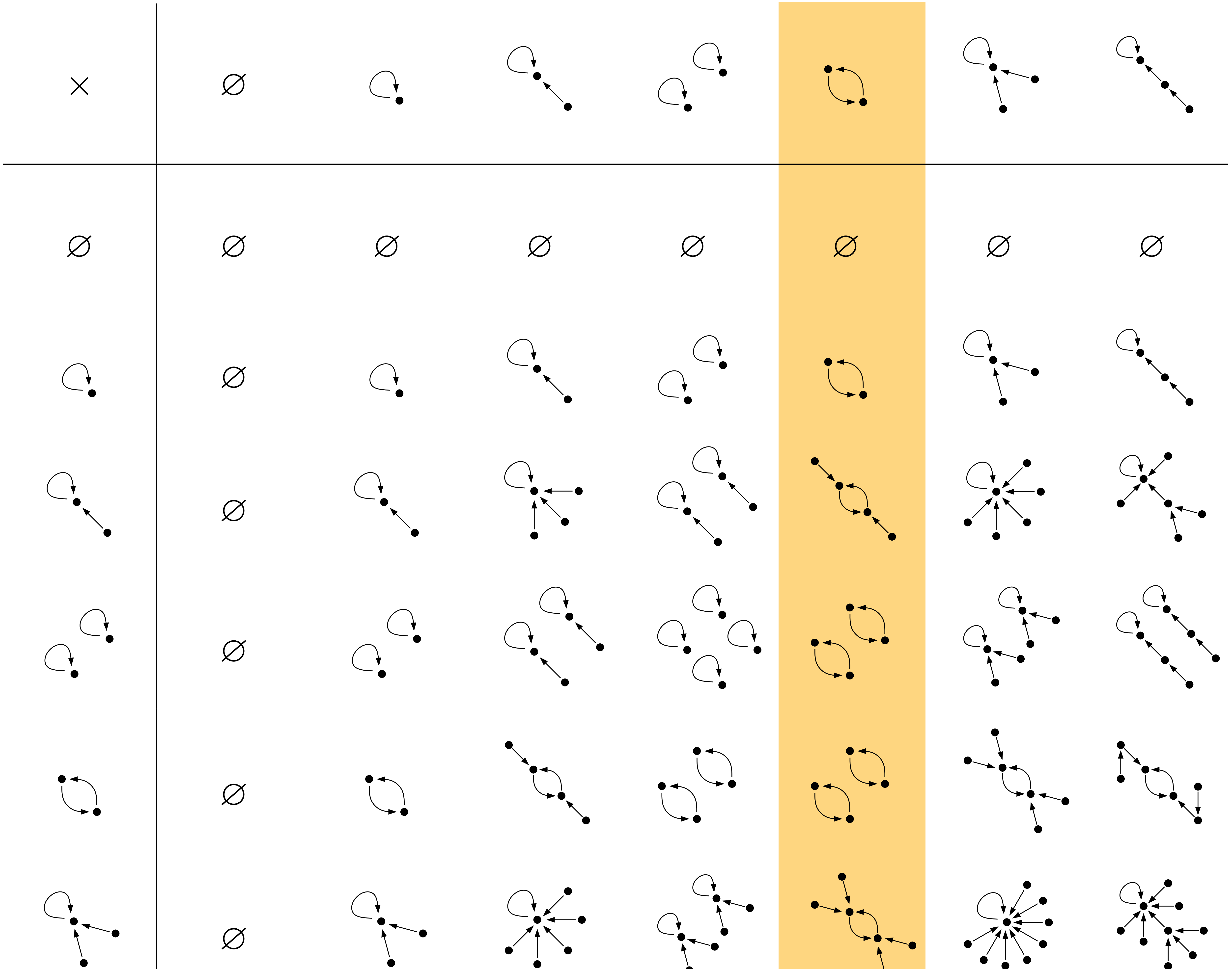
- Multiplication by zero: $\varnothing \times X = \varnothing$

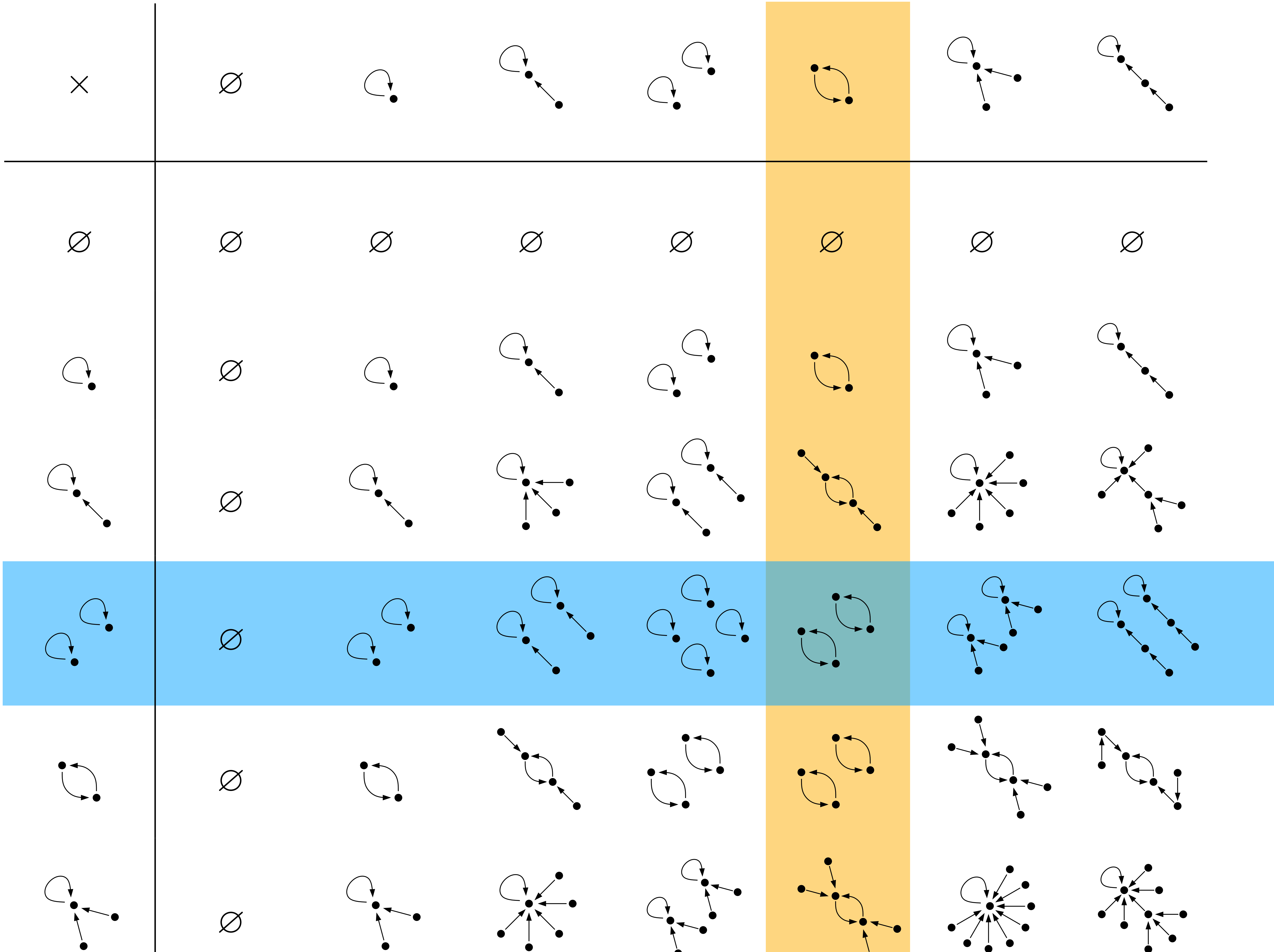
No unique factorisation 😭

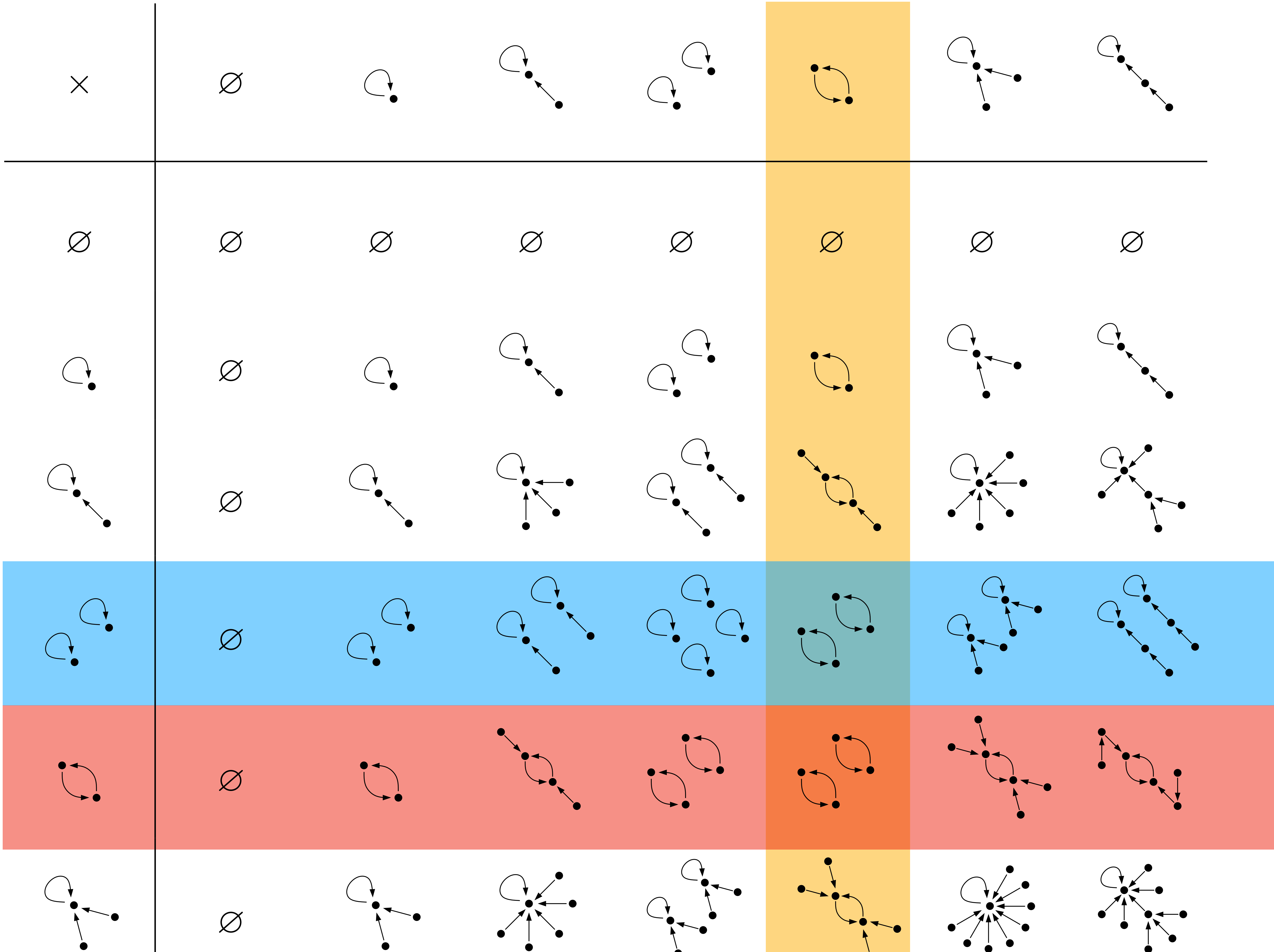
Multiplication table





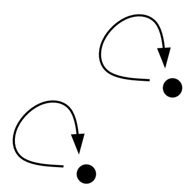
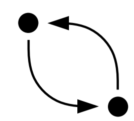
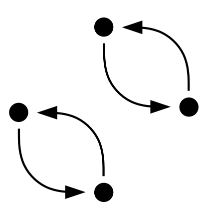


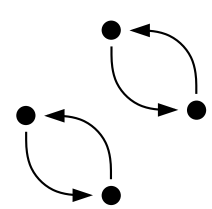
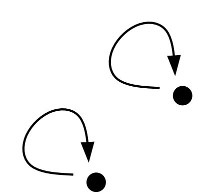
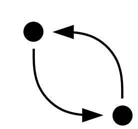


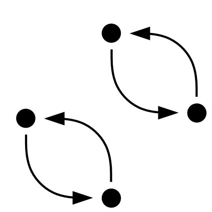
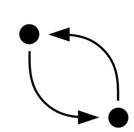
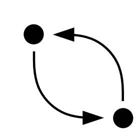


No unique factorisation

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

 $=$  $\times$ 

 $=$  $\times$ 

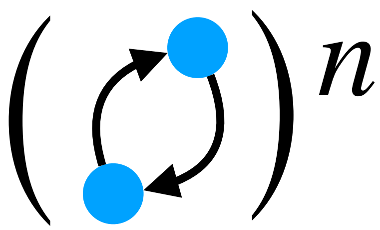
Systems with arbitrarily many factorisations

Theorem

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

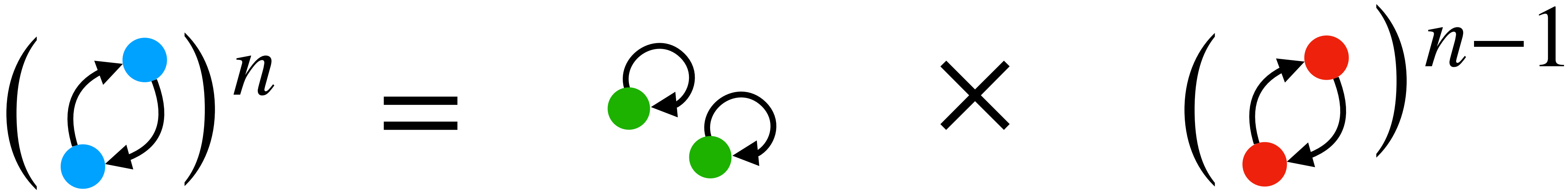
Theorem

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



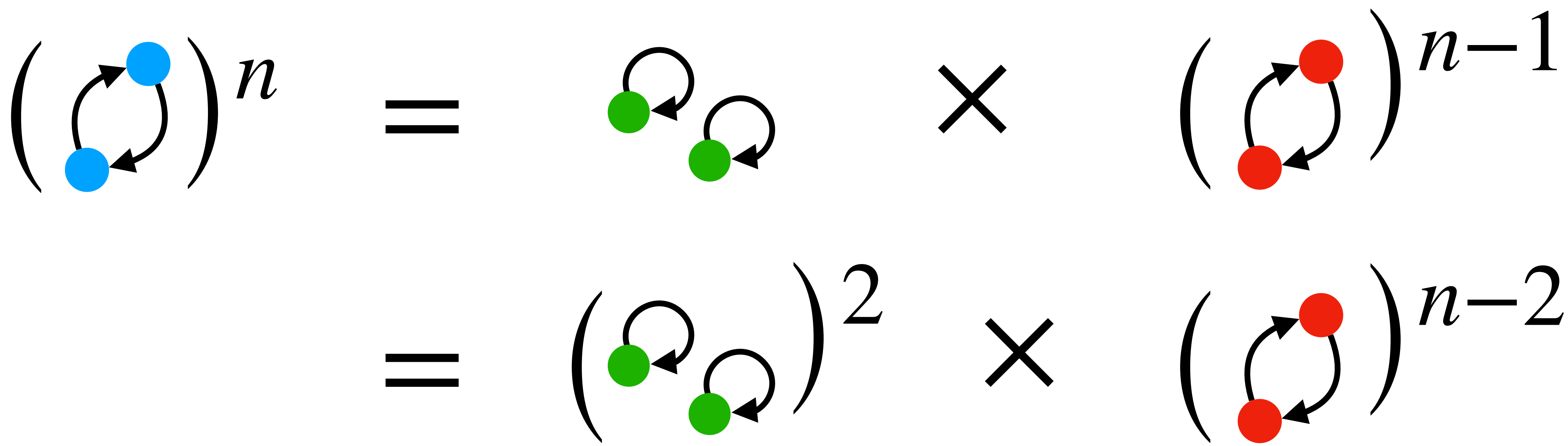
Theorem

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



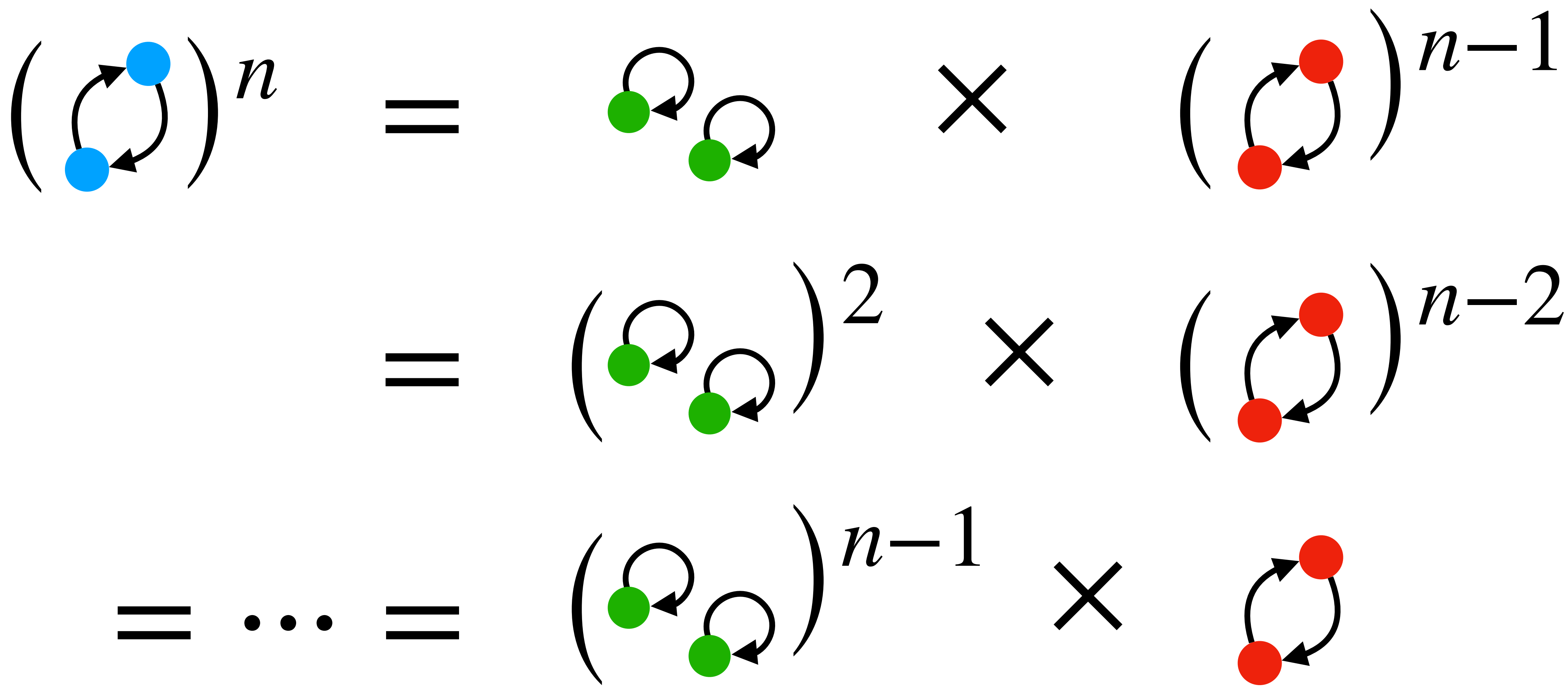
Theorem

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 $\varphi \colon \mathbb{N} \to \mathbf{D}$

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

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

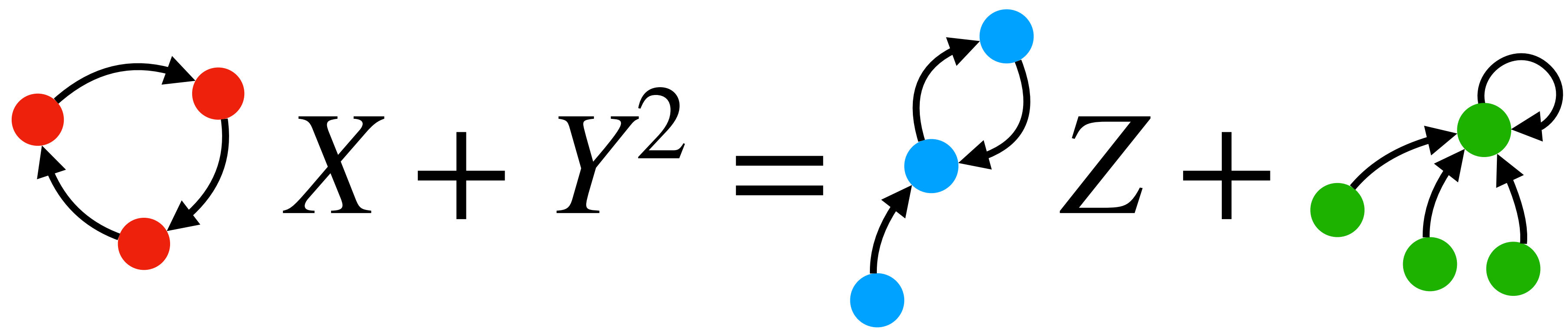
- So $\mathbf{D}$ contains a isomorphic copy of $\mathbb{N}$

Polynomial equations

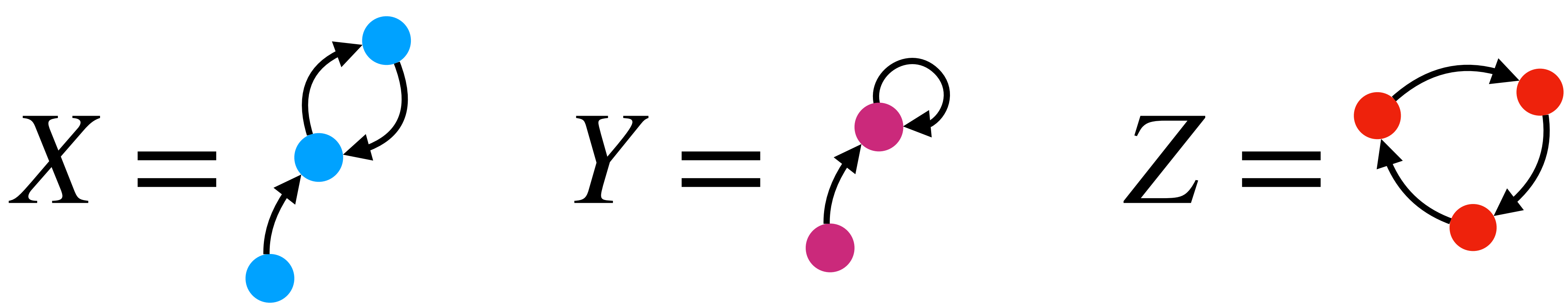
Polynomial equations over $\mathbb{D}$

For the analysis of complex systems

- Consider the equation



- There is least one solution



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 $p, q \in \mathbf{D}[\overrightarrow{X}]$

Undecidability of polynomial equations

Undecidability of polynomial equations

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

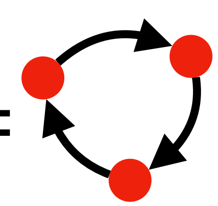
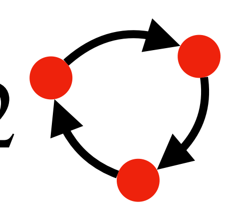
- We have showed that $\mathbb{N}$ is a subsemiring of $\mathbf{D}$
- But sometimes enlarging the solution space makes the problem actually easier: given $p, q \in \mathbb{N}[\overrightarrow{X}]$
  - Finding if $p(\overrightarrow{X}) = q(\overrightarrow{X})$ has solution in $\mathbb{N}$ is undecidable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

$X =$  $Y = 2$ 

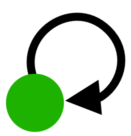
- But, of course, it also has the natural solution $X' = 3$, $Y' = 6$

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

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

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

Proof

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

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

Proof

The coefficients are natural

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

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

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

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**

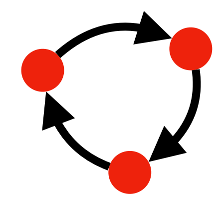
$$p(|A_1|, \ldots, |A_k|) = q(|A_1|, \ldots, |A_k|)$$

Equations with non-natural coefficients

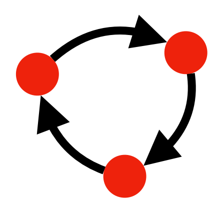
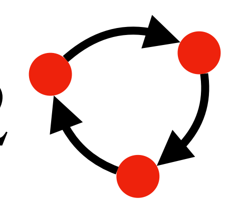
Equations without natural solutions

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 logspace
- 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)

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 $\mathbf{D}$ and more specifically over $\mathbb{N}$, 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 **NP**-complete

$\mathbf{D}$ 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

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 $p_i \in \mathbb{N}[\overrightarrow{X}]$
- Take any n cycles of distinct prime length $C_1, \ldots C_n \in \mathbf{D}$
- Then the equation $C_1 p_1(\overrightarrow{X}) + \cdots + C_n p_n(\overrightarrow{X}) = C_1 + \cdots + C_n$ is a linear equation over $\mathbf{D}[\overrightarrow{X}]$ having the same solutions as the original system
- This means that the problem is $\mathbf{NP}$-complete even for linear equations with constant right-hand side over cycles!

Irreducible systems

Most dynamical systems are irreducible

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

- Formally:

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

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

Prime system

Identifying basic building blocks

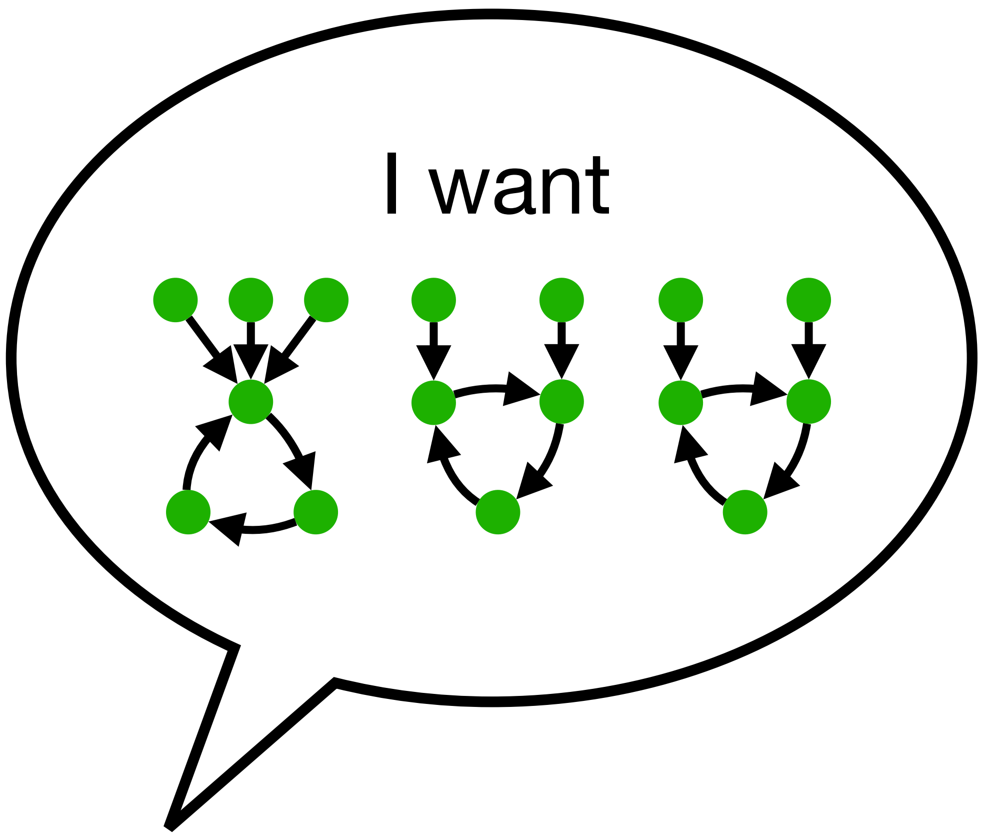
Scenario

DynaSys Inc.

Scenario

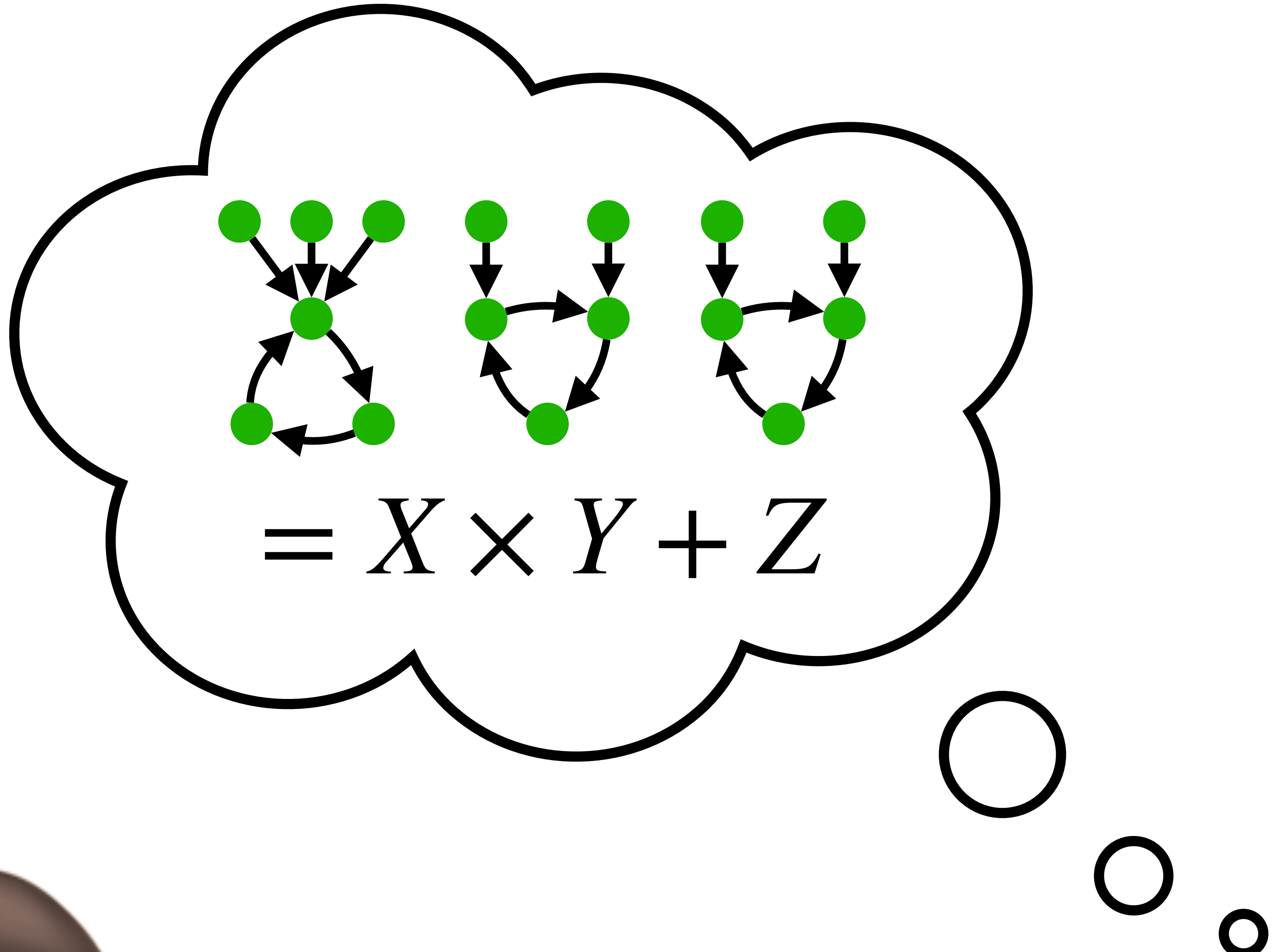
DynaSys Inc.

Scenario



DynaSys Inc.

Scenario



DynaSys Inc.

Scenario

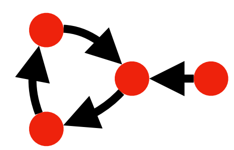
$$X =$$

$$Y =$$

$$Z =$$

DynaSys Inc.

Scenario

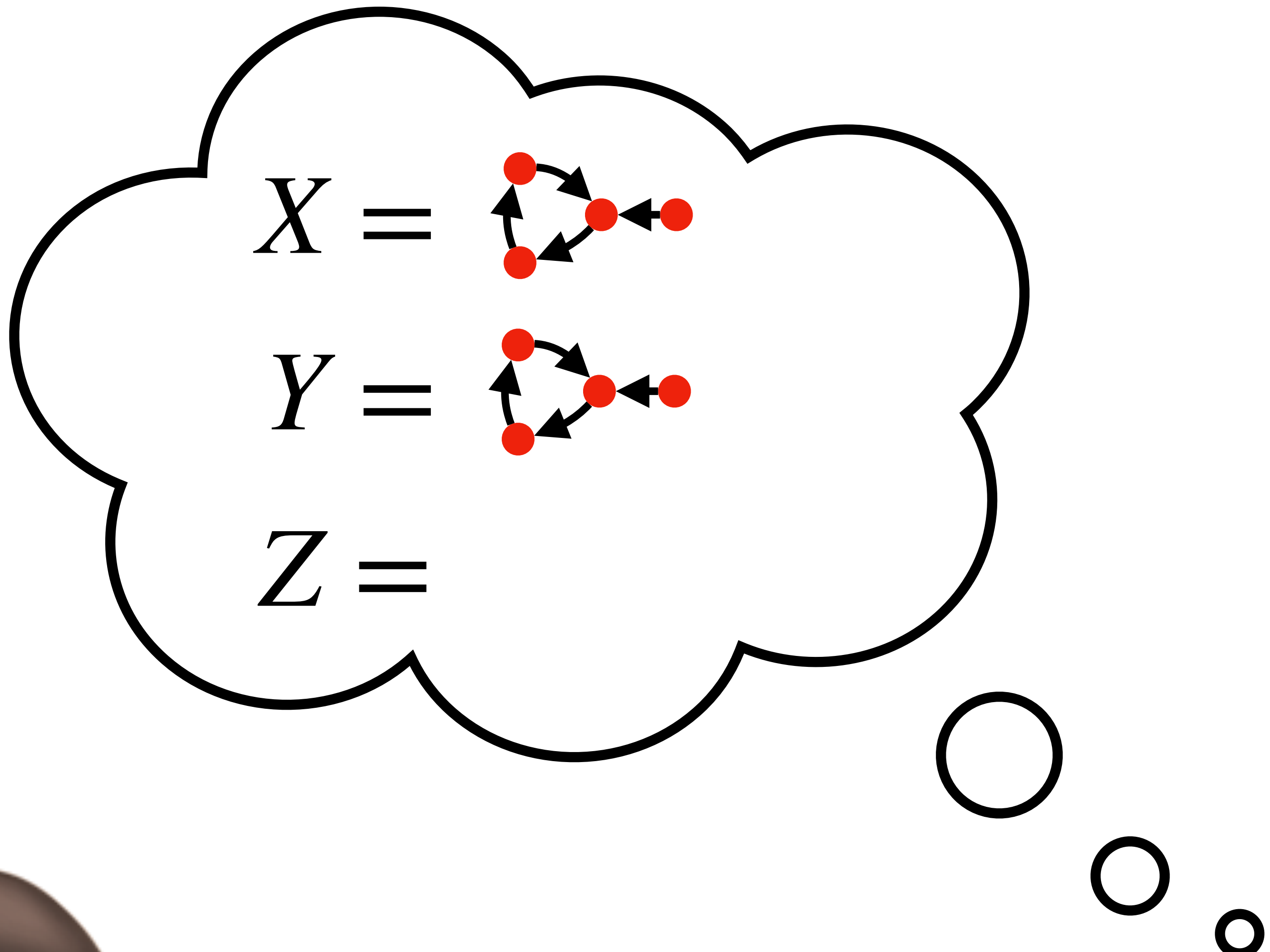
$X =$ 

$Y =$

$Z =$

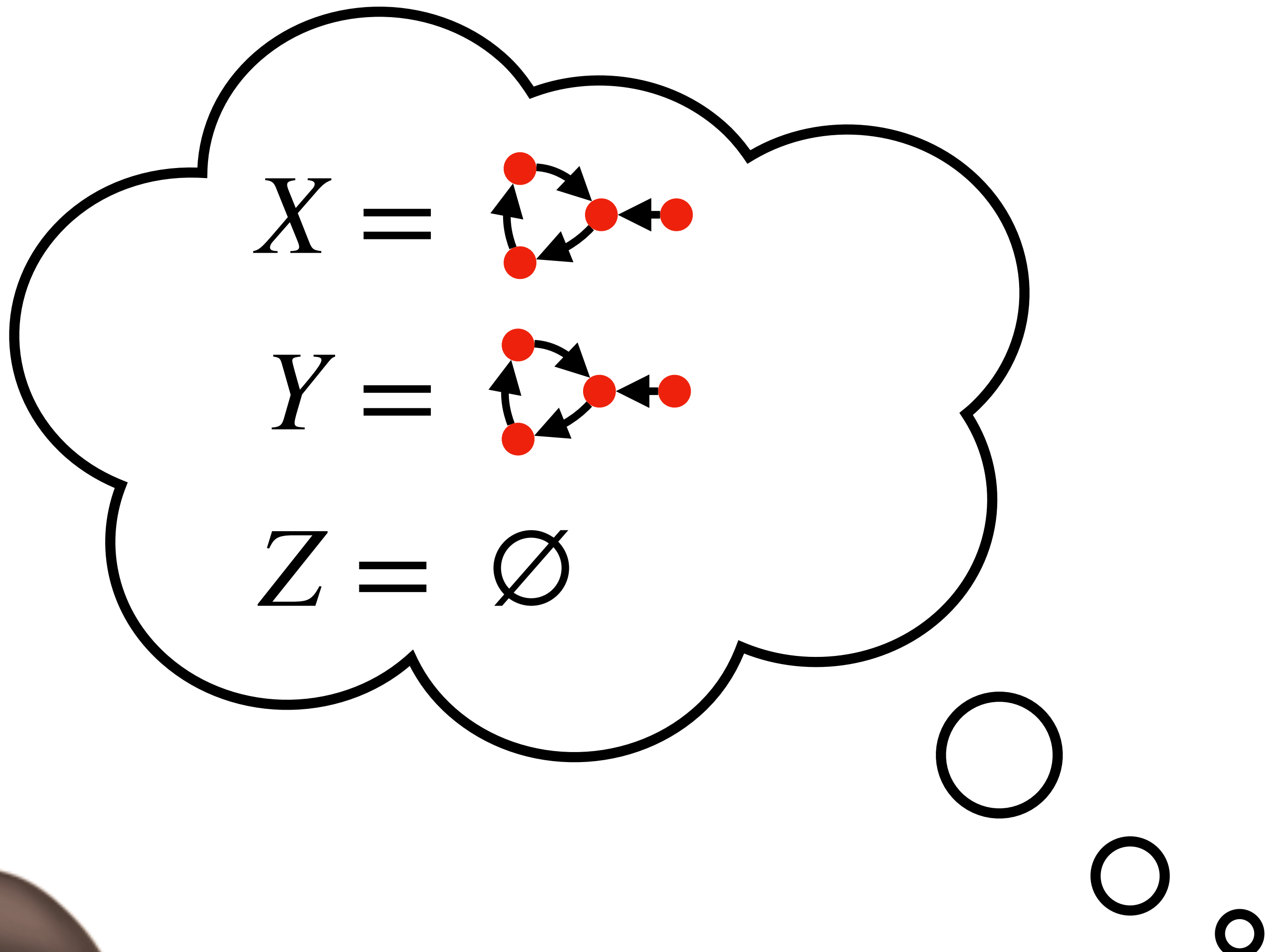
DynaSys Inc.

Scenario



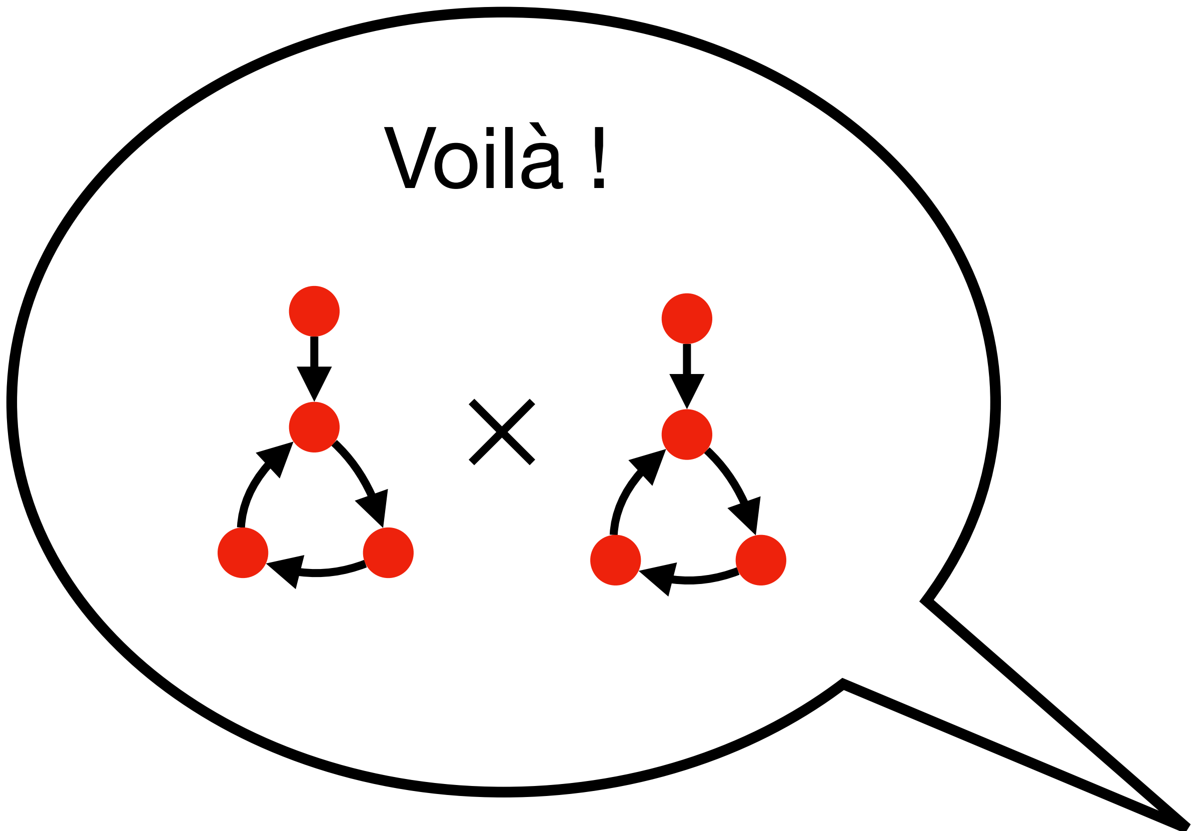
DynaSys Inc.

Scenario



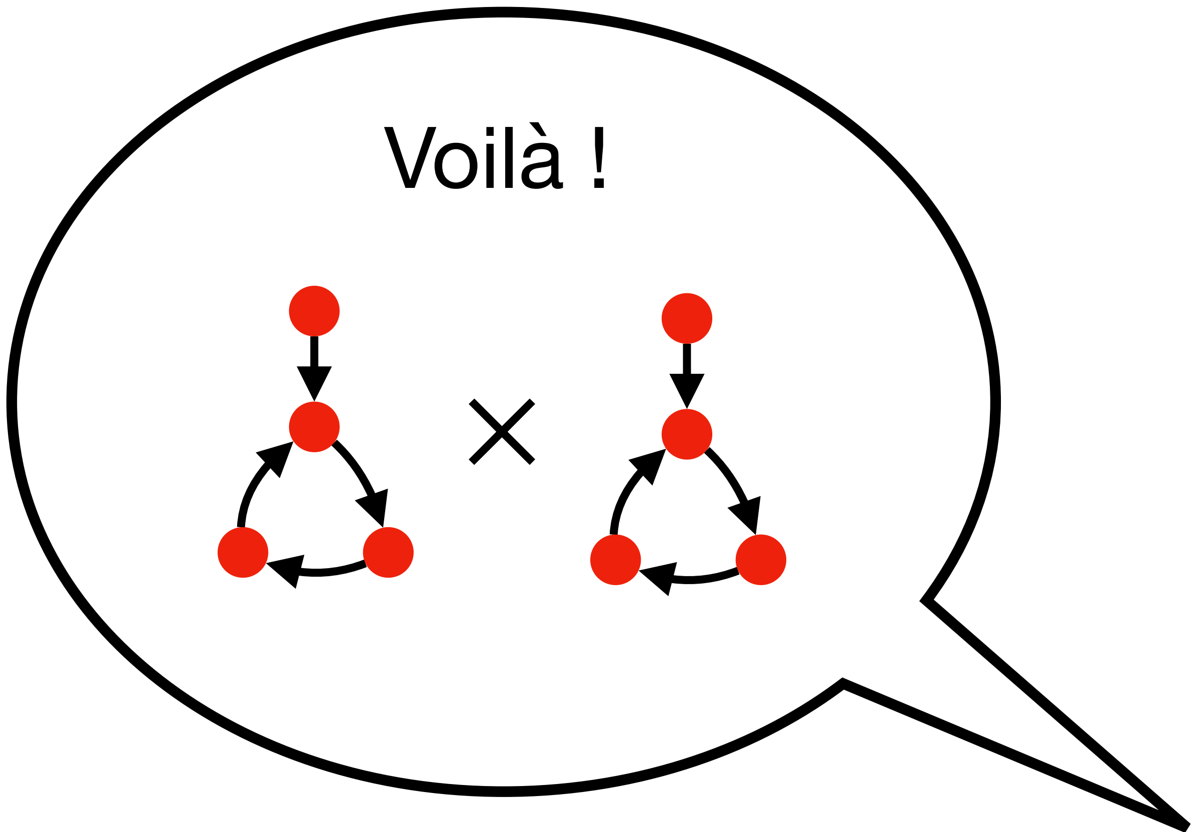
DynaSys Inc.

Scenario



DynaSys Inc.

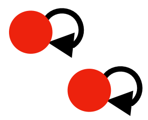
Scenario

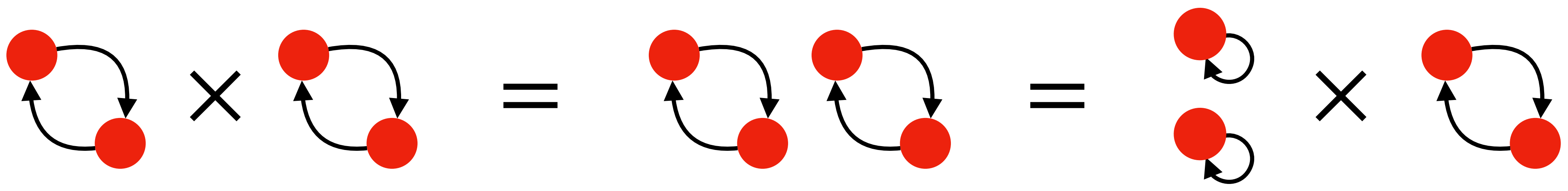


DynaSys Inc.

Prime system

$P \neq 0,1$ **is prime iff** $P \mid AB$ **implies** $P \mid A$ **or** $P \mid B$

- If a prime P appears in a factorisation into irreducibles of a system, then it appears in all factorisations
- On the contrary, non-prime systems can sometimes be replaced
- So prime systems are irreplaceable building blocks
- We don't know if prime systems exist yet!
- But we know several nonprimes, for instance 



No natural number is prime

Not even prime naturals!

- Cycles of length n sometimes behave like n fixed points

$$C_n \times C_n = n \times C_n$$

- This is based on the folklore (?) result that

$$C_m \times C_n = \gcd(m, n) \times C_{\operatorname{lcm}(m,n)}$$

More interesting classes of nonprimes

Work by Johan Couturier

- If A is disconnected, then A is not prime
- If A is connected but of period > 1, then A is not prime
- If A is connected of period 1, but

$$\gcd(A) = \gcd\{\text{\#preimages of } a : a \in A\} > 1$$

then A is not prime

- In particular, systems consisting of sums of cycles (i.e., the asymptotic behaviours of any system) are nonprime

Is primality decidable?

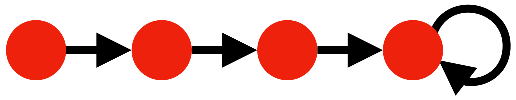
Most. Annoying. Open. Problem. Ever. 😡

- We do not know an algorithm for primality testing!
- Nonprimes are recursively enumerable
  - Enumerate systems A, B to find a counterexample to the primality of P, i.e., $P \mid AB$ but $P \nmid A$ and $P \nmid B$
  - No known way to bound the size of counterexamples
- Fun fact: if primality is undecidable, then primes do exist 😄

Open problems

Open problems

Algebraic ones

- Do prime systems exist at all? Is primality decidable?
- Is this particular guy here prime? 
- What is the complexity of deciding if $A \mid B$?
  And deciding if A is irreducible?
- Does it make any sense to adjoin the additive inverses in order to obtain a ring?
- Is it useful to find nondeterministic dynamical system (i.e., arbitrary graph) solutions to equations?
- Semirings of infinite discrete-time dynamical systems

Open problems

Solving equations

- Find larger classes of solvable equations,
  e.g., by number of variables or degree of the polynomials
- Discover classes of equations solvable efficiently
  - Probably very 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

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