Communication topologies in natural computing

Antonio E. Porreca
https://aeporreca.org

Outline

- The first and second machine classes
- Communication topologies and their role
- Membrane computing
- Complexity theory of membrane systems
- A research project

The first machine class and P

The deterministic Turing machine and all models that simulate and are simulated by it efficiently

- Random access machines with arithmetic operations + and –
- Cellular automata with finite initial configuration

Van Emde Boas: Machine models and simulations, in Van Leeuwen (ed.), Handbook of Theoretical Computer Science Volume A: Algorithms and Complexity 1–66, MIT Press, 1990

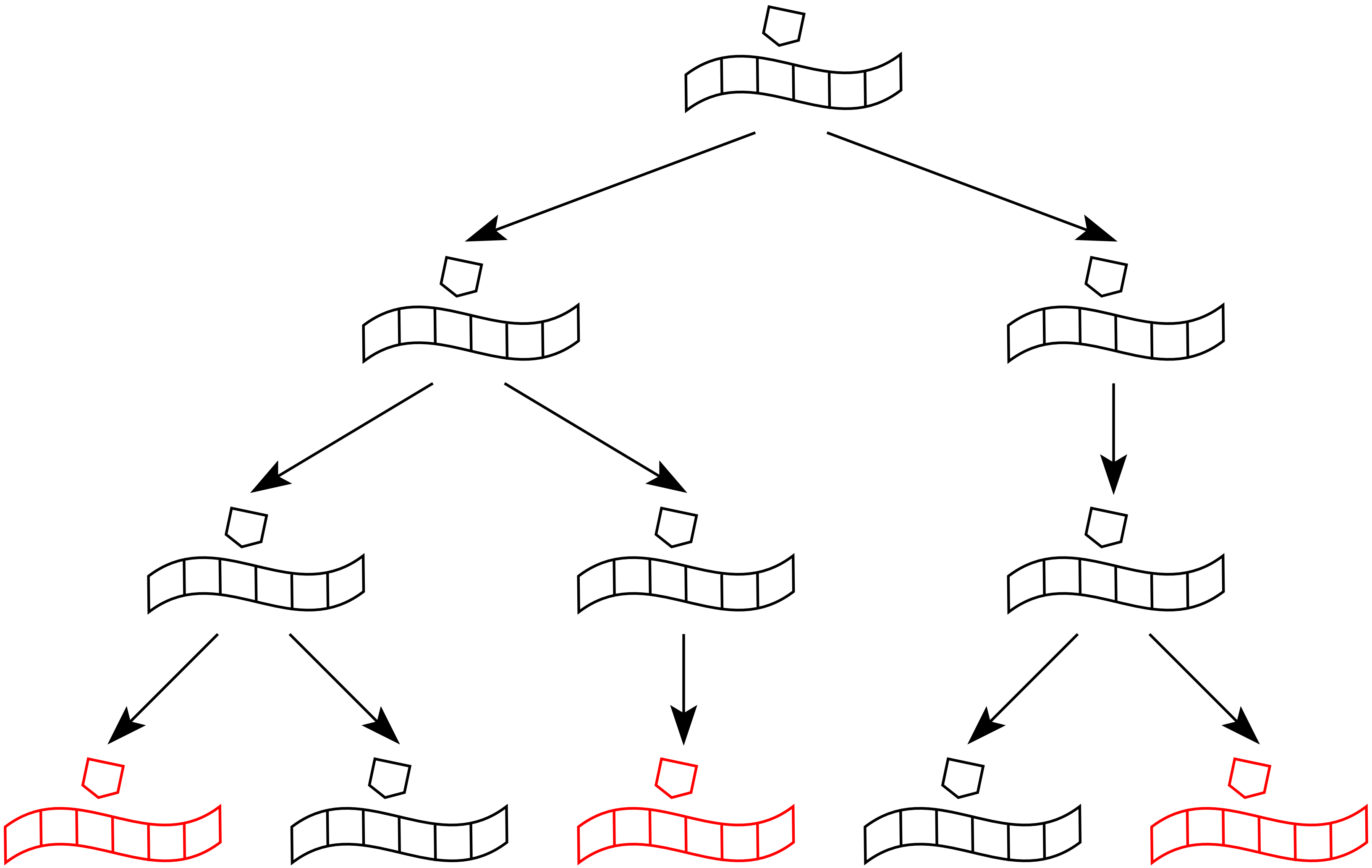
The second machine class and PSPACE

Computing models that solve in polynomial time what a Turing machine solves in polynomial space

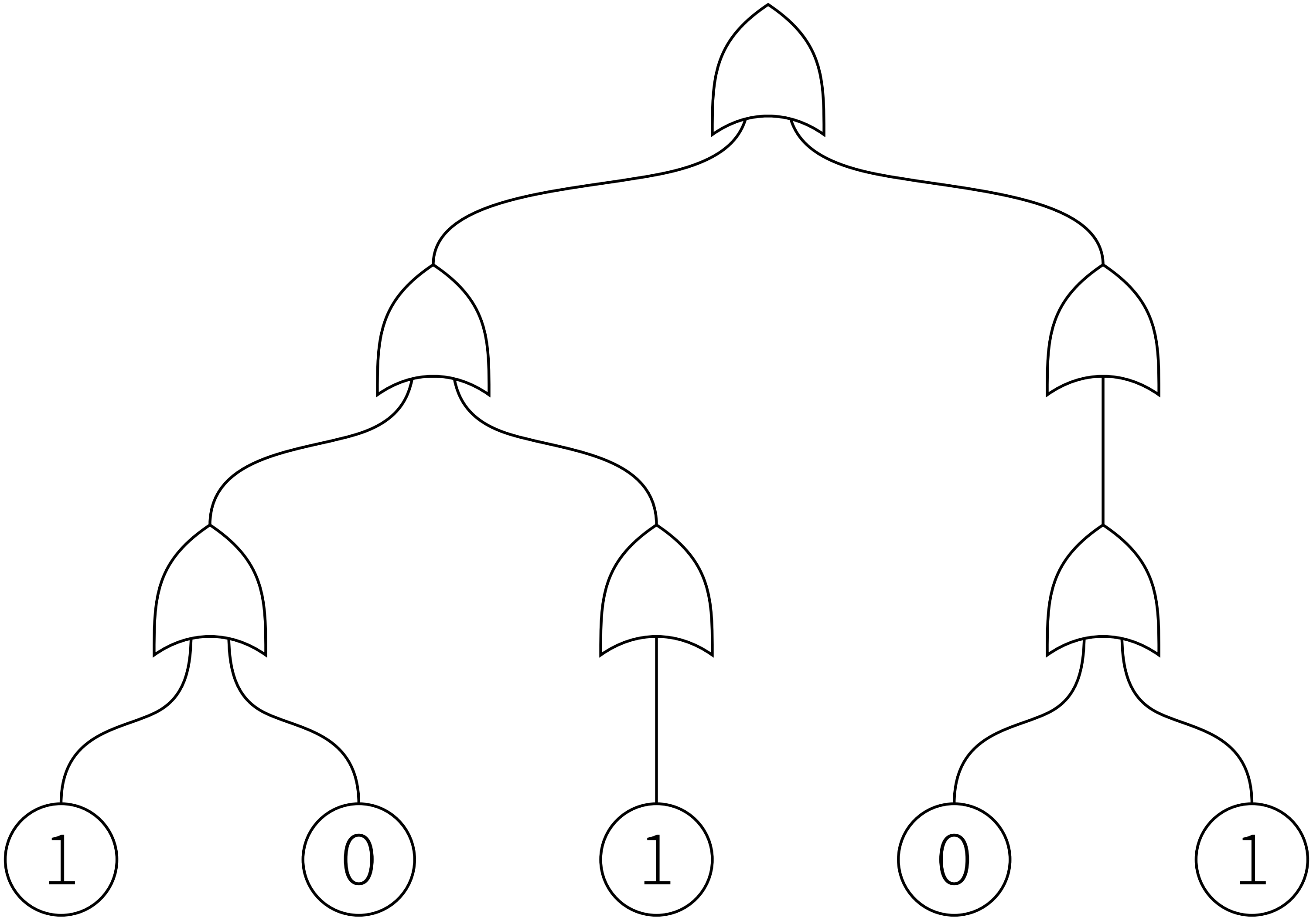
- Alternating Turing machines
- Random access machines with arithmetic operations $+ - \times \div$
- Parallel processes generated by `fork()` running on an unbounded number of processors
- Cellular automata over hyperbolic grids

Van Emde Boas: Machine models and simulations, in Van Leeuwen (ed.), Handbook of Theoretical Computer Science Volume A: Algorithms and Complexity 1–66, MIT Press, 1990

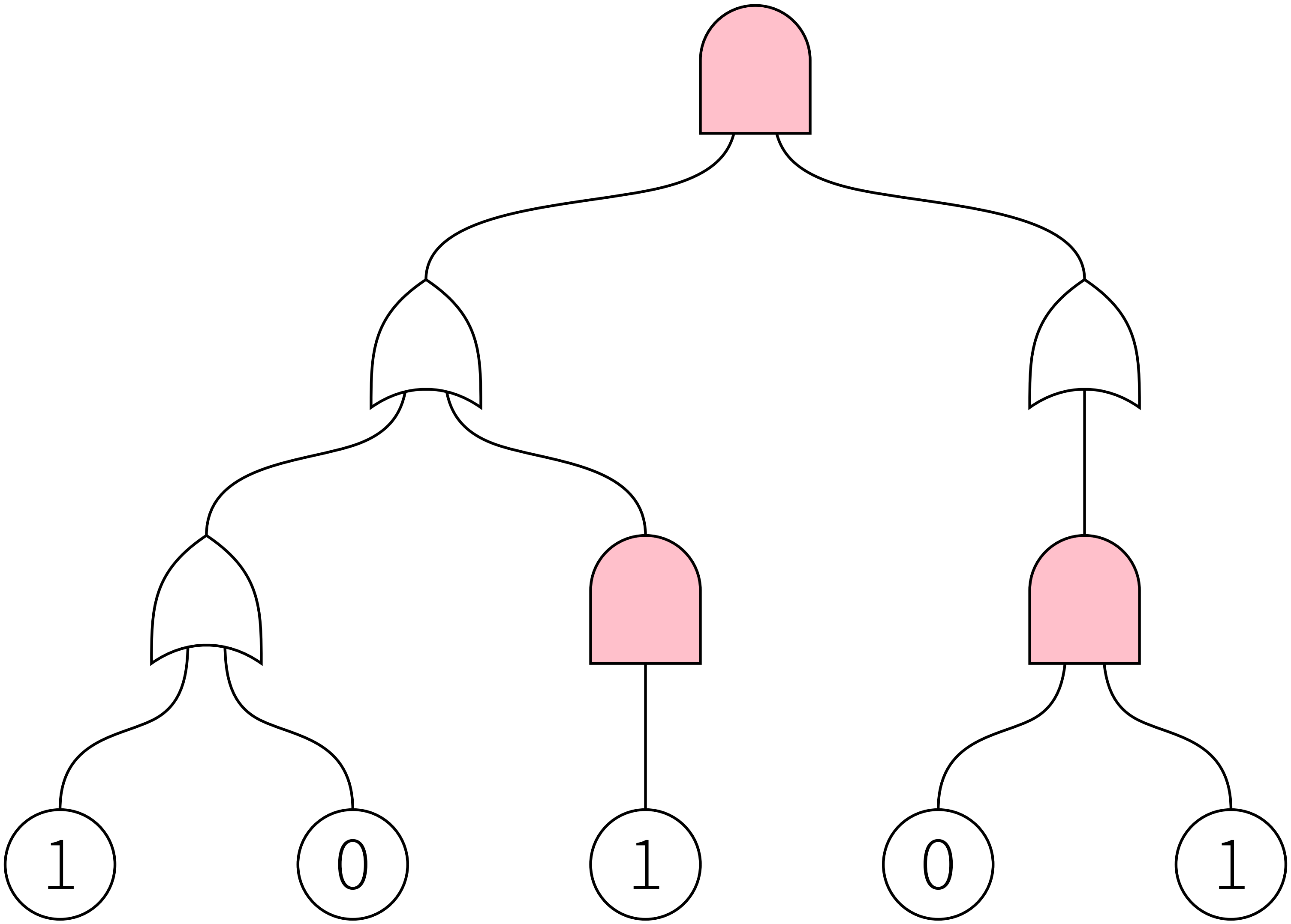
Nondeterministic Turing machines: NP



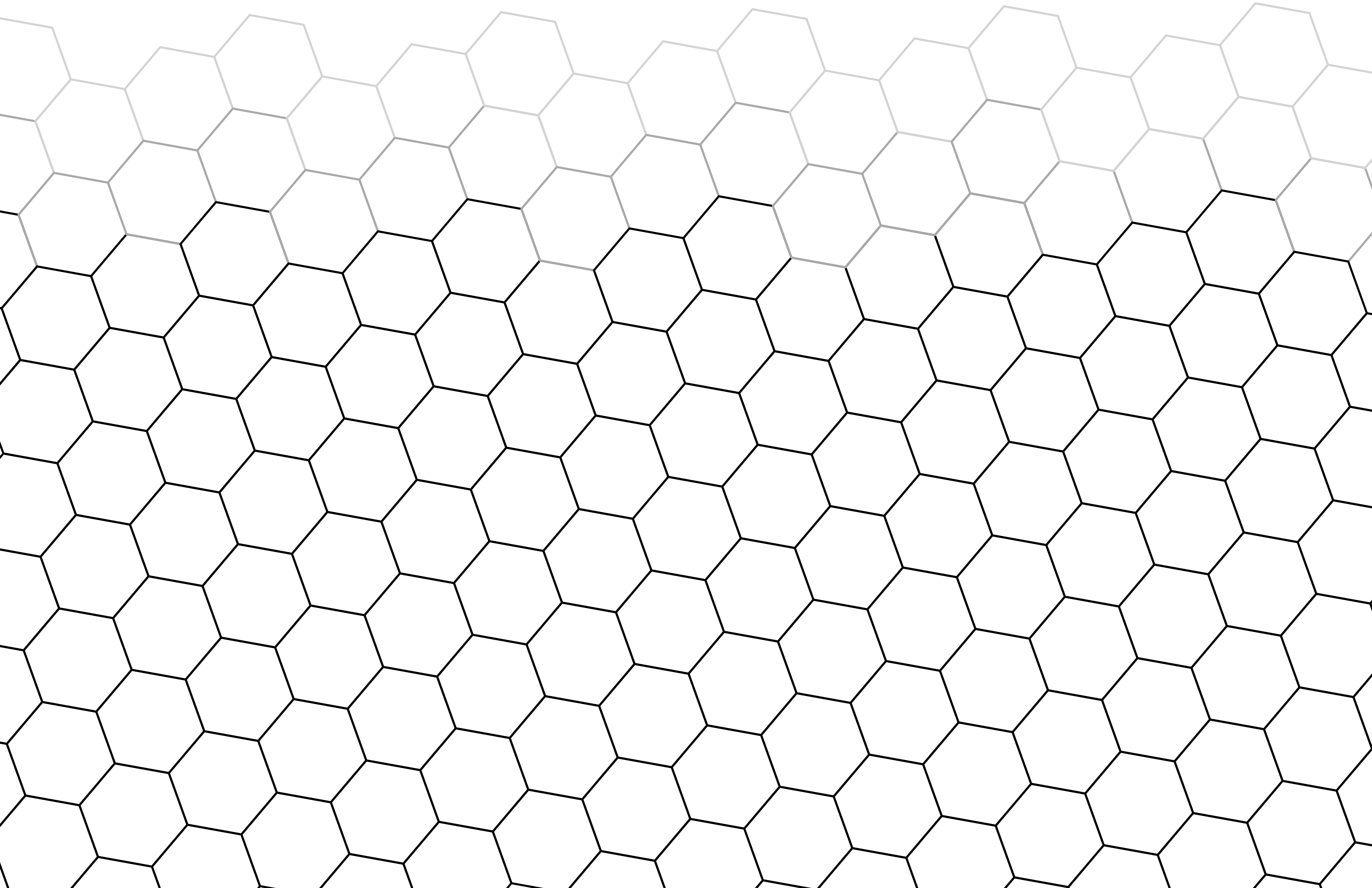
Nondeterministic Turing machines: NP



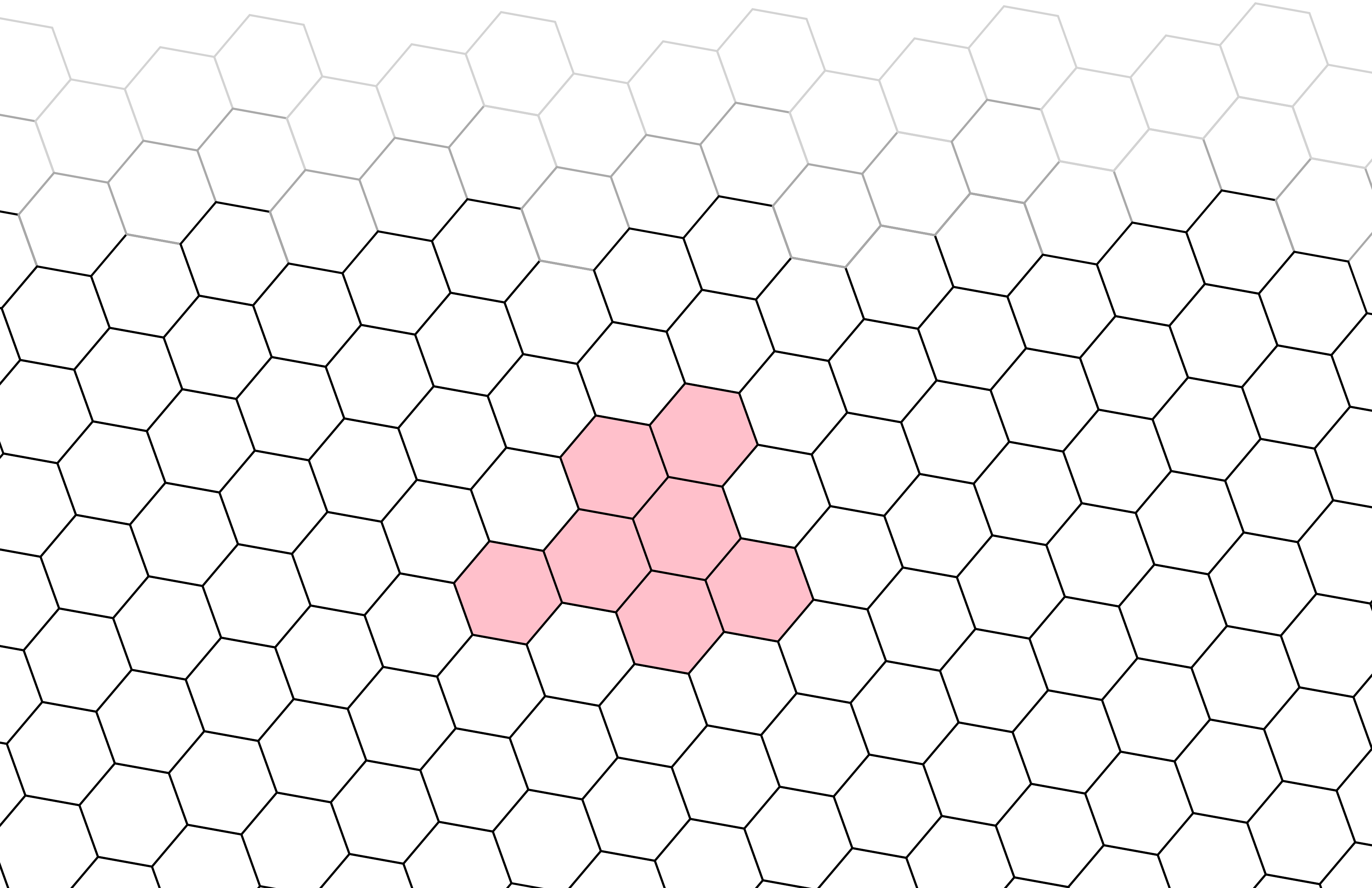
Alternating Turing machines: PSPACE



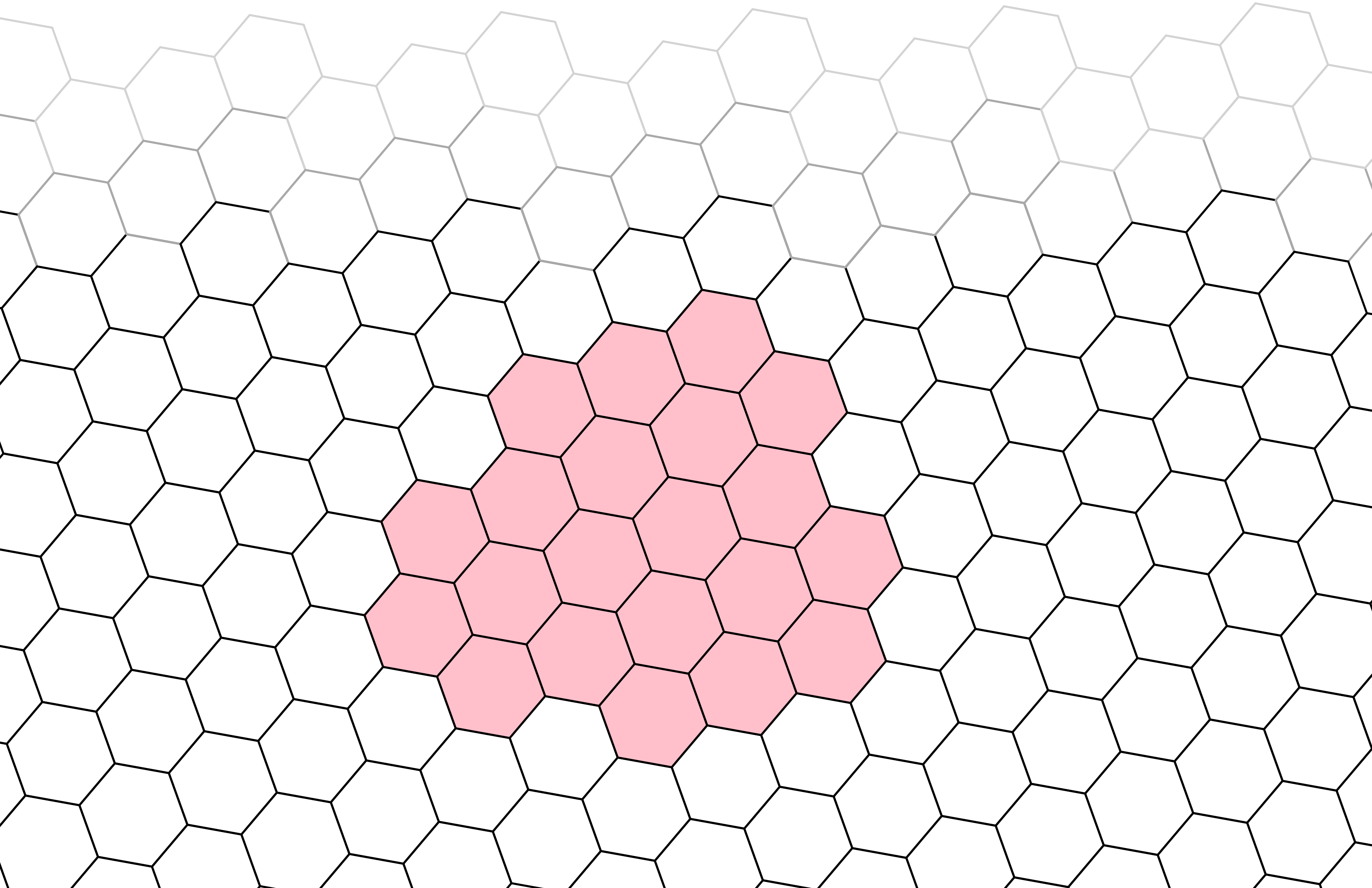
Cellular automata over a Euclidean grid



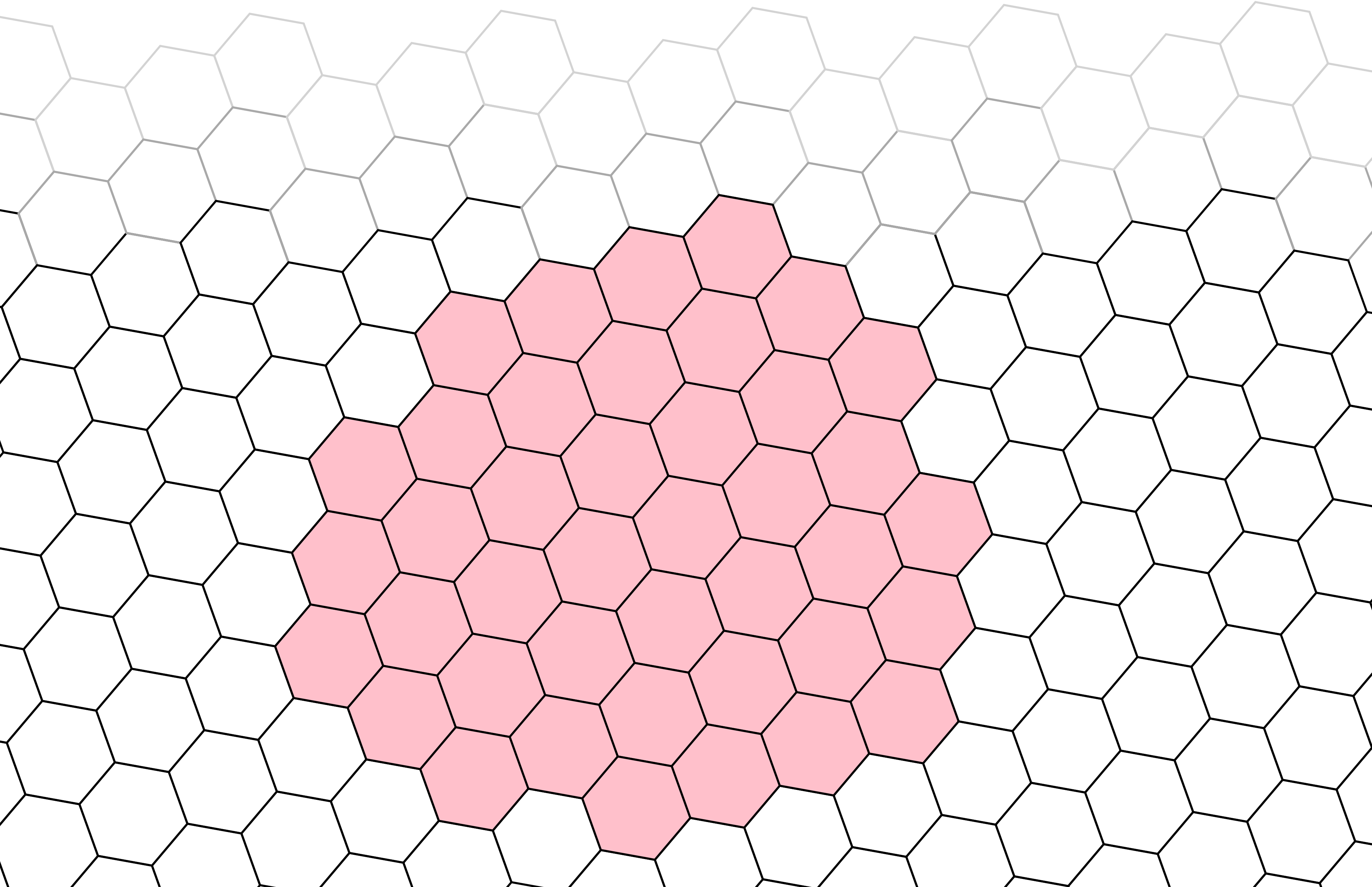
Cellular automata over a Euclidean grid



Cellular automata over a Euclidean grid

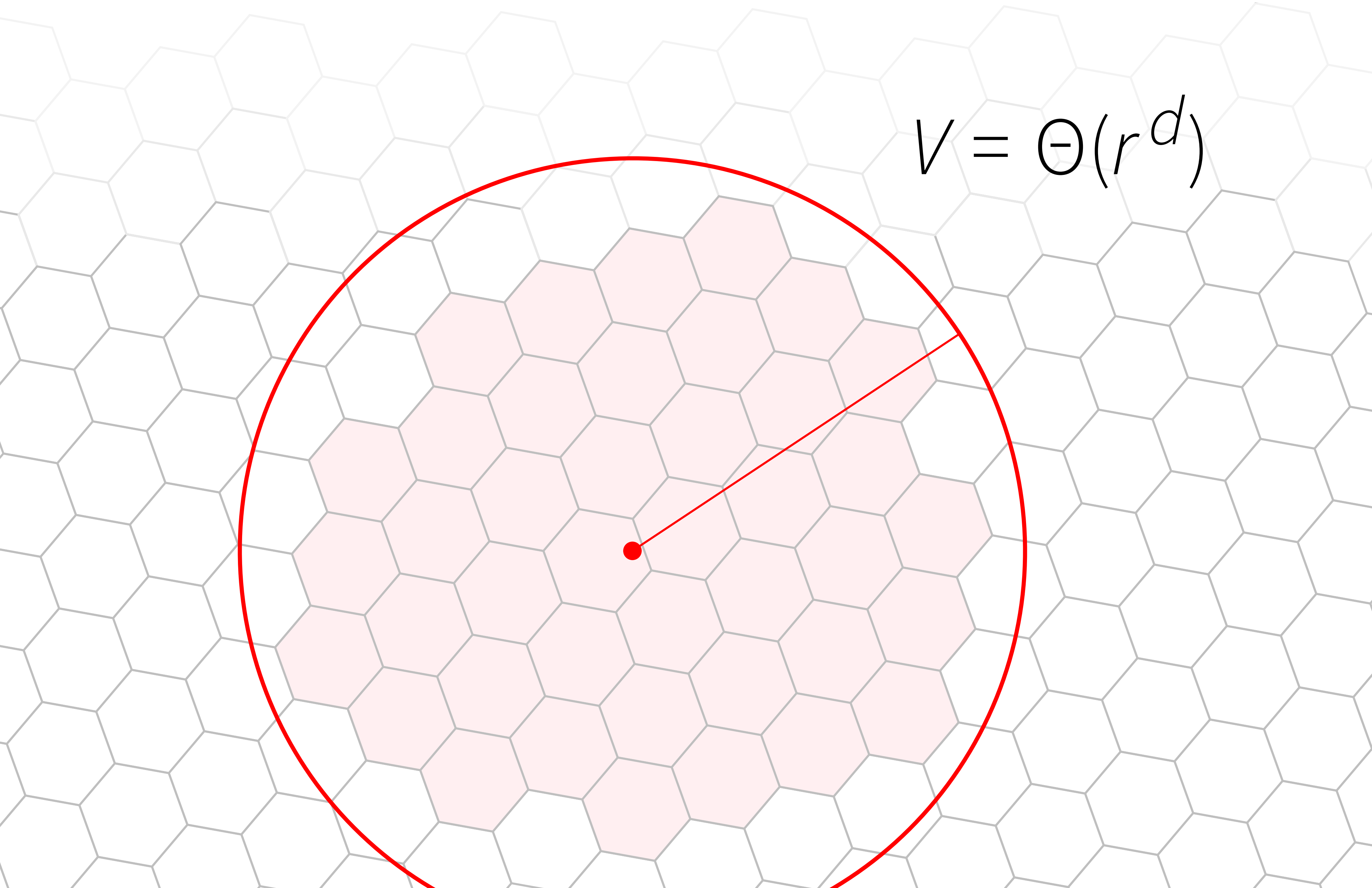


Cellular automata over a Euclidean grid

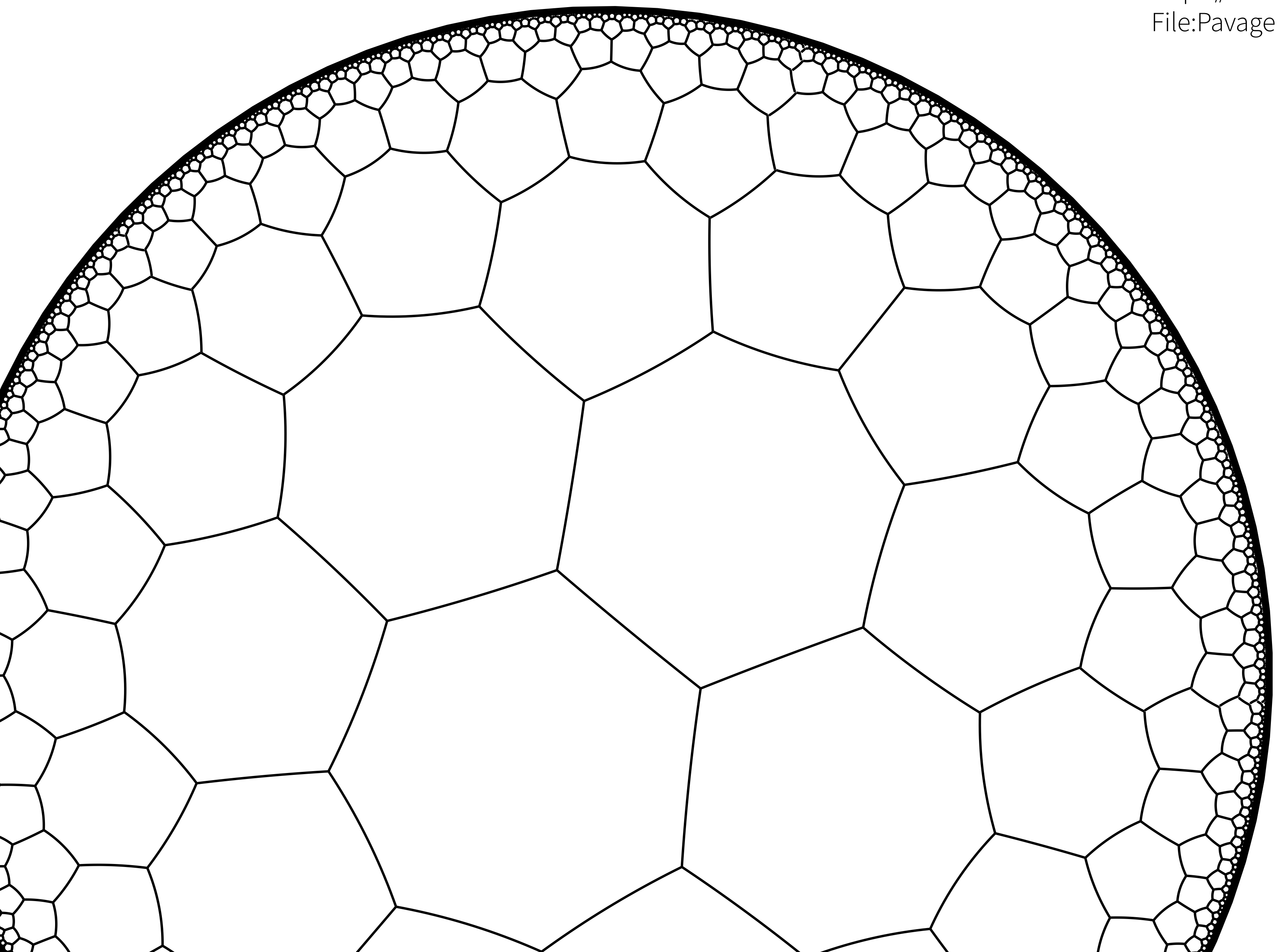


Cellular automata over a Euclidean grid

$$V = \Theta(r^d)$$



Hyperbolic cellular automata

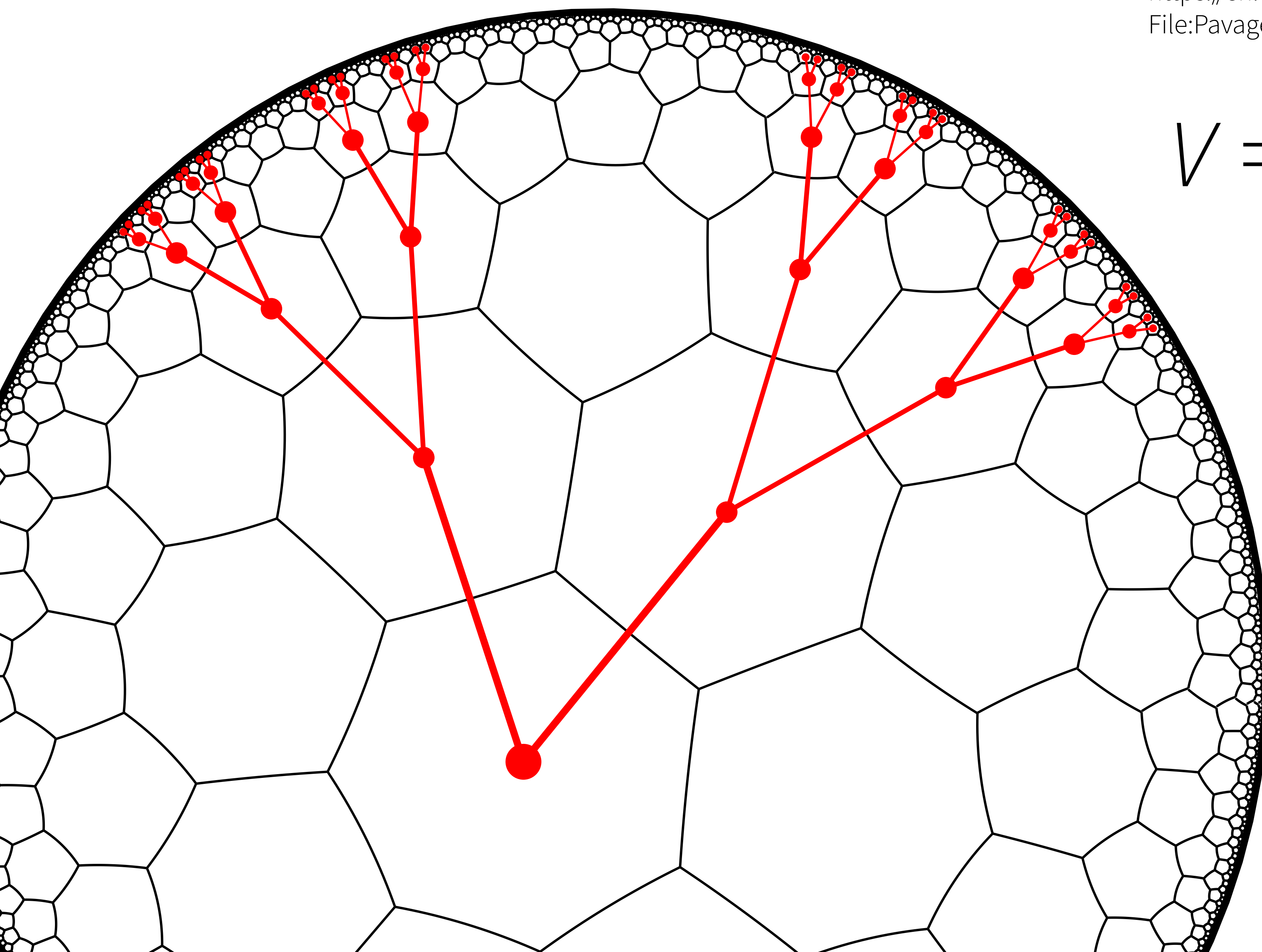


Pavage du plan hyperbolique par des heptagones, dans le modèle du disque de Poincaré. By Theon, used under CC BY-SA 3.0 https://en.wikipedia.org/wiki/File:PavageHypPoincare2.svg

Hyperbolic cellular automata

Pavage du plan hyperbolique par des heptagones, dans le modèle du disque de Poincaré. By Theon, used under CC BY-SA 3.0 https://en.wikipedia.org/wiki/File:PavageHypPoincare2.svg

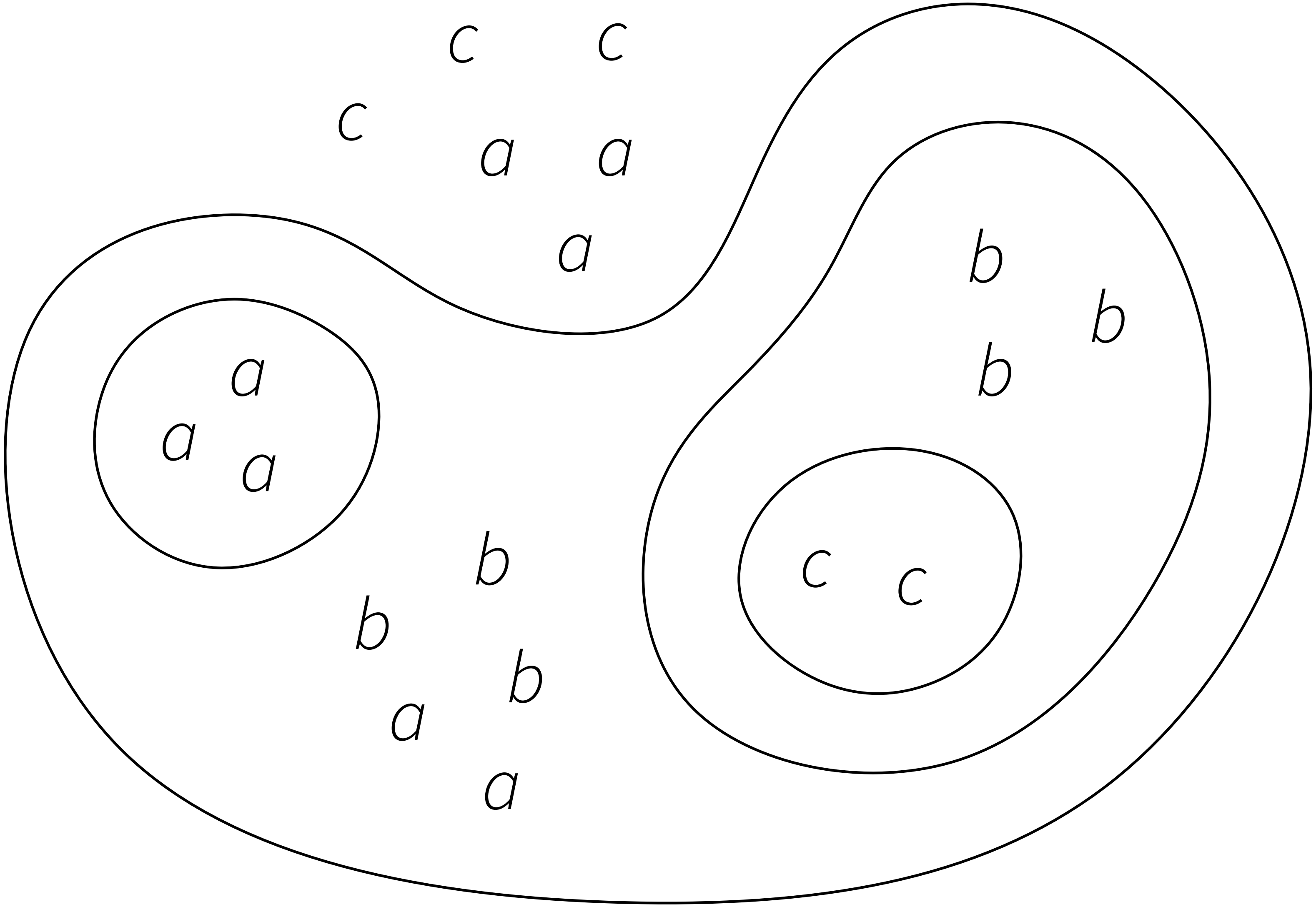
$$V = \Omega(2^r)$$



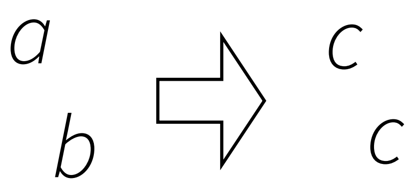
Rule of thumb

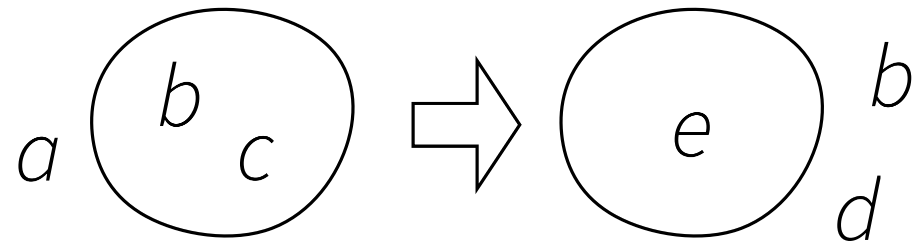
- Sequential machines are first class
- Bounded parallel machines are first class
- Unbounded parallel machines are second class

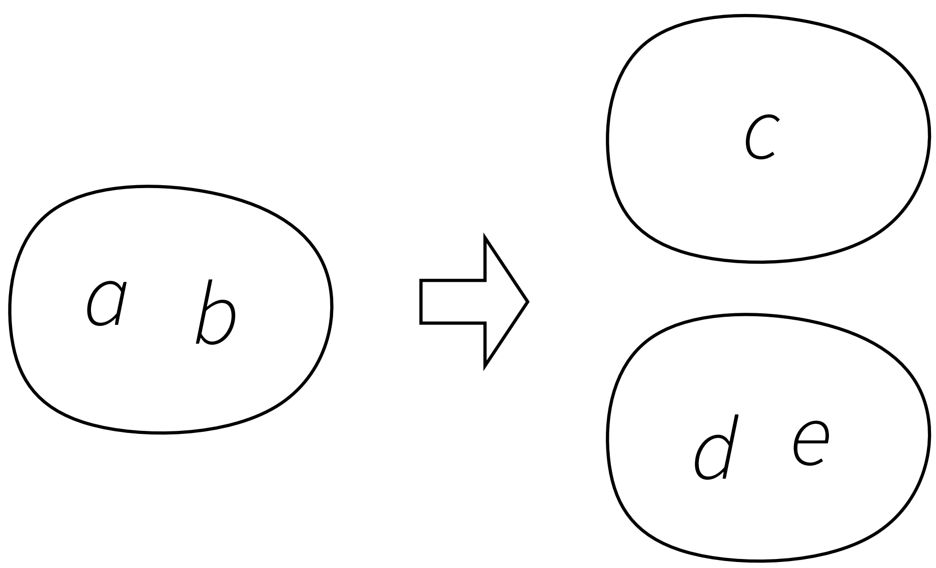
Membrane systems (P systems)



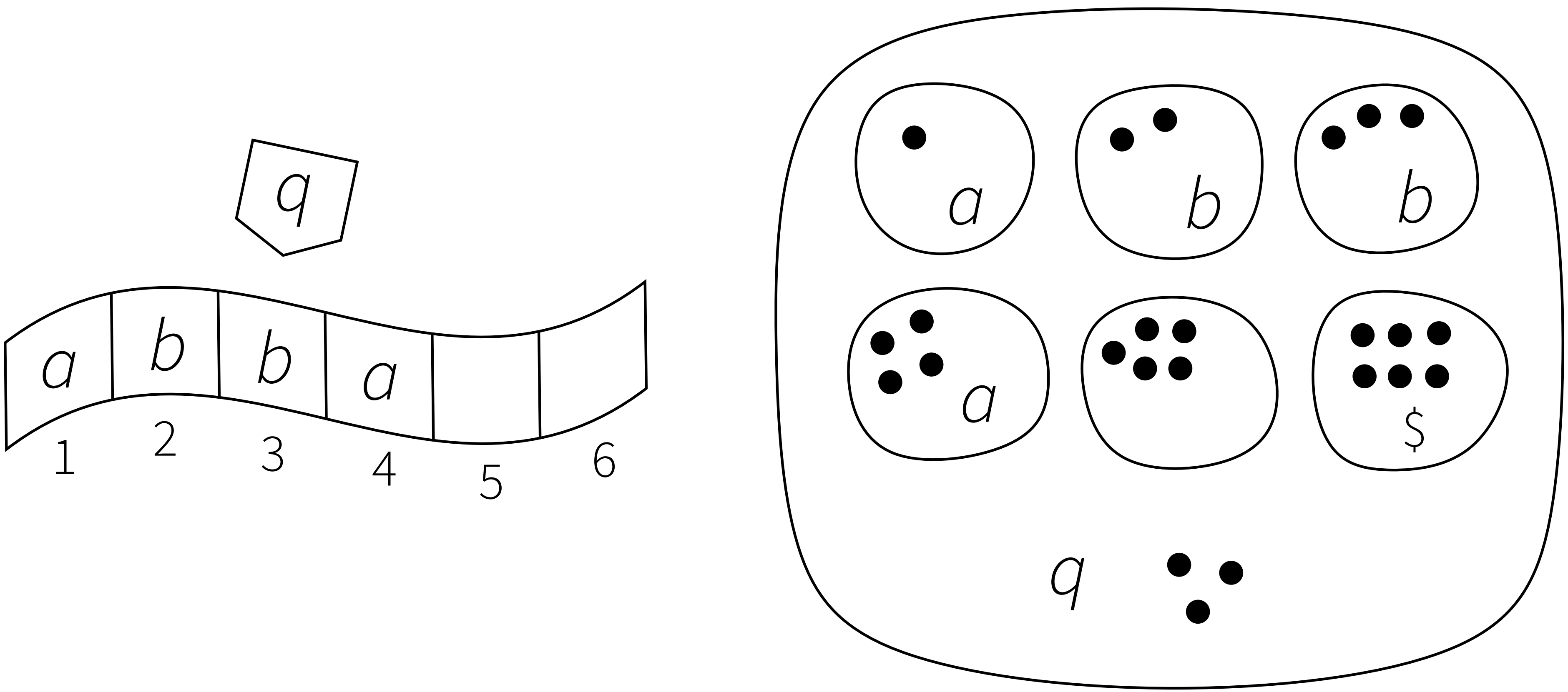
Evolution rules and parallel semantics





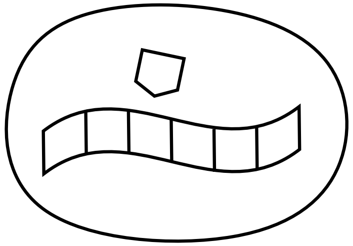


Efficient universality of membrane systems



Alhazov, Leporati, Mauri, Porreca, Zandron: Space complexity equivalence of P systems with active membranes and Turing machines. Theoretical Computer Science 529, 69–81 (2014)

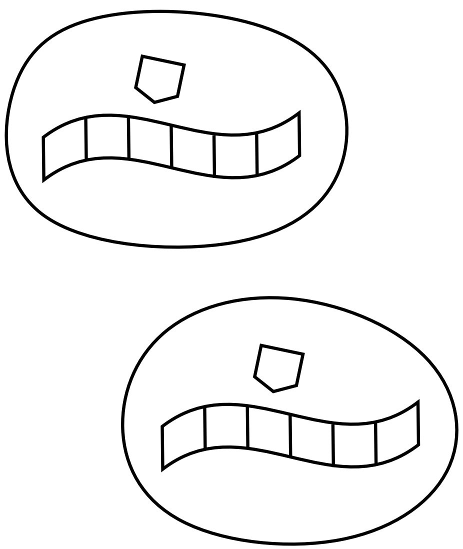
Membrane systems solving NP problems



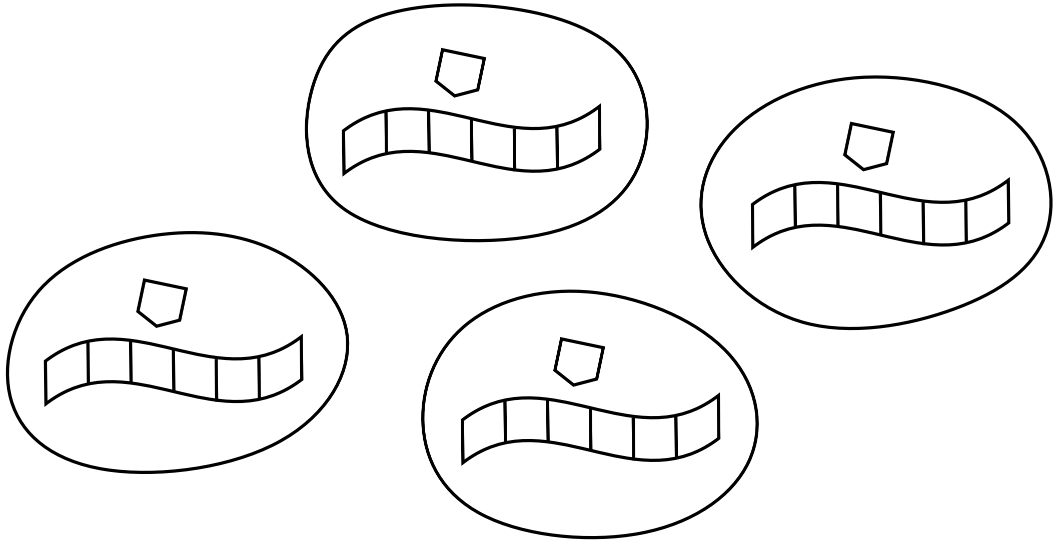
Membrane systems solving NP problems

$$\delta(q, a) = \begin{cases} (q', b, +1) \\ (q', c, -1) \end{cases}$$

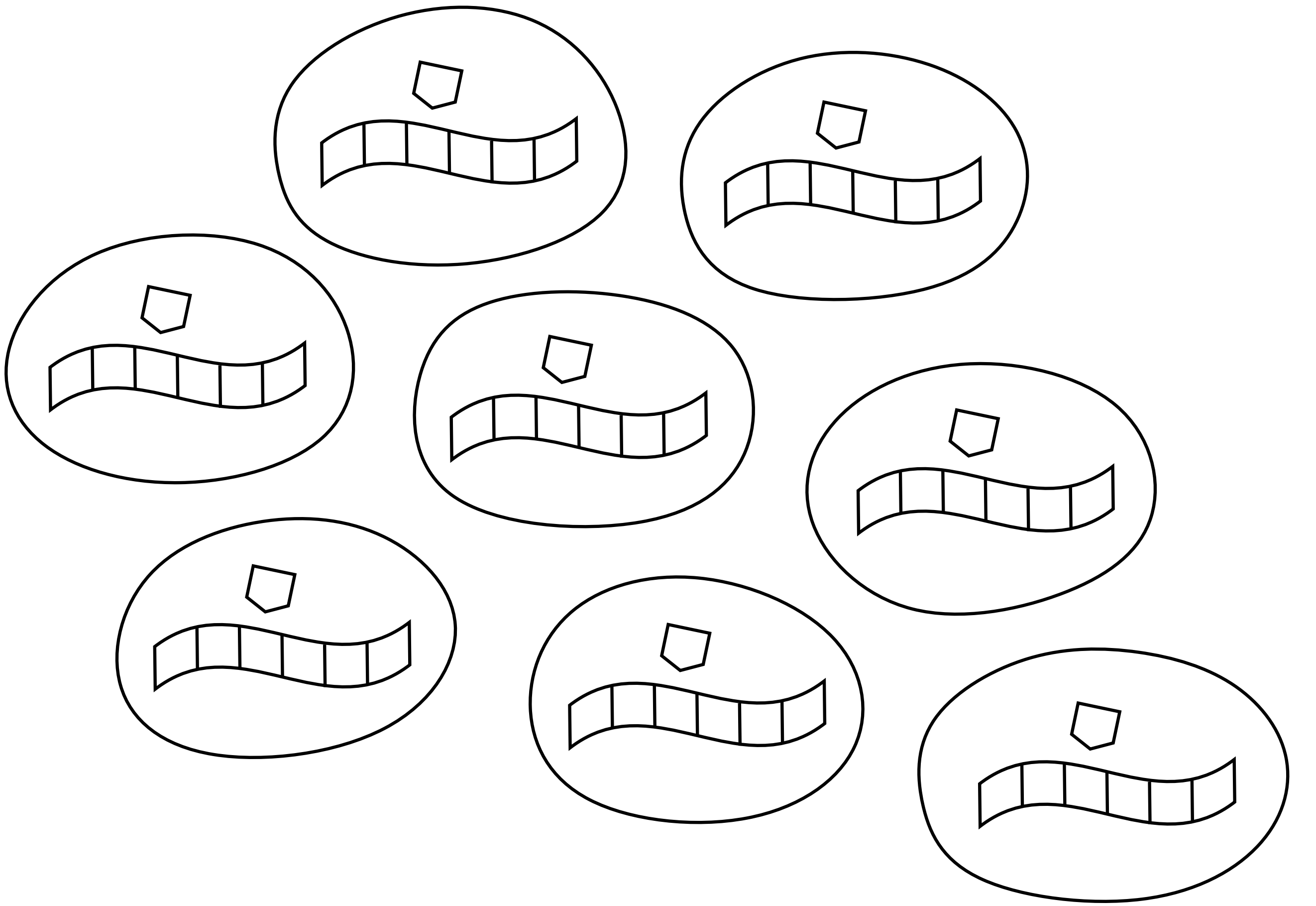
$$[q_3\ a_3] \rightarrow [q'_4\ b_3]\ [q''_2\ c_3]$$



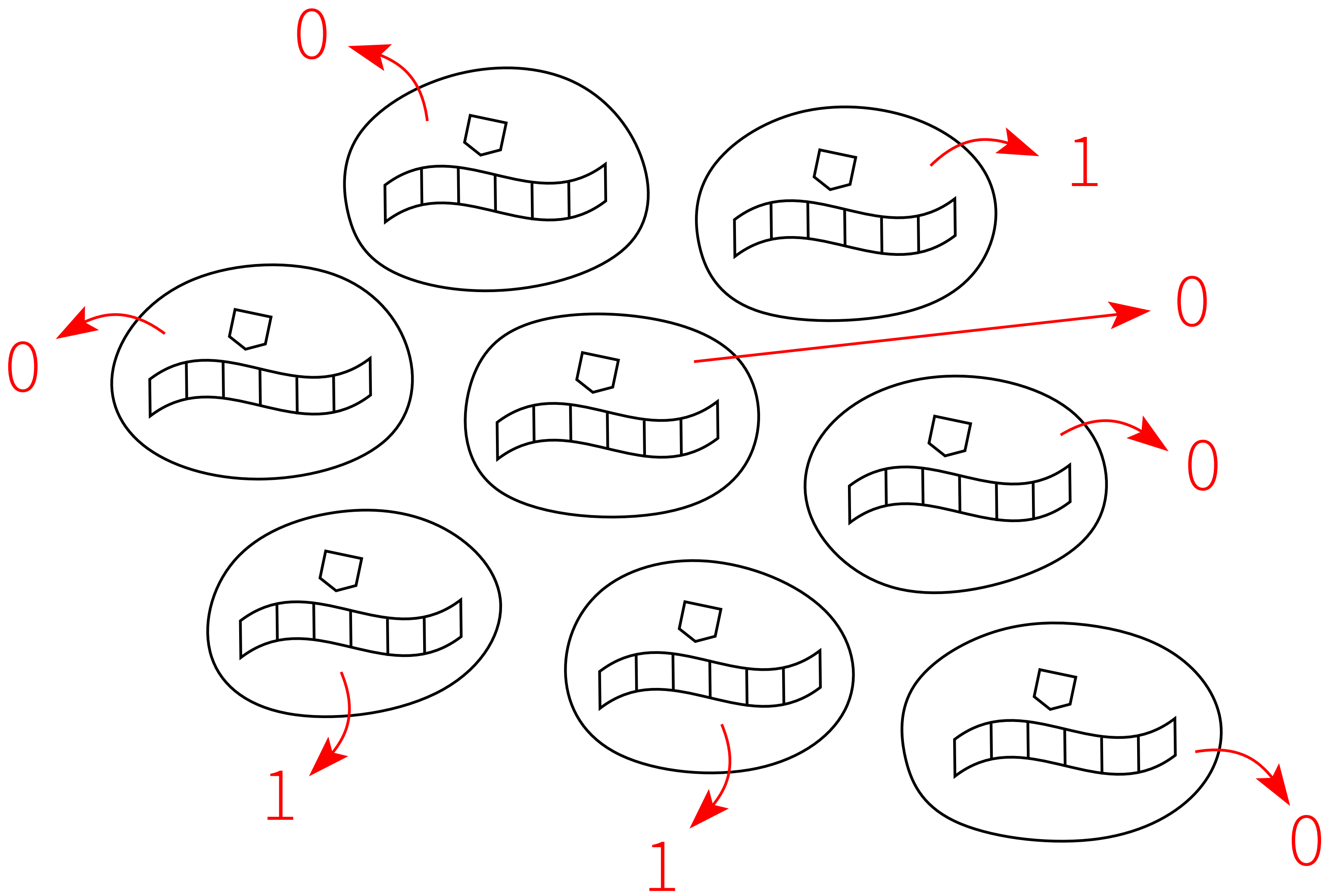
Membrane systems solving NP problems



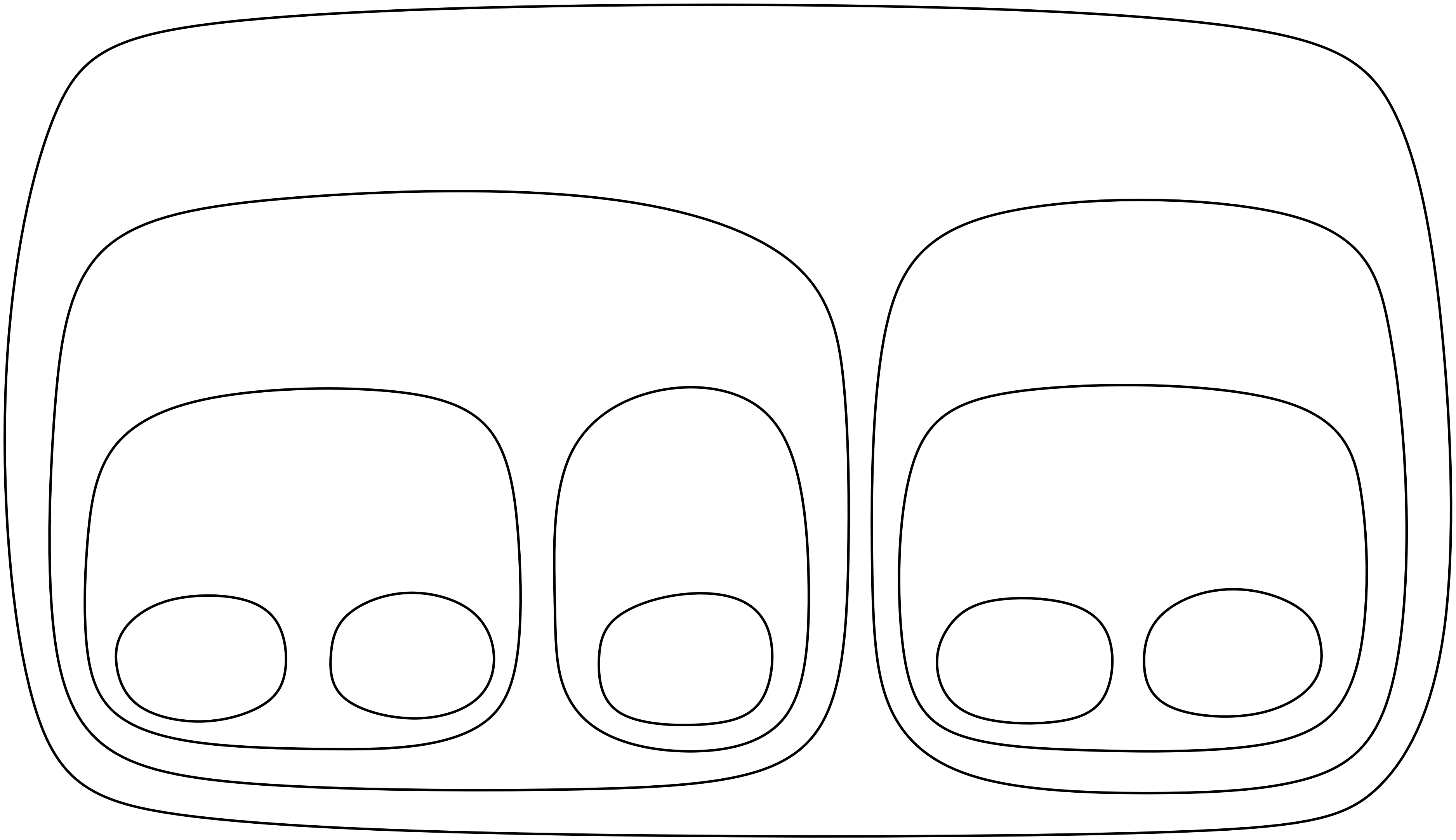
Membrane systems solving NP problems



Membrane systems solving NP problems

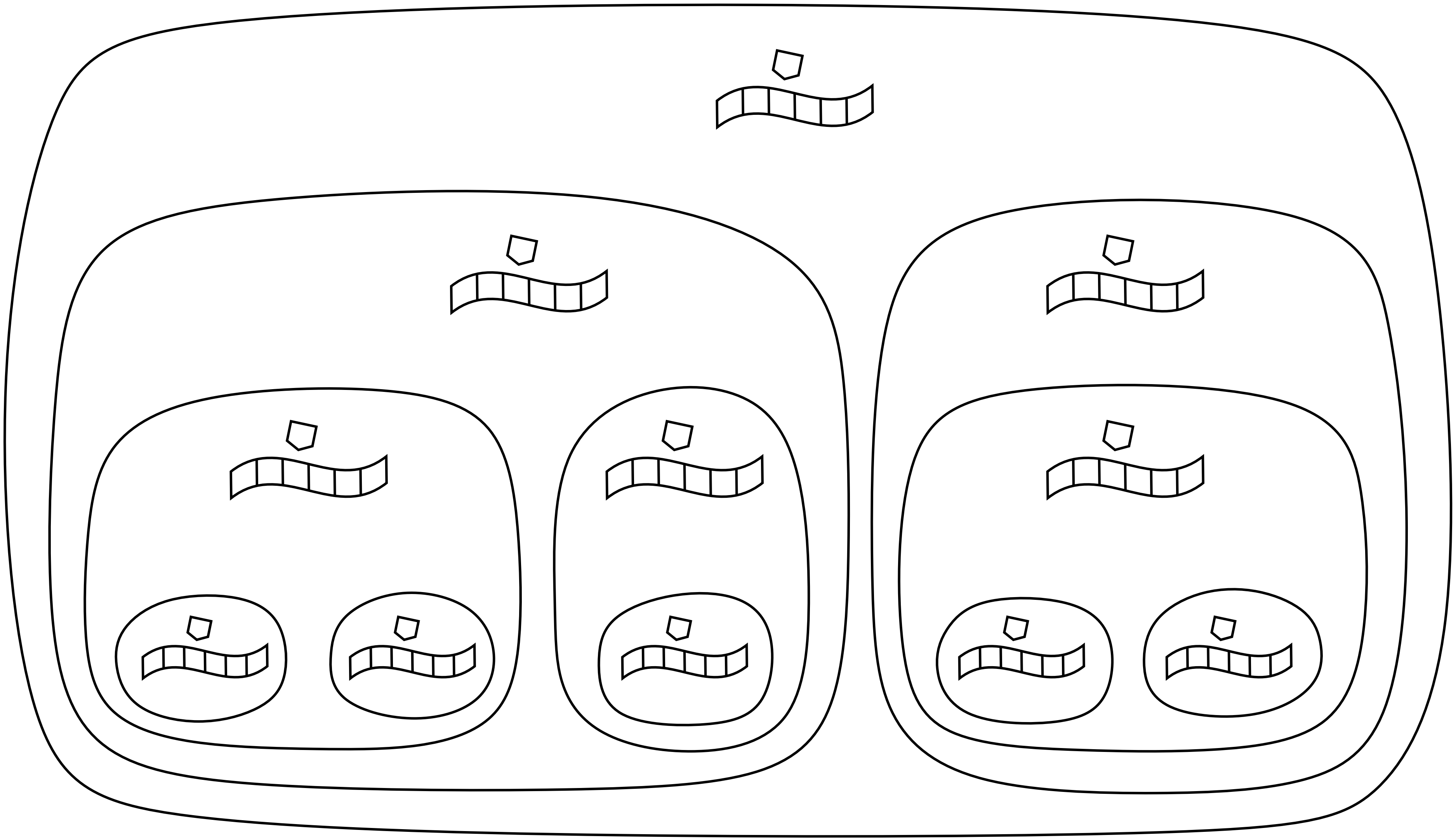


Membrane systems are second class



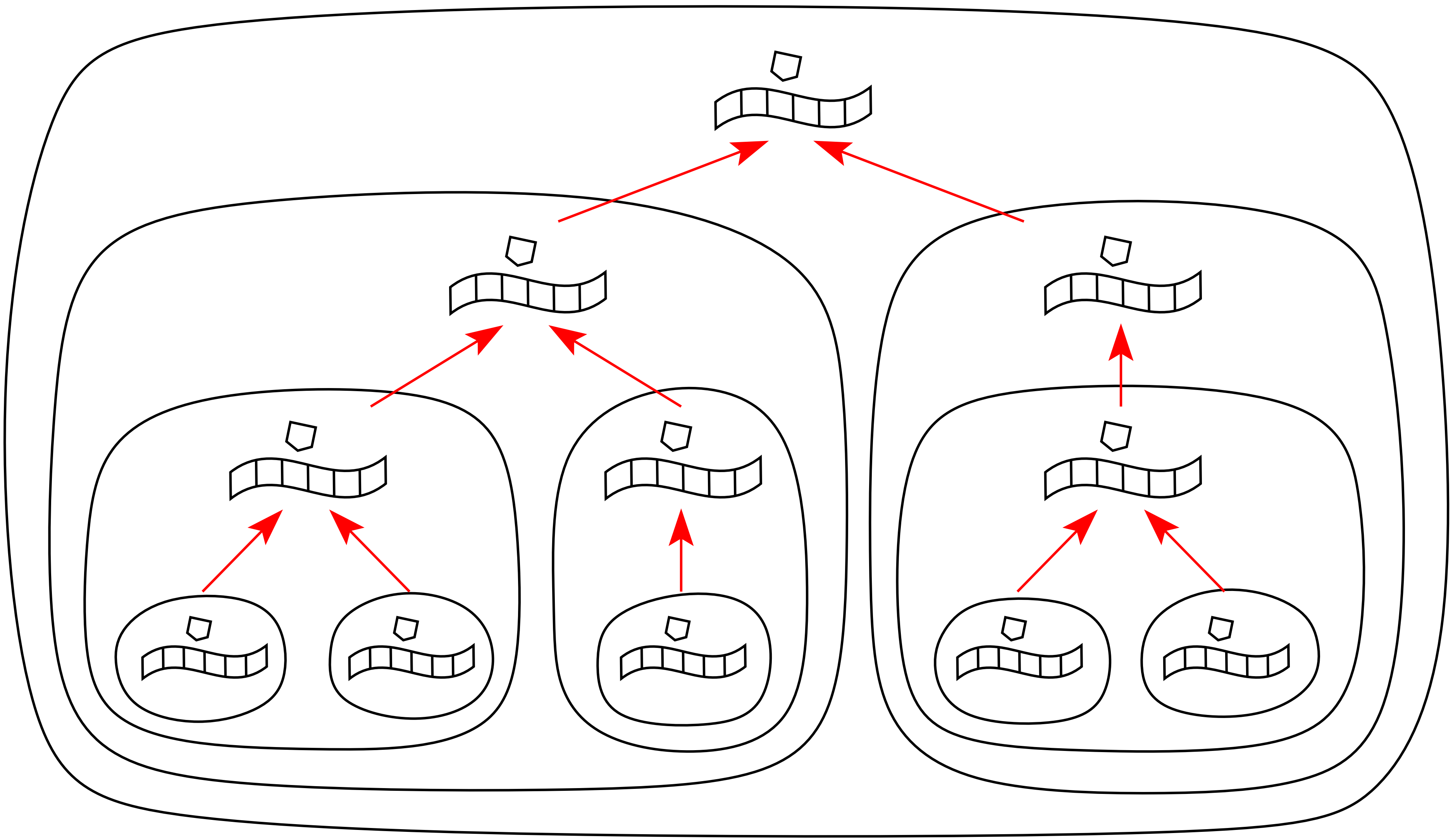
Sosík, Rodríguez-Patón: Membrane computing and complexity theory: A characterization of PSPACE. Journal of Computer and System Sciences 73(1), 137–152 (2007)

Membrane systems are second class



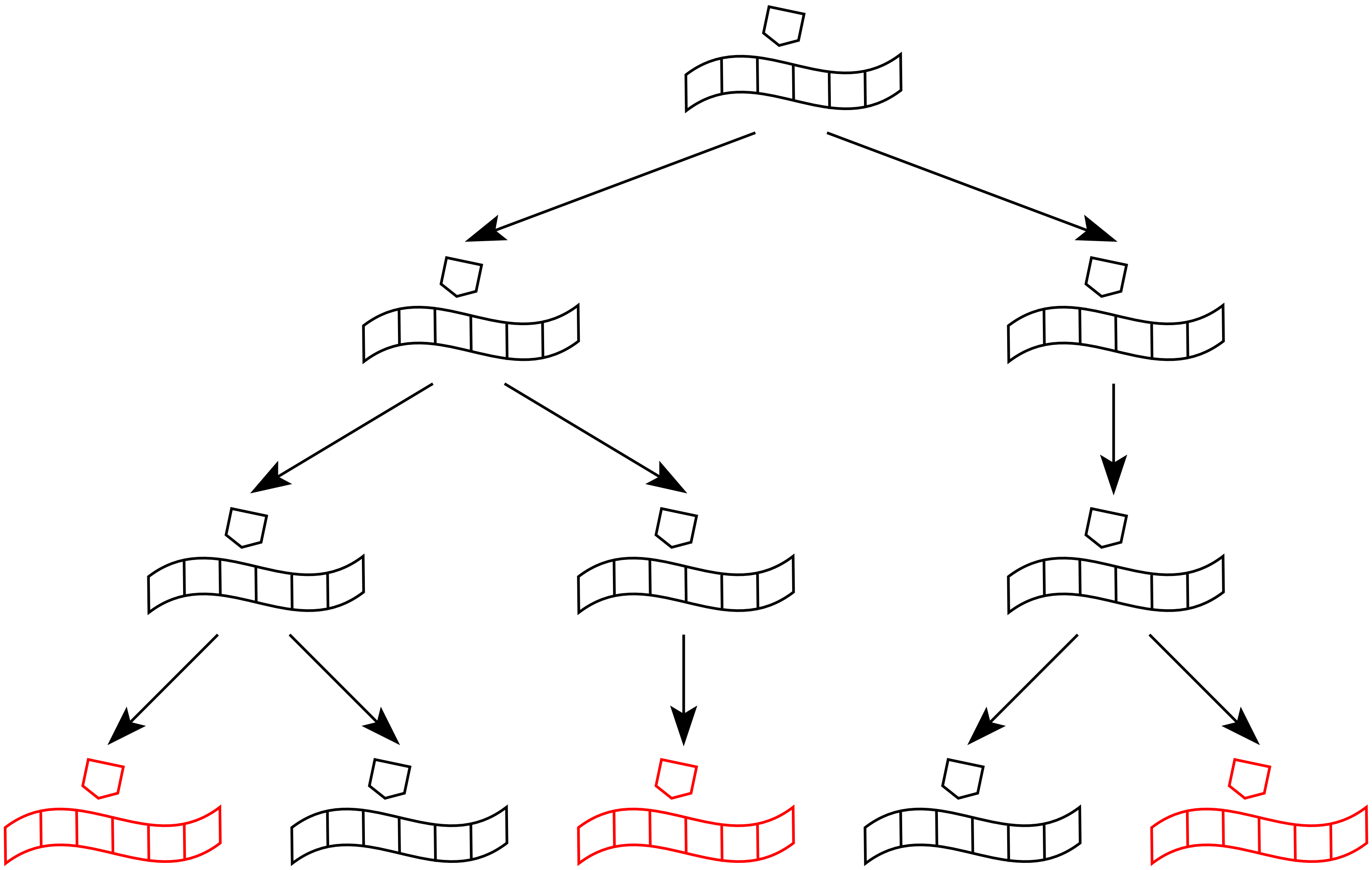
Sosík, Rodríguez-Patón: Membrane computing and complexity theory: A characterization of PSPACE. Journal of Computer and System Sciences 73(1), 137–152 (2007)

Membrane systems are second class



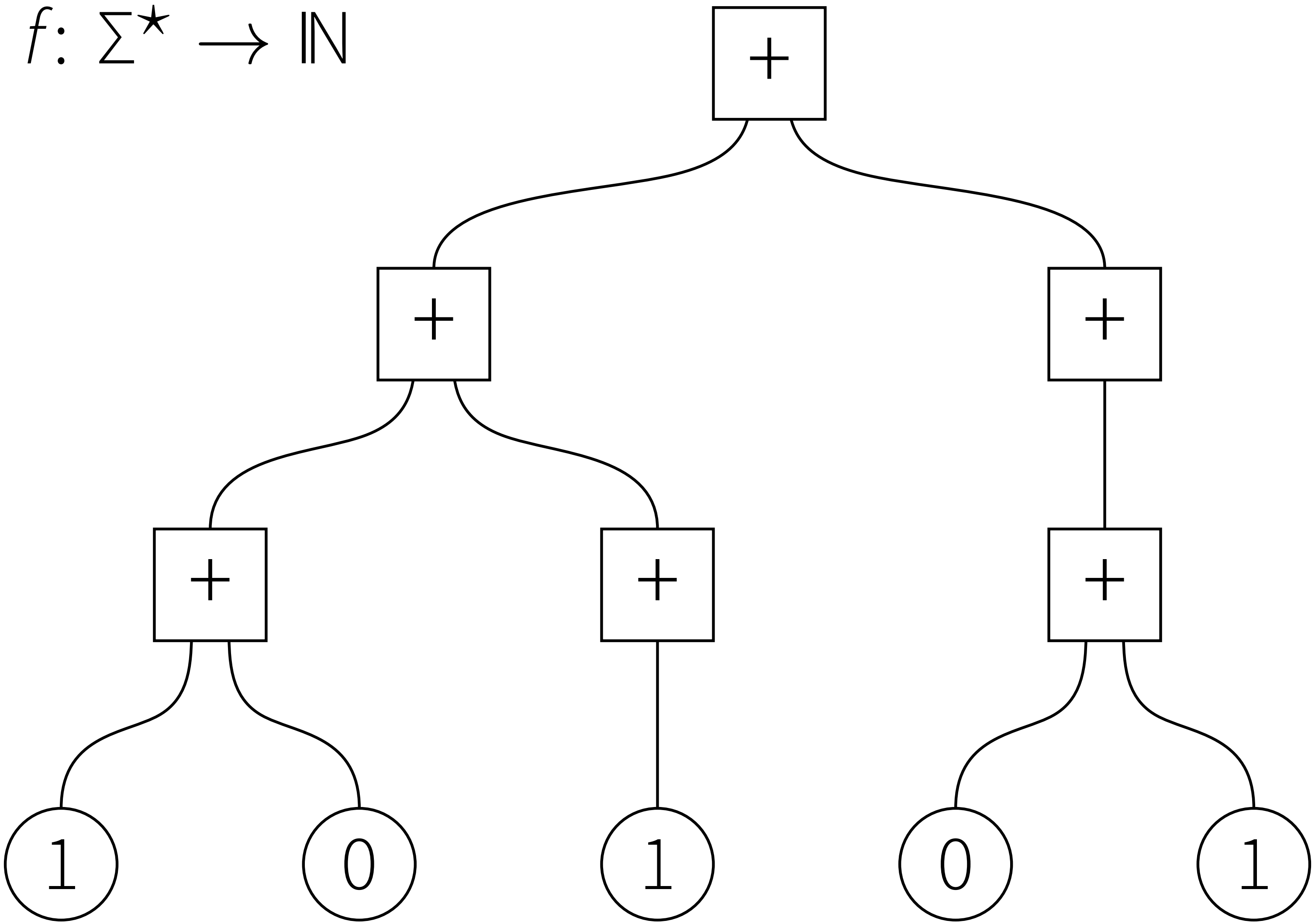
Sosík, Rodríguez-Patón: Membrane computing and complexity theory: A characterization of PSPACE. Journal of Computer and System Sciences 73(1), 137–152 (2007)

Counting Turing machines: #P

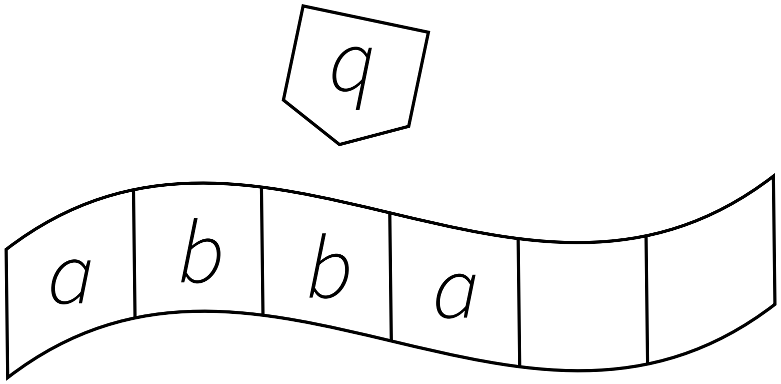


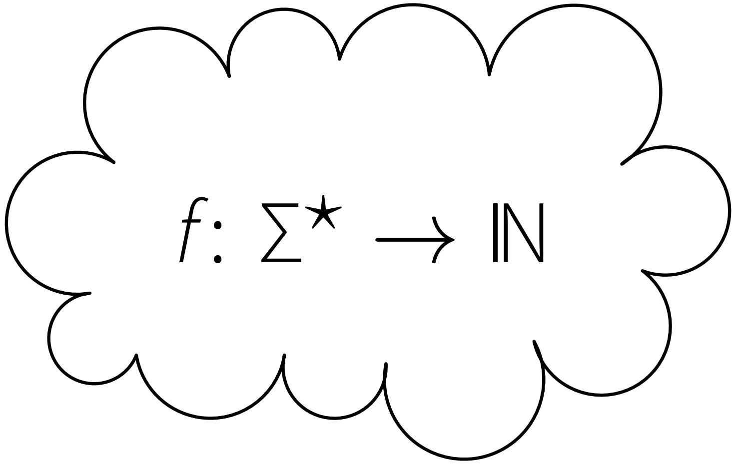
Counting Turing machines: #P

$f \colon \Sigma^\star \to \mathbb{N}$

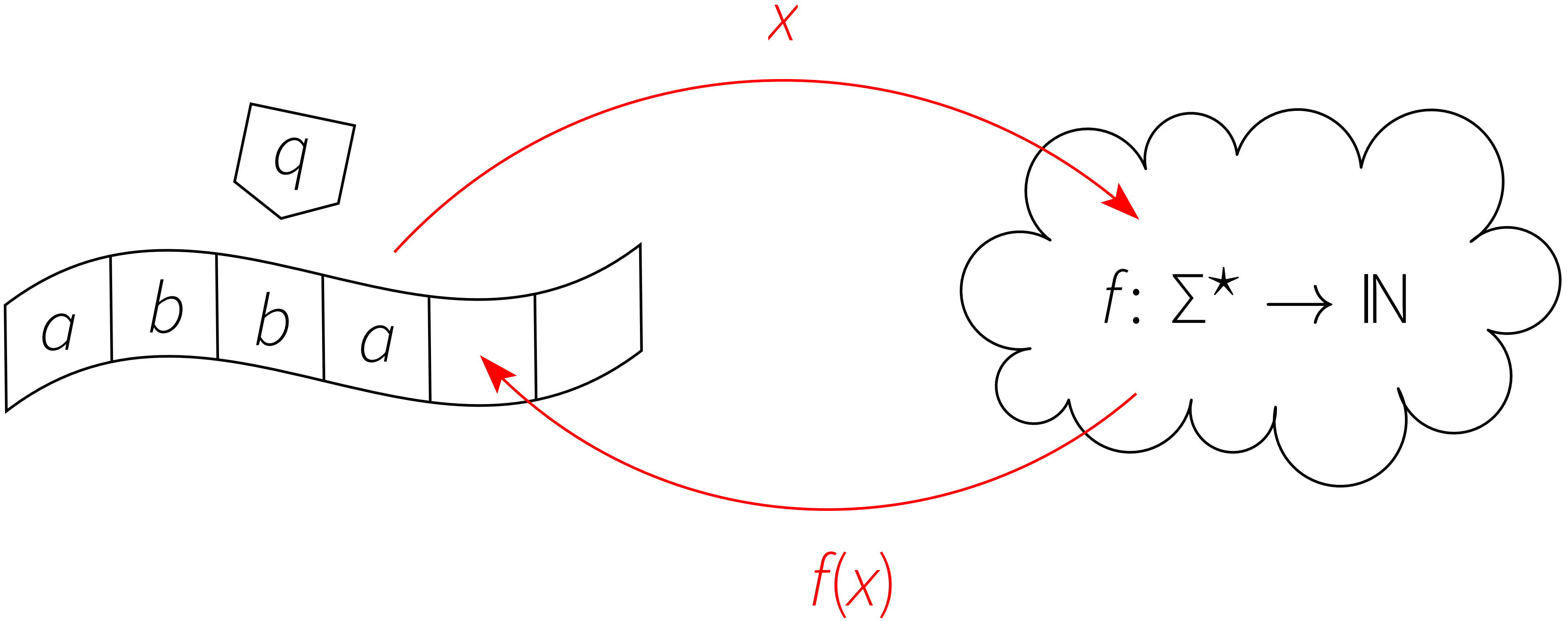


Oracle Turing machines and the class $P^{\#P}$

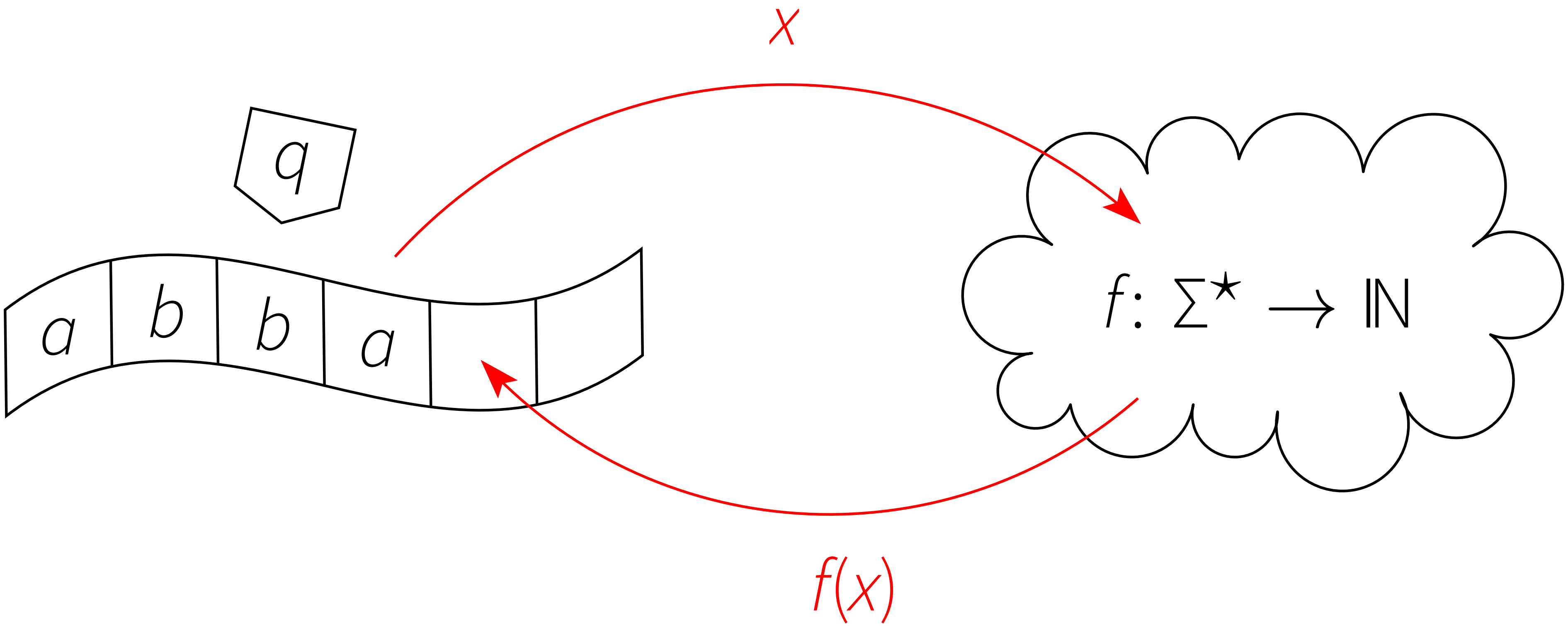




Oracle Turing machines and the class $P^{\#P}$



Oracle Turing machines and the class $\mathsf{P}^{\#\mathsf{P}}$



$$\underbrace{\mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{NP}^{\mathsf{NP}} \subseteq \mathsf{NP}^{\mathsf{NP}^{\mathsf{NP}}} \subseteq \cdots}_{\mathsf{PH}} \subseteq \mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{PSPACE}$$

Toda's Theorem

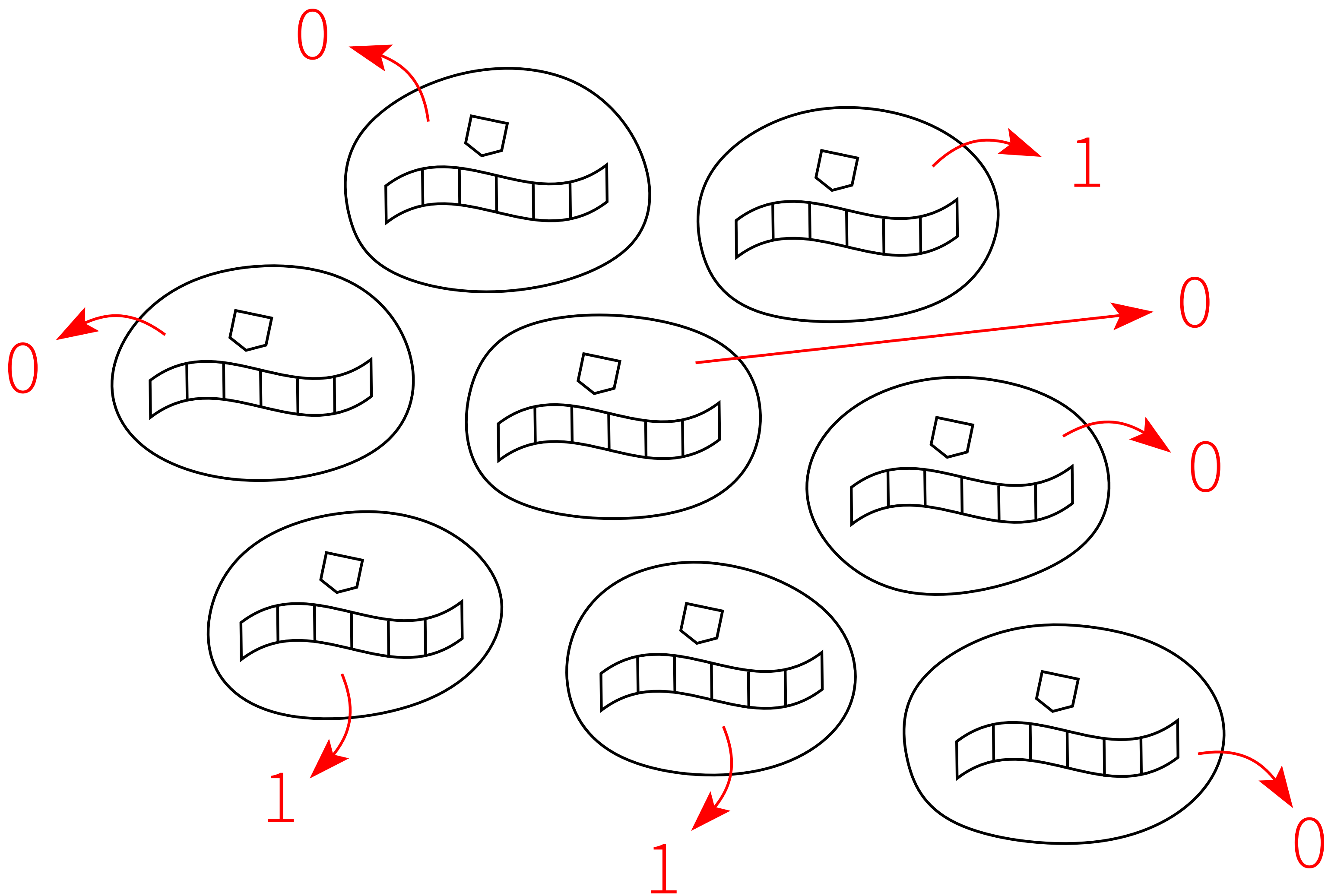
Oracle Turing machines and the class $\mathrm{P}^{\#\mathrm{P}}$

$\exists \vec{x}\, \varphi \qquad \exists \vec{x} \forall \vec{y}\, \varphi \qquad \exists \vec{x} \forall \vec{y} \exists \vec{z}\, \varphi \qquad \exists \vec{x}_1 \forall \vec{x}_2 \cdots \exists \vec{x}_n\, \varphi$

$$\underbrace{\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{NP}^{\mathrm{NP}} \subseteq \mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}}} \subseteq \cdots}_{\mathrm{PH}} \subseteq \mathrm{P}^{\#\mathrm{P}} \subseteq \mathrm{PSPACE}$$

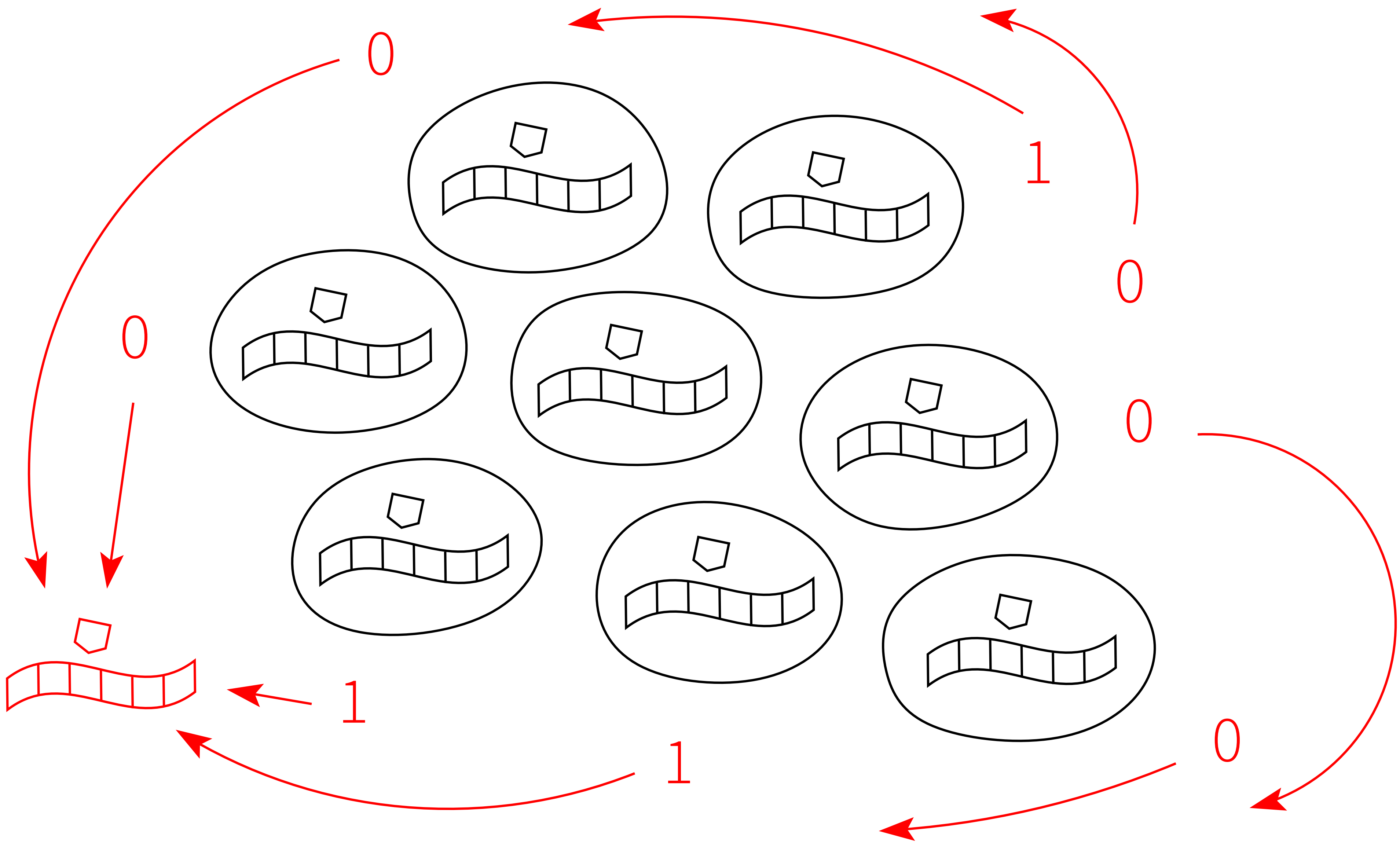
Toda's Theorem

Shallow membrane systems solve $P^{\#P}$



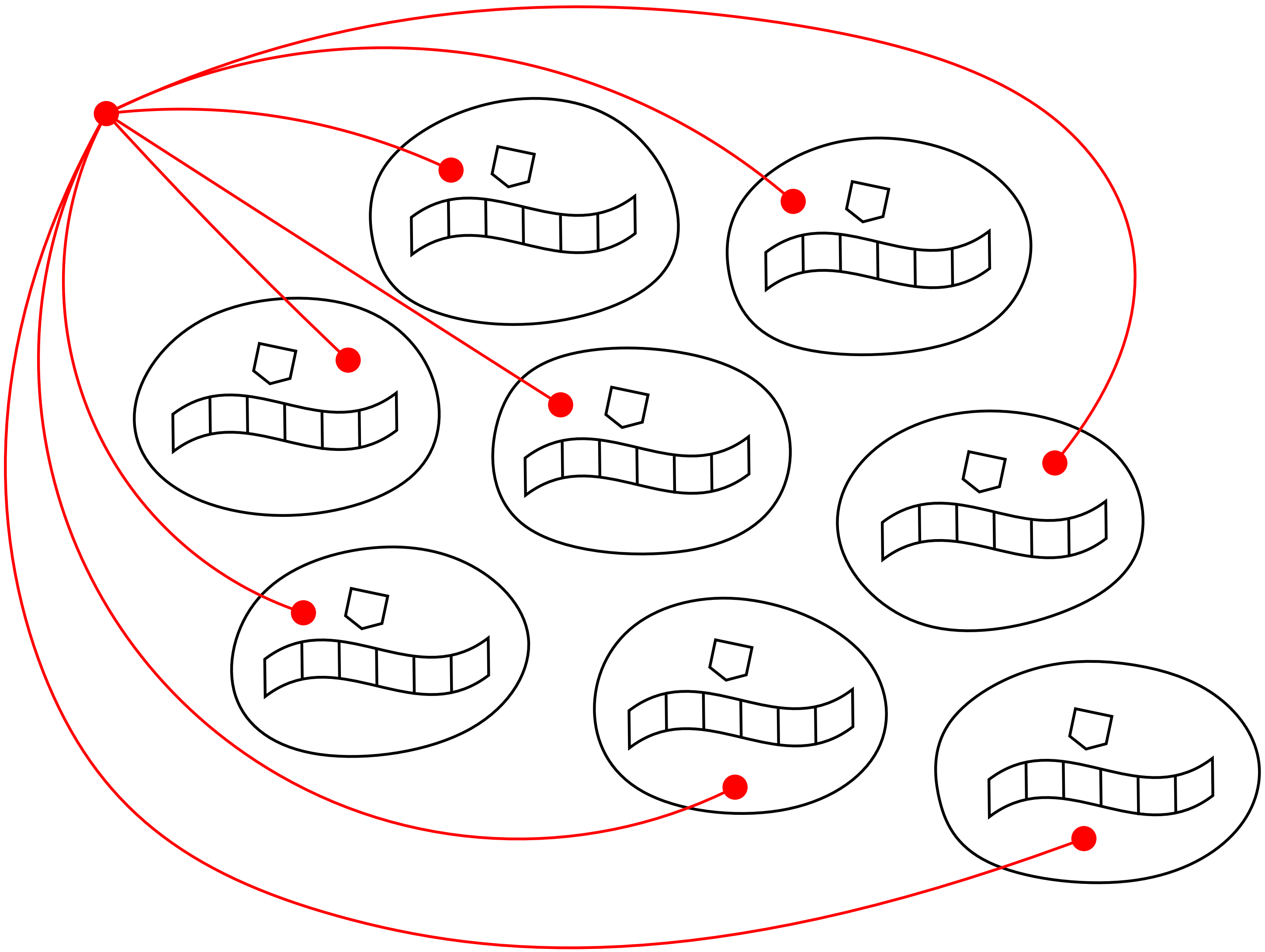
Leporati, Manzoni, Mauri, Porreca, Zandron: Characterising the complexity of tissue P systems with fission rules. Journal of Computer and System Sciences 90, 115–128 (2017)

Shallow membrane systems solve $P^{\#P}$



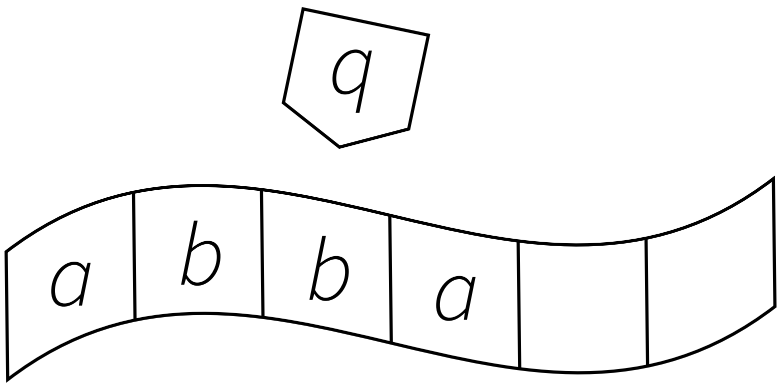
Leporati, Manzoni, Mauri, Porreca, Zandron: Characterising the complexity of tissue P systems with fission rules. Journal of Computer and System Sciences 90, 115–128 (2017)

Shallow membrane systems solve $P^{\#P}$

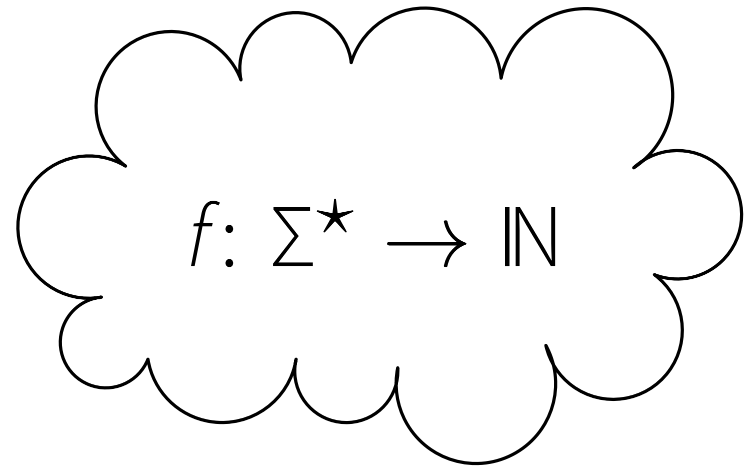


Leporati, Manzoni, Mauri, Porreca, Zandron: Characterising the complexity of tissue P systems with fission rules. Journal of Computer and System Sciences 90, 115–128 (2017)

Simulating shallow membrane systems



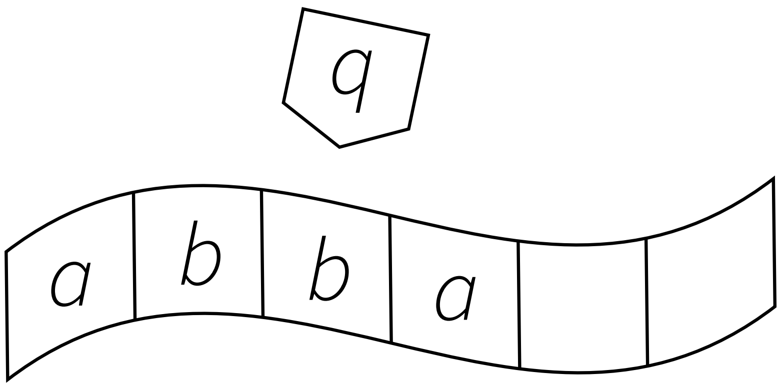
external environment



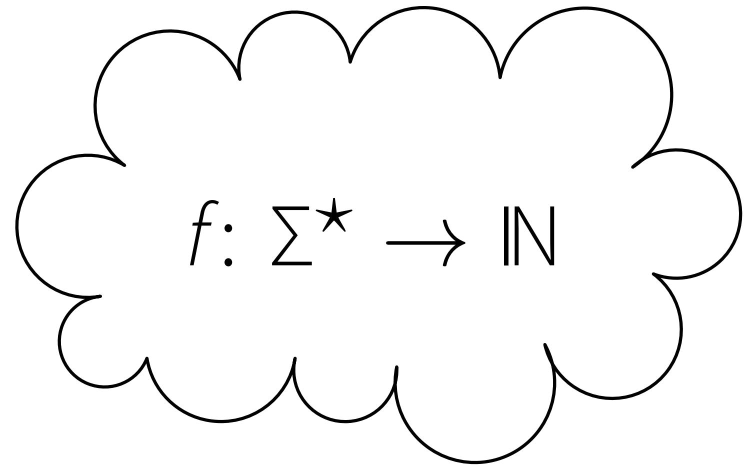
dividing membranes

Leporati, Manzoni, Mauri, Porreca, Zandron: Characterising the complexity of tissue P systems with fission rules. Journal of Computer and System Sciences 90, 115–128 (2017)

Simulating shallow membrane systems



external environment

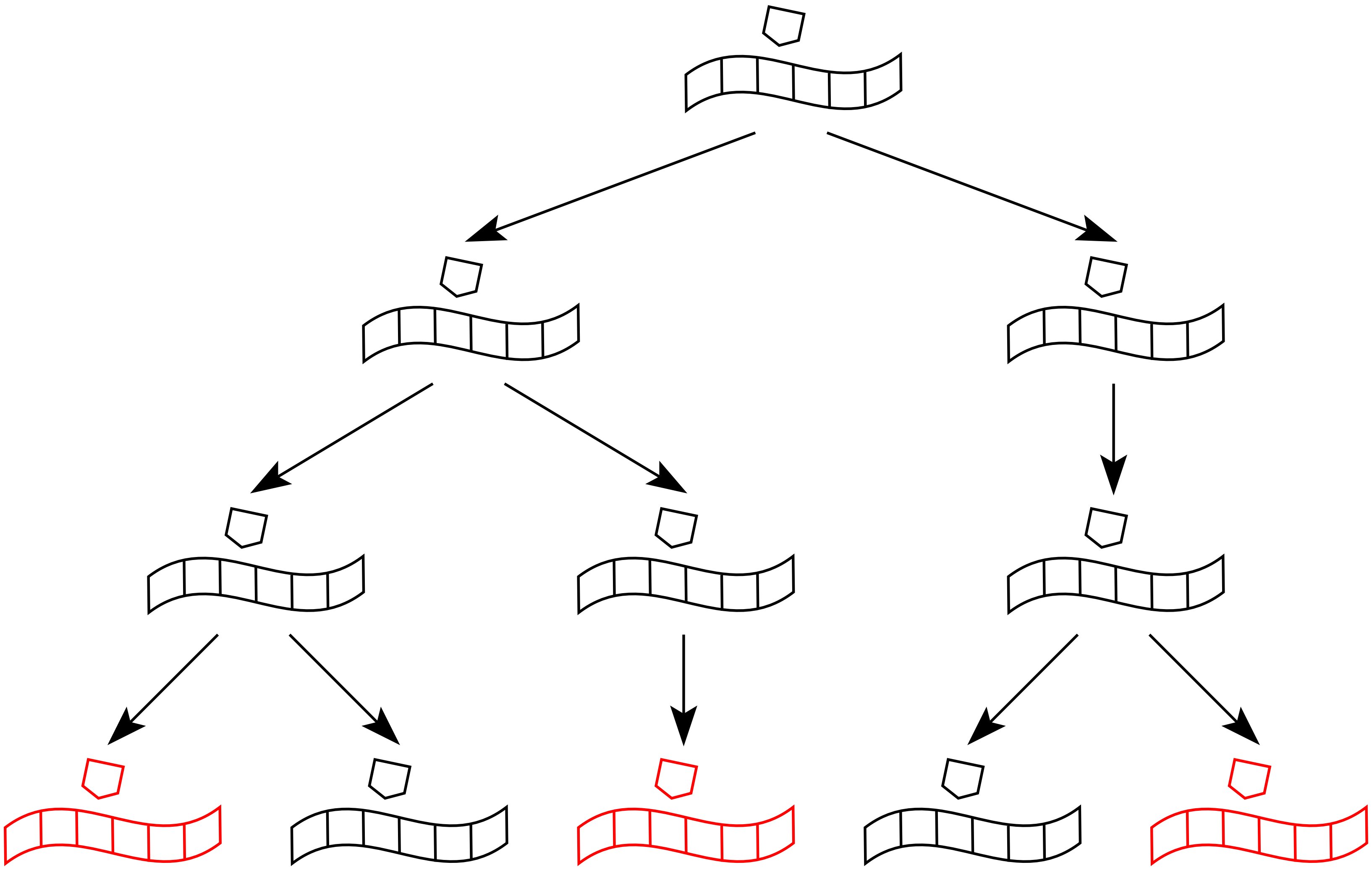


dividing membranes

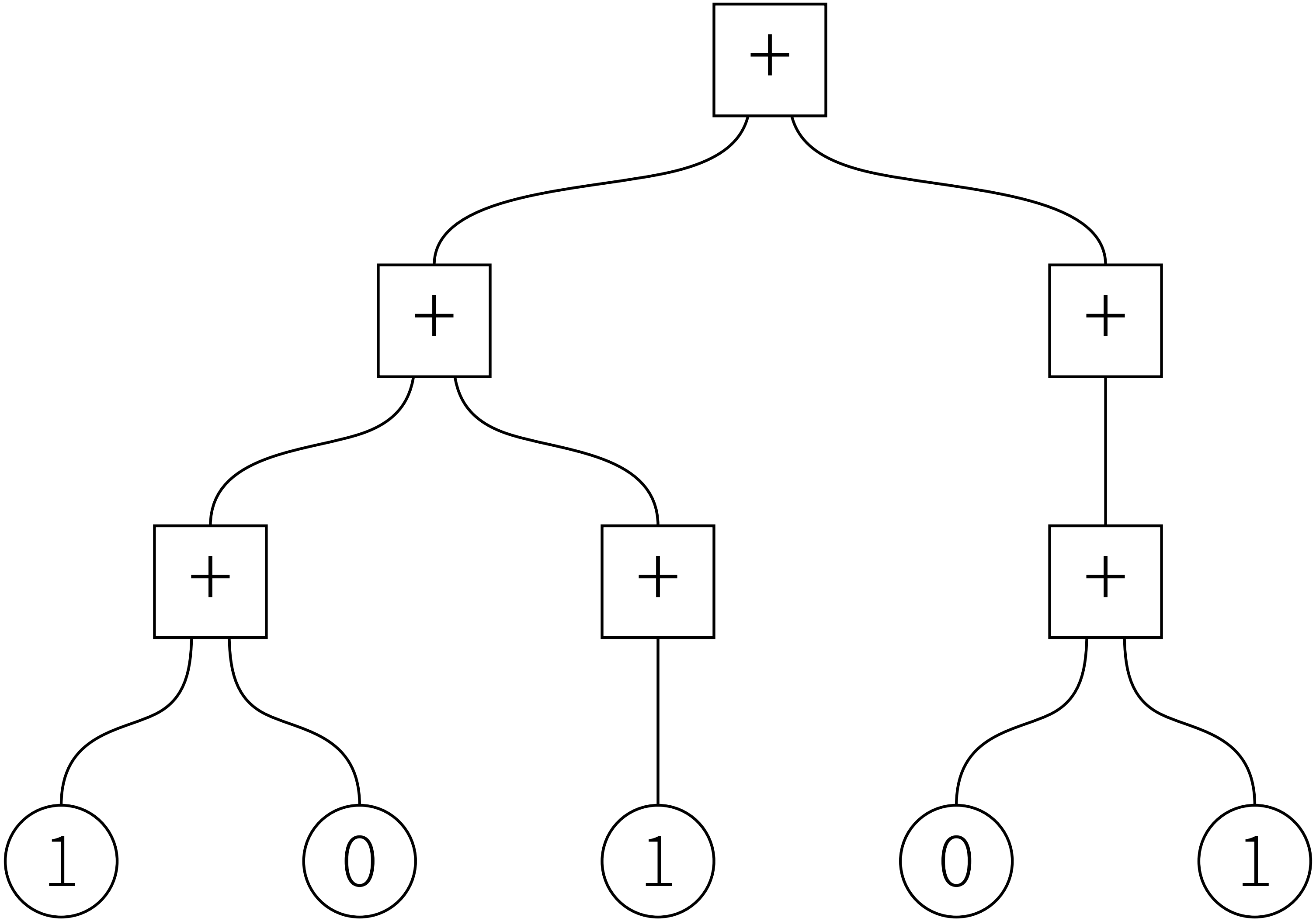
$f(x) = f(x_1, \ldots, x_t, a) =$ number of instances of a sent out at time $t+1$ by dividing membranes which receive input x_i at time $1 \leq i \leq t$

Leporati, Manzoni, Mauri, Porreca, Zandron: Characterising the complexity of tissue P systems with fission rules. Journal of Computer and System Sciences 90, 115–128 (2017)

Simulating parallelism nondeterministically



Simulating parallelism nondeterministically



Shallow membrane systems and $P^{\#P}$

The complexity class $P^{\#P}$ is exactly the class of problems solved by membrane systems with shallow (depth 1) division in polynomial time

Leporati, Manzoni, Mauri, Porreca, Zandron: Characterising the complexity of tissue P systems with fission rules. Journal of Computer and System Sciences 90, 115–128 (2017)

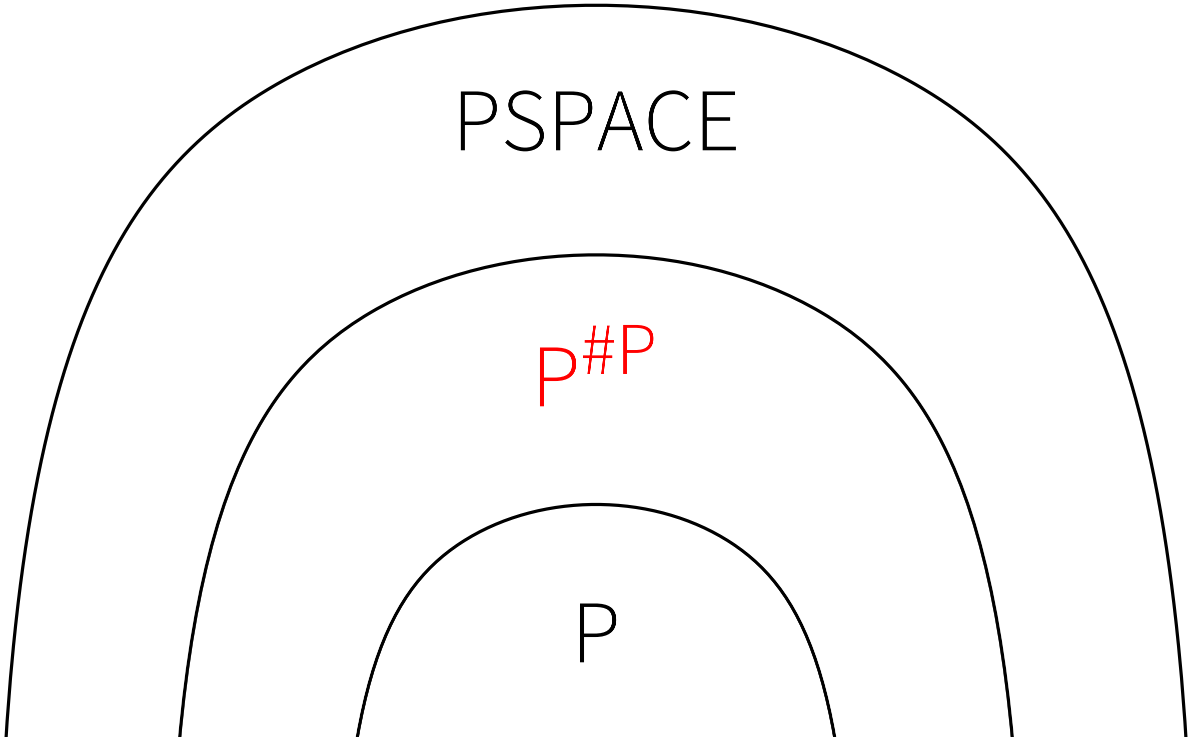
Shallow membrane systems and $P^{\#P}$

The complexity class P is exactly the class of problems solved by membrane systems without division in polynomial time

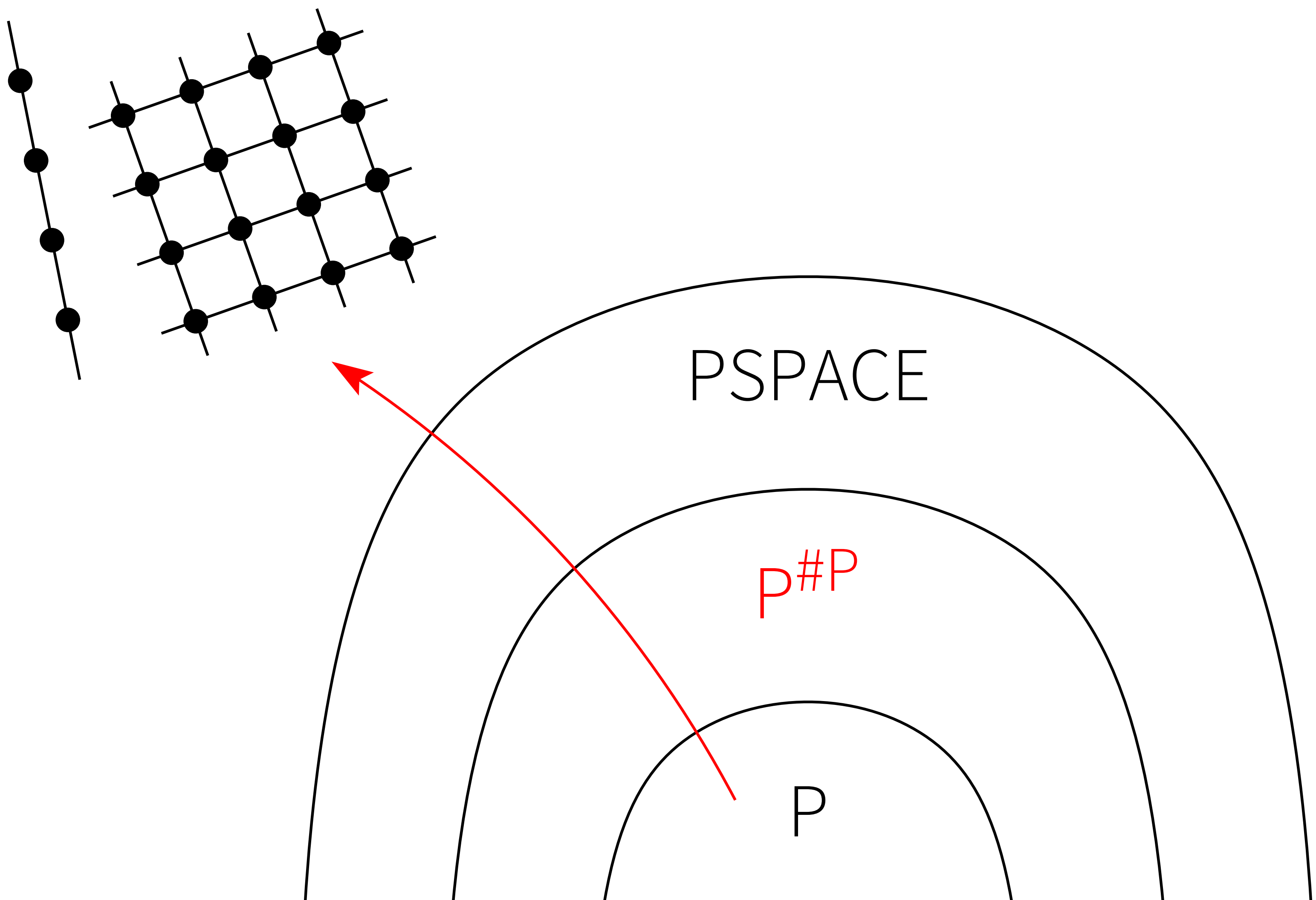
The complexity class $P^{\#P}$ is exactly the class of problems solved by membrane systems with shallow (depth 1) division in polynomial time

The complexity class PSPACE is exactly the class of problems solved by membrane systems with deep division in polynomial time

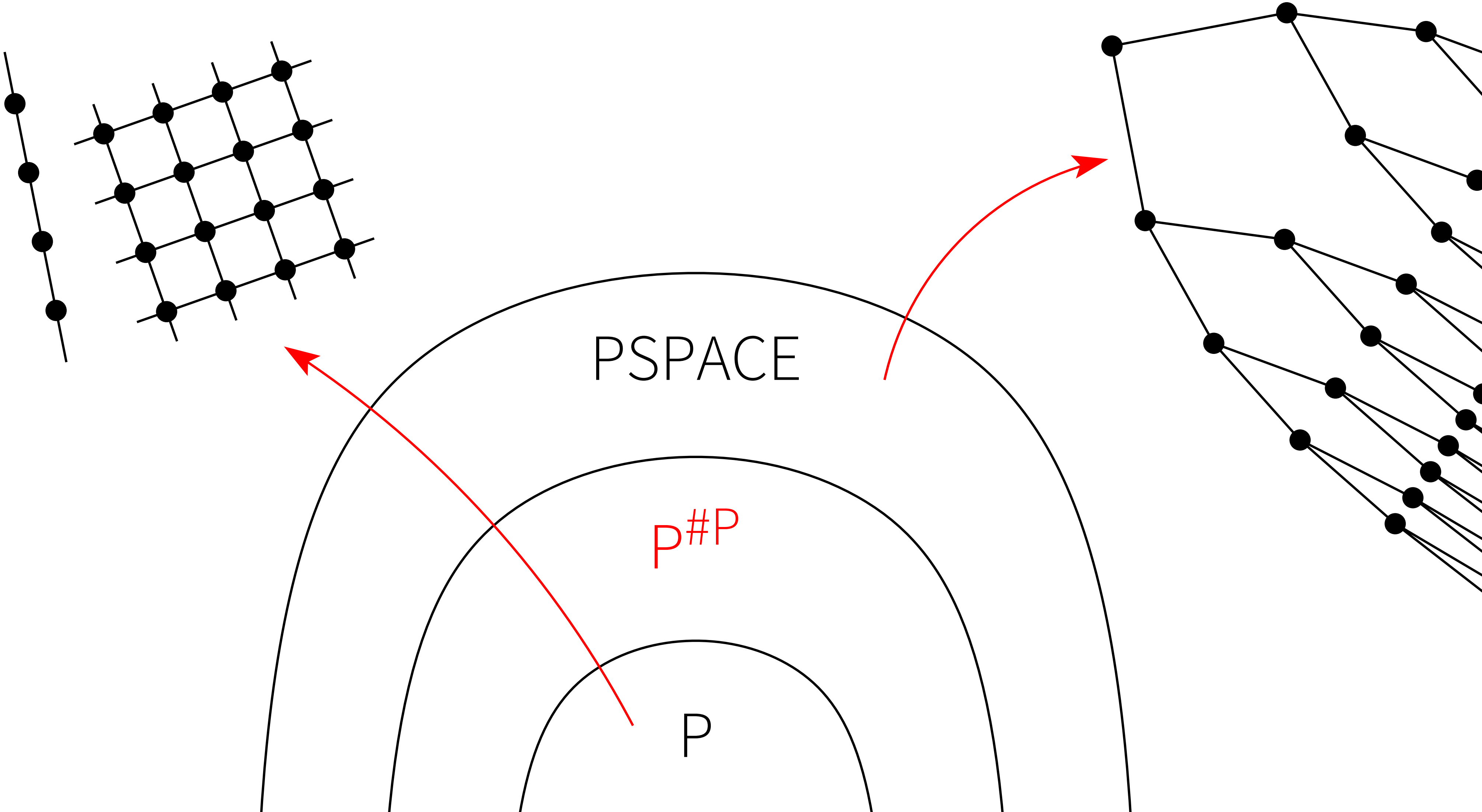
Communication topologies and complexity



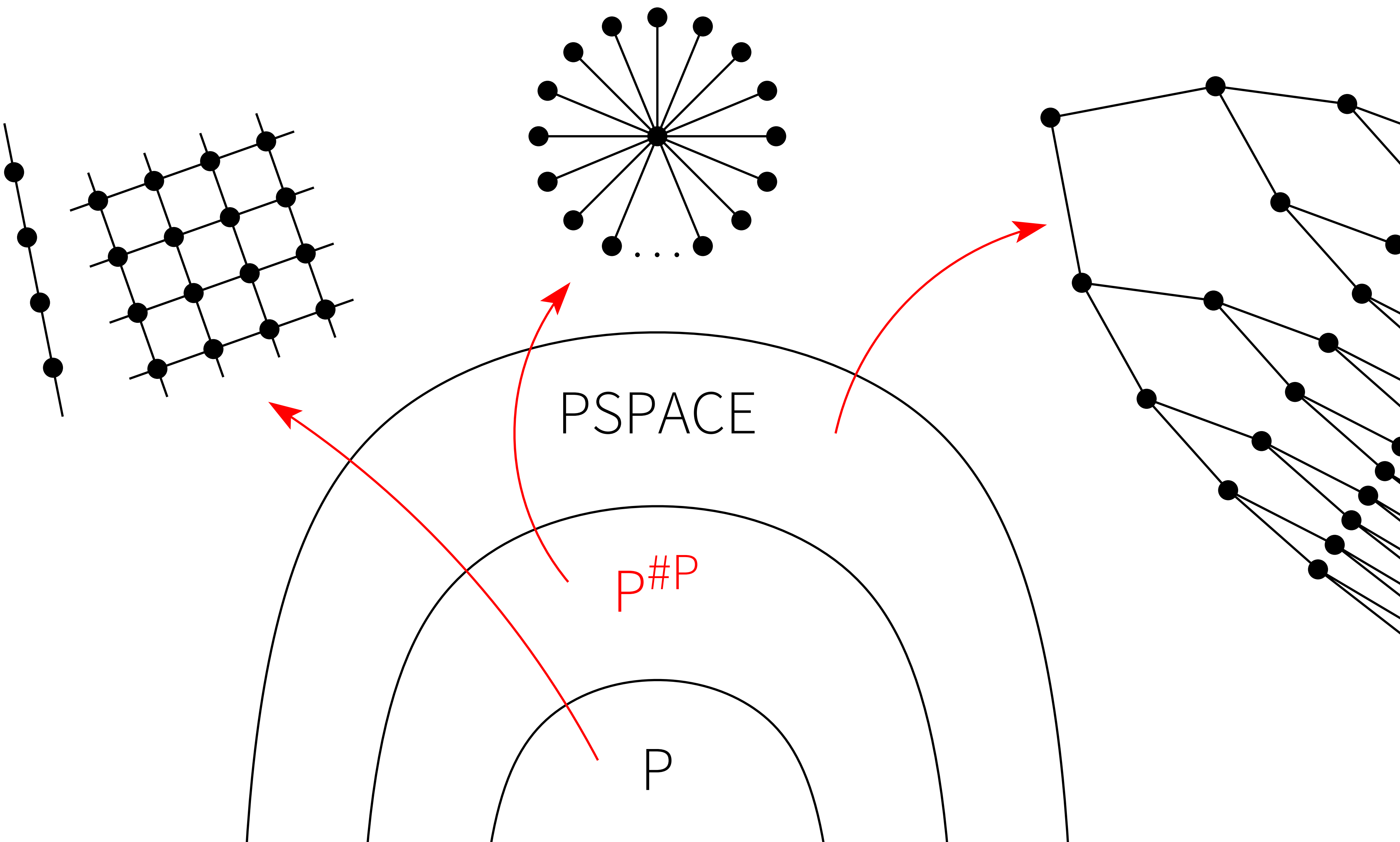
Communication topologies and complexity



Communication topologies and complexity



Communication topologies and complexity

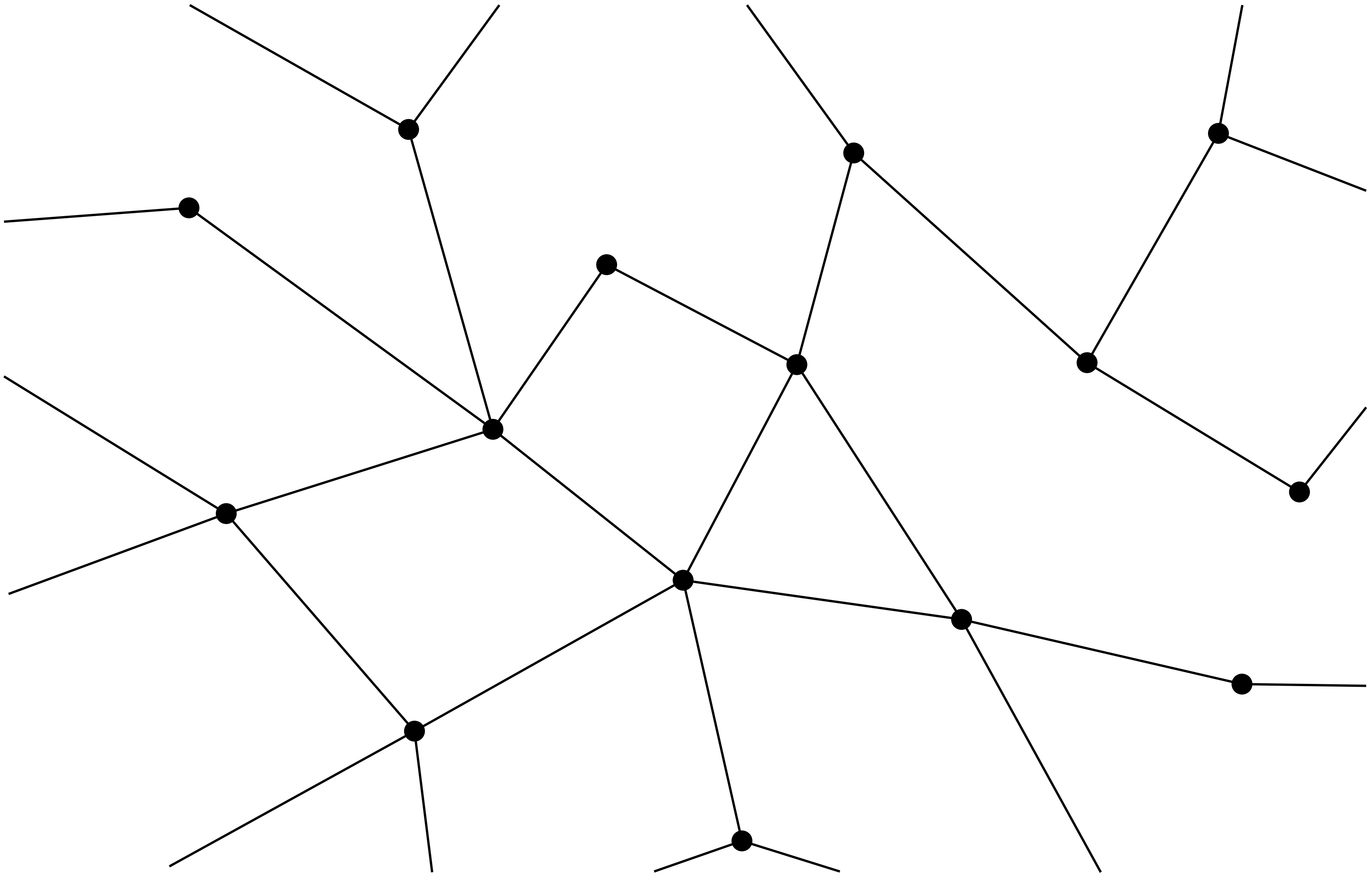


A research project

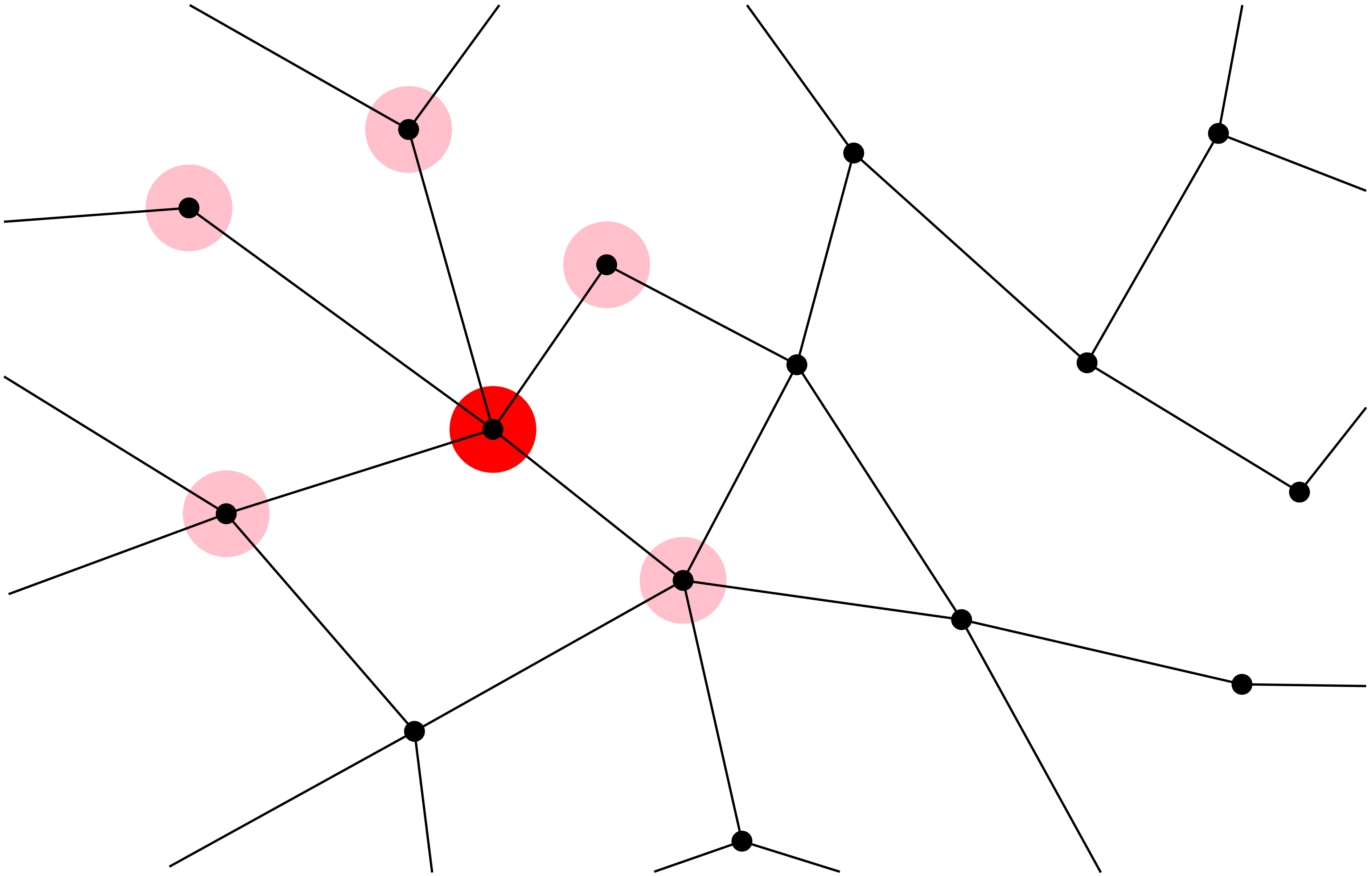
Investigate the role of communication topologies in parallel computing models, with a focus on natural computing

Which graph-theoretic, geometric, descriptive complexity properties of the communication topology influence the computational complexity?

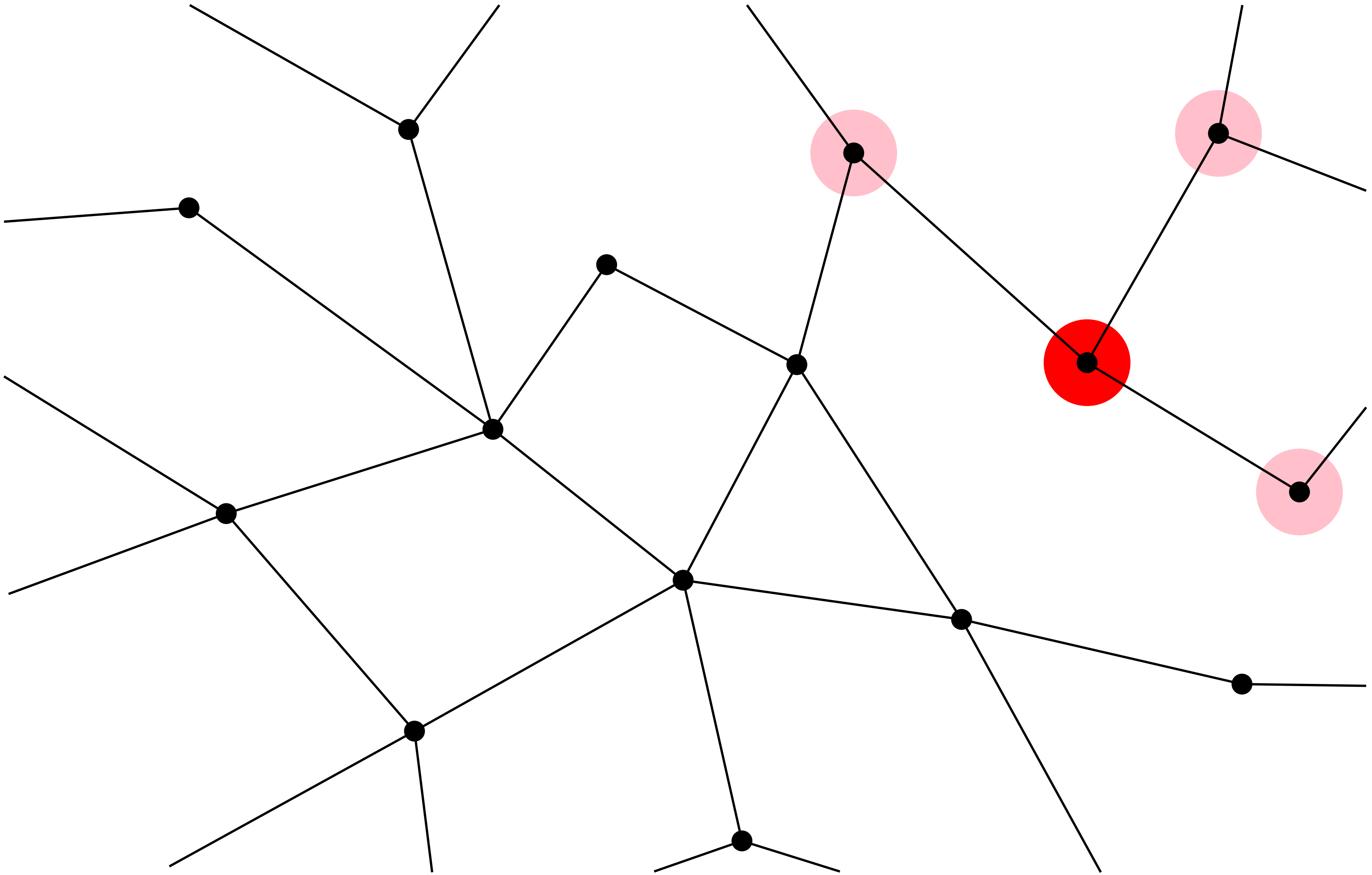
Automata networks over infinite graphs



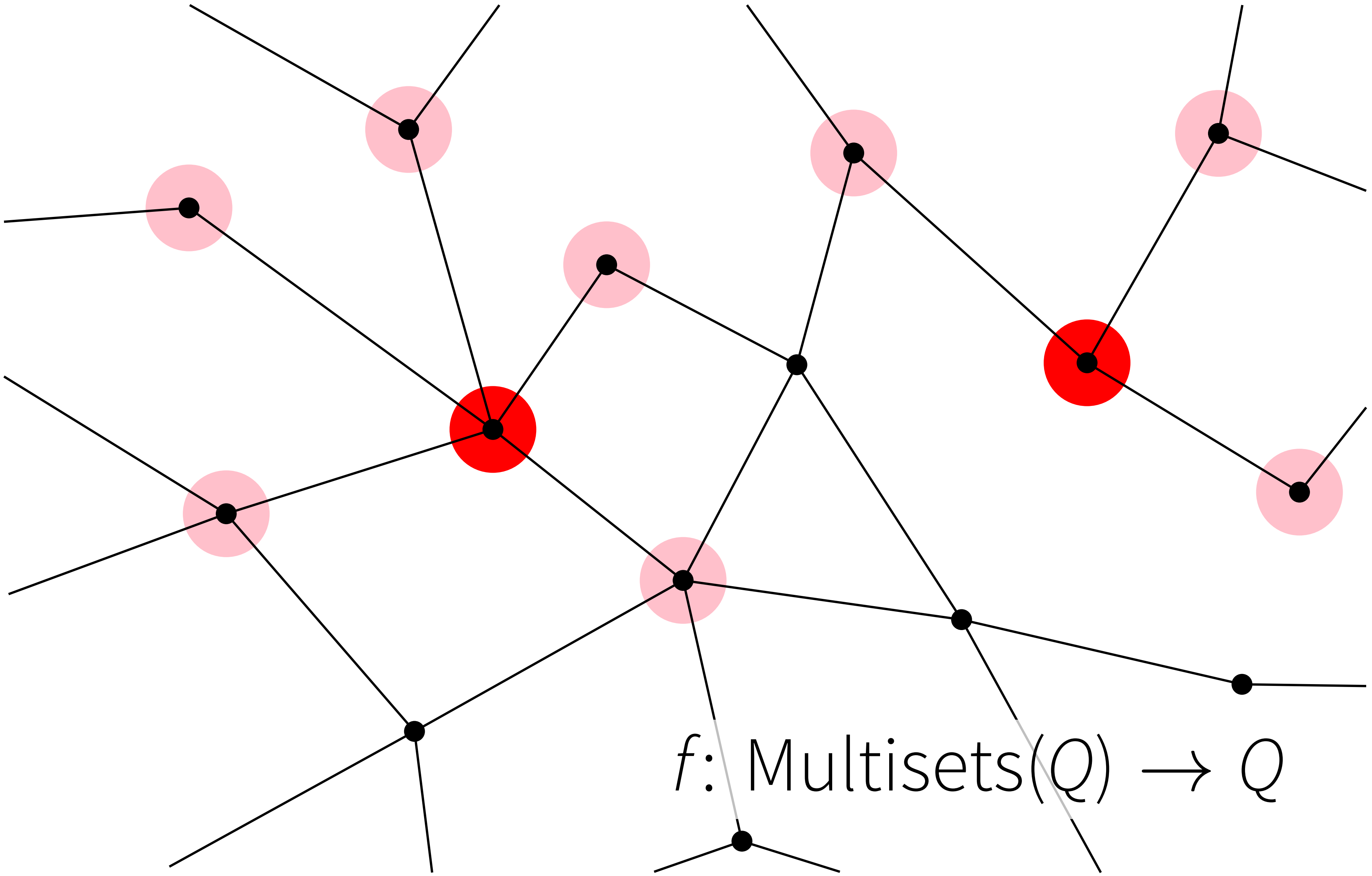
Automata networks over infinite graphs



Automata networks over infinite graphs

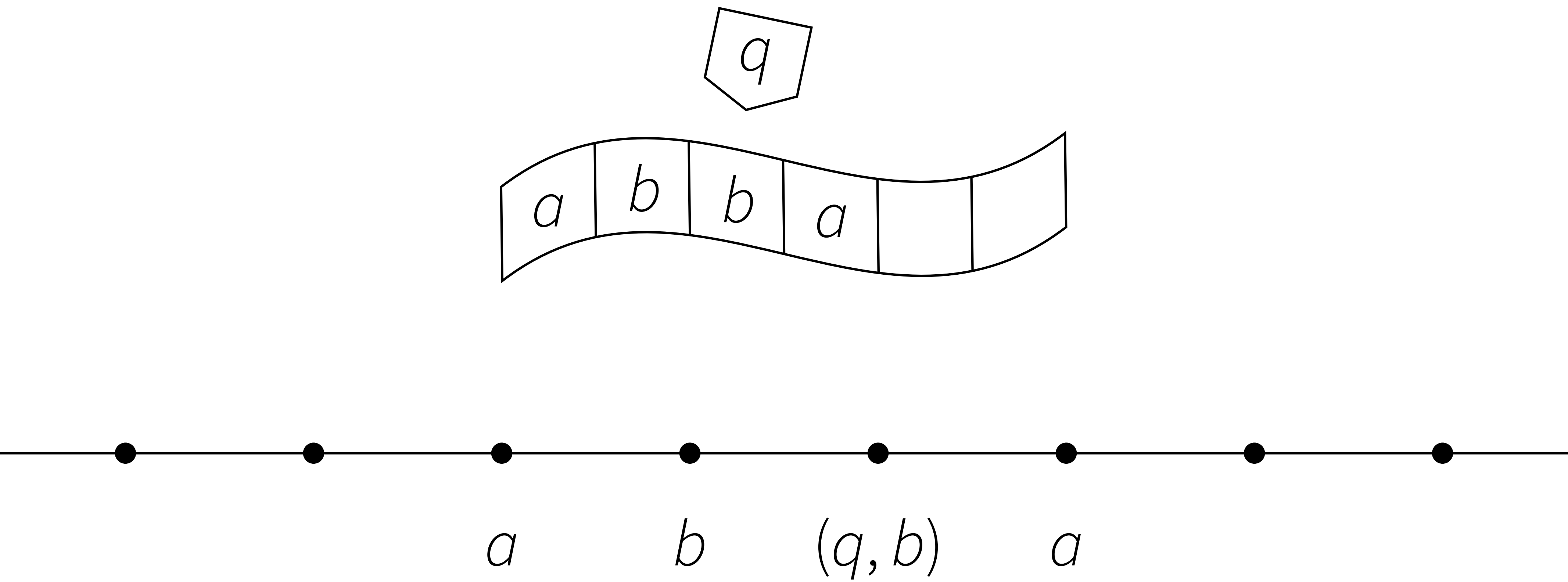


Automata networks over infinite graphs



$f\colon \mathrm{Multisets}(Q) \to Q$

1D cellular automata as generalised TMs



Descriptive complexity of infinite graphs

$$G = (V, E)$$

Descriptive complexity of infinite graphs

$\Sigma^\star$ $\qquad$ $\mathbb{N}$ $\qquad$ $\mathbb{N}^d$

$$G = (V, E)$$

$$M(v, i) = \begin{cases} i\text{-th neighbour of } v \\ \perp \text{ if it does not exist} \end{cases}$$

Things to do

- Choose restrictions on the local functions $f\colon \text{Multisets}(Q) \to Q$, e.g., threshold functions
- Choose a bound for the description complexity of the underlying graph
- Choose a way to encode the input in the initial configuration

Generalised complexity classes

- P(G) = problems solved in polynomial time over G
- PSPACE(G) = polynomial space over G
- EXPTIME(G) = exponential time over G
- ...
- LOGTIME(G) = logarithmic time over G

- NP(G) = nondeterministic polynomial time over G

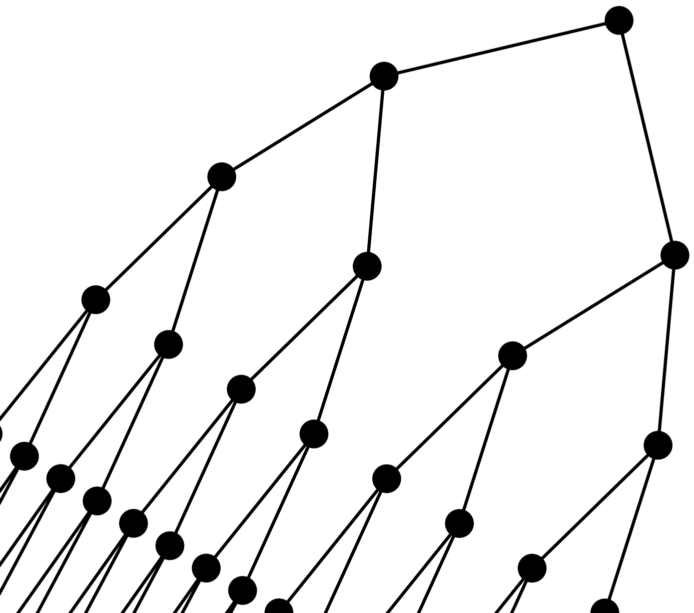
Preliminary results

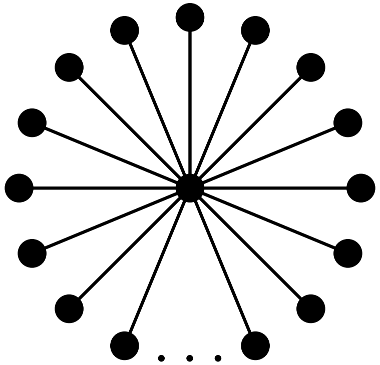
- $\mathrm{P}(G) = \mathrm{P}$
- $\mathrm{PSPACE}(G) = \mathrm{PSPACE}$
- $\mathrm{EXPTIME}(G) = \mathrm{EXPTIME}$

where G is the linear graph or, more generally, an efficiently navigable graph embeddable in the Euclidean space

Expected results

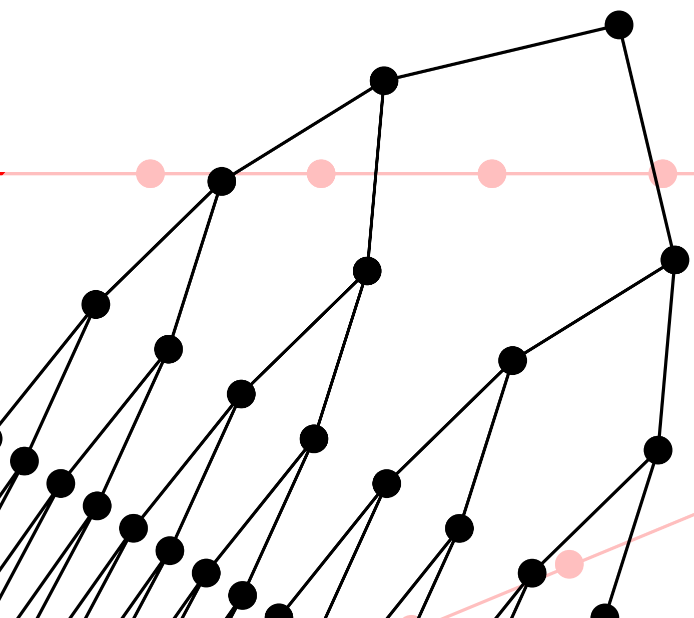
- P(infinite binary tree) = PSPACE
- P(infinite star or variant) = $P^{\#P}$
- P(non-computable graph) includes undecidable problems

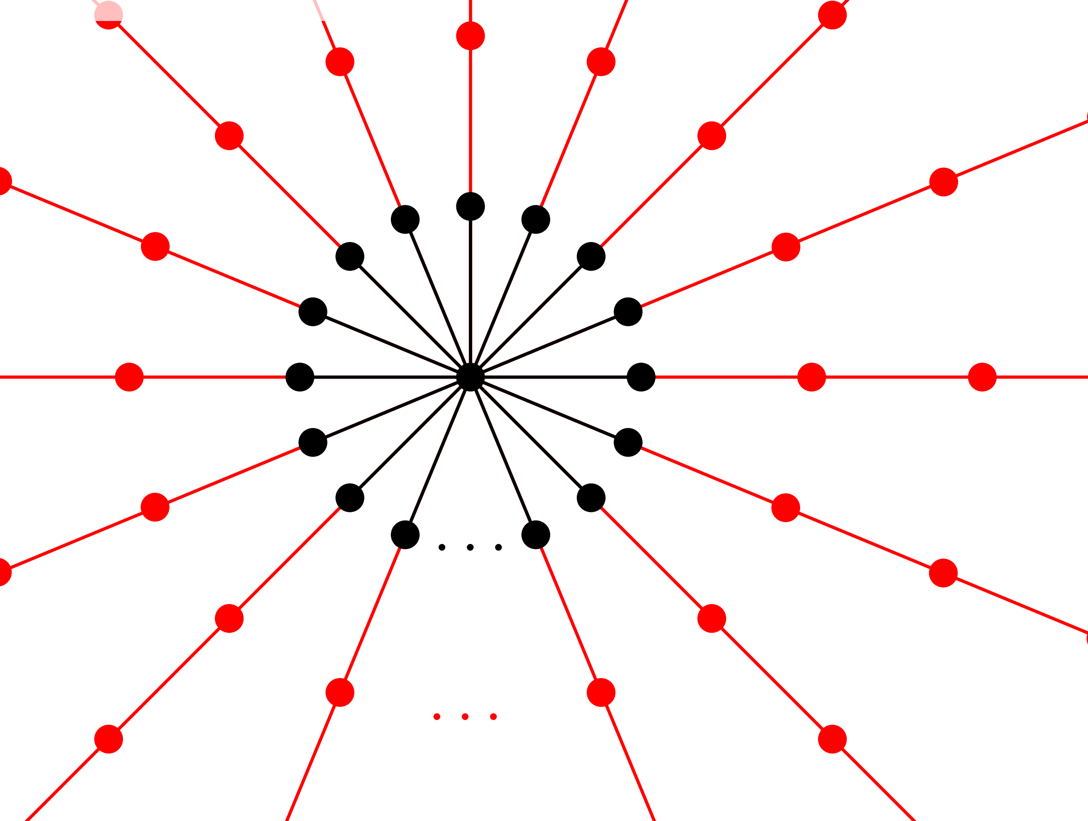




Expected results

- P(infinite binary tree) = PSPACE
- P(infinite star or variant) = $P^{\#P}$
- P(non-computable graph) includes undecidable problems





Long-term goals: theory of computation

- Find graphs characterising the standard complexity classes
- ….or prove that it is impossible
- Define new complexity classes in terms of “natural” graphs
- Examine the complexity of simulating automata networks over certain graphs

Long-term goals: distributed algorithms

- Find low-level algorithms (local rules) working with all graphs or certain families of graphs
- …or prove impossibility results
- Investigate how the graph-theoretic and geometric properties can speed up or slow down the algorithms
- Example: agorithms running in time $\Theta(n^{f(d)})$ in $\mathbb{R}^d$

Connections with other areas

- Exploit results on the dynamics of automata networks
- Provide bounds for applications of automata networks (e.g., biological modelling)
- Theory (and practice) of distributed algorithms
- Machine learning (e.g., “non-Euclidean learning”)
- Attack open problems in complexity theory

Thanks for your attention!

Merci de votre attention !

Any questions?