# Formal Framework (for P-Systems)

CS 296 – Mock Proposal / Research Direction

2019 November 18

Ren Tristan A. de la Cruz

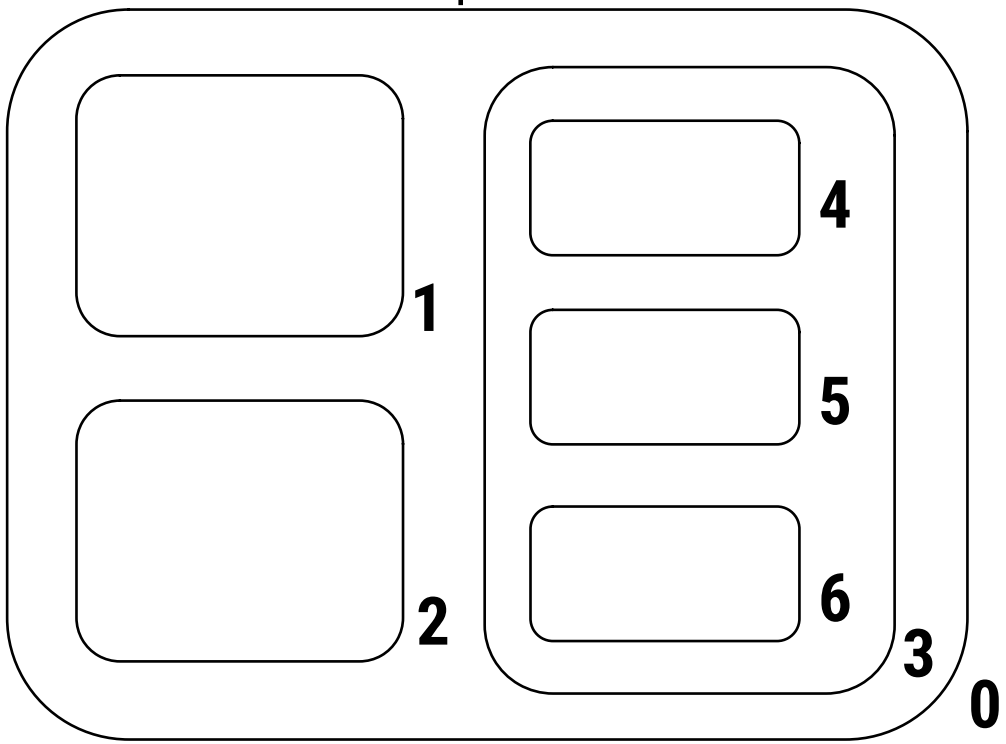
# P Systems & Membrane Computing

- *Membrane Computing* – field of theoretical CS (Natural Computing)
- Membrane computing studies different **models of computation** known as *P Systems*

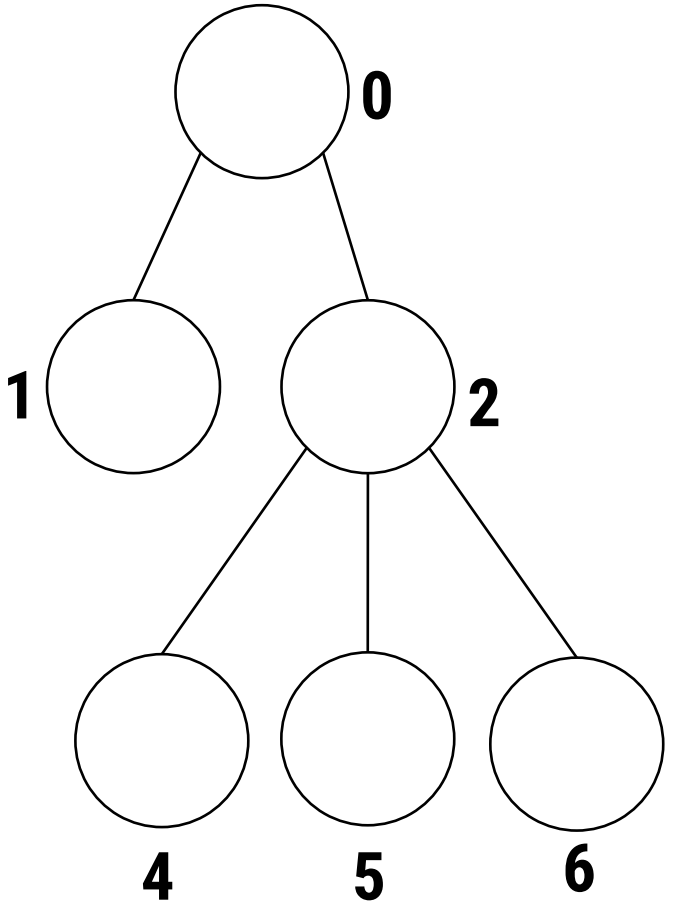
**P Systems:**
- distributed parallel computing devices
- biologically-inspired
- use the idea of membranes(and cells) to compartmentalize space
- related to multiset rewriting systems

# P Systems – Membrane Structure

Visual Representation




Tree Representation




Balanced Brackets Representation

$$[_0\ [_1]_1\ [_2]_2\ [_3\ [_4]_4\ [_5]_5\ [_6]_6\ ]_3\ ]_0$$

**[1]** *Păun*, G.. 1998. *Computing with Membranes*. In *Technical Report. Turku Centre for Computer Science*.

# P Systems – Multisets of Objects

Alphabet / Set of Objects $O = \{a,b,c\}$

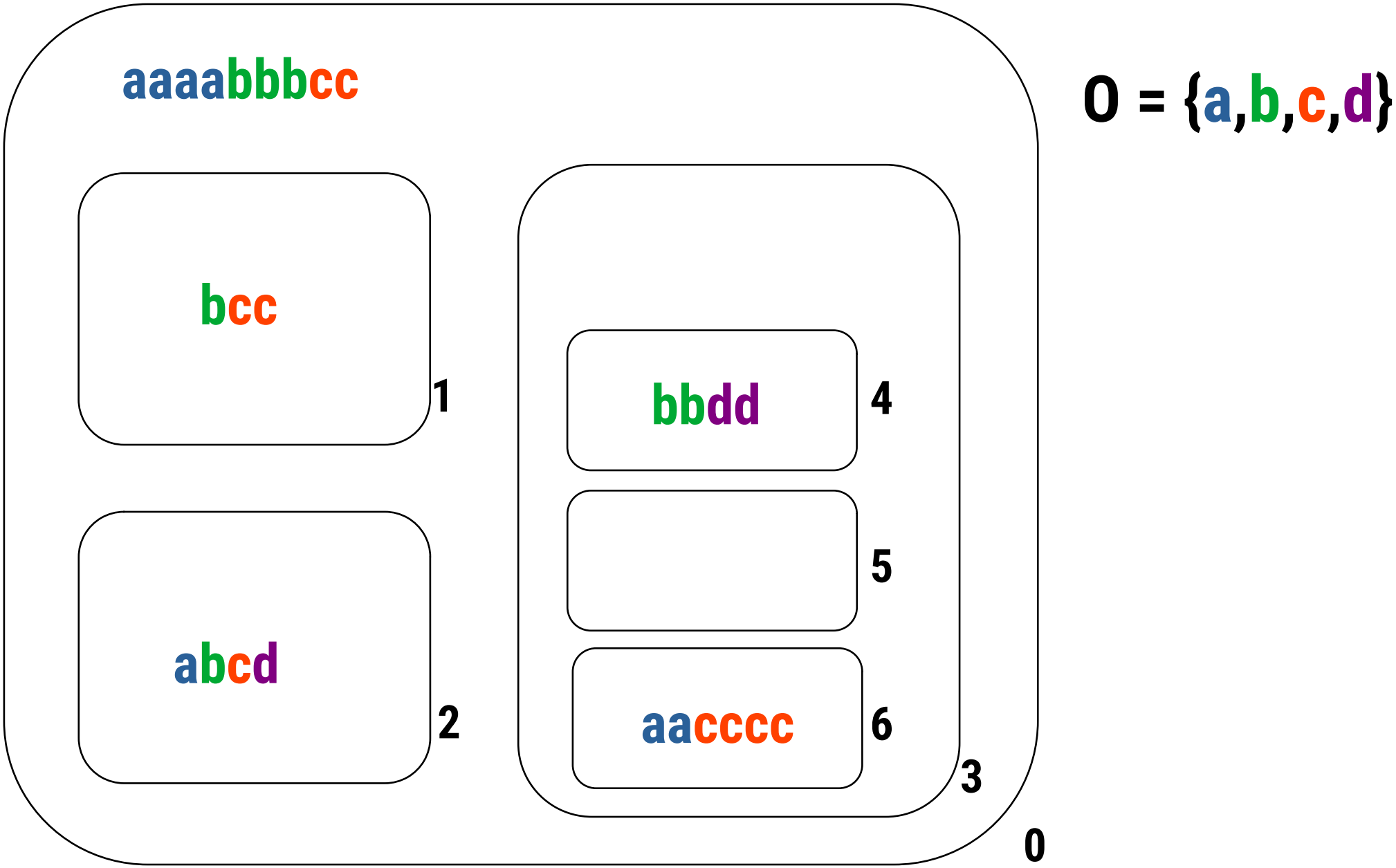
Sample Multisets over **O**

$aaaabbcc = \{a,a,a,a,b,b,c,c\} = a^4b^2c^2$

$abbccc = \{a,b,b,c,c,c\} = a^1b^2c^3$

$ac = \{a,c\} = a^1c^1 = a^1b^0c^1$

# P Systems – Multisets in Membranes



# P Systems – Multiset Rewriting

Alphabet / Set of Objects $O = \{a,b,c\}$

| | | |
|---|---|---|
| aaaabbcc | $bc \rightarrow aaa$ | $a^7bc$ |
| acc | $ac \rightarrow bb$ | bbc |
| aabbbccc | $ab \rightarrow aab$ | aaabbbccc |

# P Systems – Multiset Rewriting

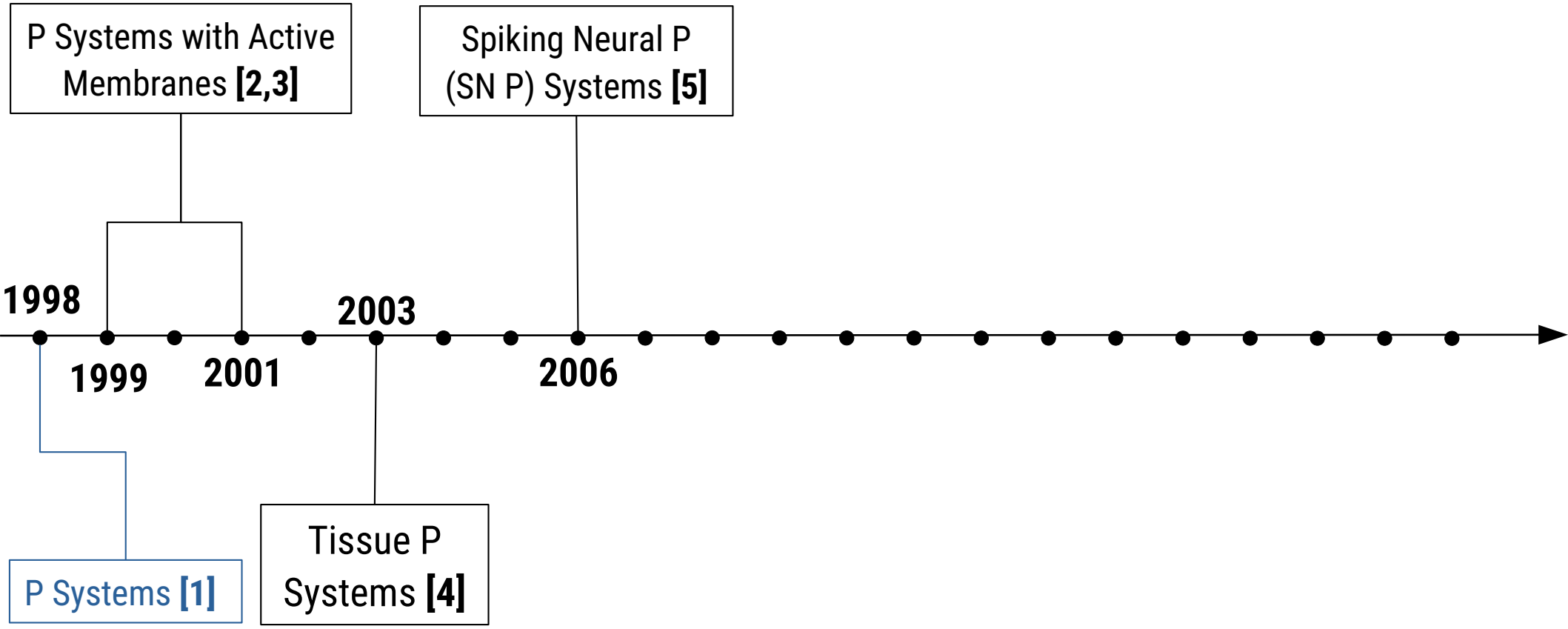
Alphabet / Set of Objects **O = {a,b,c}**

**aaaabbcc** **bc** $\xrightarrow{x2}$ **aaa** **$a^{10}$**

**acc** **ac** $\rightarrow$ **bb** **bbc**

**aabbbccc** **ab** $\xrightarrow[x2]{}$ **aab** **aaaabbbccc**

# Membrane Computing - P Systems




**[1]** *Păun*, G.. 1998. *Computing with Membranes*. In *Technical Report. Turku Centre for Computer Science*.

**[2]** *Păun*, G.. 1999. *P Systems with Active Membranes: Attacking NP-Complete Problems*. In *Centre for Discrete Mathematics and Theoretical Computer Science* (CDMTCS-102) – Research Report Series

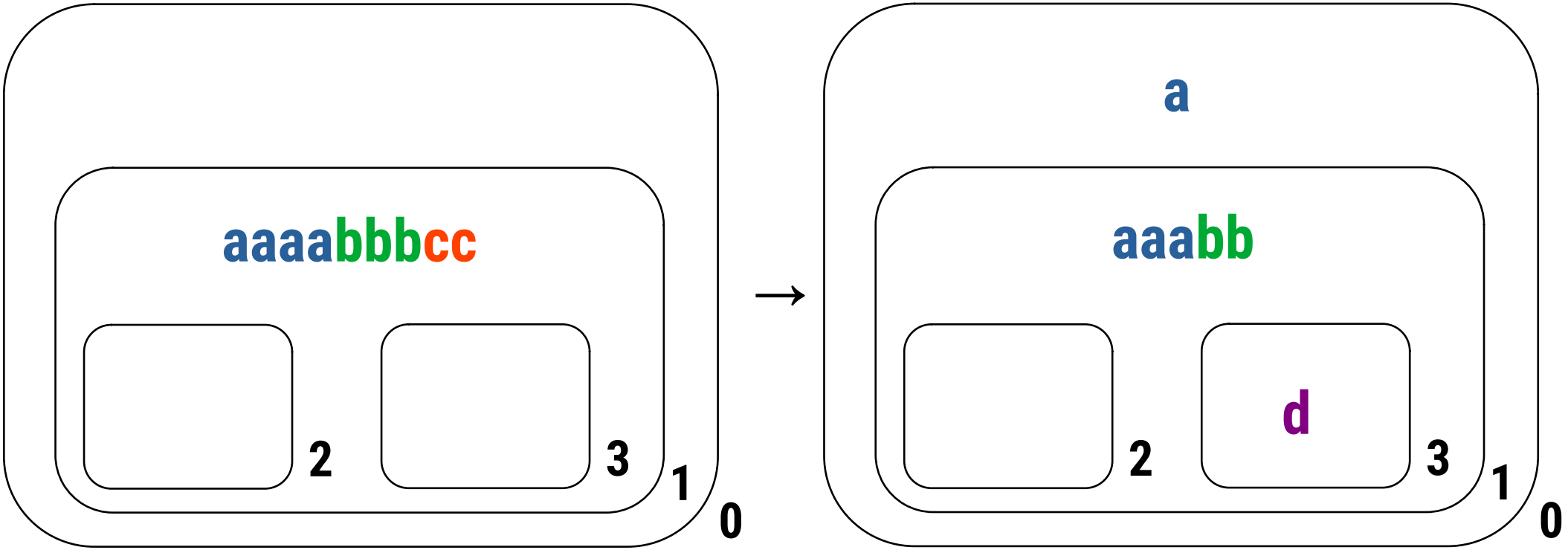
**[3]** *Păun*, G.. 2001. *P Systems with Active Membranes: Attacking NP-Complete Problems*. In *Journal of Automata, Languags, and Combinatorics.* vol.6, issue 1 (January 2001), 75-90.

**[4]** *Martín-Videa*, C., *Păun*, G., *Pazos*, J., *Rodríguez-Patón*, A.. 2003. *Tissue P Systems*. In *Theoretical Computer Science.* Elsevier.

**[5]** *Ionescu*, M., Păun, G., *Yokomori*, T.. 2006. *Spiking Neural P Systems*. In *Fundamenta Informaticae*. vol 71, issue 2,3 (February 2006), 279-308.

# P Systems – Rules

$O = \{a,b,c,d\}$



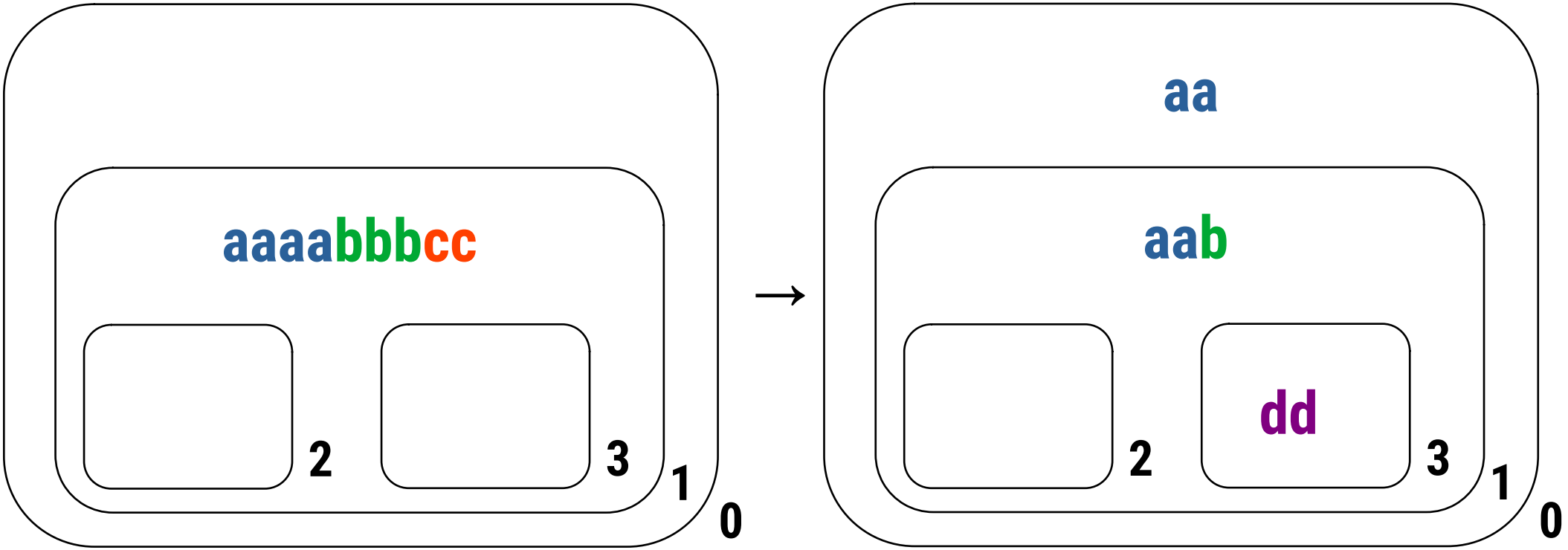

**Rule in 1:** $abcc \rightarrow (a, out)(d, in_3)$

$[abcc]_1 \rightarrow [_0\ a\ ]_0\ [_3\ d\ ]_3$

**[1]** *Păun*, G.. 1998. *Computing with Membranes*. In *Technical Report. Turku Centre for Computer Science*.

# P Systems – Rules

$O = \{a,b,c,d\}$



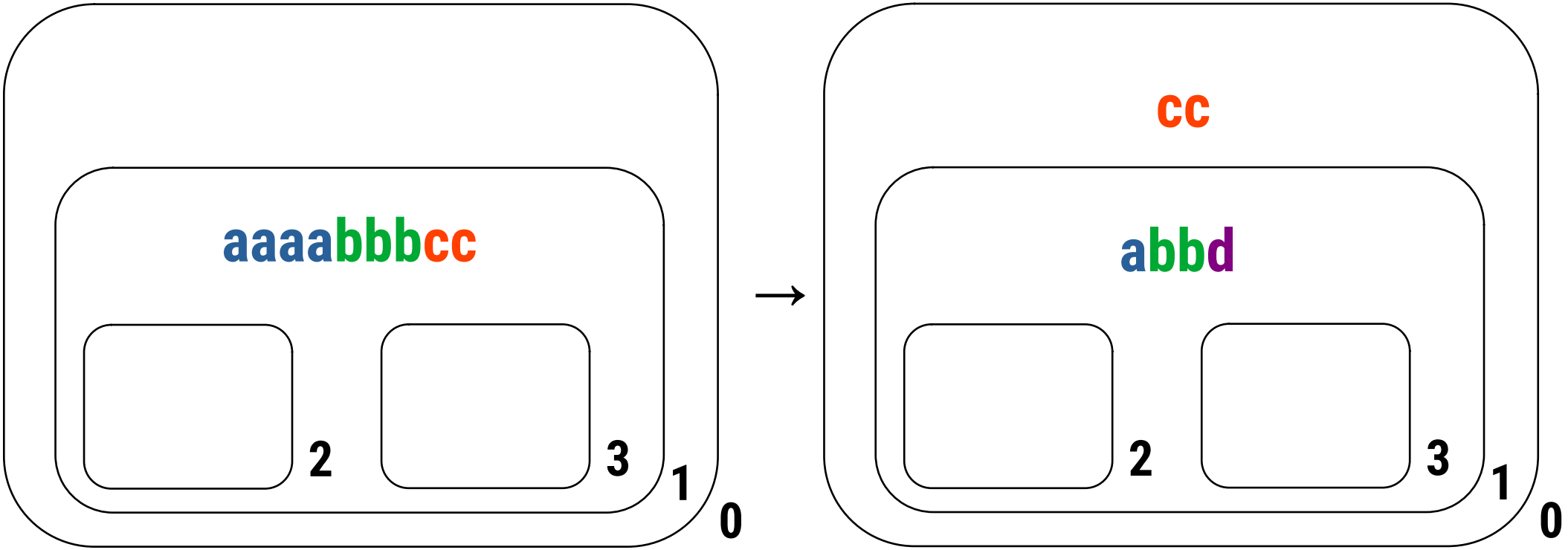

**Rule in 1:** $abc \rightarrow (a, out)(d, in_3)$ x2

$[abc]_1 \rightarrow [_0\ a\ ]_0\ [_3\ d\ ]_3$ x2

**[1]** *Păun*, G.. 1998. *Computing with Membranes*. In *Technical Report. Turku Centre for Computer Science*.

# P Systems – Rules

$O = \{a,b,c,d\}$



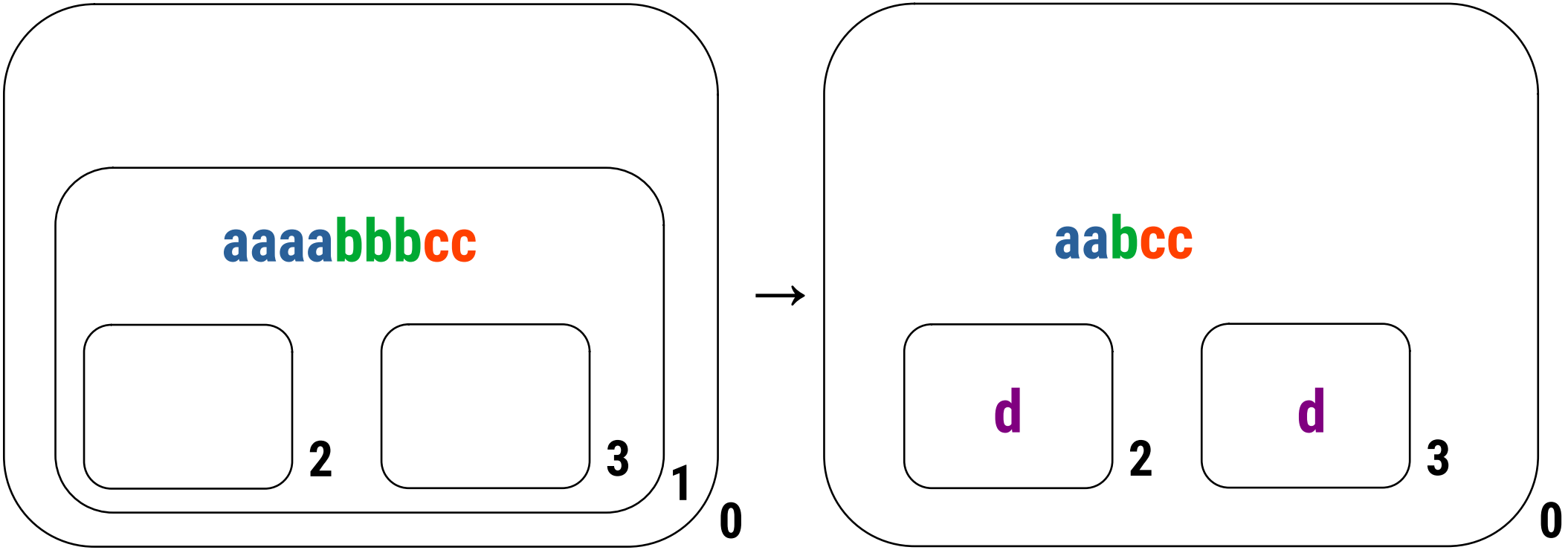

**Rule in 1:** $aaabcc \rightarrow (cc,out)(d,here)$

$$[aabcc]_1 \rightarrow [_0\ cc\ ]_0\ [_1\ d\ ]_1$$

**[1]** *Păun*, G.. 1998. *Computing with Membranes*. In *Technical Report. Turku Centre for Computer Science*.

# P Systems – Rules

$O = \{a,b,c,d\}$

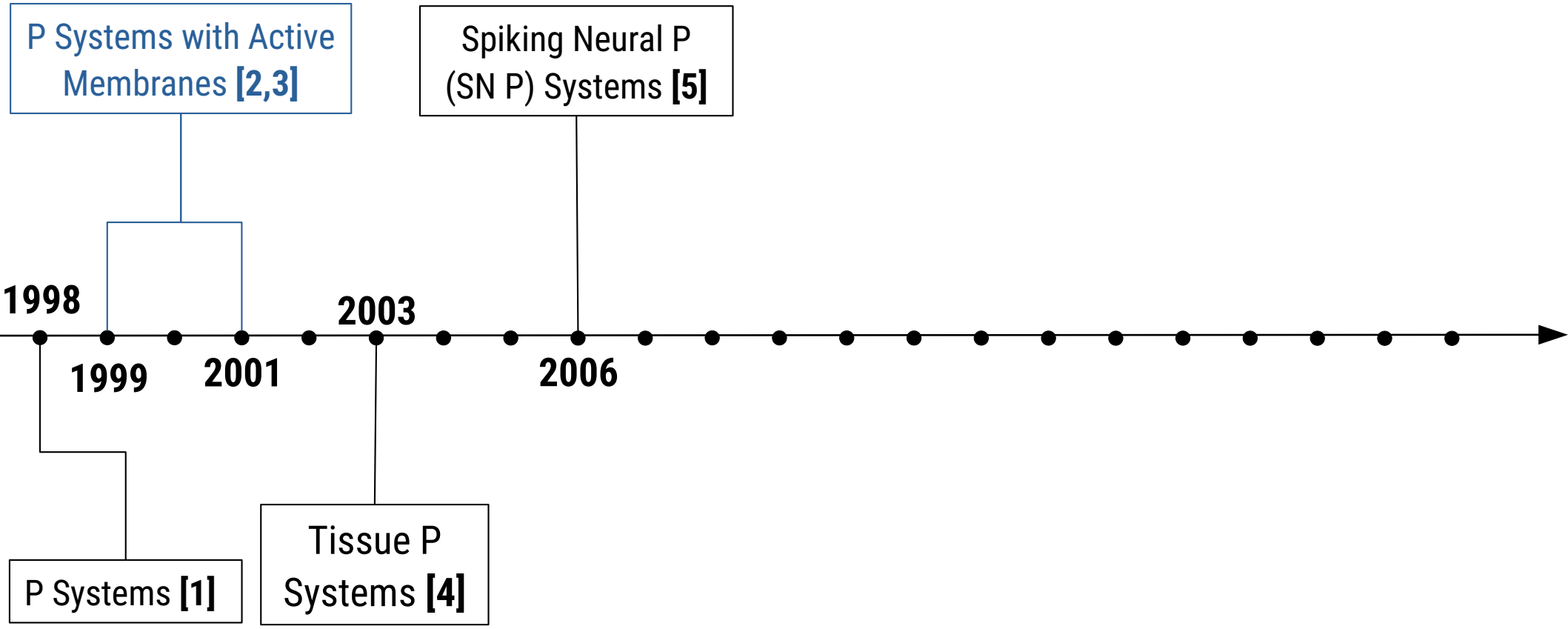



**Rule in 1:** $aabbcc \rightarrow (d,in_2)(d,in_3)\delta$

$[aabbcc]_1 \rightarrow [_2\ d\ ]_2\ [_3\ d\ ]_3\ [_1\delta\ ]_1$

**[1]** *Păun*, G.. 1998. *Computing with Membranes*. In *Technical Report. Turku Centre for Computer Science*.

# Membrane Computing - P Systems




**[1]** *Păun*, G.. 1998. *Computing with Membranes*. In *Technical Report. Turku Centre for Computer Science*.

**[2]** *Păun*, G.. 1999. *P Systems with Active Membranes: Attacking NP-Complete Problems*. In *Centre for Discrete Mathematics and Theoretical Computer Science* (CDMTCS-102) – Research Report Series

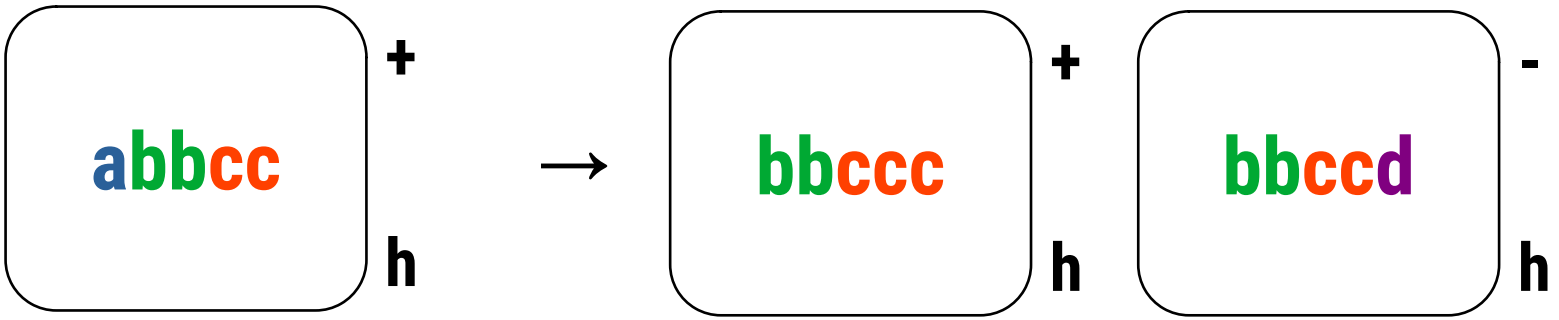
**[3]** *Păun*, G.. 2001. *P Systems with Active Membranes: Attacking NP-Complete Problems*. In *Journal of Automata, Languags, and Combinatorics.* vol.6, issue 1 (January 2001), 75-90.

**[4]** *Martín-Videa*, C., *Păun*, G., *Pazos*, J., *Rodríguez-Patón*, A.. 2003. *Tissue P Systems*. In *Theoretical Computer Science.* Elsevier.

**[5]** *Ionescu*, M., Păun, G., *Yokomori*, T.. 2006. *Spiking Neural P Systems*. In *Fundamenta Informaticae*. vol 71, issue 2,3 (February 2006), 279-308.

# P Systems with Active Membranes

**O = {a,b,c,d}  P = {+,-,0}**




$$[a]_h^+ \rightarrow [_h\ c\ ]_h^+\ [_h\ d\ ]_h^-$$

**[2]** *Păun*, G.. 1999. *P Systems with Active Membranes: Attacking NP-Complete Problems*. In *Centre for Discrete Mathematics and Theoretical Computer Science* (CDMTCS-102) – Research Report Series

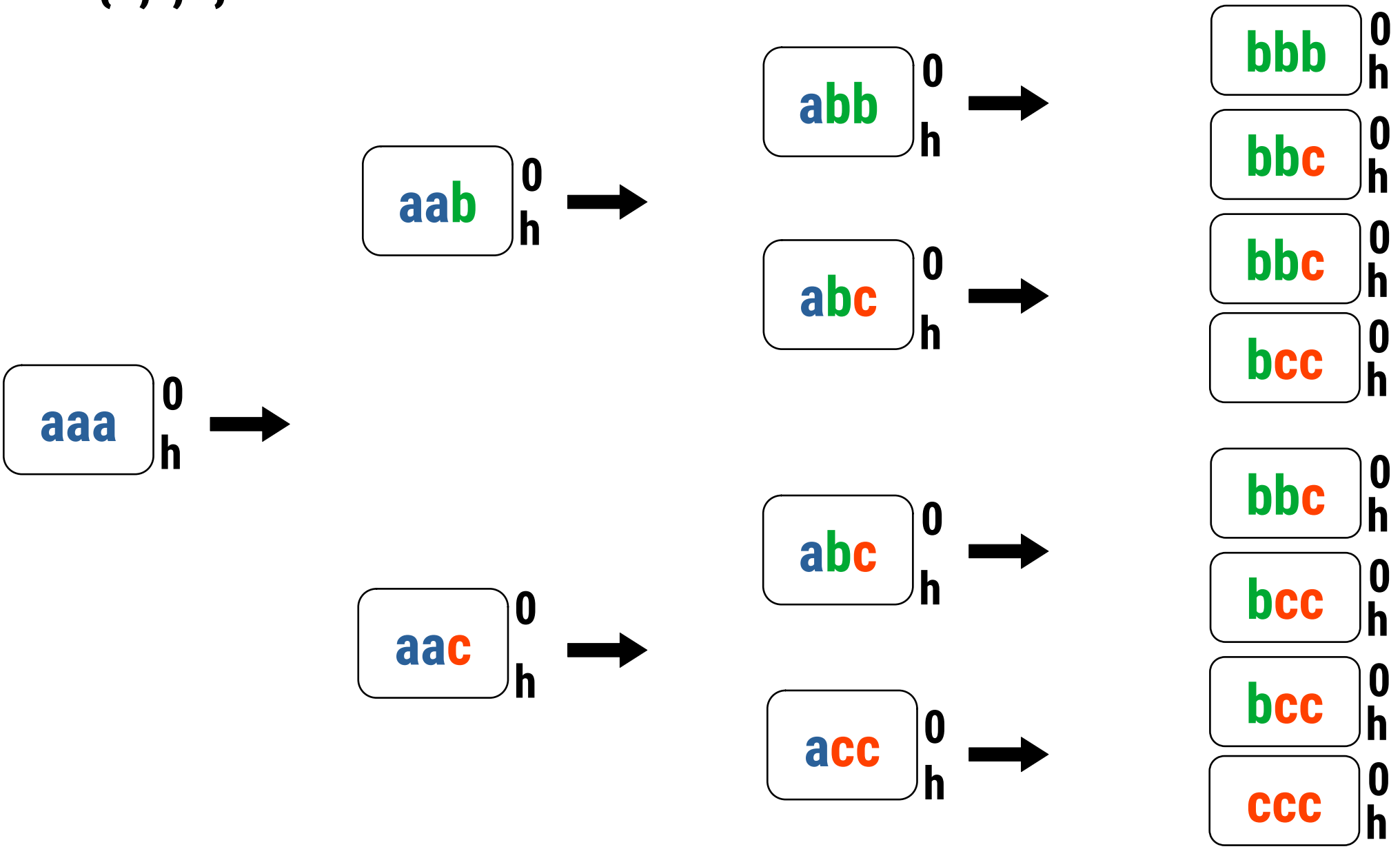
**[3]** *Păun*, G.. 2001. *P Systems with Active Membranes: Attacking NP-Complete Problems*. In *Journal of Automata, Languags, and Combinatorics.* vol.6, issue 1 (January 2001), 75-90.

# P Systems with Active Membranes

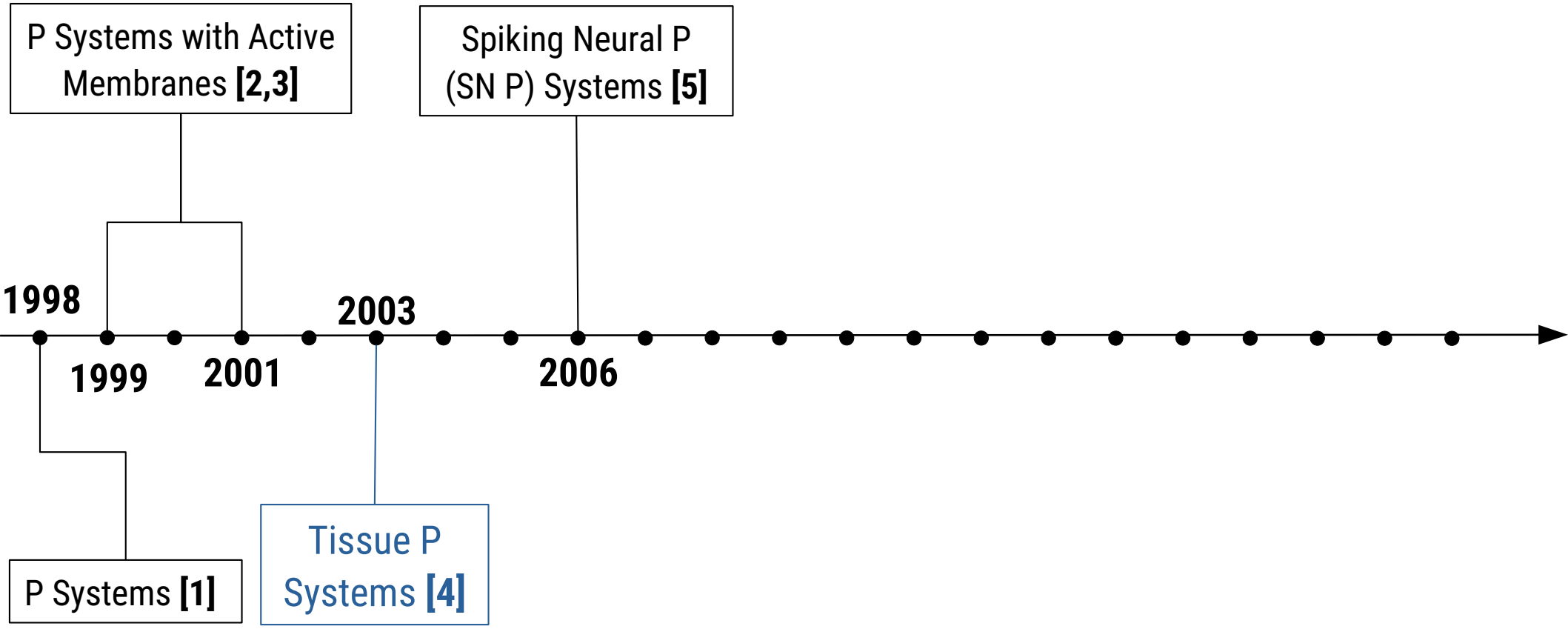
O = {a,b,c}

P = {+,-,0}

$[a]_h^0 \rightarrow [_h\ b\ ]_h^0\ [_h\ c\ ]_h^0$



# Membrane Computing - P Systems




**[1]** *Păun*, G.. 1998. *Computing with Membranes*. In *Technical Report. Turku Centre for Computer Science*.

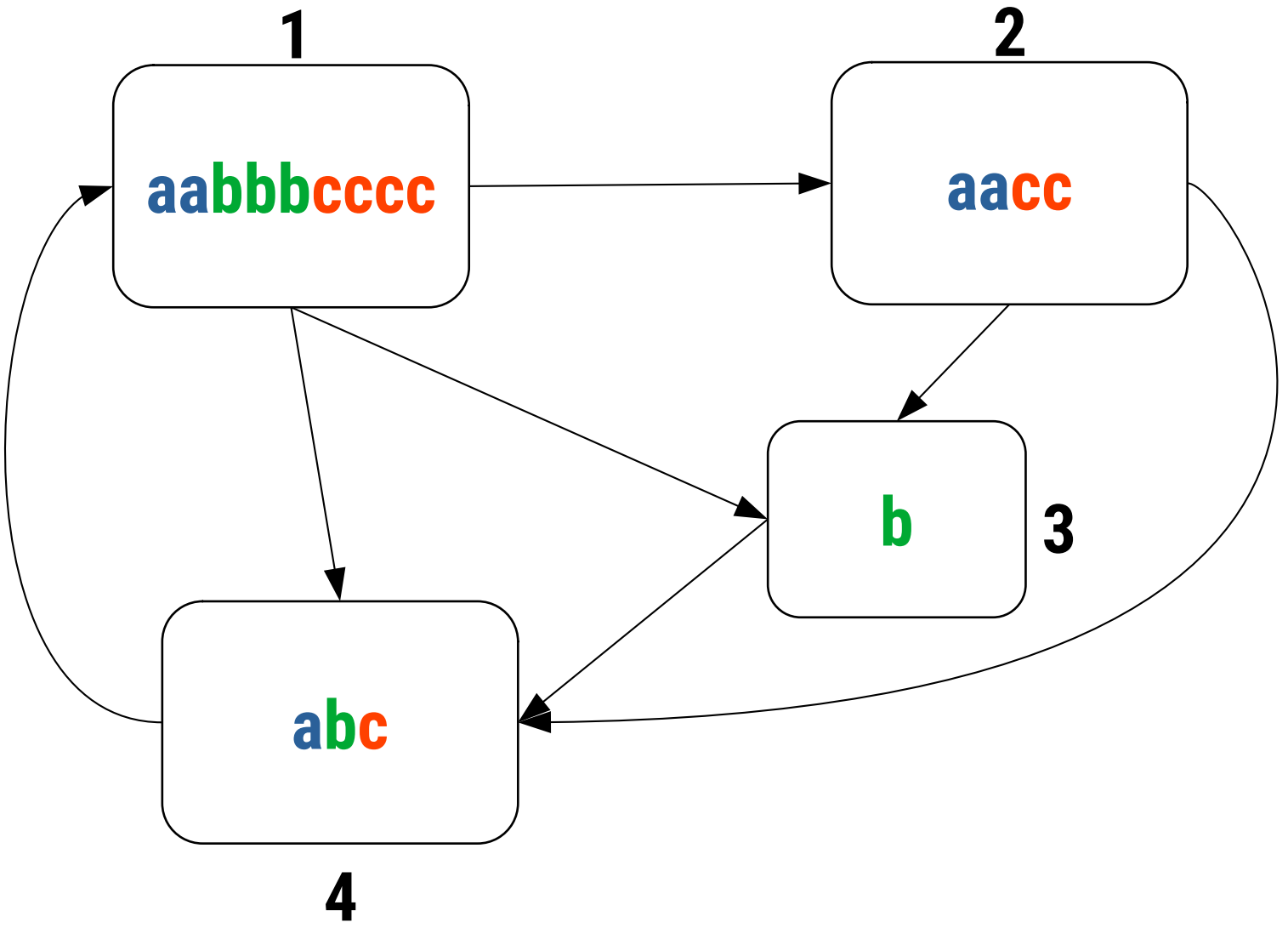
**[2]** *Păun*, G.. 1999. *P Systems with Active Membranes: Attacking NP-Complete Problems*. In *Centre for Discrete Mathematics and Theoretical Computer Science* (CDMTCS-102) – Research Report Series

**[3]** *Păun*, G.. 2001. *P Systems with Active Membranes: Attacking NP-Complete Problems*. In *Journal of Automata, Languags, and Combinatorics.* vol.6, issue 1 (January 2001), 75-90.

**[4]** *Martín-Videa*, C., *Păun*, G., *Pazos*, J., *Rodríguez-Patón*, A.. 2003. *Tissue P Systems*. In *Theoretical Computer Science.* Elsevier.

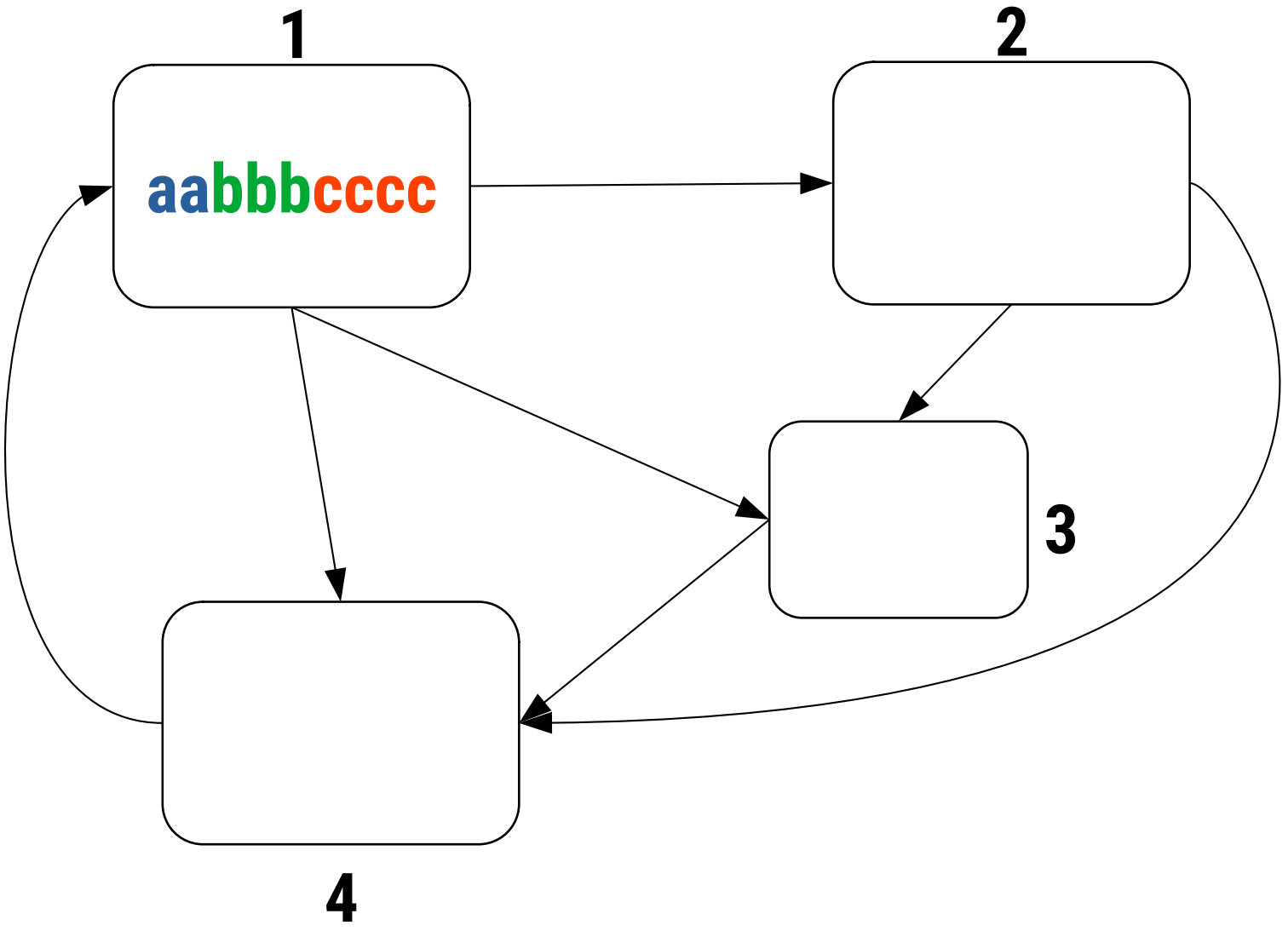
**[5]** *Ionescu*, M., Păun, G., *Yokomori*, T.. 2006. *Spiking Neural P Systems*. In *Fundamenta Informaticae*. vol 71, issue 2,3 (February 2006), 279-308.

# (Static) Tissue P Systems




**[4]** *Martín-Videa*, C., *Păun*, G., *Pazos*, J., *Rodríguez-Patón*, A.. 2003. *Tissue P Systems*. In *Theoretical Computer Science*. Elsevier.

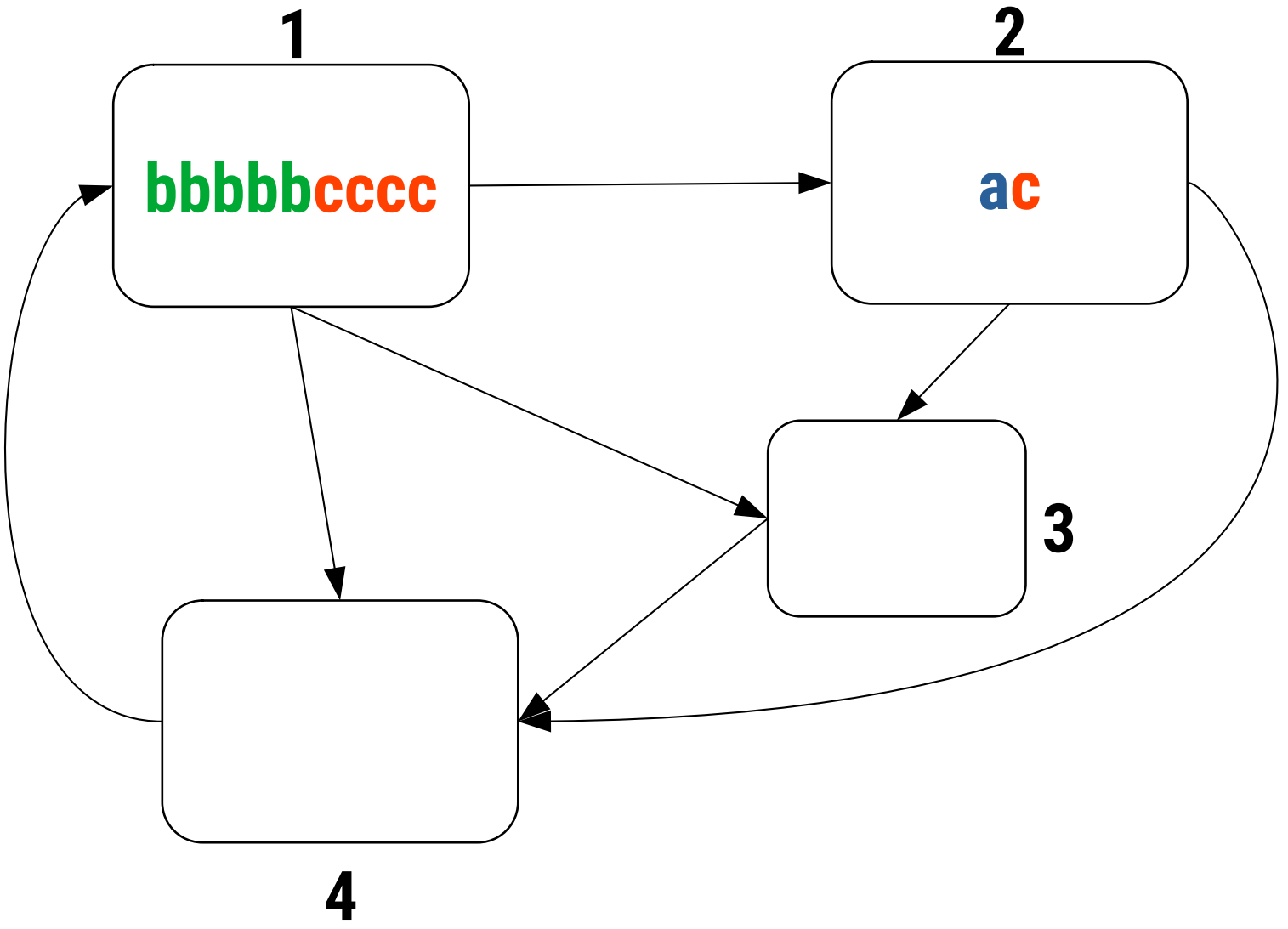
# (Static) Tissue P Systems




**Rule in 1: aa→(bb, *here*)(ac, out)**

**[4]** *Martín-Videa*, C., *Păun*, G., *Pazos*, J., *Rodríguez-Patón*, A.. 2003. *Tissue P Systems*. In *Theoretical Computer Science*. Elsevier.

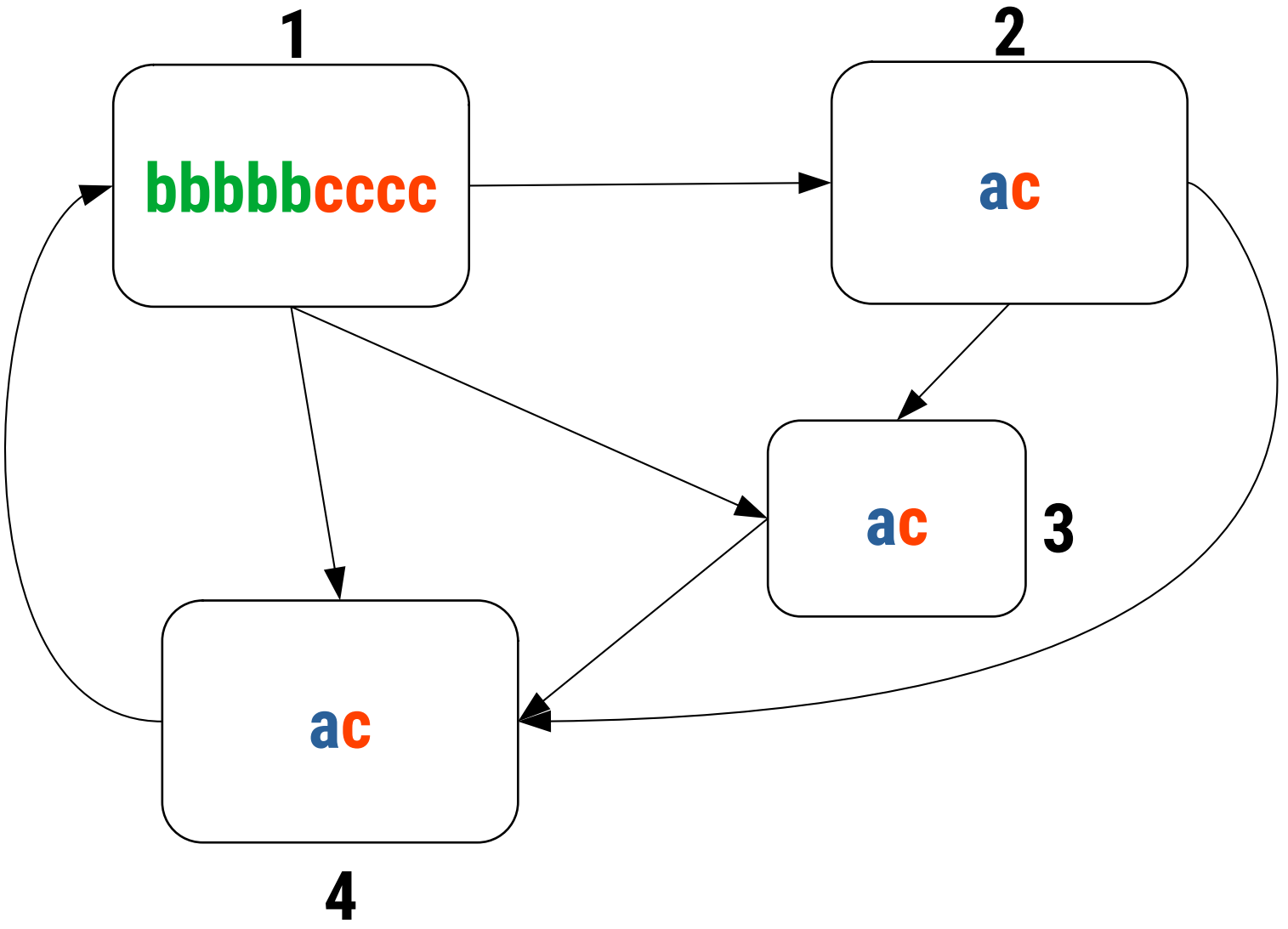
# (Static) Tissue P Systems




**Rule in 1: aa→(bb, *here*)(ac, out)**

**[4]** *Martín-Videa*, C., *Păun*, G., *Pazos*, J., *Rodríguez-Patón*, A.. 2003. *Tissue P Systems*. In *Theoretical Computer Science*. Elsevier.

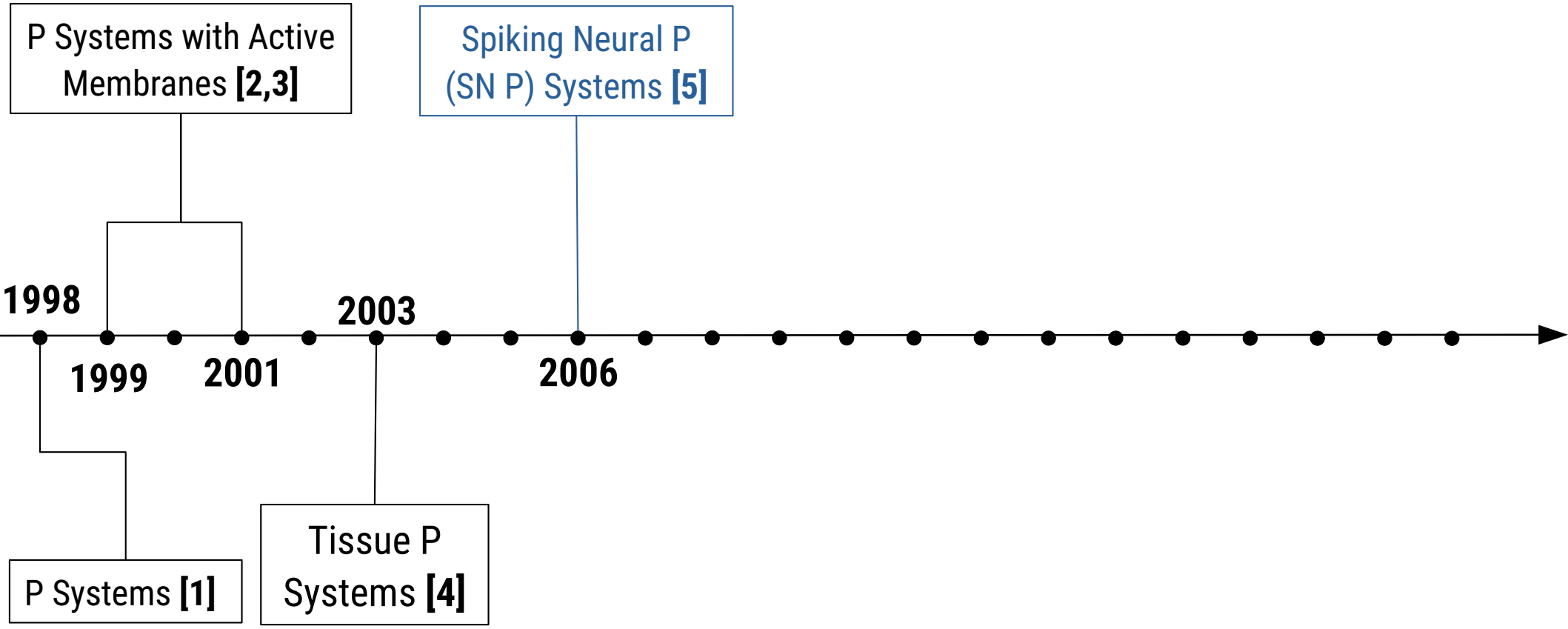
# (Static) Tissue P Systems




**Rule in 1: aa→(bb, *here*)(ac, out)**

**[4]** *Martín-Videa*, C., *Păun*, G., *Pazos*, J., *Rodríguez-Patón*, A.. 2003. *Tissue P Systems*. In *Theoretical Computer Science*. Elsevier.

# Membrane Computing - P Systems




**[1]** *Păun*, G.. 1998. *Computing with Membranes*. In *Technical Report. Turku Centre for Computer Science*.

**[2]** *Păun*, G.. 1999. *P Systems with Active Membranes: Attacking NP-Complete Problems*. In *Centre for Discrete Mathematics and Theoretical Computer Science* (CDMTCS-102) – Research Report Series

**[3]** *Păun*, G.. 2001. *P Systems with Active Membranes: Attacking NP-Complete Problems*. In *Journal of Automata, Languags, and Combinatorics.* vol.6, issue 1 (January 2001), 75-90.

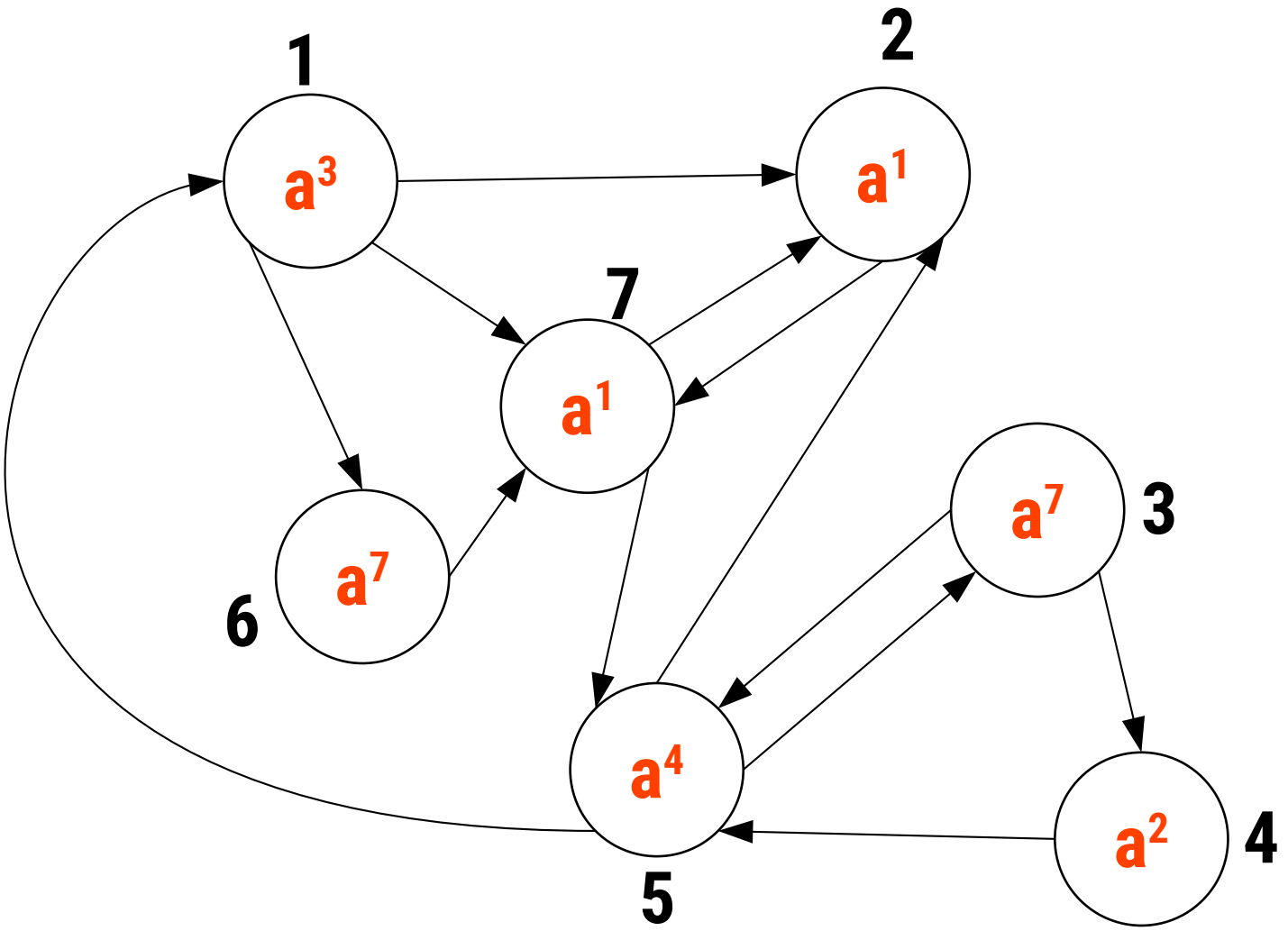
**[4]** *Martín-Videa*, C., *Păun*, G., *Pazos*, J., *Rodríguez-Patón*, A.. 2003. *Tissue P Systems*. In *Theoretical Computer Science.* Elsevier.

**[5]** *Ionescu*, M., Păun, G., *Yokomori*, T.. 2006. *Spiking Neural P Systems*. In *Fundamenta Informaticae*. vol 71, issue 2,3 (February 2006), 279-308.

# Spiking Neural P Systems

**O = {a}**

**a - spike**

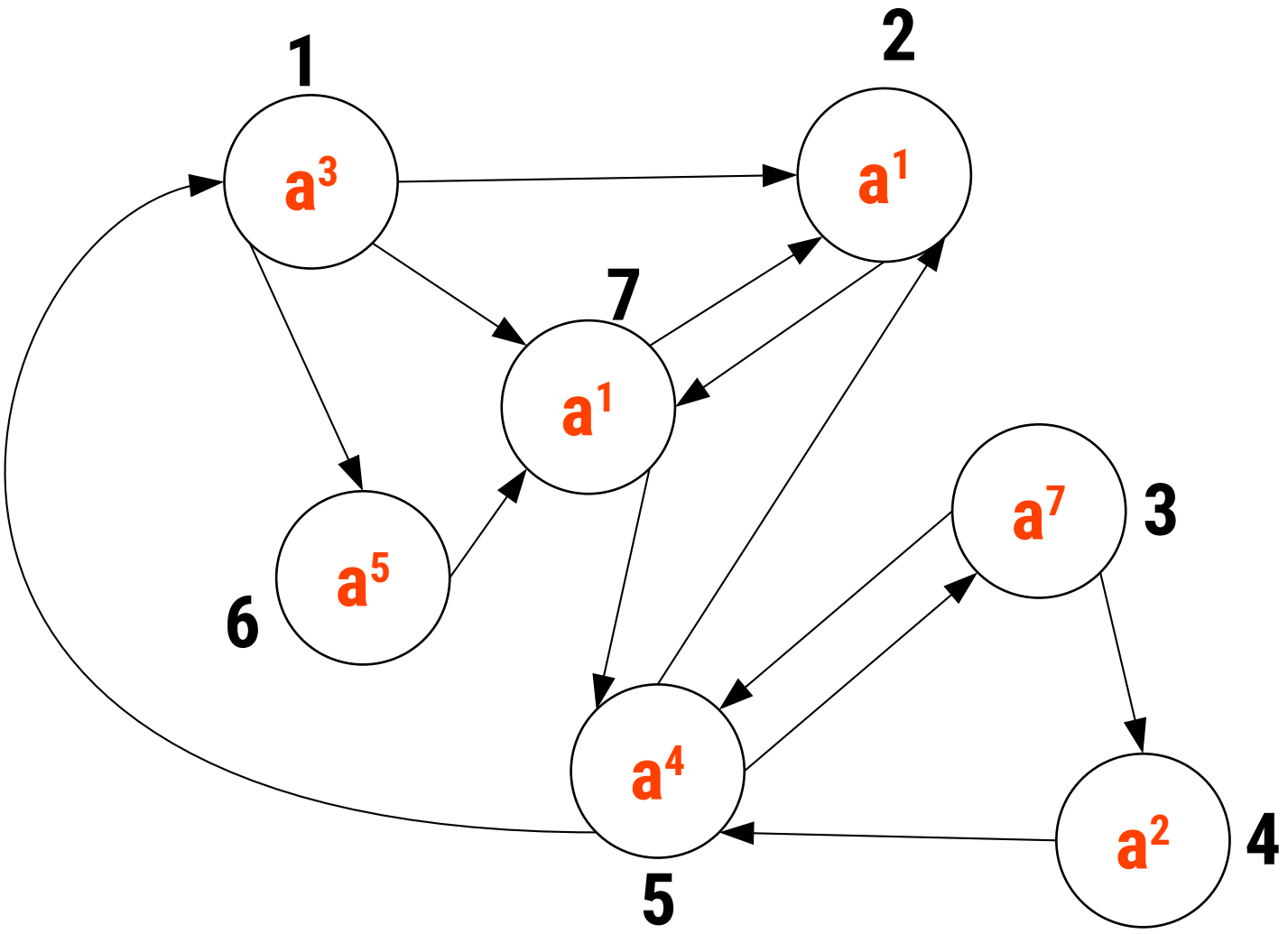



**[5]** *Ionescu*, M., Păun, G., *Yokomori*, T.. 2006. *Spiking Neural P Systems*. In *Fundamenta Informaticae*. vol 71, issue 2,3 (February 2006), 279-308.

# Spiking Neural P Systems

O = {a}

a - spike




$a^{2x+1}/a^2 \rightarrow a$: 3

**2x+1, x** -whole number

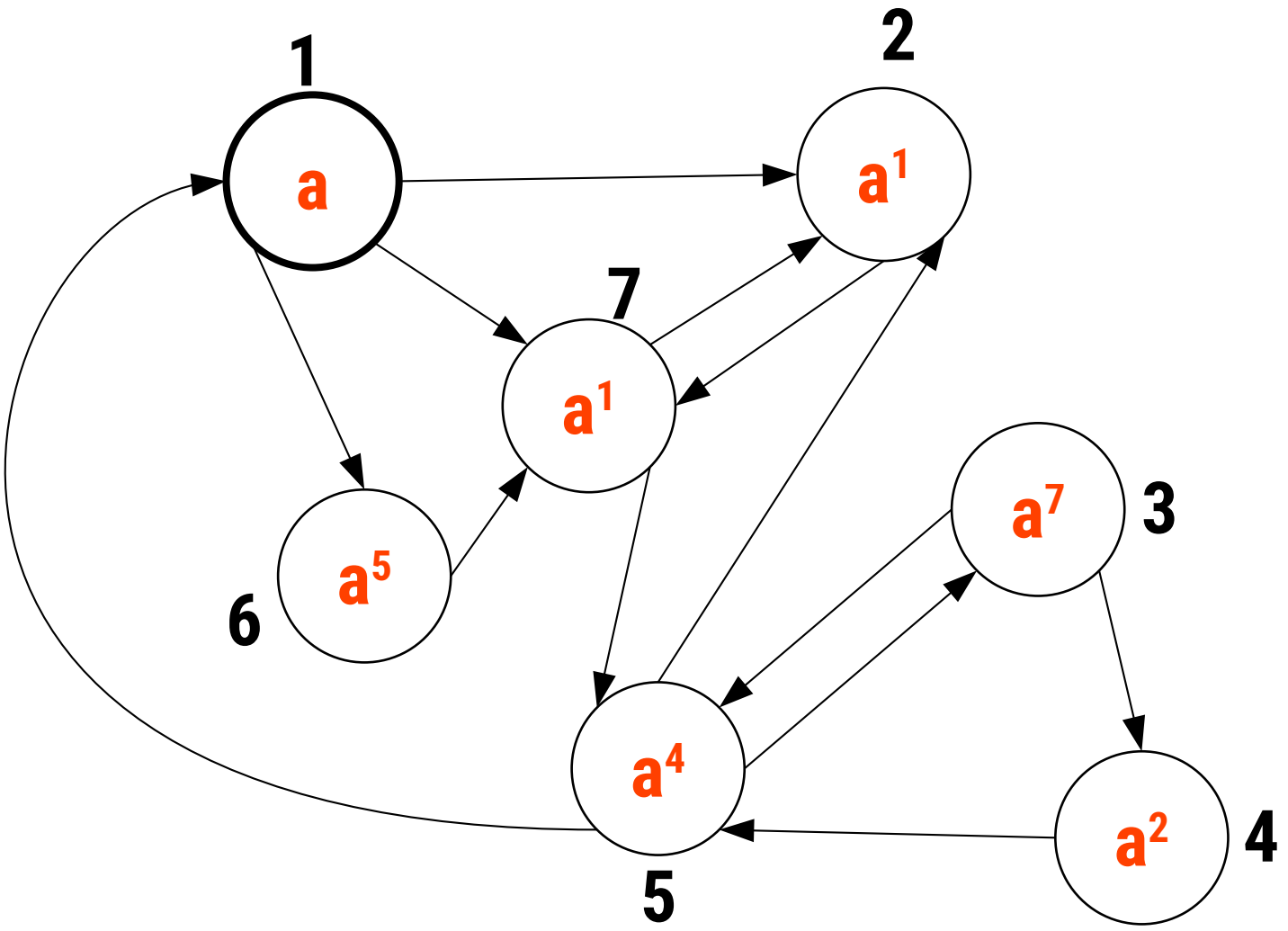
**[5]** *Ionescu*, M., Păun, G., *Yokomori*, T.. 2006. *Spiking Neural P Systems*. In *Fundamenta Informaticae*. vol 71, issue 2,3 (February 2006), 279-308.

# Spiking Neural P Systems

O = {a}

a - spike

t=0




$a^{2x+1}/a^2 \rightarrow a$: 3

2x+1, x -whole number

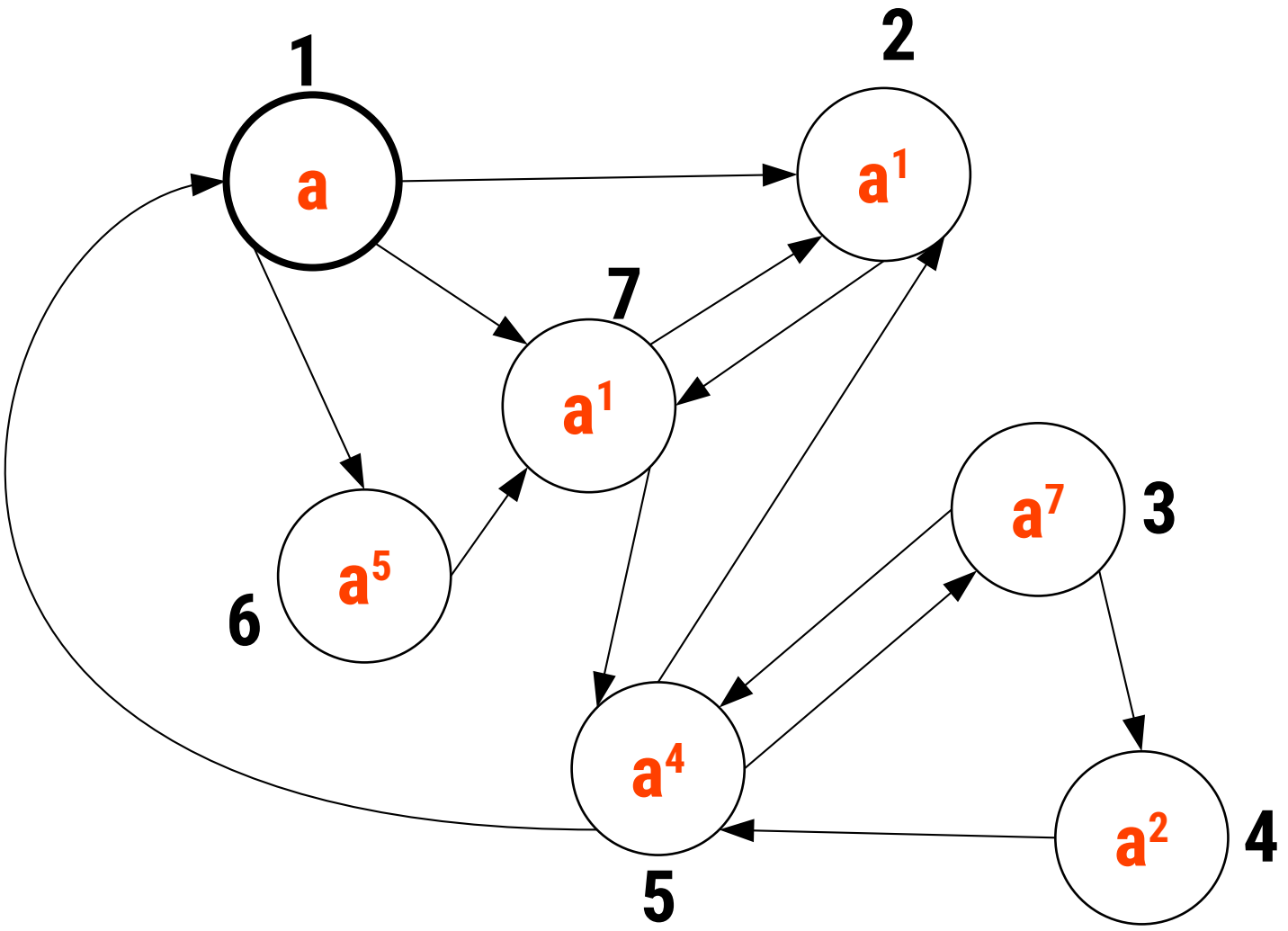
**[5]** *Ionescu*, M., Păun, G., *Yokomori*, T.. 2006. *Spiking Neural P Systems*. In *Fundamenta Informaticae*. vol 71, issue 2,3 (February 2006), 279-308.

# Spiking Neural P Systems

O = {a}

a - spike

t=1




$a^{2x+1}/a^2 \rightarrow a$: 3

**2x+1, x** -whole number

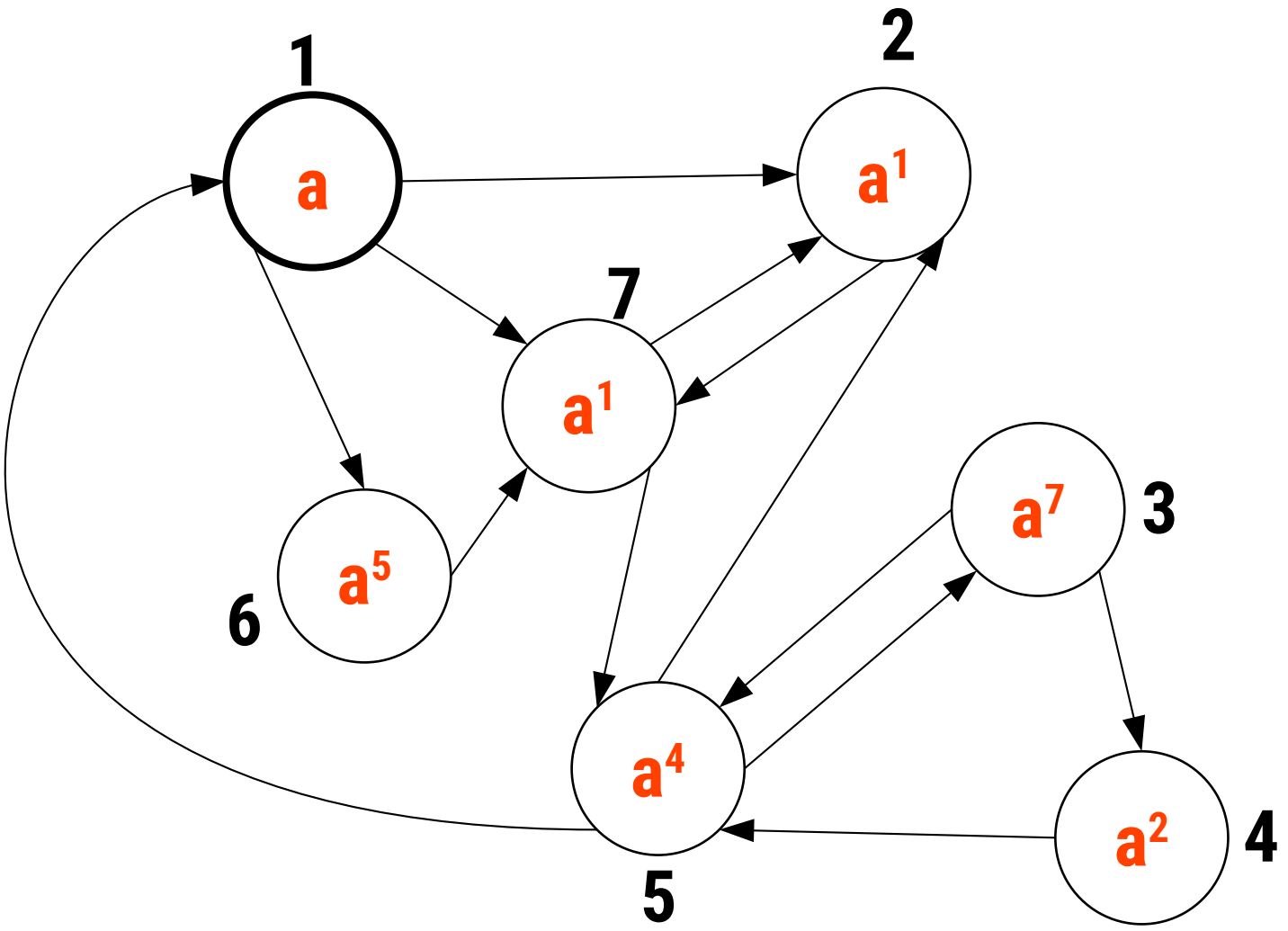
**[5]** *Ionescu*, M., Păun, G., *Yokomori*, T.. 2006. *Spiking Neural P Systems*. In *Fundamenta Informaticae*. vol 71, issue 2,3 (February 2006), 279-308.

# Spiking Neural P Systems

O = {a}

a - spike

t=2




$a^{2x+1}/a^2 \rightarrow a$: 3

**2x+1, x** -whole number

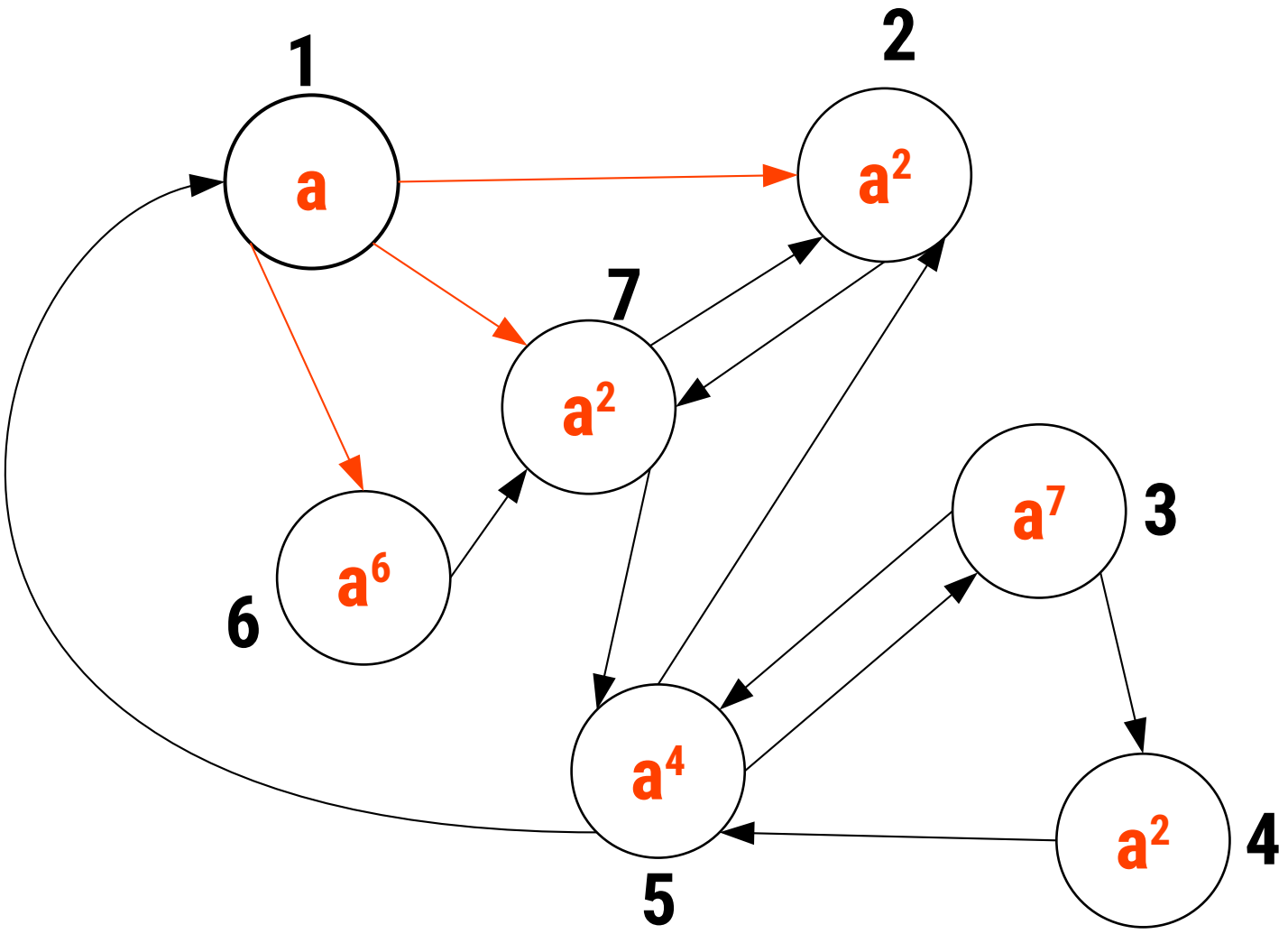
**[5]** *Ionescu*, M., Păun, G., *Yokomori*, T.. 2006. *Spiking Neural P Systems*. In *Fundamenta Informaticae*. vol 71, issue 2,3 (February 2006), 279-308.

# Spiking Neural P Systems

O = {a}

a - spike

t=3




$a^{2x+1}/a^2 \rightarrow a$: 3

2x+1, x -whole number

**[5]** *Ionescu*, M., Păun, G., *Yokomori*, T.. 2006. *Spiking Neural P Systems*. In *Fundamenta Informaticae*. vol 71, issue 2,3 (February 2006), 279-308.

# P Systems – Uses + Research

**Uses:**

- Use P systems to model natural phenomena (i.e. population dynamics)
- Use P systems to solve NP-complete problems
- Use P systems to design hardwared

+ Others applications (fault detection, image skeletonization, etc.)

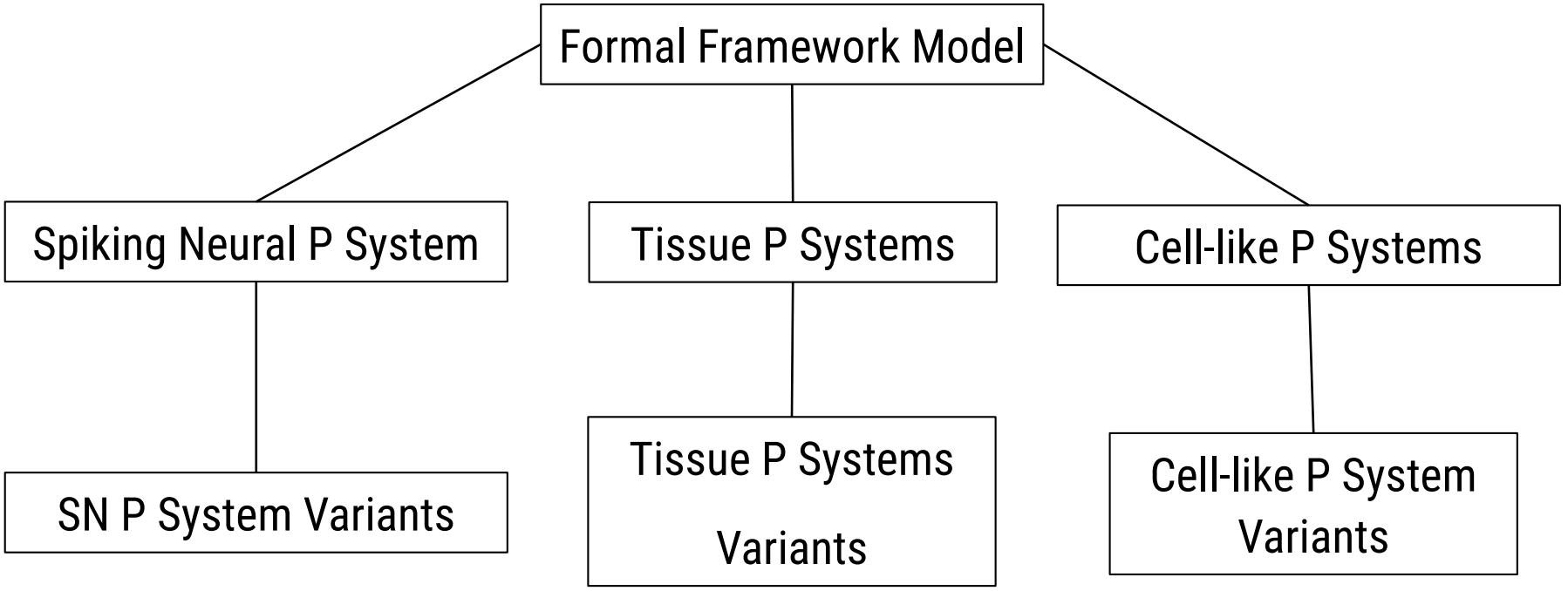
**Research Directions:**

- Checking computability of  P Systems (Turing-completeness)
- Comparing *features* of different P Systems (Feature Bisimulation)
- Creating *stronger normal forms* for different P Systems
- Looking for smallest universal system for a particular P System type

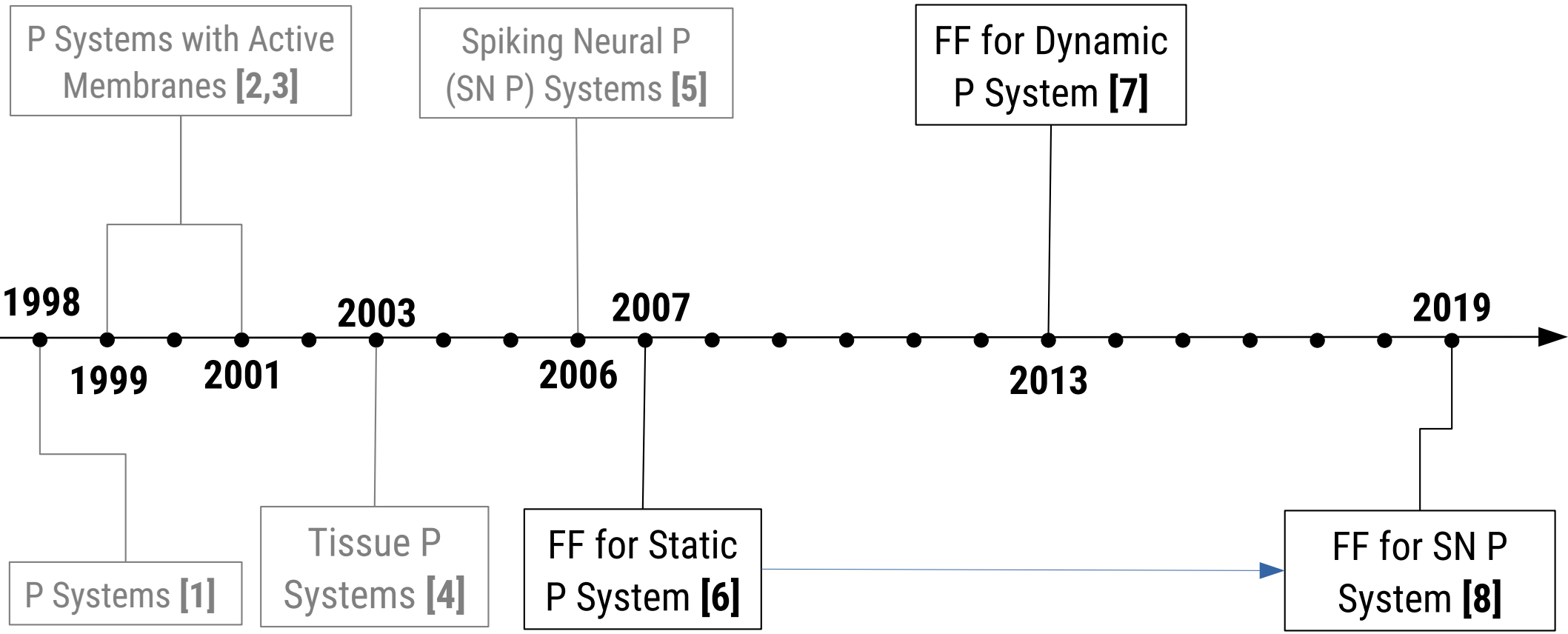
# Formal Framework – The Idea

- A framework (model) that generalizes P Systems and capture their essential properties

- Can be used as a tool to study P systems (e.g. can help answer open problems in Membrane computing, can be used to ask interesting questions)



# Formal Framework (FF) for P Systems





**[6]** *Freund* R., *Verlan* S. 2007. *A Formal Framework for Static (Tissue) P Systems*. In: Eleftherakis G., Kefalas P., Păun G., Rozenberg G., Salomaa A. (eds) *Membrane Computing*. WMC 2007. Lecture Notes in Computer Science, vol 4860. Springer, Berlin, Heidelberg

**[7]** *Freund*, R., *Pérez-Hurtado*, I., *Riscos-Núñez*, A., *Verlan*, S.. 2013. *A Formalization of Membrane Systems with Dynamically Evolving Structures.* In *International Journal of Computer Mathematics*, 90:4, 801-815

**[8]** *Verlan*, S., *Freund*, R., *Alhazov*, A., *Pan*, L.. 2008. A Formal Framework for Spiking Neural P Systems. In *Proceedings of 20th Conference on Membrane Computing (CMC20)*

# Formal Framework – Components

**1. Configuration** – the state of the system – multisets + membrane structure

**2. Generalized Rule** – general rule to transform the state of the system – multiset rewriting + membrane structure change

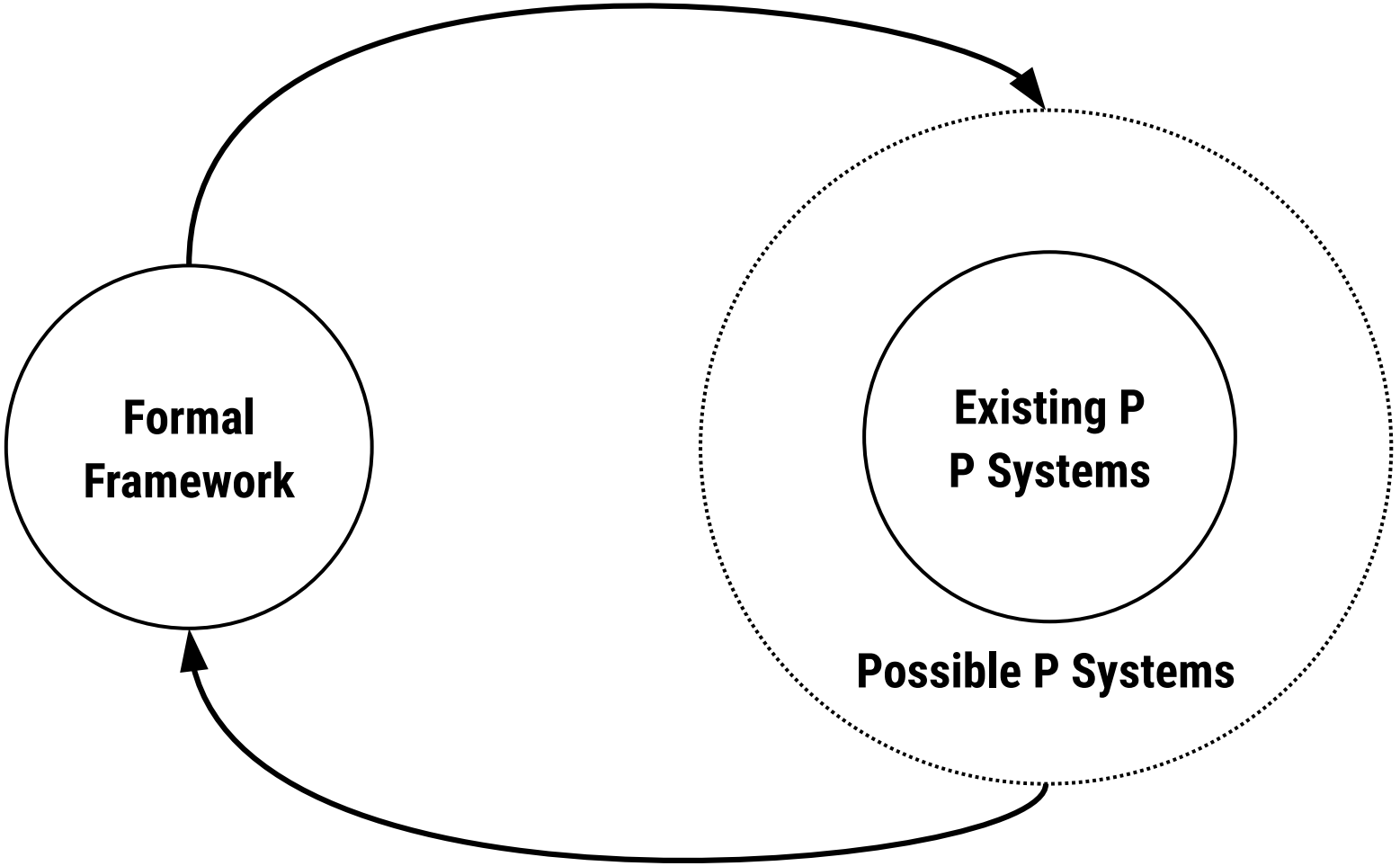
3**. Rules Application** – how to apply a set of rules

**4. Derivation Modes** – what combination of rules are allowed (different rule combination semantics)

# Formal Framework - Research Approaches

**FF→P:** Help answer open problems

**FF→P:** Help ask interesting questions




**P->FF:** Improve the framework

# Formal Framework – Research Ideas

**1.** Merge the dynamic formal framework with the SNP formal framework . **(FF)**

**2.** *Conjecture:* Many SNP system variants are ‘equivalent’. Use formal framework to check if this is true. **(FF→ P).**

**3.** Extentend the formal framework to handle self-modifying P systems. **(P→FF).**

**4.** Check if “all” P systems if they can be represented using formal framework. Extend the framework if needed. **(P→ FF)**

**5.** Reformulate rule representation as bottom-up instead of top down. **(FF)**