

# Characterization of Symport/Antiport P Systems Complexity Classes

M1 internship presentation  
September 7, 2023

Vivien Ducros<sup>1</sup>    Claudio Zandron<sup>2</sup>

<sup>1</sup>DER Informatique  
ENS Paris-Saclay, Gif-sur-Yvette

<sup>2</sup>Dipartimento di Informatica, Sistemistica e Comunicazione  
Università degli Studi di Milano-Bicocca

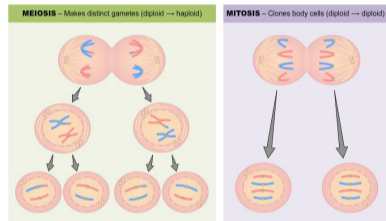
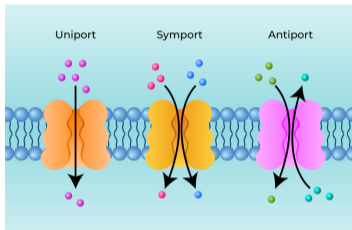
March - July 2023

# Plan

- 1 Introduction
- 2 Definitions
- 3 State of the art
- 4 Characterizing  $\text{PMC}_{\text{CSC}}$
- 5 Conclusion

# Introduction to Membrane Computing

- Bio-inspired computational model
- Introduced by Georghe Păun



- Power of this model: can create and use an exponential number of objects in a polynomial time
- My internship: study of the relations between the complexity classes of symport/antiport P systems

# Table of Contents

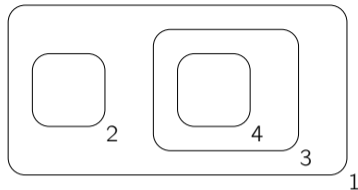
- 1 Introduction
- 2 Definitions
- 3 State of the art
- 4 Characterizing **PMC**<sub>CSC</sub>
- 5 Conclusion

# Plan

- 1 Introduction
- 2 Definitions
- 3 State of the art
- 4 Characterizing  $\text{PMC}_{\text{CSC}}$
- 5 Conclusion

## How does a symport/antiport P system work ?

- Structure: labelled membranes organized as a tree:



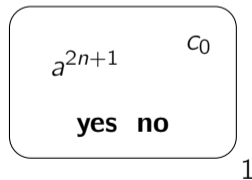
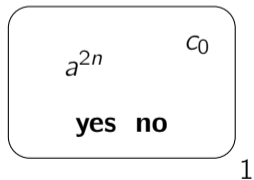
- Alphabet of objects  $\Gamma$
- Multisets of objects in each region, represented as words:  $\{\{a, a, b\}\} = a^2b$
- Associated rules ( $u, v$  multisets on  $\Gamma$ ,  $a, b, c \in \Gamma$ )
  - ▶ Symport rules:  $(u, out)$  or  $(u, in)$
  - ▶ Antiport rule:  $(u, out; v, in)$
  - ▶ Division rule:  $[a]_i \rightarrow [b]_i[c]_i$
  - ▶ Separation rule:  $[a]_i \rightarrow [\Gamma_1]_i[\Gamma_2]_i$
- Environment  $\mathcal{E} \in \Gamma$
- Input membrane (and input alphabet  $\Sigma \subseteq \Gamma$ ), output membrane

# An example of a symport/antiport P system solving **PARITY**

- Objects:  $\Gamma = \{a, c_0, c_1, d_1, d_2, \mathbf{yes}, \mathbf{no}\}$ ,  $\mathcal{E} = \{c_1, d_1, d_2\}$ ,  $\Sigma = \{a\}$
- One membrane, initial multiset:  $m_1 = \{c_0, \mathbf{yes}, \mathbf{no}\}$ , rules:  
 $\mathcal{R}_1 = \{(a^2, out), (c_0, out; c_1 d_1, in), (d_1, out; d_2, in), (c_1 a \mathbf{no}, out), (c_1 d_2 \mathbf{yes}, out)\}$
- Input membrane: membrane 1, output membrane: environment

Two cases, whether the input is even or odd:

$k = 0$



# Definitions (1)

## Recognizer symport/antiport P system

A recognizer symport/antiport P system is a tuple:

$$\Pi = (\Gamma, \mathcal{E}, \Sigma, \mu, m_1, \dots, m_q, \mathcal{R}_1, \dots, \mathcal{R}_q, i_{in}, i_{out})$$

where:

- **yes**, **no**  $\in \Gamma \setminus \Sigma$  are the answer objects,  $\exists i, j \in \llbracket 1, q \rrbracket$ , **yes**  $\in m_i$ , **no**  $\in m_j$
- $i_{out} = 0$

and such that:

- Every computation ends
- For all computations of  $\Pi$  on the same input, either **yes** (the computation is said to be accepting) or **no** (the computation is said to be rejecting) have been released in the environment **only at the last step**.

## Definitions (2)

Recognizer P system classes

- **CSC**: symport/antiport P systems with separation rule
- **CD<sub>ne</sub>C**: symport/antiport P systems with division rule
- **CD<sub>e</sub>C**: symport/antiport P systems with division rule only for elementary membranes

For  $k \in \mathbb{N}$ , **CSC**( $k$ ), **CD<sub>ne</sub>C**( $k$ ) and **CD<sub>e</sub>C**( $k$ ) when then length of the rules is less than  $k$ .

*D*-consistent family, polynomially uniform family

$\Pi = (\Pi(i))_{i \in I}$ , family of P systems

$\Pi$  is *D*-consistent if  $\forall i \in I, \Pi(i) \in \mathcal{D}$ .

It is said to be *polynomially uniform* if there exists a deterministic Turing Machine that computes the P system  $\Pi(i)$  for all  $i \in I$  in polynomial time of  $|i|$ .

## Definitions (3)

### Decision problem

$X = (I_X, \theta_X)$  is a decision problem, with  $I_X$  the set of instances and  $\theta_X : I_X \rightarrow \{0, 1\}$ .

### Recognition in a uniform way

A polynomially uniform family  $\Pi = (\Pi(n))_{n \in \mathbb{N}}$  recognizes in a uniform way the decision problem  $X = (I_X, \theta_X)$  if there exists two polynomially computable functions  $cod$  and  $s$  such that:

- $s : I_X \rightarrow \mathbb{N}$  to indicate which of the P system run
- $cod : I_X \rightarrow \bigcup_{n \in \mathbb{N}} \Sigma(n)$  to encode the input

For  $x \in I_X$ , a computation of the P system  $\Pi(s(x))$  with the input  $cod(x)$  must accept iff  $\theta(x) = 1$ .

## Definitions (4)

### $\mathbf{MC}_{\mathcal{D}}(g)$ complexity classes

$\mathcal{D}$  be a class of recognizer P systems,  $g$  be a constructible function,  $X$  a decision problem  
 $X \in \mathbf{MC}_{\mathcal{D}}(g)$  if there exists a family  $\Pi$  recognizing  $X$  in a uniform way and such that  $\Pi(s(x))$  computes with the input  $cod(x)$  in less than  $g(|x|)$  steps,  $x \in I_X$ .

### $\mathbf{PMC}_{\mathcal{D}}$ complexity classes

Let  $\mathcal{D}$  be a class of recognizer P systems. The polynomial time complexity classes of P systems are defined as follows:

$$\mathbf{PMC}_{\mathcal{D}} = \bigcup_{k \in \mathbb{N}} \mathbf{MC}_{\mathcal{D}}(O(x^k))$$

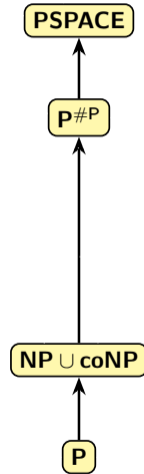
# Plan

- 1 Introduction
- 2 Definitions
- 3 State of the art
- 4 Characterizing  $\text{PMC}_{\text{CSC}}$
- 5 Conclusion

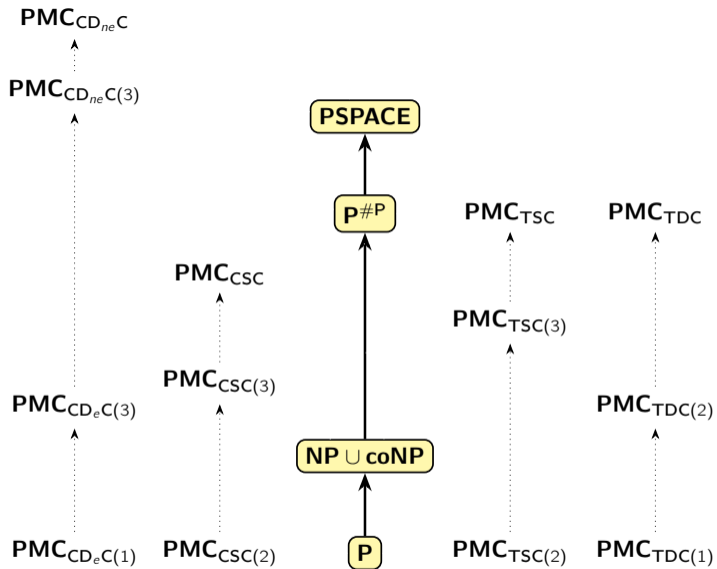
## An other variant: tissue P system

- Membranes organized in a graph structure
- Symport/antiport rules written:  $[u]_i \leftrightarrow [v]_j$
- Other classes: **TDC**( $k$ ), **TSC**( $k$ ), **TDC** and **TSC**
- **CSC**  $\subseteq$  **TSC** (submodel)
- **CD<sub>ne</sub>C**  $\not\subseteq$  **TDC**

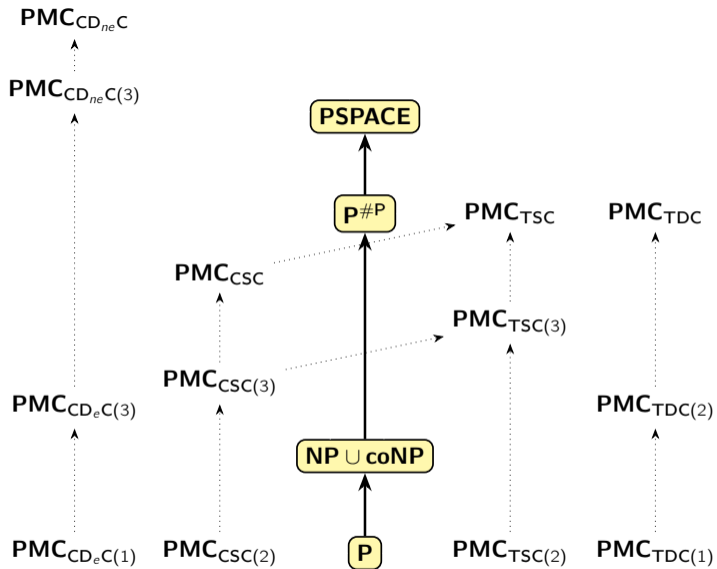
# State of the art



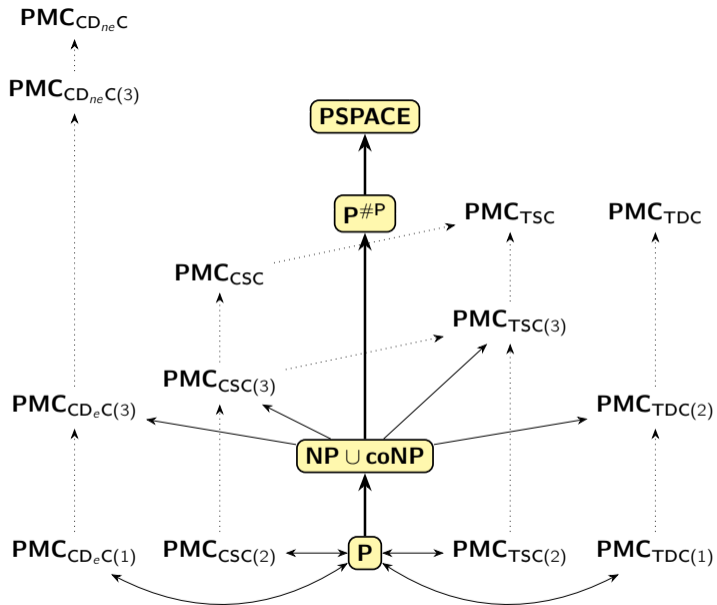
# State of the art



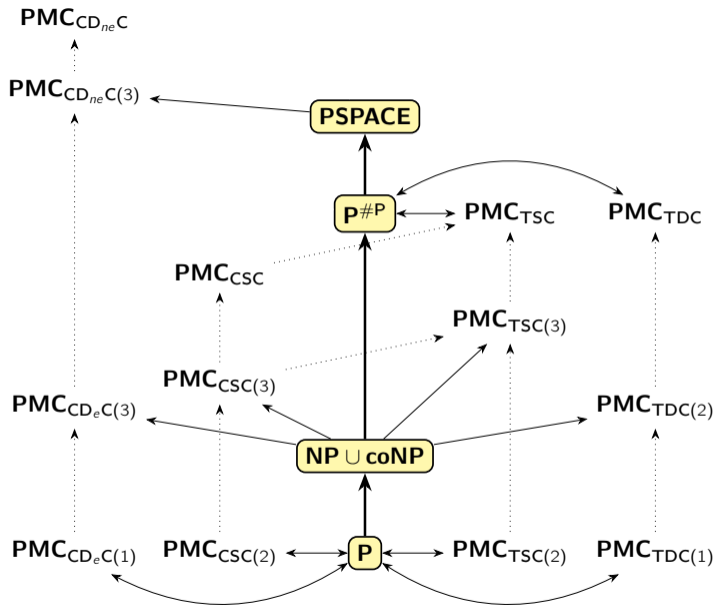
# State of the art



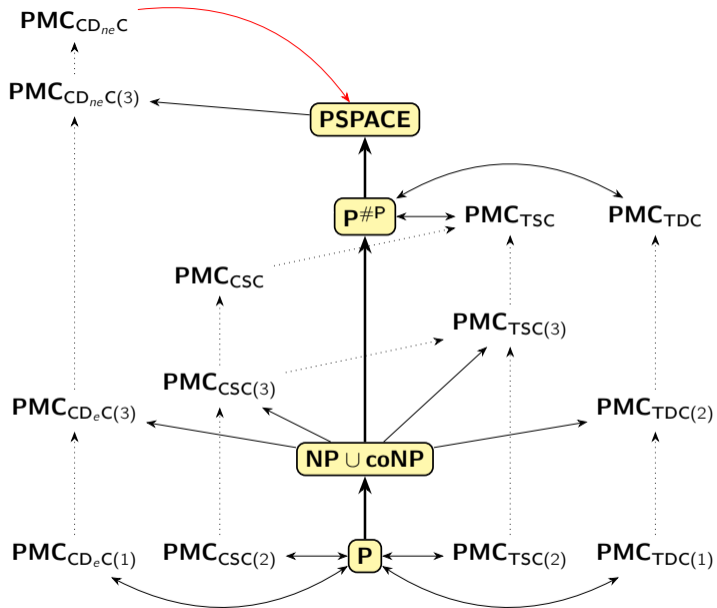
# State of the art



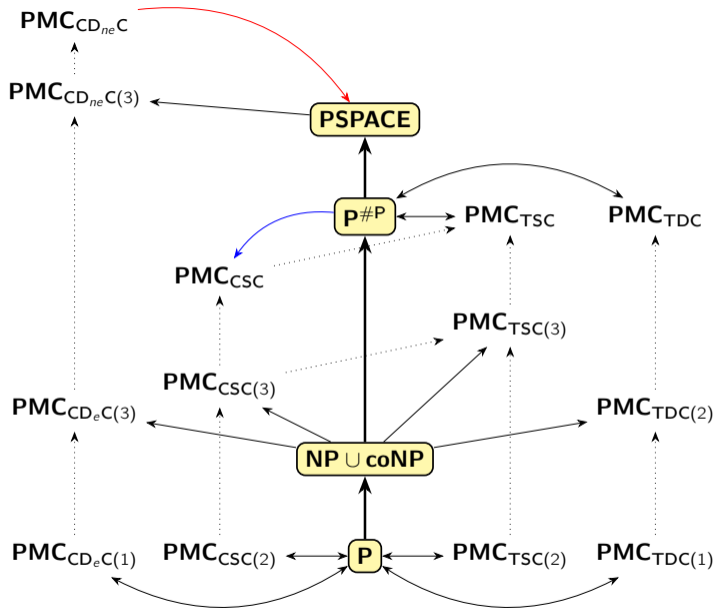
# State of the art



# State of the art



# State of the art



# Plan

- 1 Introduction
- 2 Definitions
- 3 State of the art
- 4 Characterizing **PMC<sub>CSC</sub>**
- 5 Conclusion

# A $\mathbf{P}^{\#\mathbf{P}}$ -complete problem

## LEXICAL\_MIDDLE\_3SAT

**Input:** A boolean formula  $\phi(x_1, \dots, x_n)$  in 3CNF (conjunction of clauses of three literals)

**Question:** Is  $x_n = 1$  in the lexicographically middle satisfying assignment  $x_1 \cdots x_n \in \{0, 1\}^n$  of  $\phi$ ?

### Example

With  $\phi(x_1, x_2, x_3) = (x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$ ,

the satisfying assignments are  $(0, 0, 0); (0, 1, 1); (1, 0, 1); (1, 1, 0); (1, 1, 1)$  (ordered in the lexicographic order)

Then the median is  $(1, 0, \mathbf{1})$  and the answer to the problem is **yes**.

**Remark:** When an even number of assignments satisfy the formula, the lowest of the two median assignments is chosen as the median.

# Solving **LEXICAL\_MIDDLE\_3SAT** with **CSC** P system

Solution of **SAT** by **CSC(3)** P system:

- **Generation phase:** Division of a membrane in  $2^n$  copies representing the assignments
- **Checking phase:** Each membrane computes the truth value of the formula
- **Output phase:** The system returns **yes** or **no**

# Solving **LEXICAL\_MIDDLE\_3SAT** with **CSC P** system

Solution of **SAT** by **CSC(3) P** system:

- **Generation phase:** Division of a membrane in  $2^n$  copies representing the assignments
- **Checking phase:** Each membrane computes the truth value of the formula
- **Output phase:** The system returns **yes** or **no**

Modifications:

- Deletion of the output phase
- **Activation phase:** Makes the membranes satisfying the formula able to interact
- **Search phase:** Binary search repeating  $n$  times the sub-phases:
  - ▶ Preparation sub-phase
  - ▶ Comparison sub-phase
  - ▶ Determination sub-phase
- **Output phase:** Returns the result

## Search phase: finding the median

Similar problem: finding the median  $m = b_1 \cdots b_n$  in a set  $E$  of values between 0 and  $2^n - 1$ .

- First idea: ordering the values and choose the middle one  
→ not possible for the membranes

## Search phase: finding the median

Similar problem: finding the median  $m = b_1 \cdots b_n$  in a set  $E$  of values between 0 and  $2^n - 1$ .

- First idea: ordering the values and choose the middle one  
→ not possible for the membranes
- Binary search: determining the bits by comparing the values to a well chosen number  
For  $i \in \llbracket 1, n \rrbracket$ , if we know  $b_1, \dots, b_{i-1}$ , then

$$b_i = \begin{cases} 1 & \text{if } |\{x \in E : x \geq b_1 \cdots b_{i-1} 1 0 \cdots 0\}| > |\{x \in E : x < b_1 \cdots b_{i-1} 1 0 \cdots 0\}| \\ 0 & \text{else} \end{cases}$$

## Search phase: finding the median

Similar problem: finding the median  $m = b_1 \cdots b_n$  in a set  $E$  of values between 0 and  $2^n - 1$ .

- First idea: ordering the values and choose the middle one  
→ not possible for the membranes
- Binary search: determining the bits by comparing the values to a well chosen number  
For  $i \in \llbracket 1, n \rrbracket$ , if we know  $b_1, \dots, b_{i-1}$ , then

$$b_i = \begin{cases} 1 & \text{if } |\{x \in E : x \geq b_1 \cdots b_{i-1} 1 0 \cdots 0\}| > |\{x \in E : x < b_1 \cdots b_{i-1} 1 0 \cdots 0\}| \\ 0 & \text{else} \end{cases}$$

Let's resume:

- ▶ Preparation sub-phase: introducing the reference value  $b_1 \cdots b_{i-1} 1 0 \cdots 0$  in the system
- ▶ Comparison sub-phase: comparing **all** the satisfying assignments to the reference
- ▶ Determination sub-phase: determining  $b_i$ , counting the number of greater and lower values

# Solving **LEXICAL\_MIDDLE\_3SAT** with **CSC P** system

Number of steps:

- **Generation phase:**  $3n + 1$  steps
- **Checking phase:**  $2m$  steps
- **Activation phase:** 1 step
- **Search phase:**  $n(2n + 5)$  steps
  - ▶ Preparation sub-phase:  $n + 1$  steps
  - ▶ Comparison sub-phase:  $n + 1$  steps
  - ▶ Determination sub-phase: 3 steps
- **Output phase:** 1 step

Total:  $2n^2 + 8n + 2m + 3$  steps, polynomial time

The number of objects, the number of rules and their length are polynomial too.

# Characterization

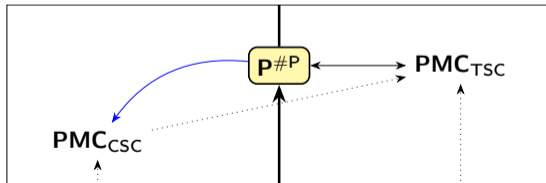
## Theorem

$\text{LEXICAL\_MIDDLE\_3SAT} \in \text{PMC}_{\text{CSC}}$

Obtained with the P system previously described.

## Corollary

$\text{PMC}_{\text{CSC}} = \text{P}\#\text{P}$



# Plan

- 1 Introduction
- 2 Definitions
- 3 State of the art
- 4 Characterizing  $\text{PMC}_{\text{CSC}}$
- 5 Conclusion

# Conclusion

- Characterization of  $\mathbf{PMC}_{\mathbf{CD}_{ne}\mathbf{C}}$ , conjecture of Petr Sosík
- Characterization of  $\mathbf{PMC}_{\mathbf{CSC}}$
- More generally: membrane computing used to tackle  $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$  and  $\mathbf{P}^{\#\mathbf{P}} \stackrel{?}{=} \mathbf{PSPACE}$  unsolved problems
- Spiking neural P systems

Thank you for listening  
Please ask your questions !