Computational Complexity in Membrane Systems

Mario J. Pérez–Jiménez

Research Group on Natural Computing
Dpt. of Computer Science and Artificial Intelligence
University of Seville, Spain
marper@us.es

UC-2011 – Turku, Finland, June 9, 2011

Outline

- Computability.
- Computability versus Complexity.
- Membrane Computing.
- Cell-like framework.
  - Complexity classes
  - Limitations and efficiency
- Tissue-like framework.
  - Complexity classes
  - Limitations and efficiency
- Open problems

Computability (I)

> As long as a branch of science offers an abundance of problems, so long it is alive ...
>
> (D. Hilbert, 1900)

Which tasks (abstract problems) can be performed (solved) efficiently?

Computability theory:

- A rigorous definition of the concept of
  - a task.
  - procedures for solving tasks.

It focuses on computational tasks and automated/mechanical procedures (computing devices, algorithms).

Computability (II)

Informal notion of algorithm:

- Producing an output from a set of inputs in a finite number of steps.

Model of computation: formal notions (1931–1936)

- Recursive functions.
- λ–calculus.
- Turing machines.

All these models are equivalent.

This realization led to the invention of the standard universal electronic computer.

Computability (III)

Computation is not merely a practical tool.

It is also a major scientific concept.

Scientists now view many natural phenomena as akin to computational processes.

Today, computational models underlie many research areas in biology and neuroscience.

Computability versus Complexity

Computability:

- What problems are computable in a (universal) model?
- Interesting tasks are inherently uncomputable.
  - Negative results: There exist infinitely many possible algorithms.
  - Computation/Algorithm is a mathematically precise notion.

Complexity (1970)

- What (computable) problems are efficiently solvable?
- Lower bounds on resources required to solve problems on a model.
  - Negative results: There exist infinitely many possible algorithms.
  - We have to prove mathematically that each algorithm solving the problem is less efficient.

Computational Complexity

Study of the intrinsic complexity of computational tasks (abstract problems).

Computational complexity theory is an infant science (about 40 years old).

- Many important results are less than 20 years old.
- Has also been used to prove some metamathematical theorems.

Complexity theory:

- Has failed (until now) to determine the intrinsic complexity of problems such as SAT or 3-COL
- Has succeeded in establishing that they are computationally equivalent

Complexity classes (I)

Complexity theory deals with *decision problems* which are problems that require a "*yes*" or "*no*" answer ($X = (I_X, \theta_X)$).

- Combinatorial optimization problems can be transformed into decision problems by supplying a target/threshold value for the quantity to be optimized, and then asking whether this value can be attained.

A complexity class is a set of (decision) problems that can be solved (in a universal computing model) within given resource bounds.

- The specific computing model does not matter!!!

Church–Turing thesis: *every physically realizable computation device can be simulated by a TM*

Strong Church–Turing thesis: *every physically realizable computation device can be simulated by a TM* with polynomial overhead

Complexity classes (II)

Solve a decision problem $\equiv$ recognize a language.

- A DTM, M, *recognizes* a language L whenever, for any input string u, if $u \in L$, then the answer of $M(u)$ is *yes*, and the answer is *no* otherwise.
- A NDTM, M, *recognizes* L if for any string u over Γ, $u \in L$ iff there exists a computation of M with input u such that the answer is *yes*.

Determinism versus nondeterminism:

- The key: how to accept (reject) an input string.

Each abstract problem has a fixed *reasonable encoding scheme* associated with it.

Complexity classes (III)

P: decision problems with feasible procedures.

- **P** is the class of all decision problems solvable by DTMs in polynomial time.

NP: decision problems whose solutions can be efficiently verified.

- **NP** is the class of all decision problems solvable by NDTMs in polynomial time

The P versus NP problem

- Many problems can be solved by exhaustive search.
- Can it be replaced by a more efficient search algorithm?
- Whether or not finding solutions is harder than checking the correctness of solutions.
- Whether or not discovering proofs is harder than verifying their correctness.

- This is essentially the famous **P** versus **NP** problem

. . . the central problem of Computational Complexity theory.

It is widely believed that it is harder

- finding (resp. proving) than checking (resp. verifying)
- solving a problem than checking the correctness of a solution
- . . . $\mathbf{P} \neq \mathbf{NP}$

Membrane Computing

- P systems provide nondeterministic models of computation.
- A computational complexity theory in Membrane Computing is proposed.
- Polynomial complexity classes associated with (cell–like and tissue–like) P systems are presented.
  - A notion of acceptance must be defined in the new (nondeterministic) framework.
    - We consider a definition of acceptance different than the classical one for nondeterministic devices.

Cell–like Framework

- P systems without input: $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, R, i_{out})$.
- P systems with input: $\Pi = (\Gamma, \Sigma, H, \mu, \mathcal{M}_1, \dots, \mathcal{M}_q, R, i_{in}, i_{out})$.
- Recognizer P systems:
  - The working alphabet contains two distinguished elements *yes* and *no*.
  - All computations halt.
  - For any computation of the system, either object *yes* or object *no* (but not both) must have been sent to the output region of the system, and only at the last step of the computation.
- Accepting/rejecting computations for recognizer P systems

Uniform families (I)

- P systems are computing devices of finite size and they have a finite description with a fixed amount of initial resources.

- In order to solve a decision problem a (possibly infinite) family of P systems is considered.

- The concept of solvability in the framework of P systems also takes into account the pre-computational process of (efficiently) constructing the family that provides the solution.

  - The terminology *uniform family* is used to denote that this construction is performed by a *single* computational machine.

Uniform families (II)

- P systems without input membrane:
  - A family $\mathbf{\Pi} = \{\Pi(w) : \; w \in I_X\}$ *associated with a decision problem* $X = (I_X, \theta_X)$ is *uniform* if there exists a DTM which constructs the system $\Pi(w)$ from the instance $w \in I_X$.
    - In such a family, each P system usually processes only one instance.

Uniform families (III)

- P systems with input membrane:
  - A family $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbb{N}\}$ is uniform if there exists a DTM which constructs the system $\Pi(n)$ from $n \in \mathbb{N}$ (which on input 1^n outputs $\Pi(n)$).
    - In such a family, the P system $\Pi(n)$ will process all the instances with numerical parameters (reasonably) encoded by n.
  - For these families the concept of *polynomial encoding* is introduced:
    - A *polynomial encoding* of X in $\mathbf{\Pi}$ is a pair (cod, s) of polynomial–time computable functions over I_X such that for each $w \in I_X$, $s(w) \in \mathbb{N}$ and $cod(w)$ is an input multiset of $\Pi(s(w))$.
    - Polynomial encodings are stable under polynomial–time reductions.

Families polynomially uniform by TM

- In both cases, the family should be constructed in an efficient way.
- *Polynomially uniform by Turing machines*: a uniform (by a single Turing machine) and effective (in polynomial time) construction of the family.
  - A family $\mathbf{\Pi}$ of recognizer P systems is *polynomially uniform by Turing machines* if there exists a DTM working in polynomial time which constructs $\Pi(w)$ (resp. $\Pi(n)$) from $w \in I_X$ (resp., from $n \in \mathbb{N}$).

Confluent P systems

- Trying to capture the true concept of algorithm by nondeterministic P systems.

- Let $X = (I_X, \theta_X)$ be a decision problem, and $\mathbf{\Pi} = \{\Pi(w) : \ w \in I_X\}$ be a family of recognizer P systems without input membrane.
  - $\mathbf{\Pi}$ is *sound* with respect to X: for each $w \in I_X$, if *there exists* an accepting computation of $\Pi(w)$, then $\theta_X(w) = 1$.
  - $\mathbf{\Pi}$ is *complete* with respect to X: for each $w \in I_X$, if $\theta_X(w) = 1$, then *every* computation of $\Pi(w)$ is an accepting computation.

- Similar definition to families of recognizer P systems with input membrane.

- Sound + Complete = Confluent

Polynomial time solvability by using P systems (I)

- Semi–uniform solutions.
  - A decision problem X is *solvable in polynomial time* by a family of recognizer P systems without input membrane $\mathbf{\Pi} = \{\Pi(w) : w \in I_X\}$, if:
    - ⋆ The family $\mathbf{\Pi}$ is polynomially uniform by Turing machines.
    - ⋆ The family $\mathbf{\Pi}$ is polynomially bounded: there exists $k \in \mathbb{N}$ such that for each $w \in I_X$, every computation of $\Pi(w)$ performs at most $|w|^k$ steps.
    - ⋆ The family $\mathbf{\Pi}$ is sound and complete with respect to X.
  - $X \in \mathbf{PMC}^*_{\mathcal{R}}$.
  - $\mathbf{PMC}^*_{\mathcal{R}}$ is closed under complement and polynomial–time reductions.

Polynomial time solvability by using P systems (II)

- Uniform solutions:
  - A decision problem X is *solvable in polynomial time* by a family of recognizer P systems without input membrane $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbb{N}\}$, if:
    - ⋆ The family $\mathbf{\Pi}$ is polynomially uniform by Turing machines.
    - ⋆ There exists a polynomial encoding (cod, s) of X in $\mathbf{\Pi}$ such that:
      - ⋆ The family $\mathbf{\Pi}$ is polynomially bounded: there exists $k \in \mathbb{N}$ such that for each $w \in I_X$, every computation of $\Pi(s(w))$ with input $cod(w)$ performs at most $|w|^k$ steps.
      - ⋆ The family $\mathbf{\Pi}$ is sound and complete with respect to X.
  - $X \in \mathbf{PMC}_{\mathcal{R}}$.
  - $\mathbf{PMC}_{\mathcal{R}}$ is closed under complement and polynomial–time reductions.
  - We have $\mathbf{PMC}_{\mathcal{R}} \subseteq \mathbf{PMC}^*_{\mathcal{R}}$

Limitations of basic transition P systems (I)

- Let M be a DTM with input alphabet Σ_M. The *decision problem associated with M* is $X_M = (I_M, \theta_M)$, where:
  - $I_M = \Sigma_M^*$.
  - For every $w \in \Sigma_M^*$, $\theta_M(w) = 1$ if and only if M accepts w.
- A Turing machine M is simulated in polynomial time by a family of recognizer P systems from $\mathcal{R}$ if $X_M \in \mathbf{PMC}_{\mathcal{R}}$.
- *Basic transition* P systems: only evolution, communication, and dissolution rules.
- $\mathcal{T}$: class of recognizer basic transition P systems.

Limitations of basic transition P systems (II)

- Every DTM working in polynomial time can be simulated in polynomial time by a family of recognizer basic transition P systems with input membrane `(Seville theorem)`.

- If a decision problem is solvable in polynomial time by a family of recognizer basic transition P systems with input membrane, then there exists a DTM solving it in polynomial time.

- **Theorem:** $\mathbf{P} = \mathbf{PMC}_{\mathcal{T}} = \mathbf{PMC}^{*}_{\mathcal{T}}$.

    - **Corollary:** $\mathbf{P} \neq \mathbf{NP}$ if and only if every, or at least one, **NP**–complete problem is not in $\mathbf{PMC}_{\mathcal{T}} = \mathbf{PMC}^{*}_{\mathcal{T}}$.

P Systems with Active Membranes

- $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, R, i_{out})$,

  (a) $[\, a \to u \,]_h^{\alpha}$, for $h \in H, \alpha \in \{+,-,0\}$, $a \in \Gamma$, $u \in \Gamma^*$ *(object evolution rules)*.

  (b) $a\,[\ \ ]_h^{\alpha_1} \to [\, b \,]_h^{\alpha_2}$, for $h \in H$, $\alpha_1, \alpha_2 \in \{+,-,0\}$, $a, b \in \Gamma$ *(send–in communication rules)*.

  (c) $[\, a \,]_h^{\alpha_1} \to [\ \ ]_h^{\alpha_2}\ b$, for $h \in H$, $\alpha_1, \alpha_2 \in \{+,-,0\}$, $a, b \in \Gamma$ *(send–out communication rules)*.

  (d) $[\, a \,]_h^{\alpha} \to b$, for $h \in H$, $\alpha \in \{+,-,0\}$, $a, b \in \Gamma$ *(dissolution rules)*.

  (e) $[\, a \,]_h^{\alpha_1} \to [\, b \,]_h^{\alpha_2}\ [\, c \,]_h^{\alpha_3}$, for $h \in H$, $\alpha_1, \alpha_2, \alpha_3 \in \{+,-,0\}$, $a, b, c \in \Gamma$ *(division rules for elementary membranes)*.

  (f) $[\,[\,]_{h_1}^{\alpha_1} \ldots [\,]_{h_k}^{\alpha_1}\ [\,]_{h_{k+1}}^{\alpha_2} \ldots [\,]_{h_n}^{\alpha_2}\,]_h^{\alpha} \to [\,[\,]_{h_1}^{\alpha_3} \ldots [\,]_{h_k}^{\alpha_3}\,]_h^{\beta}\ [\,[\,]_{h_{k+1}}^{\alpha_4} \ldots [\,]_{h_n}^{\alpha_4}\,]_h^{\gamma}$, for $k \geq 1$, $n > k$, $h, h_1, \ldots, h_n \in H$, $\alpha, \beta, \gamma, \alpha_1, \ldots, \alpha_4 \in \{+,-,0\}$ and $\{\alpha_1, \alpha_2\} = \{+,-\}$ *(division rules for non–elementary membranes)*.

- The sets $\mathcal{NAM}, \mathcal{AM}(+n)$ and $\mathcal{AM}(-n)$.

P Systems with Active Membranes: limitations

- A deterministic P system with active membranes but without membrane division can be simulated by a DTM with a polynomial slowdown `(Milano theorem)`.

Corollary $\mathbf{PMC}^{*}_{\mathcal{NAM}} \subseteq \mathbf{P}$.

Theorem: $\mathbf{P} = \mathbf{PMC}_{\mathcal{NAM}} = \mathbf{PMC}^{*}_{\mathcal{NAM}}$.

P Systems with Active Membranes: efficiency (I)

- The first efficient solutions to **NP**–complete problems by using P systems with active membranes were given in a *semi–uniform* way (S.N. Krishna and R. Rama (1999), A. Păun, Gh. Păun, C. Zandron, C. Ferretti and G. Mauri (2000), A. Obtulowicz (2001)).
  - $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}^{*}_{\mathcal{AM}(-n)}$
- In the framework of $\mathcal{AM}(-n)$, efficient *uniform* solutions **NP**–complete problems have been given (Seville team (2003), A. Alhazov, C. Martin–Vide, L. Pan (2004), etc.).
  - $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{AM}(-n)}$
- A borderline between efficiency and non–efficiency: division rules in the framework of P systems with active membranes.

P Systems with Active Membranes: efficiency (II)

- In the framework of $\mathcal{AM}(+n)$, P. Sosík (2003) gave an efficient *semi–uniform* solution to QBF-SAT.
  - $\textbf{PSPACE} \subseteq \textbf{PMC}^*_{\mathcal{AM}(+n)}$

  P. Sosík and A. Rodríguez–Patón (2007) have proven that the reverse inclusion holds as well.
  - Nevertheless, the notion of *uniform family* of P systems considered is different, although maybe the proof can be adapted. In this case the following would hold: $\textbf{PSPACE} = \textbf{PMC}^*_{\mathcal{AM}(+n)}$
- The above inclusion has been extended by A. Alhazov, C. Martin–Vide and L. Pan (2003) showing that QBF-SAT can be solved in a linear time and in a *uniform* way.
  - $\textbf{PSPACE} \subseteq \textbf{PMC}_{\mathcal{AM}(+n)}$.

P Systems with Active Membranes: efficiency (III)

- A.E. Porreca, G. Mauri and C. Zandron (2006) described a (deterministic and efficient) algorithm simulating a single computation of any confluent recognizer P system with active membranes and without input. Such P systems can be simulated by a DTM working in exponential time.
  - $\mathbf{PMC}^{*}_{\mathcal{AM}(+n)} \subseteq \mathbf{EXP}$.
  - $\mathbf{PSPACE} \subseteq \mathbf{PMC}_{\mathcal{AM}(+n)} \subseteq \mathbf{PMC}^{*}_{\mathcal{AM}(+n)} \subseteq \mathbf{EXP}$.

- **Conclusion:** the usual framework of $\mathcal{AM}$ for solving decision problems is too powerful from the complexity point of view.

- It would be interesting to investigate weaker models of P systems with active membranes able to characterize classical complexity classes below **NP** and providing borderlines between efficiency and non–efficiency.

Polarizationless P systems with active membrane

- $\Pi = (\Gamma, H, \mu, \mathcal{M}_1, \ldots, \mathcal{M}_q, R, i_{out})$,

  (a) $[\, a \to u \,]_h$, for $h \in H$, $a \in \Gamma$, $u \in \Gamma^*$ (*object evolution rules*).

  (b) $a\,[\;\,]_h \to [\, b \,]_h$, for $h \in H$, $a, b \in \Gamma$ (*send–in communication rules*).

  (c) $[\, a \,]_h \to [\;\,]_h\, b$, for $h \in H$, $a, b \in \Gamma$ (*send–out communication rules*).

  (d) $[\, a \,]_h \to b$, for $h \in H$, $a, b \in \Gamma$ (*dissolution rules*).

  (e) $[\, a \,]_h \to [\, b \,]_h\, [\, c \,]_h$, for $h \in H$, $a, b, c \in \Gamma$ (*division rules for elementary membranes* or weak *division rules for non-elementary membranes*).

  (f) $[\,[\,]_{h_1} \ldots [\,]_{h_k}\, [\,]_{h_{k+1}} \ldots [\,]_{h_n}\,]_h \to [\,[\,]_{h_1} \ldots [\,]_{h_k}\,]_h\, [\,[\,]_{h_{k+1}} \ldots [\,]_{h_n}\,]_h$, for $k \geq 1$, $n > k$, $h, h_1, \ldots, h_n \in H$, (strong *division rules for non–elementary membranes*).

- Rules of type (f) are used only for $k = 1$, $n = 2$, that is, rules of the form (f) $[\,[\;]_{h_1} [\;]_{h_2}\,]_h \to [\,[\;]_{h_1}\,]_h\, [\,[\;]_{h_2}\,]_h$.

  They can also be restricted to the case where they are controlled by the presence of a specific membrane, that is,

  rules of the form (g) $[\,[\;]_{h_1} [\;]_{h_2}\, [\;]_p\,]_h \to [\,[\;]_{h_1} [\;]_p\,]_h\, [\,[\;]_{h_2} [\;]_p\,]_h$.

- The sets $\mathcal{NAM}^0, \mathcal{AM}^0(\alpha, \beta, \gamma, \delta)$, where $\alpha \in \{-d, +d\}$, $\beta \in D = \{-n, +nw, +ns, +nsw, +nsr\}$, $\gamma \in \{-e, +e\}$, and $\delta \in \{-c, +c\}$.

A conjecture of Păun

At the beginning of 2005, Gh. Păun (problem **F** from [1]) wrote:

> *My favorite question (related to complexity aspects in P systems with active membranes and with electrical charges) is that about the number of polarizations. Can the polarizations be completely avoided? The feeling is that this is not possible – and such a result would be rather sound: passing from no polarization to two polarizations amounts to passing from non–efficiency to efficiency.*

The so–called Păun's conjecture can be formally formulated:

$$\mathbf{P} = \mathbf{PMC}^{[*]}_{\mathcal{AM}^0(+d,-n,+e,+c)}$$

[1] Gh. Păun: Further twenty six open problems in membrane computing. *Third Brainstorming Week on Membrane Computing* (M.A. Gutiérrez et al. eds.), Fénix Editora, Sevilla, 2005, pp. 249–262.

A partial affirmative answer

- Non–efficiency of polarizationless P systems with active membranes which do not make use of dissolution rules (Seville team, 2006):

Theorem: $\mathbf{P} = \mathbf{PMC}^{[*]}_{\mathcal{AM}^0(-d,\beta,+e,+c)}$, where $\beta \in D$.

  - The notion of dependency graph:
    - Simulating accepting computations in $\mathcal{AM}^0(-d,\beta,+e,+c)$ by means of reachability problems in a static directed graph.

N. Murphy and D. Woods (2007) gave a further partial affirmative answer in the case of *symmetric* division rules for elementary membranes: $[\, a\,]_h \rightarrow [\, b\,]_h [\, b\,]_h$.

  - $\mathbf{P} = \mathbf{PMC}^{[*]}_{\mathcal{AM}^0(+d,-n(sym),+e,+c)}$.

D. Woods, N. Murphy, M.J. Pérez, A. Riscos (2009) have provided a **P** upper bound on systems from $\mathcal{AM}^0(+d,-n,-e,-c)$, having an initial membrane structure that is a single (linear) path ($\mathcal{D}$).

  - $\mathbf{P} = \mathbf{PMC}^{[*]}_{\mathcal{D}}$.

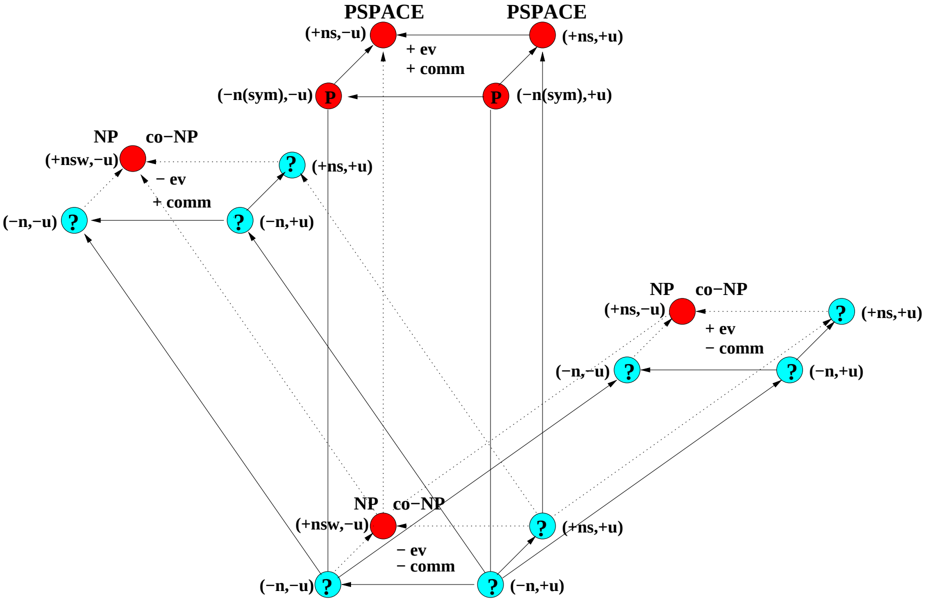
A partial negative answer

- Efficiency of polarizationless P systems with active membranes when dissolution and division for non–elementary membranes, in the strong sense, is permitted (A. Alhazov, L. Pan (SAT, 2004), Seville team (SS, 2006))
  - $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}^*_{\mathcal{AM}^0(+d,+ns,+e,+c)}$.

- A new borderline in $\mathcal{AM}^0(+ns,+e,+c)$: dissolution rules.

  This result has been improved (A. Alhazov, M.J. Pérez (QBF–SAT, 2007)):
  - $\mathbf{PSPACE} \subseteq \mathbf{PMC}_{\mathcal{AM}^0(+d,+ns,+e,+c)}$.

- Other interesting results: forbidding evolution and/or communication rules.
  - $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}^*_{\mathcal{AM}^0(+d,\beta,+e,-c)}$, where $\beta \in \{+nw, +ns\}$ (A. Alhazov, L. Pan, Gh. Păun, 2004).
  - $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}^*_{\mathcal{AM}^0(+d,+nsw,-e,-c)}$ (Milano team and Seville team, 2008).
  - $\mathbf{PSPACE} \subseteq \mathbf{PMC}^*_{\mathcal{AM}^0(+d,+nsr,-e,-c)}$ (Milano team and Seville team, 2008).



Polarizationless active membranes *using dissolution rules*

Tissue–like Framework

- *Polarizationless tissue P system of degree $q \geq 1$ with cell division*:

$$\Pi = (\Gamma, \Sigma, \Omega, \mathcal{M}_1, \dots, \mathcal{M}_q, R, i_{in}, i_{out})$$

  - Γ is a finite alphabet (called working alphabet) whose elements are called objects;
  - Σ is a finite alphabet (called input alphabet) strictly contained in Γ ;
  - $\Omega \subseteq \Gamma \setminus \Sigma$ is a finite alphabet, describing the set of objects located in the environment in arbitrarily many copies each;
  - $\mathcal{M}_1, \dots, \mathcal{M}_q$ are strings over $\Gamma \setminus \Sigma$, describing the *multisets of objects* placed in the q cells of the system;
  - R is a finite set of rules, of the following forms:
    - $(i, u/v, j)$, for $i, j \in \{0, 1, 2, \dots, q\}$, $i \neq j$, and $u, v \in \Gamma^*$ communication rules
    - $[\ a\ ]_i \rightarrow [\ b\ ]_i [\ c\ ]_i$, where $i \in \{1, 2, \dots, q\}$ and $a, b, c \in \Gamma$ *division rules;*.
  - $i_{in} \in \{1, \dots, q\}$ is the input cell, and $i_{out} \in \{0, 1, \dots, q\}$ is the output cell.

- *Length* of the communication rule $(i, u/v, j) = |u| + |v|$.

- Result of the halting computation $\mathcal{C} = \{C_i\}_{i<r}$:

$$Output(\mathcal{C}) = \Psi_{\Gamma \setminus \Omega}(M_{r-1,0})$$

where Ψ is the Parikh function, and $M_{r-1,0}$ is the multiset associated with the environment at C_{r-1}.

- Recognizer tissue P system with cell division

$$\Pi = (\Gamma, \Sigma, \Omega, \mathcal{M}_1, \ldots, \mathcal{M}_q, R, i_{in})$$

  - $\Gamma \setminus \Omega$ has two distinguished objects yes and no, present in at least one copy in some initial multisets $\mathcal{M}_1, \ldots, \mathcal{M}_q$.
  - All computations halt.
  - For each computation, either yes or no (but not both) must have been released into the environment (only in the last step of the computation).

- Result of the halting computation $\mathcal{C} = \{C_i\}_{i<r}$:

$$Output(\mathcal{C}) = \begin{cases} \texttt{yes}, & \text{if } \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{r-1,0}) = (1,0) \\ & \wedge\ \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{k,0}) = (0,0) \text{ for } k = 0, \dots, r-2 \\ \texttt{no}, & \text{if } \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{r-1,0}) = (0,1) \\ & \wedge\ \Psi_{\{\texttt{yes},\texttt{no}\}}(M_{k,0}) = (0,0) \text{ for } k = 0, \dots, r-2 \end{cases}$$

where Ψ is the Parikh function, and $M_{i,0}$ is the multiset associated with the environment at C_i.

- The sets $\mathcal{TC}$, $\mathcal{TDC}$ and $\mathcal{TDC}(k)$.

Cell separation rules

- $[\ a\ ]_i \rightarrow [\ \Gamma_1\ ]_i [\ \Gamma_2\ ]_i$

  where:

  - $i \in \{1, 2, \ldots, q\}$.
  - $a \in \Gamma$.
  - $\{\Gamma_1, \Gamma_2\}$ is a fixed partition of Γ.

- The set $\mathcal{TSC}(k)$.

Polynomial–Time Solvability

- A decision problem $X = (I_X, \theta_X)$ is solvable in polynomial time by a family $\mathbf{\Pi} = \{\Pi(n) : n \in \mathbb{N}\}$ of recognizer tissue P systems if:
  - The family $\mathbf{\Pi}$ is *polynomially uniform* by Turing machines.
  - There exists a pair (cod, s) of polynomial-time computable functions over I_X such that:
    - For each $u \in I_X$, $s(u) \in \mathbf{N}$ and $cod(u)$ is an input multiset of $\Pi(s(u))$.
    - The family $\mathbf{\Pi}$ is *polynomially bounded* with regard to (X, cod, s).
    - The family $\mathbf{\Pi}$ is *sound* and *complete* with regard to (X, cod, s).

- The complexity class $\mathbf{PMC}_{\mathcal{R}}$.

Efficiency of Tissue P Systems with cell division

- An efficient solution of the Vertex Cover problem was given[2] by using a family of recognizer tissue P systems from $\mathcal{TDC}(3)$.

  ⋆ $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{TDC}(3)}$.

- This result has been improved recently[3]

  ⋆ $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{TDC}(2)}$.

- BUT ...

  ⋆ $\mathbf{P} = \mathbf{PMC}_{\mathcal{TDC}(1)}$.

  (dependency graph technique)

[2] D. Díaz–Pernil, M.J. Pérez–Jiménez, A. Riscos–Núñez and A. Romero–Jiménez. Computational Efficiency of Cellular Division in Tissue-like Membrane Systems. *Romanian Journal of Information Science and Technology*, **11**, 3 (2008), 229–241.

[3] A.E. Porreca, N. Murphy, M.J. Pérez-Jiménez. submitted, 2011.

Tissue without cell division vs Basic transition (I)

- A family of recognizer tissue from $\mathcal{TC}$ which solves a decision problem can be efficiently simulated by a family from $\mathcal{T}$ solving the same problem.
  - Π' *efficiently simulates* Π if:
    - ⋆ Π' can be constructed from Π by a DTM working in polynomial time.
    - ⋆ There exists a bijective function, f, from $\textbf{Comp}(\Pi)$ onto $\textbf{Comp}(\Pi')$ such that:
      - ⋆ $\mathcal{C} \in \textbf{Comp}(\Pi)$ is an accepting computation iff $f(\mathcal{C}) \in \textbf{Comp}(\Pi')$ is an accepting one.
      - ⋆ There exists a polynomial $p(n)$ such that for each $\mathcal{C} \in \textbf{Comp}(\Pi)$ we have $|f(\mathcal{C})| \leq p(|\mathcal{C}|)$.

Tissue without cell division vs Basic transition (II)

Let $\Pi = (\Gamma, \Sigma, \Omega, \mathcal{M}_1, \dots, \mathcal{M}_q, \mathcal{R}, i_{in}) \in \mathcal{TC}$.
Let us consider $S(\Pi) = (\Gamma', \Sigma', \mu, \mathcal{M}'_1, \mathcal{R}', i'_{in}) \in \mathcal{T}$ defined as follows:

- $\Gamma' = \{(a, i) \,:\, a \in \Gamma \,\wedge\, i \in \{1, \dots, q\}\} \cup \{(a, 0) \,:\, a \in \Gamma \setminus \Omega\} \cup \{\mathtt{yes}, \mathtt{no}\}$.
- $\Sigma' = \{(a, i_{in}) \,:\, a \in \Sigma\}$.
- $\mu = [\;]_1$.
- $\mathcal{M}'_1 = \sum_{i=1}^{q} \sum_{a \in \Gamma \setminus \Sigma} (a, i)^{\mathcal{M}_i(a)}$.
- In the set $\mathcal{R}'$ the following rules associated with $S(\Pi)$ are included:
  - For each rule $r_\Pi \equiv (i, a_1 \dots a_m \,/\, b_1 \dots b_n, j) \in \mathcal{R}$ with $i, j \neq 0$, we consider the rule $r_{S(\Pi)}$:
    $(a_1, i) \dots (a_m, i)(b_1, j) \dots (b_n, j) \to (b_1, i) \dots (b_n, i)(a_1, j) \dots (a_m, j)$
  - For each rule $r_\Pi \equiv (i, a_1 \dots a_m \,/\, b_1 \dots b_n, 0) \in \mathcal{R}$ with $i \neq 0$, we consider the rule $r_{S(\Pi)}$:
    $(a_1, i) \dots (a_m, i)(b_1, 0) \dots (b_s, 0) \to (b_1, i) \dots (b_n, i)(a_1, 0) \dots (a_r, 0)$
    where $a_1, \dots, a_r, b_1, \dots, b_s \notin \Omega$ and $a_{r+1}, \dots, a_m, b_{s+1}, \dots, b_n \in \Omega$.
  - For each rule $r_\Pi \equiv (0, a_1 \dots a_m \,/\, b_1 \dots b_n, i) \in \mathcal{R}$ with $i \neq 0$, we consider the rule $r_{S(\Pi)}$:
    $(a_1, 0) \dots (a_r, 0)(b_1, i) \dots (b_n, i) \to (b_1, 0) \dots (b_s, 0)(a_1, i) \dots (a_m, i)$
    where $a_1, \dots, a_r, b_1, \dots, b_s \notin \Omega$ and $a_{r+1}, \dots, a_m, b_{s+1}, \dots, b_n \in \Omega$.
  - $(\mathtt{yes}, 0) \to (\mathtt{yes}, out)$; $(\mathtt{no}, 0) \to (\mathtt{no}, out)$.
- $i'_{in} = 1$.

Tissue without cell division vs Basic transition (III)

Proposition: If $\Pi \in \mathcal{TC}$, then $S(\Pi) \in \mathcal{T}$ efficiently simulates Π.

$$\boxed{\mathbf{P} = \mathbf{PMC}_{\mathcal{TC}}.}$$

Efficiency of Tissue P Systems with cell separation

- An efficient solution of the SAT problem was given[4] by using a family of recognizer tissue P systems from $\mathcal{TSC}(6)$.

  ⋆ $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{TSC}(6)}$.

- This result has been improved recently[5]

  ⋆ $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{TSC}(3)}$.

- BUT ...

  ⋆ $\mathbf{P} = \mathbf{PMC}_{\mathcal{TSC}(1)}$.

  (dependency graph technique[3])

[4] L. Pan, M.J. Pérez–Jiménez. Computational Complexity of tissue–like P systems with cell separation. *Journal of Complexity*, **26** (2010), 296–315

[5] M.J. Pérez–Jiménez, P. Sosik. On the efficiency of tissue P systems with cell separation, submitted 2011.

Open problems (I)

(1) Is there some class $\mathcal{R}$ of recognizer P systems such that the inclusion $\mathbf{PMC}_{\mathcal{R}} \subseteq \mathbf{PMC}^{*}_{\mathcal{R}}$ is strict?

(2) Is it possible to efficiently solve **PSPACE**–complete problems by using families of P systems from $\mathcal{AM}(-n)$?

(3) Is $\mathbf{P} = \mathbf{PMC}^{[*]}_{\mathcal{AM}^0(+d,-n,+e,+c)}$ true? (Păun's conjecture).

(4) It is well known that $\mathbf{PSPACE} \subseteq \mathbf{PMC}^{*}_{\mathcal{AM}^0(+d,+nsr,-e,-c)}$. Determine an upper bound for that membrane computing complexity class.

Open problems (II)

(5) What is the efficiency of P systems from $\mathcal{AM}^0(\alpha, \beta, -e, -c)$? Are there any relations with the results obtained for polarizationless P systems?

(6) Is it $\mathbf{NP} \cup \mathbf{co\text{-}NP} \subseteq \mathbf{PMC}_{\mathcal{TSC}(2)}$?

(7) Would it be possible to solve efficently **NP**–complete problems by families from $\mathcal{TDC}(2)$ where all rules of length 3 were symport?
What about $\mathcal{TSC}(3)$?

(8) Would it be possible to solve efficently **NP**–complete problems by families from $\mathcal{TDC}(2)$ or $\mathcal{TSC}(3)$ where the environment is passive (as in cell–like P systems)?