P systems simulating oracle computations

Antonio E. Porreca   Alberto Leporati   Giancarlo Mauri
Claudio Zandron

Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano-Bicocca, Italy

12th Conference on Membrane Computing
Fontainebleau, France, 24 August 2011

Summary

- We show how to reuse existing recogniser P systems as "subroutines"
- This allows us to simulate oracles
- The procedure is quite general (though technical details may vary)
- As an application, we improve the lower bound on $\mathbf{PMC}_{\mathcal{AM}(-\mathrm{d},-\mathrm{n})}$ to $\mathbf{P}^{\mathbf{PP}}$

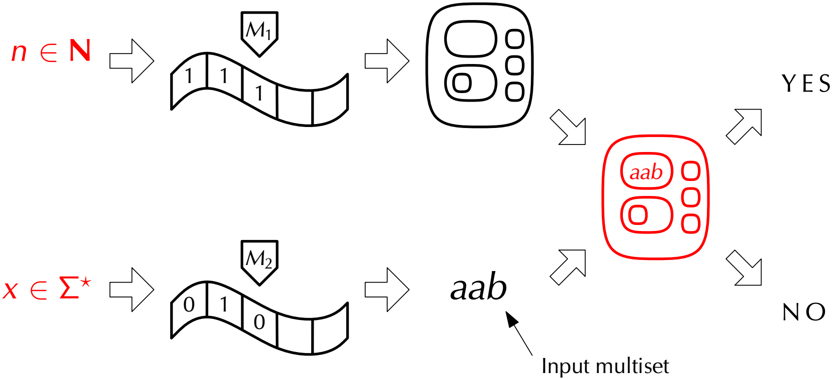
P systems with active membranes

- Known for their ability to solve computationally hard problems
- Here we focus on restricted elementary membranes (no nonelementary division, no dissolution)

Object evolution	$[a \to w]_h^\alpha$
Communication (send-in)	$a\ [\ ]_h^\alpha \to [b]_h^\beta$
Communication (send-out)	$[a]_h^\alpha \to [\ ]_h^\beta\ b$
Elementary division	$[a]_h^\alpha \to [b]_h^\beta\ [c]_h^\gamma$

Uniform families of recogniser P systems

- For each input length $n = |x|$ we construct a P system Π_n receiving as input a multiset encoding x
- Both are constructed by fixed polytime Turing machines
- The resulting P system decides if $x \in L$



The complexity class $\mathbf{PMC}_{\mathcal{AM}(-\mathrm{d},-\mathrm{n})}$

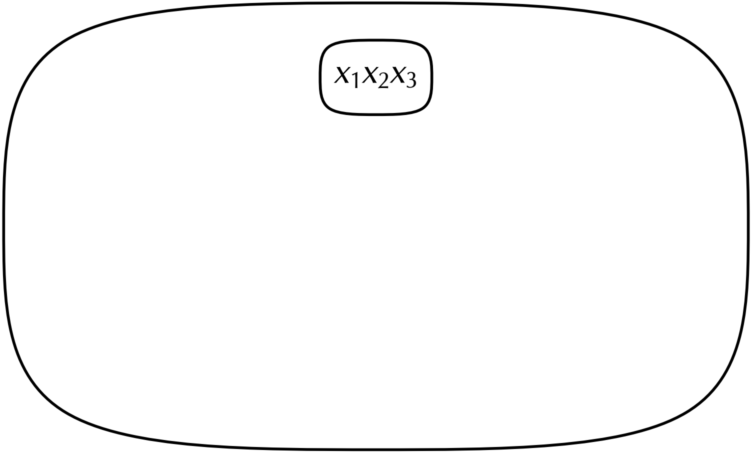
It consists of the languages recognised in polytime by uniform families of P systems with restricted elementary membranes

- Contains **NP** problems [Zandron et al. 2000] (semi-uniform solution)
- Contains **NP** problems [Pérez-Jiménez et al. 2003] (first uniform solution)
- Is contained in **PSPACE** [Sosík, Rodríguez-Patón 2007]
- Contains **PP** problems [Porreca et al. 2010, 2011]

On the other hand, by using nonelementary division (class $\mathbf{PMC}_{\mathcal{AM}}$) we obtain exactly **PSPACE**

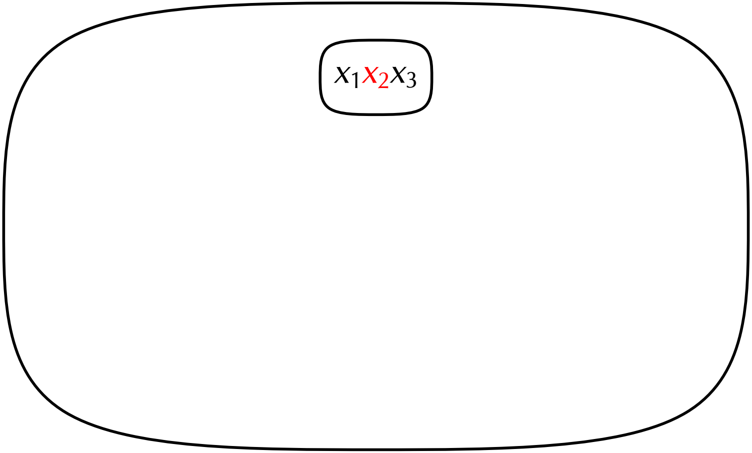
Solving 3SAT

Is $\varphi(x_1, x_2, x_3)$ satisfiable?



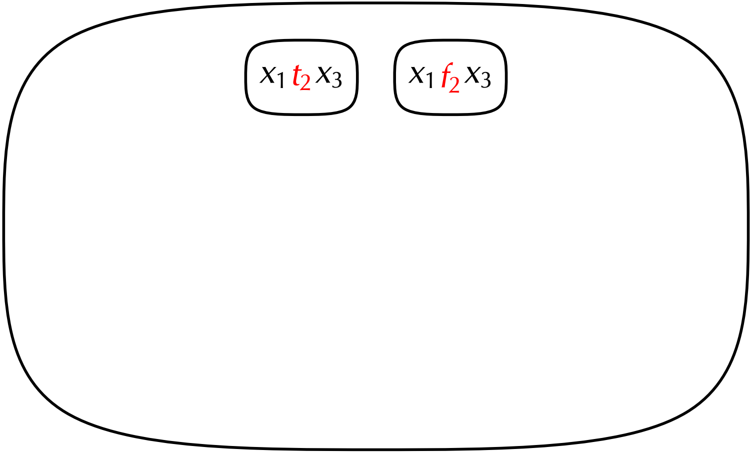
Solving 3SAT

Is $\varphi(x_1, x_2, x_3)$ satisfiable?



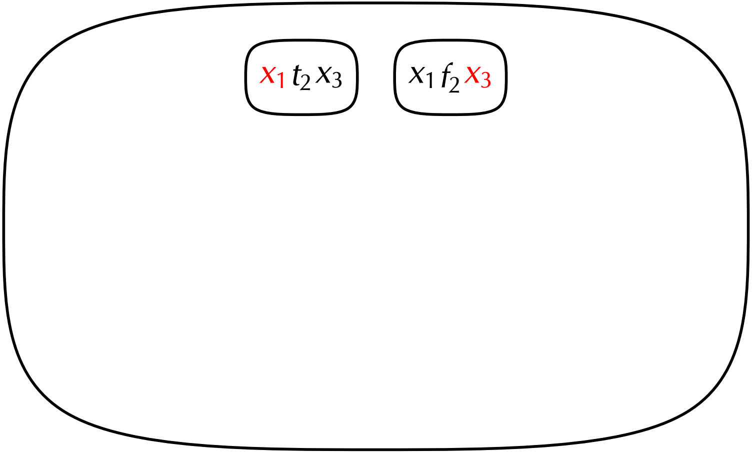
Solving 3SAT

Is $\varphi(x_1, x_2, x_3)$ satisfiable?



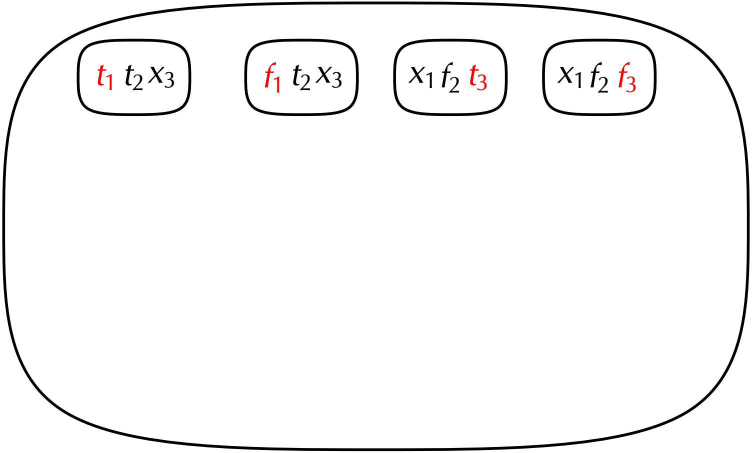
Solving 3SAT

Is $\varphi(x_1, x_2, x_3)$ satisfiable?



Solving 3SAT

Is $\varphi(x_1, x_2, x_3)$ satisfiable?



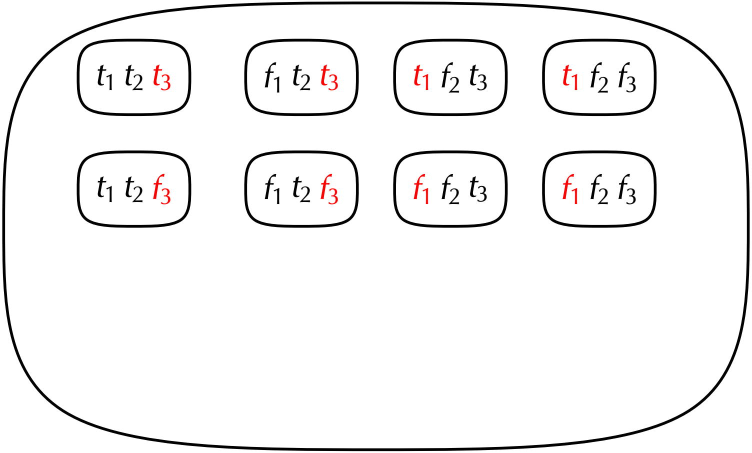
Solving 3SAT

Is $\varphi(x_1, x_2, x_3)$ satisfiable?

$\boxed{t_1\, t_2\, x_3}$ $\boxed{f_1\, t_2\, x_3}$ $\boxed{x_1\, f_2\, t_3}$ $\boxed{x_1\, f_2\, f_3}$

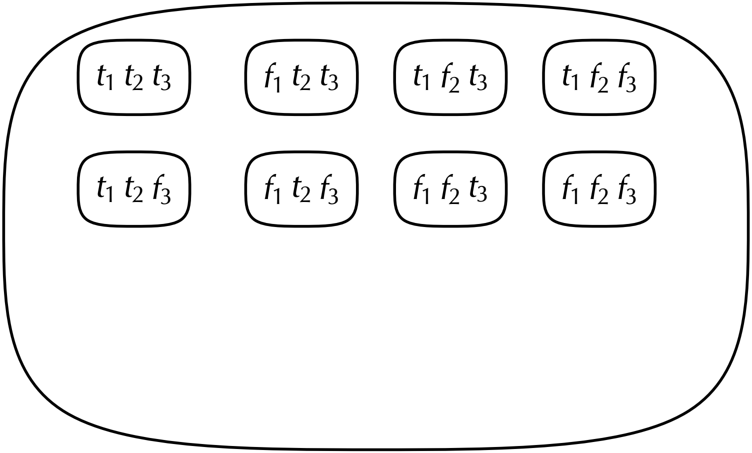
Solving 3SAT

Is $\varphi(x_1, x_2, x_3)$ satisfiable?



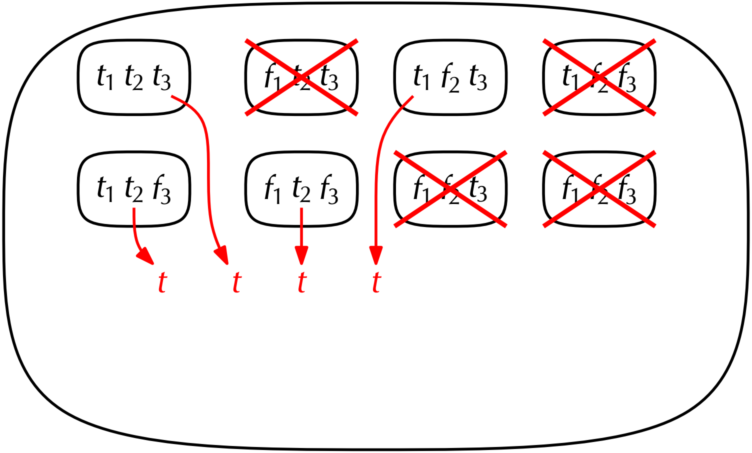
Solving 3SAT

Is $\varphi(x_1, x_2, x_3)$ satisfiable?



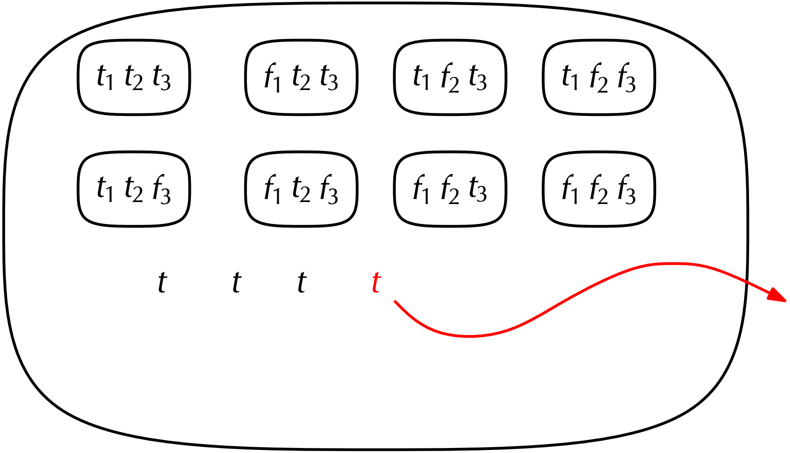
Solving 3SAT

Is $\varphi(x_1, x_2, x_3)$ satisfiable?



Solving 3SAT

Is $\varphi(x_1, x_2, x_3)$ satisfiable?



Solving 3SAT

Is $\varphi(x_1, x_2, x_3)$ satisfiable?

$t_1\ t_2\ t_3$ $f_1\ t_2\ t_3$ $t_1\ f_2\ t_3$ $t_1\ f_2\ f_3$

$t_1\ t_2\ f_3$ $f_1\ t_2\ f_3$ $f_1\ f_2\ t_3$ $f_1\ f_2\ f_3$

t t t

YES

The complexity class **PP**

Definition

$L \in \mathbf{PP}$ if it is accepted in polytime by a nondeterministic TM such that more than half of its computations are accepting

Solving **PP** is "essentially the same as" counting the number of solutions

Problem (Threshold-3SAT)

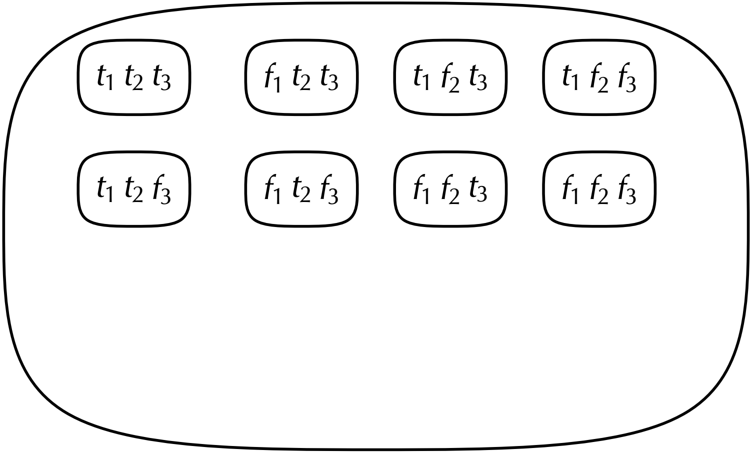
Given a Boolean formula φ over m variables and an integer $k < 2^m$, do more than k assignments out of 2^m satisfy φ?

Theorem

Threshold-3SAT *is* **PP**-*complete* $\square$

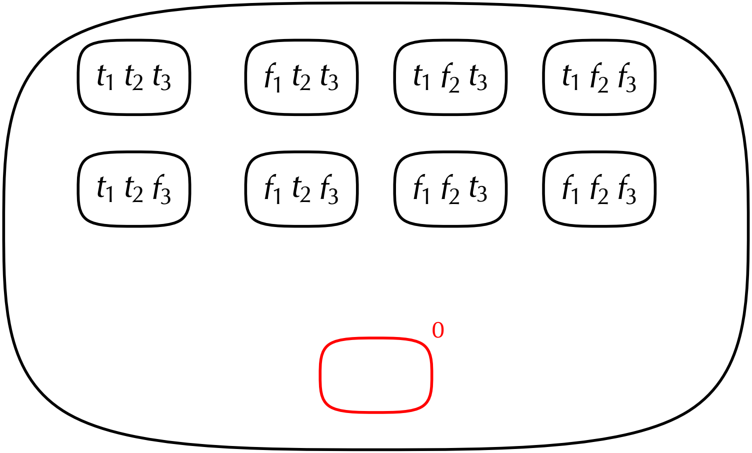
Solving THRESHOLD-3SAT

Does $\varphi(x_1, x_2, x_3)$ have more than 3 satisfying assignments?



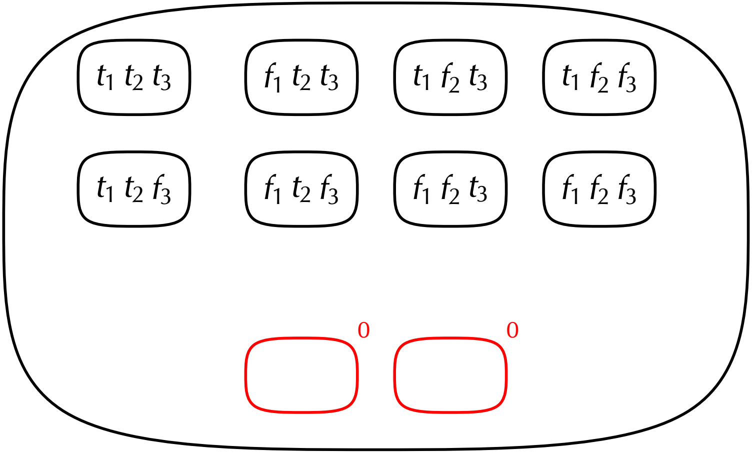
Solving THRESHOLD-3SAT

Does $\varphi(x_1, x_2, x_3)$ have more than 3 satisfying assignments?



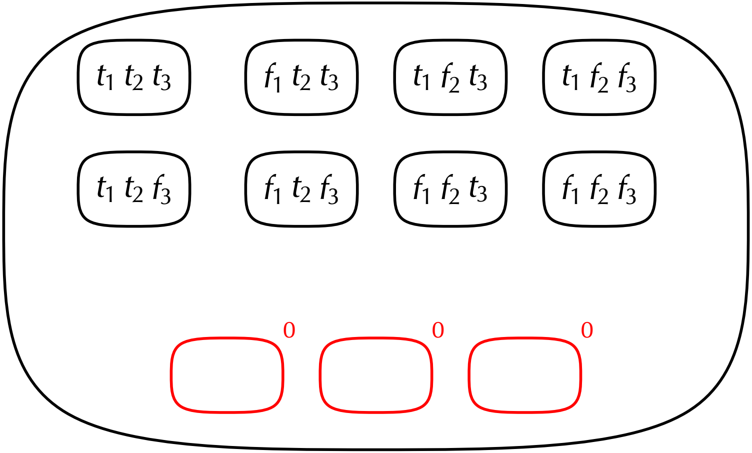
Solving THRESHOLD-3SAT

Does $\varphi(x_1, x_2, x_3)$ have more than 3 satisfying assignments?



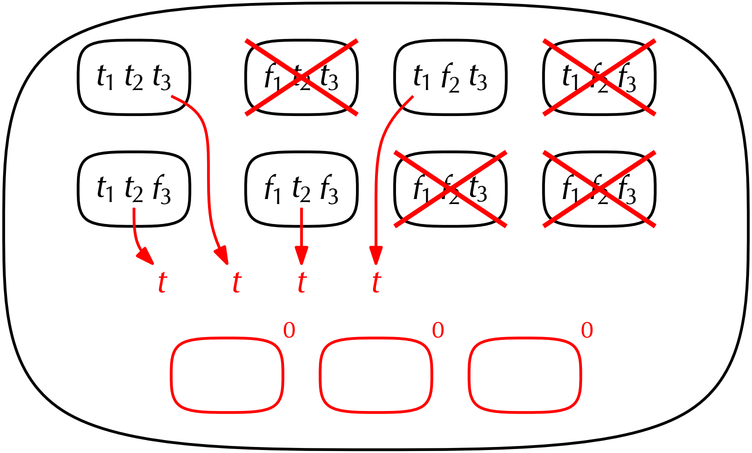
Solving THRESHOLD-3SAT

Does $\varphi(x_1, x_2, x_3)$ have more than 3 satisfying assignments?



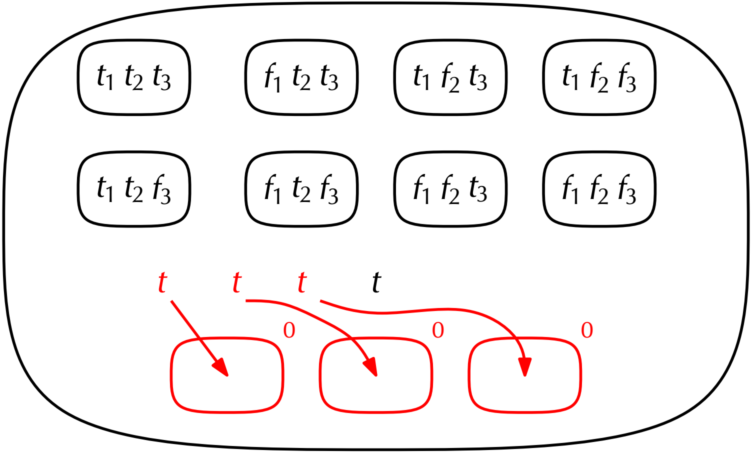
Solving THRESHOLD-3SAT

Does $\varphi(x_1, x_2, x_3)$ have more than 3 satisfying assignments?



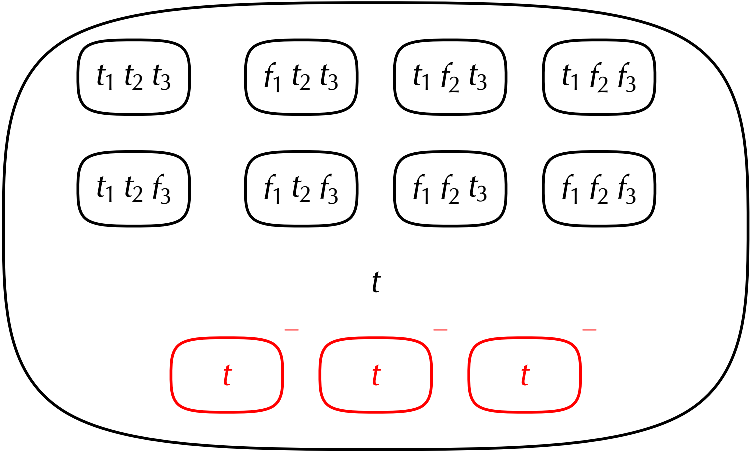
Solving THRESHOLD-3SAT

Does $\varphi(x_1, x_2, x_3)$ have more than 3 satisfying assignments?



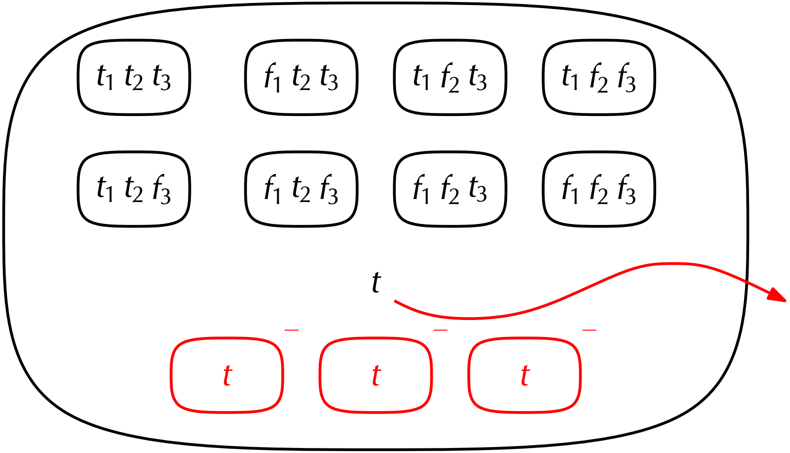
Solving THRESHOLD-3SAT

Does $\varphi(x_1, x_2, x_3)$ have more than 3 satisfying assignments?



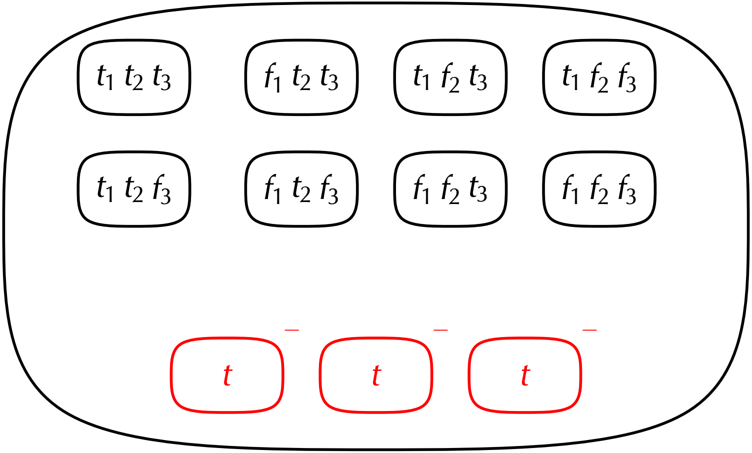
Solving THRESHOLD-3SAT

Does $\varphi(x_1, x_2, x_3)$ have more than 3 satisfying assignments?



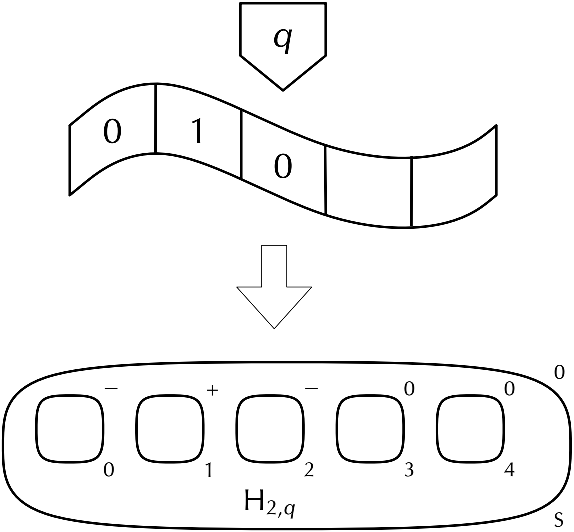
Solving THRESHOLD-3SAT

Does $\varphi(x_1, x_2, x_3)$ have more than 3 satisfying assignments?



YES

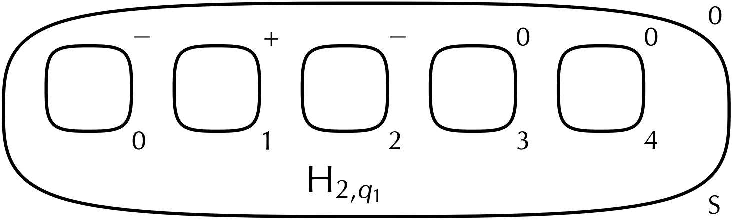
Simulating Turing machines I



Simulating Turing machines II

$$\delta(q_1, 0) = (q_2, 1, \triangleright)$$

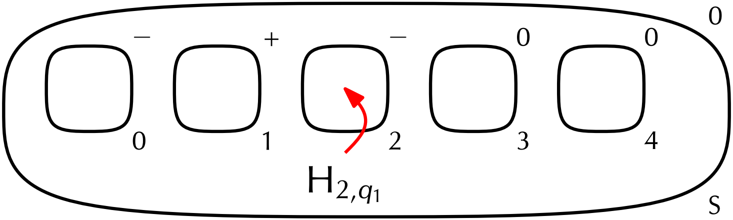
$$\begin{cases} \mathsf{H}_{q_1,i},\ [\ ]_i^- \rightarrow [\mathsf{H}_{q_2,i+1}]_i^+ \\ [\mathsf{H}_{q_2,i+1}]_i^+ \rightarrow [\ ]_i^+\ \mathsf{H}_{q_2,i+1} \end{cases}$$



Simulating Turing machines II

$$\delta(q_1, 0) = (q_2, 1, \triangleright)$$

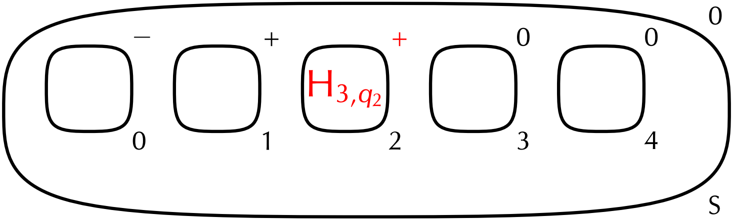
$$\begin{cases} \mathrm{H}_{q_1,i},\ [\ ]_i^- \to [\mathrm{H}_{q_2,i+1}]_i^+ \\ [\mathrm{H}_{q_2,i+1}]_i^+ \to [\ ]_i^+\ \mathrm{H}_{q_2,i+1} \end{cases}$$



Simulating Turing machines II

$$\delta(q_1, 0) = (q_2, 1, \triangleright)$$

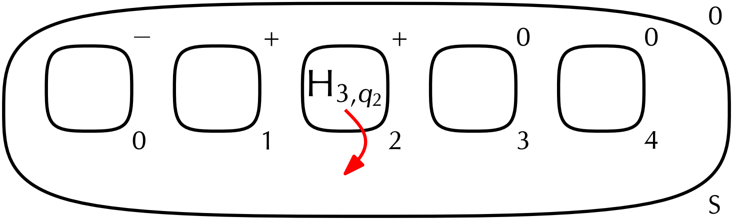
$$\begin{cases} \mathrm{H}_{q_1,i},\ [\ ]_i^- \to [\mathrm{H}_{q_2,i+1}]_i^+ \\ [\mathrm{H}_{q_2,i+1}]_i^+ \to [\ ]_i^+\ \mathrm{H}_{q_2,i+1} \end{cases}$$



Simulating Turing machines II

$$\delta(q_1, 0) = (q_2, 1, \triangleright)$$

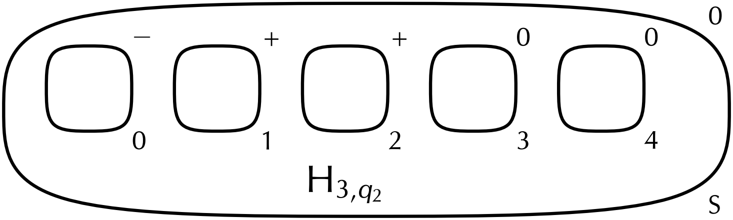
$$\begin{cases} \mathrm{H}_{q_1,i},\ [\ ]_i^- \rightarrow [\mathrm{H}_{q_2,i+1}]_i^+ \\ [\mathrm{H}_{q_2,i+1}]_i^+ \rightarrow [\ ]_i^+\ \mathrm{H}_{q_2,i+1} \end{cases}$$



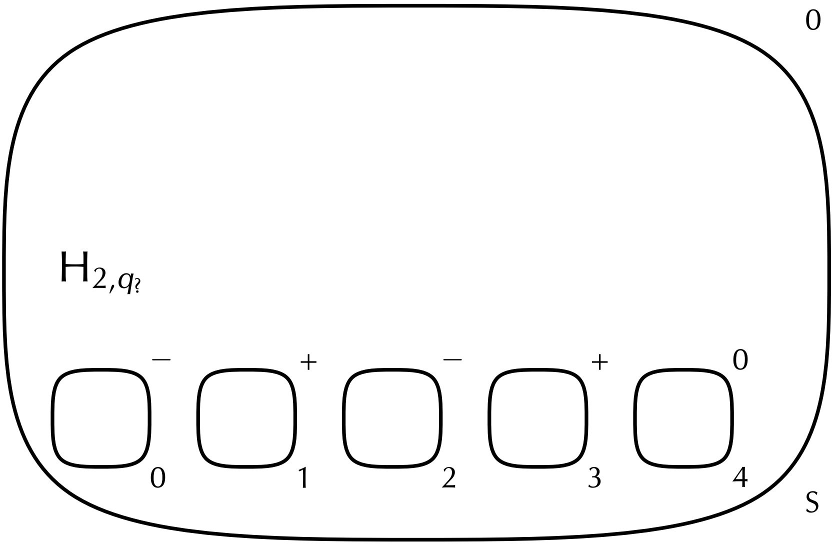
Simulating Turing machines II

$$\delta(q_1, 0) = (q_2, 1, \triangleright)$$

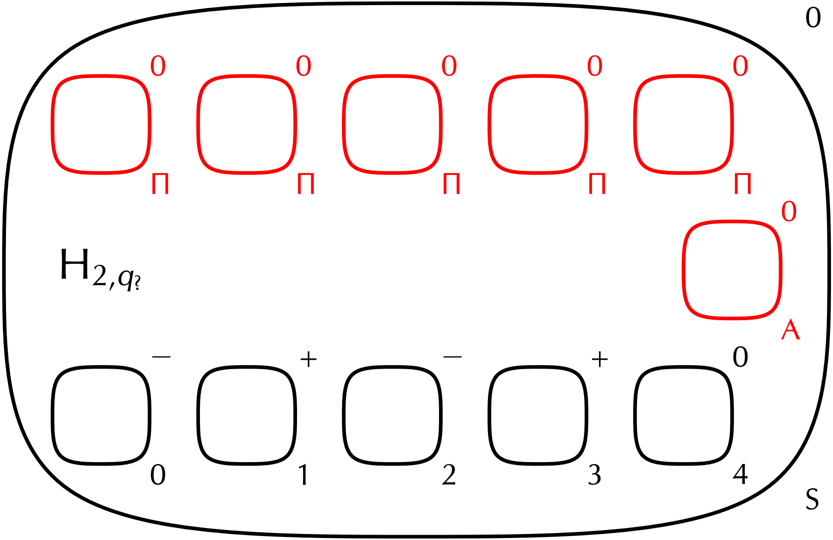
$$\begin{cases} \mathsf{H}_{q_1,i},\ [\ ]_i^- \to [\mathsf{H}_{q_2,i+1}]_i^+ \\ [\mathsf{H}_{q_2,i+1}]_i^+ \to [\ ]_i^+\ \mathsf{H}_{q_2,i+1} \end{cases}$$



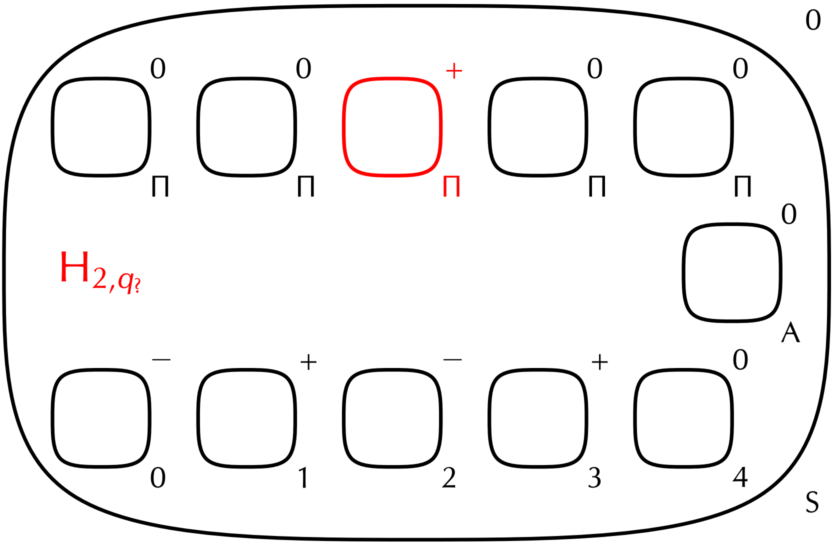
Using P systems as subroutines



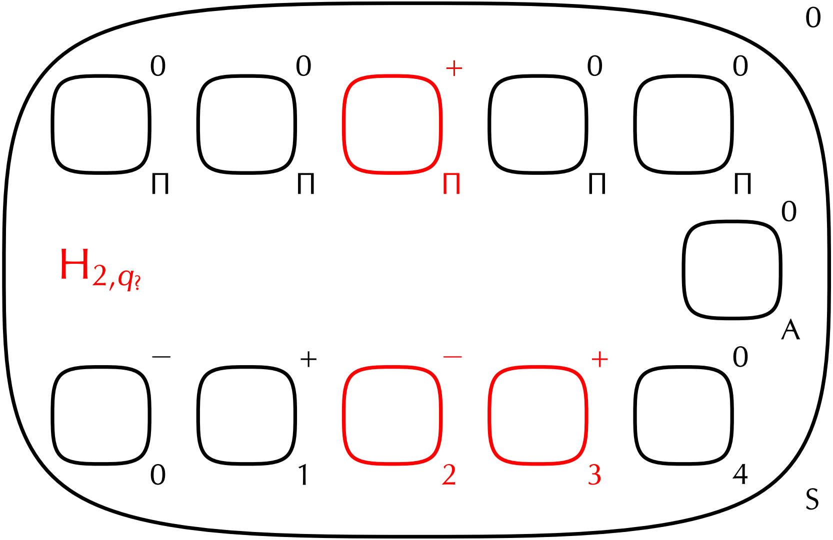
Using P systems as subroutines



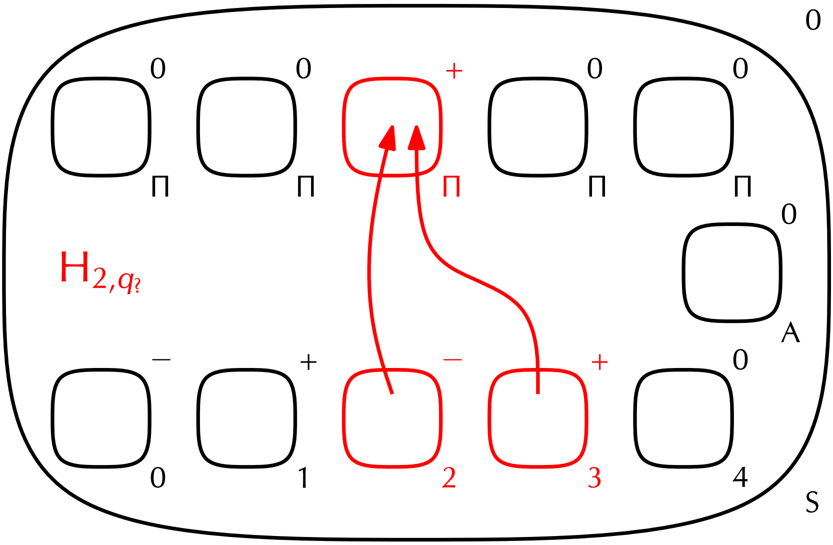
Using P systems as subroutines



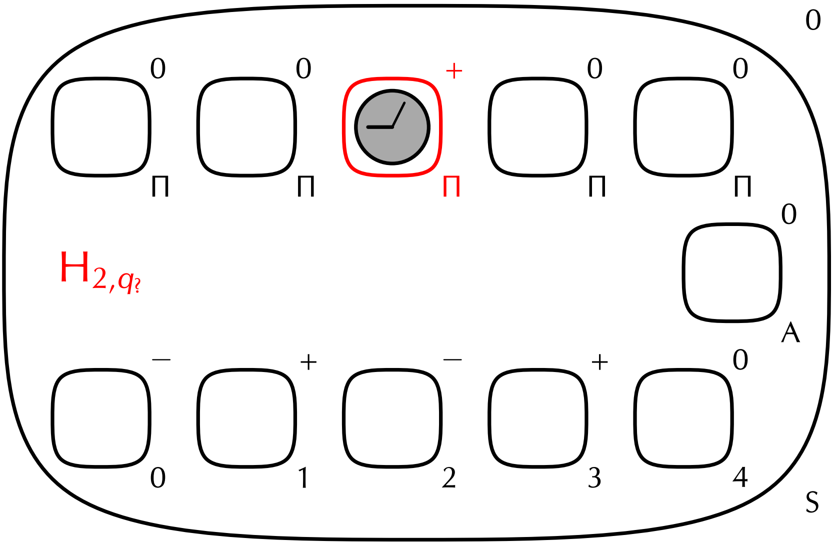
Using P systems as subroutines



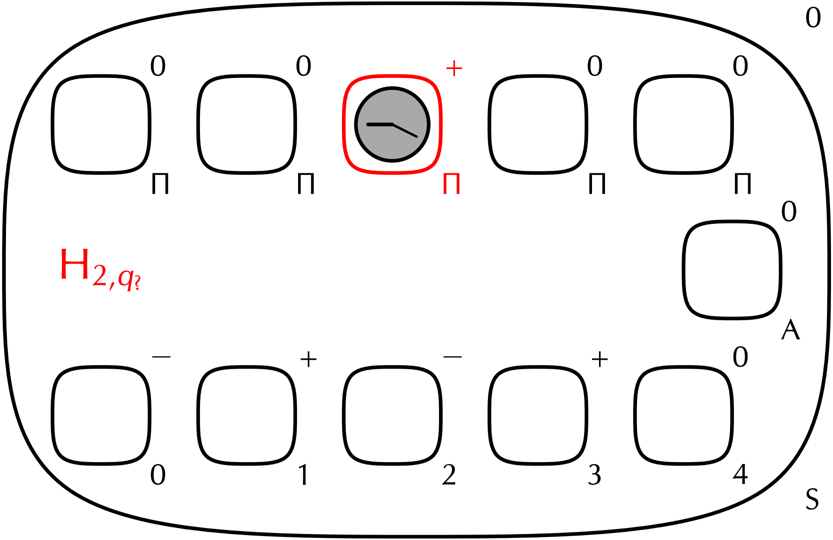
Using P systems as subroutines



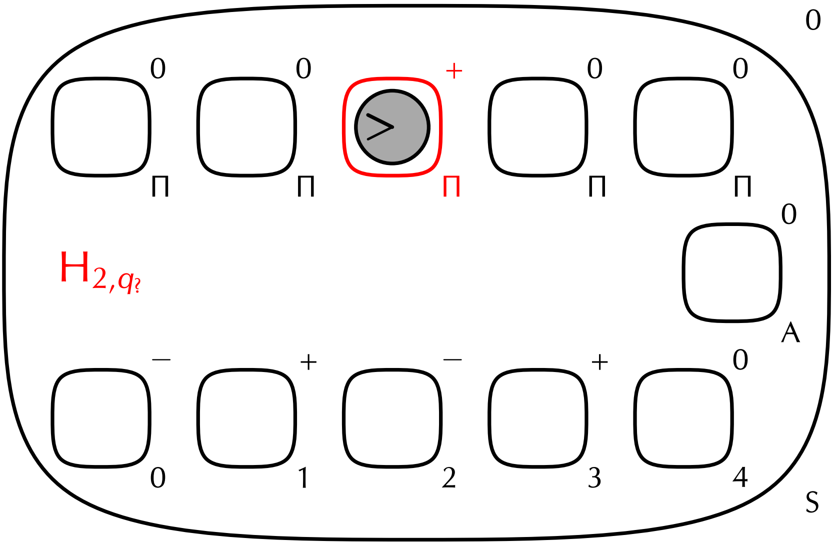
Using P systems as subroutines



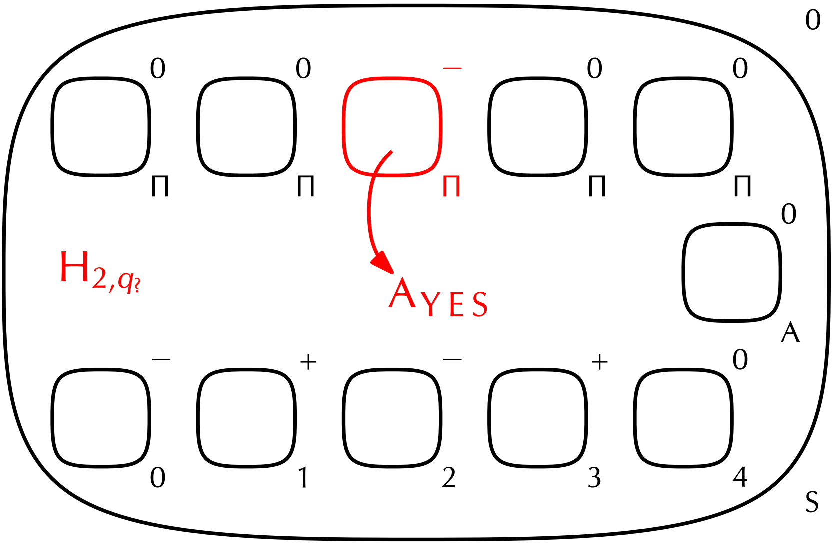
Using P systems as subroutines



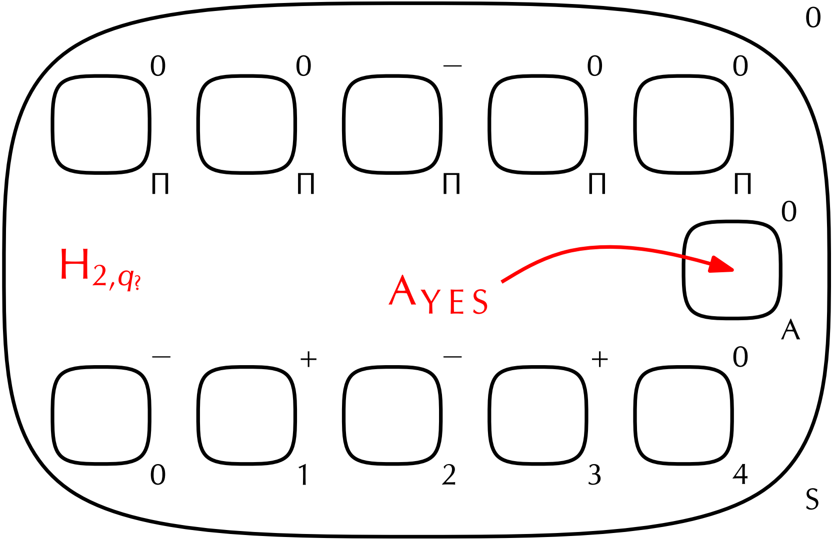
Using P systems as subroutines



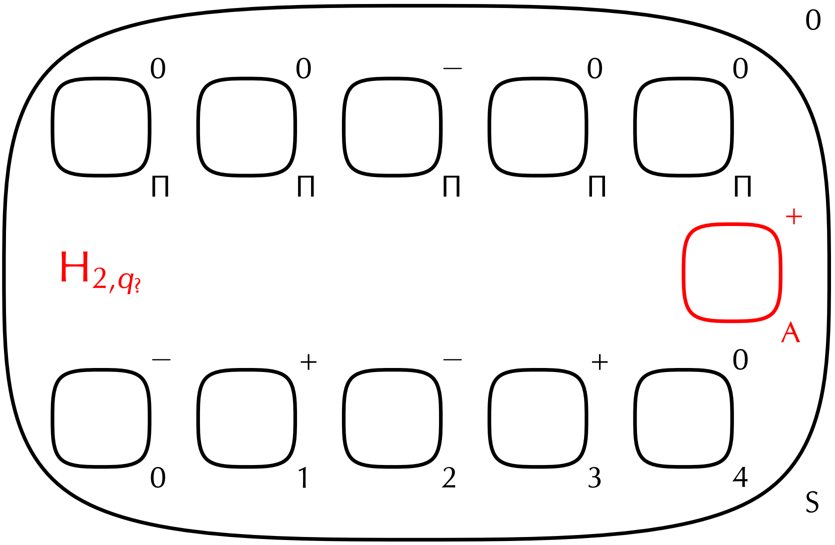
Using P systems as subroutines



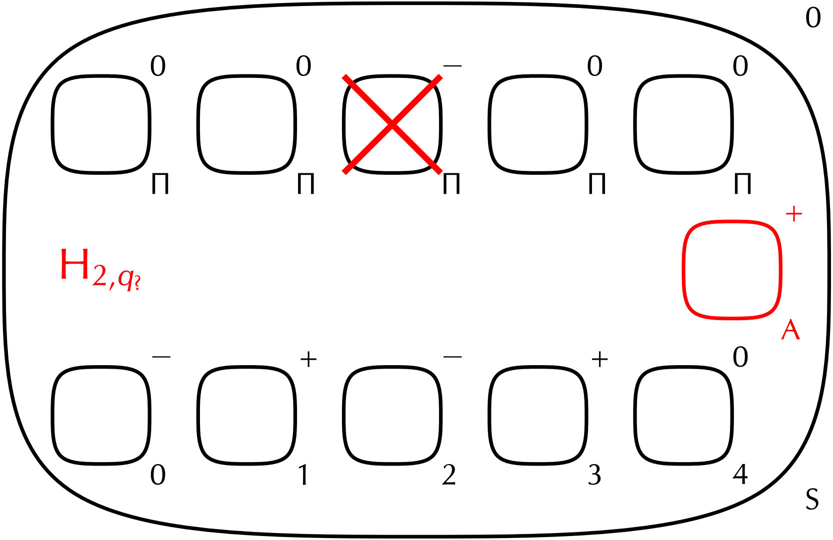
Using P systems as subroutines



Using P systems as subroutines



Using P systems as subroutines



Main result

Theorem

$\mathbf{P}^{\mathbf{PP}} \subseteq \mathbf{PMC}_{\mathcal{AM}(-\mathrm{d},-\mathrm{n})}$

Proof.

- Any polytime TM M with a **PP** oracle can be simulated by a polytime TM M' with an oracle for THRESHOLD-3SAT and only one tape
- Just apply a reduction (which always exists, since THRESHOLD-3SAT is **PP**-complete) before querying the oracle
- And we know how to simulate the TM M' with a polytime $\mathcal{AM}(-\mathrm{d}, -\mathrm{n})$ uniform family. $\square$

Discussion I

- We can solve QSAT (**PSPACE**-complete) by using nonelementary division and a membrane structure of depth $\Theta(n)$
- QSAT instances have an arbitrary number of alternations of quantifiers
- By fixing the first quantifier ($\forall$ or $\exists$) and the number of alternations, we get complete problems for all levels of the polynomial hierarchy
- Formulae with k alternations can be solved by P systems using nonelementary division and a membrane structure of depth $\Theta(k)$
- Notice that k does not depend on the input size

Discussion II

- Let **PH** be the union of the levels of the polynomial hierarchy
- Toda's theorem tells us that $\mathbf{PH} \subseteq \mathbf{P}^{\mathbf{PP}}$
- So we also have $\mathbf{PH} \subseteq \mathbf{PMC}_{\mathcal{AM}(-\mathrm{d},-\mathrm{n})}$
- This means that all levels of the polynomial hierarchy can be solved by using P systems with only elementary division and membrane structure of depth 3
- Does this mean $\mathbf{PSPACE} \subseteq \mathbf{PMC}_{\mathcal{AM}(-\mathrm{d},-\mathrm{n})}$ and so $\mathbf{PSPACE} = \mathbf{PMC}_{\mathcal{AM}(-\mathrm{d},-\mathrm{n})}$?
- Not immediately: **PH** is not known neither conjectured to be **PSPACE**

Open problems

- Prove that we can always do the oracle simulation
- If we can reset the “oracle P systems”
  then we only need a single copy of it
- It might still be possible that $\mathbf{PMC}_{\mathcal{AM}(-\mathrm{d},-\mathrm{n})} = \mathbf{PSPACE}$
  even if $\mathbf{PH} \neq \mathbf{PSPACE}$
- But maybe it would be more interesting if it turns out
  that $\mathbf{PMC}_{\mathcal{AM}(-\mathrm{d},-\mathrm{n})} = \mathbf{P}^{\mathbf{PP}}$

Merci de votre attention!

Thanks for your attention!