Space complexity of membrane systems

Antonio E. Porreca
porreca@disco.unimib.it

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

RGNC Seminar
Universidad de Sevilla
10th February 2011

Introduction

- Some variants of P systems solve classically intractable problems in polynomial time
- E.g., active membranes, tissues with cell division
- The key is trading space for time: generate exponentially many membranes (processing units) working in a maximally parallel way

Motivation

- The space/time trade-off was always described in a rather informal way
- What exactly is space in a P system?
- We want to formalise the notion of space complexity, in order to be able to prove results about it

One-sentence summary of the current results

Polynomial space
Turing machines

equal

Polynomial space
P systems
with active membranes[1]

[1] With charges, and as long as we have at least communication rules

Outline

A space complexity measure for P systems

Space complexity classes

PSPACE upper bound

Solving **PSPACE**-complete problems

Open problems

A space complexity measure for P systems[2]

- The size of a configuration is #membranes + #objects
- The space of a computation is the sup of the sizes of its configurations
- The space required by a P system is the sup of the space requirements of all its computations
- A family of P systems $\mathbf{\Pi}$ works in space f if each P system $\Pi_x \in \mathbf{\Pi}$ does not require more than $f(|x|)$ space

[2] A.E. Porreca, A. Leporati, G. Mauri, C. Zandron, Introducing a space complexity measure for P systems, *International Journal of Computers, Communications & Control* 4(3), 301–310, 2009

Rationale for the definition

- We want a simple but “realistic” notion of space
- Each membrane and molecule requires some amount of physical space
- The exact amount is not important, as long as it is polynomial wrt the input size (and normally it is, because of uniformity)
- So we just postulate that each molecule has unit size
- Membranes are just slightly larger than their contents, so we give them unit size in addition to that

Space complexity classes for P systems

- Languages decided by [non]confluent [semi]uniform families of P systems of type $\mathcal{D}$ (e.g., $\mathcal{AM}$) in space f

$$[\mathbf{N}]\mathbf{MCSPACE}_{\mathcal{D}}^{[\star]}(f)$$

- Languages decided by [non]confluent [semi]uniform families of P systems of type $\mathcal{D}$ in polynomial space

$$[\mathbf{N}]\mathbf{PMCSPACE}_{\mathcal{D}}^{[\star]} = \bigcup_{p \text{ poly}} [\mathbf{N}]\mathbf{MCSPACE}_{\mathcal{D}}^{[\star]}(p)$$

Simulating a nonconfluent polyspace P system[3]

A For each rule in R, assign to it a nondeterministically chosen set of membranes and objects to which the rule should be applied

B Check if the assignment of membranes and objects to rules is indeed maximally parallel; if this is not the case, abort the simulation by rejecting

C Apply the rules selected in step **A**, starting from the elementary membranes and going up towards the skin membrane

D If either *yes* or *no* were sent out from the skin membrane in step **C**, then halt and accept or reject accordingly; otherwise, jump to step **A**

[3] A.E. Porreca, A. Leporati, G. Mauri, C. Zandron, P systems with active membranes working in polynomial space, *International Journal of Foundations of Computer Science* 22(1), 65–73, 2011

Choosing the rules to be applied (step **A**)

A$_1$ Let R' be the set of currently unused rules; set $R' \leftarrow R$

A$_2$ If $R' = \emptyset$ then go to step **B**. Otherwise, pick a rule $r \in R'$

A$_3$ If $r = [a \to w]_h^\alpha$ then, for each membrane of the form $[\;]_h^\alpha$, nondeterministically choose an amount k of copies of object a to be rewritten into w; this amount can be anywhere from 0 to the multiplicity of a in that particular membrane. Subtract k from the number of available copies of a in h

A$_4$ If $r = a\,[\;]_h^\alpha \to [b]_h^\beta$ then, for each available membrane of the form $[\;]_h^\alpha$ having an available instance of a in the region immediately outside, nondeterministically choose whether to apply r and, in that case, assign those particular instances of a and h to r making them unavailable

[...]

A$_7$ Set $R' \leftarrow R' - \{r\}$ and go back to step **A**$_2$

PSPACE upper bound

Theorem

A nonconfluent P system with active membranes,
running in space S and having a description of length m
can be simulated nondeterministically using space $O(S \log m)$

Proof.

We only need to store the current configuration
and auxiliary data not larger than that □

Corollary

$\mathbf{NPMCSPACE}^{\star}_{\mathcal{AM}} \subseteq \mathbf{PSPACE}$ □

Solving Q3SAT in polynomial space[4]

- Design a uniform family of simple programs solving Q3SAT in polynomial space
- Compile the programs into a uniform family of register machines working in polynomial space
- Simulate them with a uniform family of P systems working in polynomial space (using only communication)

[4] A.E. Porreca, A. Leporati, G. Mauri, C. Zandron, P systems with active membranes: Trading time for space, *Natural Computing*, to appear (available online at `http://j.mp/PQ3SAT`)

Encoding Q3SAT instances

- There are $\binom{m}{3}$ sets of 3 variables out of m
- Each variable can be positive or negated (2^3 ways)
- Hence there are $n = 8\binom{m}{3}$ possible clauses
- We can represent a 3CNF formula by an n-bit string
- Checking well-formedness and recovering m from n are easy (polytime)

Example Boolean formula

- If we have 3 variables, the number of clauses is $8\binom{3}{3} = 8$

$$
\begin{array}{lll}
x_1 \vee x_2 \vee x_3 & x_1 \vee x_2 \vee \neg x_3 & x_1 \vee \neg x_2 \vee x_3 \\
x_1 \vee \neg x_2 \vee \neg x_3 & \neg x_1 \vee x_2 \vee x_3 & \neg x_1 \vee x_2 \vee \neg x_3 \\
\neg x_1 \vee \neg x_2 \vee x_3 & \neg x_1 \vee \neg x_2 \vee \neg x_3 &
\end{array}
$$

- Then the formula

$$\varphi = \underbrace{(x_1 \vee \neg x_2 \vee x_3)}_{\text{3rd}} \wedge \underbrace{(\neg x_1 \vee x_2 \vee \neg x_3)}_{\text{6th}} \wedge \underbrace{(\neg x_1 \vee \neg x_2 \vee x_3)}_{\text{7th}}$$

is encoded as

$$\langle \varphi \rangle = 0010\ 0110$$

Evaluating $\varphi(t_1, \ldots, t_m)$

```
if c₁ then
  v₁ := t₁ or t₂ or   t₃
else
  v₁ := 1
end;
[...]
if cₙ then
  vₙ := not tₘ₋₂ or not tₘ₋₁ or not tₘ
else
  vₙ := 1
end;
r := v₁ and v₂ ··· and vₙ
```

Evaluating $\forall x_1 \exists x_2 \cdots Q_m x_m \varphi(x_1, \ldots, x_m)$

```
r₁ := 1;
for t₁ := 0 to 1 do
  r₂ := 0;
  for t₂ := 0 to 1 do
    [...]
      rₘ := 1/0;
      for tₘ := 0 to 1 do
        r := phi(t₁, ..., tₘ);
        rₘ := rₘ and/or r
      end
    [...]
    r₂ := r₂ or r₃
  end;
  r₁ := r₁ and r₂
end;
```

Compiling programs into register machines

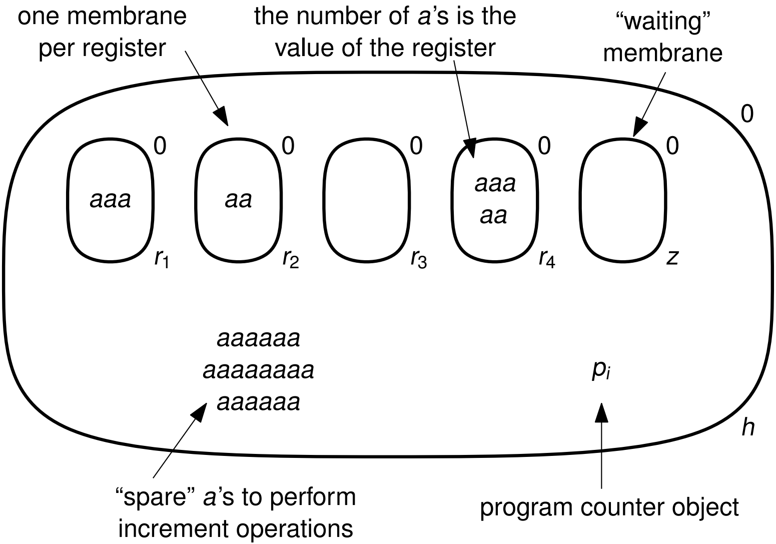
- Just an example:

```
for x := 0 to 1 do <body> end
```

- This statement is compiled as

$\mathtt{i}_1$: DEC(x), $\mathtt{i}_1$, $\mathtt{i}_2$
$\mathtt{i}_2$: <*body*>
$\mathtt{i}_3$: DEC(x), $\mathtt{i}_5$, $\mathtt{i}_4$
$\mathtt{i}_4$: INC(x), $\mathtt{i}_2$

P systems simulating register machines



Simulating the instructions

- "`i: INC(r), j`" becomes

$$p_i\,[\,]_r^0 \rightarrow [p_i']_r^+ \qquad a\,[\,]_r^+ \rightarrow [a]_r^0 \qquad [p_i']_r^0 \rightarrow [\,]_r^0\, p_j$$

- "`i: DEC(r), j, k`" becomes

$$\begin{array}{rcl} p_i\,[\,]_r^0 \rightarrow [p_i']_r^- & & [a]_r^- \rightarrow [\,]_r^0\, a \\ [p_i']_r^\alpha \rightarrow [\,]_r^\alpha\, p_i' & & p_i'\,[\,]_z^0 \rightarrow [p_i']_z^0 \\ [p_i']_z^0 \rightarrow [\,]_z^0\, p_i'' & & p_i''[\,]_r^\alpha \rightarrow [p_i'']_r^\alpha \\ [p_i'']_r^0 \rightarrow [\,]_r^0\, p_j & & [p_i'']_r^- \rightarrow [\,]_r^0\, p_k \end{array}$$

PSPACE lower bound

Theorem

If a problem can be solved in polynomial space by a uniform family of simple programs, then it can be solved in polynomial space by a uniform family of P systems with active membranes using only communication

Proof.

Just put enough copies of *a* inside the outermost membrane □

Corollary

$\mathbf{PSPACE} \subseteq \mathbf{PMCSPACE}_{\mathcal{AM}(-\mathrm{evo},-\mathrm{diss},-\mathrm{div})}$ □

Characterising polynomial space

Theorem

If $\mathcal{D} \subseteq \mathcal{AM}$ and $\mathcal{D}$ includes communication rules, then

$$[\mathbf{N}]\mathbf{PMCSPACE}_{\mathcal{D}}^{[\star]} = \mathbf{PSPACE}$$

Proof.

We proved $\mathbf{PSPACE} \subseteq \mathbf{PMCSPACE}_{\mathcal{AM}(-\text{evo},-\text{diss},-\text{div})}$
and $\mathbf{NPMCSPACE}_{\mathcal{AM}}^{\star} \subseteq \mathbf{PSPACE}$;
everything else is in between □

Open problems

- Investigate logspace families of P systems
  - Requires a weaker uniformity condition (**AC**^{0}?)[5] otherwise they trivially characterise **P**
  - Conjecture: they characterise **L**
- And also exponential space families of P systems
  - We already have an **EXPSPACE** upper bound
  - As for the lower bound, register machines don't seem to work here because we use a unary encoding to simulate them
  - No conjecture here!

[5]See N. Murphy, D. Woods, The computational power of membrane systems under tight uniformity conditions, *Natural Computing*, to appear (available online at `http://j.mp/PUniformity`)

Papers on space complexity by Milano Team I

A.E. Porreca, A. Leporati, G. Mauri, C. Zandron
Introducing a space complexity measure for P systems
International Journal of Computers, Communications & Control 4(3), 301–310, 2009

A. Valsecchi, A.E. Porreca, A. Leporati, G. Mauri, C. Zandron
An efficient simulation of polynomial-space Turing machines by P systems with active membranes
in G. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez, G. Rozenberg, A. Salomaa (eds.) *Membrane Computing, 10th International Workshop, WMC 2009*, *Lecture Notes in Computer Science* 5979, 461–478, 2010

Papers on space complexity by Milano Team II

- A.E. Porreca, A. Leporati, C. Zandron

  On a powerful class of non-universal P systems with active membranes

  in Y. Gao, H. Lu, S. Seki, S. Yu (eds.), *Developments in Language Theory, 14th International Conference, DLT 2010*, *Lecture Notes in Computer Science* 6224, 364–375, 2010

- A.E. Porreca, A. Leporati, G. Mauri, C. Zandron

  P systems with active membranes: Trading time for space

  Natural Computing, to appear (available online at `http://j.mp/PQ3SAT`)

- A.E. Porreca, A. Leporati, G. Mauri, C. Zandron

  P systems with active membranes working in polynomial space

  International Journal of Foundations of Computer Science 22(1), 65–73, 2011