Research Topics Concerning Space Complexity in P Systems

A. Alhazov, C. Zandron and Milano Team

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

16th Brainstorming Week on Membrane Computing
4-7 February 2020, Sevilla, Spain

Definition: P systems with Active Membranes

$$\Pi = (V, H, \mu, M_1, \ldots, M_n, R)$$

- V: Alphabet
- H: set of labels for membranes
- μ: Membrane structure (Ex. $[\ [\ ]_2\ [\ ]_3\ [\ [\ ]_5\ [\ ]_6\ ]_4\ ]_1$)
- M_i: String over V, initial multiset of symbols in region i
- R: Finite sets of evolution rules
- Membranes and objects can be marked using polarization: $\{+, -, 0\}$

Developmental Rules

- Assume $a \in V, w \in V^*, h \in H, \alpha_i \in \{+, -, 0\}$
- **Object evolution:** $[a \to w]_h^{\alpha_1}$
- **IN communication:** $a\,[\;]_h^{\alpha_1} \to [b]_h^{\alpha_2}$
- **OUT communication:** $[a]_h^{\alpha_1} \to [\;]_h^{\alpha_2}\, b$
- **Dissolution:** $[a]_h^{\alpha_1} \to b$
- Division for Elementary and Non–elementary membranes

Space Complexity

Space in membrane systems: definitions

- BASIC IDEA: both objects and membranes need physical space
- Let C_i be a configuration of a P system Π
- $size(C_i)$ is the sum of number of membranes in μ and the total number of objects they contain
- The space required by a halting computation $C = (C_0, C_1, \ldots, C_m)$ of Π is $size(C) = max\{size(C_0), \ldots, size(C_m)\}$
- The *space required by* Π itself is
  $size(\Pi) =$
  $= max\{size(C) : C \text{ is a halting computation of } \Pi\}$

Some basic result concerning space complexity classes

- $P \subseteq MCSPACE_{\mathcal{N}AM}(O(1))$
- $NP \cup co - NP \subseteq EXPMCSPACE_{\mathcal{E}AM}$
- $PSPACE \subseteq EXPMCSPACE_{\mathcal{A}M}$

Sublinear Space Membrane Systems

Sublinear Space Membrane Systems

- We need two distinct alphabets: *INPUT* alphabet and *WORK* alphabet
- Input objects cannot be rewritten and do not contribute to the size of a configuration
- Size of a configuration: number of membranes + total number of working objects
- Weaker uniformity condition (e.g. *DLOGTIME*-uniformity)
- RESULT: Log–space DTMs can be simulated *DLOGTIME*-uniform families of P systems in logarithmic space

Constant-space P systems

Constant space P systems

Consider P systems using constant number of working objects and constant number of membranes

Surprising result: a constant amount of space is sufficient (and trivially necessary) to solve all problems in *PSPACE*!!!
Main idea: we use the subscript and position of an input object σ_j.
Such information can be read using only a constant number of additional objects and membranes

Rethinking the definition of space

New definition of space

- Each non-input object and membrane label encodes $\Theta(\log(n))$ bits of information. E.G.
  - q_i have a subscript i ranging from 1 to n
  - Requires unitary space (definition of space)
  - The binary representation of the subscript i requires $\log p(n) = \Theta(\log n)$ bits
- We need an alternative definition to capture in a more accurate way the intuition of space

Definition

Let C be a configuration of a P system Π, Λ the set of labels of its membranes, and Γ its non-input alphabet.
The *size* $|C|$ *of* C is defined as the number of membranes in the current membrane structure multiplied by $\log |\Lambda|$, plus the total number of objects from Γ multiplied by $\log |\Gamma|$.

New definition of space

Adopting this stricter definition:

- Does not significantly change space complexity results involving polynomial or larger upper bounds
- The simulation previosly using constant-space P systems would require now logarithmic space
- The simulation previously using logarithmic-space P systems would require now $\Theta(\log n \log \log n)$

Research Topics

Representation of Sparse Data

- How many bits do we need to represent sparse data?
- Consider n indistinguishable entities *ent*: we need $bits(n) = \Theta(\log n)$ bits to represent n
- We may need more bits depending on the nature of entities, possibly requiring $bits(ent)$. In which cases?
- What about $n = 0$? $bits(0) = 0$ or we need to count the leading zeros?

Number of Configurations

- When we are only interested in number of membranes and objects, the number of their types (membrane labels and different symbols) is not really bounded apriori, but only the number of types up to their amount is relevant (due to isomorphism).
- We may be interested in $cfgs(m, t)$: $m =$ number of membranes, $t =$ number of objects
- $cfgs(m, 0)$ should be equal to the number of membrane structures
- Question: is it true that for each "reasonable" size measure *meas* for each size x there exist only a finite number $cfgs_{meas}(x)$ of possible configurations (up to isomorphism)?
- Other questions: what about different type of restrictions, like Tissue P systems or shallow P systems with active membranes?

Space complexity and Membrane Parameters 1/2

- As we saw, labels of membranes and (types of) non–input objects encode some information
- Taking or not into account the information stored there, leads to big differences: surprising result for constant space P systems
- Many questions concerning various parameters:
  - What parameters must/can we consider?
  - What if complexity is measured as $bits(cfgs(params))$?
  - What complexity classes do we characterize when we consider differend bounds on different parameters?
  - What complexity classes do we characterize when we take a combination of parameters to count the total number X of respective configurations, use $\log_2(X)$ as a space measure?

Space complexity and Membrane Parameters 2/2

- Parameters to consider on membranes: number of membranes, size of membrane label alphabet, size of polarizations alphabet
- Parameters to consider on symbols: total number of objects, bits(total number of objects), bits(number of objects in each region), bits(global multiplicity of each symbol), bits(all objects in all regions), size of the object alphabet, maximal number of objects in membranes, maximal number of global multiplicities of symbols
- Joint parameters: maximal multiplicity among symbol-membrane pairs
- Other Parameters that could be considered?

Unary vs binary representation

- General question on membrane labels, objects, parameters: unary vs binary representation.
- What space complexity classes of P systems characterize the same family of decisional problems whether the object numbers are written in unary or binary?
- If we have large multiplicities, use membrane nesting to represent these numbers in binary. When it is possible/useful?
- Simulation chain: Let's imagine S-space (where S is P or EXP) decisional P systems than can be simulated by S-space decisional Turing machines, and S-space decisional Turing machines can be simulated by S-space decisional P systems, using a constant-bounded number of objects (e.g., 1). Then s_{unary} is not more powerful than s_{binary} and, therefore, they are equivalent, proving that the true power is in membranes, not objects.

Constructibility

- Usually, when you claim some normal form for some family F of computing devices, you expect the reduction to be algorithmic.
- In other words: consider a property P, and let $P(F)$ be a subclass of F respecting P
- We could say that P does not restrict generality of F if for every system S in F there is an equivalent system S' in $P(F)$
- Constructive normal form: there esists a general algorithm $A : F \to P(F)$ s.t. $P(S)$ is equivalent to S for each S in F
- Notice: it does not work if we took the number of objects in a dedicated region (exponential), but focus is on decisional problems

Simulations

- Questions concerning complexities of different types used in in the simulations *TMsvsPsystems*: descriptional complexity, space complexity, time complexity. It could be fun to have results like: "how many membranes are 10^{12} objects worth?"
- Potential direct simulation: if we have large multiplicities, use membrane nesting to represent these numbers in binary.
  - $Bits(x)$ membranes are enough to represent multiplicity x
  - How to replace producing or consuming a copy of object a by incrementing or decrementing the number stored bitwise in this additional membrane substructure?
  - In particular, difficult with parallelism; but parallelism could be not necessary when we only care about space, not time

THANK YOU !!!