Spatial Dynamic Structures and Mobility in Computation

Bogdan Aman

Supervisor
Prof. Dr. Gabriel Ciobanu

October 2009
Institutul de Informatică Teoretică
Academia Română - Filiala Iaşi
Secția de Știința și Tehnologia Informației

Nr. 141 / 05. 10. 2009

Către D-na/Dl. .................................................................

Vă facem cunoscut că în ziua de 21.10.2009 la ora 9.00 în Sala Mică de
Conferințe a Academiei Române-Filiala Iași, va avea loc susținerea publică a tezei de
doctorat intitulată: „STRUCTURI SPĂIALE DINAMICE ȘI MOBILITATE” (în
engleză „SPATIAL DYNAMIC STRUCTURES AND MOBILITY IN COMPUTATION”),
elaborată de domnul Aman Bogdan în vederea conferirii titlului științific de doctor.

Comisia de doctorat este alcătuită din:

- Prof. dr. ing. Horia-Nicolai Teodorescu. - Președinte
  Membru Corespondent al Academiei Române
  Director la Institutul de Informatică Teoretică,
  Academia Română - Filiala Iași

- CS I dr. Gabriel Ciobanu. - Conducător științific
  Institutul de Informatică Teoretică,
  Academia Română - Filiala Iași

- CS I dr. Gheorghe Păun. - Referent
  Membru Corespondent al Academiei Române
  Institutul de Matematică “Simion Stoilow” al Academiei Române.

- Prof. dr. Dorel Lucanu. - Referent
  Universitatea „Al. I. Cuza” Iași, Facultatea de Informatică

- Prof. dr. Mitică Craus. - Referent
  Universitatea Tehnică “Gheorghe Asachi” din Iași,
  Facultatea de Automatică și Calculatoare.

Cu această ocazie vă transmitem rezumatul tezei de doctorat cu rugământa de a ne
comunica în scris aprecierile și observațiile dumneavoastră.

De asemenea vă invităm să participați la susținerea publică a tezei de doctorat.

Director,

Prof. dr. ing. Horia-Nicolai Teodorescu
Membru Corespondent al Academiei Române

Coordonator Programe Doctorale,

CS II dr. Florin Rotaru
Abstract

Membrane computing is a well-established and successful research field which belongs to the more general area of molecular computing. Membrane computing aims at defining parallel and non-deterministic computing models, called membrane systems or P Systems, which abstract from the functioning and structure of the cell. A membrane system consists of a spatial structure, a hierarchy of membranes which do not intersect, with a distinguishable membrane called skin surrounding all of them. A membrane without any other membranes inside is elementary, while a non-elementary membrane is a composite membrane. The membranes define demarcations between regions; for each membrane there is a unique associated region. Since we have a one-to-one correspondence, we sometimes use membrane instead of region, and vice-versa. The space outside the skin membrane is called the environment.

In this thesis we define and investigate variants of systems of mobile membranes as models for molecular computing and as modelling paradigms for biological systems. On one hand, we follow the standard approach of research in membrane computing: defining a notion of computation for systems of mobile membranes, and investigating the computational power of such computing devices. Specifically, we address issues concerning the power of operations for modifying the membrane structure of a system of mobile membranes by mobility: endocytosis (moving a membrane inside a neighbouring membrane) and endocytosis (moving a membrane outside the membrane where it is placed). On the other hand, we relate systems of mobile membranes to process algebra (mobile ambients, timed mobile ambients, \(\pi\)-calculus, brane calculus) by providing some encodings and adding some concepts inspired from process algebra in the framework of mobile membrane computing.
Acknowledgements

I thanks to my supervisor Dr. Gabriel Ciobanu for providing this opportunity, and for his constant support and advice throughout this PhD.

I thanks the anonymous referees for their comments and suggestions which helped improve the quality of the papers in which parts of this thesis were published.

I would also like to thanks my examiners: Dr. Gheorghe Păun, Dr. Dorel Lucanu, Mitică Craus, Matteo Cavaliere, for their careful reading of the thesis and their helpful comments.

I am grateful to all my colleagues from the group of Formal Methods Laboratory, Institute of Computer Science, Romania Academy, Iaşi Branch for their suggestions and encouragements.
# Contents

1 Introduction ........................................................................... 2  
  1.1 Context .......................................................................... 2  
  1.2 Motivation ...................................................................... 3  
  1.3 Outline ........................................................................ 4  
2 Preliminaries ................................................................. 6  
  2.1 Alphabets, strings and languages ..................................... 6  
  2.2 ChomskyGrammars ....................................................... 7  
  2.3 L Systems ..................................................................... 9  
  2.4 Matrix Grammars ....................................................... 10  
  2.5 Register Machines ..................................................... 12  
  2.6 Process Algebra ........................................................ 12  
      2.6.1 \(\pi\)-calculus ................................................... 12  
      2.6.2 Safe MobileAmbients ......................................... 13  
      2.6.3 BraneCalculi ................................................... 16  
  2.7 P Systems .................................................................... 17  
      2.7.1 Transition P Systems ........................................ 17  
      2.7.2 P Systems with Active Membranes ..................... 18  
3 Mobile Membranes .......................................................... 24  
  3.1 SimpleMobileMembranes ............................................. 25  
      3.1.1 Definition ...................................................... 25  
      3.1.2 ComputationalPower ....................................... 26  
  3.2 EnhancedMobileMembranes ........................................... 28  
      3.2.1 Motivation .................................................... 28  
      3.2.2 Definition ..................................................... 31  
      3.2.3 Modelling power ............................................. 32  
      3.2.4 ComputationalPower ....................................... 35  
  3.3 MutualMobileMembranes ............................................. 37  
      3.3.1 Motivation .................................................... 37  
      3.3.2 Definition ..................................................... 39  
      3.3.3 ComputationalPower ....................................... 39  
  3.4 Conclusions ................................................................... 45
This chapter contains some basic notions of formal language theory, as well as automata. For further information about these topics the reader is referred to the monographs [9, 12, 13, 24] and to specific papers cited in the next sections.
Mobile Membranes

In this chapter we define the systems of simple, enhanced and mutual mobile membranes and study their modelling and computational power.

The systems of simple mobile membranes are a variant of P systems with active membranes having none of the features like polarizations, label change, division of non-elementary membranes, priorities, or cooperative rules. The additional feature considered instead are the operations of endocytosis and exocytosis: moving a membrane inside a neighbouring membrane, or outside the membrane where it is placed. However, these operations are slightly different in the papers introducing them: in [8] one object is specified in each membrane involved in the operation, while in [18] one object is mentioned only in the moving membrane. Another variant of P systems with mobile membranes is mobile P systems [23] having rules inspired from mobile ambients [4]. Turing completeness is obtained by using nine membranes together with the operations of endocytosis and exocytosis [18]. Using also some contextual evolution rules (together with endocytosis and exocytosis), in [14] it is proven that four mobile membranes are enough to get the power of a Turing machine, while in [BA5] we decrease the number of membranes to three. In order to simplify the presentation, we use systems of simple mobile membranes instead of P systems with mobile membranes.

The systems of enhanced mobile membranes are a variant of membrane systems which we proposed in [BA10] for describing some biological mechanisms of the immune system. The operations governing the mobility of the systems of enhanced mobile membranes are endocytosis (endo), exocytosis (exo), forced endocytosis (fendo) and forced exocytosis (fexo). The computational power of the systems of enhanced mobile membranes using these four operations was studied in [17] where it is proved that twelve membranes can provide the computational universality, while in [BA5] we improved the result by reducing the number of membranes to nine. It is worth to note that unlike the previous results, the rewriting of object by means of context-free rules is not used in any of the results (proofs).

Following our approach from [BA14] we define systems of mutual mobile membranes representing a variant of systems of simple mobile membranes in which the endocytosis and the exocytosis work whenever the in-
volved membranes “agree” on the movement; this agreement is described by using dual objects $a$ and $\overline{a}$ in the involved membranes. The operations governing the mobility of the systems of mutual mobile membranes are mutual endocytosis (mutual endo), and mutual exocytosis (mutual exo). It is enough to consider the biologically inspired operations of mutual endocytosis and mutual exocytosis and three membranes (compartments) to get the full computational power of a Turing machine [BA8]. Three represents the minimum number of membranes in order to discuss properly about the movement provided by endocytosis and exocytosis: we work with two membranes inside a skin membrane.
Mobile Membranes with Objects on Surface

Membrane systems [21, 22] and brane calculus [3] have been inspired from the structure and the functioning of the living cell. Although these models start from the same observation, they are build having in mind different goals: membrane systems investigate formally the computational nature and power of various features of membranes, while the brane calculus is capable to give a faithful and intuitive representation of the biological reality. In [5] the initiators of these two formalisms describe the goals they had in mind: “While membrane computing is a branch of natural computing which tries to abstract computing models, in the Turing sense, from the structure and the functioning of the cell, making use especially of automata, language, and complexity theoretic tools, brane calculi pay more attention to the fidelity to the biological reality, have as a primary target systems biology, and use especially the framework of process algebra.”

In [BA18] we define a new class of systems of mobile membranes, namely the systems of mutual membranes with objects on surface. The inspiration to add objects on membrane and to use the biologically inspired rules pino/exo/phago comes from [2, 5, 6, 15, 16]. The novelty comes from the fact that we use objects and co-objects in phago and exo rules in order to illustrate the fact that both involved membranes agree on the movement. We investigate in [BA6] the computational power of systems of mutual membranes with objects on surface controlled by pairs of rules: pino/exo or phago/exo, proving that they are universal with a small number of membranes. Similar rules are used by another formalism called brane calculus [3]. We compare in [BA18] the systems of mutual membranes with objects on surface with brane calculus, and encode a fragment of brane calculus into the newly defined class of systems of mobile membranes. Even brane calculus have an interleaving semantic and membrane systems have a parallel one, by performing this translation we show that the difference between the two models is not significant.
Membrane Systems and Process Algebra

The membrane systems [21, 22] and the mobile ambients [4] have similar structures and common concepts. Both have a hierarchical structure representing locations, and are used to model various aspects of biological systems. The mobile ambients are suitable to represent the movement of ambients through ambients and the communication which takes place inside the boundaries of ambients. Membrane systems are suitable to represent the movement of objects and membranes through membranes. We consider these new computing models used in describing various biological phenomena [3, 7], and encode the ambients into membrane systems [BA11]. We present such an encoding, and use it to describe the sodium-potassium exchange pump [BA4]. We provide an operational correspondence between the safe ambients and their encodings, as well as various related properties of the membrane systems [BA4].

In [BA3] we investigate the problem of reaching a configuration from another configuration in a special class of systems of mobile membranes. We prove that the reachability can be decided by reducing it to the reachability problem of a version of pure and public ambient calculus without the capability open.

A feature of current membrane systems is the fact that objects and membranes are persistent. However, this is not quite true in the real world. In fact, cells and intracellular proteins have a well-defined lifetime. Inspired from these biological facts, we define in [BA16] a model of mobile membranes in which each membrane and each object has a timer attached representing their lifetime. This new feature is inspired from biology where cells and intracellular proteins have a well-defined lifetime. In order to simulate the passage of time we use rules of the form $a^\Delta t \rightarrow a^{\Delta(t-1)}$ and $[ ]^\Delta t_i \rightarrow [ ]^{\Delta(t-1)}_i$ for the objects and membranes which are not involved in other rules. If the timer of an objects reaches 0 then this object is consumed by applying a rule of the form $a^{\Delta 0} \rightarrow \lambda$, while if the timer of a membrane $i$ reaches 0 then the membrane is dissolved by applying a rule of the form $[ ]^{\Delta 0}_i \rightarrow [\delta]^{\Delta 0}_i$. After dissolving a membrane, all objects and mem-
branes previously present in it become elements of the immediately upper membrane, while the rules of the dissolved membrane are removed. Some results show that mutual mobile membranes with and without timers have the same computational power. Since we have defined an extension with time for mobile ambients in [BA1, BA2, BA7], and one for mobile membrane in [BA16], we study the relationship between these two extensions: timed safe mobile ambients are encoded into mutual mobile membranes with timers.

Membrane systems [21, 22] are known to be Turing complete and are often used to model biological systems and their evolution. In order to increase the modelling power of this formalism, we define in [BA9] a typed version, which leads in turn to a decrease of computational power. We enrich the symport/antiport membrane systems with a *type discipline* which allow to guarantee the soundness of reduction rules with respect to some relevant properties of the biological systems. The key technical tools we use are type inference and principal typing [25], i.e. we associate to each reduction rule the minimal set of conditions that must be satisfied in order to assure that applying this rule to a correct membrane system, then we get a correct membrane system as well. The type system for membrane systems with symport/antiport rules is (up to our knowledge) the first attempt to control the evolution of membrane systems using typing rules. The presentation of the typed sodium-potassium pump is an example how to introduce and use types in membrane systems. The $\pi$-calculus typed pump is presented in order to see what is the desired modelling power we want to have in membrane systems by introducing a type system. Type descriptions of biological inspired formalisms, along with the type inference algorithm can also be found in [BA12, 10]. Other static techniques have been applied to biological systems, such as Control Flow Analysis [1, 19, 20] and Abstract Interpretation [11].
Conclusions

Contributions

Membrane computing is a well-established and successful research field which belongs to the more general area of molecular computing. Membrane computing aims at defining parallel and non-deterministic computing models, called membrane systems or P Systems, which abstract from the functioning and structure of the cell. Since the introduction of this model, many variants have been proposed and the literature on the subject is now rapidly growing. There are two standard ways of investigating membrane systems: considering their computational power in comparison with the classical notion of Turing computability, or considering their efficiency in solving algorithmically hard problems, like NP-problems, in a polynomial time.

In this respect, we defined new classes of membrane systems which are powerful, mostly equivalent to Turing machines, and for which we established links with process algebra.

The systems of simple mobile membranes are a variant of P systems with active membranes having none of the features like polarizations, label change, division of non-elementary membranes, priorities, or cooperative rules. The additional feature considered instead are the operations of endocytosis and exocytosis: moving a membrane inside a neighbouring membrane, or outside the membrane where it is placed. In [BA5] we proved that three mobile membranes are enough to get the power of a Turing machine.

The systems of enhanced mobile membranes are a variant of systems of simple mobile membranes that we proposed in [BA10] for describing some biological mechanisms of the immune system. The operations governing the mobility of the systems of enhanced mobile membranes are endocytosis (endo), exocytosis (exo), enhanced endocytosis (fendo) and enhanced exocytosis (fexo). In [BA5] we studied the computational power of the systems of nine enhanced mobile membranes using these four operations.

Following our approach from [BA14] we defined systems of mutual mobile membranes representing a variant of systems of simple mobile mem-
branes in which the endocytosis and exocytosis work whenever the involved membranes “agree” on the movement; this agreement is described by using dual objects $a$ and $\overline{a}$ in the involved membranes. The operations governing the mobility of the systems of mutual mobile membranes are mutual endocytosis (mutual endo), and mutual exocytosis (mutual exo). It is enough to consider the biologically inspired operations of mutual endocytosis and mutual exocytosis and three membranes (compartments) to get the full computational power of a Turing machine [BA8].

In [BA18] we defined a new class of systems of mobile membranes, namely the systems of mutual membranes with objects on surface. The rules of this class are biologically inspired, namely pino/exo/phago rules. The novelty comes from the fact that we used objects and co-objects in phago and exo rules in order to illustrate the fact that both involved membranes agree on the movement. We investigated in [BA6] the computational power for systems of mutual membranes with objects on surface controlled by pairs of rules: pino/exo or phago/exo, proving that they are universal with a small number of membranes. Similar rules are used by another formalism called brane calculus [3]. In [BA18] we defined an operational semantic for systems of membranes with objects on surface and compare the systems of mutual membranes with objects on surface with brane calculus, and encoded a fragment of brane calculus into the newly defined class of systems of mobile membranes.

In [BA11] we encoded the mobile ambients into the membrane systems and provided an operational semantic for membrane systems. We presented such an encoding, and used it to describe the sodium-potassium exchange pump [BA4]. We provided an operational correspondence between the safe ambients and their encodings, as well as various related properties of the membrane systems [BA4].

In [BA3] we investigated the problem of reaching a configuration from another configuration for a subclass of systems of mobile membranes, and proved that the reachability can be decided by reducing it to the reachability problem of a version of pure and public ambient calculus without the capability open.

In [BA15, BA16] new classes of membranes are defined: timers are assigned to each membrane and each object. This new feature is inspired from biology where cells and intracellular proteins have a well-defined lifetime. In order to simulate the passage of time we use rules of the form $a^{\Delta t} \rightarrow a^{\Delta(t-1)}$ and $[i]^{\Delta t} \rightarrow [i]^{\Delta(t-1)}$ for the objects and membranes which are not involved in other rules. If the timer of an objects reaches 0 then this object is consumed by applying a rule of the form $a^{\Delta 0} \rightarrow \lambda$, while if the timer of a membrane $i$ reaches 0 then the membrane is dissolved by applying a rule of the form $[i]^{\Delta 0} \rightarrow [\delta]^{\Delta 0}$. After dissolving a membrane,
all objects and membranes previously present in it become elements of the membrane containing it, while the rules of the dissolved membrane are removed.

By adding timers to objects and membranes into a system of mutual mobile membranes, we do not obtain a more powerful formalism. According to [BA16] we have that systems of mutual mobile membranes with timers and systems of mutual mobile membranes without timers have the same power. Since an extension with time for mobile ambients already exists [BA1, BA2, BA7], and one for systems of mobile membranes is presented in [BA16], the relationship between these two extensions is studied: timed safe mobile ambients are encoded into systems of mutual mobile membranes with timers [BA16].

In [BA9] we enriched the symport/antiport membrane systems with a type discipline which allows to guarantee the soundness of reduction rules with respect to some relevant properties of the biological systems. The key technical tools we used are type inference and principal typing, i.e. we associate to each reduction rule the minimal set of conditions which must be satisfied in order to assure that applying this rule to a correct membrane system, then we get a correct membrane system as well. The type system for membrane systems with symport/antiport rules is (up to our knowledge) the first attempt to control the evolution of membrane systems using typing rules. The presentation of the typed sodium-potassium pump is an example how to introduce and use types in membrane systems. The $\pi$-calculus typed pump is presented in order to see what is the desired modelling power we want to have in membrane systems by introducing a type system.

### Other Contributions

We also focused on other formalisms which are characterized by spatial dynamic structures and mobility, namely mobile ambients and the calculus of looping sequences. In what follows we present the work done in this direction.

In [BA1] we extended mobile ambients with timers and proximities, in order to get a clear notion of location and mobility. Timers define timeouts for various resources, making them available only for a determined period of time; we add timers to ambients and capabilities. We presented an example how the new model is working. The coordination of the ambients in time and space is given by assigning specific values to timers, and by a set of coordination rules.

In [BA2] we added timers to communication channels, capabilities and ambients, and used a typing system for communication. The passage of
time is given by a discrete global time progress function. We proved that structural congruence and passage of time do not interfere with the typing system. Moreover, once well-typed, an ambient remains well-typed. A timed extension of the cab protocol illustrates how the new formalism is working.

In [BA7] we added timers to capabilities and ambients, and provided an operational semantics of the new calculus. Certain results are related to the passage of time, and some new behavioural equivalences over timed mobile ambients are defined. Timeout for network communication (TTL) can be naturally modelled by the time constraints over capabilities and ambients. The new formalism can be used to describe network protocols; Simple Network Management Protocol (SNMP) may implement its own strategy for timeout and retransmission in TCP/IP.

In [BA12] we enriched the calculus of looping sequences, a formalism for describing evolution of biological systems by means of term rewriting rules, with type disciplines to guarantee the soundness of reduction rules with respect to interesting biological properties.
Bibliography


My Publications

ISI Publications


Other Publications


[BA11] B. Aman, G. Ciobanu. Translating Mobile Ambients into P Sys-


