Preface

The design of formal calculi in which the fundamental concepts underlying interactive systems can be described and studied has been a central theme of theoretical computer science over the last two decades. In this book we refer to the formal description of *mobility* in computer science by using π-calculus, ambient calculus, bioambients, brane calculi and systems of mobile membranes.

In process algebra the moving entities are the links (π-calculus), the ambients (ambient calculus and bio-ambients) and the branes (brane calculi). In membrane systems the movement is provided by rules inspired by endocytosis and exocytosis. Cell movement is a dynamic phenomenon that is essential to a variety of biological processes (e.g., immune response).

In **Chapter 1: Mobility in Process Calculi** we refer to the formal description of *mobility* in process calculi [52]. When expressing mobility, we should mention what entities move and in what space they move. The *π-calculus* [121] is a formalism where links are the moving entities, and they move in a virtual space of linked processes (the network of web pages is a good example for this approach). This option is powerful enough to express moving processes both in a physical space of computing locations and in a virtual space of linked processes [121]. The π-calculus has a simple semantics and a tractable algebraic theory [121]; it is a widely accepted model of interacting systems with dynamically evolving communication topology and (channel) mobility. Its mobility increases the expressive power enabling the description of many high-level concurrent features.

Timed distributed π-calculus ($tD\pi$) [67] is a rigorous framework for describing distributed systems with time constraints. The timers on channels define timeouts for communications, and timers on the channel types restrict the channels' availability. Whenever the timer of either a channel or a channel type expires, the corresponding channel is discarded, and respectively the channel type is lost. $tD\pi$ combines temporal constraints with types and locations in order to give the possibility of modelling *located* and *timed* interactions between distributed processes with *time-restricted resource access*.

Another formalism able to express mobility is the *ambient calculus* [42]; it describes computation carried out on mobile devices (i.e. networks having a dynamic topology), and mobile computation (i.e. executable code able to move around the network). The primitive concept of the calculus is the ambient defined as a bounded place in which computation can occur. Ambients can be nested inside other ambients. Each ambient has a name used to control access to it. Computation is represented as the movement of ambients: they can be moved as a whole, changing their location by consuming certain capabilities: in, out, open.

Mobile ambients with timers (tMA) [9, 10, 15] represent a conservative extension of the ambient calculus. Inspired by [41], we introduce types for ambients in tMA. The type system associates to each ambient a set of types in order to control its communication by allowing only well-typed messages. For instance, if a process inside an ambient sends a message of a type which is not included in the type system of the ambient, then the process fails. In tMA the process may continue its execution after the timer of the corresponding output communication expires.

The biological inspiration is predominant in the case of *brane calculi* [40]. The operations of the two basic brane calculi, namely *pino, exo, phago* (for the PEP fragment) and *mate, bud, drip* (for the MBD fragment) are directly inspired by the biologic processes of *endocytosis, exocytosis* and *mitosis*. Since some proteins are embedded in cell membranes, and can act on both sides of the membrane simultaneously, brane calculi use both sides of the membrane, emphasizing that computation happens also on the membrane surface.

On the other hand, in **Chapter 2: Mobility in Membrane Computing**, we study mobility in the framework of natural computing. Natural computing refers to both computational models inspired by nature and biological processes. When complex natural phenomena are analyzed in terms of computational processes, our understanding of both nature and computation is enhanced. Natural computing is looking for concepts, principles and mechanisms underlying natural systems.

Membrane computing is part of natural computing, being a rule-based formalism inspired by biological cells [128]. Mobile membranes represent a formalism describing the movement of membranes inside a spatial structure by applying specific rules from a given set. We define several systems of mobile membranes: simple, enhanced and mutual mobile membranes, as well as mutual mobile membranes with objects on surface. When membrane systems are considered as computing devices, two main research directions are considered: the computational power in comparison with the classical notion of Turing computability, and the efficiency in algorithmically solving hard problems (e.g., NP-problems) in polynomial time. In this chapter we present mobile systems which are both powerful (mostly equivalent to Turing machines) and efficient (membrane system algorithms provide efficient solutions to NP-complete problems through the generation of an exponential space in polynomial time).

Reachability is the problem of deciding whether a system may reach a given configuration during its execution. This is one of the most critical properties in the verification of systems; most of the safety properties of computing systems can be reduced to the problem of checking whether a system may reach an "unintended state". We investigate the problem of reaching a certain configuration in systems of mobile membranes with replication rules, starting from a given configuration. We prove that reachability in systems of mobile membranes can be decided by reducing it to the reachability problem of a version of pure and public ambient calculus from which the open capability has been removed.

In **Chapter 3: Encodings** we establish several links between process calculi and membrane computing in order to be able to use techniques from one area in the other one. The difference between these two research areas is the fact that process algebras provide a tool for the high-level description of interactions, communications, and synchronizations between a collection of independent agents or processes, providing also algebraic laws that allow process descriptions to be manipulated and analyzed, and permit formal reasoning about equivalences between processes (e.g., using bisimulation), while membrane computing uses techniques from languages, automata, complexity, and dynamical systems. We consider our encodings as the first efforts towards bridging the gap between process calculi and mobile membranes.

In order to study the expressive power of $tD\pi$ we use a method of *embeddings among languages* introduced in [148]. The method is based on a tuple composed of a set of process expressions $\mathscr{P}$, a partial operation over $\mathscr{P}$ (in process calculi we choose the parallel composition operator) and an observational equivalence. To compare two formalisms by looking at their sets of syntactic expressions (languages) L_1 and L_2, we are required to identify the corresponding *algebraic languages* $(\mathscr{P}; |; \simeq)$ respectively $(\mathscr{P}'; |'; \simeq')$. We adapt this method and use it to show that $tD\pi$ is more expressive than the underlining $D\pi$.

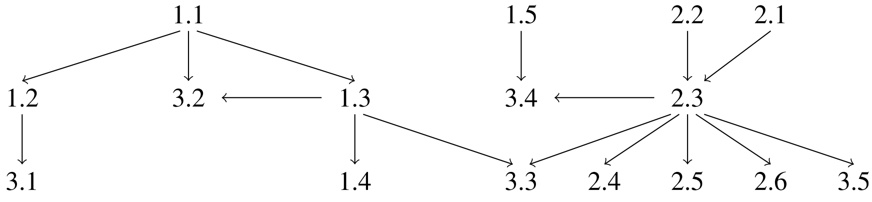
Although both the π-calculus and the calculus of mobile ambients are Turing-complete [42, 121] and they have almost the same field of application (mobile computations), it is widely believed (see [77]) that the π-calculus does not directly model phenomena such as the distribution of processes within different localities, their migrations, or their failures. We present a translation of mobile ambients into the asynchronous π-calculus: in order to imitate the spatial structure of mobile ambients we impose some very rigid restrictions on the structural congruence rules of the π-calculus. A key idea of the encoding is based on the separation of the spatial structure of mobile ambients from their operational semantics.

Membrane systems [127, 128] and mobile ambients [42] have similar structures and common concepts. Both have a hierarchical structure representing locations, and are used to model various aspects of biological systems. 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 [40, 65], and encode the ambients into membrane systems [12, 19]. We present such an encoding, and provide an operational correspondence between the safe ambients and their encodings, as well as various related properties of the membrane systems [14].

Some work has been done trying to relate membrane systems and brane calculi [35, 37, 47, 105, 106]. Inspired by brane calculi, a model of membrane systems having objects attached to membranes has been introduced in [45]. In [31], a class of membrane systems containing both free floating objects and objects attached to membranes has been proposed. We are continuing this research line, and simulate a fragment of brane calculi by using systems of mutual membranes with objects on surface. By defining an encoding of the PEP fragment of brane calculi into systems of mutual membranes with objects on surface, we show that the difference between the two models is not significant.

A relation can be established between mobile membranes and coloured Petri nets by providing an encoding of the first formalism into the second one. By considering the endocytic pathway for low-density lipoprotein degradation, we show how mobile membranes can be used to model such a biological phenomenon, while coloured Petri nets can be used to analyze and verify automatically some behavioural properties of the pathway. Some connections between membrane systems and Petri nets are presented for the first time in [78] and [137]. In [101, 102], a direct structural relationship between these two formalisms is established by defining a new class of Petri nets called Petri nets with localities. This new class of Petri nets has been used to show how maximal evolutions from membrane systems are faithfully reflected in the maximally concurrent step sequence semantics of their corresponding Petri nets with localities.

The book is devoted to researchers. However, since it contains examples and exercises, it can be used as a course support. The dependencies of its chapters and sections are represented by the following graph



The book is designed primarily for computer scientists working in concurrency (process calculi, Petri nets), in biologically inspired formalisms (brane calculi, membrane systems), and also for the mathematically inclined scientists interested in formalizing moving agents and biological phenomena. As far as we know, the book is the first monograph that treats mobility as its central topic.

Iaşi, 2011

Bogdan Aman
Gabriel Ciobanu

Mobility in Process Calculi and Natural Computing
Aman, B.; Ciobanu, G.
2011, XIV, 210 p., Hardcover
ISBN: 978-3-642-24866-5