**PhD Dissertation**


**International Doctorate School in Information and Communication Technologies**

DIT - University of Trento

# Logical Analysis of Complex Systems

## - Dynamic Epistemic Spatial Logics -

Radu Mardare

Advisor:Prof. Corrado Priami

Università degli Studi di Trento

March 2006

# Abstract

*We propose a new class of spatial logics for concurrency, Dynamic Epistemic Spatial Logics, to be used for specifying properties of concurrent and distributed systems. These logics are developed as extensions of Hennessy-Milner logic, with spatial and epistemic operators. We propose the use of epistemic operators to express contextual situations and to treat processes/programs as epistemic agents. Knowledge is thus defined as information (about the overall, global processes) that is locally available (to an agent/suprocess).*

*The novelty with respect to spatial logics comes from proposing a way to express contextual properties of dynamic systems within the limits of decidability. For our logics, satisfiability, validity and model-checking problems have been proved to be decidable in combination with temporal and dynamic operators.*

*This idea is also new for epistemic logics, as we have an algebraical structure over the class of epistemic agents that allows composed agents to be considered. Thus the processes $P$ and $Q$ are agents with their own knowledge and dynamics, but also $P|Q$, the process running $P$ and $Q$ in parallel, has its own knowledge and dynamics. Moreover, the knowledge of the composed agent $P|Q$ may be computed from the knowledge of its subsidiary agents $P$ and $Q$. This feature opens up the possibility to analyze dynamics of knowledge in multi-agent systems where individuals, societies of individuals, societies of societies of individuals, etc, are considered as agents conjointly.*

*For our logics, we develop Hilbert-style axiomatic systems identifying the basic axioms and rules that defines the behavior of classical concurrent operators. The systems have been proved to be sound and complete with respect to process-algebraical semantics.*

*Combining epistemic operators, dynamic operators and spatial operators,*

*we obtain logics expressive enough to be decidable and completely axiomatizable, thus suitable for analyzing properties of complex concurrent and distributed systems.*

**Keywords**
Process Algebra, Spatial Logics, Dynamic Logics, Epistemic Logics

*To my wife Diana*

## Acknowledgement

First I would like to thank my advisor, Corrado Priami, for his contribution to the development of my work during these years. He believed in me and constantly encouraged me in going further with my (not always orthodox) ideas. I would also like to thank him for his extraordinary generosity and for his full support and help over the years. Working together was always a pleasure.

I would like to thank Alexandru Baltag for all his insightful help that had an important role in the growing and the unfolding of the ideas presented here. His philosophical views on the foundations of mathematics and his work on non-wellfounded sets have inspired me since my college years and directed my research from Mathematics to Philosophy and, in the end, to Computer Science. I am grateful to Alexandru also for giving me the chance to visit Comlab Oxford where most of the ideas presented further have been born, as a result of the stimulating discussions we had.

I want to express my thanks to Luca Cardelli for the interesting discussions we had during these years and for his comments and feedback on the present work. His masterful knowledge of theoretical computer science played an essential role in my training in this field. His work on logics for concurrency was my main inspiration source that made my logics possible.

I want to express my thanks to Solomon Marcus for collaborating with me in the past. This collaboration was the moral and spiritual support I needed in some of the toughest moments of my life. In that time, due to him, I rediscovered Mathematics, a lost love that, eventually, saved my soul.

Last but not list I want to thank to my wife Diana for being closed to me, for giving me all her love, for supporting and taking care of me all these years. Without her this work wouldn't be possible.

# Contents

# Chapter 1

# Introduction

"*The Fundamental Postulates of General Systemantics are:*
*1. Everything is a system*
*2. Everything is part of a larger system*
*3. The Universe is infinitely systematizable, both*
*upward (larger systems) and downward (smaller systems)*
*4. All systems are infinitely complex. (The illusion of*
*simplicity comes from focusing attention on ... a few variables)*"

John Gall, *Systemantics*, 1975

The notions of *calculational process* or *algorithm* are not new in mathematics. They were studied long before the development of computing technology. Still, the invention of modern computers poses computing theorists with new problems and puts the old ideas in the light of new paradigms.

The notion of *concurrency* has a history deeply related with the evolution of technology in Computer Science. It arises from the necessity to build mathematical models for depicting the behavior of computational systems. In the beginning, such a system was just a computer able to run a few programs successively - *a stand-alone computer system.* Many

mathematical models of these systems have been built and adapted to their purposes, the main goal being to explain the way computational systems behave for external observers. Among these models, $\lambda$-calculus proved to be an appropriate and useful one.

The development of computer networks came with new challenges and new paradigms of computation. The concept of *monolithic computational systems* (one-agent system) was replaced by the *concurrent distributed computing systems* (multi-agent systems), which are not only sequential, goal-directed, deterministic or hierarchical systems, but represent programs/processors running in parallel and organized in networks of subsystems, each subsystem having its own identity. They interact, collaborate, communicate and interrupt each other.

Underlying this new paradigm is the assumption that each part of such a system has its own identity, which persists through time. We shall call these parts *agents*. We need the notion of agents in order to discriminate between the events of the system's behavior. Indeed, if we wish to identify a particular event we have little choice but to identify the agents involved. Hence the agents are separate and independently observable units of behavior and computation. They evolve in a given environment, following some primitive rules, their evolution influencing the structure of the whole system. The main feature of the agents is their ability to communicate, that is to exchange information inside their environment.

These agents are not topologically bound in the network, being able to change their relative positions. A laptop, for example, can be connected to the computer network at some point, it can start running some programs that interact with the network and, further, it might be unplugged and plugged back at a different point. Meanwhile, the laptop is running its programs independently of the whole system. Such a multi-agent system is, by nature, also spatially distributed. Hence, we discuss interactive,

concurrent and distributed behaviors and computations of agents.

The agents themselves might be hierarchically organized. If we take a concrete system, let's say a *company*, then we may consider it as a network of *departments*, each department, maybe, as a network of *sub-departments*, etc, until, in the end, we reach the point where we treat the *sub-...departments* as networks of people as the final level of analysis.

On the other hand, if we study how a virus that produces a disease acts at the level of *population*, then the network of agents will be defined in a different way. As in the company example, we consider smaller *societies* - people living in the same town. Such an abstraction might be relevant for analyzing the dynamics of the virus and imagining some prophylactic strategies. Then we may go further to the level of *families* and furthermore to people. But, in this case, we may want to go even deeper, and to consider the person itself as a system (of organs etc.) which might help in understanding more about the disease.

Hence, the level to which we proceed in decomposition depends upon our present interest and not upon the entities involved. Therefore, in a theory that we wish to apply at different levels of abstraction, we should make no distinction of kind between systems and components, nor between systems and systems without substructure. In this light, broadly, an agent is any system whose behavior consists of discrete actions. An agent which for one purpose we take to be atomic may, for other purposes, be decomposed into subagents acting concurrently and interacting. Hence agents are, by nature, compositionally and hierarchically organized. Agent $A$ might have a behavior described, on some level, by the programs/processes $P_1$ and $P_2$ running in parallel (hereafter we denote this by $P_1|P_2$) while program $P_1$ might describe, on another level, the behavior of agent $B$, and $P_2$ of agent $C$. Hence $B$ and $C$ *"working together"* can simulate/compose $A$, as they represent two complementary subsystems of $A$.

Multi-agent systems are extremely complex. In dealing with this complexity we have only to focus on a few details that, hopefully, cover everything relevant to our analysis. Hence, the success of our approach depends on the mathematical model we choose to abstract the system. We focus on two major approaches.

One is the approach inspired by $\lambda$-calculus - Process Algebra initiated by Tony Hoare [45] and Robin Milner [57]. It abstracts the agents of the system, on the level of their behavior, by using some algebraic calculi. There are algebraic operators to describe the atomic sequential evolutions of the agents ($\alpha.P$ describes the behavior of the agent that can perform an atomic action $\alpha$ and then behaves as $P$); there is an operator that describes parallel behaviors ($P|Q$ describes the behavior of the agent composed of two subsystems that act in parallel, one with the behavior $P$ and the other with the behavior $Q$); we denote by 0 the behavior of a static agent that does nothing. In addition, other operators can be added to express peculiarities of some systems, such as operators for nondeterministic choice, fresh resources, input/output on channels, named locations, movements between named locations, etc.

The other approach comes from philosophy and logics and is based on the seminal work of Hintikka [43]: reasoning about systems in terms of knowledge. This approach was successful in modelling systems with human agents (in economy, linguistics, etc.) but then it was extended to (artificial) intelligent agents and later to multi-agent systems in the most general sense. This adaptation came with the price of reconsidering the meaning of *knowledge*. In the new interpretation *the knowledge of the agent* is the sum of actions the agent may take as a function of its local state in a given environment. In this context we have an *external* notion of knowledge in

the sense that there is no notion of the agent computing his knowledge and no requirement that the agent be able to answer questions based on his knowledge.

The two approaches were developed in parallel, but to our knowledge, until now there has been no unified paradigm. We propose such a paradigm in this thesis.

## 1.1 *To be* is *to behave*

*"Esse est percipi (To be is to be perceived)."*
George Berkeley

Process algebra aims at underpinning the theories of concurrent distributed systems, as $\lambda$-calculus does for monolithic systems. Process calculi propose some abstract algebraical formalisms to model the behavior of such systems.

Starting with CCS [57, 60] and CSP [45, 11], up to $\pi$-calculus [68, 62] and lately to calculi with locations - Ambient Calculus [21, 22, 20], Distributed Mobile Processes [67], BioAmbients [66], Brane Calculi [18], Beta Binders [64, 65], a large class of process calculi have been developed and used in various applications, from security [1, 17, 50] to systems biology [66, 18, 64, 65, 51, 53].

Two complementary levels of description of a system are needed in order to provide an accurate model for it: the *intensional level*, which concerns the internal structure and mechanics of the system, and the *extensional level*, which concerns the effect produced by it over a given environment. Process algebras describe systems on both levels. On the intensional level they provide the *structural congruence* relation able to identify a system

at the level of the internal structure and the *operational semantics* [63] able to handle the mechanics of the system. On the extensional level *the bisimulation*-like relations allow us to associate systems that, for an external observer, have the same behavior.

### Logics for processes

While developing mathematical models for describing and simulating concurrent systems, more complex problems arise from applications, such as the necessity to express complex properties of these systems. We may want to express the fact that a system is composed of two different subsystems, or that, no matter how our system will evolve, it will never reach a particular state or, on the contrary, it will always/sometimes reach that state.

In studying security problems, for example, we may want to be able to specify systems that deal with fresh or secret resources such as channels, locations, and keys. We may want to express properties such as "*the agent has gone away*", "*eventually the agent crosses the firewall*", "*every agent always carries a suitcase*" or "*there is always at most one agent able to perform the action $\alpha$ here*".

In the paradigm proposed recently by systems biology we need to handle big complex systems having extreme dimensions and variable environments. A possible approach is to have a way to specify and check properties of interest. We may need to express properties such as "*somewhere there is a virus*", "*if the virus will meet the macrophage cell then it will be engulfed and hopefully destroyed*", or "*the presence of the protein $x$ will stimulate the reaction $X$*", etc. Observe that such properties do not refer directly to a specified environment. They involve, in a sense, a universal quantifier that is not present in process calculi.

Different approaches have been taken in order to express such properties.

To a very low level, some of these properties can be expressed as properties of the computational traces. Other properties can be expressed by using equations [40], but this solution seems to be quite unnatural and sometimes difficult to use.

One of the successful approaches is inspired by logics. Modal logics have emerged in many domains as a good compromise between expressiveness and abstraction. Furthermore, many logics support useful computational applications, such as model checking. *A model checker* is an algorithm that receives, as input, an initial state of a system and a property and provides, as output, the answer to the question if the given system, at the given state, has the property. If in the logic for which we projected the model-checking algorithm there are temporal operators, such as *sometime in the future*, *always in the future* or *at the next state*, etc., then we can use it to verify properties concerning the evolution of the system.

As the behavior of a concurrent system is, mainly, a succession of affine states in (possibly branching) time, it seems natural to consider the possibility of applying modal (especially temporal) logics for specifying properties of such systems. Moreover, if we succeed in developing model checkers for these logics, then we have tools for making predictions over our systems.

Hennessy-Milner logic [42] is one of the first modal logics that proposes some modal operators, indexed by actions, to describe the behavior of the system in CCS. The idea was then further developed in combination with temporal operators [69] or applied to other calculi [61, 30, 32]. The work of Mads Dam in the logics for processes [31] is relevant in this direction as he introduced a tensor that can express properties of modularity in the system, i.e. it can identify subsystems of a system. All these logics are characterized by their extensional nature with respect to the process calculi.

We say that a logic, defined in the terms of a calculus or of a model, is

*extensional* if it can only distinguish between terms with different behaviors. On the contrary, we say that a logic is *intensional* if it can distinguish between terms with different internal architecture even though their behaviors are the same. Hence a logic for processes is extensional if it cannot distinguish between processes that behave the same, even if these processes are different. On the other hand, an intensional logic for processes should identify a process up to structural congruence, i.e. up to spatially rearranging its subsystems.

Due to the complexity of the concurrent distributed systems, we need to express more than pure behaviors. We need logics that can specify a richer model able to represent computations in space. Such an increased degree of expressiveness is necessary if we want to specify and to reason about notions such as locations, resources, independence, distribution, connectivity and freshness. We may want to derive the behavior of a system by analyzing its subsystems and their behaviors. For such purposes an extensional logic is not enough.

*Spatial logics* seems to be an answer to this problem. The specific applications of mobile computing call for properties that hold at particular locations, and it becomes natural to consider spatial modalities for expressing properties that hold at a certain location, at some locations or at every location. Thus, spatial logics propose some modal spatial operators able to refer to subsystems, in addition to the modal temporal operators. A formula in a spatial logic describes a property of a particular part of the system at a particular time. These spatial modalities have an intensional flavor that distinguishes this class of logics from other modal logics for concurrency, the properties they express being invariant only for simple spatial rearrangements.

If in the extensional logics the truth of a formula is relative to a state

of the whole system at a given moment, in spatial logics the truth of a space-time modal formula is relative to *here* and *now*. Each formula talks about *the current time*, i.e. the current state of execution, and *the current place*, i.e. the location where the observer is placed. Different observers may speak about different faces of the same system at the same moment.

In the last few years spatial logics have been extensively studied, the spatial properties they focus on being, mainly, of two types: whether a system is composed of some identifiable subsystems, and whether a system restricts the use of certain resources to certain subsystems. Some logics [23, 25] have also considered whether a system is composed of named locations.

The potential applications deriving from their expressivity attracted much interest amongst researchers. Unfortunately, the price of the expressivity is quite high: the basic spatial operators, in combination with temporal operators, generate undecidable logics [15, 28, 27], even against small finite pieces of CCS. This means that we cannot solve most of the problems concerning satisfiability, validity and model checking. The situation is caused, mainly, by the presence of *the guarantee operator*, the adjunct of the parallel operator, that involves a universal quantification over the class of processes.

However, some decidable sublogics have been investigated [12, 16, 53, 50] and some model-checking algorithms exist for them. In the light of these results we have two alternatives for avoiding undecidability: either we choose a static calculus [16] which cannot describe our system in evolution, or we choose a dynamic one, but we have to avoid the use of a guarantee operator [12, 53]. The latter alternative is useful only if our system is an isolated one and we have a full description of it. In this sense the possible applications are quite limited. In problems such as those proposed by systems biology, for example, it is not acceptable, as biological systems are almost always subsystems of bigger ones with which they interact.

Very often we do not know too much about these upper systems, or we cannot decide how far up we should go with modeling the systems in order to obtain the information we are looking for. In such situations we still need a spatial operator to act as a universal quantifier, similar to the guarantee operator, but within the limits of decidability. To the best of our knowledge, no such an operator has been proposed.

## 1.2 *To be* is *to know*

*"Not ignorance, but ignorance of ignorance,*
*is the death of knowledge"*
Alfred Whitehead on Philosophy Ignorance Wonder

We now approach the problem from a different perspective.

Consider the following communication protocol proposed in [37]: there are two agents (processes), say a *sender* $S$ and a *receiver* $R$, that communicate by using a communication line. The sender starts with one bit (either 0 or 1) that it wants to communicate to the receiver.

Suppose that the communication line is not a safe one, and it may always lose any message whether it is sent by $S$ or by $R$. Assume in addition, for simplicity, that a message is either received in the same round that it is sent, or lost altogether.

Due to the risk of losing messages, $S$ sends the bit to $R$, in every round, until $S$ receives a message, $(ack)$, from $R$ acknowledging the receipt of the bit. In the round immediately after receiving the bit from $S$, $R$ starts to send the $(ack)$ message and continues to do so from then on. The reason for this behavior is that, in order to stop $S$ sending the bit, $R$ must *know* that the $(ack)$ message has been received and it was not lost. Even if $R$

does not receive any other bit from $S$ for a while, this is not a reason to believe that $S$ is not sending bits anymore, as it might be the case that the messages from $S$ aren't reaching $R$, being lost before delivery. Hence $R$ waits for an $(ack - ack)$ message from $S$ to confirm that $S$ received the $(ack)$ message and, consequently, it stops sending bits.

From this perspective, we have pushed the problem one level further: $S$ sends a message to $R$ hoping that $R$ will receive it. $S$ *cannot know* if $R$ received the $(ack - ack)$ message, so it keeps sending this message until it receives an $(ack - ack - ack)$ message from $R$ acknowledging receipt of the $(ack - ack)$ message. Even if $S$ does not receive any $(ack - ack)$ from $R$ for a while, it might be the case that $R$ is still sending these messages, but they are lost in the communication channel.

As proved in [37], this type of uncertainty is inherent in any system where communication is not guaranteed. Note the usage of the word *"know"* in the description of the previous protocol. This, of course, is not an accident. One of the best models for this type of protocol is in terms of knowledge. In order to stop transmitting a bit, the sender $S$ *has to know* that $R$ received it. Then for $R$ to stop sending $(ack)$, *he has to know that* $S$ *knows* that it received the bit. Further, $S$ will not transmit the $(ack - ack)$ message until it *knows that* $R$ *knows that it knows that* $R$ *received the bit.* The dynamics of the system are, thus, generated by the dynamics of the knowledge in the system. Each transmission of an acknowledgement message is generated by an update of information in the system: $R$ sends $(ack)$ when it receives the bit and the system remains in this state (in which $R$ keeps sending $(ack)$ and $S$ keeps sending $(bit)$) until the knowledge of $S$ is enriched with new information, i.e. until $S$ finds out that $R$ knows the message. Now the system is in a different state, where $R$ continues to send $(ack)$, but $S$ has stopped sending $(bit)$ and has started

sending $(ack - ack)$, and so on.

Certainly, we might use a process calculus to describe the behavior of this system. However we will not comprehend the dynamics of the system up to the level of the dynamics of the knowledge of the two agents involved. The knowledge, in this example, does not necessarily mean update of information about environment. The evolution of the system by itself is the *engine*. During their evolutions $R$ and $S$ do not *learn* more about the communication channel or how to send a safe message. But they start to know more about each other, i.e. more exactly what the other knows. Hence, interesting properties of the evolution of the system involve the knowledge of the two agents.

Expressing properties of the system that concern the *"epistemic side"* of the agents involved seems difficult and unnatural in spatial logics.

The modal epistemic logic [37] represents a special class of modal logics developed for describing epistemic attitudes in multi-agent systems, i.e. for formalizing concepts such as *knowledge*, *belief*, *commitment*, *intention*, etc. These logics are developed for a given class $\mathfrak{S}$ of agents, each agent $i \in \mathfrak{S}$ having an epistemic operator $K_i$ associated with it. Intuitively, $K_i\phi$ means *the agent $i$ knows $\phi$*.

In the previous system we have $\mathfrak{S} = \{S, R\}$. Hence $K_R(bit)$ means that the agent $R$ received/knows the bit sent by $S$. Further $K_S K_R(bit)$ describes the fact that $S$ knows that $R$ received the bit, and this has to be implied by $K_S(ack)$. Then $K_R K_S(ack)$, which is equivalent to $K_R K_S K_R(bit)$, means that $R$ knows that $S$ knows $(ack)$, i.e. that $R$ knows that $S$ knows that $R$ received the bit. Such a situation can happen only if $R$ received the message $(ack - ack)$, hence it has to be implied by $K_R(ack - ack)$, etc.

The knowledge-based approach of multi-agent systems [5, 3, 4, 6, 47, 70,

71, 72] uses Kripke semantics for modeling - the evolution of the system is depicted by traces in a graph with the vertices labeled by the possible states of the system, adjacent vertices representing consecutive states in a possible evolution of the system. Focusing on the agents, these logics are, fundamentally, intensional.

Returning to the distributed systems, we notice a peculiarity that does not fit very well with this approach. As discussed before, these agents are not always a society of distinct, independent individuals as is usual in epistemic logics. In our systems an agent is any definable subsystem. One agent, say $A$, might have a behavior described by the process $P|Q$ meaning that, by its protocol, it runs the processes/programs $P$ and $Q$ in parallel; another agent, $B$, might be described by $P$, while agent $C$ might have the behavior $Q|R$. All of them name subprocesses of a bigger system characterized by the behavior $P|Q|R$. Hence, the behaviors of the three agents are not independent: agent $B$ is a subsidiary of agent $A$ while agents $A$ and $C$ overlap with respect to the subsystem described by $Q$.

Consider again the example of the communication discussed at the beginning of this section. Suppose that the sender does not send just a bit, but a message enciphered by a cipher shared with the receiver. The receiver keeps receiving messages from the channel but it does not consider any information unless it recognizes the cipher. Now, our system is similar to the previous one: it contains two agents, the sender $S$ and the receiver $R$, that behave the same as in the previous case.

At another level of abstraction, we may discover that $S$ itself is a system composed of some parts. It might be the case that $S$ contains two agents: an agent $ME$ that generates an enciphered message to be sent to $R$, and a gate $G$ able to send/receive messages on the channel. The functioning of

$S$ might follow the protocol:

- $ME$ receives messages from $G$, analyzes them and replies with other messages; if it receives the message $(x)$, then it replies with $(ack - x)$;
- $G$ receives messages from $ME$ and sends them on the channel; it also receives messages from the channel, checks them for the cipher and sends those which have it to $ME$.

In this situation we write $S = ME|G$ to say that the system $S$ contains two agents (subsystems), $ME$ and $G$, running in parallel.

At another level of abstraction, we may identify another two subsystems of $S$: an agent $M$ and an agent $EG$

- $M$ generates a message $x$ until it receives the message $(next)$ from $EG$, then it starts to generate the message $(ack - x)$;
- $EG$ enciphers the messages received from $M$ and sends them on the channel; it also receives messages from the channel and checks them for the cipher; if the deciphered message is $(ack - x)$, then it sends the message $(next)$ to $M$.

In this way, we may consider $S = M|EG$.

In the same way we may discover that $S$ actually contains three agents: an agent $M$ that generates the messages, an agent $E$ that enciphers/deciphers messages and a gate $G$ that sends/receives messages on the channel. Generally speaking, $S$ functions according to the following protocol:

- $M$ generates a message $x$ until it receives the message $(next)$ from $E$, then it starts to generate the message $(ack - x)$;
- $E$ enciphers the messages received from $M$ and deciphers the messages received from $G$; if $E$ deciphers the message $(ack - x)$ received from $G$, it then sends the message $(next)$ to $M$;

- $G$ receives messages from $E$ and sends them on the channel; it also receives messages from the channel, checks them for the cipher and, if they have it, sends them to $E$.

In this last case, $S = M|E|G$.

Hence we can see how, on different levels of abstraction, our system can be composed of different subsystems $S = ME|G = M|EG = M|E|G$. Observe that agent $E$ is ontologically addicted to the agents $ME$ and $EG$, while these last two have in common the subsystem $E$. If we chose the classical epistemic logic to describe the system, then, in order to be able to express properties of all these agents, we should take $\mathfrak{S} = \{M, E, G, ME, EG\}$ and then, independently for each agent, ascribe its knowledge. This solution might be a correct one, but it is unsatisfactory. As the agents are ontologically related, we expect to derive relations between their knowledge from their ontological relations. We expect, for example, to be able to syntactically derive the knowledge of $ME$ from the knowledge of $M$ and the knowledge of $E$.

## 1.3 Our approach

As presented in the previous sections, both approaches to model multi-agent systems (the one from process algebra and the one from epistemic logics) have their limitations. The spatial logics able to express enough complex properties of the systems are undecidable, and abstracting the agents on the level of their behavior only, makes the discourse about intelligent actions - such as epistemic actions - difficult and unnatural. On the other hand, classical epistemic logic focuses on agents, but the agents, on the ontological level, are not compositional and not hierarchically organized.

Our approach tries to combine both paradigms to obtain maximum advantages while still ensuring decidability. We propose a new class of logics: *Dynamic Epistemic Spatial Logics*.

From the spatial logics perspective, our logics extend the Hennessy-Milner logic with the parallel operator (hence are spatial logics) and epistemic operators. The role of the epistemic operators is not only to focus more directly on agents, but they are also able to do most of the job of the guarantee operator while maintaining decidability. In our logics the epistemic agents are just processes, and their epistemic operators help to specify properties about the system in a given context, thus replacing the guarantee operator. In our approach $K_P\phi$ holds, *the agent having the behavior* $P$ *knows* $\phi$, iff $\phi$ is satisfied by any process in a given context (class of processes) having $P$ as a subprocess.

From the epistemic logics perspective, we propose a new class of epistemic logics by imposing an algebraical structure (CCS-like) on the class $\mathfrak{S}$ of agents. In this way we may assume compositional and hierarchically organized agents. Now the agents in our system are: individuals, societies of individuals, societies of societies of individuals, etc. Our logic can analyze the dynamic of knowledge between these complex agents.

Unlike in the classical epistemic logic, our agents are not always active. For a system in the state $\alpha.P|Q$, the active agents are $\alpha.P$ and $Q$, but $P$ is not active (yet). It will be activated in a future state after the action $\alpha$ is performed. On the other hand, after doing $\alpha$, the agent $\alpha.P$ does not exist anymore (unless a copy of $\alpha.P$ is still active inside $Q$).

We may also have clones of the same agent working in parallel. For example, in the system described by $P|Q|P$ we have two agents that can be described (only!) by $P$. Hence the knowledge $K_P\phi$ refers to the knowledge

of $P$ without referring to one of the two. This system is different and cannot be identified with $P|Q$, because in this system only one copy of $P$ works. All these features are new for epistemic logics.

By combining the peculiarities of epistemic logics with the characteristics of Hennessy-Milner logic, our approach tries to provide tools for expressing complex properties of systems modelled in CCS. Thus, we propose three systems: a Dynamic Spatial Logic, and two Dynamic Epistemic Spatial Logics with different levels of expressivity. The Dynamic Spatial Logic combines Hennessy-Milner-like dynamic operators with the parallel operator and with the operators of the classical propositional logic, while the Dynamic Epistemic Spatial Logic adds the epistemic operators indexed by processes. The extension of this last system came with some specific expressivity, being able to specify properties such as "*the system can perform the action $\alpha$ and the agent who is doing $\alpha$ is exactly the agent $P$*". Hence it can name the active agent that originates the action of the whole system we analyze. Such properties are important in causality problems - performance analysis or debugging - where we know that there is an error in our system, or there is a high-cost operation, but we do not know which agent causes this. Being able to write down a specification in which not only the actions performed by our system but also the authors of the actions are mentioned, may provide a way to handle causality problems. In this context a causality problem can be mapped into a model-checking or a validity-checking problem. If the logics are decidable, as in our case, such problems can be solved in finite time.

For these logics we propose complete Hilbert-style axiomatic systems, which helps in understanding the basic algebraical behavior of the classical process operators. To the best of our knowledge, this is the first time in

the literature that Hilbert-style axiomatic systems have been proposed for spatial logics.

We prove that our systems are sound and complete with respect to process semantics. Thus, many properties can be syntactically verified and proved. Moreover the interplay of our logical operators allows expression, inside the syntax, of validity and satisfiability for formulas. We also have characteristic formulas able to identify a process (agent) up to structural congruence (cloned copies).

Last, but not least, we prove, for all the systems, the finite model property with respect to the the chosen semantics. Thus, we can decide if a process, in a certain context, satisfies or not a formula by checking if the formula is satisfied by one of the processes in a finite set defined, univocally, by the formula. As this set is finite, we can decide on satisfiability, validity and model-checking problems in finite time. This feature, in the context of a sound, complete, axiomatic system, presents our logics as appropriate tools for specifying and verifying properties of multi-agent systems modelled by CCS.

In proving the finite model property we used a new congruence on processes - *the structural bisimulation*. To the best of our knowledge, this has not been studied before. A conceptually similar congruence has been proposed in [16], but for static processes only. The structural bisimulation is interesting in itself, as it provides a bisimulation-like description of the structural congruence. Informally, it is an approximation of the structural congruence bound by two dimensions: the *height* and the *weight* of a process. The bigger these sizes, the better approximation we obtain. At the limit, for sizes big enough, we find exactly the structural congruence.

## 1.4 Outline of the Thesis

The thesis is organized in nine chapters, as follows.

***Chapter 2*** reports the theoretical background needed in the main body of this work. In particular:
*Section 2.1* introduces the main concepts in process algebra focusing on restriction-free CCS that will be used, further, as semantics for our logics. As closely related with the topic of spatial logics, we will present, in addition, the syntaxes of $\pi$-calculus and Ambient Calculus.
*Section 2.2* presents the logics for processes, considering both the extensional and the intensional approaches.
*Section 2.3* recalls the classic modal epistemic logic, the syntax, the semantics and some completeness results.

***Chapter 3*** introduces the fragment of CCS oh which we will focus in this thesis. We define the notions of canonical representation and size of a process, and using them we develop the pruning method for processes and we propose a new congruence relation - the structural bisimulation. Eventually the results will be extended from processes to classes of processes.

***Chapter 4***, following [54], analyzes the Dynamic Spatial Logic, $\mathcal{L}_{DS}$, an extension of Hennessy-Milner logic with spatial operators. By exploiting the properties of the structural bisimulation, we prove the finite model theory for $\mathcal{L}_{DS}$ against process semantics, concluding on decidability of the satisfiability, validity and model-checking problems. A Hilbert-style axiomatic system is developed for it; the soundness and completeness of the system with respect to process semantics are proved. Interesting properties of semantics are then deduced in the system.

***Chapter 5***, following [55] proposes the first Dynamic Epistemic Spatial Logic, the system $\mathcal{L}^{\mathfrak{S}}_{DES}$. It is introduced as an extension of $\mathcal{L}_{DS}$ with epistemic operators. We define a process semantics for it and prove the finite model property, hence decidability. In addition we propose a Hilbert-style axiomatic system sound and complete against the considered semantics. We point out the similarities between the axioms of $\mathcal{L}^{\mathfrak{S}}_{DES}$ and the axioms of classic epistemic logic. Interesting properties of processes are proved in the system.

***Chapter 6*** extends the results in the previous chapter, following the lines of [56], proposing a more expressive logic, the system $\mathcal{L}^{\mathfrak{S}^+}_{DES}$, that can express some special transitions on processes labeled not only by action, but also by the agent that performs it. As for $\mathcal{L}^{\mathfrak{S}}_{DES}$, we prove the finite model property that ensures the decidability for satisfiability, validity and model checking. A Hilbert-style system, that extends the one introduced in the previous chapter, is developed for $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ and is proved to be sound and complete with respect to process semantics.

***Chapter 7*** outlines the future works. We point to possible continuations of this idea also on the level of expressivity for our logics as on semantics by extending it toward more complex calculi. Our intention is to check, from this perspective, the limits of expressivity within decidability for epistemic spatial logics.

***Chapter 8*** applies our logics in specifying properties of some real systems. Thus, we show the advantages of using our logics in security, system biology, and we discuss some possible applications in causality - debugging and performance analysis.

***Chapter 9*** eventually concludes the work by resuming the main results.

# Chapter 2

# Preliminaries

In this chapter we present the background theories that are used or referred to in this thesis. Thus we will introduce and discuss some process calculi, spatial logics and epistemic logic. The reader familiar with these topics can safely proceed to the next chapter.

## 2.1 Overview on Process Algebra

We begin this section with a brief outline of the restriction-free CCS calculus [57]. For a fragment of this calculus we will develop, later, our logics. In the end, we succinctly present the syntax of two more process calculi, $\pi$-calculus [68, 62] and Ambient Calculus [21, 27], as representing key moments in the evolution of the computational paradigm, fully reflected in topic of the spatial logics.

**Definition 2.1.1** (Processes)**.** Assume a denumerable set of *actions* $\mathbb{A}$ ranged over by $\alpha, \beta, ....$ We call *process* any term $P$ generated by the grammar in table 2.1. We denote by $\mathfrak{P}^+$ the class of all processes.[1]

[1] We choose this notation because we intend to denote by $\mathfrak{P}$ only a subclass of CCS processes introduced in the next chapter.

| | | | |
|---|---|---|---|
| $P$ | ::= | $0$ | (inactive process) |
| | | $\mid \alpha.P$ | (prefixing) |
| | | $\mid P\|P$ | (parallel composition) |
| | | $\mid P+P$ | (nondeterministic choice) |
| | | $\mid X$ | (variable) |
| | | $\mid RecX.P$ | (recursive equation) |

Table 2.1: The syntax of restriction-free CCS calculus

The intuition behind this grammar is that the behavior of any system can be depicted by using the proposed operators on top of the *inactive process*.

The *inactive process*, 0, represents any process which does nothing (at the level of abstraction we choose for describing our system).

The *prefixed process* $\alpha.P$ is used to describe the system that can perform, as the first action, $\alpha$, after which it reaches a state that can be traced by the process $P$.

The *parallel operator* $P|Q$ is used to denote the process composed by two subprocesses running in parallel. Hence, if the process $P$ can evolve to a process $P'$ and the process $Q$ is able to reach a state labeled $Q'$, then the process $P|Q$ goes either to the state $P'|Q$ or $P|Q'$.

The *nondeterministic choice operator* is a sequential operator meant to express two processes that cancel each other out. Thus if a process $P$ can evolve to $P'$ and a process $Q$ to $Q'$, then the process $P+Q$ evolves either to $P'$ or to $Q'$. In this way, we model the behavior of a process which nondeterministically chooses between the two possible futures for the system.

The *recursive variable* and *recursive equations* were introduced in the syntax in order to allow us to manipulate systems with recursive behaviors. As usual in calculi with recursion, we will define $P\{X \Leftarrow Q\}$ as the term obtained by replacing, in the syntax of $P$, all the occurrences of the variable $X$ by $Q$. We call the terms containing variables *open terms* and we denote

such a term by $P[X]$, while by $P[Q]$ we denote $P[X]\{X \Leftarrow Q\}$.

The actions collected in $\mathbb{A}$ are given in pairs of distinct (co)actions, characterized by the involution $co : \mathbb{A} \longrightarrow \mathbb{A}$. This function associates each action $\alpha$ with the action $\overline{\alpha}$ such that $\overline{(\overline{\alpha})} = \alpha$. The reason for adding such a structure onto $\mathbb{A}$ is to be able to model communication between processes. Thus the system characterized, in the current state, by the process $\alpha.P|\overline{\alpha}.Q$ can evolve, by the communication $(\alpha, \overline{\alpha})$, to the state described by $P|Q$. Hence, the use of the involution is meant to model the input/output attributes of communication.

In the calculus, we will identify different syntactic representations for the same process. In this respect, the nondeterministic choice and the parallel operator are commutative, associative and the inactive process 0 is a neutral element for them. In order to encode all these properties we introduce *the structural congruence*. Two structurally congruent processes are two processes that describe, in a syntactically different manner, the same system. The structural congruence is not only an equivalence relation, but also a congruence. Namely, replacing structurally congruent processes $P$ and $Q$ in any CCS-context (open term) $C[X]$ results in structurally congruent agents $C[P]$ and $C[Q]$.

**Definition 2.1.2** (Structural congruence)**.** The *relation of structural congruence* is defined as the least congruence $\equiv$ on processes satisfying the axioms in table 2.2.

Further, we focus on defining the transitions for our systems. Usually, for process description languages (such as CCS), this is given by an *operational semantics* defined in the style of Plotkin [63]. The behavior of any

| | |
|---|---|
| $P\|0 \equiv P$ | $P + 0 \equiv P$ |
| $P\|Q \equiv Q\|P$ | $P + Q \equiv Q + P$ |
| $P\|(Q\|R) \equiv (P\|Q)\|R$ | $P + (Q + R) \equiv (P + Q) + R$ |
| $RecX.P \equiv P\{X \Leftarrow RecX.P\}$ | |

Table 2.2: The axioms the structural congruence

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \qquad \frac{P \equiv Q \qquad P \xrightarrow{\mu} P'}{Q \xrightarrow{\mu} P'}$$

$$\frac{P \xrightarrow{\mu} P'}{P|Q \xrightarrow{\mu} P'|Q} \qquad \frac{P \xrightarrow{\mu} P'}{P + Q \xrightarrow{\mu} P}$$

$$\frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\overline{\alpha}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

Table 2.3: The transition system of restriction-free CCS

process $P$ is described in terms of the transitional relations, denoted by $\longrightarrow$, and introduced by a set of rules called the *labeled transition system*. So, for instance, $P \xrightarrow{\alpha} Q$ means that the process $P$ can evolve into $Q$ by performing the action $\alpha$.

**Definition 2.1.3** (Transition systems)**.** Given the set $\mathbb{A}^+ \stackrel{def}{=} \mathbb{A} \cup \{\tau\}$ of labels (where $\tau \notin \mathbb{A}$), the standard *transition system* for CCS, usually called *interleaving*, is $ITS = \langle \mathfrak{P}^+, \mathbb{A}^+, \longrightarrow \rangle$, where $\longrightarrow \subset \mathfrak{P}^+ \times \mathbb{A}^+ \times \mathfrak{P}^+$ is the *transition relation* defined by the rules in table 2.3, with the assumption that $P \xrightarrow{\mu} Q$ denotes $\langle P, \mu, Q \rangle \in \longrightarrow$, $\mu$ was used to range over $\mathbb{A}^+$ and $\alpha$ to range over $\mathbb{A}$. We denote by $\longrightarrow^*$ the transitive closure of $\longrightarrow$.

The role of $\tau$ in the previous definition is to denote the *silent action*. In any process algebra, the action alphabet contains a distinguished element $\tau$

meant to label transitions which cannot be influenced by interactions with the external environment [59], called the silent action. An internal communication, for example, is usually labeled as a silent action, as the system changes its state without exchanging information with the environment.

Observe that we can, naturally, associate to any process $P$ a labeled tree that depicts the unfolding of the transition system together with the evolution of the process. It has the nodes labeled by the processes $Q$ such that $P \longrightarrow^* Q$ and the edges defined by the transition relation and labeled by the actions performed. Consider, for example, the process $P \equiv \alpha.(\beta.0|\gamma.0)$. We can depict it, following the previous observations, using the unfolding tree in the left part of figure 2.1.

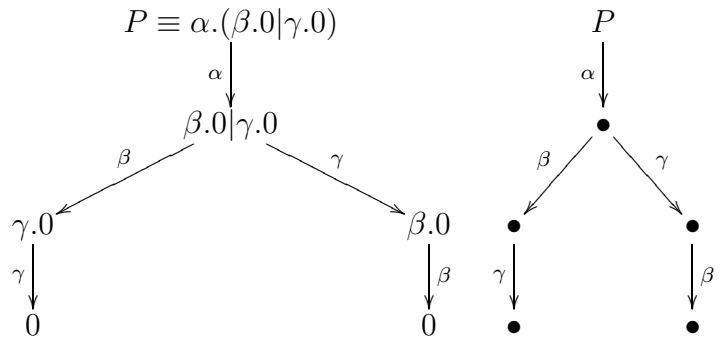


Figure 2.1: The transition system associated to the process $P \equiv \alpha.(\beta.0|\gamma.0)$

If we focus only on the transition system, then we can draw the tree in the right side of figure 2.1.

The operational semantics are generally too *intensional*, in the sense that they discriminate also behaviors that should reasonably be considered interchangeable. Informally, we will call any two systems that behave the same *bisimilar*, i.e. an external observer cannot distinguish them.

A serious challenge in process algebra is to provide the right definition for bisimulation. By fixing the notion of *observation* for processes, the

emerging equivalences generate different semantics. A large number of such equivalences have been proposed in the literature, each of them stressing a distinct peculiarity of process behavior.

It is not in the scope of this thesis to go deeper into this matter. We only want to present "the flavor" of the bisimulation problem and to conclude about its relation with the structural congruence. For this purpose we propose only the definition of *strong bisimulation* [58]. We call two processes $P$ and $Q$ *strongly bisimilar*, denoted by $P \sim Q$, if each transition from $P$ can be mimicked by a transition form $Q$ and vice versa, both transitions leading to strongly bisimilar derivatives $P'$ and $Q'$.

As for the case of structural congruence, the strong bisimulation over processes in a CCS-like environment is not only an equivalence relation, but also a congruence. We recall that this mean that using strong bisimilar processes $P \sim Q$ in the same CCS-context (open term) $C[X]$ results in strong bisimilar agents $C[P]$ and $C[Q]$.

**Definition 2.1.4** (Strong bisimulation)**.** A *strong bisimulation* is a binary relation $\mathcal{S}$ over processes having the property that $P\mathcal{S}Q$ implies

- if $P \xrightarrow{\alpha} P'$ then for some $Q'$, $Q \xrightarrow{\alpha} Q'$ and $P'\mathcal{S}Q'$
- if $Q \xrightarrow{\alpha} Q'$ then for some $P'$, $P \xrightarrow{\alpha} P'$ and $P'\mathcal{S}Q'$

$P$ is *strong bisimilar* with $Q$, written $P \sim Q$, if $P\mathcal{S}Q$ for some strong bisimulation $\mathcal{S}$.

**Example 2.1.1.** *To comprehend the difference between the structural congruence and the bisimulation, consider the processes*

$$P \equiv \alpha.0|\beta.\gamma.0$$

$$Q \equiv \alpha.\beta.\gamma.0 + \beta.(\alpha.0|\gamma.0)$$

*The system described by $P$ can perform either action $\alpha$ reaching the state $\beta.\gamma.0$ where it can perform $\beta$ then $\gamma$ and stops, or it can perform $\beta$ reaching the state $\alpha.0|\gamma.0$ where it can go further by doing $\alpha$ followed by $\gamma$ or $\gamma$ followed by $\alpha$ and stops. See the left side of figure 2.2.*

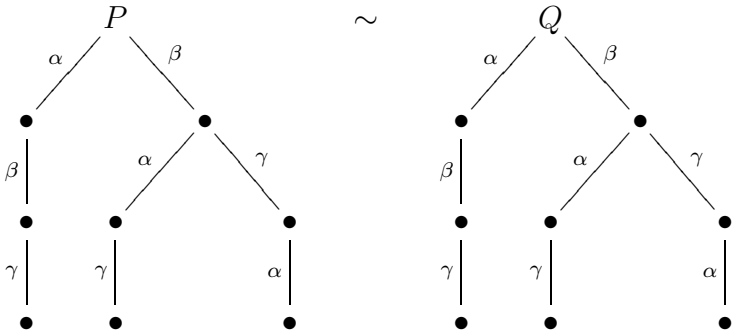


Figure 2.2: The transition trees of $P$ and $Q$

*Consider now the system described by $Q$: it can either perform $\alpha$ followed by $\beta$, $\gamma$ and stops, or it can choose to perform $\beta$ reaching the state $\alpha.0|\gamma.0$ where it can perform $\alpha$ then $\gamma$ and stops or $\gamma$ then $\alpha$ and stops. Hence, its behavior is depicted in the right side of figure 2.2:*

*Thus the behaviors of the two systems are identical. An external observer cannot distinguish between them and cannot say if they are different or clones. Hence the systems are bisimilar.*

*Still, if we look inside them, we notice big differences. While $P$ contains two parallel agents, the two agents that are subsystems of $Q$ are related by $+$, meaning that the action of one will cancel out the future of the other. Moreover, one of the agents of $Q$ is formed, itself, by two parallel subsystems, which is not the case with any of the two agents of $P$.*

*Hence the systems are not structurally congruent. If this is not important for an external observer, it may be important if the two systems interact with the environment. Indeed, if the systems work as subsystems in*

*some interacting environments, then the existing differences between them may induce different behaviors for the upper systems.*

The standard syntax of CCS contains, in addition to the subcalculus presented here, the *restriction operator* $(\nu x)P$. This operator acts as a static binder for the action name $x$ local to the process $P$ that it prefixes. In other words, $x$ is a unique name (for action) in $P$ that is different from all the external names. The restriction operator is beyond the scope of this paper and for this reason we will not proceed further by presenting its algebra.

To end this section we briefly present two more calculi to which we contextually refer further, as having direct relation with the problematic of spatial logics - $\pi$-calculus and Ambient Calculus. However, our logics will only focus on a fragment of CCS, and for this reason we do not need to go into too much detail concerning these other calculi.

### $\pi$-calculus

The $\pi$-calculus [62, 68] is the result of going further with modeling communication issues. It arises from the necessity to have a calculus of communicating systems in which one can naturally express processes with changing structure. Not only may the component agents of a system be arbitrarily linked, but communication between neighbors may carry information that changes that linkage. CCS can at best express this mobility indirectly. Consequently, $\pi$-calculus evolved from the CCS calculus by replacing the syntax of actions with a more complex one.

| | | | |
|---|---|---|---|
| $P$ | ::= | $0$ | (inactive process) |
| | | $\mid P \mid P$ | (parallel composition) |
| | | $\mid P + P$ | (nondeterministic choice) |
| | | $\mid \overline{y}x.P$ | (output prefix) |
| | | $\mid y(x).P$ | (input prefix) |
| | | $\mid \tau.P$ | (silent prefix) |
| | | $\mid [x = y]\pi.P$ | (matching prefix) |
| | | $\mid\ !P$ | (replication) |
| | | $\mid (\nu x).P$ | (restriction) |

Table 2.4: The syntax of $\pi$-calculus

The syntax of the $\pi$-calculus is constructed for an infinite set $\mathcal{N}$ of names ranged over by $x, y, ...$ and is presented in table 2.4.

The actions of CCS were replaced here by capabilities. The process $\pi.P$ has a single *capability*, expressed by $\pi$; the process $P$ cannot proceed until that capability has been exercised. The capabilities are of four types:

- $\overline{y}x.$ is called an *output prefix*. $\overline{y}$ may be thought of as an *output port* of an agent which contains it. $\overline{y}x.P$ outputs the name $x$ at port $y$ and then behaves like $P$.

- $y(x)$ is called an *input prefix*. A name $y$ may be thought of as an input port of an agent. $y(x).P$ inputs an arbitrary name $z$ at port $y$ and then behaves like $P\{x \Leftarrow z\}$ (we use this to denote the substitution of $x$ by $z$, in all its occurrences in the syntax of $P$).

- $\tau.$ is called the *silent prefix*. $\tau.P$ performs the silent action and then behaves like $P$.

- $[x = y]\pi.P$ can evolve as $\pi.P$ if $x$ and $y$ are the same name, but can do nothing otherwise.

The *replication operator* $!P$ has been proposed to express infinite recursive behaviors, thus replacing the recursive equations of CCS. $!P$ can be

thought of as an infinite composition $P|P|...$ or, equivalently, as a process that can generate a copy of $P$ at any time, hence having the property that $!P$ and $P|!P$ are different syntaxes for the same process.

### Ambient Calculus

With Ambient Calculus [21], the complexity of modeling multi-agent systems goes further in trying to specify spatial structures for distributed concurrent systems. The intuition was to model systems where not only the linkage between agents can be reconfigured, but the agents themselves can be moved in the topology of the system. Thus, the Ambient Calculus came with the notion of named locations and with capabilities denoting movements between these locations.

Briefly, the syntax is presented in table 2.5, being constructed for a set $\mathcal{N}$ of names ranged over by $n, m....$ These names denote *ambients*.

An *ambient* is a bounded place in the system where the computation happens. Hence there exists a boundary that defines/separates the inside of the ambient from the outside. This boundary can physically move in the sense that the ambient behaves as a box enclosing a computational device. If we reconnect a laptop to a different network, all the address spaces and file systems within it move accordingly and automatically. If we move an agent from one computer to another, its local data should move accordingly and automatically. In order to integrate such syntactic constructs in our system we accept that the ambients have names that can be used to refer to them. We have, thus, a set $\mathcal{N}$ of ambient names.

An ambient can be nested in other ambients, hence our system will be represented as a hierarchical network of locations. Consequently, each

| | | | |
|---|---|---|---|
| $P$ | $::=$ | | (processes) |
| | | $0$ | (inactive process) |
| | | $\mid P \mid P$ | (parallel composition) |
| | | $\mid (\nu x).P$ | (restriction) |
| | | $\mid\ !P$ | (replication) |
| | | $\mid \langle M \rangle$ | (output) |
| | | $\mid (n).P$ | (input) |
| | | $\mid n[P]$ | (ambient) |
| | | $\mid M.P$ | (capability action) |
| $M$ | $::=$ | | (capabilities) |
| | | $in\ n$ | (can enter into $n$) |
| | | $out\ n$ | (can exit out of $n$) |
| | | $open\ n$ | (can open $n$) |
| | | $M.M'$ | (path) |

Table 2.5: The syntax of Ambient Calculus

$$m[P] \mid n[in\ m.Q \mid R] \xrightarrow{in\ m} m[\ n[Q|R] \mid P\ ]$$

$$m[\ n[out\ m.Q|R] \mid P\ ] \xrightarrow{out\ m} m[P] \mid n[Q \mid R]$$

$$m[P] \mid open\ m.Q \xrightarrow{open\ m} P|Q$$

Table 2.6: Spatial transitions of Ambient Calculus

ambient has a collection of local agents/processes. These are the computations that run directly within the ambient and, in a sense, control the ambient. For example, they can instruct the ambient to move.

Unlike in other calculi, in Ambient Calculus communication is not the only possible action. There are capabilities that model topological movements in the system. These spatial transitions are presented in table 2.6.

## 2.2 Logics for processes

In this section we present some of the logics developed for processes. These logics, extensions of the classical logics, aim to underpin the process calculi in order to provide a way to specify complex properties concerning processes. The main intuition is to describe the specification in the language of logics and to use the satisfiability relation to assert that a process $P$ has the property $\phi$, i.e. $P \models \phi$.

We start with the extensional approaches, as, historically, they were the first, and then we will focus on spatial logics.

### 2.2.1 Hennessy-Milner Logic

Hennessy-Milner logic [42], is an extension of the classic propositional logic with some modal operators indexed by CCS actions. The full syntax is given by the grammar:

$$\phi ::= \top \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \langle\mu\rangle\phi$$

The satisfaction relation, $P \models \phi$, is introduced similarly to classical modal logics, by interpreting the graph of labeled transitions of a CCS process $P$ as a Kripke structure. This means that we associate to the process $P$ a graph having the vertices labeled by the processes describing transition moments of $P$ and the edges connecting the processes directly related by the transition relation. Treating such a graph as a Kripke structure, we can introduce the semantics as for classical modal logics.

$P \models \top$ always
$P \models \phi_1 \wedge \phi_2$ iff $P \models \phi_1$ and $P \models \phi_2$
$P \models \neg\phi$ iff $P \not\models \phi$

$P \models \langle\mu\rangle\phi$ iff there is a transition $P \xrightarrow{\mu} Q$ and $Q \models \phi$

Observe the similarity of $\langle\mu\rangle$ operator to the modal diamond operator. On the basis of this similarity we can propose, by duality, the derived operator $[\mu]$ having the following semantics:

$P \models [\mu]\phi$ iff for any transition $P \xrightarrow{\mu} Q$ (if any) we have $Q \models \phi$

The syntax can be easily generalized from actions of CCS to sets of actions, by replacing the operators $\langle\mu\rangle$ and $[\mu]$ by the operators $\langle A\rangle$ and $[A]$, with $A \subset \mathbb{A}^+$ a set of CCS actions. The semantics will be defined following the same intuition:

$P \models \langle A\rangle\phi$ iff $\exists\mu \in A$ such that $P \xrightarrow{\mu} Q$ and $Q \models \phi$
$P \models [A]\phi$ iff $\forall\mu \in A$ such that $P \xrightarrow{\mu} Q$ (if any) we have $Q \models \phi$

Hennessy-Milner logic have been studied also in relation to temporal operators [69]:

$$\phi ::= \top \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \langle\mu\rangle\phi \mid AG\phi \mid EF\phi \mid AF\phi \mid EG\phi$$

The associated semantics combines the semantics of Hennessy-Milner logic with the classic semantics of temporal logics [36]:

$P \models AG\phi$ iff for all runs $P \xrightarrow{\mu_1} P_1 \xrightarrow{\mu_2} P_2 \xrightarrow{\mu_3} \ldots$ and all $i \geq 0$, $P_i \models \phi$
$P \models EF\phi$ iff for some run $P \xrightarrow{\mu_1} P_1 \xrightarrow{\mu_2} P_2 \xrightarrow{\mu_3} \ldots$ and some $i$, $P_i \models \phi$
$P \models AF\phi$ iff for all runs $P \xrightarrow{\mu_1} P_1 \xrightarrow{\mu_2} P_2 \xrightarrow{\mu_3} \ldots$ for some $i \geq 0$, $P_i \models \phi$
$P \models EG\phi$ iff for some run $P \xrightarrow{\mu_1} P_1 \xrightarrow{\mu_2} P_2 \xrightarrow{\mu_3} \ldots$ and all $i \geq 0$, $P_i \models \phi$

The meanings of the temporal operators are, thus, the standard ones:

$P \models AG\phi$ means that *all the processes reachable from P satisfy* $\phi$;
$P \models EF\phi$ means that *some processes reachable from P satisfy* $\phi$;
$P \models AF\phi$ means *eventually a process will be reached from P satisfying* $\phi$;
$P \models EG\phi$ means *some runs always satisfy* $\phi$.

Further, this logic was extended to other process calculi [30, 32, 61].
What is common to all of these is that they satisfy theorem 2.2.1 proving their extensional nature.

**Theorem 2.2.1.** *If* $P \sim Q$ *and* $P \models \phi$ *then* $Q \models \phi$.

### 2.2.2 Spatial Logics

The need to describe spatial properties for distributed systems has recently arisen, generating the class of *spatial logics*. The initial motivation for studying these logics was the necessity to specify systems that deal with fresh or secret resources such as keys, nonces, channels, locations, and to express complex properties of such systems.

Informally, these are properties such as "*the agent has gone away*", "*eventually the agent crosses the firewall*", "*every agent always carries a suitcase*", "*somewhere there is a virus*" or "*there is always at most one agent called n here*". Several ways of formalizing these assertions have been tried, such as equational approaches [40], most of them being either difficult or unnatural.

Modal logics inspired a solution, but for specifying properties such as those exemplified before, the extensional logics, such as Hennessy-Milner logic or temporal extensions of it, were not sufficiently expressive. We need logics that can specify a richer model able to represent computation in space. Such an increased degree of expressiveness is necessary if we want

to specify and reason about notions like locations, resources, independence, distribution, connectivity or freshness. We may want to derive properties of a system by analyzing its subsystems and their behaviors in the context of their topology. In this respect, speaking only about pure behaviors is not enough; it is necessary to specify properties that hold at a certain location, at some locations or at every location.

To fulfill these requirements, *spatial logics* were developed. They represent extensions of classic propositional logic with some modal spatial operators and temporal operators. Being the specific applications of mobile computing, it makes sense to consider spatial modalities for expressing properties that hold at a certain location. Thus, spatial logics propose, in addition to the modal temporal operators, some modal spatial operators able to refer to subsystems. A formula in a spatial logic describes a property of a particular part of the system at a particular time. It is thus sensitive to the current time, i.e. the current state of execution, and the current place, i.e. the location where the observer is placed.

These spatial modalities have an intensional flavor that distinguishes this class of logics from other modal logics for concurrency, the properties studied by them being invariant only for simple spatial rearrangements.

The spatial properties spatial logics consider are essentially of two kinds: whether a system is composed of two or more identifiable subsystems, and whether a system restricts the use of certain resources to certain subsystems. Some logics [23, 25] also consider whether a system is composed of named locations. In the last few years these logics have been studied in depth.

In this section we will present, briefly, spatial logics and we will indicate the results relevant for our approach. As our interest focuses on express-

ing spatial properties in decidability against restriction-free CCS, we will only mention the additional features studied by the spatial logics, such as the fresh names quantifiers [24, 44] or location operators [23, 25], without proceeding further by presenting details of them.

#### Syntax of Spatial Logics

We present further the syntax of a spatial logic [14, 13] developed for asynchronous $\pi$-calculus [68]. We have chosen to present this, as it is one of the most complete, but have chosen not to add the second-order variable and quantifiers, as in [14], as we are mainly interested in the spatial issues.

**Definition 2.2.1.** Given the set $\mathcal{N}$ of names of $\pi$-calculus and an infinite set $\mathcal{X}$ of name variables (mutually disjoint from $\mathcal{N}$), formulas of spatial logics are presented in table 2.7.

As expected from such a rich syntax the expressivity of the logic is very high. Moreover, in [14] also second-order quantification is used. We will now outline the intension behind each operator.

The formula 0 will be satisfied by any process in the structural congruence class of 0. It will be, hereafter, distinguished[2] by $\bot$, which is the formula meant to express *false*.

The *parallel operator* $\phi|\psi$ it is used to specify any process that can be decomposed into processes that satisfy respectively $\phi$ and $\psi$.

*Guarantee* is the logical adjunct of composition: $\phi \triangleright \psi$ is satisfied by those processes whose composition with any process satisfying $\phi$ results in a process satisfying $\psi$. It was anticipated by Mads Dam in his work

[2]Some syntaxes of classical logics use 0 for $\bot$; this is not the case here.

| | | | |
|---|---|---|---|
| $x, y, z$ | $\in$ | $\mathcal{X}$ | (name variables) |
| $\mu$ | $\in$ | $\mathcal{N} \cup \mathcal{X}$ | (names or name variables) |
| | | | |
| $\phi, \psi$ | ::= | | (formulas) |
| | | $\bot$ | (false) |
| | | $\phi \wedge \psi$ | (conjunction) |
| | | $\phi \rightarrow \psi$ | (implication) |
| | | $0$ | (void) |
| | | $\phi \vert \psi$ | (composition) |
| | | $\phi \triangleright \psi$ | (guarantee) |
| | | $\mu \circledR \phi$ | (revelation) |
| | | $\phi \oslash \mu$ | (hiding) |
| | | $\mu\langle\mu'\rangle$ | (message) |
| | | $\forall x.\phi$ | (first order universal quantification) |
| | | $Nx.\phi$ | (fresh name quantifier) |
| | | $\blacklozenge\phi$ | (next step) |

Table 2.7: The syntax of Spatial Logics

on Relevance logics [31] where it was used for expressing modularity for processes.

The *hiding* and *revelation* operators are used to express the semantics of the new name operator in process calculus. Hence a process satisfies $n \circledR \phi$ if it belongs to the equivalence class of the process $(\nu n)P$ where $P$ satisfies $\phi$; similarly, a process $P$ satisfies $\phi \oslash n$ if the process $(\nu n).P$ satisfies $\phi$.

The *fresh name quantification* refers to the same problem. A process satisfies $Nx.\phi$ if for (some/all) fresh names $n$ (fresh in the process and in the formula), it satisfies $\phi\{x \Leftarrow n\}$. This quantifier exhibits the universal/existential ambivalence typical of freshness: a property holding for some fresh names should also hold for any other fresh name. The development of this operator follows the direction of [38].

The message $m\langle n\rangle$ holds for processes structurally congruent with a

message $m\langle n\rangle$ in $\pi$-calculus.[3]

The next step operator is the classical temporal operator used to describe the future of the system. So a process $P$ will satisfy $\blacklozenge\phi$ when it can perform a transition $P \longrightarrow Q$ and $Q$ satisfies $\phi$.

#### Semantics of Spatial Logics

We now present the semantics of the spatial logic introduced before. We define the satisfaction relation $P \models \phi$, where $P$ is a process and $\phi$ a closed formula, inductively as follows:

$P \not\models \bot$ for any process $P$
$P \models \phi \wedge \psi$ iff $P \models \phi$ and $P \models \psi$
$P \models \phi \rightarrow \psi$ iff $P \models \phi$ implies $P \models \psi$
$P \models 0$ iff $P \equiv 0$
$P \models \phi|\psi$ iff $P \equiv Q|R$, $Q \models \phi$ and $R \models \psi$
$P \models \phi \triangleright \psi$ iff for any process $Q \models \phi$ we have $P|Q \models \psi$
$P \models n\textcircled{R}\phi$ iff $P \equiv (\nu n)Q$ and $Q \models \phi$
$P \models \phi \oslash n$ iff $(\nu n)P \models \phi$
$P \models m\langle n\rangle$ iff $P \equiv m\langle n\rangle$
$P \models \forall x.\phi$ iff for any $n \in \mathcal{N}$ we have $P \models \phi\{x \Leftarrow n\}$
$P \models \blacklozenge\phi$ iff there exists a process $Q$ with $P \longrightarrow Q$ and $Q \models \phi$
$P \models Nx.\phi$ iff $\exists n \in \mathcal{N} \setminus (fn(P) \cup fn(\phi))$ such that $P \models \phi\{x \Leftarrow n\}$

where we denoted by $fn(P)$ the set of the names that appears in the syntax of $P$ and are not bounded by an input or a new name operator, and by $fn(\phi)$ the set of all the names and free variables in the syntax of

[3]In asynchronous $\pi$-calculus the output capability $m\langle n\rangle$ cannot be succeeded by other processes but inactive one.

$\phi$. Observe the existential/universal character of the $Nx$ operator. Indeed, if the satisfiability requirements are fulfilled by one such name, then they are fulfilled by any name with the same property.

#### Ambient Logic

Ambient Logic [23, 25] was the first spatial logic proposed in the literature, being developed for Ambient Calculus [21]. With respect to the syntax presented before, this logic contains two more spatial operators: the location operator $n[\phi]$ and the placement operator $\phi@n$.

The formula $n[\phi]$, involving the *location operator*, is satisfied by the ambient process $n[P]$ when $P$ satisfies $\phi$.

Similarly, a formula involving the *placement operator*, $\phi@n$ is satisfied by a process $P$, which, if it is placed in the ambient $n$, will satisfy $\phi$. Hence the semantics of the two operators are the following:

$P \models n[\phi]$ iff $P \equiv n[Q]$ and $Q \models \phi$
$P \models \phi@n$ iff $n[P] \models \phi$

#### On decidability of Spatial Logics

From the beginning, the main reason for introducing spatial logics was to provide appropriate techniques for specification and model checking concurrent distributed systems, therefore most of the work done in this field points to the decidability problems.

*A model checker* is an algorithm that receives, as input, an initial state of a system and a property and provides, as output, the answer to the question if the given system, at the given state, has the property. If in the logic for which we projected the model-checking algorithm there are

temporal operators, such as *sometime in the future*, *always in the future* or *at the next state*, etc., then we can use it to verify the properties of our system in evolution.

Hence, to prove that model checking is decidable for a logic against the process semantics, we must prove that there exists a procedure that takes a process $P$ and a formula $\phi$ as input and provides, in finite time, as output, the yes/no answer to the question "*is it the case that* $P \models \phi$*?*".

Similarly, to prove that satisfiability is decidable, we have to prove that there is a finite algorithm taking a formula $\phi$ as input and giving, in a finite time, the yes/no answer to the question "exists a process $P$ such that $P \models \phi$?".

We will present further some decidability results that exist in the literature of spatial logics.

Decidability of model checking for the adjunct-free ambient logic against the replication free calculus was settled in [23] and extended to logics with constructs for restricted names and to the finite-control ambient calculus in [26].

Validity and model checking of ambient calculus against spatial logics with existential quantifiers was shown to be undecidable in [28].

Model checking the $\pi$-calculus against full adjunct-free spatial logic with behavioral modalities, hidden and fresh name quantifiers, and recursive operators was shown to be decidable in [12].

Decidability of validity in a static spatial logic for trees with adjunct operator was shown in [16].

One of the most clarifying results concerning the complexity and decidability of spatial logics is presented in [15] where a minimal spatial logic is considered against a small finite fragment of CCS. Thus this logics contains the basic spatial operators: void, composition and its adjunct, and

the next step modality. No quantifiers or operators involving names are considered. It is shown that this "*core logic*" can encode its own extension with quantification over actions, and modalities for actions, and that both model checking and satisfiability problems for it are undecidable.

The fragment of CCS to which we referred in the previous paragraph is the one generated by the next syntax, where $\alpha \in \mathbb{A}$ and these processes are collected in the set $\mathfrak{P}$:

$$P ::= 0 \mid \alpha.P \mid P|P$$

For it we consider two spatial logics:
$\mathcal{L}_{spat}$ given by the syntax

$$\phi ::= \top \mid 0 \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \phi_1|\phi_2 \mid \phi_1 \triangleright \phi_2 \mid \diamond\phi$$

$\mathcal{L}_{mod}$ given by the syntax

$$\phi ::= \top \mid 0 \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \phi_1|\phi_2 \mid \phi_1 \triangleright \phi_2 \mid \diamond\phi \mid \langle x\rangle\phi \mid \exists x.\phi$$

with the semantics already presented. In [15] it is proved that $\mathcal{L}_{spat}$ can encode $\mathcal{L}_{mod}$, hence they are equally expressive. Then it is proved that model-checking and validity/satisfiability checking for $\mathcal{L}_{spat}$ with respect to the finite fragment of CCS are both undecidable.

Concluding, the price of the expressivity of spatial logics is quite high: the basic spatial operators, in combination with temporal operators, generate undecidable logics [15, 28, 27], even against small finite pieces of CCS. This situation is induced mainly, by the presence of *the guarantee operator*, the adjunct of the parallel operator, having the semantics that involves a universal quantification over the class of processes. Indeed, while for the

static spatial logics (which are decidable [16]) the guarantee operator can be eliminated [48], in the dynamic spatial logics it has been proved [15] that the composition adjunct adds to the expressivity of the logic, and these logics are undecidable.

In the light of these results we have two alternatives: either we choose a static calculus, such as [16], which cannot describe our system in evolution, or we choose a dynamic one, but we have to avoid the use of the guarantee operator. This last alternative is useful only if our system is an isolated one and we have a full description of it, as in [12]. In this sense the possible applications are quite limited and, anyway, beyond the very first intentions of the spatial logics.

In problems such as those proposed by systems biology, for example, the systems we want to specify are always subsystems of bigger ones with which they interact. Very often we do not know too much about these upper systems, or we have only statistical information that shows how diverse the environment can be. In such situations we still need a spatial operator to act as universal quantifier, similar to the guarantee operator, but within the limits of decidability.

The aim of this thesis is to propose such an operator inspired by the epistemic logics.

## 2.3 Epistemic Logic

Epistemic logic was invented in the early 1960's, in top of the seminal work of Hintikka [43], as a tool for describing epistemic concepts such as *knowledge* and *belief* formally. At the beginning, the main interest was to

find inherent properties of knowledge (and related concepts) and to apply the analysis to epistemology. Thus, axioms for knowledge were suggested, attacked and defended. More recently, researchers from disciplines such as linguistics, economics, game theory, and computer science have become increasingly interested in reasoning about knowledge.

Within computer science, reasoning about knowledge plays an extremely important role in contemporary theories of (intelligent) agents.

In the transition from human agents to (artificial) intelligent agents and latter to the multi-agent system in the most general sense, the meaning of the term *"knowledge"* evolved. It was originally used in its ordinary language meaning: to say that an agent knows a sentence either means that it consciously assents to it, or that it immediately sees it to be true when the question is presented. Then, in the new interpretation, *the knowledge of the agent* is understood as the sum of actions the agent may take as a function of its local state in a given environment. Thus the agent knows its *protocol* in a given system. In this context we have an *external* notion of knowledge in the sense that there is no notion of the agent computing his knowledge and no requirement that the agent being able to answer questions based on his knowledge.

In addition to the more traditional topics, many other questions have become relevant for those who are more interested in applications, e.g., questions about computational complexities or the relationship between and agent's knowledge and his action. Also the multi-agent systems have been considered, this paradigm asking for a combination of epistemic and dynamic features [5, 3, 4, 6, 47, 70, 71, 72].

We present hereafter the main lines of the classic modal epistemic logic.

### The language of epistemic logic

Suppose that we have a group consisting of $n$ agents. Then we augment the language of propositional logic by $n$ knowledge operators $K_1, ...K_n$ (one for each agent), and form formulas in the obvious way. A statement like $K_1\phi$ is read *"agent* 1 *knows* $\phi$*"*[4]. The state that *"agent* 1 *knows that agent* 2 *knows* $\phi$*"* is formalized by $K_1K_2\phi$. A formula like $K_1\phi \wedge K_1(\phi \rightarrow \psi) \rightarrow K_1\psi$ is interpreted: *"if agent* 1 *knows* $\phi$ *and* $\phi \rightarrow \psi$ *then it knows* $\psi$*"*.

Formally, the language of modal epistemic logic is defined as follows:

**Definition 2.3.1** (The language of epistemic logic)**.** Let $\Phi$ be a nonempty, countable set of atomic formulae and $\mathfrak{S} = \{1, ...n\}$ a set of agents. We introduce the language of epistemic logic as the least set $\mathcal{F}^{\mathfrak{S}}$ of formulas such that:

- $Atom \subseteq \mathcal{F}^{\mathfrak{S}}$
- if $\phi \in \mathcal{F}^{\mathfrak{S}}$ then $\neg\phi \in \mathcal{F}^{\mathfrak{S}}$
- if $\phi, \psi \in \mathcal{F}^{\mathfrak{S}}$ then $\phi \wedge \psi \in \mathcal{F}^{\mathfrak{S}}$
- if $\alpha \in \mathcal{F}^{\mathfrak{S}}$ and $i \in \mathfrak{S}$ then $K_i\phi \in \mathcal{F}^{\mathfrak{S}}$

[4]The truth values of epistemic statements also depend on other parameters such as time, location, context. However, it is a common practice in epistemic logic to take only agents into consideration and to assume certain standard values for the other parameters, i.e., the sentences are interpreted relative to the "current" situation.

### Possible-worlds semantics for epistemic logic

One approach to defining semantics for epistemic logic is in terms of possible worlds. The intuitive idea behind the possible worlds approach is that an agent can build different models of the world using some suitable language. He usually does not know exactly which one of the models is the right model of the world. However, he does not consider all these models equally possible. Some world models are incompatible with his current information state, so he can exclude these incompatible models from the set of his possible world models. Only a subset of the set of all (logically) possible models are considered possible by the agent.

The set of worlds considered possible by an agent $i$ depends on the "actual world", or the agent's actual state of information. This dependency can be captured formally by introducing a binary relation, say $\mathcal{R}_i$, on the set of possible worlds. To express the idea that for agent $i$, the world $t$ is compatible with his information state when he is in the world $s$, we require that the relation $\mathcal{R}_i$ holds between $s$ and $t$. One says that $t$ is an epistemic alternative to $s$ (for agent $i$). If a sentence $\phi$ is true in all worlds which agent $i$ considers possible then we say that this agent knows $\phi$.

Formally, the concept of models is defined in terms of Kripke structures, as follows:

**Definition 2.3.2.** A model $\mathcal{M}$ for the language $\mathcal{F}^{\mathfrak{S}}$ is a Kripke structure for the agents in $\mathfrak{S}$ over $\Phi$, i.e. is a structure $\mathcal{M} = (S, \pi, (\mathcal{R}_i)_{i\in\mathfrak{S}})$ where

- $S$ is a nonempty set of possible worlds (states)
- $\pi$ is an *interpretation* which associates with each state in $S$ a truth assignment to the primitive propositions in $\Phi$ (i.e. for $s \in S$, $\pi(s) : \Phi \to \{\top, \bot\}$)

- $\mathcal{R}_i$ is a binary relation on $S$ associated to the agent $i \in \mathfrak{S}$

The satisfaction relation $\models$ is defined recursively on $\mathcal{F}^{\mathfrak{S}}$ as follows:

- $\mathcal{M}, s \models p$ iff $\pi(s)(p) = \top$ for any $p \in \Phi$
- $\mathcal{M}, s \models \neg\phi$ iff $\mathcal{M}, s \not\models \phi$
- $\mathcal{M}, s \models \phi \wedge \psi$ iff $\mathcal{M}, s \models \phi$ and $\mathcal{M}, s \models \psi$
- $\mathcal{M}, s \models K_i\phi$ iff for all $t \in S$ such that $s\mathcal{R}_i t$ we have $\mathcal{M}, t \models \phi$

### Axioms for modal epistemic logic

A modal epistemic logic for the agents in $\mathfrak{S}$ is obtained by joining together $n$ modal logics [8], one for each agent in $\mathfrak{S}$. It is usually assumed that the agents are homogeneous, i.e., they can be described by the same logic. So an epistemic logic for $n$ agents consists of $n$ copies of a certain modal logic. Such a system over $\mathfrak{S}$ will be denoted by the same name as the modal system, but with the superscript $\mathfrak{S}$.

**Definition 2.3.3** (Modal epistemic logic $K^{\mathfrak{S}}$)**.** The modal epistemic logic $K^{\mathfrak{S}}$ is the logic specified by the following axioms and rules of inference, where $i \in \mathfrak{S}$:

**(PC):** All propositional tautologies.

**(K):** $\vdash K_i\phi \wedge K_i(\phi \to \psi) \to K_i\psi$

**(MP):** Modus ponens: if $\vdash \phi$ and $\vdash \phi \to \psi$ then $\vdash \psi$

**(NEC):** Necessity: if $\vdash \phi$ then $\vdash K_i\phi$

Stronger logics can be obtained by adding additional principles, which express the desirable properties of the concept of knowledge, to the basic system $K^{\mathfrak{S}}$. The following properties are often considered:

**(T):** Knowledge axiom: $\vdash K_i\phi \rightarrow \phi$
**(D):** Consistency axiom: $\vdash K_i\phi \rightarrow \neg K_i\neg\phi$
**(4):** Positive introspection: $\vdash K_i\phi \rightarrow K_iK_i\phi$
**(5):** Negative introspection: $\vdash \neg K_i\phi \rightarrow K_i\neg K_i\phi$

The formula (T) states that knowledge must be true. In the doxastic logic this axiom is taken to be the major one distinguishing knowledge from belief. For that reason (T) is called the Knowledge Axiom or the Truth Axiom (for knowledge). Systems containing the schema (T) (such as **S4** and **S5**) are then called logics of knowledge, and logics without the schema (T) are called logics of belief.

The property (D), called the Consistency Axiom, requires that agents be consistent in their knowledge: they do not know both a formula and its negation. Generally, (D) is a weaker condition than (T).

The properties (4) and (5) are called positive and negative introspection axioms, respectively. They say that an agent is aware of what he knows and what he does not know. Their converses, i.e., the formulae $\vdash K_iK_i\phi \rightarrow K_i\phi$ and $\vdash K_i\neg K_i\phi \rightarrow \neg K_i\phi$, are instances of the schema (T). Taking (4) and (5) together with their converses we have $\vdash K_iK_i\phi \leftrightarrow K_i\phi$ and $\vdash K_i\neg K_i\phi \leftrightarrow \neg K_i\phi$, which allow to reduce multiple knowledge operators to a single (positive or negative) knowledge operator.

The commonly used epistemic logics are specified as follows:

- $T^{\mathfrak{S}}$ is $K^{\mathfrak{S}}$ plus (T)

- $S4^{\mathfrak{S}}$ is $T^{\mathfrak{S}}$ plus (4)
- $S5^{\mathfrak{S}}$ is $S4^{\mathfrak{S}}$ plus (5)
- $KD^{\mathfrak{S}}$ is $K^{\mathfrak{S}}$ plus (D)
- $KD4^{\mathfrak{S}}$ is $KD^{\mathfrak{S}}$ plus (4)
- $KD45^{\mathfrak{S}}$ is $KD4^{\mathfrak{S}}$ plus (5)

The following theorem summarizes some completeness and decidability results for modal epistemic logic [29, 46, 39, 41].

**Theorem 2.3.1.** *1. $K^{\mathfrak{S}}$ is determined by the class of all models with accessibility relations indexed by elements in $\mathfrak{S}$.*

2. *$T^{\mathfrak{S}}$ is determined by the class of models with reflexive accessibility relations.*
3. *$S4^{\mathfrak{S}}$ is determined by the class of models with reflexive and transitive accessibility relations.*
4. *$S5^{\mathfrak{S}}$ is determined by the class of models with equivalence relations as accessibility relations.*
5. *$KD^{\mathfrak{S}}$ is determined by the class of models with serial accessibility relations.*
6. *$KD4^{\mathfrak{S}}$ is determined by the class of models with serial and transitive accessibility relations.*
7. *$KD45^{\mathfrak{S}}$ is determined by the class of models with serial, transitive and Euclidean accessibility relations.*

*8.* $K^{\mathfrak{S}}$, $T^{\mathfrak{S}}$, $S4^{\mathfrak{S}}$, $S5^{\mathfrak{S}}$, $KD^{\mathfrak{S}}$, $KD4^{\mathfrak{S}}$, *and* $KD45^{\mathfrak{S}}$ *are all decidable.*

# Chapter 3

# The joy of processes

In this chapter we return to CCS and we consider a subcalculus for which we will construct our logics. We propose some new concepts that will help the future constructs. One of the most important is a new congruence on processes - *the structural bisimulation*. This relation will be used, further, to prove the finite model property for our logics against the process semantics in combination with the concept of *pruning processes*.

The structural bisimulation is interesting in itself as it provides a bisimulation-like definition for structural congruence. Informally, it is an approximation of the structural congruence bounded by two sizes: the *height* (the depth of the syntactic tree) and the *weight* (the maximum number of bisimilar subprocesses that can be found in a node of the syntactic tree) of a process. The bigger these sizes, the better approximation we obtain. At the limit, for sizes big enough, we find exactly the structural congruence.

To the best of our knowledge, a similar relation, which to provide a bisimulation-like description of structural congruence, has not been proposed for process calculi. A conceptually similar congruence was proposed in [16] for analyzing trees of location for the static ambient calculus.

On the two sizes defined for processes, *height* and *weight*, we will introduce an effective method to construct, given process $P$, a minimal process

$Q$ that has an established size $(h, w)$ and is structurally bisimilar to $P$ on this size. Because, for a small size, the construction is supposed to prune the syntactic tree of $P$, we will call this method *pruning*, and we refer to $Q$ as *the pruned of $P$ on the size* $(h, w)$.

Eventually we will extend the notions of *size*, *structural bisimulation* and *pruning* from processes to classes of processes. We focus our interest on *contexts*, defined as being special classes of processes that contain, in a maximal manner, processes of interest for us (that might model completely or partially our system together with all its subsystems). The contexts will be used, in the next chapters, as the sets of processes on which we will define the satisfiability relation for the logics.

We begin by introducing the subcalculus of CCS on which we will focus for the rest of the paper.

**Definition 3.0.4.** Let $\mathbb{A}$ be a finite class of actions called *basic actions*. The class $\mathfrak{P}$ of processes is given by the following grammar

$$P := \ 0 \mid P|P \mid \alpha.P, \text{ where } \alpha \in \mathbb{A}.$$

Hereafter we use $\alpha$ to range over $\mathbb{A}$.

**Definition 3.0.5** (Subprocesses)**.** We define the *subprocesses* of a given process $P$ as being the elements of the set $sub(P)$ inductively defined by:

1. $sub(0) \stackrel{def}{=} \{0\}$
2. $sub(P|Q) \stackrel{def}{=} \{P|Q\} \cup sub(P) \cup sub(Q)$
3. $sub(\alpha.P) \stackrel{def}{=} \{\alpha.P\} \cup sub(P)$

We will say that the process $P$ is an *active subprocess* of $Q$ iff for some process $R$ we have $Q \equiv P|R$.

**Definition 3.0.6.** We call a process $P$ *guarded* iff $P \equiv \alpha.Q$ for $\alpha \in \mathbb{A}$. We introduce the notation $P^k \stackrel{def}{=} \underbrace{P|...|P}_{k}$, and convey to denote $P^0 \equiv 0$.

We will not consider additional features for the actions, such as pairs of names, etc., as we want to avoid all the syntactic sugar that is irrelevant from the point of view of the logic. We might define an involution on $\mathbb{A}$ and the null action, but all these will be irrelevant for our logic, as a communication, in a dynamic logic, can be introduced as a derived operator:

$$\langle \alpha, \overline{\alpha} \rangle \phi \stackrel{def}{=} \bigvee_{\phi \leftrightarrow \phi_1|\phi_2} \langle \alpha \rangle \phi_1 | \langle \overline{\alpha} \rangle \phi_2$$

where we denoted by $(\alpha, \overline{\alpha})$ the communication action and the disjunction is considered up to logically-equivalent decompositions $\phi \leftrightarrow \phi_1|\phi_2$ that ensures the use of a finitary formula.

## 3.1 Canonical representation of processes

**Assumption** (Representativeness modulo structural congruence)**.** *By definition, $\equiv$ is a congruence (thence an equivalence relation) over $\mathfrak{P}$. Consequently, we convey to identify processes up to structural congruence, because the structural congruence is the ultimate level of expressivity we want for our logic. Hereafter in the paper, if it is not explicitly otherwise stated, we will speak about processes up to structural congruence.*

Being the previous assumption, it is useful to define a *canonical representative* for each class of structurally congruent processes. We will do this by assuming a lexicographical order on $\mathbb{A}$.

**Definition 3.1.1** (Canonical representation of a process)**.** Suppose that we are given a lexicographical order, $\ll$, over the set of actions $\mathbb{A}$ that syntactically induces a order $\ll$ on processes. We introduce the *canonical representation of the process* $P$ as being the process $\overline{P}$ inductively defined by:

- if $P \equiv 0$ then $\overline{P} \stackrel{def}{=} 0$
- if $P \equiv Q|0$ then $\overline{P} \stackrel{def}{=} \overline{Q}$
- if $P \equiv (\alpha_1.P_1)^{k_1}|(\alpha_2.P_2)^{k_2}|...|(\alpha_n.P_n)^{k_n}$ with $k_i \neq 0$, $\alpha_i < \alpha_j$ for $i < j$ or if $\alpha_i = \alpha_j$ then $P_i \ll P_j$, and for $i \neq j$, $\alpha_i.P_i \not\equiv \alpha_j.P_j$, then

$$\overline{P} \stackrel{def}{=} (\alpha_1.\overline{P_1})^{k_1}|(\alpha_2.\overline{P_2})^{k_2}|...|(\alpha_n.\overline{P_n})^{k_n}$$

**Example 3.1.1.** *Consider $\alpha, \beta, \gamma \in \mathbb{A}$ with $\beta < \gamma$ and the process*

$$P = \alpha.(\beta.0|0)|\alpha.(0|\beta.(0|0))|\alpha.(\beta.0|\gamma.0)|\alpha.\beta.0|\alpha.(\gamma.0|\beta.0).$$

*Because $\beta.0|0 \equiv \beta.0$, we obtain that $\overline{\beta.0|0} = \beta.\overline{0} = \beta.0$;*
*the same about $0|\beta.(0|0) \equiv \beta.(0|0)$, thus $\overline{0|\beta.(0|0)} = \beta.\overline{(0|0)} = \beta.0$; because $\beta < \gamma$ and $\gamma.0|\beta.0 \equiv \beta.0|\gamma.0$ we obtain $\overline{\beta.0|\gamma.0} = \overline{\gamma.0|\beta.0} = \gamma.0|\beta.0$.*
*Thence we will obtain*

$$\overline{P} = (\alpha.\beta.0)^3|(\alpha.(\beta.0|\gamma.0))^2$$

*Remark* 3.1.1. Observe that, by construction, $\overline{P} \equiv P$ and the canonical representation of a process is unique for each class of structurally congruent processes. For this reason, hereafter, if needed in some proofs, we will safely replace a process by its canonical representative.

## 3.2 Size of a process

Further we propose a definition for the *size of a process*, following a similar idea developed in [16] for sizes of trees. The intuition is that the process has a *height* given by the vertical size of its syntactic tree, and a *width* equal to the maximum number of bisimilar subprocesses that can be identified in a node of the syntactic tree.

**Definition 3.2.1** (Size of a process)**.** We define *the size (height and width) of a process* $P$, denoted by $[\![P]\!]$, by:

- $[\![0]\!] \stackrel{def}{=} (0,0)$
- $[\![P]\!] \stackrel{def}{=} (h,w)$ iff
  - $\overline{P} = (\alpha_1.Q_1)^{k_1}|(\alpha_2.Q_2)^{k_2}|...|(\alpha_j.Q_j)^{k_j}$ and $[\![Q_i]\!] = (h_i, w_i)$, $i \in 1..j$
  - $h = 1 + max(h_1, ..., h_k)$, $w = max(k_1, ..., k_j, w_1, ..., w_j)$

where we used $h$ for *height* and $w$ for *width*. We convey to write $(h_1, w_1) \leq (h_2, w_2)$ for $h_1 \leq h_2$ and $w_1 \leq w_2$ and $(h_1, w_1) < (h_2, w_2)$ for $h_1 < h_2$ and $w_1 < w_2$.

**Example 3.2.1.** *We show further the size for some processes:*

- $[\![0]\!] = (0,0)$
- $[\![\alpha.0]\!] = (1,1)$

- $[\![\alpha.0|\beta.0]\!] = (1,1)$
- $[\![\alpha.0|\alpha.0]\!] = (1,2)$
- $[\![\alpha.\alpha.0]\!] = [\![\alpha.\beta.0]\!] = (2,1)$
- $[\![\alpha.(\beta.0|\beta.0)]\!] = (2,2)$

*Remark* 3.2.1. Observe that, by construction, the size of a process is unique up to structural congruence, as the canonical representation of it is unique. Moreover, if $[\![P]\!] = (h,w)$ then for any subprocess $P'$ of $P$ we have $[\![P']\!] \leq (h,w)$.

## 3.3 Structural bisimulation

In this section we introduce the *structural bisimulation*, a congruence relation on processes bounded by size. It analyzes the behavior of a process focusing on a boundary of its syntactic tree. This relation will be used in the next chapter to prove the finite model property for our logics.

The intuition behind the structural bisimulation is that $P \approx_h^w Q$ ($P$ and $Q$ are structurally bisimilar on size $(h,w)$) iff when we consider for both processes their syntactic trees up to the depth $h$ only (we prune them on the height $h$) and we ignore the presence of more than $w$ parallel bisimilar subprocesses in any node of the syntactic trees (we prune the trees on weight $w$), we obtain syntactic trees depicting two structurally congruent processes.

The relation between the structural bisimulation and the structural congruence is interesting. We will see that the structural bisimulation depicts,

step by step, the structural congruence being, in a sense, a bisimulation-like approximation of it on a given size. We will see further how $P \approx_h^w Q$ entails that, if we choose any subprocess of $P$ with the size smaller than $(h, w)$, then there exists a subprocess of $Q$ structurally congruent with it, and vice versa. Now, if the size indexing the structural bisimulation is bigger than the size of the processes, then our relation will describe structurally congruent processes. Moreover, the structural bisimulation is preserved by transitions with the price of decreasing the size.

**Definition 3.3.1** (Structural bisimulation)**.** Let $P, Q$ be any processes. We define $P \approx_h^w Q$ by:

- $P \approx_0^w Q$ always
- $P \approx_{h+1}^w Q$ iff for any $i \in 1..w$ and any $\alpha \in \mathbb{A}$ we have
  - if $P \equiv \alpha.P_1|...|\alpha.P_i|P'$ then $Q \equiv \alpha.Q_1|...|\alpha.Q_i|Q'$ with $P_j \approx_h^w Q_j$, for $j = 1..i$
  - if $Q \equiv \alpha.Q_1|...|\alpha.Q_i|Q'$ then $P \equiv \alpha.P_1|...|\alpha.P_i|P'$ with $Q_j \approx_h^w P_j$, for $j = 1..i$

**Example 3.3.1.** *Consider the processes*

$$P \equiv \alpha.\beta.0 \text{ and } Q \equiv \alpha.\beta.\gamma.0.$$

*Verifying the definition 3.3.1 we obtain $P \approx_0^w Q$, $P \approx_1^w Q$, $P \approx_2^w Q$ but for a bigger height we obtain $P \not\approx_3^w Q$. This is so because, going on the syntactic trees on the depth 2, $Q$ can perform $\gamma$ (in figure 3.1 marked with dashed arrow) while $P$ cannot.*

*Consider now the processes*

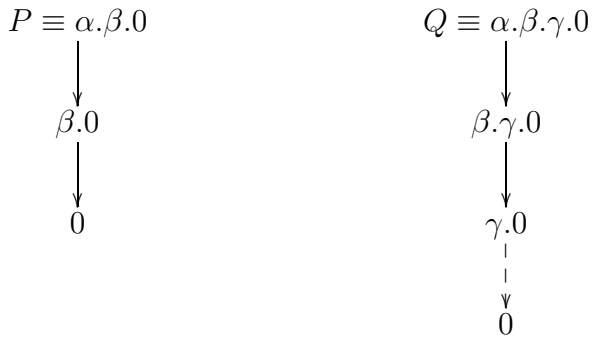


Figure 3.1:

$$R \equiv \alpha.(\beta.0|\beta.0|\beta.0)|\alpha.\beta.0 \text{ and } S \equiv \alpha.(\beta.0|\beta.0)|\alpha.\beta.\alpha.0$$

*We can verify the requirements of the definition 3.3.1 and decide that* $R \approx_2^2 S$ *but* $R \not\approx_3^2 S$ *because on the depth 2* $S$ *can perform* $\alpha$ *(in figure 3.2 marked with a dashed arrow) while* $R$ *cannot (because the height of* $R$ *is only 2). Also* $R \not\approx_2^3 S$ *because* $S$ *contains only 2 bisimilar copies of* $\beta.0$ *on depth 2, while* $R$ *contains 3 (the extra one is marked with a dashed arrow).*

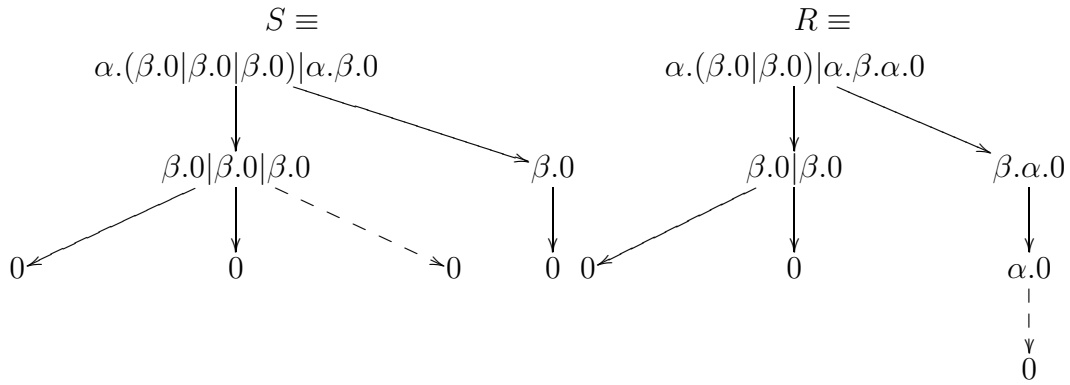


Figure 3.2:

*Hence, for any weight bigger than 2 this feature will show the two processes as different. But if we remain on depth 1 we have* $R \approx_1^3 S$*, as on this deep the two processes have the same number of bisimilar subprocesses, i.e. any of them can perform* $\alpha$ *in two ways giving, further, processes in*

*the relation* $\approx_0^3$. *Indeed*

$$R \equiv \alpha R' | \alpha R'', \text{ where } R' \equiv \beta.0|\beta.0|\beta.0 \text{ and } R'' \equiv \beta.0$$
$$S \equiv \alpha.S' | \alpha.S'', \text{ where } S' \equiv \beta.0|\beta.0 \text{ and } S'' \equiv \beta.\alpha.0$$

*By definition,* $R' \approx_0^3 S'$ *and* $R'' \approx_0^3 S''$

We focus further on the properties of the relation $\approx_h^w$. We start by proving that structural bisimulation is a congruence relation.

**Theorem 3.3.1** (Equivalence Relation)**.** *The relation* $\approx_h^w$ *on processes is an equivalence relation.*

*Proof.* We verify the reflexivity, symmetry and transitivity directly.

**Reflexivity:** $P \approx_h^w P$ - we prove it by induction on $h$

**the case** $h = 0$: we have $P \approx_0^w P$ from the definition 3.3.1.

**the case** $h + 1$: suppose that $P \equiv \alpha.P_1|...|\alpha.P_i|P'$ for $i \in 1..w$ and some $\alpha \in \mathbb{A}$. The inductive hypotheses gives $P_j \approx_h^w P_j$ for each $j = 1..i$. Further we obtain, by the definition 3.3.1, that $P \approx_h^w P$.

**Symmetry:** if $P \approx_h^w Q$ then $Q \approx_h^w P$

Suppose that $P \equiv \alpha.P_1|...|\alpha.P_i|P'$ for some $i \in 1..w$ and $\alpha \in \mathbb{A}$ then, by the definition 3.3.1, exists $Q \equiv \alpha.Q_1|...|\alpha.Q_i|Q'$ with $P_j \approx_{h-1}^w Q_j$ for $j = 1..i$ and vice versa. Similarly, if we start from $Q \equiv \beta.R_1|...|\beta.R_k|R'$ for $k \in 1..w$ and $\beta \in \mathbb{A}$ we obtain $P \equiv \beta.S_1|...|\beta.S_k|S'$ for some $S_j$, with $R_j \approx_{h-1}^w S_j$ for $j = 1..k$ and vice versa. Hence $Q \approx_h^w P$.

**Transitivity:** if $P \approx_h^w Q$ and $Q \approx_h^w R$ then $P \approx_h^w R$ - we prove it by induction on $h$.

**the case** $h = 0$ is trivial, because by the definition 3.3.1, for any two processes $P, R$ we have $P \approx_0^w R$

**the case** $h + 1$: suppose that $P \equiv \alpha.P_1|...|\alpha.P_i|P'$ for some $i \in 1..w$

and $\alpha \in \mathbb{A}$. Then from $P \approx_h^w Q$ we obtain, by the definition 3.3.1, that $Q \equiv \alpha.Q_1|...|\alpha.Q_i|Q'$ with $P_j \approx_{h-1}^w Q_j$ for $j = 1..i$ and vice versa. Further, because $Q \approx_h^w R$, we obtain that $R \equiv \alpha.R_1|...|\alpha.R_i|R'$ with $Q_j \approx_{h-1}^w R_j$ for $j = 1..i$ and vice versa.

As $P_j \approx_{h-1}^w Q_j$ and $Q_j \approx_{h-1}^w R_j$ for $j = 1..i$, we obtain, using the inductive hypothesis that $P_j \approx_{h-1}^w R_j$ for $j = 1..i$.

Hence, for $P \equiv \alpha.P_1|...|\alpha.P_i|P'$, some $i \in 1..w$ and $\alpha \in \mathbb{A}$ we have that $R \equiv \alpha.R_1|...|\alpha.R_i|R'$ with $Q_j \approx_{h-1}^w R_j$ for $j = 1..i$ and vice versa. This entails $P \approx_h^w R$. □

**Theorem 3.3.2.** *If $P \approx_h^w Q$ and $Q \equiv R$ then $P \approx_h^w R$.*

*Proof.* Suppose that $P \equiv \alpha.P_1|...|\alpha.P_i|P'$ for some $i \in 1..w$ and $\alpha \in \mathbb{A}$. As $P \approx_h^w Q$, we obtain $Q \equiv \alpha.Q_1|...|\alpha.Q_i|Q'$ with $P_j \approx_{h-1}^w Q_j$ for $j = 1..i$ and vice versa. But $Q \equiv R$, so $R \equiv \alpha.Q_1|...|\alpha.Q_i|Q'$ with $P_j \approx_{h-1}^w Q_j$ for $j = 1..i$ and vice versa. Hence $P \approx_h^w R$. □

**Theorem 3.3.3** (Antimonotonicity)**.** *If $P \approx_h^w Q$ and $(h', w') \leq (h, w)$ then $P \approx_{h'}^{w'} Q$.*

*Proof.* We prove it by induction on $h$.

**The case** $h = 0$ is trivial, as $(h', w') \leq (0, w)$ gives $h' = 0$ and for any processes $P, Q$ we have $P \approx_0^w Q$.

**The case** $h + 1$ in the context of the inductive hypothesis:
Suppose that $P \approx_{h+1}^w Q$ and $(h', w') \leq (h + 1, w)$.
If $h' = 0$ we are, again, in a trivial case as for any two processes $P, Q$ we have $P \approx_0^w Q$.
If $h' = h'' + 1$ then consider any $i \in 1..w'$, and any $\alpha \in \mathbb{A}$ such that

$P \equiv \alpha.P_1|...|\alpha.P_i|P'$. Because $i \leq w' \leq w$, and as $P \approx^w_{h+1} Q$, we have $Q \equiv \alpha.Q_1|...|\alpha_i.Q_i|Q'$ with $P_j \approx^w_h Q_j$, for $j = 1..i$. A similar argument can de developed if we start the analysis from $Q$.
But $(h'', w') \leq (h, w)$, so we can use the inductive hypothesis that gives $P_j \approx^{w'}_{h''} Q_j$ for $j = 1..i$. Hence $P \approx^{w'}_{h''+1} Q$, that is, $P \approx^{w'}_{h'} Q$ q.e.d. □

**Theorem 3.3.4** (Congruence)**.** *The following holds:*

1. *if* $P \approx^w_h Q$ *then* $\alpha.P \approx^w_{h+1} \alpha.Q$
2. *if* $P \approx^w_h P'$ *and* $Q \approx^w_h Q'$ *then* $P|Q \approx^w_h P'|Q'$

*Proof.* 1.: Suppose that $P \approx^w_h Q$. Because $\alpha.P$ is guarded, it cannot be represented as $P \equiv \alpha.P'|P''$ for $P'' \not\equiv 0$. The same about $\alpha.Q$. But this observation, together with $P \approx^w_h Q$ gives, in the light of definition 3.3.1, $\alpha.P \approx^w_{h+1} \alpha.Q$.

2.: We prove it by induction on $h$.

**If** $h = 0$ then the conclusion is immediate.

**For** $h + 1$, suppose that $P \approx^w_{h+1} P'$ and $Q \approx^w_{h+1} Q'$; then consider any $i = 1..w$, $\alpha$ and $R_j$ for $j = 1..i$ such that

$$P|Q \equiv \alpha.R_1|...|\alpha.R_i|R_{i+1}$$

Suppose, without loss of generality, that $R_j$ are ordered in such a way that there exist $k \in 1..i$, $P'', Q''$ such that

$$P \equiv \alpha.R_1|...|\alpha.R_k|P''$$
$$Q \equiv \alpha.R_{k+1}|...|\alpha.R_i|Q''$$
$$R_{i+1} \equiv P''|Q''$$

Because $k \in 1..w$, from $P \approx^w_{h+1} P'$ we have $P' \equiv \alpha.P'_1|...|\alpha.P'_k|P_0$ such that $R_j \approx^w_h P'_j$ for $j = 1..k$.

Similarly, from $Q \approx^w_{h+1} Q'$ we have $Q' \equiv \alpha.Q'_{k+1}|...|\alpha.Q'_i|Q_0$ such that $Rj \approx^w_h Q'_j$ for $j = (k+1)..i$. Hence, we have

$$P'|Q' \equiv \alpha.P'_1|...|\alpha.P'_k|\alpha.Q'_{k+1}|...|\alpha.Q'_i|P_0|Q_0$$

As $R_j \approx^w_h P'_j$ for $j = 1..k$ and $R_j \approx^w_h Q'_j$ for $j = (k+1)..i$, and because a similar argument starting from $P'|Q'$ is possible, we proved that $P|Q \approx^w_{h+1} P'|Q'$. □

**Theorem 3.3.5** (Inversion)**.** *If $P'|P'' \approx^{w_1+w_2}_h Q$ then exists $Q', Q''$ such that $Q \equiv Q'|Q''$ and $P' \approx^{w_1}_h Q'$, $P'' \approx^{w_2}_h Q''$.*

*Proof.* Let $w = w_1 + w_2$. We prove the theorem by induction on $h$:
**The case** $h = 0$**:** is trivial.
**The case** $h + 1$**:** Suppose that $P'|P'' \approx^w_{h+1} Q$.

Consider the following definition: a process $P$ is in $(h, w)$-*normal form* if whenever $P \equiv \alpha_1.P_1|\alpha_2.P_2|P_3$ and $P_1 \approx^w_h P_2$ then $P_1 \equiv P_2$. Note that $P \approx^w_{h+1} \alpha_1.P_1|\alpha_2.P_1|P_3$. This shows that for any $P$ and any $(h, w)$ we can find a $P_0$ such that $P_0$ is in $(h, w)$-normal form and $P \approx^w_{h+1} P_0$.

Now, we can suppose, without loosing generality, that the canonical representations of $P', P''$ and $Q$ are[1]:

$$P' \equiv (\alpha_1.P_1)^{k'_1}|...|(\alpha_n.P_n)^{k'_n}$$
$$P'' \equiv (\alpha_1.P_1)^{k''_1}|...|(\alpha_n.P_n)^{k''_n}$$
$$Q \equiv (\alpha_1.P_1)^{l_1}|...|(\alpha_n.P_n)^{l_n}$$

For each $i \in 1..n$ we split $l_i = l'_i + l''_i$ in order to obtain a splitting of $Q$. We define the splitting of $l_i$ such that $(\alpha_i.P_i)^{k'_i} \approx_{h+1,w_1} (\alpha_i.P_i)^{l'_i}$ and $(\alpha_i.P_i)^{k''_i} \approx_{h+1,w_2} (\alpha_i.P_i)^{l''_i}$. We do this as follows:

[1]Else we can replace $P', P''$ with $(h+1, w)$-related processes having the same $(h, w)$-normal forms

- if $k'_i + k''_i < w_1 + w_2$ then $P'|P'' \approx^w_{h+1} Q$ implies $l_i = k'_i + k''_i$, so we can choose $l'_i = k'_i$ and $l''_i = k''_i$.
- if $k'_i + k''_i \geq w_1 + w_2$ then $P'|P'' \approx^w_{h+1} Q$ implies $l_i \geq w_1 + w_2$. We meet the following subcases:
  - $k'_i \geq w_1$ and $k''_i \geq w_2$. We choose $l'_i = w_1$ and $l''_i = l_i - w_1$ (note that as $l_i \geq w_1 + w_2$, we have $l''_i \geq w_2$).
  - $k'_i < w_1$, then we must have $k''_i \geq w_2$. We choose $l'_i = k'_i$ and $l''_i = l_i - k'_i$. So $l''_i \geq w_2$ as $l_i \geq w_1 + w_2$ and $l'_i < w_1$.
  - $k''_i < w_2$ is similar with the previous one. We choose $l''_i = k''_i$ and $l'_i = l_i - k''_i$.

Now for $Q' \equiv (\alpha_1.P_1)^{l'_1}|...|(\alpha_n.P_n)^{l'_n}$ and $Q'' \equiv (\alpha_1.P_1)^{l''_1}|...|(\alpha_n.P_n)^{l''_n}$ the theorem is verified by repeatedly using theorem 3.3.4. □

The next theorems point out the relation between the structural bisimulation and the structural congruence. We will prove that for a well-chosen boundary, which depends on the processes involved, the structural bisimulation guarantees the structural congruence. $P \approx^w_h Q$ entails that if we choose any subprocess of $P$ having the size smaller than $(h, w)$, we will find a subprocess of $Q$ structurally congruent with it, and vice versa. Now, if the size indexing the structural bisimulation is bigger than the size of the processes, then our relation will describe structurally congruent processes. We also prove that the structural bisimulation is preserved by transitions with the price of decreasing the size.

**Theorem 3.3.6.** *If* $[\![P]\!] \leq (h, w)$ *and* $[\![P']\!] \leq (h, w)$ *then* $P \approx^w_h P'$ *iff* $P \equiv P'$.

*Proof.* $P \equiv P'$ implies $P \approx_h^w P'$, because by reflexivity $P \approx_h^w P$ and then we can apply theorem 3.3.2.
We prove further that $P \approx_h^w P'$ implies $P \equiv P'$. We'll do it by induction on $h$.
**The case** $h = 0$**:** $[\![P]\!] \leq (0, w)$ and $[\![P']\!] \leq (0, w)$ means $P \equiv 0$ and $P' \equiv 0$, hence $P \equiv P'$.
**The case** $h + 1$**:** suppose that $[\![P]\!] \leq (h + 1, w)$, $[\![P']\!] \leq (h + 1, w)$ and $P \approx_{h+1}^w P'$. We can suppose, without loosing generality[2], that

$$P \equiv (\alpha_1.Q_1)^{k_1}|...|(\alpha_n.Q_n)^{k_n}$$
$$P' \equiv (\alpha_1.Q_1)^{l_1}|...|(\alpha_n.Q_n)^{l_n}$$

where for $i \neq j$, $\alpha_i.Q_i \not\equiv \alpha_j.Q_j$. Obviously, as $[\![P]\!] \leq (h + 1, w)$ and $[\![P']\!] \leq (h + 1, w)$ we have $k_i \leq w$ and $l_i \leq w$.

We show that $k_i \leq l_i$. If $k_i = 0$ then, obviously, $k_i \leq l_i$. If $k_i \neq 0$ then $P \equiv (\alpha_i.Q_i)^{k_i}|P_i$ and $P \approx_{h+1}^w P'$ provides that $P' \equiv \alpha_i.Q_1''|...\alpha_i.Q_{k_i}''|R$ with $Q_i \approx_h^w Q_j''$ for $j = 1..k_i$. By construction, $[\![Q_i]\!] \leq ((h + 1) - 1, w) = (h, w)$ and $[\![Q_j'']\!] \leq ((h + 1) - 1, w) = (h, w)$. So, we can apply the inductive hypothesis that provides $Q_i \equiv Q_j''$ for $j = 1..i$. Hence $P' \equiv (\alpha_i.Q_i)^{k_i}|R$ that gives $k_i \leq l_i$.

With a symmetrical argument we can prove that $l_i \leq k_i$ that gives $k_i = l_i$ and, finally, $P \equiv P'$. □

**Theorem 3.3.7.** *If* $P \approx_h^w Q$ *and* $[\![P]\!] < (h, w)$ *then* $P \equiv Q$.

*Proof.* Suppose that $[\![P]\!] = (h', w')$ and $P \equiv (\alpha_1.P_1)^{k_1}|...|(\alpha_n.P_n)^{k_n}$ with $\alpha_i.P_i \not\equiv \alpha_j.P_j$ for $i \neq j$. Obviously we have $k_i \leq w' < w$.

We prove the theorem by induction on $h$. The first case is $h = 1$ (because $h > h'$).

[2]We obtain such representation by considering the canonical representation of each process augmented, if necessary, with parallel components having $k_i = 0$

**The case** $h = 1$**:** we have $h' = 0$ that gives $P \equiv 0$. Further $0 \approx_1^w Q$ gives $Q \equiv 0$, because else $Q \equiv \alpha.Q'|Q''$ asks for $0 \equiv \alpha.P'|P''$ - impossible. Hence $P \equiv Q \equiv 0$.

**The case** $h+1$**:** as $P \equiv (\alpha_i.P_i)^{k_i}|P^+$, $P \approx_h^w Q$ and $k_i < w$, we obtain that $Q \equiv \alpha_i.R_1|...|\alpha_i.R_{k_i}|R^+$ with $P_i \approx_{h-1}^w R_j$ for any $j = 1..k_i$.

But $P_i \approx_{h-1}^w R_j$ allows us to use the inductive hypothesis, because $[\![P_i]\!] \leq (h' - 1, w') < (h - 1, w)$, that gives $P_i \equiv R_j$ for any $j = 1..k_i$. Hence $Q \equiv (\alpha_i.P_i)^{k_i}|R^+$ and this is sustained for each $i = 1..n$. As $\alpha_i.P_i \not\equiv \alpha_j.P_j$ for $i \neq j$, we derive $Q \equiv (\alpha_1.P_1)^{k_1}|...|(\alpha_n.P_n)^{k_n}|R$.

We prove now that $R \equiv 0$. Suppose that $R \equiv (\alpha.R')|R''$. Then $Q \equiv \alpha.R'|R^-$, and as $P \approx_h^w Q$, we obtain that there is an $i = 1..n$ such that $\alpha = \alpha_i$ and $R' \approx_{h-1,w} P_i$.

Because $[\![P_i]\!] \leq (h'-1, w') < (h-1, w)$, we can use the inductive hypothesis and obtain $R' \equiv P_i$. Therefore $R \equiv \alpha_i.P_i|R''$, that gives further

$$Q \equiv (\alpha_1.P_1)^{k_1}|...(\alpha_{i-1}.P_{i-1})^{k_{(i-1)}}|(\alpha_i.P_i)^{k_i+1}|(\alpha_{i+1}.P_{i+1})^{k_{(i+1)}}|...|(\alpha_n.P_n)^{k_n}|R$$

So, we can consider $Q \equiv (\alpha_i.P_i)^{k_i+1}|Q^+$. Because $P \approx_h^w Q$ and $k_i + 1 \leq w' + 1 \leq w$, we obtain that $P \equiv \alpha_i.P_1'|...|\alpha_i.P_{k_i+1}'|P'$ with $P_j' \approx_{h-1}^w P_i$ for any $j = 1..k_i + 1$.

But $[\![P_i]\!] \leq (h' - 1, w') < (h - 1, w)$, consequently we can use the inductive hypothesis and obtain $P_j' \equiv P_i$ for any $j = 1..k_i + 1$.

Hence $P \equiv (\alpha_i.P_i)^{k_i+1}|P''$ which is impossible because we supposed that $P \equiv (\alpha_1.P_1)^{k_1}|...|(\alpha_n.P_n)^{k_n}$ with $\alpha_i.P_i \not\equiv \alpha_j.P_j$ for $i \neq j$.

Concluding, $R \equiv 0$ and $Q \equiv (\alpha_1.P_1)^{k_1}|...|(\alpha_n.P_n)^{k_n}$, i.e. $Q \equiv P$. □

**Theorem 3.3.8.** *If* $P \equiv R|P'$, $P \approx_h^w Q$ *and* $[\![R]\!] < (h, w)$ *then* $Q \equiv R|Q'$.

*Proof.* Suppose that $[\![R]\!] = (h', w') < (h, w)$. Because $P \equiv R|P'$ and $P \approx_h^w Q$, using theorem 3.3.5, we obtain that exists $Q_1, Q_2$ such that $Q \equiv Q_1|Q_2$ and $R \approx_h^{w'+1} Q_1$ and $P' \approx_h^{w-(w'+1)} Q_2$. Further, as $R \approx_h^{w'+1} Q_1$ and $[\![R]\!] = (h', w') < (h, w' + 1)$ we obtain, by using theorem 3.3.7, that $Q_1 \equiv R$, hence $Q \equiv R|Q_2$. □

**Theorem 3.3.9.** *Let $P \approx_h^w Q$. If $P \equiv \alpha.P'|P''$ then $Q \equiv \alpha.Q'|Q''$ and $P'|P'' \approx_{h-1}^{w-1} Q'|Q''$*

*Proof.* As $P \approx_h^w Q$ and $P \equiv \alpha.P'|P''$, we obtain that, indeed, $Q \equiv \alpha.Q'|Q''$ with $P' \approx_{h-1}^w Q'$. We will prove that $P'|P'' \approx_{h-1}^{w-1} Q'|Q''$. Consider any $i = 1..w-1$ and $\beta \in \mathbb{A}$ such that:

$$P'|P'' \equiv \beta.P_1|...|\beta.P_i|P^\star \tag{3.3.1}$$

We can suppose, without loos of generality that for some $k \leq i$ we have

$$P' \equiv \beta.P_1|...|\beta.P_k|P^+$$
$$P'' \equiv \beta.P_{k+1}|...|\beta.P_i|P^-$$
$$P^\star \equiv P^+|P^-$$

Because $P' \approx_{h-1}^w Q'$ and $k \leq i \leq w-1$, we obtain that $Q' \equiv \beta.Q_1|...|\beta.Q_k|Q^+$ with $P_j \approx_{h-2}^w Q_j$ for $j = 1..k$. Further we distinguish two cases:

- if $\alpha \neq \beta$ then we have

  $$P \equiv \beta.P_{k+1}|...|\beta.P_i|(P^-|\alpha.P')$$

  and because $P \approx_h^w Q$, we obtain

  $$Q \equiv \beta.R_{k+1}|...|\beta.R_i|R^\star \text{ with } R_j \approx_{h-1}^w P_j \text{ for } j = k+1..i$$

  But $Q \equiv \alpha.Q'|Q''$ and because $\alpha \neq \beta$, we obtain $Q'' \equiv \beta.R_{k+1}|...|\beta.R_i|R^+$ that gives us in the end

$$Q'|Q'' \equiv \beta.Q_1|...|\beta.Q_k|\beta.R_{k+1}|...|\beta.R_i|(R^+|Q^+)$$

with $P_j \approx^w_{h-2} Q_j$ for $j = 1..k$ (hence $P_j \approx^{w-1}_{h-2} Q_j$) and $P_j \approx^w_{h-1} R_j$ for $j = k+1..i$ (hence $P_j \approx^{w-1}_{h-2} R_j$).

- if $\alpha = \beta$ then we have

$$P \equiv \alpha.P_{k+1}|...|\alpha.P_i|\alpha.P'|P^-$$

and as $P \approx^w_h Q$ and $i \leq w-1$, we obtain

$$Q \equiv \alpha.R_{k+1}|...|\alpha.R_i|\alpha.R'|R^\star$$

with $R_j \approx^w_{h-1} P_j$ for $j = k+1..i$ and $R' \approx^w_{h-1} P'$. Because $P' \approx^w_{h-1} Q'$ and $\approx^w_h$ is an equivalence relation, we can suppose that $R' \equiv Q'$ (Indeed, if $\alpha.Q'$ is a subprocess of $R^\star$ then we can just substitute $R'$ with $Q'$; if $\alpha.Q' \equiv \alpha.R_s$, then $Q' \approx^w_{h-1} P_s$ and as $Q' \approx^w_{h-1} P'$ and $P' \approx^w_{h-1} R'$ we derive $R' \approx^w_{h-1} P_s$ and $Q' \approx^w_{h-1} P'$, so we can consider this correspondence). So

$$Q \equiv \alpha.R_{k+1}|...|\alpha.R_i|\alpha.Q'|R^\star$$

that gives

$$Q'' \equiv \alpha.R_{k+1}|...|\alpha.R_i|R^\star$$

which entails further

$$Q'|Q'' \equiv \alpha.Q_1|...|\alpha.Q_k|\alpha.R_{k+1}|...|\alpha.R_i|(R^\star|Q^+)$$

with $P_j \approx^w_{h-2} Q_j$ for $j = 1..k$ (hence $P_j \approx^{w-1}_{h-2} Q_j$) and $P_j \approx^w_{h-1} R_j$ for $j = k+1..i$ (hence $P_j \approx^{w-1}_{h-2} R_j$).

All these prove that $P'|P'' \approx^{w-1}_{h-1} Q'|Q''$ (as we can develop a symmetric argument starting in (3.3.1) with $Q|Q'$). □

**Theorem 3.3.10** (Behavioral simulation)**.** *Let $P \approx_h^w Q$. If $P \xrightarrow{\alpha} P'$ then exists a transition $Q \xrightarrow{\alpha} Q'$ such that $P' \approx_{h-1}^{w-1} Q'$.*

*Proof.* If $P \xrightarrow{\alpha} P'$ then $P \equiv \alpha.R'|R''$ and $P' \equiv R'|R''$. But $P \approx_h^w Q$ gives, using theorem 3.3.9 that $Q \equiv \alpha.S'|S''$ and $R'|R'' \approx_{h-1}^{w-1} S'|S''$. And because $Q \xrightarrow{\alpha} S'|S''$, we can take $Q' \equiv S'|S''$. □

## 3.4 Pruning processes

The main goal of this section is to prove the pruning theorem, stating that for a given process $P$ and a given size $(h, w)$, we can always find a process $Q$ having the size at most equal with $(h, w)$ such that $P \approx_h^w Q$. Moreover, in the proof of the theorem we will present a method for constructing such a process from $P$, by pruning its syntactic tree to the desired size.

**Theorem 3.4.1** (Pruning theorem)**.** *For any process $P \in \mathfrak{P}$ and any $(h, w)$ exists a process $Q \in \mathfrak{P}$ with $P \approx_h^w Q$ and $[\![Q]\!] \leq (h, w)$.*

*Proof.* We describe the construction[3] of $Q$ by induction on $h$.

**For $h = 0$:** we just take $Q \equiv 0$. Because for any process $R$ we have $P \approx_0^w R$, we have also $P \approx_0^w Q$ and $[\![0]\!] = (0, 0)$ as desired.

**For $h + 1$:** suppose that $P \equiv \alpha_1.P_1|...|\alpha_n.P_n$.
Let $P_i'$ be the result of pruning $P_i$ by $(h, w)$ (we use the inductive step of construction) and $P' \equiv \alpha_1.P_1'|...|\alpha_n.P_n'$.
As for any $i = 1..n$ we have $P_i \approx_h^w P_i'$ (by the inductive hypothesis), we obtain, using the congruence theorem 3.3.4, that $\alpha_i.P_i \approx_{h+1}^w \alpha_i.P_i'$ i.e. further, using the same theorem, $P \approx_{h+1}^w P'$.

[3]This construction is not necessarily unique.

Consider the canonical representation of $P' \equiv (\beta_1.Q_1)^{k_1}|...|(\beta_m.Q_m)^{k_m}$.
Let $l_i = min(k_i, w)$ for $i = 1..m$. Then we define $Q \equiv (\beta_1.Q_1)^{l_1}|...|(\beta_m.Q_m)^{l_m}$.
Obviously $Q \approx^w_{h+1} P'$ and as $P \approx^w_{h+1} P'$, we obtain $P \approx^w_{h+1} Q$. By construction, $[\![Q]\!] \leq (h+1, w)$. □

**Definition 3.4.1** (Pruning processes)**.** For a process $P$ and for a tuple $(h, w)$ we denote by $P_{(h,w)}$ the process obtained by pruning $P$ to the size $(h, w)$ by the method described in the proof of theorem 3.4.1.

**Example 3.4.1.** *Consider the process*

$$P \equiv \alpha.(\ \beta.(\gamma.0|\gamma.0|\gamma.0) \mid \beta.\gamma.0\ ) \mid \alpha.\beta.\gamma.0$$

*Observe that* $[\![P]\!] = (3, 3)$*, hence*

$$P_{(3,3)} = \alpha.(\ \beta.(\gamma.0|\gamma.0|\gamma.0) \mid \beta.\gamma.0\ ) \mid \alpha.\beta.\gamma.0 \equiv P$$

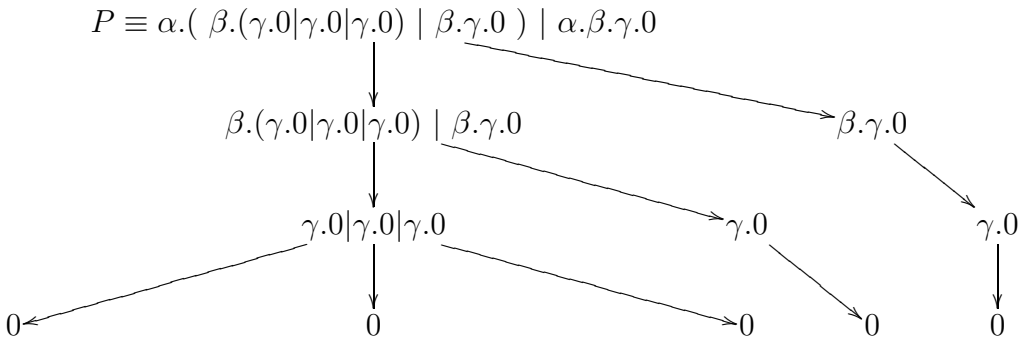


Figure 3.3: The syntactic tree of $P$

*Now we want to prune* $P$ *to the size* $(3, 2)$*, thus we have to prune the syntactic tree drawn in figure 3.3 such that to not exist, in any node, more than two bisimilar branches. In the tree drawn in figure 3.4 we marked with*

*dashed arrow the branch that has to be removed (in this case is only one). Hence*

$$P_{(3,2)} = \alpha.(\ \beta.(\gamma.0|\gamma.0)\ |\ \gamma.0)\ |\ \alpha.\beta.\gamma.0$$

*and its tree is drawn in figure 3.5.*

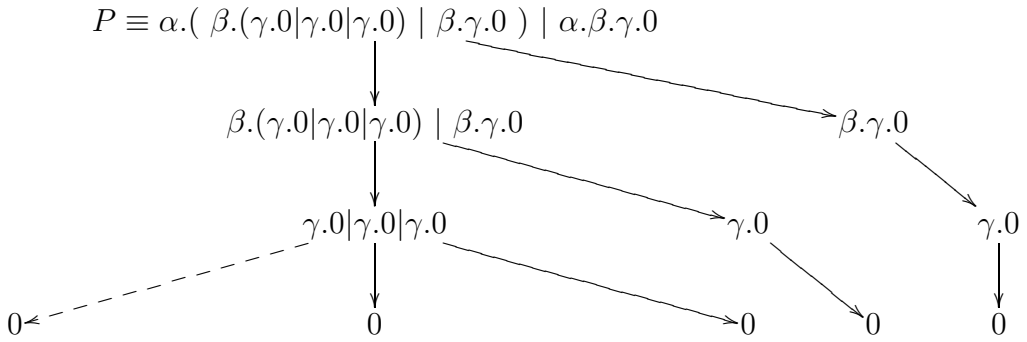


Figure 3.4: The syntactic tree of $P$

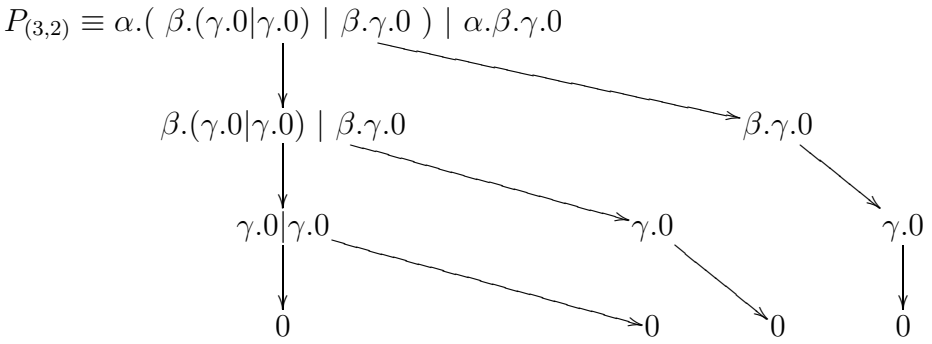


Figure 3.5: The syntactic tree of $P_{(3,2)}$

*If we want to prune $P$ on the size $(3,1)$ then we have to prune the syntactic tree such that, in any node, there are no bisimilar branches. Hence the ones marked dashed in figure 3.6 have to be removed. The pruned process will then be $P_{(3,1)} = \alpha.\beta.\gamma.0$ having the syntactic tree drawn in figure 3.7.*
*Suppose that we want to prune $P$ on the size $(2,2)$. Then we have to replace all the processes in the nodes depth $2$ by $0$ and in the new tree we have to let, in any node, a maximum of two bisimilar branches. With these*

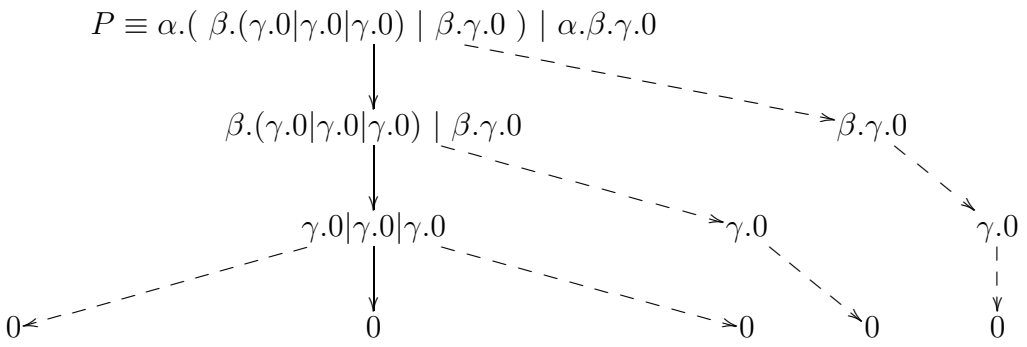


Figure 3.6: The syntactic tree of $P$

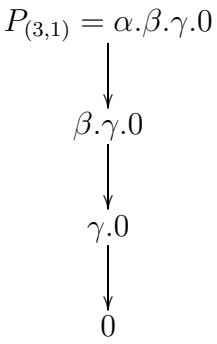


Figure 3.7: The syntactic tree of $P_{(3,1)}$

*modifications on the leaves of the tree, we have to reconstruct the whole tree. In figure 3.8 we marked with dashed arrows the branches that have to be pruned and with* $Q \Leftarrow 0$ *the nodes where the process* $Q$ *has to be replaced by the process* $0$*. So, reconstructing the syntactic tree, we obtain* $P_{(2,2)} = \alpha.(\beta.0|\beta.0) \mid \alpha.\beta.0$ *and its tree is drawn in figure 3.9.*

*We can go further with pruning and we will obtain the smaller processes*

$P_{(0,0)} = 0$

$P_{(1,1)} = \alpha.0$

$P_{(1,2)} = \alpha.0|\alpha.0$

$P_{(2,1)} = \alpha.\beta.0$

**Corollary 3.4.2.** *If* $P \equiv Q$ *then* $P_{(h,w)} \equiv Q_{(h,w)}$.

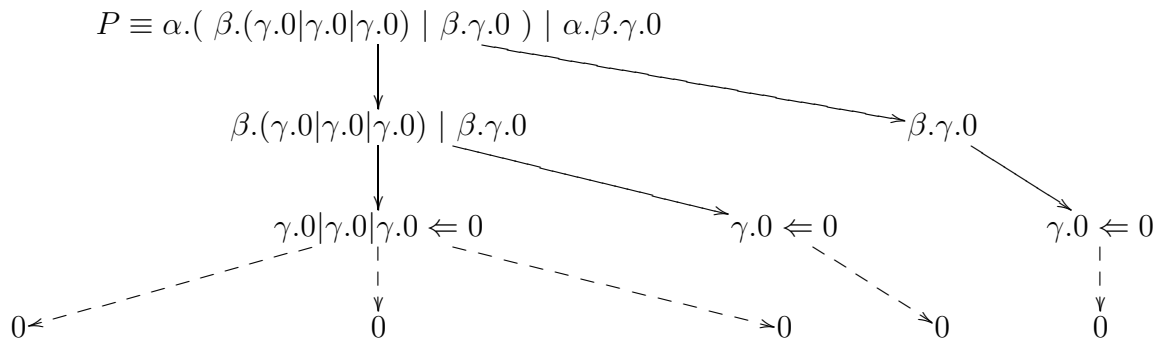


Figure 3.8: The syntactic tree of $P$

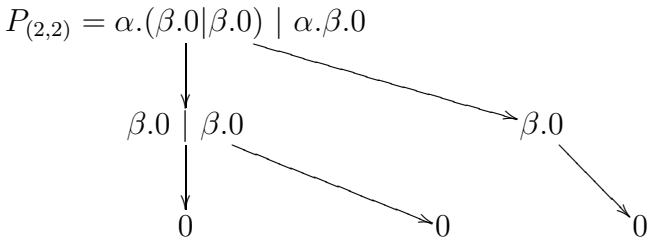


Figure 3.9: The syntactic tree of $P_{(2,2)}$

*Proof.* Because the canonical representation of a process is unique up to structural congruence, the result can be derived trivially, following the construction in the proof of theorem 3.4.1. □

**Corollary 3.4.3.** $[\![P]\!] \leq (h, w)$ *iff* $P_{(h,w)} \equiv P$.

*Proof.* ($\Rightarrow$) If $[\![P]\!] \leq (h, w)$, then, by construction, $[\![P_{(h,w)}]\!] \leq (h, w)$ and $P \approx_h^w P_{(h,w)}$, we can use theorem 3.3.6 and obtain $P_{(h,w)} \equiv P$.

($\Leftarrow$) Suppose that $P_{(h,w)} \equiv P$. Suppose, in addition that $[\![P]\!] > (h, w)$. By construction, $[\![P_{(h,w)}]\!] \leq (h, w)$, hence $[\![P_{(h,w)}]\!] \leq (h, w) < [\![P]\!]$, i.e. $[\![P_{(h,w)}]\!] \neq [\![P]\!]$. But this is impossible, because the size of a process is unique up to structural congruence, see remark 3.2.1. □

We will define further the set of all processes having a size smaller than a given tuple $(h, w)$ and we will prove that for the fragment of CCS we considered they are finitely many (modulo $\equiv$).

**Definition 3.4.2.** Consider defined the set

$$\mathfrak{P}_{(h,w)} \stackrel{def}{=} \{P \in \mathfrak{P} \mid [\![P]\!] \leq (h, w)\}$$

**Theorem 3.4.4.** $\mathfrak{P}_{(h,w)}$ *is finite*[4].

*Proof.* We will prove more: if we denote by $n = (w+1)^{card(\mathbb{A})}$, then

$$card(\mathfrak{P}_{(h,w)}) = \begin{cases} 1 & \text{if } h = 0 \\ \underbrace{n^{n^{\cdots^{n}}}}_{h} & \text{if } h \neq 0 \end{cases}$$

We prove this by induction on $h$.

**The case** $h = 0$**:** we have $[\![Q]\!] = (0, w)$ iff $Q \equiv 0$, so $\mathfrak{P}_{(0,w)} = \{0\}$ and $card(\mathfrak{P}_{(0,w)}) = 1 = n^0$.

**The case** $h = 1$**:** let $Q \in \mathfrak{P}_{(1,w)}$. Then

$Q \equiv (\alpha_1.Q_1)^{k_1}|...|(\alpha_s.Q_s)^{k_s}$ with $Q_i \in \mathfrak{P}_{(0,w)}$ and $\alpha_i.Q_i \not\equiv \alpha_j.Q_j$ for $i \neq j$.

But $Q_i \in \mathfrak{P}_{(0,w)}$ means $Q_i \equiv 0$, hence

$$Q \equiv (\alpha_1.0)^{k_1}|...|(\alpha_s.0)^{k_s}$$

Because $[\![Q]\!] \leq (1, w)$ we obtain that $k_i \leq w$. The number of guarded processes $\alpha.0$ with $\alpha \in \mathbb{A}$ is $card(\mathbb{A})$ and as $k_i \in 0..w$, the number of processes in $\mathfrak{P}_{(1,w)}$ is $(w+1)^{card(\mathbb{A})} = n^1$.

**The case** $h + 1$**:** let $Q \in \mathfrak{P}_{(h+1,w)}$. Then

$Q \equiv (\alpha_1.Q_1)^{k_1}|...|(\alpha_s.Q_s)^{k_s}$ with $Q_i \in \mathfrak{P}_{(h,w)}$ and $\alpha_i.Q_i \not\equiv \alpha_j.Q_j$ for $i \neq j$.

[4]We count the processes up to structural congruence.

Because $[\![Q]\!] \leq (h+1, w)$ we obtain that $k_i \leq w$. The number of guarded processes $\alpha.R$ with $\alpha \in \mathbb{A}$ and $R \in \mathfrak{P}_{(h,w)}$ is $card(\mathbb{A}) \times card(\mathfrak{P}_{(h,w)})$ and as $k_i \in 0..w$, the number of processes in $\mathfrak{P}_{(h+1,w)}$ is $(w+1)^{card(\mathbb{A}) \times card(\mathfrak{P}_{(h,w)})} = ((w+1)^{card(\mathbb{A})})^{card(\mathfrak{P}_{(h,w)})} = n^{card(\mathfrak{P}_{(h,w)})}$. But the inductive hypothesis gives $card(\mathfrak{P}_{(h,w)}) = \underbrace{n^{n^{\cdots^{n}}}}_{h}$, so $card(\mathfrak{P}_{(h+1,w)}) = \underbrace{n^{n^{\cdots^{n}}}}_{h+1}$. □

## 3.5 Contexts

In this section we introduce *the contexts*, sets of processes that will be used to evaluate formulas of our logics. The intuition is that a *context* $\mathcal{M}$ is a (possibly infinite) set of processes that contains, in a maximal manner, any process representing a possible state (or a possible approximation on a given size of such state) of our system or of a subsystem of our system. Our intention is to allow the possibility to have big contexts able to represent not only the system we analyze, but also the (possibly unknown) environment in which our system functions and with which it might interact. This intuition motivates the next definition. Then we will go further and check some properties of the contexts.

**Definition 3.5.1** (Context)**.** *A context* is a set $\mathcal{M} \subseteq \mathfrak{P}$ of processes such that

- if $P|Q \in \mathcal{M}$ then $P, Q \in \mathcal{M}$
- if $P \in \mathcal{M}$ and $P \longrightarrow P'$ then $P' \in \mathcal{M}$
- if $P \in \mathcal{M}$, and $(h, w) \leq [\![P]\!]$ then $P_{(h,w)} \in \mathcal{M}$

We extend the definition of size from processes to sets of processes.

**Definition 3.5.2** (Size of a set of processes)**.** Let $M \subset \mathfrak{P}$. We write $[\![M]\!] = (h, w)$ iff $(h, w) = max\{[\![P]\!] \mid P \in M\}$.
As the sets of processes may be infinite, not for all of them this definition works, in the sense that some sets may have infinite sizes[5]. For this reason we convey to extend the order, and when $M$ has infinite size, to still write $(h, w) \leq [\![M]\!]$ and $(h, w) < [\![M]\!]$ for any $(h, w)$.

Observe that, due to the closure clauses in definition 3.5.1, we can consider the possibility to define systems of generators for a context, as a class of processes that, using the rules in definition 3.5.1 can generate the full context.

**Definition 3.5.3** (System of generators for a context)**.** We say that the set $M \subset \mathfrak{P}$ is a system of generators for the context $\mathcal{M}$ if $\mathcal{M}$ is the smallest context that contains $M$. We denote this by $\overline{M} = \mathcal{M}$ or, reverse $\underline{\mathcal{M}} = M$.

**Theorem 3.5.1.** *If $M \in \mathfrak{P}$ is a finite set of processes, then $\overline{M}$ is a finite context.*

*Proof.* Trivial. □

[5]Such a situation is in the case of the set $\mathcal{M} = \{0, \alpha.0, \alpha.\alpha.0, ..., \alpha....\alpha.0, ...\}$.

## 3.6 Structural bisimulation on contexts

In this section we will extend the definitions of structural bisimulation from processes to contexts. This will allow us to prove the *context pruning theorem*, a result similar to the pruning theorem proved for processes.

**Definition 3.6.1** (Structural bisimulation over contexts)**.** Let $\mathcal{M}, \mathcal{N}$ be two contexts. We write $\mathcal{M} \approx_h^w \mathcal{N}$ iff

- for any $P \in \mathcal{M}$ there is a $Q \in \mathcal{N}$ with $P \approx_h^w Q$
- for any $Q \in \mathcal{N}$ there is a $P \in \mathcal{M}$ with $P \approx_h^w Q$

**Theorem 3.6.1** (Antimonotonicity over contexts)**.** *If* $\mathcal{M} \approx_h^w \mathcal{N}$ *and* $(h', w') \leq (h, w)$ *then* $\mathcal{M} \approx_{h'}^{w'} \mathcal{N}$.

*Proof.* For any process $P \in \mathcal{M}$ there exists a process $Q \in \mathcal{N}$ such that $P \approx_h^w Q$ and using theorem 3.3.3 we obtain $P \approx_{h'}^{w'} Q$. And the same if we start from a process $Q \in N$. These proves that $\mathcal{M} \approx_{h'}^{w'} \mathcal{N}$. □

## 3.7 Pruning contexts

As for processes, we can define the pruning of a context $\mathcal{M}$ as the context generated by the set of pruned processes of $\mathcal{M}$, taken as system of generators.

**Definition 3.7.1** (Pruning contexts)**.** For any context $\mathcal{M}$ and any $(h, w)$ we define

$$\mathcal{M}_{(h,w)} \stackrel{def}{=} \overline{\{P_{(h,w)} \mid P \in \mathcal{M}\}}$$

**Theorem 3.7.1.** *For any context $\mathcal{M}$, and any size $(h, w)$ we have $\mathcal{M}_{(h,w)} \approx_w^h \mathcal{M}$.*

*Proof.* Denote by

$$M = \{P_{(h,w)} \mid P \in \mathcal{M}\}$$

Let $P \in \mathcal{M}$. Then it exists a process $Q \in \mathcal{M}_{(h,w)}$, more exactly $Q \equiv P_{(h,w)}$ such that $P \approx_w^h Q$.
Let $Q \in \mathcal{M}_{(h,w)}$. Since $\overline{M}$ is the smallest context containing $M$, and because, by construction, $M \subseteq \mathcal{M}$ we derive that $\overline{M} \subseteq \mathcal{M}$. Hence, for any process $Q \in \overline{M}$ there is a process $P \in \mathcal{M}$, more exactly $P \equiv Q$ such that $P \approx_w^h Q$ (since $P \equiv Q$ implies $P \approx_w^h Q$). □

**Definition 3.7.2.** We denote by $\mathfrak{M}_{(h,w)}$ the set of all contexts generated by systems with the size at most $(h, w)$

$$\mathfrak{M}_{(h,w)} \stackrel{def}{=} \{\overline{M} \subset \mathfrak{P} \mid [\![M]\!] \leq (h, w)\}$$

**Theorem 3.7.2.** *The following hold:*

1. *If $[\![M]\!] \leq (h, w)$ then $\overline{M}$ is a finite context.*
2. *$\mathfrak{M}_{(h,w)}$ is finite.*

*Proof.* 1.: If $[\![M]\!] \leq (h, w)$, then $M \subset \mathfrak{P}_{(h,w)}$. But $\mathfrak{P}_{(h,w)}$ is finite, by theorem 3.4.4. Thus, by theorem 3.5.1 $\overline{M}$ is a finite context.

2.: As $\mathfrak{P}_{(h,w)}$ is finite by theorem 3.4.4, the set of its subsets is finite, and as all the elements of $\mathfrak{M}_{(h,w)}$ are generated by subsets of $\mathfrak{P}_{(h,w)}$, we obtain that $\mathfrak{M}_{(h,w)}$ is finite. □

**Theorem 3.7.3** (Pruning theorem)**.** *Let $\mathcal{M}$ be a context. Then for any $(h, w)$ there is a context $\mathcal{N} \in \mathfrak{M}_{(h,w)}$ such that $\mathcal{M} \approx_h^w \mathcal{N}$.*

*Proof.* The context $\mathcal{N} = \mathcal{M}_{(h,w)}$ fulfills the requirements of the theorem, by construction. Indeed, it is a context, and it is generated by the set $N = \{P_{(h,w)} \mid P \in \mathcal{M}\}$. But $[\![N]\!] \leq (h, w)$, hence $\mathcal{N} \in \mathfrak{M}_{(h,w)}$. Moreover, by theorem 3.7.1, $\mathcal{M} \approx_w^h \mathcal{N}$. □

## 3.8 Concluding remarks

In this chapter we developed some new concepts for the chosen fragment of CCS that concerns this thesis.

A new congruence on processes has been proposed, the *structural bisimulation*, that is a bisimilar-like approximation of the structural congruence sensitive to sizes of the processes - *height* (the depth of the syntactic tree of a process) and *weight* (the maximum number of bisimilar subprocesses that can be found in a node of the syntactic tree of a process).

Then we presented an effective method to construct, given a process $P$, a minimal process $Q$ that has an established size $(h, w)$ and is structurally bisimilar to $P$ at the same size. This method consists in pruning the syntactic tree of $P$ by eliminating all the branches that *"are not seen"* by the structural bisimulation relation indexed with the given size.

This analysis is important for the next chapters where we will prove that a spatial logic formula is sensitive up to the level of structural bisimulation indexed by the size of the logical formula. Thus, it will not distinguish between two structurally bisimilar processes at this size. Based on this idea we will be able to prove the finite model properties for our logics.

At the end of the chapter we extended the *structural bisimulation* and the *pruning method* from processes to classes of processes. We have a special interest in *contexts*, defined as special classes of processes that contain, in a maximal way, the processes needed for modeling a given system together with all the other processes that might represent subprocesses for our system, or pruning-approximations of these.

The contexts will be used in defining the semantics for our logics, as they will be the classes of processes on which we will evaluate the formulas.

# Chapter 4

# Dynamic Spatial Logic

In this chapter we introduce the Dynamic Spatial Logic, $\mathcal{L}_{DS}$, as an extension of Hennessy-Milner logic with the parallel operator. For it we extend the process semantics of Hennessy-Milner logic with the definition of satisfiability for the parallel operator, as usual in spatial logics. The satisfiability relation will evaluate a formula to a process in a context.

Although a similar logic has been considered before in the literature [15], the results presented here are all new.

$\mathcal{L}_{DS}$ will distinguish processes up to structural congruence level, as the other spatial logics. On the level of formulas, after introducing the notion of size of a formula, we will prove that each formula describes a process not up to structural congruence, but up to the structural bisimulation indexed by its size. Hence two processes that are structurally bisimilar on the size of the formula, cannot be distinguished. As a consequence, choosing the right size, we can define characteristic formulas for our processes. To the best of our knowledge, a similar result has not been proved for spatial logics before.

For our logic, we propose a Hilbert-style axiomatic system and we prove it to be sound and complete with respect to process semantics. This allows us to use the syntax to derive properties of the semantics. Combining

these features with the finite model property, which we will prove for the system against the process semantics, we find that, for $\mathcal{L}_{DS}$, the problems of satisfiability, validity and model checking are decidable.

The decidability has been anticipated before [15], but to our knowledge it has not been proved. Also new is the Hilbert-style approach to spatial logics.

## 4.1 Syntax of Dynamic Spatial Logic

**Definition 4.1.1** (Language of $\mathcal{L}_{DS}$)**.** We define the language of Dynamic Spatial Logic, as the formulas collected in the set $\mathcal{F}_{DS}$ introduced by:

$$\phi := \ 0 \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi|\phi \mid \langle\alpha\rangle\phi$$

where $\alpha \in \mathbb{A}$.

**Definition 4.1.2** (Derived operators)**.** In addition we introduce some derived operators:

1. $\bot \stackrel{def}{=} \neg\top$
2. $\phi \vee \psi \stackrel{def}{=} \neg((\neg\phi) \wedge (\neg\psi))$
3. $\phi \rightarrow \psi \stackrel{def}{=} (\neg\phi) \vee \psi$
4. $[\alpha]\phi \stackrel{def}{=} \neg(\langle\alpha\rangle(\neg\phi))$
5. $\bigwedge_{\phi\in M} \phi \stackrel{def}{=} (...((\phi_1 \wedge \phi_2) \wedge \phi_3)...\phi_k)$ for any finite set $M = \{\phi_1, \phi_2, ..., \phi_k\}$ of formulas.
6. $\bigvee_{\phi\in M} \phi \stackrel{def}{=} (...((\phi_1 \vee \phi_2) \vee \phi_3)...\phi_k)$ for any finite set $M = \{\phi_1, \phi_2, ..., \phi_k\}$ of formulas.

7. $|_{\phi \in M}\phi \stackrel{def}{=} (...((\phi_1|\phi_2)|\phi_3)...|\phi_k)$ for any finite set $M = \{\phi_1, \phi_2, ..., \phi_k\}$ of formulas.
8. $1 \stackrel{def}{=} \neg((\neg 0) \mid (\neg 0))$.
9. $\langle !\alpha \rangle \psi \stackrel{def}{=} (\langle \alpha \rangle \psi) \wedge 1$
10. $\diamond \phi \stackrel{def}{=} \bigvee_{\alpha \in \mathbb{A}} \langle \alpha \rangle \phi$
11. $\Box \phi \stackrel{def}{=} \neg(\diamond(\neg \phi))$

Anticipating the semantics, we will outline here the intuition that motivates the choice of the formulas. Mainly it is similar to that of Hennessy-Milner and spatial logics.

The formula 0 is meant to characterize any process structurally congruent with 0 (and only these) in any context, expressing *"there is no activity here"*. It should not be confused with "*false*".[1]

$\top$ will be satisfied by any process in any context.

The reason for introducing the parallel operator $\phi|\psi$ is that we want to be able to express, as in other spatial logics, the situation in which our system is composed by two parallel subsystems, one satisfying $\phi$ and the other satisfying $\psi$.

The dynamic-like operator $\langle \alpha \rangle \phi$ is meant to be used, as in Hennessy-Milner logic, to speak about the transitions of our system. It expresses *"the system may perform the action $\alpha$ thus meeting a state described by $\phi$"*.

$\perp$ will be used to express the inconsistent behavior of the system. For this reason no process, in any context, will satisfy $\perp$.

The dynamic-like operator $[\alpha]\phi$, the dual operator of $\langle \alpha \rangle \phi$, expresses the situation where either the system cannot perform $\alpha$, or if the system

[1]We insist on this aspect as some syntaxes of classical logic use 0 for denoting *false*. This is not our intention. We use $\perp$ to denote *false*.

can perform $\alpha$ then any future state that can be reached by performing $\alpha$ can be described by $\phi$.

The formula 1 is meant to describe the situation in which the system cannot be decomposed into two non-trivial subsystems. 1 can describe also the trivial system 0.

The formula $\langle !\alpha \rangle \psi$ expresses a process guarded by $\alpha$, which, after consuming $\alpha$, will satisfy $\psi$.

$\diamond\phi$ describes the system that has at least one next state that can be described by $\phi$.

By duality, $\Box\phi$ describes the system that either does not have a next state, or if it has one, all the next states satisfy $\phi$.

To relax the syntax of our logic, we propose a convention regarding the precedence of the operators.

**Assumption.** *We convey that the precedence order of the operators in the syntax of $\mathcal{L}_{DS}$ is*

$$\neg \ , \ \langle\alpha\rangle \ , \ | \ , \ \wedge \ , \ \vee, \ \rightarrow$$

*where $\neg$ have precedence over all other operators.*

**Example 4.1.1.** *Consider the formula*

$$\theta = (\neg(\langle\alpha\rangle\phi))|(\psi \wedge (\neg\rho))$$

*then, using the previous assumption, it can be written as*

$$\theta = \neg\langle\alpha\rangle\phi \mid (\psi \wedge \neg\rho)$$

## 4.2 Process Semantics

Hereafter we introduce the process semantics of $\mathcal{L}_{DS}$. A formula will be evaluated to processes in a given context. We will prove later that for $\mathcal{L}_{DS}$ the context itself is not relevant. This logic is not expressive enough to describe contextual situations. But the future extensions of $\mathcal{L}_{DS}$ with epistemic operators are sensitive to the context, meaning that the same process will satisfy different formulas in different contexts. For uniformity of presentation, we chose to introduce the semantics by contexts.

**Definition 4.2.1** (Models and satisfaction)**.** A model of $\mathcal{L}_{DS}$ is a context $\mathcal{M}$ for which we define the satisfaction relation, for $P \in \mathcal{M}$, as follows:

- $\mathcal{M}, P \models \top$ always
- $\mathcal{M}, P \models 0$ iff $P \equiv 0$
- $\mathcal{M}, P \models \neg\phi$ iff $\mathcal{M}, P \not\models \phi$
- $\mathcal{M}, P \models \phi \wedge \psi$ iff $\mathcal{M}, P \models \phi$ and $\mathcal{M}, P \models \psi$
- $\mathcal{M}, P \models \phi|\psi$ iff $P \equiv Q|R$ and $\mathcal{M}, Q \models \phi$, $\mathcal{M}, R \models \psi$
- $\mathcal{M}, P \models \langle\alpha\rangle\phi$ iff there exists a transition $P \xrightarrow{\alpha} P'$ and $\mathcal{M}, P' \models \phi$

Then the semantics of the derived operators will be:

- $\mathcal{M}, P \not\models \bot$ always
- $\mathcal{M}, P \models \phi \vee \psi$ iff $\mathcal{M}, P \models \phi$ or $\mathcal{M}, P \models \psi$
- $\mathcal{M}, P \models [\alpha]\phi$ iff
  - either there is no transition $P \xrightarrow{\alpha} P'$

  - or for any $P' \in \mathcal{M}$ such that $P \xrightarrow{\alpha} P'$ we have $\mathcal{M}, P' \models \phi$

- $\mathcal{M}, P \models 1$ iff $P \equiv 0$ or $P \equiv \alpha.Q$ ($P$ is guarded)
- $\mathcal{M}, P \models \Diamond\phi$ iff there is a transition $P \longrightarrow P'$ and $\mathcal{M}, P' \models \phi$
- $\mathcal{M}, P \models \Box\phi$ iff
  - either there is no transition $P \longrightarrow P'$
  - or for any $P' \in \mathcal{M}$ such that $P \longrightarrow P'$ we have $\mathcal{M}, P' \models \phi$

In the end of this section we recall some classic definitions.

**Definition 4.2.2.** We call a formula $\phi \in \mathcal{F}_{DS}$ *satisfiable* if there exists a context $\mathcal{M}$ and a process $P \in \mathcal{M}$ such that $\mathcal{M}, P \models \phi$.
We call a formula $\phi \in \mathcal{F}_{DS}$ *validity* if for any context $\mathcal{M}$ and any process $P \in \mathcal{M}$ we have $\mathcal{M}, P \models \phi$. In such a situation we write $\models \phi$.
Given a context $\mathcal{M}$, we denote by $\mathcal{M} \models \phi$ the situation when for any $P \in \mathcal{M}$ we have $\mathcal{M}, P \models \phi$.

*Remark* 4.2.1. $\phi$ is satisfiable iff $\neg\phi$ is not a validity, and vice versa, $\phi$ is a validity iff $\neg\phi$ is not satisfiable.

## 4.3 Finite model property and decidability

In this section we will prove the finite model property for $\mathcal{L}_{DS}$, i.e. we will prove that for any satisfiable formula $\phi$ there exists a process $P$ in a context $\mathcal{M}$, belonging to a finite class of such couples, such that $\mathcal{M}, P \models \phi$. Put more concretely, given a formula $\phi$, we can construct a finite class $C_\phi$ of

couples $(\mathcal{M}, P)$ (where $\mathcal{M}$ is a context and $P \in \mathcal{M}$), depending on $\phi$, such that if $\phi$ is satisfiable then one of these couples must satisfy it as well.

This makes it possible to verify, in a finite way, the satisfiability of a formula. Indeed, as $C_\phi$ is finite, for deciding if $\phi$ is satisfiable, it is sufficient to verify if $\mathcal{M}, P \models \phi$ for each $(\mathcal{M}, P) \in C_\phi$.

The intuition that leads us in the construction of $C_\phi$ is that in the relation $\mathcal{M}, P \models \phi$ what matters is the structure of the process $P$ on a size $(h, w)$ that depends on $\phi$. Deeper $\phi$ is not "sensitive". In other words, we can derive from the structure of $\phi$ a size $(h, w)$ such that if $\mathcal{M}, P \models \phi$ then any process $Q \approx_w^h P$, in any context $\mathcal{N} \ni Q$ has the property $\mathcal{N}, Q \models \phi$.

As, by pruning theorem, for any size $(h, w)$ and any process $P$ we can find a process $Q$ with $P \approx_w^h Q$, we are motivated to believe that the class $\mathfrak{P}_{(h,w)}$ is the projection of $C_\phi$ onto $\mathfrak{P}$ (indeed $\mathfrak{P}_{(h,w)}$ contains a finite number of processes, by theorem 3.4.4). Further we will prove that these intuitions are correct, and we will try to identify $C_\phi$.

We start by introducing *the size of a formula* of our logic in a way similar to the size defined for processes.

**Definition 4.3.1** (Size of a formula)**.** We define *the sizes of a formula*, $(\!|\phi|\!)$ (*height* and *width*), inductively on $\mathcal{F}_{DS}$, by:

$(\!|0|\!) \stackrel{def}{=} (1, 1)$

$(\!|\top|\!) \stackrel{def}{=} (0, 0)$

$(\!|\neg\phi|\!) \stackrel{def}{=} (\!|\phi|\!)$

$(\!|\phi \wedge \psi|\!) \stackrel{def}{=} (max(h_1, h_2), max(w_1, w_2))$ if $(\!|\phi|\!) = (h_1, w_1)$, $(\!|\psi|\!) = (h_2, w_2)$

$(\!|\phi|\psi|\!) = (max(h_1, h_2), w_1 + w_2)$ where $(\!|\phi|\!) = (h_1, w_1)$ and $(\!|\psi|\!) = (h_2, w_2)$

$(\!|\langle\alpha\rangle\phi|\!) = (1 + h, 1 + w)$ where $(\!|\phi|\!) = (h, w)$

**Lemma 4.3.1.** *If* $(\!|\phi|\!) = (h, w)$, $\mathcal{M}, P \models \phi$ *and* $P \approx_h^w Q$ *then for any context* $\mathcal{M}'$ *with* $Q \in \mathcal{M}'$, *we have* $\mathcal{M}', Q \models \phi$.

*Proof.* We prove it by induction on the structure of $\phi$.

- **The case** $\phi = 0$**:** gives $(\!|\phi|\!) = (1, 1)$ and $\mathcal{M}, P \models 0$ implies $P \equiv 0$. As $P \approx_1^1 Q$, we should have $Q \equiv 0$, because else $Q \equiv \alpha.Q'|Q''$ asks for $P \equiv \alpha.P'|P''$ for some $P', P''$, but this is impossible because $P \equiv 0$. So $Q \equiv 0$ and for any $\mathcal{M}'$ we have, indeed, $\mathcal{M}', Q \models 0$.

- **The case** $\phi = \top$**:** is a trivial case because $\mathcal{M}', Q \models \top$ always.

- **The case** $\phi = \phi_1 \wedge \phi_2$**:** denote by $(h_i, w_i) = (\!|\phi_i|\!)$ for $i = 1, 2$. We have $(\!|\phi|\!) = (max(h_1, h_2), max(w_1, w_2))$.
  $\mathcal{M}, P \models \phi$ is equivalent with $\mathcal{M}, P \models \phi_1$ and $\mathcal{M}, P \models \phi_2$.
  As $P \approx_{max(h_1,h_2)}^{max(w_1,w_2)} Q$ we obtain, by using theorem 3.3.3, that $P \approx_{h_1}^{w_1} Q$ and $P \approx_{h_2}^{w_2} Q$.
  For $\mathcal{M}, P \models \phi_1$ and $P \approx_{h_1}^{w_1} Q$ we can apply the inductive hypothesis and obtain that for any context $\mathcal{M}' \ni Q$ we have $\mathcal{M}', Q \models \phi_1$.
  Similarly, $\mathcal{M}, P \models \phi_2$ and $P \approx_{h_2}^{w_2} Q$ gives that for any context $\mathcal{M}' \ni Q$ we have $\mathcal{M}', Q \models \phi_2$.
  Hence for any context $\mathcal{M}' \ni Q$ we have $\mathcal{M}', Q \models \phi$.

- **The case** $\phi = \neg\phi'$**:** we have $(\!|\phi|\!) = (\!|\phi'|\!) = (h, w)$, $\mathcal{M}, P \models \neg\phi'$ and $P \approx_h^w Q$.
  If for some context $\mathcal{M}' \ni Q$ we have $\mathcal{M}', Q \not\models \neg\phi'$, then
  $\mathcal{M}', Q \models \neg\neg\phi'$, hence $\mathcal{M}', Q \models \phi'$.
  But $\mathcal{M}', Q \models \phi'$ and $P \approx_h^w Q$ give, by using the inductive hypothesis, that for any context $\mathcal{M}'' \ni P$ we have $\mathcal{M}'', P \models \phi'$. So, we have also $\mathcal{M}, P \models \phi'$ and as $\mathcal{M}, P \models \neg\phi'$ we obtain $\mathcal{M}, P \models \bot$ - impossible.
  Hence for any context $\mathcal{M}' \ni Q$ we have $\mathcal{M}', Q \models \phi$.

- **The case** $\phi = \phi_1|\phi_2$**:** suppose that $(\!|\phi_i|\!) = (h_i, w_i)$ for $i = 1, 2$. Then

$(\!|\phi|\!) = (max(h_1, h_2), w_1 + w_2)$.
Now $\mathcal{M}, P \models \phi_1|\phi_2$ implies $P \equiv P_1|P_2$, with $\mathcal{M}, P_1 \models \phi_1$ and $\mathcal{M}, P_2 \models \phi_2$.
Because $P \approx^{w_1+w_2}_{max(h_1,h_2)} Q$, using theorem 3.3.5, we obtain $Q \equiv Q_1|Q_2$ with $P_i \approx^{w_i}_{max(h_1,h_2)} Q_i$ for $i = 1, 2$. Further, using theorem 3.3.3 we obtain $P_i \approx^{w_i}_{h_i} Q_i$.
Now $\mathcal{M}, P_i \models \phi_i$ and $P_i \approx^{w_i}_{h_i} Q_i$ give, by the inductive hypothesis, that for any context $\mathcal{M}'' \ni Q_1$ we have $\mathcal{M}'', Q_1 \models \phi_1$ and for any context $\mathcal{M}''' \ni Q_2$ we have $\mathcal{M}''', Q_2 \models \phi_2$.
Then, for any context $\mathcal{M}' \ni Q \equiv Q_1|Q_2$ we have $\mathcal{M}', Q_i \models \phi_i$ (as a context that contains $Q_1|Q_2$ contains also $Q_1$ and $Q_2$).
Hence $\mathcal{M}', Q \models \phi$.

- **The case** $\phi = \langle\alpha\rangle\phi'$**:** suppose that $(\!|\phi'|\!) = (h, w)$. We have $(\!|\langle\alpha\rangle\phi'|\!) = (1 + h, 1 + w)$.
$\mathcal{M}, P \models \langle\alpha\rangle\phi'$ means that $P \xrightarrow{\alpha} P'$ and $\mathcal{M}, P' \models \phi'$.
But because $P \approx^{1+w}_{1+h} Q$, using theorem 3.3.10, we obtain that $Q \xrightarrow{\alpha} Q'$ and $P' \approx^{w}_{h} Q'$.
Now from $\mathcal{M}, P' \models \phi'$ and $P' \approx^{w}_{h} Q'$, we obtain, by using the inductive hypothesis, that for any context $\mathcal{M}'' \ni Q'$ we have $\mathcal{M}'', Q' \models \phi'$. As $Q \xrightarrow{\alpha} Q'$ and because any context that contains $Q$ contains $Q'$ as well, we obtain further that for any context $\mathcal{M}' \ni Q$ we have $\mathcal{M}', Q' \models \phi'$, hence $\mathcal{M}, Q \models \phi$.

□

Hence, lemma 4.3.1 proved that if a process in a context satisfies a formula then any other process structurally bisimilar to our process on the size of the formula satisfies the formula in any context. It is then natural to try to find such a process in a finite context.

**Theorem 4.3.2.** *If $\mathcal{M}, P \models \phi$ then $\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \phi$.*

*Proof.* Let $\mathcal{M}, P \models \phi$ and $(h, w) = (\!|\phi|\!)$. Then, by pruning theorem, 3.4.1, exists the process $P_{(h,w)}$ with $P \approx_h^w P_{(h,w)}$. Then, using lemma 4.3.1, we obtain that for any context $\mathcal{M}'$ such that $P_{(h,w)} \in \mathcal{M}'$ we have $\mathcal{M}', P_{(h,w)} \models \phi$. But $P_{(h,w)} \in \mathcal{M}_{(h,w)}$. Hence $\mathcal{M}_{(h,w)}, P_{(h,w)} \models \phi$. □

As $\mathfrak{M}_{(\!|\phi|\!)}$ is finite and $\mathcal{M}_{(\!|\phi|\!)} \in \mathfrak{M}_{(\!|\phi|\!)}$ is also finite, it seems that couples $(P_{(\!|\phi|\!)}, \mathcal{M}_{(\!|\phi|\!)})$ with $P_{(\!|\phi|\!)} \in \mathcal{M}_{(\!|\phi|\!)}$ are finitely many, and their set depends directly on $\phi$. Thus, it might be the case that this set is $C_\phi$.

**Theorem 4.3.3** (Finite model property)**.** *If $\mathcal{M}, P \models \phi$ and $(\!|\phi|\!) = (h, w)$, then exists a context $\mathcal{N} \in \mathfrak{M}_{(h,w)}$ and a process $Q \in \mathcal{N}$ such that $\mathcal{N}, Q \models \phi$.*

*Proof.* If we take $\mathcal{N} = \mathcal{M}_{(h,w)}$ (because $\mathcal{M}_{(h,w)} \in \mathfrak{M}_{(h,w)}$) and $Q = P_{(h,w)}$, then theorem 4.3.2 gives $\mathcal{N}, Q \models \phi$, q.e.d. □

The fact that our logic has the finite model property against the process semantics entails:

- Satisfiability checking is decidable, meaning that a finite procedure exists such that, taking the formula $\phi$ as input, decides, in a finite manner, if there exists a process satisfying it in a context; indeed, this procedure may construct $C_\phi$ in the manner presented before and then browsing it to find such a model - the searching is finite because $C_\phi$ is finite.
- Validity checking is decidable, because $\phi$ is valid iff $\neg\phi$ is not satisfiable; but the satisfiability of $\neg\phi$ can be decided by following the finite-time procedure shown before.

- Model checking is decidable because, given a process $P$ in a context $\mathcal{M}$ and a formula $\phi$ then $\mathcal{M}, P \models \phi$ is equivalent with $\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \phi$ that requires a finite verification.

**Theorem 4.3.4** (Decidability of $\mathcal{L}_{DS}$)**.** *For $\mathcal{L}_{DS}$ against process semantics, satisfiability, validity and model checking are decidable.*

## 4.4 Axioms of $\mathcal{L}_{DS}$

In this section we propose a Hilbert-style axiomatic system for the Dynamic Spatial Logic, $\mathcal{L}_{DS}$. The system will be constructed in top of the classical propositional logic. Hence all the *axioms and rules of propositional logic* are available. In addition we will have a class of *spatial axioms and rules* that describe, mainly, the behavior of the parallel operator, and a class of *dynamic axioms and rules* regarding the behavior of the dynamic operators in relation with the parallel one. In the next sections we will prove that the system is sound and complete with respect to process semantics.

We begin by defining, inductively on processes, a special class of formulas that characterize a process up to structural congruence.

**Definition 4.4.1** (Characteristic formulas)**.** We define a class of formulas $(c_P)_{P \in \mathfrak{P}}$, indexed by ($\equiv$-equivalence classes of) processes, as follows:

1. $c_0 \stackrel{def}{=} 0$
2. $c_{P|Q} \stackrel{def}{=} c_P | c_Q$
3. $c_{\alpha.P} \stackrel{def}{=} \langle !\alpha \rangle c_P$

**Spatial axioms**

**Axiom D 1.** $\vdash \top|\bot \rightarrow \bot$

**Axiom D 2.** $\vdash \phi|0 \leftrightarrow \phi$

**Axiom D 3.** $\vdash \phi|\psi \rightarrow \psi|\phi$

**Axiom D 4.** $\vdash (\phi|\psi)|\rho \rightarrow \phi|(\psi|\rho)$

**Axiom D 5.** $\vdash \phi|(\psi \vee \rho) \rightarrow (\phi|\psi) \vee (\phi|\rho)$

**Axiom D 6.** $\vdash (c_P \wedge \phi|\psi) \rightarrow \bigvee_{P \equiv Q|R}(c_Q \wedge \phi)|(c_R \wedge \psi)$

**Spatial rules**

**Rule $\mathbf{D}_R$ 1.** *If* $\vdash \phi \rightarrow \psi$ *then* $\vdash \phi|\rho \rightarrow \psi|\rho$

**Dynamic axioms**

**Axiom D 7.** $\vdash \langle\alpha\rangle\phi|\psi \rightarrow \langle\alpha\rangle(\phi|\psi)$

**Axiom D 8.** $\vdash [\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [a]\psi)$

**Axiom D 9.** $\vdash 0 \rightarrow [\alpha]\bot$

**Axiom D 10.** *If* $\beta \neq \alpha_i$ *for* $i = 1..n$ *then* $\vdash \langle!\alpha_1\rangle\top|...|\langle!\alpha_n\rangle\top \rightarrow [\beta]\bot$

**Axiom D 11.** $\vdash \langle!\alpha\rangle\phi \rightarrow [\alpha]\phi$

**Dynamic rules**

**Rule $\mathbf{D}_R$ 2.** *If* $\vdash \phi$ *then* $\vdash [\alpha]\phi$

**Rule $\mathbf{D}_R$ 3.** *If* $\vdash \phi \rightarrow [\alpha]\phi'$ *and* $\vdash \psi \rightarrow [\alpha]\psi'$ *then* $\vdash \phi|\psi \rightarrow [\alpha](\phi'|\psi \vee \phi|\psi')$.

**Rule $\mathrm{D}_R$ 4.** *If* $\vdash \bigvee_{[\![Q]\!] \leq (\!|\phi|\!)} c_Q \rightarrow \phi$ *then* $\vdash \phi$.

Axiom D1 states the propagation of the inconsistency from a subsystem to the upper system.

Axioms D2, D3 and D4 depict the structure of abelian monoid projected by the parallel operator on the class of processes.

Concerning axiom D6, observe that the disjunction involved has a finite number of terms, as we considered the processes up to structural congruence level. The theorem states that if system has a property expressed by parallel composition of specifications, then it must have two parallel complementary subsystems, each of them satisfying one of the specifications.

Rule $\mathrm{D}_R1$ states a monotony property for the parallel operator.

The first dynamic axiom, axiom D7, presents a domain extrusion property for the dynamic operator. It expresses the fact that if an active subsystem of a bigger system performs the action $\alpha$, then the bigger system performs it as a whole.

Axiom D8 is just the (K)-axiom for the dynamic operator.

Axiom D9 states that an inactive system cannot perform any action.

Given a complex process that can be exhaustively decomposed in $n$ parallel subprocesses, each of them being able to perform one action only, $\alpha_i$, for $i = 1..n$, axiom D10 ensures us that the entire system, as a whole, cannot perform another action $\beta \neq \alpha_i$ for $i = 1..n$.

Recalling that the operator $\langle !\alpha \rangle$ describes processes guarded by $\alpha$, axiom D11 states that a system described by a guarded process can perform one and only one action, the guarding one.

Rule $\mathrm{D}_R2$ is the classic necessity rule used for the dynamic operator.

Rule $\mathrm{D}_R3$ is, in a sense, a counterpart of axiom D7 establishing the action of the operator $[\alpha]$ in relation to the parallel operator.

Rule $D_R4$ comes as a consequence of the finite model property and provides a rule that characterizes, in a finite manner, the validity of a formula. Observe that the disjunction in the first part of the rule has a finite number of terms as $[\![P]\!] \leq (\!|\phi|\!)$ defines the set $\mathfrak{P}_{(\!|\phi|\!)}$, which is finite (modulo $\equiv$).

## 4.5 Soundness of $\mathcal{L}_{DS}$ with respect to process semantics

In this section we will prove that our intuition behind the axioms and rules is correct and that, indeed, these describe real behaviors of processes. We will do this by proving *the soundness theorem*. Such a theorem states that our axioms and rules are correct descriptions of the semantics, i.e. of the algebra of processes and, in consequence, everything that can be proved using our axiomatic system will be true about processes in the given interpretation (via satisfiability relation).

**Theorem 4.5.1** (Process-Soundness)**.** *The system $\mathcal{L}_{DS}$ is a sound system with respect to the process semantics.*

*Proof.* The proof derives, as a consequence, from the soundness of all the axioms and rules of the system. These are proved further, in this section. □

### 4.5.1 Soundness of the spatial axioms and rules

We start with proving the soundness of the spatial axioms and rules.

**Lemma 4.5.2** (Soundness of axiom D1)**.** $\models \top|\bot \rightarrow \bot$

*Proof.* Suppose that it exists a context $\mathcal{M}$ and a process $P \in \mathcal{M}$ such that $\mathcal{M}, P \models \top|\bot$. Then $P \equiv Q|R$ with $\mathcal{M}, Q \models \top$ and $\mathcal{M}, R \models \bot$; i.e. $\mathcal{M}, R \not\models \top$. But this is not possible. Hence, there is no context $\mathcal{M}$ and process $P \in \mathcal{M}$ such that $\mathcal{M}, P \models \top|\bot$, i.e. for any context $\mathcal{M}$ and any process $P \in \mathcal{M}$ we have $\mathcal{M}, P \models \neg(\top|\bot)$, i.e. $\mathcal{M}, P \models \top|\bot \rightarrow \bot$. □

**Lemma 4.5.3** (Soundness of axiom D2)**.** $\models \phi|0 \leftrightarrow \phi$.

*Proof.* $\mathcal{M}, P \models \phi|0$ iff $P \equiv Q|R$, $\mathcal{M}, Q \models \phi$ and $\mathcal{M}, R \models 0$. Then $R \equiv 0$, so $P \equiv Q$, hence $\mathcal{M}, P \models \phi$.
If $\mathcal{M}, P \models \phi$, because $\mathcal{M}, 0 \models 0$ and $P \equiv P|0 \in \mathcal{M}$ we obtain that $\mathcal{M}, P \models \phi|0$. □

**Lemma 4.5.4** (Soundness of axiom D3)**.** $\models \phi|\psi \rightarrow \psi|\phi$.

*Proof.* $\mathcal{M}, P \models \phi|\psi$ means that $P \equiv Q|R$, $\mathcal{M}, Q \models \phi$ and $\mathcal{M}, R \models \psi$. But $P \equiv R|Q \in \mathcal{M}$, hence $\mathcal{M}, P \models \psi|\phi$. □

**Lemma 4.5.5** (Soundness of axiom D4)**.** $\models (\phi|\psi)|\rho \rightarrow \phi|(\psi|\rho)$.

*Proof.* $\mathcal{M}, P \models (\phi|\psi)|\rho$ implies that $P \equiv Q|R$, $\mathcal{M}, Q \models \phi|\psi$ and $\mathcal{M}, R \models \rho$. Then $Q \equiv S|V$ with $\mathcal{M}, S \models \phi$ and $\mathcal{M}, V \models \psi$. But $P \equiv (S|V)|R \equiv S|(V|R)$, where $\mathcal{M}, S \models \phi$ and $\mathcal{M}, V|R \models \psi|\rho$. Hence $\mathcal{M}, P \models \phi|(\psi|\rho)$. □

**Lemma 4.5.6** (Soundness of axiom D5)**.**

$$\models \phi|(\psi \vee \rho) \rightarrow (\phi|\psi) \vee (\phi|\rho)$$

*Proof.* $\mathcal{M}, P \models \phi|(\psi \vee \rho)$ means that $P \equiv Q|R$, $\mathcal{M}, P \models \phi$ and $\mathcal{M}, R \models \psi \vee \rho$, i.e. $\mathcal{M}, R \models \psi$ or $\mathcal{M}, R \models \rho$. Hence $\mathcal{M}, P \models \phi|\psi$ or $\mathcal{M}, P \models \phi|\rho$. So $\mathcal{M}, P \models (\phi|\psi) \vee (\phi|\rho)$. □

On this point we have enough information to prove two expected results: first that $c_P$ is, indeed, satisfied by the process $P$ and second, that the formula $c_P$ is satisfied by the whole $\equiv$-equivalence class of $P$. These results will be useful in proving the rest of the soundness lemmas.

**Theorem 4.5.7.** *If $P \in \mathcal{M}$, then $\mathcal{M}, P \models c_P$.*

*Proof.* We prove it by induction on the structure of the process $P$.
**The case $P \equiv 0$:** $\mathcal{M}, 0 \models c_0$, because $0 \in \mathcal{M}$, $c_0 = 0$ and $\mathcal{M}, 0 \models 0$.
**The case $P \equiv Q|R$:** we have $Q, R \in \mathcal{M}$ and $c_P = c_Q|c_R$. By the inductive hypothesis $\mathcal{M}, Q \models c_Q$ and $\mathcal{M}, R \models c_R$, so $\mathcal{M}, Q|R \models c_Q|c_R$. Hence $\mathcal{M}, P \models c_P$.
**The case $P \equiv \alpha.Q$:** we have $P \xrightarrow{\alpha} Q$, hence $Q \in \mathcal{M}$. Moreover, $c_P = \langle\alpha\rangle c_Q \wedge 1$. By the inductive hypothesis $\mathcal{M}, Q \models c_Q$. Because $P \xrightarrow{\alpha} Q$, we obtain $\mathcal{M}, P \models \langle\alpha\rangle c_Q$, and because $P \equiv \alpha.Q$ is a guarded process, we have also $\mathcal{M}, P \models 1$. Hence $\mathcal{M}, P \models c_P$. □

**Theorem 4.5.8.** *$\mathcal{M}, P \models c_Q$ iff $P \equiv Q$.*

*Proof.* ($\Leftarrow$) We prove it by verifying that $\mathcal{M}, P \models c_Q$ for any $P, Q$ involved in the equivalence rules.

- if $P = R|S$ and $Q = S|R$, we have $\mathcal{M}, R|S \models c_R|c_S$ and using the soundness of axiom D3, we obtain $\mathcal{M}, R|S \models c_S|c_R$, i.e. $\mathcal{M}, P \models c_Q$

- if $P = (R|S)|U$ and $Q = R|(S|U)$ we have $\mathcal{M}, P \models (c_R|c_S)|c_U$. Using the soundness of axiom D4, we obtain $\mathcal{M}, P \models c_Q$. Similarly $\mathcal{M}, Q \models c_P$, using the soundness of axioms D3 and D4.

- if $P = Q|0$ then $\mathcal{M}, P \models c_Q|0$, i.e., by using the soundness of axiom D2, $\mathcal{M}, P \models c_Q$. Similarly reverse, form $\mathcal{M}, Q \models c_Q$ we derive, by using the soundness of axiom D2, $\mathcal{M}, Q \models c_Q|0$, i.e. $\mathcal{M}, Q \models c_P$.

- if $P = P'|R$ and $Q = Q'|R$ with $P' \equiv Q'$ and $\mathcal{M}, P' \models c_{Q'}$, because $\mathcal{M}, R \models c_R$, we obtain that $\mathcal{M}, P \models c_{Q'}|c_R$, i.e. $\mathcal{M}, P \models c_Q$.

- if $P = \alpha.P'$ and $Q = \alpha.Q'$ with $P' \equiv Q'$ and $\mathcal{M}, P' \models c_{Q'}$, as $P \xrightarrow{\alpha} P'$, then $\mathcal{M}, P \models \langle\alpha\rangle c_{Q'}$. But $\mathcal{M}, P \models 1$, because $P$ is a guarded process, hence $\mathcal{M}, P \models \langle\alpha\rangle c_{Q'} \wedge 1$, i.e. $\mathcal{M}, P \models c_Q$.

($\Rightarrow$) We prove the implication in this sense by induction on the structure of $Q$.

- if $Q \equiv 0$, then $\mathcal{M}, P \models c_0$, means $\mathcal{M}, P \models 0$. Hence $P \equiv 0$.

- if $Q \equiv R|S$ then $\mathcal{M}, P \models c_Q$ is equivalent with $\mathcal{M}, P \models c_R|c_S$. So $P \equiv U|V$, $\mathcal{M}, U \models c_R$ and $\mathcal{M}, V \models c_S$. By the inductive hypothesis we obtain that $U \equiv R$ and $V \equiv S$. Hence $P \equiv Q$.

- if $Q \equiv \alpha.R$, then $\mathcal{M}, P \models c_Q$ is equivalent with $\mathcal{M}, P \models \langle\alpha\rangle c_R \wedge 1$. So $P \xrightarrow{\alpha} P'$ with $\mathcal{M}, P' \models c_R$. By the inductive hypothesis, $P' \equiv R$. And because $\mathcal{M}, P \models 1$ we obtain that $P \equiv \alpha.R$, i.e. $P \equiv Q$.

□

**Lemma 4.5.9** (Soundness of axiom D6)**.**

$$\models (c_P \wedge \phi|\psi) \rightarrow \bigvee_{P \equiv Q|R} (c_Q \wedge \phi)|(c_R \wedge \psi)$$

*Proof.* Suppose that $\mathcal{M}, S \models c_P \wedge \phi|\psi$. Then $S \equiv P$ (by theorem 4.5.8) and $S \equiv S_1|S_2$ with $\mathcal{M}, S_1 \models \phi$ and $\mathcal{M}, S_2 \models \psi$.
But $\mathcal{M}, S_1 \models c_{S_1}$ and $\mathcal{M}, S_2 \models c_{S_2}$, by theorem 4.5.7.
Hence $\mathcal{M}, S_1 \models \phi \wedge c_{S_1}$ and $\mathcal{M}, S_2 \models \psi \wedge c_{S_2}$.
And because $P \equiv S \equiv S_1|S_2$, we obtain $\mathcal{M}, P \models (\phi \wedge c_{S_1})|(\psi \wedge c_{S_2})$, hence $\mathcal{M}, P \models \bigvee_{P \equiv Q|R}(c_Q \wedge \phi)|(c_R \wedge \psi)$, q.e.d. □

**Lemma 4.5.10** (Soundness of rule $\mathrm{D}_R1$)**.**

$$\textit{If } \models \phi \to \psi \textit{ then } \models \phi|\rho \to \psi|\rho$$

*Proof.* If $\mathcal{M}, P \models \phi|\rho$ then $P \equiv Q|R$, $\mathcal{M}, Q \models \phi$ and $\mathcal{M}, R \models \rho$. But from the hypothesis, $\mathcal{M}, Q \models \phi \to \psi$, hence $\mathcal{M}, Q \models \psi$. Then $\mathcal{M}, P \models \psi|\rho$, so $\models \phi|\rho \to \psi|\rho$. □

### 4.5.2 Soundness of the dynamic axioms and rules

We prove now the soundness for the class of dynamic axioms and rules.

**Lemma 4.5.11** (Soundness of axiom D7)**.** $\models \langle\alpha\rangle\phi|\psi \to \langle\alpha\rangle(\phi|\psi)$.

*Proof.* If $\mathcal{M}, P \models \langle\alpha\rangle\phi|\psi$, then $P \equiv R|S$, $\mathcal{M}, R \models \langle\alpha\rangle\phi$ and $\mathcal{M}, S \models \psi$. So $\exists R \xrightarrow{\alpha} R'$ and $\mathcal{M}, R' \models \phi$. So $\exists P \equiv R|S \xrightarrow{\alpha} P' \equiv R'|S$ and $\mathcal{M}, P' \models \phi|\psi$. Hence $\mathcal{M}, P \models \langle\alpha\rangle(\phi|\psi)$. □

**Lemma 4.5.12** (Soundness of axiom D8)**.**

$$\models [\alpha](\phi \to \psi) \to ([\alpha]\phi \to [\alpha]\psi)$$

*Proof.* Let $\mathcal{M}, P \models [\alpha](\phi \to \psi)$ and $\mathcal{M}, P \models [\alpha]\phi$. If there is no $P'$ such that $P \xrightarrow{\alpha} P'$, then $\mathcal{M}, P \models [\alpha]\psi$. Suppose that exists such $P'$. Then for any such $P'$ we have $\mathcal{M}, P' \models \phi \to \psi$ and $\mathcal{M}, P' \models \phi$. Hence $\mathcal{M}, P' \models \psi$, i.e. $\mathcal{M}, P \models [\alpha]\psi$. □

**Lemma 4.5.13** (Soundness of axiom D9)**.** $\models 0 \to [\alpha]\bot$

*Proof.* If $\mathcal{M}, P \models 0$ then $P \equiv 0$ and there is no transition $0 \xrightarrow{\alpha} P'$, hence $\mathcal{M}, P \not\models \langle\alpha\rangle\top$, i.e. $\mathcal{M}, P \models [\alpha]\bot$. □

**Lemma 4.5.14** (Soundness of axiom D10)**.**

$$\textit{If } \beta \neq \alpha_i \textit{ for } i = 1..n, \textit{ then } \models \langle!\alpha_1\rangle\top|...|\langle!\alpha_n\rangle\top \to [\beta]\bot$$

*Proof.* Suppose that $\mathcal{M}, P \models \langle!\alpha_1\rangle\top|...|\langle!\alpha_n\rangle\top$. Then necessarily $P \equiv \alpha_1.P_1|...|\alpha_n.P_n$. But if $\alpha_i \neq \beta$ for $i = 1..n$, there is no transition

$$\alpha_1.P_1|...|\alpha_n.P_n \xrightarrow{\beta} P'$$

hence $\mathcal{M}, P \not\models \langle\beta\rangle\top$, i.e. $\mathcal{M}, P \models [\beta]\bot$. □

**Lemma 4.5.15** (Soundness of axiom D11)**.** $\models \langle!\alpha\rangle\phi \to [\alpha]\phi$

*Proof.* Suppose that $\mathcal{M}, P \models \langle!\alpha\rangle\phi$, then $\mathcal{M}, P \models 1$ and $\mathcal{M}, P \models \langle\alpha\rangle\phi$. Then necessarily $P \equiv \alpha.P'$ and $\mathcal{M}, P' \models \phi$. But there is only one reduction that $P$ can do, $P \xrightarrow{\alpha} P'$. So, for any reduction $P \xrightarrow{\alpha} P''$ (because there is only one), we have $\mathcal{M}, P'' \models \phi$, i.e. $\mathcal{M}, P \models [\alpha]\phi$ □

**Lemma 4.5.16** (Soundness of rule $\mathrm{D}_R2$)**.** *If* $\models \phi$ *then* $\models [\alpha]\phi$.

*Proof.* Let $\mathcal{M}$ be a context and $P \in \mathcal{M}$ a process. If there is no $P'$ such that $P \xrightarrow{\alpha} P'$, then $\mathcal{M}, P \models [\alpha]\phi$. Suppose that exists such $P'$ (obviously $P' \in \mathcal{M}$). Then for any such $P'$ we have $\mathcal{M}, P' \models \phi$, due to the hypothesis $\models \phi$. Hence $\mathcal{M}, P \models [\alpha]\phi$. □

**Lemma 4.5.17** (Soundness of rule $\mathrm{D}_R3$)**.**

$$\textit{If } \models \phi \to [\alpha]\phi' \textit{ and } \models \psi \to [\alpha]\psi' \textit{ then } \models \phi|\psi \to [\alpha](\phi'|\psi \vee \phi|\psi')$$

*Proof.* Suppose that $\mathcal{M}, P \models \phi|\psi$, then $P \equiv Q|R$, $\mathcal{M}, Q \models \phi$ and $\mathcal{M}, R \models \psi$. Because $\models \phi \to [\alpha]\phi'$ and $\models \psi \to [\alpha]\psi'$, we derive $\mathcal{M}, Q \models [\alpha]\phi'$ and $\mathcal{M}, R \models [\alpha]\psi'$. We analyze some cases:

- if $P$ cannot perform a transition by $\alpha$, then $\mathcal{M}, P \models [\alpha]\bot$, and using the soundness of axiom D8 and rule $\mathrm{D}_R2$ we derive

  $$\models [\alpha]\bot \to [\alpha](\phi'|\psi \vee \phi|\psi')$$

  hence, we obtain in the end $\mathcal{M}, P \models [\alpha](\phi'|\psi \vee \phi|\psi')$.

- if $Q \xrightarrow{\alpha} Q'$ and $R$ cannot perform a transition by $\alpha$, then $Q|R \xrightarrow{\alpha} Q'|R$ and the transitions of $P \equiv Q|R$ by $\alpha$ have always this form.
  But $\mathcal{M}, Q \models [\alpha]\phi'$, so for any such $Q'$ we have $\mathcal{M}, Q' \models \phi'$, thus $\mathcal{M}, Q'|R \models \phi'|\psi$, i.e. $\mathcal{M}, Q'|R \models (\phi'|\psi \vee \phi|\psi')$.
  Hence for any transition $P \xrightarrow{\alpha} P'$ we have $\mathcal{M}, P' \models (\phi'|\psi \vee \phi|\psi')$. In conclusion, $\mathcal{M}, P \models [\alpha](\phi'|\psi \vee \phi|\psi')$.

- if $Q$ cannot perform a transition by $\alpha$ and $R \xrightarrow{\alpha} R'$, similarly as in the previous case, we can derive $\mathcal{M}, P \models [\alpha](\phi'|\psi \vee \phi|\psi')$.

- if $Q \xrightarrow{\alpha} Q'$ and $R \xrightarrow{\alpha} R'$ then $P \xrightarrow{\alpha} P'$ has either the form $Q|R \xrightarrow{\alpha} Q'|R$ or $Q|R \xrightarrow{\alpha} Q|R'$. But $\mathcal{M}, Q'|R \models \phi'|\psi$, hence $\mathcal{M}, Q'|R \models (\phi'|\psi \vee \phi|\psi')$ and $\mathcal{M}, Q|R' \models \phi|\psi'$, hence $\mathcal{M}, Q|R' \models (\phi'|\psi \vee \phi|\psi')$. Thus, for any transition $P \xrightarrow{\alpha} P'$ we have $\mathcal{M}, P' \models (\phi'|\psi \vee \phi|\psi')$, i.e. $\mathcal{M}, P \models [\alpha](\phi'|\psi \vee \phi|\psi')$.

So, in any case $\mathcal{M}, P \models [\alpha](\phi'|\psi \vee \phi|\psi')$, that concludes the proof. □

**Lemma 4.5.18** (Soundness of rule $\mathrm{D}_R4$)**.**

$$\textit{If } \models \bigvee_{[\![Q]\!] \leq (\!|\phi|\!)} c_Q \to \phi \textit{ then } \models \phi$$

*Proof.* Suppose that $\models \bigvee_{[\![Q]\!] \leq (\!|\phi|\!)} c_Q \to \phi$ but exists a model $\mathcal{M}$ and a process $P \in \mathcal{M}$ with $\mathcal{M}, P \not\models \phi$. Then $\mathcal{M}, P \models \neg\phi$. Further, using the finite model property, theorem 4.3.3, we obtain that $\mathcal{M}_{(\!|\neg\phi|\!)}, P_{(\!|\neg\phi|\!)} \models \neg\phi$. But $(\!|\phi|\!) = (\!|\neg\phi|\!)$, so $\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \neg\phi$.
Further, because $\models \bigvee_{[\![Q]\!] \leq (\!|\phi|\!)} c_Q \to \phi$, we have

$$\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \bigvee_{[\![Q]\!] \leq (\!|\phi|\!)} c_Q \to \phi$$

But $[\![P_{(\!|\phi|\!)}]\!] \leq (\!|\phi|\!)$, and as $\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models c_{P_{(\!|\phi|\!)}}$, we obtain

$$\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \bigvee_{[\![Q]\!] \leq (\!|\phi|\!)} c_Q$$

Further $\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \phi$, so $\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \bot$ - impossible!
Hence for any model $\mathcal{M}$ and any process $P \in \mathcal{M}$ we have $\mathcal{M}, P \models \phi$. But this means $\models \phi$. □

## 4.6 Theorems of $\mathcal{L}_{DS}$

*A mathematician is a device*
*for turning coffee into theorems.*
Paul Erdos

We proved, in the previous section, that our axiomatic system is sound with respect to the process semantics, hence any provable result is a sound result, i.e. it says something true about processes. In this section we will prove some interesting theorems in $\mathcal{L}_{DS}$ and eventually we will interpret the nontrivial ones in the process semantics.

### 4.6.1 Spatial results

We start with the results that can be proved on the basis of the spatial theorems and rules only. They reflect the behavior of the parallel operator in relation to the operators of the classical logic.

**Theorem 4.6.1.** $\vdash \top|\top \leftrightarrow \top$

*Proof.* Obviously $\vdash \top|\top \rightarrow \top$. As $\vdash 0 \rightarrow \top$, using rule $\mathrm{D}_R1$, we obtain $\vdash \top|0 \rightarrow \top|\top$. Further axiom D2 gives us $\vdash \top \rightarrow \top|\top$. □

**Theorem 4.6.2.** *If* $\vdash \phi$ *then* $\vdash \theta|\rho \rightarrow \phi|\rho$

*Proof.* Because $\vdash \phi$ implies $\vdash \theta \rightarrow \phi$, using rule $\mathrm{D}_R1$ we obtain the result. □

**Theorem 4.6.3.** $\vdash \phi|\psi \leftrightarrow \psi|\phi$

*Proof.* We use axiom D3 in both directions. □

**Theorem 4.6.4.** $\vdash (\phi|\psi)|\rho \leftrightarrow \phi|(\psi|\rho)$

*Proof.* We use axiom D4 and theorem 4.6.3. □

**Theorem 4.6.5.** $\vdash \phi|(\psi \vee \rho) \leftrightarrow (\phi|\psi) \vee (\phi|\rho)$

*Proof.* $\vdash \psi \rightarrow \psi \vee \rho$ so, using rule $\mathrm{D}_R1$, $\vdash \phi|\psi \rightarrow \phi|(\psi \vee \rho)$. Similarly, $\vdash \phi|\rho \rightarrow \phi|(\psi \vee \rho)$. Hence $\vdash (\phi|\psi) \vee (\phi|\rho) \rightarrow \phi|(\psi \vee \rho)$. The other direction is stated by axiom D5. □

**Theorem 4.6.6.** $\vdash \phi|(\psi \wedge \rho) \rightarrow (\phi|\psi) \wedge (\phi|\rho)$

*Proof.* Because $\vdash \psi \wedge \rho \rightarrow \psi$, by applying rule $\mathrm{D}_R1$, we have $\vdash \phi|(\psi \wedge \rho) \rightarrow \phi|\psi$. Similarly $\vdash \phi|(\psi \wedge \rho) \rightarrow \phi|\rho$. □

The next result proves a strong version of monotonicity of the parallel composition.

**Theorem 4.6.7.** *If* $\vdash \phi \rightarrow \rho$ *and* $\vdash \psi \rightarrow \theta$ *then* $\vdash \phi|\psi \rightarrow \rho|\theta$.

*Proof.* If $\vdash \phi \rightarrow \rho$ then rule $\mathrm{D}_R1$ gives us $\vdash \phi|\psi \rightarrow \rho|\psi$. If $\vdash \psi \rightarrow \theta$, then the same rule gives $\vdash \rho|\psi \rightarrow \rho|\theta$. Hence $\vdash \phi|\psi \rightarrow \rho|\theta$. □

The next result speaks about the negative parallel decomposition of a specification. It states that, given two specifications, $\phi$ and $\psi$, if considering any parallel decomposition of our system (process) $P \equiv Q|R$, we obtain that either $Q$ doesn't satisfy $\phi$ or $R$ doesn't satisfy $\psi$, then our system $P$ does not satisfy the parallel composition of the two specifications, $\phi|\psi$.

**Theorem 4.6.8.** *If for any decomposition $P \equiv Q|R$ we have $\vdash c_Q \rightarrow \neg\phi$ or $\vdash c_R \rightarrow \neg\psi$ then $\vdash c_P \rightarrow \neg(\phi|\psi)$.*

*Proof.* $\vdash c_Q \rightarrow \neg\phi$ is equivalent with $\vdash c_Q \wedge \phi \rightarrow \bot$ and because $\vdash c_R \wedge \psi \rightarrow \top$, we obtain, by theorem 4.6.7 $\vdash (c_Q \wedge \phi)|(c_R \wedge \psi) \rightarrow \bot|\top$. And using axiom D1, we derive

$$\vdash (c_Q \wedge \phi)|(c_R \wedge \psi) \rightarrow \bot$$

Similarly, from $\vdash c_R \rightarrow \neg\psi$ we can derive

$$\vdash (c_Q \wedge \phi)|(c_R \wedge \psi) \rightarrow \bot$$

Hence, the hypothesis of the theorem says that for any decomposition $P \equiv Q|R$ we have $\vdash (c_Q \wedge \phi)|(c_R \wedge \psi) \rightarrow \bot$, i.e.

$$\vdash \bigvee_{P \equiv Q|R} (c_Q \wedge \phi)|(c_R \wedge \psi) \rightarrow \bot$$

But axiom D6 gives

$$\vdash (c_P \wedge \phi|\psi) \rightarrow \bigvee_{P \equiv Q|R} (c_Q \wedge \phi)|(c_R \wedge \psi)$$

hence

$$\vdash (c_P \wedge \phi|\psi) \rightarrow \bot, \ i.e. \ \vdash c_P \rightarrow \neg(\phi|\psi).$$

□

*Remark* 4.6.1. Related to the same topic of the relation between negation and the parallel operator, observe that the negation is not distributive with respect to parallel. This is the reason why, in the previous theorem, we had to ask in the premises that the condition $\vdash c_Q \to \neg\phi$ or $\vdash c_R \to \neg\psi$ be fulfilled by all the possible decompositions of $P$. If only a decomposition $P \equiv Q|R$ exists such that $\vdash c_Q \to \neg\phi$ or $\vdash c_R \to \neg\psi$, this is not enough to derive $\mathcal{M}, P \models \neg(\phi|\psi)$. Indeed suppose that $\mathcal{M}, Q \models \phi$ but $\mathcal{M}, Q \not\models \psi$ and $\mathcal{M}, R \models \psi$ but $\mathcal{M}, R \not\models \phi$. Then from $\mathcal{M}, Q \models \phi$ and $\mathcal{M}, R \models \psi$ we derive $\mathcal{M}, P \models \phi|\psi$. It is not the case that, from the additional information $\mathcal{M}, Q \not\models \psi$ and $\mathcal{M}, R \not\models \phi$, $\mathcal{M}, P \models \neg(\phi|\psi)$ to be derived. All we can derive from the unused information is that $\mathcal{M}, P \models \neg\phi|\neg\psi$, which does not contradict $\mathcal{M}, P \models \phi|\psi$.

### 4.6.2 Dynamic results

Now we focus of the theorems that derive from the class of dynamic axioms and rules. Remark the *modal behaviors* of the epistemic operators.

The next result states the monotonicity of the diamond operator.

**Theorem 4.6.9** (Monotonicity)**.** *If* $\vdash \phi \to \psi$ *then* $\vdash \langle\alpha\rangle\phi \to \langle\alpha\rangle\psi$.

*Proof.* $\vdash \phi \to \psi$ implies $\vdash \neg\psi \to \neg\phi$. Using rule $\mathrm{D}_R2$ we obtain $\vdash [\alpha](\neg\psi \to \neg\phi)$ and axiom D8 gives $\vdash [\alpha]\neg\psi \to [\alpha]\neg\phi$. This is equivalent with $\vdash \neg\langle\alpha\rangle\psi \to \neg\langle\alpha\rangle\phi$, i.e. $\vdash \langle\alpha\rangle\phi \to \langle\alpha\rangle\psi$. □

**Theorem 4.6.10.** *If* $\vdash \phi \to \psi$ *then* $\vdash [\alpha]\neg\psi \to [\alpha]\neg\phi$.

*Proof.* If $\vdash \phi \to \psi$ then, by theorem 4.6.9, $\vdash \langle\alpha\rangle\phi \to \langle\alpha\rangle\psi$, hence $\vdash \neg\langle\alpha\rangle\psi \to \neg\langle\alpha\rangle\phi$, that gives $\vdash [\alpha]\neg\psi \to [\alpha]\neg\phi$. □

The next theorems confirm the intuition that the formulas $c_P$, in their interrelations, mimic the transitions of the processes (the dynamic operators mimic the transition labeled by the action it has as index).

**Theorem 4.6.11.** *If $P$ cannot do any transition by $\alpha$ then $\vdash c_P \to [\alpha]\bot$.*

*Proof.* We prove it by induction on the structure of $P$.
**The case $P \equiv 0$:** axiom D9 implies $\vdash 0 \to [\alpha]\bot$ which proves this case, because $c_0 = 0$.
**The case $P \equiv \alpha_1.P_1|...|\alpha_n.P_n$:** as $P$ cannot perform $\alpha$ we have $\alpha \neq \alpha_i$ for $i = 1..n$. We have $c_P = (\langle\alpha_1\rangle c_{P_1} \wedge 1)|...|(\langle\alpha_n\rangle c_{P_n} \wedge 1)$. From $\vdash c_{P_i} \to \top$ we derive, using theorem 4.6.9, $\vdash (\langle\alpha_i\rangle c_{P_i} \wedge 1) \to (\langle\alpha_i\rangle\top \wedge 1)$. Further, we apply theorem 4.6.7 and obtain $\vdash c_P \to (\langle\alpha_1\rangle\top \wedge 1)|...|(\langle\alpha_n\rangle\top \wedge 1)$. Axiom D10 gives that for $\alpha \neq \alpha_i$, $\vdash (\langle\alpha_1\rangle\top \wedge 1)|...|(\langle\alpha_n\rangle\top \wedge 1) \to [\alpha]\bot$. Hence $\vdash c_P \to [\alpha]\bot$. □

**Theorem 4.6.12.** $\vdash c_P \to [\alpha]\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\}$

*Proof.* We prove it by induction on $P$.

**The case $P \not\equiv \alpha.P'|P''$ for some $P', P''$:** then $P$ cannot preform a transition by $\alpha$, hence, by theorem 4.6.11, $\vdash c_P \to [\alpha]\bot$. But $\vdash \neg\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \to \top$, and using theorem 4.6.10, we derive

$$\vdash [\alpha]\bot \to [\alpha]\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\}$$

Combining this with $\vdash c_P \to [\alpha]\bot$, we derive

$$\vdash c_P \to [\alpha]\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\}$$

**The case $P \equiv \alpha.P'$:** then $\{c_Q \mid P \xrightarrow{\alpha} Q\} = \{c_{P'}\}$ and $c_P = \langle\alpha\rangle c_{P'} \wedge 1$. Applying axiom D11 we obtain $\vdash c_P \rightarrow [\alpha]c_{P'}$. Hence

$$\vdash c_P \rightarrow [\alpha] \bigvee \{c_Q \mid P \xrightarrow{\alpha} Q\}$$

**The case $P \equiv \alpha.P'|P''$ with $P'' \not\equiv 0$:** we apply the inductive hypothesis to $\alpha.P'$ and $P''$ respectively, and we obtain

$$\vdash c_{\alpha.P'} \rightarrow [\alpha] \bigvee \{c_{Q'} \mid \alpha.P' \xrightarrow{\alpha} Q'\}$$

and

$$\vdash c_{P''} \rightarrow [\alpha] \bigvee \{c_{Q''} \mid P'' \xrightarrow{\alpha} Q''\}$$

We apply rule $\mathrm{D}_R3$ and obtain

$$\vdash c_P \rightarrow [\alpha](c_{\alpha.P'}| \bigvee \{c_{Q''} \mid P'' \xrightarrow{\alpha} Q''\} \vee \bigvee \{c_{Q'} \mid \alpha.P' \xrightarrow{\alpha} Q'\}|c_{P''})$$

Using theorem 4.6.5, we obtain this result equivalent with

$$\vdash c_P \rightarrow [\alpha] \bigvee \{c_Q \mid P \xrightarrow{\alpha} Q\}$$

□

**Theorem 4.6.13.** *If* $\vdash \bigvee \{c_Q \mid P \xrightarrow{\alpha} Q\} \rightarrow \phi$ *then* $\vdash c_P \rightarrow [\alpha]\phi$

*Proof.* If $\vdash \bigvee \{c_Q \mid P \xrightarrow{\alpha} Q\} \rightarrow \phi$ then rule $\mathrm{D}_R2$ gives

$$\vdash [\alpha](\bigvee \{c_Q \mid P \xrightarrow{\alpha} Q\} \rightarrow \phi)$$

and further axiom D8 gives $\vdash [\alpha] \bigvee \{c_Q \mid P \xrightarrow{\alpha} Q\} \rightarrow [\alpha]\phi$. But theorem 4.6.12 gives $\vdash c_P \rightarrow [\alpha] \bigvee \{c_Q \mid P \xrightarrow{\alpha} Q\}$, hence $\vdash c_P \rightarrow [\alpha]\phi$. □

## 4.7 Characteristic formulas

In this section we will focus on the class $(c_P)_{P \in \mathfrak{P}}$ of formulas and we will prove that, indeed, they characterize, up to structural congruence, their indexes (processes). Hence they provide univocal syntactical descriptions for the $\equiv$-equivalence classes of processes. We will use this peculiarity of our syntax, in the next section, for proving the completeness of our system with respect to process semantics.

We begin by restating some relevant results, proved before, in order to offer to the reader a full picture of the problem.

**Theorem 4.7.1.** $\mathcal{M}, P \models c_P$.

*Proof.* It has been proved as theorem 4.5.7. □

**Theorem 4.7.2.** $\mathcal{M}, P \models c_Q$ *iff* $P \equiv Q$.

*Proof.* It has been proved as theorem 4.5.8. □

The next theorems show that $c_P$ could provide a syntactic characterization of the process $P$, stating that the conjunction of two such formulas, $c_P$ and $c_Q$, is inconsistent if the indexes are not structurally congruent, and respectively that two structurally congruent indexes generate logical equivalent formulas.

**Theorem 4.7.3.** *If* $P \not\equiv Q$ *then* $\vdash c_P \rightarrow \neg c_Q$.

*Proof.* We prove it by induction on $P$.

- **the case** $P \equiv 0$**:** as $P \not\equiv Q$ we obtain that $Q \equiv \alpha.R|S$. So $c_Q = \langle\alpha\rangle c_R \wedge 1|c_S$ that implies, using theorem 4.6.6, $\vdash c_Q \rightarrow \langle\alpha\rangle c_R|c_S$, and applying axiom D7, $\vdash c_Q \rightarrow \langle\alpha\rangle(c_R|c_S)$.
  But $\vdash c_R|c_S \rightarrow \top$ and applying theorem 4.6.9, we obtain
  $\vdash \langle\alpha\rangle(c_R|c_S) \rightarrow \langle\alpha\rangle\top$.
  Hence, $\vdash c_Q \rightarrow \langle\alpha\rangle\top$. Then $\vdash \neg\langle\alpha\rangle\top \rightarrow \neg c_Q$.
  Axiom D9 gives $\vdash 0 \rightarrow \neg\langle\alpha\rangle\top$ hence, in the end, $\vdash 0 \rightarrow \neg c_Q$, i.e. $\vdash c_P \rightarrow \neg c_Q$.

- **the case** $P \equiv P'|P''$**:** we have $c_P = c_{P'}|c_{P''}$. Because $P \not\equiv Q$, we obtain that for any decomposition $Q \equiv Q'|Q''$ we have either $P' \not\equiv Q'$ or $P'' \not\equiv Q''$. Using the inductive hypothesis, we derive that either $\vdash c_{Q'} \rightarrow \neg c_{P'}$ or $\vdash c_{Q''} \rightarrow \neg c_{P''}$. Because this is happening for any decomposition of $Q$, we can apply theorem 4.6.8 and we obtain
  $\vdash c_Q \rightarrow \neg(c_{P'}|c_{P''})$, i.e. $\vdash c_Q \rightarrow \neg c_P$. Hence $\vdash c_P \rightarrow \neg c_Q$.

- **the case** $P \equiv \alpha.P'$**:** $c_P = 1 \wedge \langle\alpha\rangle c_{P'}$, so $\vdash c_P \rightarrow 1 \wedge \langle\alpha\rangle\top$.
  But axiom D10 gives $\vdash \langle\alpha\rangle\top \wedge 1 \rightarrow \neg\langle\beta\rangle\top$ for any $\beta \neq \alpha$.
  Hence, for any $\beta \neq \alpha$ we have $\vdash c_P \rightarrow \neg\langle\beta\rangle\top$.

  - if $Q \equiv 0$ we already proved that $\vdash c_Q \rightarrow \neg c_P$ (because $P \not\equiv 0$), so $\vdash c_P \rightarrow \neg c_Q$
  - if $Q \equiv \beta.Q'|Q''$ for some $\beta \neq \alpha$, then $\vdash c_Q \rightarrow \langle\beta\rangle\top$, hence $\vdash \neg\langle\beta\rangle\top \rightarrow \neg c_Q$. But we proved that $\vdash c_P \rightarrow \neg\langle\beta\rangle\top$. Hence $\vdash c_P \rightarrow \neg c_Q$.
  - if $Q \equiv \alpha.Q_1|...|\alpha.Q_k$ for $k > 1$, then $\vdash c_Q \rightarrow \neg 0|\neg 0$ (as $\vdash 0 \rightarrow \neg c_{\alpha.Q_1}$ and $\vdash 0 \rightarrow \neg c_{\alpha.Q_2|...|\alpha.Q_k}$). Then $\vdash c_Q \rightarrow \neg 1$, i.e.
    $\vdash 1 \rightarrow \neg c_Q$. But $\vdash c_P \rightarrow 1$. Hence $\vdash c_P \rightarrow \neg c_Q$.
  - if $Q \equiv \alpha Q'$: then $P \not\equiv Q$ gives $P' \not\equiv Q'$. For this case we can use the inductive hypothesis and we obtain $\vdash c_{Q'} \rightarrow \neg c_{P'}$. Further,

applying theorem 4.6.10, we obtain $\vdash [\alpha]c_{P'} \rightarrow [\alpha]\neg c'_Q$, i.e.
$\vdash [\alpha]c_{P'} \rightarrow \neg\langle\alpha\rangle c_{Q'}$ that gives, because $c_Q = 1 \wedge \langle\alpha\rangle c_{Q'}$,
$\vdash [\alpha]c_{P'} \rightarrow \neg c_Q$.
Now, using axiom D11, $\vdash 1 \wedge \langle\alpha\rangle c_{P'} \rightarrow [\alpha]c_{P'}$, so $\vdash c_P \rightarrow [\alpha]c_{P'}$,
and, combining it with the previous result, we derive $\vdash c_P \rightarrow \neg c_Q$.

□

**Theorem 4.7.4.** *If $P \equiv Q$ then $\vdash c_P \leftrightarrow c_Q$.*

*Proof.* We prove it verifying the congruence rules:

- if $P = R|S$ and $Q = S|R$ then $\vdash c_R|c_S \leftrightarrow c_S|c_R$ from theorem 4.6.3, i.e. $\vdash c_P \leftrightarrow c_Q$
- if $P = (R|S)|U$ and $Q = R|(S|U)$ then theorem 4.6.4 we have $\vdash (c_R|c_S)|c_U \leftrightarrow c_R|(c_S|c_U)$, i.e. $\vdash c_P \leftrightarrow c_Q$
- if $P = Q|0$ then axiom D2 gives $\vdash c_Q|0 \leftrightarrow c_Q$, i.e. $\vdash c_P \leftrightarrow c_Q$.
- if $P = P'|R$ and $Q = Q'|R$ with $P' \equiv Q'$ and $\vdash c_{P'} \leftrightarrow c_{Q'}$ then rule $\mathrm{D}_R1$ gives $\vdash c_{P'}|c_R \leftrightarrow c_{Q'}|c_R$. Hence $\vdash c_P \leftrightarrow c_Q$.
- if $P = \alpha.P'$ and $Q = \alpha.Q'$ with $P' \equiv Q'$ and $\vdash c_{P'} \leftrightarrow c_{Q'}$ then theorem 4.6.9 gives $\vdash \langle\alpha\rangle c_{P'} \leftrightarrow \langle\alpha\rangle c_{Q'}$, so $\vdash (\langle\alpha\rangle c_{P'} \wedge 1) \leftrightarrow (\langle\alpha\rangle c_{Q'} \wedge 1)$. Hence $\vdash c_P \leftrightarrow c_Q$.

□

We will use, now, the characteristic formula to obtain a syntactic characterization of the satisfiability relation. The intuition is that, as far as a

process $P$ in a context can be characterized by the formula $c_P$, it is expected that $\mathcal{M}, P \models \phi$ and $\vdash c_P \to \phi$ are equivalent. The last relation, if provable (hence sound), states that if a process satisfies $c_P$ then it also satisfies $\phi$. But a process satisfies $c_P$ only if it belongs to the $\equiv$-equivalence class of $P$. But $\mathcal{M}, P \models \phi$ states exactly the same thing!

In the next lemma we will prove that this intuition is correct.

**Lemma 4.7.5** (Syntactic characterization of satisfiability)**.**

*If $\mathcal{M}$ is a context and $P \in \mathcal{M}$, then $\mathcal{M}, P \models \phi$ iff $\vdash c_P \to \phi$*

*Proof.* ($\Longrightarrow$) We prove it by induction on the syntactical structure of $\phi$.

- **The case $\phi = 0$:** $\mathcal{M}, P \models 0$ implies $P \equiv 0$. But $c_0 = 0$ and $\vdash 0 \to 0$, hence $\vdash c_P \to \phi$.
- **The case $\phi = \top$:** we have always $\mathcal{M}, P \models \top$, and always $\vdash c_P \to \top$.
- **The case $\phi = \phi_1 \wedge \phi_2$:** $\mathcal{M}, P \models \phi$ iff $\mathcal{M}, P \models \phi_1$ and $\mathcal{M}, P \models \phi_2$.
  Using the inductive hypothesis, we obtain $\vdash c_P \to \phi_1$ and $\vdash c_P \to \phi_2$.
  Hence $\vdash c_P \to (\phi_1 \wedge \phi_2)$, i.e. $\vdash c_P \to \phi$.
- **The case $\phi = \phi_1 | \phi_2$:** $\mathcal{M}, P \models \phi$ iff $P \equiv Q|R$, $\mathcal{M}, Q \models \phi_1$ and $\mathcal{M}, R \models \phi_2$.
  Using the inductive hypothesis, $\vdash c_Q \to \phi_1$ and $\vdash c_R \to \phi_2$.
  Hence, using theorem 4.6.7 $\vdash (c_Q | c_R) \to (\phi_1 | \phi_2)$, i.e. $\vdash c_P \to \phi$.
- **The case $\phi = \langle\alpha\rangle\psi$:** $\mathcal{M}, P \models \langle\alpha\rangle\psi$ means that exists $P' \in \mathcal{M}$ such that $P \xrightarrow{\alpha} P'$ and $\mathcal{M}, P' \models \psi$. Then the inductive hypothesis gives $\vdash c_{P'} \to \psi$.

  But $P \xrightarrow{\alpha} P'$ means that $P \equiv \alpha.R|S$ and $P' \equiv R|S$, so
  $c_P = (\langle\alpha\rangle c_R \wedge 1)|c_S$ and $c_{P'} = c_R|c_S$.

Then $\vdash c_{P'} \to \psi$ is equivalent with $\vdash c_R|c_S \to \psi$. Further, using theorem 4.6.9, we obtain $\vdash \langle\alpha\rangle(c_R|c_S) \to \langle\alpha\rangle\psi$.

As $c_P = (\langle\alpha\rangle c_R \wedge 1)|c_S$ theorem 4.6.6 gives $\vdash c_P \to (\langle\alpha\rangle c_R|c_S) \wedge (1|c_S)$, hence $\vdash c_P \to \langle\alpha\rangle c_R|c_S$.

Further, axiom D7 gives $\vdash \langle\alpha\rangle c_R|c_S \to \langle\alpha\rangle(c_R|c_S)$.

Hence we proved that $\vdash c_P \to \langle\alpha\rangle c_R|c_S$, that $\vdash \langle\alpha\rangle c_R|c_S \to \langle\alpha\rangle(c_R|c_S)$ and that $\vdash \langle\alpha\rangle(c_R|c_S) \to \langle\alpha\rangle\psi$. Hence $\vdash c_P \to \langle\alpha\rangle\psi$ q.e.d.

- **The case $\phi = \neg\psi$:** we argue by induction on the syntactical structure of $\psi$.
  - **the subcase $\psi = 0$:** $\mathcal{M}, P \models \neg 0$ means that $P \not\equiv 0$, and using theorem 4.7.3, $\vdash c_P \to \neg c_0$, i.e. $\vdash c_P \to \neg 0$, q.e.d.
  - **the subcase $\psi = \top$:** is an impossible one as we cannot have $\mathcal{M}, P \models \neg\top$, equivalent with $\mathcal{M}, P \not\models \top$.
  - **the subcase $\psi = \psi_1 \wedge \psi_2$:** $\mathcal{M}, P \models \neg(\psi_1 \wedge \psi_2)$ is equivalent with $\mathcal{M}, P \models \neg\psi_1 \vee \neg\psi_2$, i.e. $\mathcal{M}, P \models \neg\psi_1$ or $\mathcal{M}, P \models \neg\psi_2$.
    By the inductive hypothesis, $\vdash c_P \to \neg\psi_1$ or $\vdash c_P \to \neg\psi_2$, where from we obtain $\vdash c_P \to \neg\psi_1 \vee \neg\psi_2$, i.e. $\vdash c_P \to \neg(\psi_1 \wedge \psi_2)$, q.e.d.
  - **the subcase $\psi = \neg\psi_1$:** $\mathcal{M}, P \models \neg\psi$ is equivalent with $\mathcal{M}, P \models \neg\neg\psi_1$, i.e. $\mathcal{M}, P \models \psi_1$ where we can use the inductive hypothesis $\vdash c_P \to \psi_1$ equivalent with $\vdash c_P \to \phi$.
  - **the subcase $\psi = \psi_1|\psi_2$:** $\mathcal{M}, P \models \neg(\psi_1|\psi_2)$ means that for any parallel decomposition of $P \equiv Q|R$, $\mathcal{M}, Q \not\models \psi_1$ (i.e. $\mathcal{M}, Q \models \neg\psi_1$) or $\mathcal{M}, R \not\models \psi_2$ (i.e. $\mathcal{M}, R \models \neg\psi_2$).
    These implies, using the inductive hypothesis, that for any decomposition $P \equiv Q|R$, $\vdash c_Q \to \neg\psi_1$ or $\vdash c_R \to \neg\psi_2$.
    Further, applying theorem 4.6.8, we obtain $\vdash c_P \to \neg(\psi_1|\psi_2)$, q.e.d.

– **the subcase** $\psi = \langle\alpha\rangle\psi_1$**:** $\mathcal{M}, P \models \neg\langle\alpha\rangle\psi_1$ is equivalent with $\mathcal{M}, P \models [\alpha]\neg\psi_1$.
If $P$ cannot perform $\alpha$, then, by theorem 4.6.11 $\vdash c_P \rightarrow [\alpha]\bot$ that implies further $\vdash c_P \rightarrow [\alpha]\neg\psi_1$ (because $\vdash \psi_1 \rightarrow \top$).
If $P$ can perform $\alpha$, then $\mathcal{M}, P \models [\alpha]\neg\psi_1$ implies that for any $Q \in \mathcal{M}$ with $P \xrightarrow{\alpha} Q$, $\mathcal{M}, Q \models \neg\psi_1$.
Using the inductive hypothesis we obtain that for any $Q \in \mathcal{M}$ such that $P \xrightarrow{\alpha} Q$ we have $\vdash c_Q \rightarrow \neg\psi_1$, i.e.
$\vdash \bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \rightarrow \neg\psi_1$.
Now, using theorem 4.6.13, we obtain $\vdash c_P \rightarrow [\alpha]\neg\psi_1$ q.e.d.

($\Longleftarrow$) Let $\vdash c_P \rightarrow \phi$ and $\mathcal{M}$ a context that contains $P$.
Suppose that $\mathcal{M}, P \not\models \phi$. Then $\mathcal{M}, P \models \neg\phi$. Using the reversed implication we obtain $\vdash c_P \rightarrow \neg\phi$, hence $\vdash c_P \rightarrow \bot$.
But $\mathcal{M}, P \models c_P$ which, using the soundness, gives $\mathcal{M}, P \models \bot$ impossible!
Hence $\mathcal{M}, P \models \phi$. □

The next corollary ensures us that $\mathcal{L}_{DS}$ is not sensitive to contexts. This is an expected result that can be corroborated by the fact that, in [15], it was proved that in spatial logics with dynamic operators the guarantee operator (dual of parallel) cannot be derived from the other operators. Now our next theorem explains why: because the dynamic spatial logic is not sensitive to contexts, while the guarantee operator is.

**Corollary 4.7.6.** *If $\mathcal{M}, \mathcal{M}'$ are contexts, $\mathcal{M}, P \models \phi$ and $P \in \mathcal{M}'$, then $\mathcal{M}', P \models \phi$.*

*Proof.* If $\mathcal{M}, P \models \phi$ then by lemma 4.7.5 we obtain $\vdash c_P \rightarrow \phi$. As $P \in \mathcal{M}'$ and $\vdash c_P \rightarrow \phi$ we can apply the same lemma once again and obtain $\mathcal{M}', P \models \phi$ q.e.d. □

**Corollary 4.7.7.** *If* $\vdash c_P \to c_Q$ *then* $P \equiv Q$.

*Proof.* If $\vdash c_P \to c_Q$ then, by lemma 4.7.5, $\mathcal{M}, P \models c_Q$, for any context $\mathcal{M} \ni P$, hence, by theorem 4.5.8 $P \equiv Q$. □

## 4.8 Completeness of $\mathcal{L}_{DS}$ against process semantics

In this section we will prove that our axiomatic system proposed for $\mathcal{L}_{DS}$ is a complete axiomatic system for process semantics. This means that everything that can be derived in semantics can be also proved, as a theorem, in our system. In this way we show that the axioms of our system are comprehensive enough to fully describe what can happen in the process calculus we considered.

This result, in relation to the soundness result proved in section 4.5 reveals a duality between our axiomatic system and the fragment of CCS we consider as semantics: everything that can be derived in semantics can be proved in the syntax, and everything that can be proved in the syntax can be derived in semantics.

**Definition 4.8.1** (Provability and consistency)**.** We say that a formula $\phi \in \mathcal{F}_{DS}$ is *provable in* $\mathcal{L}_{DS}$ (or $\mathcal{L}_{DS}$*-provable* for short), if $\phi$ can be derived, as a theorem, using the axioms and the rules of $\mathcal{L}_{DS}$. We denote this by $\vdash \phi$.
We say that a formula $\phi \in \mathcal{F}_{DS}$ is *consistent in* $\mathcal{L}_{DS}$ (or $\mathcal{L}_{DS}$*-consistent* for short) if $\neg\phi$ is not $\mathcal{L}_{DS}$-provable.

In the next lemma we will prove that the consistency is the syntactic counterpart of satisfiability. This will be eventually used to prove the completeness.

**Lemma 4.8.1.** *If $\phi$ is $\mathcal{L}_{DS}$-consistent then exists a context $\mathcal{M}$ and a process $P \in \mathcal{M}$ such that $\mathcal{M}, P \models \phi$.*

*Proof.* Suppose that $\phi$ is $\mathcal{L}_{DS}$-consistent, but for any context $\mathcal{M}$ and any process $P \in \mathcal{M}$ we do not have $\mathcal{M}, P \models \phi$, i.e. $\mathcal{M}, P \not\models \phi$.
Then for any process $P \in \mathfrak{P}$ and any context $\mathcal{M} \ni P$ we have $\mathcal{M}, P \models \neg\phi$. Further lemma 4.7.5 gives $\vdash c_P \to \neg\phi$, for any $P \in \mathfrak{P}$. Thus we also have $\vdash c_P \to \neg\phi$ for any $P$ such that $[\![P]\!] \leq (\!|\neg\phi|\!)$, i.e.

$$\vdash \bigvee_{[\![P]\!] \leq (\!|\neg\phi|\!)} c_P \to \neg\phi.$$

Further, using rule $\mathrm{D}_R4$, we obtain $\vdash \neg\phi$. This contradicts with the hypothesis of $\mathcal{L}_{DS}$-consistency of $\phi$.

In consequence, there exists a context $\mathcal{M}$ and a process $P \in \mathcal{M}$ such that $\mathcal{M}, P \models \phi$. □

**Theorem 4.8.2** (Completeness)**.** *The $\mathcal{L}_{DS}$ system is complete with respect to process semantics.*

*Proof.* Suppose that $\phi$ is a valid formula with respect to process semantics, but $\phi$ is not provable in the system $\mathcal{L}_{DS}$. Then neither is $\neg\neg\phi$, so, by definition, $\neg\phi$ is $\mathcal{L}_{DS}$-consistent. It follows from lemma 4.8.1 that $\neg\phi$ is satisfiable with respect to our semantics, contradicting the validity of $\phi$.
Hence, if $\phi$ is valid, then it is $\mathcal{L}_{DS}$-provable. □

## 4.9 Concluding remarks

In this chapter we introduced the Dynamic Spatial Logic, $\mathcal{L}_{DS}$, as an extension of the Hennessy-Milner logic with the parallel operator. This logic distinguishes processes up to structural congruence level.

We developed a Hilbert-style axiomatic system for it and we proved that the system is sound and complete with respect to process semantics.

We proved that $\mathcal{L}_{DS}$ against the process semantics satisfies the finite model property that entails decidability for satisfiability, validity and model-checking problems.

# Chapter 5

# Dynamic Epistemic Spatial Logic

*...you act, and you know why you act,*
*but you don't know why you know*
*that you know what you do.*
(Umberto Eco, *The name of the Rose*)

In this chapter we will introduce the first Dynamic Epistemic Spatial Logic, $\mathcal{L}^{\mathfrak{S}}_{DES}$, which extends $\mathcal{L}_{DS}$ with epistemic operators meant to express global properties over contexts. The intuition is to define the knowledge of the process $P$ in the context $\mathcal{M}$ as the common properties of the processes in $\mathcal{M}$ that contain $P$ as an active subprocess. Hence the knowledge implies a kind of universal quantifier over $\mathcal{M}$. We find this enough for expressing most of the properties considered in the spatial logic literature, which requires the use of the guarantee operator.

By using the *structural bisimulation* and *pruning method*, we will prove the finite model property for $\mathcal{L}^{\mathfrak{S}}_{DES}$ in relation to the semantics we considered. Consequently, we obtain decidability for satisfiability/validity and model checking.

For $\mathcal{L}^{\mathfrak{S}}_{DES}$ we will develop a Hilbert-style axiomatic system that will be proved to be sound and complete with respect to process semantics. Thus

we identify the main axioms and rules that regularize the behavior of the classical, spatial, dynamic and epistemic logical operators. We will stress the similarities between our axioms and the classical axioms of epistemic logic, and we will prove some meaningful theorems.

Combined with the decidability, the properties of soundness and completeness make our logic a useful tool in analyzing complex multi-agent systems.

## 5.1 The signature

To introduce epistemic operators into our syntax we need to specify, for the beginning, the *epistemic agents*. As in classic epistemic logic, we may start with a class of agents, each agent pointing to a predefined subsystem (subprocess) of the system we consider. In this respect, we should consider quite a large class of agents, also for the processes that are not active in the current state but might be activated in future.

Hence for a system containing an agent associated with the process $\alpha.P|Q$, we might want to have also agents associated with $\alpha.P$, $P$, $P|Q$ and $Q$ respectively.

To avoid a syntax that is too complex, we decided to identify the agents with the processes they represent. Hence, in our logic the class of epistemic agents is just a subclass of $\mathfrak{P}$. We will call this class *signature*, as it contains processes that will be part of the syntax as indexes of the epistemic agents. To denote the signature of our logic we will use the symbol $\mathfrak{S}$.

We associate to each process $P \in \mathfrak{S}$ an epistemic operator $K_P\phi$ meaning *the agent (process) $P$ knows $\phi$*. Obviously, for our agents the notion of knowledge is different than in the standard approaches to intelligent

agents, in the sense that we do not expect our agents to answer questions concerning their knowledge or to compute it. The knowledge of the agent $P$ in a context $\mathcal{M}$ is strictly related to the spectrum of actions $P$ can perform in this environment.

Anticipating the semantics, we will have $\mathcal{M}, Q \models K_P\phi$, i.e. in the state $Q$ of our system (which *"works"* in the context $\mathcal{M}$), $P$ *knows* $\phi$ iff

- $P$ is active in the state $Q$ (hence, following the definition of context, $P \in \mathcal{M}$ is a precondition) and
- if in any other state of $\mathcal{M}$, which *looks the same as $Q$ from the point of view of $P$* (i.e. where $P$ is active, because the only *"sensitivity"* that $P$ has concerns its own activity/inactivity), $\phi$ is satisfied.

Hence $\mathcal{M}, Q \models K_P\phi$ iff $P \equiv Q|R$ and for any process $Q|R' \in \mathcal{M}$ we have $\mathcal{M}, Q|R' \models \phi$.

Consider, for example, the next context:

$$\mathcal{M} = \{\alpha.0|\beta.0,\ \alpha.0|\gamma.0,\ \theta.\alpha.0,\ \alpha.0,\ \beta.0,\ \gamma.0,\ \theta.0,\ 0\}$$

Suppose that $\alpha.0 \in \mathfrak{S}$. The processes of $\mathcal{M}$ where $\alpha.0$ is active are

$$\{\alpha.0|\beta.0,\ \alpha.0|\gamma.0,\ \alpha.0\}$$

hence we can state $K_{\alpha.0}\phi$ only to one of these processes, and only if $\phi$ is satisfied by each of these processes.

$$\mathcal{M}, \alpha.0|\beta.0 \models K_{\alpha.0}\langle\alpha\rangle\top$$

because we have $\mathcal{M}, \alpha.0|\beta.0 \models \langle\alpha\rangle\top$, $\mathcal{M}, \alpha.0|\gamma.0 \models \langle\alpha\rangle\top$ and $\mathcal{M}, \alpha.0 \models \langle\alpha\rangle\top$. But $\mathcal{M}, \theta.\alpha.0 \not\models K_{\alpha.0}\langle\alpha\rangle\top$ because in $\theta.\alpha.0$ the agent $\alpha.0$ is not active.

$$\mathcal{M}, \alpha.0|\gamma.0 \models K_{\alpha.0}\neg\langle\gamma\rangle\top$$

but $\mathcal{M}, \alpha.0|\gamma.0 \models \langle\gamma\rangle\top$. Still, we do not have $\mathcal{M}, \alpha \models \langle\gamma\rangle\top$, and for this reason we cannot state $\mathcal{M}, \alpha.0|\gamma.0 \models K_{\alpha.0}\langle\gamma\rangle\top$.

In our approach an inactive agent does not have a knowledge. This is an expected fact, as an inactive agent does not exist. Indeed, approaching systems from the point of view of behavior, *to be* is *to behave*. This aspect is new for the class of epistemic logic where, always, all the agents exist and know at least the tautologies.

**Definition 5.1.1** (Signature)**.** *A signature over* $\mathfrak{P}$ is a set of processes $\mathfrak{S} \subset \mathfrak{P}$, hereafter called *epistemic agents*, satisfying the conditions:

- if $P|Q \in \mathfrak{S}$ then $P, Q \in \mathfrak{S}$
- if $P \in \mathfrak{S}$ and $P \longrightarrow Q$, then $Q \in \mathfrak{S}$

Observe that, by the previous definition, any signature $\mathfrak{S}$ contains 0.

## 5.2 Syntax of $\mathcal{L}^{\mathfrak{S}}_{DES}$

In this section we introduce the syntax of $\mathcal{L}^{\mathfrak{S}}_{DES}$. We will add to the operators of $\mathcal{L}_{DS}$ the epistemic operators indexed by the epistemic agents in the signature $\mathfrak{S}$.

**Definition 5.2.1** (Syntax of $\mathcal{L}^{\mathfrak{S}}_{DES}$)**.** Let $\mathfrak{S}$ be a signature over $\mathfrak{P}$. We define the language of Dynamic Epistemic Spatial Logic over $\mathfrak{S}$, $\mathcal{F}^{\mathfrak{S}}_{DES}$, by the following grammar:

$$\phi := \ 0 \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi|\phi \mid \langle\alpha\rangle\phi \mid K_Q\phi$$

where $Q \in \mathfrak{S}$ and $\alpha \in \mathbb{A}$.

We consider the derived operators proposed for $\mathcal{F}_{DS}$ to be already defined. In addition we propose the dual of the epistemic operator

$$\widetilde{K}_Q\phi \stackrel{def}{=} \neg K_Q \neg\phi$$

Further, the size of a formula in $\mathcal{F}_{DES}^{\mathfrak{S}}$ is defined. For this, we recall the definition 4.3.1, where the sizes of formulas in $\mathcal{F}_{DE}$ were defined. We only add the definition for the formulas involving the epistemic operator.

**Definition 5.2.2** (Size of a formula)**.** We extend the definition 4.3.1 with the case of epistemic operators.
Consider $R \in \mathfrak{S}$ and suppose that $[\![R]\!] = (h_R, w_R)$ and $(\!|\phi|\!) = (h, w)$. Then:

$$(\!|K_R\phi|\!) \stackrel{def}{=} (1 + max(h, h_R), 1 + max(w, w_R))$$

## 5.3 Extending the process semantics

The semantics of $\mathcal{L}_{DES}^{\mathfrak{S}}$ will be introduced also as an extension of the semantics of $\mathcal{L}_{DS}$. We will add only the definition of satisfaction for the epistemic operator.

**Definition 5.3.1.** A model of $\mathcal{L}_{DES}^{\mathfrak{S}}$ is a context $\mathcal{M}$ for which we define the satisfaction relation by extending the satisfaction relation defined for $\mathcal{L}_{DS}$ with the rule:

- $\mathcal{M}, P \models K_Q\phi$ iff $P \equiv Q|R$ and $\forall Q|R' \in \mathcal{M}$ we have $\mathcal{M}, Q|R' \models \phi$

In the light of this definition the semantics of the $\widetilde{K}_Q\phi$ operator will be:
$\mathcal{M}, P \models \widetilde{K}_Q\phi$ iff

- either $P \not\equiv Q|R$ for any $R$, or
- it exists $Q|S \in \mathcal{M}$ such that $\mathcal{M}, Q|S \models \phi$

Remark the interesting semantics of the operators $K_0$ and $\widetilde{K}_0$, because for any process we have $Q \equiv Q|0$:
$\mathcal{M}, P \models K_0\phi$ iff for any $Q \in \mathcal{M}$ we have $\mathcal{M}, Q \models \phi$
$\mathcal{M}, P \models \widetilde{K}_0\phi$ iff it exists a process $Q \in \mathcal{M}$ such that $\mathcal{M}, Q \models \phi$
If a process $P \in \mathcal{M}$ satisfies $K_0\phi$ then $\phi$ is valid in $\mathcal{M}$ (the same about $K_0\phi$) and vice versa. Hence we can encode, in the syntax, the validity with respect to a given context.
If a process $P \in \mathcal{M}$ satisfies $\widetilde{K}_0\phi$ (then all the processes in $\mathcal{M}$ satisfy $\widetilde{K}_0\phi$) then a process $Q \in \mathcal{M}$ exists that satisfies $\phi$ and vice versa. Hence $\widetilde{K}_0\phi$ provides a way to encode the satisfiability with respect to a given model.

## 5.4 Characterizing contexts

We now use the peculiarities of the operators $K_0$ and $\widetilde{K}_0$ to define characteristic formulas for finite contexts. Such formulas will be useful in proving, later, the finite model theory and the completeness for our system.

**Definition 5.4.1** (Characteristic formulas for contexts)**.** If $\mathcal{M}$ is a finite context, we define its *characteristic formula* by:

$$c_{\mathcal{M}} = K_0(\bigvee_{Q\in\mathcal{M}} c_Q) \wedge (\bigwedge_{Q\in\mathcal{M}} \widetilde{K}_0 c_Q) \tag{5.4.1}$$

We prove now that the intuition behind the previous definition is correct and, indeed, $c_{\mathcal{M}}$ can be used to characterize $\mathcal{M}$.

**Theorem 5.4.1.** *If $\mathcal{M}$ is a finite context and $P \in \mathcal{M}$ then $\mathcal{M}, P \models c_{\mathcal{M}}$.*

*Proof.* Obviously $\mathcal{M}, P \models c_P$, hence $\mathcal{M}, P \models \bigvee_{Q\in\mathcal{M}} c_Q$.
Similarly, for any $R \in \mathcal{M}$ we have $\mathcal{M}, R \models \bigvee_{Q\in\mathcal{M}} c_Q$, and because $R \equiv R|0$ and $P \equiv P|0$, we derive $\mathcal{M}, P \models K_0(\bigvee_{Q\in\mathcal{M}} c_Q)$.
As for any $R \in \mathcal{M}$ there exists a process $U \in \mathcal{M}$ (more exactly $U = R$) such that $\mathcal{M}, U \models c_R$, we obtain that for each $R \in \mathcal{M}$ we have $\mathcal{M}, P \models \widetilde{K}_0 c_R$, hence $\mathcal{M}, P \models \bigwedge_{Q\in\mathcal{M}} \widetilde{K}_0 c_Q$. □

**Corollary 5.4.2.** *If $\mathcal{M}$ is a finite context and $P \in \mathcal{M}$ then*

$$\mathcal{M}, P \models c_{\mathcal{M}} \wedge c_P$$

**Theorem 5.4.3.** *If $\mathcal{M}, P \models c_{\mathcal{N}}$ then $\mathcal{N} = \mathcal{M}$.*

*Proof.* Suppose that $\mathcal{M}, P \models c_{\mathcal{N}}$, then $\mathcal{M}, P \models K_0(\bigvee_{Q\in\mathcal{N}} c_Q)$, i.e. for any $R \in \mathcal{M}$ we have $\mathcal{M}, R \models \bigvee_{Q\in\mathcal{N}} c_Q$. Hence, for any $R \in \mathcal{M}$ there exists a process $Q \in \mathcal{N}$ with $\mathcal{M}, R \models c_Q$, or equivalently, $R \equiv Q$.
Now $\mathcal{M}, P \models \bigwedge_{Q\in\mathcal{N}} \widetilde{K}_0 c_Q$ gives that for any $Q \in \mathcal{N}$ we have

$\mathcal{M}, P \models \widetilde{K}_0 c_Q$, i.e. there exists a process $R \in \mathcal{M}$ such that $\mathcal{M}, R \models c_Q$, or equivalently, $R \equiv Q$.
Hence, we proved that for any $R \in \mathcal{M}$ there exists $Q \in \mathcal{N}$ such that $R \equiv Q$, and for any $Q \in \mathcal{N}$ there exists $R \in \mathcal{M}$ such that $R \equiv Q$. Because we identify processes up to structural congruence, we decide that $M = N$. □

## 5.5 Finite model property and decidability

Now we can prove the finite model property. Recall that this means to prove that we can construct, starting from a formula $\phi$, a finite class $C_\phi$ of couples $(\mathcal{M}, P)$ with $\mathcal{M}$ context and $P \in \mathcal{M}$ such that if $\phi$ is satisfiable then, necessarily, an element $(\mathcal{M}, P) \in C_\phi$ exists such that $\mathcal{M}, P \models \phi$. Anticipating, we will prove that the wanted set $C_\phi$ is

$$C_\phi = \{(\mathcal{M}, P) \mid \mathcal{M} \in \mathfrak{M}_{(\!|\phi|\!)},\ P \in \mathcal{M}\}$$

We have already proved, in theorem 3.7.2 that $\mathfrak{M}_{(\!|\phi|\!)}$ is finite and each context $\mathcal{M} \in \mathfrak{M}_{(\!|\phi|\!)}$ contains a finite number of processes. Thus $C_\phi$ will be finite.

We begin by proving that the relation $\mathcal{M}, P \models \phi$ is conserved by substituting the couple $(M, P)$ with any other couple structurally bisimilar to it at the size $(\!|\phi|\!)$. In other words $\phi$ is *"sensitive"* only up to $(\!|\phi|\!)$.

**Definition 5.5.1** (Extending the structural bisimulation)**.** We write $(\mathcal{M}, P) \approx_h^w (\mathcal{N}, Q)$ for the case when $P \in \mathcal{M}$, $Q \in \mathcal{N}$, $P \approx_h^w Q$ and $\mathcal{M} \approx_h^w \mathcal{N}$.

**Lemma 5.5.1.** *If* $(\!|\phi|\!) = (h, w)$, $\mathcal{M}, P \models \phi$ *and* $(\mathcal{M}, P) \approx_h^w (\mathcal{N}, Q)$ *then* $\mathcal{N}, Q \models \phi$.

*Proof.* We prove it by induction on the syntactical structure of $\phi$.

- **The case** $\phi = 0$**:** $(\!|\phi|\!) = (1, 1)$.
  $\mathcal{M}, P \models 0$ implies $P \equiv 0$.
  As $P \approx_1^1 Q$ we should have $Q \equiv 0$ as well, because else $Q \equiv \alpha.Q'|Q''$ asks for $P \equiv \alpha.P'|P''$ for some $P', P''$, but this is impossible because $P \equiv 0$.
  So $Q \equiv 0 \in \mathcal{N}$ and we have $\mathcal{N}, Q \models 0$, q.e.d.

- **The case** $\phi = \top$**:** is a trivial case as $\mathcal{N}, Q \models \top$ always.

- **The case** $\phi = \phi_1 \wedge \phi_2$**:** denote by $(h_i, w_i) = (\!|\phi_i|\!)$ for $i = 1, 2$. Then we have $(\!|\phi|\!) = (max(h_1, h_2), max(w_1, w_2))$.

  $\mathcal{M}, P \models \phi$ is equivalent with $\mathcal{M}, P \models \phi_1$ and $\mathcal{M}, P \models \phi_2$.

  Because $(\mathcal{M}, P) \approx_{max(h_1,h_2)}^{max(w_1,w_2)} (\mathcal{N}, Q)$ we obtain, by using theorem 3.6.1, that $(\mathcal{M}, P) \approx_{h_1}^{w_1} (\mathcal{N}, Q)$ and $(\mathcal{M}, P) \approx_{h_2}^{w_2} (\mathcal{N}, Q)$.

  Now $(\mathcal{M}, P) \approx_{h_1}^{w_1} (\mathcal{N}, Q)$ and $\mathcal{M}, P \models \phi_1$ give, by inductive hypothesis, $\mathcal{N}, Q \models \phi_1$, while $(\mathcal{M}, P) \approx_{h_2}^{w_2} (\mathcal{N}, Q)$ and $\mathcal{M}, P \models \phi_2$ give, by inductive hypothesis $\mathcal{N}, Q \models \phi_2$.
  Hence $\mathcal{N}, Q \models \phi_1 \wedge \phi_2$, q.e.d.

- **The case** $\phi = \neg\phi'$**:** $(\!|\phi|\!) = (\!|\phi'|\!) = (h, w)$.

  We have $\mathcal{M}, P \models \neg\phi'$ and $(\mathcal{M}, P) \approx_h^w (\mathcal{N}, Q)$.

  If $\mathcal{N}, Q \not\models \neg\phi'$, then $\mathcal{N}, Q \models \neg\neg\phi'$, i.e. $\mathcal{N}, Q \models \phi'$.
  Because $(\mathcal{M}, P) \approx_h^w (\mathcal{N}, Q)$ and $\mathcal{N}, Q \models \phi'$, the inductive hypothesis gives that $\mathcal{M}, P \models \phi'$, which combined with $\mathcal{M}, P \models \neg\phi'$ gives $\mathcal{M}, P \models \bot$ - impossible. Hence $\mathcal{N}, Q \models \neg\phi'$.

- **The case** $\phi = \phi_1|\phi_2$**:** suppose that $(\!|\phi_i|\!) = (h_i, w_i)$ for $i = 1, 2$. Then $(\!|\phi|\!) = (max(h_1, h_2), w_1 + w_2)$.

  Further, $\mathcal{M}, P \models \phi_1|\phi_2$ requires $P \equiv P_1|P_2$, with $\mathcal{M}, P_1 \models \phi_1$ and $\mathcal{M}, P_2 \models \phi_2$.

  As $(\mathcal{M}, P) \approx^{w_1+w_2}_{max(h_1,h_2)} (\mathcal{N}, Q)$ we obtain $P \approx^{w_1+w_2}_{max(h_1,h_2)} Q$. Than, from $P \equiv P_1|P_2$, using theorem 3.3.5, we obtain $Q \equiv Q_1|Q_2$ and $P_i \approx^{w_i}_{max(h_1,h_2)} Q_i$ for $i = 1, 2$. Hence, using theorem 3.6.1, $(\mathcal{M}, P_i) \approx^{w_i}_{max(h_1,h_2)} (\mathcal{N}, Q_i)$. Further, using again theorem 3.6.1, we obtain $(\mathcal{M}, P_i) \approx^{w_i}_{h_i} (\mathcal{N}, Q_i)$, and using the inductive hypothesis, $\mathcal{N}, Q_1 \models \phi_1$ and $\mathcal{N}, Q_2 \models \phi_2$. Hence $\mathcal{N}, Q \models \phi$.

- **The case** $\phi = \langle\alpha\rangle\phi'$**:** suppose that $(\!|\phi'|\!) = (h', w')$. We have $(\!|\langle\alpha\rangle\phi'|\!) = (1 + h', 1 + w')$.

  $\mathcal{M}, P \models \langle\alpha\rangle\phi'$ means that $P \xrightarrow{\alpha} P'$ and $\mathcal{M}, P' \models \phi'$.

  Now $(\mathcal{M}, P) \approx^{1+w'}_{1+h'} (\mathcal{N}, Q)$ gives $P \approx^{1+w'}_{1+h'} Q$, and using theorem 3.3.10, we obtain that $Q \xrightarrow{\alpha} Q'$ and $P' \approx^{w'}_{h'} Q'$.

  But $(\mathcal{M}, P) \approx^{1+w'}_{1+h'} (\mathcal{N}, Q)$ gives also $\mathcal{M} \approx^{w'+1}_{h'+1} \mathcal{N}$, so using theorem 3.6.1, $\mathcal{M} \approx^{w'}_{h'} \mathcal{N}$. Hence $(\mathcal{M}, P') \approx^{w'}_{h'} (\mathcal{N}, Q')$.

  Now from $\mathcal{M}, P' \models \phi'$ and $(\mathcal{M}, P') \approx^{w'}_{h'} (\mathcal{N}, Q')$, we obtain, by using the inductive hypothesis, that $\mathcal{N}, Q' \models \phi'$, and as $Q \xrightarrow{\alpha} Q'$, we obtain further that $\mathcal{N}, Q \models \phi$.

- **The case** $\phi = K_R\phi'$ **with** $R \in \mathfrak{S}$**:** suppose that $(\!|\phi'|\!) = (h', w')$ and $[\![R]\!] = (h_R, w_R)$.

  Then $(\!|K_R\phi'|\!) = (1 + max(h', h_R), 1 + max(w', w_R))$.

  Now $\mathcal{M}, P \models K_R\phi'$ gives $P \equiv R|P'$ and for any $R|S \in \mathcal{M}$ we have $\mathcal{M}, R|S \models \phi'$.

  As $(\mathcal{M}, P) \approx^{1+max(w',w_R)}_{1+max(h',h_R)} (\mathcal{N}, Q)$ then $P \approx^{1+max(w',w_R)}_{1+max(h',h_R)} Q$ and because

$P \equiv R|P'$ and $[\![R]\!] = (h_R, w_R) < (1 + max(h', h_R), 1 + max(w', w_R))$, we obtain, using theorem 3.3.7, that $Q \equiv R|Q'$.

Let $R|S' \in \mathcal{N}$ be an arbitrary process. Because $\mathcal{M} \approx^{1+max(w',w_R)}_{1+max(h',h_R)} \mathcal{N}$ we obtain that exists a process $P'' \in \mathcal{M}$ such that $P'' \approx^{1+max(w',w_R)}_{1+max(h',h_R)} R|S'$. But $[\![R]\!] < (1+max(h', h_R), 1+max(w', w_R))$, so, using theorem 3.3.7, $P'' \equiv R|S''$.

Then $\mathcal{M}, R|S'' \models \phi'$, as $\mathcal{M}, R|S \models \phi'$ for any $R|S \in \mathcal{M}$.

From the other side, $(\mathcal{M}, P) \approx^{1+max(w',w_R)}_{1+max(h',h_R)} (\mathcal{N}, Q)$ gives, using theorem 3.6.1, $(\mathcal{M}, P) \approx^{w'}_{h'} (\mathcal{N}, Q)$ where from we obtain $\mathcal{M} \approx^{w'}_{h'} \mathcal{N}$.

Also $R|S'' \approx^{1+max(w',w_R)}_{1+max(h',h_R)} R|S'$ gives $R|S'' \approx^{w'}_{h'} R|S'$, i.e. $(\mathcal{M}, R|S'') \approx^{w'}_{h'} (\mathcal{N}, R|S')$.

Now $\mathcal{M}, R|S'' \models \phi'$ and $(\mathcal{M}, R|S'') \approx^{w'}_{h'} (\mathcal{N}, R|S')$ give, using the inductive hypothesis, that $\mathcal{N}, R|S' \models \phi'$.

Concluding, we obtained that $Q \equiv R|Q'$ and for any $R|S' \in \mathcal{N}$ we have $\mathcal{N}, R|S' \models \phi'$. These two give $\mathcal{N}, Q \models K_R\phi'$ q.e.d.

□

Now, using this lemma, we conclude that if a process, in a context, satisfies $\phi$ then by pruning the process and the context on the size $(\!|\phi|\!)$, we still have satisfiability for $\phi$.

**Theorem 5.5.2.** *If $\mathcal{M}, P \models \phi$ then $\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \phi$.*

*Proof.* Let $(\!|\phi|\!) = (h, w)$. By contexts pruning theorem 3.7.3, we have $\mathcal{M} \approx^h_w \mathcal{M}_{(h,w)}$. By process pruning theorem 3.4.1, we have $P \approx^h_w P_{(h,w)}$ and $P_{(h,w)} \in \mathcal{M}_{(h,w)}$. Hence $(\mathcal{M}, P) \approx^h_w (\mathcal{M}_{(h,w)}, P_{(h,w)})$. Further lemma 5.5.1 establishes $\mathcal{M}_{(h,w)}, P_{(h,w)} \models \phi$ q.e.d. □

With all these results proved we can proceed with the finite model property. Indeed, because from $\mathcal{M}, P \models \phi$ we can derive $\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \phi$, and because $\mathcal{M}_{(\!|\phi|\!)} \in \mathfrak{M}_{(\!|\phi|\!)}$ we can decide on the satisfiability of $\phi$ just by browsing the couples $(\mathcal{M}, P)$ with $P \in \mathcal{M} \in \mathfrak{M}_{(\!|\phi|\!)}$. But this means searching in a finite set. Indeed in theorem 3.7.2 we proved that $\mathfrak{M}_{(\!|\phi|\!)}$ is finite and each context $\mathcal{M} \in \mathfrak{M}_{(\!|\phi|\!)}$ contains a finite number of processes.

**Theorem 5.5.3** (Finite model property)**.**

$$\textit{If } \mathcal{M}, P \models \phi \textit{ then } \exists \mathcal{N} \in \mathfrak{M}_{(\!|\phi|\!)} \textit{ and } Q \in \mathcal{N} \textit{ such that } \mathcal{N}, Q \models \phi$$

*Proof.* We can take directly $\mathcal{N} = \mathcal{M}_{(h,w)} \in \mathfrak{M}_{(h,w)}$ and $Q = P_{(h,w)} \in \mathcal{M}_{(h,w)}$, and theorem 5.5.2 proves the finite model property. □

The finite model property provides a practical finite procedure for deciding on satisfiability, validity and model-checking problems.

Suppose that we have a formula $\phi$ and we want to know if there exists a process $P$, in a context $\mathcal{M}$, which satisfies $\phi$. Using the finite model property, if such a situation exists, then we should find, in a context $\mathcal{M} \in \mathfrak{M}_{(\!|\phi|\!)}$, a process that satisfies $\phi$. If, after checking the satisfiability of $\phi$ for all such cases (which are finitely many), we do not find any process in a context that satisfies $\phi$, we can decide that $\phi$ is not satisfiable. Otherwise $\phi$ is satisfiable and we also find a model of it.

Suppose that we want to check the validity of the formula $\phi$. Then it is sufficient to check if $\neg\phi$ is satisfiable (we saw how to do this by a finite procedure).

Going further, suppose that we have a context $\mathcal{M}$, a process $P \in \mathcal{M}$ and a formula $\phi$ and we want to decide, in a finite manner, if it is the case that $\mathcal{M}, P \models \phi$. The only nontrivial case is when $\mathcal{M}$ is infinite, because the epistemic operators, if involved in $\phi$, act as universal quantifiers on the level of $\mathcal{M}$. In such a case we can apply the finite model property and reduce the problem to the case $\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \phi$, which, as $\mathcal{M}_{(\!|\phi|\!)}$ is finite, goes to a finite problem.

**Theorem 5.5.4** (Decidability)**.** *For $\mathcal{L}^{\mathfrak{S}}_{DES}$ validity, satisfiability and model checking are decidable against the process semantics.*

## 5.6 Axioms of $\mathcal{L}^{\mathfrak{S}}_{DES}$

Starting with this section we will present the Hilbert-style axiomatic system for $\mathcal{L}^{\mathfrak{S}}_{DES}$. It will be just an extension of the system proposed for $\mathcal{L}_{DS}$ with a set of epistemic axioms and rules. Hence, we assume the axioms and the rules of propositional logic, the spatial axioms and rules of $\mathcal{L}_{DS}$ (although the same, for this new system we will use the label $E$ for axioms and $E_R$ for rules) and the dynamic axioms and rules of $\mathcal{L}_{DS}$, with the only exception of rule $\mathrm{D}_R4$ which will be slightly modified. In addition, our system will contain a specific class of epistemic axioms and rules.

### Spatial axioms

**Axiom E 1.** $\vdash \top|\bot \rightarrow \bot$

**Axiom E 2.** $\vdash \phi|0 \leftrightarrow \phi$

**Axiom E 3.** $\vdash \phi|\psi \to \psi|\phi$

**Axiom E 4.** $\vdash (\phi|\psi)|\rho \to \phi|(\psi|\rho)$

**Axiom E 5.** $\vdash \phi|(\psi \vee \rho) \to (\phi|\psi) \vee (\phi|\rho)$

**Axiom E 6.** $\vdash (c_P \wedge \phi|\psi) \to \bigvee_{P \equiv Q|R}(c_Q \wedge \phi)|(c_R \wedge \psi)$

**Spatial rules**

**Rule $\mathbf{E}_R$ 1.** *If* $\vdash \phi \to \psi$ *then* $\vdash \phi|\rho \to \psi|\rho$

**Dynamic axioms**

**Axiom E 7.** $\vdash \langle\alpha\rangle\phi|\psi \to \langle\alpha\rangle(\phi|\psi)$

**Axiom E 8.** $\vdash [\alpha](\phi \to \psi) \to ([\alpha]\phi \to [a]\psi)$

**Axiom E 9.** $\vdash 0 \to [\alpha]\bot$

**Axiom E 10.** *If* $\beta \neq \alpha_i$ *for* $i = 1..n$ *then* $\vdash \langle!\alpha_1\rangle\top|...|\langle!\alpha_n\rangle\top \to [\beta]\bot$

**Axiom E 11.** $\vdash \langle!\alpha\rangle\phi \to [\alpha]\phi$

**Dynamic rules**

**Rule $\mathbf{E}_R$ 2.** *If* $\vdash \phi$ *then* $\vdash [\alpha]\phi$

**Rule $\mathbf{E}_R$ 3.** *If* $\vdash \phi \to [\alpha]\phi'$ *and* $\vdash \psi \to [\alpha]\psi'$ *then* $\vdash \phi|\psi \to [\alpha](\phi'|\psi \vee \phi|\psi')$.

**Rule $\mathbf{E}_R$ 4.** *If* $\vdash \bigvee_{[\![M]\!] \leq (\!|\phi|\!)} c_{\overline{M}} \to \phi$ *then* $\vdash \phi$.

We will not comment again on these axioms and rules, as the remarks for case of $\mathcal{L}_{DS}$ remain available.

The only observation we make concerns rule $\mathrm{E}_R4$, which has been modified with respect to the dynamic system. Because the dynamic epistemic semantics are sensitive to contexts, a fact revealed also by the finite model property, we had to replace the hypothesis of $\mathrm{D}_R4$ with a stronger one. The condition $[\![M]\!] \leq (\!|\phi|\!)$ is equivalent to $\overline{M} \in \mathfrak{M}_{(\!|\phi|\!)}$, so the disjunction involved in the axiom contains a finite number of disjuncts, as $\mathfrak{M}_{(\!|\phi|\!)}$ is finite.

## Epistemic axioms

**Axiom E 12.** *If* $P \in \mathfrak{S}$ *then* $\vdash K_P\top \leftrightarrow c_P|\top$

**Axiom E 13.** $\vdash K_Q\phi \wedge K_Q(\phi \to \psi) \to K_Q\psi$

**Axiom E 14.** $\vdash K_Q\phi \to \phi$

**Axiom E 15.** $\vdash K_Q\phi \to K_QK_Q\phi$.

**Axiom E 16.** $\vdash K_Q\top \to (\neg K_Q\phi \to K_Q\neg K_Q\phi)$

**Axiom E 17.** $\vdash K_Q\phi \leftrightarrow (K_Q\top \wedge K_0(K_Q\top \to \phi))$

**Axiom E 18.** $\vdash K_0\phi \wedge \psi|\rho \to (K_0\phi \wedge \psi)|(K_0\phi \wedge \rho)$

**Axiom E 19.** $\vdash K_0\phi \to [\alpha]K_0\phi$

**Axiom E 20.** $\vdash K_0\phi \to (K_Q\top \to K_QK_0\phi)$

## Epistemic rules

**Rule $\mathbf{E}_R$ 5.** *If* $\vdash \phi$ *then* $\vdash K_Q\top \to K_Q\phi$.

**Rule $\mathbf{E}_R$ 6.** *If* $\mathcal{M} \ni P$ *is a finite context and* $\vdash c_{\mathcal{M}} \wedge c_P \to K_0\phi$ *then* $\vdash c_{\mathcal{M}} \to \phi$.

Axiom E12 states the equivalence between *to be active* and *to know* for the epistemic agents. Indeed $\mathcal{M}, Q \models K_P\top$ means exactly $P$ is an active subsystem of $Q$ and nothing more. The same can be expressed by $\mathcal{M}, Q \models c_P|\top$.

Axiom E13 is the classical (K)-axiom stating that our epistemic operator is a normal one. This is an expected axiom as all the epistemic logics have it.

The same remark on axiom E14 which is just the axiom (T) - necessity axiom, for the epistemic operator.

Also axiom E15 is well known in epistemic logics. It states that our epistemic agents satisfy *the positive introspection property*, i.e. if $P$ knows something then it knows that it knows that thing.

Axiom E16 states a variant of the *negative introspection*, saying that if an agent $P$ is active and if it doesn't know $\phi$, then it knows that it doesn't know $\phi$. The novelty in our axiom is the precondition $K_P\top$ of the negative introspection. This precondition guarantees that the agent really exists, i.e. it is active. Such a precondition does not appear in the other epistemic logics for the reason that, in those cases, the agents exists always and they knows, always, at least the tautologies.

Axiom E17 provides a full description of the $K_Q$ operator by means of $K_0$ and $K_Q\top$. As, by axiom E12, $K_Q\top$ can be expressed by the epistemic operators, our system might be reduced to one epistemic operator only, $K_0$. We leave for future work the analysis of minimality for our axiomatic system. For the moment we consider it interesting to have all these epistemic operators that provide links with the rest of epistemic logics.

Axioms E18, E19 and E20 present $K_0\phi$ as a syntactic encryption of validity, stating that once $K_0\phi$ can be stated for a real system, it will be propagated to all the levels of it.

Rule $\mathrm{E}_R5$ states that any active agent knows all the tautologies. As

in the case of the negative introspection, we deal with a well known epistemic rule, widely spread in epistemic logics, but our rules work under the assumption that the agent is active.

Also rule $\mathrm{E}_R6$ depicts the fact that $K_0\phi$ is an encoding of the validity in a given context.

## 5.7 The soundness of $\mathcal{L}^{\mathfrak{S}}_{DES}$ against the process semantics

In this section we will motivate the choice of the axioms by proving the soundness of our system with respect to process semantics. In this way we will prove that everything expressed by our axioms and rules about the process semantics is correct and, in conclusion, using our system, we can derive only theorems that can be meaningfully interpreted.

**Theorem 5.7.1** (Process-Soundness)**.** *The system $\mathcal{L}^{\mathfrak{S}}_{DES}$ is sound against the process semantics.*

*Proof.* The soundness of $\mathcal{L}^{\mathfrak{S}}_{DES}$ will be sustained by the soundness of all spatial, dynamic and epistemic axioms and rules.

Referring to section 4.5, we remind the reader that the proofs of soundness for the spatial and dynamic parts of the system have already been showed (as these axioms and rules are identical in $\mathcal{L}^{\mathfrak{S}}_{DES}$ and $\mathcal{L}_{DS}$). The only exception is rule $\mathrm{E}_R4$, for which we will explicitly prove the soundness further.

We will also prove the soundness for the epistemic axioms and rules. □

*Remark* 5.7.1. Observe that the theorems proved in section 4.7 concerning the properties of the characteristic formulas for processes are also available for the system $\mathcal{L}^{\mathfrak{S}}_{DES}$, excepting the syntactic characterization of satisfiability, lemma 4.7.5. Indeed our system has the same semantics with respect to spatial and dynamic operators and the same axioms on these levels (excepting rule $\mathrm{D}_R4$, which is not used in the proofs of these theorems). Thus, we will reuse, if the case, these results.

**Lemma 5.7.2** (Soundness of rule $\mathrm{E}_R4$)**.** *If* $\models \bigvee_{[\![M]\!]\leq(\!|\phi|\!)} c_{\overline{M}} \rightarrow \phi$ *then* $\models \phi$.

*Proof.* Suppose that $\models \bigvee_{[\![M]\!]\leq(\!|\phi|\!)} c_{\overline{M}} \rightarrow \phi$ but exists a model $\mathcal{N}$ and a process $Q \in \mathcal{N}$ with $\mathcal{N}, Q \not\models \phi$. Then $\mathcal{N}, Q \models \neg\phi$.
Further, using the finite model property, theorem 5.5.3, we obtain that exists a context $\mathcal{N}' \in \mathfrak{M}_{(\!|\neg\phi|\!)}$ and a process $R \in \mathcal{N}'$ with $\mathcal{N}', R \models \neg\phi$.
But $(\!|\phi|\!) = (\!|\neg\phi|\!)$, so exists the context $\mathcal{N}' \in \mathfrak{M}_{(\!|\phi|\!)}$ and a process $R \in \mathcal{N}'$ with $\mathcal{N}', R \models \neg\phi$. Moreover, $\mathcal{N}' \in \mathfrak{M}_{(\!|\phi|\!)}$ gives $\mathcal{N}' = \overline{N}$ with $N \subset \mathfrak{P}$ and $[\![N]\!] \leq (\!|\phi|\!)$.
Thus, because $\mathcal{N}', R \models c_{\mathcal{N}'}$, we derive $\mathcal{N}', R \models \bigvee_{[\![M]\!]\leq(\!|\phi|\!)} c_{\overline{M}}$.
But $\models \bigvee_{[\![M]\!]\leq(\!|\phi|\!)} c_{\overline{M}} \rightarrow \phi$ implies $\mathcal{N}', R \models \bigvee_{[\![M]\!]\leq(\!|\phi|\!)} c_{\overline{M}} \rightarrow \phi$ which combined with $\mathcal{N}', R \models \bigvee_{[\![M]\!]\leq(\!|\phi|\!)} c_{\overline{M}}$ entails $\mathcal{N}', R \models \phi$.
As we also have $\mathcal{N}', R \models \neg\phi$, we obtain $\mathcal{N}', R \models \bot$ - impossible!
Then, for any model $\mathcal{N}$ and any process $P \in \mathcal{N}$ we have $\mathcal{N}, P \models \phi$, i.e. $\models \phi$. □

Hereafter we prove the soundness for the epistemic axioms and rules.

**Lemma 5.7.3** (Soundness of axiom E12)**.**

$$\textit{If } Q \in \mathfrak{S} \textit{ then } \models c_Q|\top \leftrightarrow K_Q\top$$

*Proof.* If $\mathcal{M}, P \models c_Q|\top$ then $P \equiv R|S$, with $\mathcal{M}, S \models c_Q$. Then theorem 4.7.2 gives $S \equiv Q$, hence $P \equiv Q|R$. And because for any $Q|R' \in \mathcal{M}$ we have $\mathcal{M}, Q|R' \models \top$, we derive $\mathcal{M}, P \models K_Q\top$.
Suppose now the reverse, i.e. that $\mathcal{M}, P \models K_Q\top$. Then $P \equiv Q|R$. But $\mathcal{M}, P \models c_P$, hence $\mathcal{M}, P \models c_Q|c_R$.
Because $\models c_Q \to \top$, using the soundness of rule $\mathrm{E}_R1$, we derive $\models c_Q|c_R \to c_Q|\top$ from where we conclude that $\mathcal{M}, P \models c_Q|\top$. □

**Lemma 5.7.4** (Soundness of axiom E13)**.**

$$\models K_Q\phi \wedge K_Q(\phi \to \psi) \to K_Q\psi$$

*Proof.* Suppose that $\mathcal{M}, P \models K_Q\phi$ and that $\mathcal{M}, P \models K_Q(\phi \to \psi)$. Then $P \equiv Q|R$ and for any $S$ such that $S|Q \in \mathcal{M}$ we have $\mathcal{M}, S|Q \models \phi$ and $\mathcal{M}, Q|S \models \phi \to \psi$. Hence for any such $Q|S$ we have $\mathcal{M}, Q|S \models \psi$ and because $P \equiv Q|R$ we obtain that $\mathcal{M}, P \models K_Q\psi$. □

**Lemma 5.7.5** (Soundness of axiom E14)**.** $\models K_Q\phi \to \phi$.

*Proof.* If $\mathcal{M}, P \models K_Q\phi$ then $P \equiv Q|R$ and for any $Q|S \in \mathcal{M}$ we have $\mathcal{M}, Q|S \models \phi$, i.e. $\mathcal{M}, Q|R \models \phi$, so $\mathcal{M}, P \models \phi$. □

**Lemma 5.7.6** (Soundness of axiom E15)**.** $\models K_Q\phi \to K_QK_Q\phi$.

*Proof.* Suppose that $\mathcal{M}, P \models K_Q\phi$, then $P \equiv Q|R$ and for any $Q|S \in \mathcal{M}$ we have $\mathcal{M}, Q|S \models \phi$. Let $Q|S' \in \mathcal{M}$ be arbitrarily chosen. As for any $Q|S \in \mathcal{M}$ we have $\mathcal{M}, Q|S \models \phi$, we derive that $\mathcal{M}, Q|S' \models K_Q\phi$. But $Q|S'$ has been arbitrarily chosen, so for any $Q|S \in \mathcal{M}$ we have $\mathcal{M}, Q|S \models K_Q\phi$, and because $P \equiv Q|R$ we obtain $\mathcal{M}, P \models K_QK_Q\phi$. □

**Lemma 5.7.7** (Soundness of axiom E16)**.**

$$\models K_Q\top \rightarrow (\neg K_Q\phi \rightarrow K_Q\neg K_Q\phi)$$

*Proof.* Suppose that $\mathcal{M}, P \models K_Q\top$ and $\mathcal{M}, P \models \neg K_Q\phi$. Then $P \equiv Q|R$ and $\exists S$ such that $\mathcal{M}, S|Q \models \neg\phi$. But then for any $U$ such that $U|Q \in \mathcal{M}$ we have $\mathcal{M}, U|Q \models \neg K_Q\phi$. Hence $\mathcal{M}, P \models K_Q\neg K_Q\phi$. □

**Lemma 5.7.8** (Soundness of axiom E17)**.**

$$\models K_Q\phi \leftrightarrow (K_Q\top \wedge K_0(K_Q\top \rightarrow \phi))$$

*Proof.* Suppose that $\mathcal{M}, P \models K_Q\phi$. Then $P \equiv Q|R$ and for any $Q|S \in \mathcal{M}$ we have $\mathcal{M}, Q|S \models \phi$. From $P \equiv Q|R$, because for any $Q|S \in \mathcal{M}$ we have $\mathcal{M}, Q|S \models \top$, we derive $\mathcal{M}, P \models K_Q\top$. Consider now an arbitrary process $S \in \mathcal{M}$. If $\mathcal{M}, S \not\models K_Q\top$, then $\mathcal{M}, S \models K_Q\top \rightarrow \phi$.
If $\mathcal{M}, S \models K_Q\top$ we derive that $S \equiv Q|S'$, hence $\mathcal{M}, S \models \phi$.
So, for an arbitrarily chosen $S \in \mathcal{M}$ we have $\mathcal{M}, S \models K_Q\top \rightarrow \phi$.
Because $P \equiv P|0$ and for any process $S \equiv S|0 \in \mathcal{M}$ we have
$\mathcal{M}, S \models K_Q\top \rightarrow \phi$, we derive that $\mathcal{M}, P \models K_0(K_Q\top \rightarrow \phi)$. Hence $\models K_Q\phi \rightarrow (K_Q\top \wedge K_0(K_Q\top \rightarrow \phi))$.

Suppose now that $\mathcal{M}, P \models K_Q\top \wedge K_0(K_Q\top \rightarrow \phi)$. From $\mathcal{M}, P \models K_Q\top$ we derive $P \equiv Q|R$.
Because $\mathcal{M}, P \models K_0(K_Q\top \rightarrow \phi)$, we obtain that for any process $S \in \mathcal{M}$ we have $\mathcal{M}, S \models K_Q\top \rightarrow \phi$. Hence, for any process $S|Q \in \mathcal{M}$ we have $\mathcal{M}, S|Q \models \phi$ (because $\mathcal{M}, S|Q \models K_Q\top$). And because $P \equiv Q|R$, we derive $\mathcal{M}, P \models K_Q\phi$. □

**Lemma 5.7.9** (Soundness of axiom E18)**.**

$$\models K_0\phi \wedge \psi|\rho \rightarrow (K_0\phi \wedge \psi)|(K_0\phi \wedge \rho)$$

*Proof.* Suppose that $\mathcal{M}, P \models K_0\phi \wedge \psi|\rho$ then $\mathcal{M}, P \models K_0\phi$ and $\mathcal{M}, P \models \psi|\rho$.
$\mathcal{M}, P \models K_0\phi$ gives that for any $R \in \mathcal{M}$ we have $\mathcal{M}, R \models \phi$.
$\mathcal{M}, P \models \psi|\rho$ gives that $P \equiv P'|P''$ and $\mathcal{M}, P' \models \psi$, $\mathcal{M}, P'' \models \rho$. Because $P', P'' \in \mathcal{M}$ and because for any $R \in \mathcal{M}$, $\mathcal{M}, R \models \phi$ we derive that $\mathcal{M}, P' \models K_0\phi$ and $\mathcal{M}, P'' \models K_0\phi$.
Hence $\mathcal{M}, P' \models \psi \wedge K_0\phi$ and $\mathcal{M}, P'' \models \rho \wedge K_0\phi$. As $P \equiv P'|P''$, we obtain further $\mathcal{M}, P \models (K_0\phi \wedge \psi)|(K_0\phi \wedge \rho)$. □

**Lemma 5.7.10** (Soundness of axiom E19)**.** $\models K_0\phi \rightarrow [\alpha]K_0\phi$

*Proof.* Suppose that $\mathcal{M}, P \models K_0\phi$, then for any $R \in \mathcal{M}$ we have $\mathcal{M}, R \models \phi$.
If $P$ cannot perform a transition by $\alpha$, we have $\mathcal{M}, P \models [\alpha]K_0\phi$.
If $P$ can perform such transitions, then for any $P \xrightarrow{\alpha} P'$ we have $\mathcal{M}, P' \models K_0\phi$ (as for any $R \in \mathcal{M}$ we have $\mathcal{M}, R \models \phi$). This means $\mathcal{M}, P \models [\alpha]K_0\phi$. □

**Lemma 5.7.11** (Soundness of axiom E20)**.**

$$\models K_0\phi \rightarrow (K_Q\top \rightarrow K_QK_0\phi)$$

*Proof.* Suppose that $\mathcal{M}, P \models K_0\phi$ and $\mathcal{M}, P \models K_Q\top$.
$\mathcal{M}, P \models K_0\phi$ gives that for any $R \in \mathcal{M}$ we have $\mathcal{M}, R \models \phi$.
$\mathcal{M}, P \models K_Q\top$ means that $P \equiv Q|S$. Because for any $R \in \mathcal{M}$ we have $\mathcal{M}, R \models \phi$, we obtain that for any $Q|S' \in \mathcal{M}$ we have $\mathcal{M}, Q|S' \models K_0\phi$, and because $P \equiv Q|S$ we obtain $\mathcal{M}, P \models K_QK_0\phi$. □

**Lemma 5.7.12** (Soundness of rule $\mathrm{E}_R5$)**.**

$$\textit{If } \models \phi \textit{ then } \models K_Q\top \to K_Q\phi$$

*Proof.* If $\models \phi$ then for any context $\mathcal{M}$ and any process $P \in \mathcal{M}$ we have $\mathcal{M}, P \models \phi$. Suppose now that $\mathcal{M}, P \models K_Q\top$. Then $P \equiv Q|R$. Because $\mathcal{M}, S \models \phi$ for each $S \in \mathcal{M}$, we derive that for any $S|Q \in \mathcal{M}$ we have $\mathcal{M}, S|Q \models \phi$. Hence $\mathcal{M}, P \models K_Q\phi$. □

**Lemma 5.7.13** (Soundness of rule $\mathrm{E}_R6$)**.**

$$\textit{If } \mathcal{M} \ni P \textit{ is a finite context and } \models c_{\mathcal{M}} \wedge c_P \to K_0\phi \textit{ then } \models c_M \to \phi$$

*Proof.* Suppose that $\models c_{\mathcal{M}} \wedge c_P \to K_0\phi$ and $\mathcal{N}$ is an arbitrary context with $Q \in \mathcal{N}$.
If $\mathcal{N}, Q \not\models c_{\mathcal{M}}$ then $\mathcal{N}, Q \models c_{\mathcal{M}} \to \phi$.
If $\mathcal{N}, Q \models c_{\mathcal{M}}$, then $\mathcal{N} = \mathcal{M}$. Further $\mathcal{M}, P \models c_P \wedge c_{\mathcal{M}}$ gives $\mathcal{M}, P \models K_0\phi$, i.e. for each $S|0 \equiv S \in \mathcal{M}$ we have $\mathcal{M}, S \models \phi$. Now, because $\mathcal{N} = \mathcal{M}$ and $Q \in \mathcal{M}$ we obtain $\mathcal{N}, Q \models \phi$. Hence, also in this case $\mathcal{N}, Q \models c_M \to \phi$. Thus $\models c_{\mathcal{M}} \to \phi$. □

Hence we have a sound system and all the theorems that can be proved with it are sound results with respect to process semantics.

## 5.8 Theorems of $\mathcal{L}^{\mathfrak{S}}_{DES}$

In this section we will derive some theorems for $\mathcal{L}^{\mathfrak{S}}_{DES}$. As, by soundness, the theorems specify *"facts"* about processes, we will try to interpret the nontrivial ones.

Observe that, as no proof of a theorem in section 4.6 uses rule $\mathrm{D}_R4$ and all the other spatial and dynamic axioms and rules are identical in $\mathcal{L}_{DS}$ and $\mathcal{L}_{DES}^{\mathfrak{S}}$, the theorems in section 4.6 hold for $\mathcal{L}_{DES}^{\mathfrak{S}}$ as well.

### 5.8.1 Epistemic results

We begin by stating that 0 is always an active agent: it always performs its *"inactivity"* expressed by 0.

**Theorem 5.8.1.** $\vdash K_0\top$.

*Proof.* Trivial consequence of axiom E12 and axiom E2. □

The next result states that an agent knows something only if it is active. Hence *to know* implies *to be*.

**Theorem 5.8.2.** $\vdash K_P\phi \to K_P\top$.

*Proof.* Trivial consequence of axiom E17. □

Further we prove another obvious property of knowledge: if $Q$ knows $\phi$ and $Q$ knows $\psi$, this is equivalent with $Q$ knows $\phi \wedge \psi$.

**Theorem 5.8.3.** $\vdash K_Q\phi \wedge K_Q\psi \leftrightarrow K_Q(\phi \wedge \psi)$

*Proof.* $\vdash \phi \to (\psi \to (\phi \wedge \psi))$. Using rule $\mathrm{E}_R5$, we obtain

$$\vdash K_Q\top \to K_Q[\phi \to (\psi \to (\phi \wedge \psi))]$$

We apply axiom E13 twice, and obtain

$$\vdash K_Q\top \to [K_Q\phi \to (K_Q\psi \to K_Q(\phi \wedge \psi))]$$

i.e.

$$\vdash K_Q\top \wedge K_Q\phi \to [K_Q\psi \to K_Q(\phi \wedge \psi)]$$

But $\vdash K_Q\phi \to K_Q\top$, hence $\vdash K_Q\phi \to [K_Q\psi \to K_Q(\phi \wedge \psi)]$, i.e.

$$\vdash K_Q\phi \wedge K_Q\psi \to K_Q(\phi \wedge \psi)$$

Reverse, we apply rule $\mathrm{E}_R5$ to $\vdash \phi \wedge \psi \to \psi$ and then axiom E13, and obtain $\vdash K_Q\top \to (K_Q(\phi \wedge \psi) \to K_Q\phi)$. But $\vdash K_Q(\phi \wedge \psi) \to K_Q\top$, hence $\vdash K_Q(\phi \wedge \psi) \to K_Q\phi$.
Similarly $\vdash K_Q(\phi \wedge \psi) \to K_Q\psi$. □

The knowledge is redundant and introspective: if $Q$ knows $\phi$ this is equivalent with the fact that $Q$ knows that $Q$ knows $\phi$.

**Theorem 5.8.4.** $\vdash K_QK_Q\phi \leftrightarrow K_Q\phi$.

*Proof.* Axiom E15 gives $\vdash K_Q\phi \to K_QK_Q\phi$, and axiom E14 gives $\vdash K_QK_Q\phi \to K_Q\phi$. □

**Theorem 5.8.5** (Monotonicity of knowledge)**.**

$$\textit{If } \vdash \phi \to \psi \textit{ then } \vdash K_P\phi \to K_P\psi$$

*Proof.* Because $\vdash \phi \to \psi$, we can use rule $\mathrm{E}_R5$ and obtain
$\vdash K_P\top \to K_P(\phi \to \psi)$. But theorem 5.8.2 gives $\vdash K_P\phi \to K_P\top$, hence $\vdash K_P\phi \to K_P(\phi \to \psi)$ where from we derive

$$\vdash K_P\phi \to (K_P\phi \wedge K_P(\phi \to \psi))$$

This entails, using axiom E13, $\vdash K_P\phi \to K_P\psi$. □

The existence of an agent entails the existence of its active sub-agents, as proved further. This is a knowledge-like description of the ontological topology of agents. It relies on *to be* is *to know*.

**Theorem 5.8.6.** $\vdash K_{P|Q}\top \rightarrow K_P\top$.

*Proof.* Axiom E12 gives $\vdash K_{P|Q}\top \leftrightarrow c_P|c_Q|\top$ and $\vdash K_P\top \leftrightarrow c_P|\top$. But $\vdash c_Q \rightarrow \top$ and applying rule $\mathrm{E}_R1$, we obtain $\vdash c_P|c_Q|\top \rightarrow c_P|\top$. Hence $\vdash K_{P|Q}\top \rightarrow K_P\top$. □

The knowledge of an agent is consistent: if it knows $\neg\phi$ (it knows that $\phi$ is false) then it cannot know $\phi$ as well. This is proved in the next two theorems.

**Theorem 5.8.7.** $\vdash K_Q\neg\phi \rightarrow \neg K_Q\phi$.

*Proof.* Axiom E14 gives $\vdash K_Q\neg\phi \rightarrow \neg\phi$ and $\vdash K_Q\phi \rightarrow \phi$. The last is equivalent with $\vdash \neg\phi \rightarrow \neg K_Q\phi$, and combined with the first entails $\vdash K_Q\neg\phi \rightarrow \neg K_Q\phi$. □

**Theorem 5.8.8** (Consistency theorem)**.** $\vdash K_Q\phi \rightarrow \neg K_Q\neg\phi$.

*Proof.* By using the negative form of theorem 5.8.7 □

In the next four theorems we will focus on the knowledge of the agent 0. It represents *"the most ignorant"* agent in $\mathcal{M}$ in the sense that if it knows something then everybody else knows it as well. This property might be

exploited in the sense that what 0 knows is a validity in $\mathcal{M}$. And the dual of knowledge operator applied to 0 gives the satisfiability in $\mathcal{M}$.

**Theorem 5.8.9.** $\vdash K_0\phi \to (K_Q\top \to K_Q\phi)$

*Proof.* Axioms E14 gives $\vdash K_0\phi \to \phi$ and applying the monotonicity of knowledge, $\vdash K_QK_0\phi \to K_Q\phi$.
Now axiom E20 provides $\vdash K_0\phi \wedge K_Q\top \to K_QK_0\phi$. Thus $\vdash K_0\phi \wedge K_Q\top \to K_Q\phi$, that is equivalent with $\vdash K_0\phi \to (K_Q\top \to K_Q\phi)$. □

**Theorem 5.8.10.** $\vdash \widetilde{K}_0\phi \leftrightarrow K_0\widetilde{K}_0\phi$

*Proof.* By definition, we have $\vdash \widetilde{K}_0\phi \leftrightarrow \neg K_0\neg\phi$, and because $\vdash K_0\top$, we derive $\vdash \widetilde{K}_0\phi \to (\neg K_0\neg\phi \wedge K_0\top)$.
But axiom E16 entails $\vdash (\neg K_0\neg\phi \wedge K_0\top) \to K_0\neg K_0\neg\phi$, i.e.

$$\vdash (\neg K_0\neg\phi \wedge K_0\top) \to K_0\widetilde{K}_0\phi$$

Hence $\vdash \widetilde{K}_0\phi \to K_0\widetilde{K}_0\phi$.
We have also $\vdash K_0\widetilde{K}_0\phi \to \widetilde{K}_0\phi$, by applying axiom E14. □

**Theorem 5.8.11.** $\vdash \widetilde{K}_0\phi \wedge \psi|\rho \to (\widetilde{K}_0\phi \wedge \psi)|(\widetilde{K}_0\phi \wedge \rho)$

*Proof.* Axiom E18 instantiated with $\phi = \widetilde{K}_0\phi$ gives

$$\vdash K_0\widetilde{K}_0\phi \wedge \psi|\rho \to (K_0\widetilde{K}_0\phi \wedge \psi)|(K_0\widetilde{K}_0\phi \wedge \rho)$$

Further, using theorem 5.8.10, we obtain the wanted result. □

**Theorem 5.8.12.** $\vdash \widetilde{K}_0\phi \rightarrow [\alpha]\widetilde{K}_0\phi$

*Proof.* Axiom E19 instantiated with $\phi = \widetilde{K}_0\phi$ gives

$$\vdash K_0\widetilde{K}_0\phi \rightarrow [\alpha]K_0\widetilde{K}_0\phi$$

Further, using theorem 5.8.10, we obtain the wanted result. □

**Theorem 5.8.13.** $\vdash \widetilde{K}_0\phi \rightarrow (K_Q\top \rightarrow K_Q\widetilde{K}_0\phi)$

*Proof.* Axiom E20 instantiated with $\phi = \widetilde{K}_0\phi$ gives

$$\vdash K_0\widetilde{K}_0\phi \rightarrow (K_Q\top \rightarrow K_QK_0\widetilde{K}_0\phi)$$

Further, using theorem 5.8.10, we obtain the wanted result. □

### 5.8.2 Theorems referring to contexts

In this section we focus on results that involve the characteristic formulas of finite contexts. We try to show, in this way, how sensitive our system is with respect to contexts. Further, these results will be used in proving the completeness for $\mathcal{L}^{\mathfrak{S}}_{DES}$.

**Theorem 5.8.14.** *If $\mathcal{M}$ is a finite context and $R \notin \mathcal{M}$ then* $\vdash c_{\mathcal{M}} \rightarrow \neg c_R$.

*Proof.* Because $c_{\mathcal{M}} = K_0(\bigvee_{P\in\mathcal{M}} c_P) \wedge (\bigwedge_{P\in\mathcal{M}} \widetilde{K}_0 c_P)$ we derive that

$$\vdash c_{\mathcal{M}} \rightarrow K_0(\bigvee_{P\in\mathcal{M}} c_P)$$

But from axiom E14 $\vdash K_0(\bigvee_{P\in\mathcal{M}} c_P) \rightarrow \bigvee_{P\in\mathcal{M}} c_P$, so $\vdash c_{\mathcal{M}} \rightarrow \bigvee_{P\in\mathcal{M}} c_P$. Further theorem 4.7.3 gives $\vdash c_P \rightarrow \neg c_R$ (as $R \notin \mathcal{M}$ and $P \in \mathcal{M}$ implies $R \not\equiv P$) which implies $\vdash \bigvee_{P\in\mathcal{M}} c_P \rightarrow \neg c_R$. But we proved that $\vdash c_{\mathcal{M}} \rightarrow \bigvee_{P\in\mathcal{M}} c_P$. Hence $\vdash c_{\mathcal{M}} \rightarrow \neg c_R$. □

**Theorem 5.8.15.** *If $\mathcal{M}$ is a finite context then*

$$\vdash (c_{\mathcal{M}} \wedge \phi|\psi) \rightarrow (c_{\mathcal{M}} \wedge \phi)|(c_{\mathcal{M}} \wedge \psi)$$

*Proof.* Observe that, by applying axiom E18, we obtain

$$\vdash (K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \wedge \phi|\psi \rightarrow (\widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \wedge (K_0\theta_1 \wedge \phi)|(K_0\theta_1 \wedge \psi) \tag{5.8.1}$$

If, further, we apply theorem 5.8.11 once, we obtain

$$\vdash (\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2) \wedge (K_0\theta_1 \wedge \phi)|(K_0\theta_1 \wedge \psi) \rightarrow \widetilde{K}_0\theta_3 \wedge (\widetilde{K}_0\theta_2 \wedge K_0\theta_1 \wedge \phi)|(\widetilde{K}_0\theta_2 \wedge K_0\theta_1 \wedge \psi)$$

Hence

$$\vdash (K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \wedge \phi|\psi \rightarrow \widetilde{K}_0\theta_3 \wedge (\widetilde{K}_0\theta_2 \wedge K_0\theta_1 \wedge \phi)|(\widetilde{K}_0\theta_2 \wedge K_0\theta_1 \wedge \psi)$$

If we apply again theorem 5.8.11 we obtain

$$\vdash \widetilde{K}_0\theta_3 \wedge (\widetilde{K}_0\theta_2 \wedge K_0\theta_1 \wedge \phi)|(\widetilde{K}_0\theta_2 \wedge K_0\theta_1 \wedge \psi) \rightarrow (\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2 \wedge K_0\theta_1 \wedge \phi)|(\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2 \wedge K_0\theta_1 \wedge \psi)$$

hence

$$\vdash (K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \wedge \phi|\psi \rightarrow (\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2 \wedge K_0\theta_1 \wedge \phi)|(\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2 \wedge K_0\theta_1 \wedge \psi)$$

Because $c_{\mathcal{M}} = K_0(\bigvee_{Q\in\mathcal{M}} c_Q) \wedge (\bigwedge_{Q\in\mathcal{M}} \widetilde{K}_0 c_Q)$, we can use the same idea, applying theorem 5.8.11 once for each process in $\mathcal{M}$ (being finite) and we obtain

$$\vdash (c_{\mathcal{M}} \wedge \phi|\psi) \rightarrow (c_{\mathcal{M}} \wedge \phi)|(c_{\mathcal{M}} \wedge \psi)$$

□

**Theorem 5.8.16.** *If $\mathcal{M}$ is a finite context then* $\vdash (c_{\mathcal{M}} \wedge \phi|\psi) \rightarrow (c_{\mathcal{M}} \wedge \phi)|\psi$

*Proof.* From the previous theorem, 5.8.15, we have

$$\vdash (c_{\mathcal{M}} \wedge \phi|\psi) \rightarrow (c_{\mathcal{M}} \wedge \phi)|(c_{\mathcal{M}} \wedge \psi)$$

Theorem 4.6.6 gives

$$(c_{\mathcal{M}} \wedge \phi)|(c_{\mathcal{M}} \wedge \psi) \rightarrow ((c_{\mathcal{M}} \wedge \phi)|c_{\mathcal{M}}) \wedge ((c_{\mathcal{M}} \wedge \phi)|\psi))$$

Hence $\vdash (c_{\mathcal{M}} \wedge \phi|\psi) \rightarrow (c_{\mathcal{M}} \wedge \phi)|\psi$. □

**Theorem 5.8.17.** *If $\mathcal{M}$ is a finite context then $\vdash c_{\mathcal{M}} \rightarrow [\alpha]c_{\mathcal{M}}$*

*Proof.* Observe that, by applying axiom E19, we obtain

$$\vdash K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3 \rightarrow (\widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \wedge [\alpha]K_0\theta_1$$

If, further, we apply theorem 5.8.12 once, we obtain

$$\vdash (\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2) \wedge [\alpha]K_0\theta_1 \rightarrow \widetilde{K}_0\theta_3 \wedge [\alpha]\widetilde{K}_0\theta_2 \wedge [\alpha]K_0\theta_1, \ i.e.$$

$$\vdash (\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2) \wedge [\alpha]K_0\theta_1 \rightarrow \widetilde{K}_0\theta_3 \wedge [\alpha](\widetilde{K}_0\theta_2 \wedge K_0\theta_1)$$

Hence

$$\vdash (K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \rightarrow \widetilde{K}_0\theta_3 \wedge [\alpha](\widetilde{K}_0\theta_2 \wedge K_0\theta_1)$$

If we apply again theorem 5.8.12 we obtain

$$\vdash \widetilde{K}_0\theta_3 \wedge [a](\widetilde{K}_0\theta_2 \wedge K_0\theta_1) \rightarrow [\alpha](\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2 \wedge K_0\theta_1)$$

hence

$$\vdash (K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \rightarrow [\alpha](\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2 \wedge K_0\theta_1)$$

As $c_{\mathcal{M}} = K_0(\bigvee_{Q\in\mathcal{M}} c_Q) \wedge (\bigwedge_{Q\in\mathcal{M}} \widetilde{K}_0 c_Q)$, we can use the same idea, applying theorem 5.8.12 once for each process in $\mathcal{M}$ (being finite) and we obtain

$$\vdash c_{\mathcal{M}} \rightarrow [\alpha]c_{\mathcal{M}}$$

□

**Theorem 5.8.18.** *If $\mathcal{M}$ is a finite context then* $\vdash c_{\mathcal{M}} \rightarrow (K_Q\top \rightarrow K_Q c_{\mathcal{M}})$

*Proof.* Observe that, by applying axiom E20, we obtain

$$\vdash K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3 \rightarrow (\widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \wedge (K_Q\top \rightarrow K_Q K_0\theta_1)$$

If, further, we apply theorem 5.8.13 once, we obtain

$$\vdash (\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2) \wedge (K_Q\top \rightarrow K_Q K_0\theta_1) \rightarrow$$
$$\widetilde{K}_0\theta_3 \wedge (K_Q\top \rightarrow K_Q\widetilde{K}_0\theta_2) \wedge (K_Q\top \rightarrow K_Q K_0\theta_1),\ i.e.$$

$$\vdash (\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2) \wedge (K_Q\top \rightarrow K_Q K_0\theta_1) \rightarrow \widetilde{K}_0\theta_3 \wedge (K_Q\top \rightarrow (K_Q\widetilde{K}_0\theta_2 \wedge K_Q K_0\theta_1))$$

i.e., using 5.8.3,

$$\vdash (\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2) \wedge (K_Q\top \rightarrow K_Q K_0\theta_1) \rightarrow \widetilde{K}_0\theta_3 \wedge (K_Q\top \rightarrow K_Q(\widetilde{K}_0\theta_2 \wedge K_0\theta_1))$$

Hence

$$\vdash (K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \rightarrow \widetilde{K}_0\theta_3 \wedge (K_Q\top \rightarrow K_Q(\widetilde{K}_0\theta_2 \wedge K_0\theta_1))$$

If we apply again the theorems 5.8.13 and 5.8.3 we obtain

$$\vdash [\widetilde{K}_0\theta_3 \wedge (K_Q\top \rightarrow K_Q(\widetilde{K}_0\theta_2 \wedge K_0\theta_1))] \rightarrow [K_Q\top \rightarrow K_Q(\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2 \wedge K_0\theta_1)]$$

hence

$$\vdash (K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \rightarrow [K_Q\top \rightarrow K_Q(\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2 \wedge K_0\theta_1)]$$

Because $c_{\mathcal{M}} = K_0(\bigvee_{Q\in\mathcal{M}} c_Q) \wedge (\bigwedge_{Q\in\mathcal{M}} \widetilde{K}_0 c_Q)$, we can use the same idea, applying theorem 5.8.13 once for each process in $\mathcal{M}$ (being finite) and we obtain

$$\vdash c_{\mathcal{M}} \rightarrow (K_Q\top \rightarrow K_Q c_{\mathcal{M}})$$

□

Now we prove a context sensitive version of rule $\mathrm{E}_R1$.

**Theorem 5.8.19.** *If $\mathcal{M}$ is a finite context and $\vdash c_{\mathcal{M}} \to (\phi \to \psi)$ then $\vdash c_{\mathcal{M}} \to (\phi|\rho \to \psi|\rho)$.*

*Proof.* $\vdash c_{\mathcal{M}} \to (\phi \to \psi)$ implies $\vdash (c_{\mathcal{M}} \wedge \phi) \to \psi$ where we apply rule $\mathrm{E}_R1$ and obtain $\vdash (c_{\mathcal{M}} \wedge \phi)|\rho \to \psi|\rho$. But theorem 5.8.16 gives $\vdash (c_{\mathcal{M}} \wedge \phi|\rho) \to (c_{\mathcal{M}} \wedge \phi)|\rho$. Combining these two results we obtain
$\vdash (c_{\mathcal{M}} \wedge \phi|\rho) \to \psi|\rho$, i.e. $\vdash c_{\mathcal{M}} \to (\phi|\rho \to \psi|\rho)$. □

A context-sensitive version of theorem 4.6.8 is also available.

**Theorem 5.8.20.** *If for a finite context $\mathcal{M} \ni P$ and any decomposition $P \equiv Q|R$ we have*

$\vdash c_{\mathcal{M}} \to (c_Q \to \neg\phi)$ *or* $\vdash c_{\mathcal{M}} \to (c_R \to \neg\psi)$ *then* $\vdash c_{\mathcal{M}} \to (c_P \to \neg(\phi|\psi))$.

*Proof.* If $\vdash c_{\mathcal{M}} \to (c_Q \to \neg\phi)$ then we have, equivalently, $\vdash c_{\mathcal{M}} \wedge c_Q \to \neg\phi$, i.e. $\vdash c_Q \to (c_{\mathcal{M}} \to \neg\phi)$, hence $\vdash c_Q \to \neg(c_{\mathcal{M}} \wedge \phi)$.
Similarly $\vdash c_{\mathcal{M}} \to (c_R \to \neg\psi)$ gives $\vdash c_R \to \neg(c_{\mathcal{M}} \wedge \psi)$.

Hence the hypothesis of the theorem can be rewritten as: for any decomposition $P \equiv Q|R$ we have

$$\vdash c_Q \to \neg(c_{\mathcal{M}} \wedge \phi) \text{ or } \vdash c_R \to \neg(c_{\mathcal{M}} \wedge \psi).$$

Then we can apply theorem 4.6.8 and we obtain

$$\vdash c_P \to \neg((c_{\mathcal{M}} \wedge \phi)|(c_{\mathcal{M}} \wedge \psi)) \tag{5.8.2}$$

But theorem 5.8.15 entails $\vdash c_{\mathcal{M}} \wedge \phi|\psi \to (c_{\mathcal{M}} \wedge \phi)|(c_{\mathcal{M}} \wedge \psi)$, hence $\vdash \neg((c_{\mathcal{M}} \wedge \phi)|(c_{\mathcal{M}} \wedge \psi)) \to \neg(c_{\mathcal{M}} \wedge \phi|\psi)$, and applying this result to (5.8.2), we obtain

$\vdash c_P \to \neg(c_{\mathcal{M}} \wedge \phi|\psi)$ *that is equivalent with* $\vdash c_{\mathcal{M}} \to (c_P \to \neg(\phi|\psi))$

□

Further we prove a context-sensitive version of rule $\mathrm{E}_R2$.

**Theorem 5.8.21.** *If* $\vdash c_{\mathcal{M}} \to \phi$ *then* $\vdash c_{\mathcal{M}} \to [\alpha]\phi$.

*Proof.* If we apply rule $\mathrm{E}_R2$ to $\vdash c_{\mathcal{M}} \to \phi$ we obtain $\vdash [\alpha](c_{\mathcal{M}} \to \phi)$. But axiom E8 gives $\vdash [\alpha](c_{\mathcal{M}} \to \phi) \to ([\alpha]c_{\mathcal{M}} \to [\alpha]\phi)$, hence $\vdash [\alpha]c_{\mathcal{M}} \to [\alpha]\phi$. Theorem 5.8.17 proves that $\vdash c_{\mathcal{M}} \to [\alpha]c_{\mathcal{M}}$ which gives further $\vdash c_{\mathcal{M}} \to [\alpha]\phi$. □

The next result is a context-sensitive variant of rule $\mathrm{E}_R5$.

**Theorem 5.8.22.** *If* $\vdash c_{\mathcal{M}} \to \phi$ *then* $\vdash c_{\mathcal{M}} \to (K_Q\top \to K_Q\phi)$.

*Proof.* If we apply rule $\mathrm{E}_R5$ to $\vdash c_{\mathcal{M}} \to \phi$, we obtain

$$\vdash K_Q\top \to K_Q(c_{\mathcal{M}} \to \phi)$$

But axiom E13 gives further $\vdash K_Q(c_{\mathcal{M}} \to \phi) \to (K_Qc_{\mathcal{M}} \to K_Q\phi)$. Hence $\vdash K_Q\top \wedge K_Qc_{\mathcal{M}} \to K_Q\phi$ that is equivalent with

$$\vdash K_Qc_{\mathcal{M}} \to (K_Q\top \to K_Q\phi)$$

Now, theorem 5.8.18 ensures that $\vdash c_{\mathcal{M}} \to (K_Q\top \to K_Qc_{\mathcal{M}})$.
Hence $\vdash c_{\mathcal{M}} \to (K_Q\top \to K_Q\phi)$. □

**Theorem 5.8.23.** *If* $\vdash c_{\mathcal{M}} \to (K_Q\psi \to \phi)$ *then* $\vdash c_{\mathcal{M}} \to (K_Q\psi \to K_Q\phi)$.

*Proof.* We apply theorem 5.8.22 to $\vdash c_{\mathcal{M}} \rightarrow (K_Q\psi \rightarrow \phi)$ and we obtain $\vdash c_{\mathcal{M}} \rightarrow (K_Q\top \rightarrow K_Q(K_Q\psi \rightarrow \phi))$, i.e. $\vdash (c_{\mathcal{M}} \wedge K_Q\top) \rightarrow K_Q(K_Q\psi \rightarrow \phi)$. But axiom E13 gives $\vdash K_Q(K_Q\psi \rightarrow \phi) \rightarrow (K_QK_Q\psi \rightarrow K_Q\phi)$. Now if we use theorem 5.8.4 we obtain further

$$\vdash K_Q(K_Q\psi \rightarrow \phi) \rightarrow (K_Q\psi \rightarrow K_Q\phi)$$

All these proved that $\vdash (c_{\mathcal{M}} \wedge K_Q\top) \rightarrow (K_Q\psi \rightarrow K_Q\phi)$, i.e.

$$\vdash c_{\mathcal{M}} \rightarrow (K_Q\top \rightarrow (K_Q\psi \rightarrow K_Q\phi))$$

which is equivalent with $\vdash c_{\mathcal{M}} \rightarrow (K_Q\top \wedge K_Q\psi \rightarrow K_Q\phi)$.
Theorem 5.8.2 proved that $\vdash K_Q\psi \rightarrow K_Q\top$, result which, combined with the previous one, gives further $\vdash c_{\mathcal{M}} \rightarrow (K_Q\psi \rightarrow K_Q\phi)$. □

**Theorem 5.8.24.** *If $Q|R \in \mathcal{M}$ then $\vdash c_{\mathcal{M}} \rightarrow (c_Q|c_R \rightarrow \neg\phi)$ implies $\vdash c_{\mathcal{M}} \rightarrow \neg K_Q\phi$.*

*Proof.* Because $\vdash c_R \rightarrow \top$, rule $E_R1$ gives $\vdash c_Q|c_R \rightarrow c_Q|\top$ that gives further $\vdash c_{\mathcal{M}} \rightarrow (c_Q|c_R \rightarrow c_Q|\top)$. Combining this result with the hypothesis of the theorem, $\vdash c_{\mathcal{M}} \rightarrow (c_Q|c_R \rightarrow \neg\phi)$, we obtain

$$\vdash (c_{\mathcal{M}} \wedge c_Q|c_R) \rightarrow (c_Q|\top \wedge \neg\phi), \text{ i.e. } \vdash c_{\mathcal{M}} \rightarrow (c_Q|c_R \rightarrow (c_Q|\top \wedge \neg\phi))$$

But $\vdash (c_Q|\top \wedge \neg\phi) \leftrightarrow \neg(c_Q|\top \rightarrow \phi)$, hence

$$\vdash c_{\mathcal{M}} \rightarrow (c_Q|c_R \rightarrow \neg(c_Q|\top \rightarrow \phi)) \tag{5.8.3}$$

Axiom E14 ensure that $\vdash K_0(c_Q|\top \rightarrow \phi) \rightarrow (c_Q|\top \rightarrow \phi)$ or, equivalently, $\vdash \neg(c_Q|\top \rightarrow \phi) \rightarrow \neg K_0(c_Q|\top \rightarrow \phi)$, that, used in (5.8.3) gives

$$\vdash c_{\mathcal{M}} \rightarrow (c_Q|c_R \rightarrow \neg K_0(c_Q|\top \rightarrow \phi)) \tag{5.8.4}$$

But theorem 5.8.1 gives $\vdash K_0\top$, that can be used in (5.8.4) providing

$$\vdash c_{\mathcal{M}} \rightarrow (c_Q|c_R \rightarrow (K_0\top \wedge \neg K_0(c_Q|\top \rightarrow \phi))) \tag{5.8.5}$$

The negative introspection, axiom E16, infers

$$\vdash (K_0\top \wedge \neg K_0(c_Q|\top \rightarrow \phi)) \rightarrow K_0\neg K_0(c_Q|\top \rightarrow \phi) \tag{5.8.6}$$

Combining (5.8.5) and (5.8.6) we obtain

$$\vdash c_{\mathcal{M}} \rightarrow (c_Q|c_R \rightarrow K_0\neg K_0(c_Q|\top \rightarrow \phi)) \tag{5.8.7}$$

But (5.8.7) is equivalent with $\vdash (c_{\mathcal{M}} \wedge c_Q|c_R) \rightarrow K_0\neg K_0(c_Q|\top \rightarrow \phi)$, and because $Q|R \in \mathcal{M}$, we can apply rule $\mathrm{E}_R6$ and obtain

$$\vdash c_{\mathcal{M}} \rightarrow \neg K_0(K_Q\top \rightarrow \phi) \tag{5.8.8}$$

But from axiom E17 we derive $\vdash K_Q\phi \rightarrow K_0(K_Q\top \rightarrow \phi)$, hence

$$\vdash \neg K_0(K_Q\top \rightarrow \phi) \rightarrow \neg K_Q\phi \tag{5.8.9}$$

Combining (5.8.8) with (5.8.9) we obtain $\vdash c_{\mathcal{M}} \rightarrow \neg K_Q\phi$, q.e.d. □

The next result is a context-sensitive version of theorem 4.6.7.

**Theorem 5.8.25.** *If* $\vdash c_{\mathcal{M}} \rightarrow (\phi \rightarrow \psi)$ *and* $\vdash c_{\mathcal{M}} \rightarrow (\rho \rightarrow \theta)$ *then* $\vdash c_{\mathcal{M}} \rightarrow (\phi|\rho \rightarrow \psi|\theta)$.

*Proof.* To $\vdash c_{\mathcal{M}} \rightarrow (\phi \rightarrow \psi)$ we can apply theorem 5.8.19 and we obtain $\vdash c_{\mathcal{M}} \rightarrow (\phi|\rho \rightarrow \psi|\rho)$, i.e. $\vdash (c_{\mathcal{M}} \wedge \phi|\rho) \rightarrow \psi|\rho$ which implies

$$\vdash (c_{\mathcal{M}} \wedge \phi|\rho) \rightarrow (c_{\mathcal{M}} \wedge \psi|\rho) \tag{5.8.10}$$

The same theorem 5.8.19 can be applied to $\vdash c_{\mathcal{M}} \to (\rho \to \theta)$ giving $\vdash c_{\mathcal{M}} \to (\psi|\rho \to \psi|\theta)$, i.e.

$$\vdash (c_{\mathcal{M}} \wedge \psi|\rho) \to \psi|\theta \tag{5.8.11}$$

Further, combining (5.8.10) and (5.8.11) we derive $\vdash (c_{\mathcal{M}} \wedge \phi|\psi) \to \psi|\theta$, hence $\vdash c_{\mathcal{M}} \to (\phi|\psi \to \psi|\theta)$. □

We prove further a contextual version of theorem 4.6.9.

**Theorem 5.8.26.** *If* $\vdash c_{\mathcal{M}} \to (\phi \to \psi)$ *then* $\vdash c_{\mathcal{M}} \to (\langle\alpha\rangle\phi \to \langle\alpha\rangle\psi)$.

*Proof.* $\vdash c_{\mathcal{M}} \to (\phi \to \psi)$ implies $\vdash c_{\mathcal{M}} \to (\neg\psi \to \neg\phi)$ where, applying theorem 5.8.21, we obtain $\vdash c_{\mathcal{M}} \to [\alpha](\neg\psi \to \neg\phi)$. But axiom E8 gives $\vdash [\alpha](\neg\psi \to \neg\phi) \to ([\alpha]\neg\psi \to [\alpha]\neg\phi)$. Hence $\vdash c_{\mathcal{M}} \to ([\alpha]\neg\psi \to [\alpha]\neg\phi)$, i.e. $\vdash c_{\mathcal{M}} \to (\neg\langle\alpha\rangle\psi \to \neg\langle\alpha\rangle\phi)$. Concluding, $\vdash c_{\mathcal{M}} \to (\langle\alpha\rangle\phi \to \langle\alpha\rangle\psi)$. □

The next result is a variant of theorem 4.6.13, but sensitive to the context.

**Theorem 5.8.27.**

$$\textit{If } \vdash c_{\mathcal{M}} \to (\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \to \phi) \textit{ then } \vdash c_{\mathcal{M}} \to (c_P \to [\alpha]\phi)$$

*Proof.* If $\vdash c_{\mathcal{M}} \to (\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \to \phi)$ then theorem 5.8.21 gives $\vdash c_{\mathcal{M}} \to [\alpha](\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \to \phi)$ and further axiom E8 gives

$$\vdash c_{\mathcal{M}} \to ([\alpha]\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \to [\alpha]\phi)$$

But theorem 4.6.12 gives

$$\vdash c_P \to [\alpha]\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\}$$

hence $\vdash c_{\mathcal{M}} \wedge c_P \to [\alpha]\phi$, i.e. $\vdash c_{\mathcal{M}} \to (c_P \to [\alpha]\phi)$. □

## 5.9 Completeness of $\mathcal{L}^{\mathfrak{S}}_{DES}$ against process semantics

Now we will prove the completeness of $\mathcal{L}^{\mathfrak{S}}_{DES}$ with respect to process semantics. We recall that completeness ensures that everything that can be derived in the semantics can be proved in the syntax. In this way we have the possibility to syntactically verify properties.

In the context of a decidable system, as ours is, the completeness provides a powerful tool for making predictions on the evolution of the system we analyze. Indeed, knowing the state of our system, we can characterize it syntactically. And because any other state can be characterized, we can project our problem into the syntax and verify its satisfiability. Hence if our system can reach that state, we will obtain that the formula is satisfiable and the method will provide also a minimal model that satisfies it. Thus we made a prediction without investigating (simulating) the full evolution of the system that might cause, sometimes, unsolvable computational problems (usually the time is branching generating exponential complexity).

As in the case of the system $\mathcal{L}_{DS}$, we start by proving a lemma that provides a syntactic characterization of the satisfiability. The intuition is that, because $c_P$ and $c_{\mathcal{M}}$ are characteristic formulas, we should have an equivalence between $\mathcal{M}, P \models \phi$ and $\vdash c_{\mathcal{M}} \wedge c_P \to \phi$ (of course for finite contexts) as both can be read as *the process $P$ in the context $\mathcal{M}$ has the property $\phi$.*

**Lemma 5.9.1.** *If $\mathcal{M}$ is a finite context then $\mathcal{M}, P \models \phi$ iff $\vdash c_{\mathcal{M}} \wedge c_P \to \phi$.*

*Proof.* ($\Longrightarrow$) We prove it by induction on the syntactical structure of $\phi$.

- **The case $\phi = 0$:** $\mathcal{M}, P \models 0$ implies $P \equiv 0$. But $c_0 = 0$ and $\vdash 0 \to 0$, hence $\vdash 0 \wedge c_{\mathcal{M}} \to 0$. This gives $\vdash c_{\mathcal{M}} \wedge c_P \to \phi$.

- **The case $\phi = \top$:** we have always $\mathcal{M}, P \models \top$ and $\vdash c_P \wedge c_{\mathcal{M}} \rightarrow \top$, hence $\vdash c_P \wedge c_{\mathcal{M}} \rightarrow \phi$.

- **The case $\phi = \phi_1 \wedge \phi_2$:** $\mathcal{M}, P \models \phi$ iff $\mathcal{M}, P \models \phi_1$ and $\mathcal{M}, P \models \phi_2$. Further, using the inductive hypothesis, we obtain $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \phi_1$ and $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \phi_2$. Hence $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow (\phi_1 \wedge \phi_2)$, i.e. $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \phi$.

- **The case $\phi = \phi_1 | \phi_2$:** $\mathcal{M}, P \models \phi$ iff $P \equiv Q|R$, $\mathcal{M}, Q \models \phi_1$ and $\mathcal{M}, R \models \phi_2$.
  Using the inductive hypothesis,
  $\vdash c_{\mathcal{M}} \wedge c_Q \rightarrow \phi_1$ and $\vdash c_{\mathcal{M}} \wedge c_R \rightarrow \phi_2$, i.e.
  $\vdash c_{\mathcal{M}} \rightarrow (c_Q \rightarrow \phi_1)$ and $\vdash c_{\mathcal{M}} \rightarrow (c_R \rightarrow \phi_2)$.
  Hence, using theorem 5.8.25 we obtain $\vdash c_{\mathcal{M}} \rightarrow (c_Q | c_R \rightarrow \phi_1 | \phi_2)$, i.e. $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \phi$.

- **The case $\phi = K_Q \top$:** $\mathcal{M}, P \models K_Q \top$ iff $P \equiv Q|R$, iff $c_P = c_Q | c_R$. Using rule $\mathrm{E}_R 1$ we obtain $\vdash c_Q | c_R \rightarrow c_Q | \top$, further using axiom E12 $\vdash c_Q | c_R \rightarrow K_Q \top$, i.e. $\vdash c_P \rightarrow K_Q \top$. Hence $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \phi$.

- **The case $\phi = K_Q \psi$:** $\mathcal{M}, P \models K_Q \psi$, and because $\vdash K_Q \psi \rightarrow K_Q \top$ (by theorem 5.8.2), using the soundness, we obtain that $\mathcal{M}, P \models K_Q \top$. Now, we apply the previous case that gives

$$\vdash c_{\mathcal{M}} \wedge c_P \rightarrow K_Q \top \tag{5.9.1}$$

  $\mathcal{M}, P \models K_Q \psi$ is equivalent with $P \equiv Q|R$ and for any $Q|S \in \mathcal{M}$ we have $\mathcal{M}, Q|S \models \psi$. Then the inductive hypothesis gives

$$\text{for any } Q|S \in \mathcal{M} \text{ we have } \vdash (c_{\mathcal{M}} \wedge c_Q | c_S) \rightarrow \psi \tag{5.9.2}$$

  Consider now a process $Q|S \notin \mathcal{M}$. Because $\mathcal{M}$ is finite, we apply theorem 5.8.14 and obtain $\vdash c_{\mathcal{M}} \rightarrow \neg(c_Q | c_S)$ or equivalent,
  $\vdash c_{\mathcal{M}} \wedge (c_Q | c_S) \rightarrow \bot$. But $\vdash \bot \rightarrow \psi$, hence

$$\text{for any } Q|S \notin \mathcal{M} \text{ we have } \vdash (c_{\mathcal{M}} \wedge c_Q | c_S) \rightarrow \psi \tag{5.9.3}$$

Now (5.9.2) and (5.9.3) together give

$$\text{for any } S \in \mathcal{M} \text{ we have } \vdash (c_{\mathcal{M}} \wedge c_Q | c_S) \to \psi \tag{5.9.4}$$

i.e., using theorem 4.6.5,

$$\vdash (c_{\mathcal{M}} \wedge c_Q | \bigvee_{S \in \mathcal{M}} c_S) \to \psi \tag{5.9.5}$$

But

$$\vdash K_0(\bigvee_{S \in \mathcal{M}} c_S) \to \bigvee_{S \in \mathcal{M}} c_S, \ \textit{hence} \ \vdash c_{\mathcal{M}} \to \bigvee_{S \in \mathcal{M}} c_S$$

Now, we can apply rule $\mathrm{E}_R1$ and obtain

$$\vdash c_Q | c_{\mathcal{M}} \to c_Q | \bigvee_{S \in \mathcal{M}} c_S, \ \textit{hence} \ \vdash (c_Q | c_{\mathcal{M}} \wedge c_{\mathcal{M}}) \to (c_Q | \bigvee_{S \in \mathcal{M}} c_S \wedge c_{\mathcal{M}})$$

In this point, using (5.9.5) we obtain

$$\vdash (c_Q | c_{\mathcal{M}} \wedge c_{\mathcal{M}}) \to \psi \tag{5.9.6}$$

We have $\vdash c_{\mathcal{M}} \to (\top \to c_{\mathcal{M}})$ and $\vdash c_{\mathcal{M}} \to (c_Q \to c_Q)$ where from, applying theorem 5.8.19, we can derive $\vdash c_{\mathcal{M}} \to (c_Q | \top \to c_Q | c_{\mathcal{M}})$, i.e. $\vdash c_M \wedge c_Q | \top \to c_Q | c_{\mathcal{M}}$ and further

$$\vdash (c_M \wedge c_Q | \top) \to (c_{\mathcal{M}} \wedge c_Q | c_{\mathcal{M}})$$

Using this result together with (5.9.6), we obtain further

$$\vdash (c_M \wedge c_Q | \top) \to \psi, \ \textit{i.e.} \ \vdash c_M \to (c_Q | \top \to \psi)$$

where we can apply axiom E12 that gives

$$\vdash c_{\mathcal{M}} \to (K_Q \top \to \psi)$$

applying theorem 5.8.23, we obtain

$$\vdash c_{\mathcal{M}} \to (K_Q \top \to K_Q \psi), \ \textit{i.e.} \ \vdash (c_{\mathcal{M}} \wedge K_Q \top) \to K_Q \psi \tag{5.9.7}$$

But (5.9.1) gives

$$\vdash c_{\mathcal{M}} \wedge c_P \rightarrow K_Q \top \text{ where from } \vdash (c_{\mathcal{M}} \wedge c_P) \rightarrow (c_{\mathcal{M}} \wedge K_Q \top)$$

and using this in (5.9.7),

$$\vdash (c_{\mathcal{M}} \wedge c_P) \rightarrow K_Q \psi \; i.e. \; \vdash (c_{\mathcal{M}} \wedge c_P) \rightarrow \phi.$$

- **The case $\phi = \langle\alpha\rangle\psi$:** $\mathcal{M}, P \models \langle\alpha\rangle\psi$ means that exists $P' \in \mathcal{M}$ such that $P \xrightarrow{\alpha} P'$ and $\mathcal{M}, P' \models \psi$. Then the inductive hypothesis gives

  $$\vdash c_{\mathcal{M}} \wedge c_{P'} \rightarrow \psi$$

  $P \xrightarrow{\alpha} P'$ means that $P \equiv \alpha.R|S$ and $P' \equiv R|S$, so $c_P = (\langle\alpha\rangle c_R \wedge 1)|c_S$ and $c_{P'} = c_R|c_S$. So $\vdash c_{\mathcal{M}} \wedge c_R|c_S \rightarrow \psi$, i.e. $\vdash c_{\mathcal{M}} \rightarrow (c_R|c_S \rightarrow \psi)$ and using theorem 5.8.26

  $$\vdash c_{\mathcal{M}} \rightarrow (\langle\alpha\rangle(c_R|c_S) \rightarrow \langle\alpha\rangle\psi) \tag{5.9.8}$$

  theorem 4.6.6 gives $\vdash c_P \rightarrow \langle\alpha\rangle c_R|c_S \wedge 1|c_S$, hence

  $$\vdash c_P \rightarrow \langle\alpha\rangle c_R|c_S \tag{5.9.9}$$

  Axiom E7 gives

  $$\vdash \langle\alpha\rangle c_R|c_S \rightarrow \langle\alpha\rangle(c_R|c_S) \tag{5.9.10}$$

  Hence, from (5.9.8), (5.9.9) and (5.9.10) we derive

  $$\vdash c_{\mathcal{M}} \rightarrow (c_P \rightarrow \langle\alpha\rangle\psi), \; i.e. \; \vdash (c_{\mathcal{M}} \wedge c_P) \rightarrow \langle\alpha\rangle\psi$$

- **The case $\phi = \neg\psi$:** we argue by induction on the syntactical structure of $\psi$.

  - **the subcase $\psi = 0$:** $\mathcal{M}, P \models \neg 0$ means that $P \not\equiv 0$. Then we can apply theorem 4.7.3 and obtain $\vdash c_P \rightarrow \neg 0$.
    So $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \neg 0$.

– **the subcase** $\psi = \top$**:** is an impossible one as we cannot have $\mathcal{M}, P \models \bot$.

– **the subcase** $\psi = \psi_1 \wedge \psi_2$**:** $\mathcal{M}, P \models \neg(\psi_1 \wedge \psi_2)$ is equivalent with $\mathcal{M}, P \models \neg\psi_1 \vee \neg\psi_2$, i.e. $\mathcal{M}, P \models \neg\psi_1$ or $\mathcal{M}, P \models \neg\psi_2$. By the inductive hypothesis, $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \neg\psi_1$ or $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \neg\psi_2$, where from we obtain $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \psi$

– **the subcase** $\psi = \neg\psi_1$**:** $\mathcal{M}, P \models \neg\psi$ is equivalent with $\mathcal{M}, P \models \neg\neg\psi_1$, i.e. $\mathcal{M}, P \models \psi_1$ where we can use the inductive hypothesis $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \psi_1$ which is equivalent with $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \phi$.

– **the subcase** $\psi = \psi_1|\psi_2$**:** $\mathcal{M}, P \models \neg(\psi_1|\psi_2)$ means that for any parallel decomposition of $P \equiv Q|R$, $\mathcal{M}, Q \models \neg\psi_1$ or $\mathcal{M}, R \models \neg\psi_2$. These imply, using the inductive hypothesis, that for any decomposition $P \equiv Q|R$ we have

$$\vdash c_{\mathcal{M}} \rightarrow (c_Q \rightarrow \neg\psi_1) \text{ or } \vdash c_{\mathcal{M}} \rightarrow (c_R \rightarrow \neg\psi_2)$$

then we can apply theorem 5.8.20 that gives

$$\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \neg\psi.$$

– **the subcase** $\psi = K_0\psi_1$**:** $\mathcal{M}, P \models \neg K_0\psi_1$ means $\exists R \in \mathcal{M}$ such that $\mathcal{M}, R \models \neg\psi_1$. Using the inductive hypothesis,
$\vdash c_{\mathcal{M}} \wedge c_R \rightarrow \neg\psi_1$, i.e. $\vdash c_{\mathcal{M}} \rightarrow (c_R|c_0 \rightarrow \neg\psi_1)$. Now theorem 5.8.24 gives $\vdash c_{\mathcal{M}} \rightarrow \neg K_0\psi_1$, hence $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \neg K_0\psi_1$.

– **the subcase** $\psi = K_Q\psi_1$**,** $Q \not\equiv 0$**:** we distinguish two cases

  * **the sub-subcase** $\psi_1 = \top$**:** $\mathcal{M}, P \models \neg K_Q\top$ implies that $Q$ is not a subprocess of $P$. Then for any $R \in \mathcal{M}$ we have $P \not\equiv Q|R$. Then theorem 4.7.3 gives us $\vdash c_{Q|R} \rightarrow \neg c_P$, i.e. $\vdash c_Q|c_R \rightarrow \neg c_P$. From here we can infer

$$\vdash c_Q|\bigvee_{S\in\mathcal{M}} c_S \rightarrow \neg c_P \tag{5.9.11}$$

But

$$\vdash K_0(\bigvee_{S\in\mathcal{M}} c_S) \to \bigvee_{S\in\mathcal{M}} c_S,\ \textit{hence}\ \vdash c_{\mathcal{M}} \to \bigvee_{S\in\mathcal{M}} c_S$$

Now, we can apply rule $\mathrm{E}_R1$ and obtain

$$\vdash c_Q|c_{\mathcal{M}} \to c_Q|\bigvee_{S\in\mathcal{M}} c_S$$

In this point, using (5.9.11) we obtain

$$\vdash c_Q|c_{\mathcal{M}} \to \neg c_P \tag{5.9.12}$$

We have $\vdash c_{\mathcal{M}} \to (\top \to c_{\mathcal{M}})$ and $\vdash c_{\mathcal{M}} \to (c_Q \to c_Q)$ where from, applying theorem 5.8.19, we can derive $\vdash c_{\mathcal{M}} \to (c_Q|\top \to c_Q|c_{\mathcal{M}})$, i.e. $\vdash c_M \wedge c_Q|\top \to c_Q|c_{\mathcal{M}}$ Using this result together with (5.9.12), we obtain further

$$\vdash (c_M \wedge c_Q|\top) \to \neg c_P,\ \textit{i.e.}\ \vdash c_M \wedge c_P \to \neg(c_Q|\top)$$

and axiom E12 gives

$$\vdash c_{\mathcal{M}} \wedge c_P \to \neg K_Q\top.$$

∗ **the sub-subcase** $\psi_1 \neq \top$**:** we distinguish two more cases $\mathcal{M}, P \models \neg K_Q\top$ and $\mathcal{M}, P \models K_Q\top$.

· **if** $\mathcal{M}, P \models \neg K_Q\psi_1$ and $\mathcal{M}, P \models \neg K_Q\top$, we have $\vdash c_{\mathcal{M}} \wedge c_P \to \neg K_Q\top$ (proved before). Moreover, because $\vdash K_Q\psi_1 \to K_Q\top$ (theorem 5.8.2) we have $\vdash \neg K_Q\top \to \neg K_Q\psi_1$ which gives $\vdash c_{\mathcal{M}} \wedge c_P \to \neg K_Q\psi_1$.

· **if** $\mathcal{M}, P \models \neg K_Q\psi_1$ and $\mathcal{M}, P \models K_Q\top$, $\exists Q|S \in \mathcal{M}$ with $\mathcal{M}, S|Q \models \neg\psi_1$. Using the inductive hypothesis we obtain $\vdash c_{\mathcal{M}} \to (c_S|c_Q \to \neg\psi_1)$ and from theorem 5.8.24 that $\vdash c_{\mathcal{M}} \to \neg K_Q\psi_1$. Hence $\vdash c_{\mathcal{M}} \wedge c_P \to \neg K_Q\psi_1$.

– **the subcase** $\psi = \langle\alpha\rangle\psi_1$**:** $\mathcal{M}, P \models \neg\langle\alpha\rangle\psi_1$ is equivalent with $\mathcal{M}, P \models [\alpha]\neg\psi_1$.

If there is a process $Q \in \mathcal{M}$ such that $P \xrightarrow{\alpha} Q$, then for any $Q \in \mathcal{M}$ such that $P \xrightarrow{\alpha} Q$ we have $\mathcal{M}, Q \models \neg\psi_1$. Using the inductive hypothesis we obtain that for any $Q \in \mathcal{M}$ such that $P \xrightarrow{\alpha} Q$ we have $\vdash c_{\mathcal{M}} \wedge c_Q \rightarrow \neg\psi_1$, i.e.

$$\vdash c_{\mathcal{M}} \wedge \bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \rightarrow \neg\psi_1$$

or equivalently

$$\vdash c_{\mathcal{M}} \rightarrow (\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \rightarrow \neg\psi_1)$$

Using theorem 5.8.27, we obtain $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow [\alpha]\neg\psi_1$.
If there is no process $Q \in \mathcal{M}$ such that $P \xrightarrow{\alpha} Q$ then theorem 4.6.11 gives $\vdash c_P \rightarrow [\alpha]\bot$. But $\vdash \psi_1 \rightarrow \top$, hence $\vdash [\alpha]\bot \rightarrow [\alpha]\neg\psi_1$. So, also in this case we have $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow [\alpha]\neg\psi_1$.

($\Longleftarrow$) Let $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \phi$. Suppose that $\mathcal{M}, P \not\models \phi$. Then $\mathcal{M}, P \models \neg\phi$. Using the reversed implication we obtain $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \neg\phi$, thus
$\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \bot$. But from corollary 5.4.2 we have $\mathcal{M}, P \models c_{\mathcal{M}} \wedge c_P$ which, using the soundness, gives $\mathcal{M}, P \models \bot$ impossible!
Hence $\mathcal{M}, P \models \phi$. □

We recall the definitions of provability, consistency, satisfiability and validity.

**Definition 5.9.1** (Provability and consistency)**.** We say that a formula $\phi \in \mathcal{F}^{\mathfrak{S}}_{DES}$ is *provable in* $\mathcal{L}^{\mathfrak{S}}_{DES}$ (or $\mathcal{L}^{\mathfrak{S}}_{DES}$*-provable* for short), if $\phi$ can be derived, as a theorem, using the axioms and the rules of $\mathcal{L}^{\mathfrak{S}}_{DES}$. We denote this by $\vdash \phi$.
We say that a formula $\phi \in \mathcal{F}^{\mathfrak{S}}_{DES}$ is *consistent in* $\mathcal{L}^{\mathfrak{S}}_{DES}$ (or $\mathcal{L}^{\mathfrak{S}}_{DES}$*-consistent* for short) if $\neg\phi$ is not $\mathcal{L}^{\mathfrak{S}}_{DES}$-provable.

**Definition 5.9.2** (Satisfiability and validity)**.** We call a formula $\phi \in \mathcal{F}^{\mathfrak{S}}_{DES}$ *satisfiable* if there exists a context $\mathcal{M}$ and a process $P \in \mathcal{M}$ such that $\mathcal{M}, P \models \phi$.
We call a formula $\phi \in \mathcal{F}^{\mathfrak{S}}_{DES}$ *validity* if for any context $\mathcal{M}$ and any process $P \in \mathcal{M}$ we have $\mathcal{M}, P \models \phi$. In such a situation we write $\models \phi$.
Given a context $\mathcal{M}$, we denote by $\mathcal{M} \models \phi$ the situation when for any $P \in \mathcal{M}$ we have $\mathcal{M}, P \models \phi$.

*Remark* 5.9.1. $\phi$ is satisfiable iff $\neg\phi$ is not a validity, and vice versa, $\phi$ is a validity iff $\neg\phi$ is not satisfiable.

**Lemma 5.9.2.** *If $\phi$ is $\mathcal{L}^{\mathfrak{S}}_{DES}$-consistent then exists a context $\mathcal{M}$ and a process $P \in \mathcal{M}$ such that $\mathcal{M}, P \models \phi$.*

*Proof.* Suppose that for any context $\mathcal{M}$ and any process $P \in \mathcal{M}$ we do not have $\mathcal{M}, P \models \phi$, i.e. we have $\mathcal{M}, P \models \neg\phi$.
Hence, for any finite context $\mathcal{M}$ and any process $P \in \mathcal{M}$ we have
$\mathcal{M}, P \models \neg\phi$.
Using lemma 5.9.1, we obtain $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \neg\phi$ for any finite context $\mathcal{M} \ni P$. Hence $\vdash c_{\mathcal{M}} \wedge \bigvee_{P\in\mathcal{M}} c_P \rightarrow \neg\phi$. But $\vdash c_{\mathcal{M}} \rightarrow \bigvee_{P\in\mathcal{M}} c_P$ which, combined with the previous result, implies $\vdash c_{\mathcal{M}} \rightarrow \neg\phi$.
Thus for each finite context $\mathcal{M}$ we have $\vdash c_{\mathcal{M}} \rightarrow \neg\phi$. But then for each context $\overline{M}$ with $[\![M]\!] \leq (\!|\neg\phi|\!)$ we have $\vdash c_{\overline{M}} \rightarrow \neg\phi$, because these contexts are finite. Each of these contexts belongs to $\mathfrak{M}_{(\!|\neg\phi|\!)}$ which is also finite. Hence, we can infer further

$$\vdash \bigvee_{[\![M]\!]\leq(\!|\neg\phi|\!)} c_{\overline{M}} \rightarrow \neg\phi$$

Now, applying rule $\mathrm{E}_R4$, we obtain $\vdash \neg\phi$. This contradicts with the hypothesis of consistency of $\phi$.

Hence, it exists a context $\mathcal{M}$ and a process $P \in \mathcal{M}$ such that

$$\mathcal{M}, P \models \phi$$

□

**Theorem 5.9.3** (Completeness)**.** *The $\mathcal{L}^{\mathfrak{S}}_{DES}$ system is complete with respect to process semantics.*

*Proof.* Suppose that $\phi$ is a valid formula with respect to our semantics, but $\phi$ is not provable in the system $\mathcal{L}^{\mathfrak{S}}_{DES}$. Then neither is $\neg\neg\phi$, so, by definition, $\neg\phi$ is $\mathcal{L}^{\mathfrak{S}}_{DES}$-consistent. It follows, from lemma 5.9.2, that $\neg\phi$ is satisfiable with respect to process semantics, contradicting the validity of $\phi$. □

## 5.10 Concluding remarks

In this chapter we developed the Dynamic Epistemic Spatial Logic, $\mathcal{L}^{\mathfrak{S}}_{DES}$, which extends the Dynamic Spatial system $\mathcal{L}_{DS}$ with epistemic operators meant to express global properties over contexts. We propose these operators as alternative to the *guarantee* operator, from the classic spatial logics, obtaining a logic adequately expressive and decidable.

For $\mathcal{L}^{\mathfrak{S}}_{DES}$ we proposed a Hilbert-style axiomatic system that was proved to be sound and complete with respect to process semantics. The axioms and rules considered are very similar to the classical axioms and rules in epistemic logic, and some derivable theorems state meaningful properties of epistemic agents.

By using the *structural bisimulation* and *pruning method*, we proved the finite model property for $\mathcal{L}^{\mathfrak{S}}_{DES}$ in relation to the semantics we considered.

Consequently, we have decidability for satisfiability, validity and model-checking problems.

# Chapter 6

# Extending Dynamic Epistemic Spatial Logic

In this chapter we consider an extension of the system $\mathcal{L}^{\mathfrak{S}}_{DES}$ with some special dynamic operators inspired by the semantics.

As the agents in the signature $\mathfrak{S}$ are *epistemic agents*, they have, between the other agents, a special ontological statute. This could mean that the external observer developed a special interest for them, or that they are the only agents known in a possible unknown large system. This being the situation, we might want more than an analysis of their knowledge and behavior. We might keep observing them and register their actions. In other words, we find it interesting to analyze the perspective that these agents can "mark" their actions. In this way, we will distinguish between the action $\alpha$ performed, in an anonymous way by the system, and the action $\alpha$ done by the system due to an agent $P \in \mathfrak{S}$.

Following this intuition, we consider, in addition to the actions in $\mathbb{A}$, also *composed actions* of type $(P : \alpha)$, which expresses the action $\alpha$ performed by $P$.

Considering the composed actions is not the same as augmenting $\mathbb{A}$, as $(P : \alpha)$ is still an $\alpha$ action. Hence, if we consider dynamic operators of type $\langle P : \alpha \rangle$, associated with the composed actions, we expect that $\vdash \langle P : \alpha \rangle \phi \rightarrow \langle \alpha \rangle \phi$, i.e. if our system can perform an action $\alpha$, performed by its agent $P \in \mathfrak{S}$, and after it satisfies $\phi$, we can also say that the system can perform an action $\alpha$ and then it satisfies $\phi$.

In this chapter we will present a new Dynamic Epistemic Spatial Logic, the system $\mathcal{L}^{\mathfrak{S}^+}_{DES}$, which extends the system $\mathcal{L}^{\mathfrak{S}}_{DES}$ considering, in the syntax, also dynamic operators for composed actions and the associated semantics.

We will develop a Hilbert-style axiomatic system and we will prove it sound and complete against the process semantics. The system will be just an extension of the axiomatic system of $\mathcal{L}^{\mathfrak{S}}_{DES}$ constructed by reconsidering the older axioms to be sensitive to the cases of the new dynamic operators. Some more axioms are added that reflect the peculiarities of the composed actions. Still, the similarities to the epistemic logic are still obvious and some meaningful theorems are derived and interpreted in the semantics.

We will also prove the finite model property for $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ against the process semantics, which entails decidability for validity, satisfiability and model checking.

## 6.1 Composed actions

In this section we define the composed actions and their transitions, and we will prove some properties for them.

**Definition 6.1.1** (Composed transitions)**.** We introduce the transition $P \xrightarrow{Q:\alpha} P'$ for the situation when $Q$ is the active subprocess of $P$ that does $\alpha$, i.e.

$$P \equiv Q|R,\ P' \equiv Q'|R \text{ and } Q \xrightarrow{\alpha} Q'$$

We denote such an action by $(Q : \alpha)$ and call it *composed action* to distinguish it from the usual actions in $\mathbb{A}$ called, hereafter, *basic actions*.

**Theorem 6.1.1.** *If $\alpha \in \mathbb{A}$ and $P \xrightarrow{Q|S:\alpha} P'$ then $P \xrightarrow{Q:\alpha} P'$ or $P \xrightarrow{S:\alpha} P'$*

**Definition 6.1.2** (Actions over a signature)**.** For a given signature $\mathfrak{S}$ we consider the set $\mathbb{A}^{\mathfrak{S}}$ of actions over $\mathfrak{S}$ defined by

$$\mathbb{A}^{\mathfrak{S}} = \mathbb{A} \cup \{(Q : \alpha) \mid \textit{for } Q \in \mathfrak{S} \textit{ and } \alpha \textit{ basic action active in } Q\}$$

Hereafter we use $a$ to range over $\mathbb{A}^{\mathfrak{S}}$, while $\alpha$ will be used to range over $\mathbb{A}$, as before.

**Theorem 6.1.2** (Behavioral simulation for composed actions)**.** *If $R \in \mathfrak{S}$ with $[\![R]\!] < (h, w)$, $P \xrightarrow{R:\alpha} P'$, and $P \approx_h^w Q$, then exists a transition $Q \xrightarrow{R:\alpha} Q'$ such that $P' \approx_{h-1}^{w-1} Q'$.*

*Proof.* Because $P \xrightarrow{R:\alpha} P'$, we have $P \equiv R|S$. Further, $P \approx_h^w Q$ and $[\![R]\!] < (h, w)$ gives, using theorem 3.3.8, $Q \equiv R|S'$. Hence, exists a transition $Q \xrightarrow{R:\alpha} Q'$.

We prove further that $P' \approx_{h-1}^{w-1} Q'$.
$P \xrightarrow{R:\alpha} P'$ gives $R \equiv \alpha.R'|R''$, so $P \equiv \alpha.R'|R''|S$ and $P' \equiv R'|R''|S$. Similarly, $Q \xrightarrow{R:\alpha} Q'$ gives $Q \equiv \alpha.R'|R''|S'$ with $Q' \equiv R'|R''|S'$. Because $P \xrightarrow{\alpha} P'$ and $Q \xrightarrow{\alpha} Q'$, using theorem 3.3.10 we can infer
$P' \approx_{h-1}^{w-1} Q'$. □

## 6.2 Syntax of $\mathcal{L}^{\mathfrak{S}^+}_{DES}$

The syntax of our new system $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ is obtained by adding to the syntax of $\mathcal{L}^{\mathfrak{S}}_{DES}$ the dynamic operators for composed actions.

**Definition 6.2.1.** Let $\mathfrak{S}$ be a finite signature over $\mathfrak{P}$. We define the language $\mathcal{F}^{\mathfrak{S}^+}_{DES}$ by adding dynamic operators for composed actions to $\mathcal{F}^{\mathfrak{S}}_{DES}$.

$$\phi := \ 0 \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi|\phi \mid K_Q\phi \mid \langle\alpha\rangle\phi \mid \langle Q:\alpha\rangle\phi$$

where $Q \in \mathfrak{S}$ is an agent that can perform $\alpha$.

As new dynamic operators have been considered, we propose for them the dual operators as well, defined as derived operators, for $Q \in \mathfrak{S}$, by:

$$[Q:\alpha]\phi \stackrel{def}{=} \neg\langle Q:\alpha\rangle\neg\phi$$

We extend the definition of size for a formula by adding to the definition for $\mathcal{F}^{\mathfrak{S}}_{DES}$ the clause for the new dynamic operator.

**Definition 6.2.2** (Size of a formula)**.** Let $R \in \mathfrak{S}$ and suppose that $[\![R]\!] = (h_R, w_R)$ and $(\!|\phi|\!) = (h, w)$. Then:

- $(\!|\langle R:\alpha\rangle\phi|\!) = (1 + max(h, h_R), 1 + max(w, w_R))$

## 6.3 Extending the process semantics

The semantics will just extend the process semantics for $\mathcal{L}^{\mathfrak{S}}_{DES}$ by adding the definition of satisfaction for the dynamic operators of the composed actions.

**Definition 6.3.1.** A model of $\mathcal{L}_{DES}^{\mathfrak{S}^+}$ is a context $\mathcal{M}$ for which we define the satisfaction relation by extending the satisfaction relation defined for $\mathcal{L}_{DES}^{\mathfrak{S}}$ with the rule:

- $\mathcal{M}, P \models \langle Q : \alpha \rangle \phi$ iff exists a transition $P \xrightarrow{Q:\alpha} P'$ and $\mathcal{M}, P' \models \phi$

In consequence, the semantics for the dual operator associated with a composed action will be:
$\mathcal{M}, P \models [Q : \alpha]\phi$ iff

- either $P$ cannot perform the composed action $(Q : \alpha)$
- or for each $P'$ such that $P \xrightarrow{Q:\alpha} P'$ we have $\mathcal{M}, P' \models \phi$

Because all the results concerning the characteristic formulas for processes and finite contexts do not involve composed action operators, they remain available and will be used, if needed.

## 6.4 Finite model property and decidability

We will now approach the problem of finite model property for the new system. Mainly we will follow the same steps as in proving it for the system $\mathcal{L}_{DES}^{\mathfrak{S}}$, see section 5.5, but reconsidering the theorems also for the case of composed dynamic operators.

For the beginning we prove that lemma 5.5.1 still holds.

**Lemma 6.4.1.** *If* $(\!|\phi|\!) = (h, w)$, $\mathcal{M}, P \models \phi$ *and* $(\mathcal{M}, P) \approx_h^w (\mathcal{N}, Q)$ *then* $\mathcal{N}, Q \models \phi$.

*Proof.* It can be proved by induction on the syntactical structure of $\phi$. We will prove, here, only the case of the composed dynamic operators. For the rest of the cases we recall the proof of theorem 5.5.1, where they have been already considered.

- **The case $\phi = \langle R : \alpha \rangle \phi'$ with $R \in \mathfrak{S}$:** suppose that $(\!|\phi'|\!) = (h', w')$ and $[\![R]\!] = (h_R, w_R)$.

  We have $(\!|\langle R : \alpha \rangle \phi'|\!) = (1 + max(h', h_R), 1 + max(w', w_R))$.

  $\mathcal{M}, P \models \langle R : \alpha \rangle \phi'$ means that $P \xrightarrow{R:\alpha} P'$ and $\mathcal{M}, P' \models \phi'$.

  Now $(\mathcal{M}, P) \approx^{1+max(w',w_R)}_{1+max(h',h_R)} (\mathcal{N}, Q)$ gives $P \approx^{1+max(w',w_R)}_{1+max(h',h_R)} Q$, and as $(h_R, w_R) < (1 + max(h', h_R), 1 + max(w', w_R))$, using theorem 6.1.2, we obtain that $Q \xrightarrow{R:\alpha} Q'$ and $P' \approx^{w'}_{h'} Q'$.

  $(\mathcal{M}, P) \approx^{1+max(w',w_R)}_{1+max(h',h_R)} (\mathcal{N}, Q)$ gives also $\mathcal{M} \approx^{1+max(w',w_R)}_{1+max(h',h_R)} \mathcal{N}$, so using theorem 3.6.1, $\mathcal{M} \approx^{w'}_{h'} \mathcal{N}$. Hence $(\mathcal{M}, P') \approx^{w'}_{h'} (\mathcal{N}, Q')$.

  From $\mathcal{M}, P' \models \phi'$ and $(\mathcal{M}, P') \approx^{w'}_{h'} (\mathcal{N}, Q')$, we obtain, by using the inductive hypothesis, that $\mathcal{N}, Q' \models \phi'$, and because $Q \xrightarrow{R:\alpha} Q'$, we obtain further that $\mathcal{N}, Q \models \phi$.

□

Lemma 6.4.1 entails the next theorem, which is the correspondent of theorem 5.5.2. It states that the satisfiability relation $\mathcal{M}, P \models \phi$ is preserved if we prune the context and the process to the size of $\phi$.

**Theorem 6.4.2.** *If $\mathcal{M}, P \models \phi$ and $(\!|\phi|\!) = (h, w)$ then $\mathcal{M}_{(h,w)}, P_{(h,w)} \models \phi$.*

*Proof.* By contexts pruning theorem 3.7.3, we have $\mathcal{M} \approx^h_w \mathcal{M}_{(h,w)}$. By process pruning theorem 3.4.1, we have $P \approx^h_w P_{(h,w)}$.
Hence $(\mathcal{M}, P) \approx^h_w (\mathcal{M}_{(h,w)}, P_{(h,w)})$. Further lemma 6.4.1 establishes $\mathcal{M}_{(h,w)}, P_{(h,w)} \models \phi$ q.e.d. □

As an immediate consequence, we can prove the finite model property. Indeed, because we can derive $\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \phi$ from $\mathcal{M}, P \models \phi$, we can decide on the satisfiability of $\phi$ only by browsing the couples $(\mathcal{M}, P)$ with $P \in \mathcal{M} \in \mathfrak{M}_{(\!|\phi|\!)}$, i.e. by a finite search.

**Theorem 6.4.3** (Finite model property)**.**

$$\textit{If } \mathcal{M}, P \models \phi \textit{ then } \exists \mathcal{N} \in \mathfrak{M}_{(\!|\phi|\!)} \textit{ and } Q \in \mathcal{N} \textit{ such that } \mathcal{N}, Q \models \phi$$

*Proof.* We can take directly $\mathcal{N} = \mathcal{M}_{(h,w)} \in \mathfrak{M}_{(h,w)}$ and $Q = P_{(h,w)} \in \mathcal{M}_{(h,w)}$, and theorem 6.4.2 proves the finite model property. □

Hence, as in the cases of the other systems, we can use these results to prove the decidability.

**Theorem 6.4.4** (Decidability)**.** *For $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ validity, satisfiability and model checking are decidable against the process semantics.*

## 6.5 Axioms of $\mathcal{L}^{\mathfrak{S}^+}_{DES}$

Hereafter we introduce and analyze a Hilbert-style axiomatic system for $\mathcal{L}^{\mathfrak{S}^+}_{DES}$. The system will be constructed on top of the system $\mathcal{L}^{\mathfrak{S}}_{DES}$ to which we will add some specific axioms describing the behavior of the dynamic operators for composed actions.

As before, we take the axioms and the rules of propositional logic, together with the spatial axioms and rules, as in $\mathcal{L}_{DS}$ and $\mathcal{L}^{\mathfrak{S}}_{DES}$. Some of the dynamic axioms and rules will be modified to fit also with the composed actions, and three more axioms, specific to composed actions, will

be added. In addition, the system $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ will contain also the class of epistemic axioms and rules, the same as $\mathcal{L}^{\mathfrak{S}}_{DES}$, the only difference being that axiom E19 will be modified for composed actions.

As for the syntax, we remind the reader that we will use $\alpha, \beta$ to range over $\mathbb{A}$, while for ranging over $\mathbb{A}^{\mathfrak{S}}$ we will use $a$ (recall that $\mathbb{A} \subset \mathbb{A}^{\mathfrak{S}}$).

### Spatial axioms

**Axiom $\mathbf{E}^+$ 1.** $\vdash \top|\bot \to \bot$

**Axiom $\mathbf{E}^+$ 2.** $\vdash \phi|0 \leftrightarrow \phi$

**Axiom $\mathbf{E}^+$ 3.** $\vdash \phi|\psi \to \psi|\phi$

**Axiom $\mathbf{E}^+$ 4.** $\vdash (\phi|\psi)|\rho \to \phi|(\psi|\rho)$

**Axiom $\mathbf{E}^+$ 5.** $\vdash \phi|(\psi \vee \rho) \to (\phi|\psi) \vee (\phi|\rho)$

**Axiom $\mathbf{E}^+$ 6.** $\vdash (c_P \wedge \phi|\psi) \to \bigvee_{P \equiv Q|R}(c_Q \wedge \phi)|(c_R \wedge \psi)$

### Spatial rules

**Rule $\mathbf{E}^+_R$ 1.** *If* $\vdash \phi \to \psi$ *then* $\vdash \phi|\rho \to \psi|\rho$

Observe that no modification concerns these axioms. In consequence, all the spatial results proved in section 4.6 hold for our system.

### Dynamic axioms

**Axiom $\mathbf{E}^+$ 7.** $\vdash \langle a\rangle\phi|\psi \to \langle a\rangle(\phi|\psi)$

**Axiom $\mathbf{E}^+$ 8.** $\vdash [a](\phi \to \psi) \to ([a]\phi \to [a]\psi)$

**Axiom $\mathbf{E}^+$ 9.** $\vdash 0 \rightarrow [\alpha]\bot$

**Axiom $\mathbf{E}^+$ 10.** *If* $\beta \neq \alpha_i$ *for* $i = 1..n$ *then* $\vdash \langle !\alpha_1\rangle\top|...|\langle !\alpha_n\rangle\top \rightarrow [\beta]\bot$

**Axiom $\mathbf{E}^+$ 11.** $\vdash \langle !\alpha\rangle\phi \rightarrow [\alpha]\phi$

**Dynamic axioms for composed actions**

**Axiom $\mathbf{E}^+$ 12.** $\vdash \langle Q : \alpha\rangle\top \rightarrow c_Q|\top$.

**Axiom $\mathbf{E}^+$ 13.** *If* $R \in \mathfrak{S}$ *then* $\vdash c_R \rightarrow (\langle\alpha\rangle\phi \leftrightarrow \langle R : \alpha\rangle\phi)$

**Axiom $\mathbf{E}^+$ 14.** $\vdash \langle P : \alpha\rangle\phi \wedge \langle P|Q : \alpha\rangle\top \rightarrow \langle P|Q : \alpha\rangle\phi$

**Dynamic rules**

**Rule $\mathbf{E}^+_R$ 2.** *If* $\vdash \phi$ *then* $\vdash [a]\phi$

**Rule $\mathbf{E}^+_R$ 3.** *If* $\vdash \phi \rightarrow [a]\phi' \wedge \langle a\rangle\top$ *and* $\vdash \psi \rightarrow [a]\psi'$
*then* $\vdash \phi|\psi \rightarrow [a](\phi|\psi' \vee \phi'|\psi)$

**Rule $\mathbf{E}^+_R$ 4.** *If* $\vdash \bigvee_{[\![M]\!]\leq(\!|\phi|\!)} c_{\overline{M}} \rightarrow \phi$ *then* $\vdash \phi$.

Note the modifications with respect to the class of dynamic axioms and rules in $\mathcal{L}^{\mathfrak{S}}_{DES}$.

Axioms $\mathrm{E}^+7$ and $\mathrm{E}^+8$ generalize axioms E7 and E8 respectively, by stating the properties also for the dynamic operators of composed actions. Hence, in this new system E7 and E8 are theorems.

Axioms $\mathrm{E}^+9$, $\mathrm{E}^+10$ and $\mathrm{E}^+11$ are exactly axioms E9, E10 and E11 respectively, without any change.

Observe that rules $\mathrm{E}^+_R2$ and $\mathrm{E}^+_R3$ have been also modified to include the composed action case, while rule $\mathrm{E}^+_R4$ remains unchanged.

With respect to the system $\mathcal{L}^{\mathfrak{S}}_{DES}$, three more axioms have been added, all of them focused on the dynamic operators of the composed actions.

Axiom $E^+12$ states that an agent in signature can perform an action only if it is active.

Axiom $E^+13$ states that if a system can be fully described by an agent in signature, then any action performed by our system can be described as performed by the agent.

Axiom $E^+14$ shows how the ontological dependencies of the agents in $\mathfrak{S}$ can be reflected in the syntax of dynamic operators for the composed actions.

## Epistemic axioms

**Axiom $E^+$ 15.** *If* $P \in \mathfrak{S}$ *then* $\vdash K_P\top \leftrightarrow c_P|\top$

**Axiom $E^+$ 16.** $\vdash K_Q\phi \wedge K_Q(\phi \rightarrow \psi) \rightarrow K_Q\psi$

**Axiom $E^+$ 17.** $\vdash K_Q\phi \rightarrow \phi$

**Axiom $E^+$ 18.** $\vdash K_Q\phi \rightarrow K_QK_Q\phi$.

**Axiom $E^+$ 19.** $\vdash \neg K_Q\phi \rightarrow (K_Q\top \rightarrow K_Q\neg K_Q\phi)$

**Axiom $E^+$ 20.** $\vdash K_Q\phi \leftrightarrow (K_Q\top \wedge K_0(K_Q\top \rightarrow \phi))$

**Axiom $E^+$ 21.** $\vdash K_0\phi \wedge \psi|\rho \rightarrow (K_0\phi \wedge \psi)|(K_0\phi \wedge \rho)$

**Axiom $E^+$ 22.** $\vdash K_0\phi \rightarrow [a]K_0\phi$

**Axiom $E^+$ 23.** $\vdash K_0\phi \rightarrow (K_Q\top \rightarrow K_QK_0\phi)$

### Epistemic rules

**Rule $\mathbf{E}_R^+$ 5.** *If* $\vdash \phi$ *then* $\vdash K_Q\top \rightarrow K_Q\phi$.

**Rule $\mathbf{E}_R^+$ 6.** *If* $\mathcal{M} \ni P$ *is a finite context and* $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow K_0\phi$ *then* $\vdash c_{\mathcal{M}} \rightarrow \phi$.

Note the only difference that exists between the class of epistemic axioms and rules in $\mathcal{L}_{DES}^{\mathfrak{S}}$ and in $\mathcal{L}_{DES}^{\mathfrak{S}^+}$: axiom $\mathrm{E}^+22$ had been modified to include, also, the case of composed actions. As before, we remark that this generalization entails E19 as theorem. Hence, all the epistemic theorems already proved for $\mathcal{L}_{DES}^{\mathfrak{S}}$ are still available.

## 6.6 The soundness of $\mathcal{L}_{DES}^{\mathfrak{S}^+}$ against process semantics

In this section we prove that the system $\mathcal{L}_{DES}^{\mathfrak{S}^+}$ is sound. Hence, anything that can be proved as theorem says something about a *real fact* in semantics.

Considering the previous observations concerning the similarity between some of the axioms of $\mathcal{L}_{DES}^{\mathfrak{S}^+}$, and axioms of $\mathcal{L}_{DS}$ or $\mathcal{L}_{DES}^{\mathfrak{S}}$, and because the three systems have been developed for the same semantics, we will not reconsider proving the soundness for the axioms for which these proofs go similarly with the proofs done in one of the previous chapters.

Due to the fact that the axioms of $\mathcal{L}_{DES}^{\mathfrak{S}}$ are just theorems for $\mathcal{L}_{DES}^{\mathfrak{S}^+}$, and because the semantics of $\mathcal{L}_{DES}^{\mathfrak{S}^+}$ just extend the semantics of $\mathcal{L}_{DES}^{\mathfrak{S}}$, all the theorems proved about characteristic formulas for processes and contexts remain true, and we will freely use them further, if necessary.

**Theorem 6.6.1** (Process-Soundness)**.** *The system* $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ *is sound with respect to process semantics.*

*Proof.* The soundness of $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ is sustained by the soundness of the axioms and rules, proved hereafter. □

The soundness of the spatial axioms and rules was proved in chapter 4, thus we take it granted.

**Lemma 6.6.2** (Soundness of axiom $\mathrm{E}^+7$)**.** $\models \langle a\rangle\phi|\psi \rightarrow \langle a\rangle(\phi|\psi)$.

*Proof.* If $\mathcal{M}, P \models \langle a\rangle\phi|\psi$, then $P \equiv R|S$, $\mathcal{M}, R \models \langle a\rangle\phi$ and $\mathcal{M}, S \models \psi$. So $\exists R \xrightarrow{a} R'$ and $\mathcal{M}, R' \models \phi$. So $\exists P \equiv R|S \xrightarrow{a} P' \equiv R'|S$ and $\mathcal{M}, P' \models \phi|\psi$. Hence $\mathcal{M}, P \models \langle a\rangle(\phi|\psi)$. □

**Lemma 6.6.3** (Soundness of axiom $\mathrm{E}^+8$)**.**

$$\models [a](\phi \rightarrow \psi) \rightarrow ([a]\phi \rightarrow [a]\psi)$$

*Proof.* Let $\mathcal{M}, P \models [a](\phi \rightarrow \psi)$ and $\mathcal{M}, P \models [a]\phi$.
If there is no $P'$ such that $P \xrightarrow{a} P'$, then $\mathcal{M}, P \models [a]\psi$.
Suppose that exists such $P'$. Then for any such $P'$ we have $\mathcal{M}, P' \models \phi \rightarrow \psi$ and $\mathcal{M}, P' \models \phi$. Hence $\mathcal{M}, P' \models \psi$, i.e. $\mathcal{M}, P \models [a]\psi$. □

The soundness for axioms $\mathrm{E}^+9$, $\mathrm{E}^+10$ and $\mathrm{E}^+11$ had been proved in chapter 5.

**Lemma 6.6.4** (Soundness of axiom $\mathrm{E}^+12$)**.**

$$\models \langle Q : \alpha \rangle \top \rightarrow c_Q | \top$$

*Proof.* Suppose that $\mathcal{M}, P \models \langle Q : \alpha \rangle \top$ then there exists a reduction $P \xrightarrow{Q:\alpha} P'$, hence $P \equiv Q|R$. Now, because $\mathcal{M}, Q \models c_Q$ and $\mathcal{M}, R \models \top$ we derive that $\mathcal{M}, P \models c_Q | \top$. □

**Lemma 6.6.5** (Soundness of axiom $\mathrm{E}^+13$)**.**

$$\textit{If } R \in \mathfrak{S} \textit{ then } \models c_R \rightarrow (\langle \alpha \rangle \phi \leftrightarrow \langle R : \alpha \rangle \phi)$$

*Proof.* If $\mathcal{M}, P \models c_R \wedge \langle \alpha \rangle \phi$ then $P \equiv R$ and $\mathcal{M}, R \models \langle \alpha \rangle \phi$. So, there is a transition $R \xrightarrow{\alpha} R'$ with $\mathcal{M}, R' \models \phi$. But $R \xrightarrow{\alpha} R'$ is equivalent with $R \xrightarrow{R:\alpha} R'$. Hence $\mathcal{M}, P \models \langle R : \alpha \rangle \phi$.
Reverse, if $\mathcal{M}, P \models c_R \wedge \langle R : \alpha \rangle \phi$ then $P \equiv R$ and $\mathcal{M}, R \models \langle R : \alpha \rangle \phi$. So, there is a transition $R \xrightarrow{R:\alpha} R'$ with $\mathcal{M}, R' \models \phi$. But $R \xrightarrow{R:\alpha} R'$ is equivalent with $R \xrightarrow{\alpha} R'$. Hence $\mathcal{M}, P \models \langle \alpha \rangle \phi$. □

**Lemma 6.6.6** (Soundness of axiom $\mathrm{E}^+14$)**.**

$$\models \langle P : \alpha \rangle \phi \wedge \langle P|Q : \alpha \rangle \top \rightarrow \langle P|Q : \alpha \rangle \phi$$

*Proof.* Suppose that $\mathcal{M}, R \models \langle P : \alpha \rangle \phi \wedge \langle P|Q : \alpha \rangle \top$.
Then $\mathcal{M}, R \models \langle P : \alpha \rangle \phi$ and $\mathcal{M}, R \models \langle P|Q : \alpha \rangle \top$.
But $\mathcal{M}, R \models \langle P : \alpha \rangle \phi$ means that it exists the reduction $R \xrightarrow{P:\alpha} R'$ and $\mathcal{M}, R' \models \phi$, i.e. $R \equiv P|S$, $P \xrightarrow{\alpha} P'$ and $R' \equiv P'|S$.
$\mathcal{M}, R \models \langle P|Q : \alpha \rangle \top$ means that $R \equiv P|Q|V$, i.e. $S \equiv Q|V$.
But $P \xrightarrow{P:\alpha} P'$ gives $P|Q \xrightarrow{P|Q:\alpha} P'|Q$, hence $R \xrightarrow{P|Q:\alpha} R'$ and $\mathcal{M}, R' \models \phi$, that means $\mathcal{M}, R \models \langle P|Q : \alpha \rangle \phi$. □

**Lemma 6.6.7** (Soundness of rule $\mathrm{E}_R^+2$)**.** *If* $\models \phi$ *then* $\models [a]\phi$.

*Proof.* Let $\mathcal{M}$ be a context and $P \in \mathcal{M}$ a process. If there is no $P'$ such that $P \xrightarrow{a} P'$, then $\mathcal{M}, P \models [a]\phi$. Suppose that exists such $P'$ (obviously $P' \in \mathcal{M}$). Then for any such $P'$ we have $\mathcal{M}, P' \models \phi$, due to the hypothesis $\models \phi$. Hence $\mathcal{M}, P \models [a]\phi$. □

**Lemma 6.6.8** (Soundness of rule $\mathrm{E}_R^+3$)**.**

*If* $\models \phi \to [a]\phi' \wedge \langle a\rangle\top$ *and* $\models \psi \to [a]\psi'$ *then* $\models \phi|\psi \to [a](\phi'|\psi \vee \phi|\psi')$

*Proof.* Suppose that $\mathcal{M}, P \models \phi|\psi$, then $P \equiv Q|R$, $\mathcal{M}, Q \models \phi$ and $\mathcal{M}, R \models \psi$. Because $\models \phi \to [a]\phi' \wedge \langle a\rangle\top$ and $\models \psi \to [a]\psi'$, we derive $\mathcal{M}, Q \models [a]\phi'$, $\mathcal{M}, Q \models \langle a\rangle\top$ (i.e. there exists a transition $Q \xrightarrow{a} Q'$) and $\mathcal{M}, R \models [a]\psi'$. We analyze some cases:

- if $R$ cannot perform a transition by $a$, then $Q|R \xrightarrow{a} Q'|R$ and the transitions of $P \equiv Q|R$ by $a$ have always this form. But $\mathcal{M}, Q \models [a]\phi'$, so for any such $Q'$ we have $\mathcal{M}, Q' \models \phi'$, thus $\mathcal{M}, Q'|R \models \phi'|\psi$, i.e. $\mathcal{M}, Q'|R \models (\phi'|\psi \vee \phi|\psi')$. Hence for any transition $P \xrightarrow{a} P'$ we have $\mathcal{M}, P' \models (\phi'|\psi \vee \phi|\psi')$. In conclusion, $\mathcal{M}, P \models [a](\phi'|\psi \vee \phi|\psi')$.

- if $R \xrightarrow{a} R'$ then $P \xrightarrow{a} P'$ has either the form $Q|R \xrightarrow{a} Q'|R$ or $Q|R \xrightarrow{a} Q|R'$. But $\mathcal{M}, Q'|R \models \phi'|\psi$, hence $\mathcal{M}, Q'|R \models (\phi'|\psi \vee \phi|\psi')$ and $\mathcal{M}, Q|R' \models \phi|\psi'$, hence $\mathcal{M}, Q|R' \models (\phi'|\psi \vee \phi|\psi')$. Thus, for any transition $P \xrightarrow{a} P'$ we have $\mathcal{M}, P' \models (\phi'|\psi \vee \phi|\psi')$, i.e. $\mathcal{M}, P \models [a](\phi'|\psi \vee \phi|\psi')$.

So, in any case $\mathcal{M}, P \models [a](\phi'|\psi \vee \phi|\psi')$, that concludes the proof of the theorem. □

The soundness of rule $\mathrm{E}_R^+4$ goes similarly with the soundness of rule $\mathrm{E}_R4$ proved in chapter 5.

The soundness of all the epistemic axioms goes similarly with the proofs done in chapter 5. The only case we will consider is of axiom $\mathrm{E}^+22$ that involves the new dynamic operators.

**Lemma 6.6.9** (Soundness of axiom $\mathrm{E}^+22$)**.** $\models K_0\phi \rightarrow [a]K_0\phi$

*Proof.* Suppose that $\mathcal{M}, P \models K_0\phi$.
Then for any $R \in \mathcal{M}$ we have $\mathcal{M}, R \models \phi$.
If $P$ cannot perform a transition by $a$, we have $\mathcal{M}, P \models [a]K_0\phi$.
If $P$ can perform such transitions, then for any $P \xrightarrow{a} P'$ we have $\mathcal{M}, P' \models K_0\phi$, because for any $R \in \mathcal{M}$ we have $\mathcal{M}, R \models \phi$. Thus $\mathcal{M}, P \models [a]K_0\phi$. □

In this way we conclude that our system $\mathcal{L}_{DES}^{\mathfrak{S}^+}$ is sound with respect to process semantics, so everything that can be proved from true premisses is true.

## 6.7 Theorems of $\mathcal{L}_{DES}^{\mathfrak{S}^+}$

In this section we will prove some theorems in the system $\mathcal{L}_{DES}^{\mathfrak{S}}$. Some of them are theorems already proved for the case of basic actions and we just show that they are available also for the composed case; others are new concerning the behavior of the dynamics operators for the composed action in the context of the other operators.

We begin by proving that for an epistemic agent to be active is equivalent with possibly performing an action.

**Theorem 6.7.1.** *If* $Q \in \mathfrak{S}$ *and* $Q \xrightarrow{\alpha} Q'$ *then* $\vdash \langle Q : \alpha \rangle \top \leftrightarrow c_Q|\top$.

*Proof.* From axiom $\mathrm{E}^+13$ we have $\vdash c_Q \rightarrow (\langle \alpha \rangle \top \leftrightarrow \langle Q : \alpha \rangle \top)$. But because $Q \xrightarrow{\alpha} Q'$ we derive that $Q \equiv \alpha.Q_1|Q_2$, thus $\vdash c_Q \rightarrow \langle \alpha \rangle \top$. Hence $\vdash c_Q \rightarrow \langle Q : \alpha \rangle \top$. Rule $\mathrm{E}^+_R1$ entails $\vdash c_Q|\top \rightarrow \langle Q : \alpha \rangle \top|\top$, and using axiom $\mathrm{E}^+7$ we derive further $\vdash c_Q|\top \rightarrow \langle Q : \alpha \rangle \top$.
Further we combine this result with axiom $\mathrm{E}^+12$. □

**Theorem 6.7.2.** $\vdash \langle Q : \alpha \rangle \top \rightarrow K_Q \top$

*Proof.* Theorem 6.7.1 and axiom $\mathrm{E}^+15$ give the result. □

The next theorem states the monotonicity for the dynamic operator, thus generalizing theorem 4.6.9.

**Theorem 6.7.3** (Monotonicity)**.** *If* $\vdash \phi \rightarrow \psi$ *then* $\vdash \langle a \rangle \phi \rightarrow \langle a \rangle \psi$.

*Proof.* $\vdash \phi \rightarrow \psi$ implies $\vdash \neg\psi \rightarrow \neg\phi$. Using rule $\mathrm{E}^+_R2$ we obtain $\vdash [a](\neg\psi \rightarrow \neg\phi)$ and axiom $\mathrm{E}^+8$ gives $\vdash [a]\neg\psi \rightarrow [a]\neg\phi$. This is equivalent with $\vdash \neg\langle a \rangle \psi \rightarrow \neg\langle a \rangle \phi$, i.e. $\vdash \langle a \rangle \phi \rightarrow \langle a \rangle \psi$. □

Further we consider the generalization of theorem 4.6.10.

**Theorem 6.7.4.** *If* $\vdash \phi \rightarrow \psi$ *then* $\vdash [a]\neg\psi \rightarrow [a]\neg\phi$.

*Proof.* If $\vdash \phi \rightarrow \psi$ then, by theorem 6.7.3, $\vdash \langle a\rangle\phi \rightarrow \langle a\rangle\psi$, hence $\vdash \neg\langle a\rangle\psi \rightarrow \neg\langle a\rangle\phi$, that gives $\vdash [a]\neg\psi \rightarrow [a]\neg\phi$. □

Also the generalization of theorem 4.6.11 is available, projecting, in the syntax, the property that $P$ cannot perform an action $a$.

**Theorem 6.7.5.** *If $P$ cannot perform a transition by $a$ then $\vdash c_P \rightarrow [a]\bot$.*

*Proof.* **The case** $a = \alpha$ was stated and proved by axiom 4.6.11.

**The case** $a = (R : \alpha)$ then $R$ is not an active subprocess of $P$, hence for any decomposition $P \equiv Q|S$, we have $Q \not\equiv R$ or equivalently, using theorem 4.7.3, $\vdash c_Q \rightarrow \neg c_R$, or using theorem 4.6.8, $\vdash c_P \rightarrow \neg(c_R|\top)$. But from theorem 6.7.1 $\vdash c_R|\top \leftrightarrow \langle R : \alpha\rangle\top$, hence $\vdash c_P \rightarrow \neg\langle R : \alpha\rangle\top$, i.e $\vdash c_P \rightarrow [a]\bot$. □

The next theorem confirms the intuition that the formulas $c_P$, in their interrelations, mimic the transitions of the processes (the dynamic operator mimics the transition labeled by the action it has as index). This generalizes theorem 4.6.12.

**Theorem 6.7.6.** $\vdash c_P \rightarrow [a]\bigvee\{c_Q \mid P \xrightarrow{a} Q\}$

*Proof.* We prove it by induction on the structure of $P$.
**The case** $a = \alpha$ was proved by theorem 4.6.12.
**The case** $a = (R : \alpha)$**:** we distinguish the subcases:

- **if $R$ is not an active subprocess of $P$**, then $P$ cannot perform the transition by $a$, hence, using theorem 6.7.5, we obtain $\vdash c_P \rightarrow [a]\bot$ that implies $\vdash c_P \rightarrow [\alpha]\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\}$.

- **if** $P \equiv R$, then using the theorem for basic actions, already proved, we obtain

$$\vdash c_R \to [\alpha] \bigvee \{c_Q \mid R \xrightarrow{\alpha} Q\}$$

But axiom $\mathrm{E}^{+}13$ gives

$$\vdash c_R \to ([\alpha] \bigvee \{c_Q \mid R \xrightarrow{\alpha} Q\} \leftrightarrow [R:\alpha] \bigvee \{c_Q \mid R \xrightarrow{\alpha} Q\})$$

and because $R \xrightarrow{\alpha} Q$ iff $R \xrightarrow{R:\alpha} Q$, we derive

$$\vdash c_R \to [R:\alpha] \bigvee \{c_Q \mid R \xrightarrow{R:\alpha} Q\}$$

- **if** $P \equiv R|S$ **and** $S \not\equiv 0$ then we apply the inductive hypothesis to $R$ and $S$ respectively, that gives

$$\vdash c_R \to [a] \bigvee \{c_{Q'} \mid R \xrightarrow{a} Q'\}$$

and

$$\vdash c_S \to [a] \bigvee \{c_{Q''} \mid S \xrightarrow{a} Q''\}$$

As $\vdash c_R \to \langle a \rangle \top$, we can apply rule $\mathrm{E}^{+}_{R}3$ and obtain

$$\vdash c_P \to [a](\bigvee \{c_{Q'} \mid R \xrightarrow{a} Q'\} | c_S \vee c_R | \bigvee \{c_{Q''} \mid S \xrightarrow{a} Q''\})$$

which is equivalent with $\vdash c_P \to [a] \bigvee \{c_Q \mid P \xrightarrow{a} Q\}$.

□

The dynamic operator $[a]$ is transparent with respect to $\widetilde{K}_0$, similarly to the situation for $[\alpha]$, stated in theorem 5.8.12.

**Theorem 6.7.7.** $\vdash \widetilde{K}_0\phi \to [a]\widetilde{K}_0\phi$

*Proof.* Axiom $\mathrm{E}^+22$ instantiated with $\phi = \widetilde{K}_0\phi$ gives

$$\vdash K_0\widetilde{K}_0\phi \rightarrow [a]K_0\widetilde{K}_0\phi$$

Further, using theorem 5.8.10, we obtain the wanted result. □

Now we prove that also the theorems involving the context-sensitive properties, proved for the dynamic operators associated to basic actions, can be stated for the composed actions cases. Thus the next result generalizes theorem 5.8.17.

**Theorem 6.7.8.** *If $\mathcal{M}$ is a finite context then $\vdash c_{\mathcal{M}} \rightarrow [a]c_{\mathcal{M}}$*

*Proof.* Observe that, by applying axiom $\mathrm{E}^+22$, we obtain

$$\vdash K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3 \rightarrow (\widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \wedge [a]K_0\theta_1$$

If, further, we apply theorem 6.7.7 once, we obtain

$$\vdash (\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2) \wedge [a]K_0\theta_1 \rightarrow \widetilde{K}_0\theta_3 \wedge [a]\widetilde{K}_0\theta_2 \wedge [a]K_0\theta_1, \ i.e.$$
$$\vdash (\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2) \wedge [a]K_0\theta_1 \rightarrow \widetilde{K}_0\theta_3 \wedge [a](\widetilde{K}_0\theta_2 \wedge K_0\theta_1)$$

Hence

$$\vdash (K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \rightarrow \widetilde{K}_0\theta_3 \wedge [a](\widetilde{K}_0\theta_2 \wedge K_0\theta_1)$$

If we apply again theorem 6.7.7 we obtain

$$\vdash \widetilde{K}_0\theta_3 \wedge [a](\widetilde{K}_0\theta_2 \wedge K_0\theta_1) \rightarrow [a](\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2 \wedge K_0\theta_1)$$

hence

$$\vdash (K_0\theta_1 \wedge \widetilde{K}_0\theta_2 \wedge \widetilde{K}_0\theta_3) \rightarrow [a](\widetilde{K}_0\theta_3 \wedge \widetilde{K}_0\theta_2 \wedge K_0\theta_1)$$

As $c_{\mathcal{M}} = K_0(\bigvee_{Q\in\mathcal{M}} c_Q)\wedge(\bigwedge_{Q\in\mathcal{M}} \widetilde{K}_0 c_Q)$, we can use the same idea, applying theorem 6.7.7 once for each process in $\mathcal{M}$ (being finite) and we obtain

$$\vdash c_{\mathcal{M}} \rightarrow [a]c_{\mathcal{M}}$$

□

We generalize, now, theorem 5.8.21.

**Theorem 6.7.9.** *If* $\vdash c_{\mathcal{M}} \to \phi$ *then* $\vdash c_{\mathcal{M}} \to [a]\phi$.

*Proof.* If we apply rule $\mathrm{E}^{+}_{R}2$ to $\vdash c_{\mathcal{M}} \to \phi$ we obtain $\vdash [a](c_{\mathcal{M}} \to \phi)$. But axiom $\mathrm{E}^{+}8$ gives $\vdash [a](c_{\mathcal{M}} \to \phi) \to ([a]c_{\mathcal{M}} \to [a]\phi)$, hence $\vdash [a]c_{\mathcal{M}} \to [a]\phi$. theorem 6.7.8 proves that $\vdash c_{\mathcal{M}} \to [a]c_{\mathcal{M}}$ which gives further $\vdash c_{\mathcal{M}} \to [a]\phi$. □

The monotonicity of the dynamic operator in a given context holds.

**Theorem 6.7.10.** *If* $\vdash c_{\mathcal{M}} \to (\phi \to \psi)$ *then* $\vdash c_{\mathcal{M}} \to (\langle a\rangle\phi \to \langle a\rangle\psi)$.

*Proof.* $\vdash c_{\mathcal{M}} \to (\phi \to \psi)$ implies $\vdash c_{\mathcal{M}} \to (\neg\psi \to \neg\phi)$ where, applying theorem 6.7.9, we obtain $\vdash c_{\mathcal{M}} \to [a](\neg\psi \to \neg\phi)$. But axiom $\mathrm{E}^{+}8$ gives $\vdash [a](\neg\psi \to \neg\phi) \to ([a]\neg\psi \to [a]\neg\phi)$. Hence $\vdash c_{\mathcal{M}} \to ([a]\neg\psi \to [a]\neg\phi)$, i.e. $\vdash c_{\mathcal{M}} \to (\neg\langle a\rangle\psi \to \neg\langle a\rangle\phi)$. Concluding, $\vdash c_{\mathcal{M}} \to (\langle a\rangle\phi \to \langle a\rangle\psi)$. □

The generalization of theorem 5.8.27 can be proved as well.

**Theorem 6.7.11.**

$$\textit{If } \vdash c_{\mathcal{M}} \to (\bigvee\{c_Q \mid P \xrightarrow{a} Q\} \to \phi) \textit{ then } \vdash c_{\mathcal{M}} \to (c_P \to [a]\phi)$$

*Proof.* If $\vdash c_{\mathcal{M}} \to (\bigvee\{c_Q \mid P \xrightarrow{a} Q\} \to \phi)$ then theorem 6.7.9 gives $\vdash c_{\mathcal{M}} \to [a](\bigvee\{c_Q \mid P \xrightarrow{a} Q\} \to \phi)$ and further axiom $\mathrm{E}^{+}8$ gives

$$\vdash c_{\mathcal{M}} \to ([a]\bigvee\{c_Q \mid P \xrightarrow{a} Q\} \to [a]\phi)$$

But theorem 6.7.6 gives

$$\vdash c_P \to [a] \bigvee \{c_Q \mid P \xrightarrow{a} Q\}$$

hence $\vdash c_{\mathcal{M}} \wedge c_P \to [a]\phi$, i.e. $\vdash c_{\mathcal{M}} \to (c_P \to [a]\phi)$. □

## 6.8 Completeness of $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ against process semantics

We are ready now to prove the completeness for our system $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ with respect to process semantics, and in this way to prove that also the most expressive logic we considered can specify correctly and completely *the soup of processes*.

As for the other systems, we begin by proving the lemma that provides a syntactic characterization of satisfiability relation. Observe that lemma 6.8.1 states exactly what the similar lemma 5.9.1 stated for the system $\mathcal{L}^{\mathfrak{S}}_{DES}$. Hence we have a generalization of lemma 5.9.1 to the case that includes the dynamic operators for composed actions.

**Lemma 6.8.1.** *If $\mathcal{M}$ is a finite context then $\mathcal{M}, P \models \phi$ iff $\vdash c_{\mathcal{M}} \wedge c_P \to \phi$.*

*Proof.* ($\Longrightarrow$) We prove it by induction on the syntactical structure of $\phi$. The only cases we analyze are those involving the dynamic operator for composed actions, because the proof for the rest of the cases is identical to that in lemma 5.9.1

- **The case $\phi = \langle Q : \alpha \rangle \psi$:** $\mathcal{M}, P \models \langle Q : \alpha \rangle \psi$ ensures us that $\alpha$ is active in $Q \in \mathfrak{S}$.

– **the subcase** $\psi = \top$**:** $\mathcal{M}, P \models \langle Q : \alpha\rangle\top$ gives $P \equiv Q|R$, hence $c_P = c_Q|c_R$. But $\vdash c_R \to \top$ and, using rule $\mathrm{E}^+_R1$, $\vdash c_Q|c_R \to c_Q|\top$. Now using theorem 6.7.1 we obtain

$$\vdash c_P \to \langle Q : \alpha\rangle\top, \ \textit{hence} \ \vdash c_{\mathcal{M}} \wedge c_P \to \langle Q : \alpha\rangle\top$$

– **the subcase** $\psi \neq \top$**:** $\mathcal{M}, P \models \langle Q : \alpha\rangle\psi$ implies $P \equiv Q|R$, exists $Q' \in \mathcal{M}$ such that $Q \xrightarrow{\alpha} Q'$ and $\mathcal{M}, Q'|R \models \psi$. Using the inductive hypothesis we obtain

$$\vdash c_{\mathcal{M}} \wedge c_{Q'|R} \to \psi$$

But $Q \xrightarrow{\alpha} Q'$ means that $Q \equiv \alpha.Q''|S$ and $Q' \equiv Q''|S$. Then $\vdash c_{\mathcal{M}} \wedge c_{Q'|R} \to \psi$ means $\vdash c_{\mathcal{M}} \wedge c_{Q''}|c_S|c_R \to \psi$, i.e.

$$\vdash c_{\mathcal{M}} \to (c_{Q''}|c_S|c_R \to \psi)$$

Using theorem 6.7.10 we obtain

$$\vdash c_{\mathcal{M}} \to (\langle\alpha.Q'' : \alpha\rangle(c_{Q''}|c_S|c_R) \to \langle\alpha.Q'' : \alpha\rangle\psi)$$

while axiom $\mathrm{E}^+7$ gives

$$\vdash \langle\alpha.Q'' : \alpha\rangle c_{Q''}|c_S|c_R \to \langle\alpha.Q'' : \alpha\rangle(c_{Q''}|c_S|c_R),$$

hence

$$\vdash c_{\mathcal{M}} \to (\langle\alpha.Q'' : \alpha\rangle c_{Q''}|c_S|c_R \to \langle\alpha.Q'' : \alpha\rangle\psi)$$

and because $\vdash c_P \to \langle\alpha.Q'' : \alpha\rangle c_{Q''}|c_S|c_R$, due to axiom $\mathrm{E}^+13$, we derive further

$$\vdash c_{\mathcal{M}} \to (c_P \to \langle\alpha.Q'' : \alpha\rangle\psi) \tag{6.8.1}$$

But $\mathcal{M}, P \models \langle Q : \alpha\rangle\psi$ gives $\mathcal{M}, P \models \langle Q : \alpha\rangle\top$ (because $\vdash \psi \to \top$ and using theorem 6.7.3 $\vdash \langle Q : \alpha\rangle\psi \to \langle Q : \alpha\rangle\top$). But, from the

previous case, $\mathcal{M}, P \models \langle Q : \alpha\rangle\top$ is equivalent with $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \langle Q : \alpha\rangle\top$. Hence

$$\vdash c_{\mathcal{M}} \rightarrow (c_P \rightarrow \langle \alpha.Q''|S : \alpha\rangle\top) \tag{6.8.2}$$

Axiom $\mathrm{E}^+14$ gives

$$\vdash \langle \alpha.Q'' : \alpha\rangle\psi \wedge \langle \alpha.Q''|S : \alpha\rangle\top \rightarrow \langle \alpha.Q''|S : \alpha\rangle\psi$$

and as (6.8.1) and (6.8.2) give

$$\vdash c_{\mathcal{M}} \rightarrow (c_P \rightarrow \langle \alpha.Q'' : \alpha\rangle\psi \wedge \langle \alpha.Q''|S : \alpha\rangle\top)$$

we obtain further

$$\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \langle Q : \alpha\rangle\psi$$

- **The case $\phi = \neg\langle a\rangle\psi_1$:** $\mathcal{M}, P \models \neg\langle a\rangle\psi_1$ is equivalent with $\mathcal{M}, P \models [a]\neg\psi_1$.

  If there is a process $Q \in \mathcal{M}$ such that $P \xrightarrow{a} Q$, then for any $Q \in \mathcal{M}$ such that $P \xrightarrow{a} Q$ we have $\mathcal{M}, Q \models \neg\psi_1$. Using the inductive hypothesis we obtain that for any $Q \in \mathcal{M}$ such that $P \xrightarrow{a} Q$ we have $\vdash c_{\mathcal{M}} \wedge c_Q \rightarrow \neg\psi_1$, i.e.

  $$\vdash c_{\mathcal{M}} \wedge \bigvee\{c_Q \mid P \xrightarrow{a} Q\} \rightarrow \neg\psi_1$$

  or equivalently

  $$\vdash c_{\mathcal{M}} \rightarrow (\bigvee\{c_Q \mid P \xrightarrow{a} Q\} \rightarrow \neg\psi_1)$$

  Using theorem 6.7.11, we obtain $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow [a]\neg\psi_1$.
  If there is no process $Q \in \mathcal{M}$ such that $P \xrightarrow{a} Q$ then theorem 6.7.6 gives $\vdash c_P \rightarrow [a]\bot$. But $\vdash \psi_1 \rightarrow \top$, hence $\vdash [a]\bot \rightarrow [a]\neg\psi_1$. So, also in this case we have $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow [a]\neg\psi_1$.

($\Longleftarrow$) Let $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \phi$. Suppose that $\mathcal{M}, P \not\models \phi$. Then $\mathcal{M}, P \models \neg\phi$. Using the reversed implication we obtain $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \neg\phi$, hence $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \bot$. But from corollary 5.4.2 we have $\mathcal{M}, P \models c_{\mathcal{M}} \wedge c_P$ which, using the soundness, gives $\mathcal{M}, P \models \bot$ impossible! Hence $\mathcal{M}, P \models \phi$. $\square$

We recall the definition of provability, consistency, satisfiability and validity adapted to the system $\mathcal{L}^{\mathfrak{S}^+}_{DES}$.

**Definition 6.8.1** (Provability and consistency)**.** We say that a formula $\phi \in \mathcal{F}^{\mathfrak{S}^+}_{DES}$ is *provable in* $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ (or $\mathcal{L}^{\mathfrak{S}^+}_{DES}$*-provable* for short), if $\phi$ can be derived, as a theorem, using the axioms and the rules of $\mathcal{L}^{\mathfrak{S}^+}_{DES}$. We denote this by $\vdash \phi$.
We say that a formula $\phi \in \mathcal{F}^{\mathfrak{S}^+}_{DES}$ is *consistent in* $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ (or $\mathcal{L}^{\mathfrak{S}^+}_{DES}$*-consistent* for short) if $\neg\phi$ is not $\mathcal{L}^{\mathfrak{S}^+}_{DES}$-provable.

**Definition 6.8.2** (Satisfiability and validity)**.** We call a formula $\phi \in \mathcal{F}^{\mathfrak{S}^+}_{DES}$ *satisfiable* if there exists a context $\mathcal{M}$ and a process $P \in \mathcal{M}$ such that $\mathcal{M}, P \models \phi$.
We call a formula $\phi \in \mathcal{F}^{\mathfrak{S}^+}_{DES}$ *validity* if for any context $\mathcal{M}$ and any process $P \in \mathcal{M}$ we have $\mathcal{M}, P \models \phi$. In such a situation we write $\models \phi$.
Given a context $\mathcal{M}$, we denote by $\mathcal{M} \models \phi$ the situation when for any $P \in \mathcal{M}$ we have $\mathcal{M}, P \models \phi$.

*Remark* 6.8.1. $\phi$ is satisfiable iff $\neg\phi$ is not a validity, and vice versa, $\phi$ is a validity iff $\neg\phi$ is not satisfiable.

Further we prove that the consistency implies satisfiability. This result will be used to prove the completeness.

**Lemma 6.8.2.** *If $\phi$ is $\mathcal{L}_{DES}^{\mathfrak{S}^+}$-consistent then exists a context $\mathcal{M}$ and a process $P \in \mathcal{M}$ such that $\mathcal{M}, P \models \phi$.*

*Proof.* Suppose that for any context $\mathcal{M}$ and any process $P \in \mathcal{M}$ we do not have $\mathcal{M}, P \models \phi$, i.e. we have $\mathcal{M}, P \models \neg\phi$.
Hence, for any finite context $\mathcal{M}$ and any process $P \in \mathcal{M}$ we have
$\mathcal{M}, P \models \neg\phi$.
Using lemma 6.8.1, we obtain $\vdash c_{\mathcal{M}} \wedge c_P \rightarrow \neg\phi$ for any finite context $\mathcal{M} \ni P$. Hence $\vdash c_{\mathcal{M}} \wedge \bigvee_{P\in\mathcal{M}} c_P \rightarrow \neg\phi$. But $\vdash c_{\mathcal{M}} \rightarrow \bigvee_{P\in\mathcal{M}} c_P$ which, combined with the previous result, implies $\vdash c_{\mathcal{M}} \rightarrow \neg\phi$.
Thus for each finite context $\mathcal{M}$ we have $\vdash c_{\mathcal{M}} \rightarrow \neg\phi$. But then for each context $\overline{M}$ with $[\![M]\!] \leq (\!|\neg\phi|\!)$ we have $\vdash c_{\overline{M}} \rightarrow \neg\phi$, as these contexts are finite. Each of these contexts belongs to $\mathfrak{M}_{(\!|\neg\phi|\!)}$ which is also finite. Hence, we can infer further

$$\vdash \bigvee_{[\![M]\!]\leq(\!|\neg\phi|\!)} c_{\overline{M}} \rightarrow \neg\phi$$

Now, applying rule $\mathrm{E}_R^+4$, we obtain $\vdash \neg\phi$. This contradicts with the hypothesis of consistency of $\phi$.

Hence, it exists a context $\mathcal{M}$ and a process $P \in \mathcal{M}$ such that

$$\mathcal{M}, P \models \phi$$

□

**Theorem 6.8.3** (Completeness)**.** *The $\mathcal{L}_{DES}^{\mathfrak{S}^+}$ system is complete with respect to process semantics.*

*Proof.* Suppose that $\phi$ is a valid formula, but $\phi$ is not provable in the system $\mathcal{L}^{\mathfrak{S}^+}_{DES}$. Then neither is $\neg\neg\phi$, so, by definition, $\neg\phi$ is $\mathcal{L}^{\mathfrak{S}^+}_{DES}$-consistent. It follows, from lemma 6.8.2, that $\neg\phi$ is satisfiable with respect to process semantics, contradicting the validity of $\phi$. □

## 6.9 Concluding remarks

In this chapter we extended the system $\mathcal{L}^{\mathfrak{S}}_{DES}$ by adding, in the syntax, dynamic operators for composed actions. The composed actions are actions marked by their authors - the epistemic agents. Thus the system is more expressive allowing properties such as *"the system performed the action $\alpha$ and the one responsible of it is the agent $P$"* to be specified. This kind of specification may be used in causality issues and may provide solutions in problems such as debugging, performance analysis, etc.

For our system we developed a Hilbert-style axiomatic that depicts the joint behavior of the classical, spatial, dynamic and epistemic operators. The similarities with the classical epistemic logic are still noticed. The system was proved to be sound and complete with respect to the semantics.

By proving the finite model theory we show the decidability of the system in problems of satisfiability, validity and model checking against the process semantics.

# Chapter 7

# Future works

We envisage some different directions for exploring further the idea of a dynamic-epistemic approach to concurrency. In this chapter we will present, briefly, the most interesting.

## 7.1 Extending CCS semantics for an infinite class of actions

As presented before, the fragment of CCS we considered was introduced for a finite class of actions $\mathbb{A}$ following the intuition that, in any application, it is improbable to face the situation that the external observer *knows* an infinite number of actions possible in the system. This situation will be immediately related to the question of the possibility of any agent whatsoever having an actual infinite knowledge, as modeling supposes an actual and not a potential knowledge. Indeed you need first to know the reality and then you might consider modeling it.

Still, someone might want to implement a counter in the system or any other recursive behavior that asks, in combination, for example, with the new name operator an unbounded set of actions.

Our latest research has revealed some results in this direction that we will present further.

The intuition is that also for a denumerable set of actions $\mathbb{A}$ our logics remain decidable. The finite model property can be proved for our systems, but in a stronger version.

Informally, if $\mathcal{M}, P \models \phi$, and if in $\mathcal{M}$ (and in $P$) some actions appear that are not present in $\phi$, then replacing these actions with any other that did not appear in the syntax of $\phi$ should not change the satisfaction relation. Indeed, if in $\mathcal{M}$ and $P$ we have the action $\alpha$ that is not in $\phi$ and if $\beta$ is another action that didn't appear in the syntax of $\phi$, then what $\phi$ states did not involve $\alpha$ or $\beta$, and it should be stated also about $\mathcal{M}^\sigma$ and $P^\sigma$ (where by $\sigma$ we denote the substitution $\sigma \stackrel{def}{=} \{\alpha \Leftarrow \beta\}$ and by $\mathcal{M}^\sigma$ and $P^\sigma$ respectively the result of substituting $\alpha$ by $\beta$ in all its occurrences in the syntaxes of the processes in $\mathcal{M}$, and in $P$ respectively).

Because $\phi$ contains, in its syntax, a finite number of actions, and because on $\mathbb{A}$ we might assume a lexicographic order, we can just take the successor of the biggest action in $\phi$, let's denote it by $\alpha_\phi$, and substitute all the actions in $\mathcal{M}$ that do not appear in $\phi$ with $\alpha_\phi$. In this way we will obtain the set $\mathcal{M}^\sigma$ that contains a finite number of actions and the relation $\mathcal{M}, P \models \phi$ is equivalent to $\mathcal{M}^\sigma, P^\sigma \models \phi$. Thus we projected the problem into a decidable one.

Now we will formalize these intuitions.

**Definition 7.1.1** (The set of actions of a process)**.** We define the set of actions of $P$, $Act(P) \subset \mathbb{A}$, inductively by:

- if $P \equiv 0$ then $Act(P) \stackrel{def}{=} \emptyset$
- if $P \equiv \alpha.Q$ then $Act(P) \stackrel{def}{=} \{\alpha\} \cup Act(Q)$
- if $P \equiv Q|R$ then $ActP \stackrel{def}{=} Act(Q) \cup Act(R)$

For a set $M \subset \mathfrak{P}$ of processes we define $Act(M) \stackrel{def}{=} \bigcup_{P \in M} Act(P)$.

**Definition 7.1.2** (The set of actions of a formula)**.** We define the set of actions of a formula $\phi$, $act(\phi) \subset \mathbb{A}$, inductively by:

- $act(0) \stackrel{def}{=} \emptyset$
- $act(\top) \stackrel{def}{=} \emptyset$
- $act(\neg\phi) = act(\phi)$
- $act(\phi \wedge \psi) = act(\phi|\psi) \stackrel{def}{=} act(\phi) \cup act(\psi)$
- $act(\langle\alpha\rangle\phi) \stackrel{def}{=} \{\alpha\} \cup act(\phi)$
- $act(\langle R : \alpha\rangle\phi) = act(K_R\phi) \stackrel{def}{=} Act(R) \cup act(\phi)$

**Definition 7.1.3** (Action substitution)**.** We call *action substitution* any function $\sigma : \mathbb{A} \longrightarrow \mathbb{A}$. We extend it further, syntactically, from actions to processes, $\sigma : \mathfrak{P} \longrightarrow \mathfrak{P}$, by

$$\sigma(P) = \begin{cases} 0 & \text{if } P \equiv 0 \\ \sigma(Q)|\sigma(R) & \text{if } P \equiv Q|R \\ \sigma(\gamma).\sigma(R) & \text{if } P \equiv \gamma.R \end{cases}$$

We extend $\sigma$ for sets of processes $M \subset \mathfrak{P}$ by $\sigma(M) \stackrel{def}{=} \{\sigma(P) \mid P \in M\}$. We will use the more convenient notation $\mathcal{M}^\sigma$ and $P^\sigma$ for denoting $\sigma(\mathcal{M})$ and $\sigma(P)$ respectively.

**Lemma 7.1.1.** *If $\alpha, \beta \notin act(\phi)$ and $\mathcal{M}, P \models \phi$ then $\mathcal{M}^\sigma, P^\sigma \models \phi$, where we denoted by $\sigma = \{\alpha \Leftarrow \beta\}$.*

***A possible proof:*** We prove, simultaneously, by induction on $\phi$ that

1. if $\mathcal{M}, P \models \phi$ then $\sigma(\mathcal{M}), \sigma(P) \models \phi$
2. if $\mathcal{M}, P \not\models \phi$ then $\sigma(\mathcal{M}), \sigma(P) \not\models \phi$

**The case $\phi = 0$:**

1. $\mathcal{M}, P \models 0$ iff $P \equiv 0$. Then $\sigma(P) \equiv 0$ and $\sigma(\mathcal{M}), \sigma(0) \models 0$ q.e.d.
2. $\mathcal{M}, P \not\models 0$ iff $P \not\equiv 0$, iff $\sigma(P) \not\equiv 0$. Hence $\sigma(\mathcal{M}), \sigma(P) \not\models 0$.

**The case $\phi = \top$:**

1. $\mathcal{M}, P \models \top$ implies $\sigma(\mathcal{M}), \sigma(P) \models \top$, because this is happening for any context and process.
2. $\mathcal{M}, P \not\models \top$ is an impossible case.

**The case $\phi = \psi_1 \wedge \psi_2$:**

1. $\mathcal{M}, P \models \psi_1 \wedge \psi_2$ implies that $\mathcal{M}, P \models \psi_1$ and $\mathcal{M}, P \models \psi_2$. Because $\alpha, \beta \notin act(\phi)$ we derive that $\alpha, \beta \notin act(\psi_1)$ and $\alpha, \beta \notin act(\psi_2)$. Further, applying the inductive hypothesis we obtain $\sigma(\mathcal{M}), \sigma(P) \models \psi_1$ and $\sigma(\mathcal{M}), \sigma(P) \models \psi_2$ that implies $\sigma(\mathcal{M}), \sigma(P) \models \psi_1 \wedge \psi_2$.
2. $\mathcal{M}, P \not\models \psi_1 \wedge \psi_2$ implies that $\mathcal{M}, P \not\models \psi_1$ or $\mathcal{M}, P \not\models \psi_2$. But, as argued before, $\alpha, \beta \notin act(\psi_1)$ and $\alpha, \beta \notin act(\psi_2)$, hence we can apply the inductive hypothesis that entails $\sigma(\mathcal{M}), \sigma(P) \not\models \psi_1$ or $\sigma(\mathcal{M}), \sigma(P) \not\models \psi_2$. Thus $\sigma(\mathcal{M}), \sigma(P) \not\models \psi_1 \wedge \psi_2$.

**The case $\phi = \neg\psi$:**

1. $\mathcal{M}, P \models \neg\psi$ is equivalent with $\mathcal{M}, P \not\models \psi$ and because $\alpha, \beta \notin act(\phi)$ guarantees that $\alpha, \beta \notin act(\psi)$, we ca apply the inductive hypothesis and we obtain $\sigma(\mathcal{M}), \sigma(P) \not\models \psi$ which is equivalent with $\sigma(\mathcal{M}), \sigma(P) \models \neg\psi$.

2. $\mathcal{M}, P \not\models \neg\psi$ is equivalent with $\mathcal{M}, P \models \psi$ and applying the inductive hypothesis, $\sigma(\mathcal{M}), \sigma(P) \models \psi$, i.e. $\sigma(\mathcal{M}), \sigma(P) \not\models \neg\psi$.

**The case $\phi = \psi_1|\psi_2$:**

1. $\mathcal{M}, P \models \psi_1|\psi_2$ implies that $P \equiv Q|R$, $\mathcal{M}, Q \models \psi_1$ and $\mathcal{M}, R \models \psi_2$. As $\alpha, \beta \notin act(\phi)$ we have $\alpha, \beta \notin act(\psi_1)$ and $\alpha, \beta \notin act(\psi_2)$. Then we can apply the inductive hypothesis and obtain $\sigma(\mathcal{M}), \sigma(Q) \models \psi_1$ and $\sigma(\mathcal{M}), \sigma(R) \models \psi_2$. But $\sigma(P) \equiv \sigma(Q)|\sigma(R)$, hence $\sigma(\mathcal{M}), \sigma(P) \models \phi$.

2. $\mathcal{M}, P \not\models \psi_1|\psi_2$ implies that for any decomposition $P \equiv Q|R$ we have either $\mathcal{M}, Q \not\models \psi_1$ or $\mathcal{M}, R \not\models \psi_2$. But, as before, from $\alpha, \beta \notin act(\phi)$ guarantees that $\alpha, \beta \notin act(\psi_1)$ and $\alpha, \beta \notin act(\psi_2)$. Hence, we can apply the inductive hypothesis and consequently, for any decomposition $P \equiv Q|R$ we have either $\sigma(\mathcal{M}), \sigma(Q) \not\models \psi_1$ or $\sigma(\mathcal{M}), \sigma(R) \not\models \psi_2$.
Consider an arbitrary decomposition $\sigma(P) \equiv P'|P''$. There exists $P \equiv Q|R$ such that $\sigma(Q) \equiv P'$ and $\sigma(R) \equiv P''$. Then because $\sigma(P) \equiv \sigma(Q)|\sigma(R)$, we obtain $\sigma(\mathcal{M}), \sigma(P) \not\models \psi_1|\psi_2$.

**The case $\phi = \langle\gamma\rangle\psi$:**

1. $\mathcal{M}, P \models \langle\gamma\rangle\psi$ means that there is a transition $P \xrightarrow{\gamma} Q$ and $\mathcal{M}, Q \models \psi$. Because $\alpha, \beta \notin act(\langle\gamma\rangle\psi)$ implies $\alpha, \beta \notin act(\psi)$ and $\alpha \neq \gamma$ and $\beta \neq \gamma$. We can apply the inductive hypothesis and derive $\sigma(\mathcal{M}), \sigma(Q) \models \psi$. As $P \xrightarrow{\gamma} Q$ we have $P \equiv \gamma.P'|P''$ and $Q \equiv P'|P''$. This mean that $\sigma(P) \equiv \sigma(\gamma).\sigma(P')|\sigma(P'') \equiv \gamma.\sigma(P')|\sigma(P'')$ and $\sigma(Q) \equiv \sigma(P')|\sigma(P'')$. Hence $\sigma(P) \xrightarrow{\gamma} \sigma(Q)$. Now because $\sigma(\mathcal{M}), \sigma(Q) \models \psi$, we derive $\sigma(\mathcal{M}), \sigma(P) \models \langle\gamma\rangle\psi$.

2. $\mathcal{M}, P \not\models \langle\gamma\rangle\psi$ implies one of two cases: either there is no transition of $P$ by $\gamma$, or there is such a transition and for any transition $P \xrightarrow{\gamma} Q$ we have $\mathcal{M}, Q \not\models \psi$.
If there is no transition of $P$ by $\gamma$ then $P \equiv \alpha_1.P_1|...|\alpha_k.P_k$ with $\alpha_i \neq \gamma$

for each $i \neq 1..k$. Because $\sigma(P) \equiv \sigma(\alpha_1).\sigma(P_1)|...|\sigma(\alpha_k).\sigma(P_k)$, and because $\gamma \neq \alpha_i$, $\gamma \neq \alpha$ and $\gamma \neq \beta$, we can state that $\sigma(P)$ cannot perform a transition by $\gamma$, hence $\sigma(\mathcal{M}), \sigma(P) \not\models \langle\gamma\rangle\psi$.
If there are transitions of $P$ by $\gamma$, and for any such a transition $P \xrightarrow{\gamma} Q$ we have $\mathcal{M}, Q \not\models \psi$: then, because from $\alpha, \beta \notin act(\langle\gamma\rangle\psi)$ we can derive $\alpha, \beta \notin act(\psi)$, the inductive hypothesis can be applied and we obtain $\sigma(\mathcal{M}), \sigma(Q) \not\models \psi$. But because $\gamma \neq \alpha$ and $\gamma \neq \beta$ we obtain $\sigma(P) \xrightarrow{\gamma} \sigma(Q)$. Hence $\sigma(\mathcal{M}), \sigma(P) \not\models \langle\gamma\rangle\psi$.

**The case $\phi = \langle R : \gamma\rangle\psi$:**

1. $\mathcal{M}, P \models \langle R : \gamma\rangle\psi$ means that there is a transition $P \xrightarrow{R:\gamma} Q$ and $\mathcal{M}, Q \models \psi$. As $\alpha, \beta \notin act(\langle R : \gamma\rangle\psi)$ implies $\alpha, \beta \notin act(\psi)$ and $\alpha \neq \gamma$ and $\beta \neq \gamma$. We can apply the inductive hypothesis and derive $\sigma(\mathcal{M}), \sigma(Q) \models \psi$. Because $P \xrightarrow{R:\gamma} Q$ we have $P \equiv R|P'$, $R \equiv \gamma.R'|R''$ and $Q \equiv R'|R''|P'$. This entails $\sigma(P) \equiv \sigma(\gamma).\sigma(R')|\sigma(R'')|\sigma(P') \equiv \gamma.R'|R''|\sigma(P')$ and $\sigma(Q) \equiv R'|R''|\sigma(P')$. Hence $\sigma(P) \xrightarrow{R:\gamma} \sigma(Q)$. Now because $\sigma(\mathcal{M}), \sigma(Q) \models \psi$, we derive $\sigma(\mathcal{M}), \sigma(P) \models \langle R : \gamma\rangle\psi$.

2. $\mathcal{M}, P \not\models \langle R : \gamma\rangle\psi$ implies one of two cases: either there is no transition of $P$ by $(R : \gamma)$, or there is such a transition and for any transition $P \xrightarrow{R:\gamma} Q$ we have $\mathcal{M}, Q \not\models \psi$.
If there is no transition of $P$ by $(R : \gamma)$ then $P \not\equiv R|P'$. Because $\alpha, \beta \notin act(R)$ we derive that $\sigma(P) \not\equiv R|S$ for any $S$. Hence, we can state that $\sigma(P)$ cannot perform a transition by $(R : \gamma)$, i.e. $\sigma(\mathcal{M}), \sigma(P) \not\models \langle\gamma\rangle\psi$.
If there are transitions of $P$ by $\gamma$, and for any such a transition $P \xrightarrow{\gamma} Q$ we have $\mathcal{M}, Q \not\models \psi$: then, as from $\alpha, \beta \notin act(\langle\gamma\rangle\psi)$ we can derive $\alpha, \beta \notin act(\psi)$, the inductive hypothesis can be applied and we obtain $\sigma(\mathcal{M}), \sigma(Q) \not\models \psi$. But because $\gamma \neq \alpha$ and $\gamma \neq \beta$ we obtain $\sigma(P) \xrightarrow{\gamma} \sigma(Q)$. Hence $\sigma(\mathcal{M}), \sigma(P) \not\models \langle\gamma\rangle\psi$.

**The case $\phi = K_R\psi$:**

1. $\mathcal{M}, P \models K_R\psi$ implies $P \equiv R|S$ and for any $R|S' \in \mathcal{M}$ we have $\mathcal{M}, R|S' \models \psi$. From $\alpha, \beta \notin act(\phi)$ we derive $\alpha, \beta \notin act(\psi)$ and $\alpha, \beta \notin Act(R)$. So, we can apply the inductive hypothesis that gives $\sigma(\mathcal{M}), \sigma(R|S') \models \psi$ and, further $\sigma(M), R|\sigma(S') \models \psi$. Consider an arbitrary process $R|S'' \in \sigma(\mathcal{M})$. There exists a process $R|S' \in \mathcal{M}$ such that $R|\sigma(S') \equiv R|S''$. Thus $\sigma(\mathcal{M}), R|S'' \models \psi$ and as $R|S''$ was arbitrarily chosen from $\sigma(\mathcal{M})$, we conclude that $\sigma(M), \sigma(P) \models K_R\psi$.

2. $\mathcal{M}, P \not\models K_R\psi$ implies that either $P \not\equiv R|S$ for no $S$, or $P \equiv R|S$ for some $S$ and there exists a process $R|S' \in \mathcal{M}$ such that $\mathcal{M}, R|S' \not\models \psi$.
   If $P \not\equiv R|P'$, because $\alpha, \beta \notin act(R)$ we derive that $\sigma(P) \not\equiv R|S$ for no $S$. Hence, we can state that $\sigma(\mathcal{M}), \sigma(P) \not\models K_R\psi$.
   If $P \equiv R|S$ for some $S$ and there exists a process $R|S' \in \mathcal{M}$ such that $\mathcal{M}, R|S' \not\models \psi$, then the inductive hypothesis gives $\sigma(\mathcal{M}), \sigma(R)|\sigma(S') \not\models \psi$. But $\sigma(R)|\sigma(S') \equiv R|\sigma(S')$, and $\sigma(P) \equiv R|\sigma(S)$ thus $\sigma(\mathcal{M}), R|\sigma(S') \not\models \psi$ implies $\sigma(\mathcal{M}), \sigma(P) \not\models K_R\psi$.

□

But we have **an open problem**:

if $\mathcal{M}$ is a context, is $\mathcal{M}^\sigma$ still a context?

This was taken as an assumption in the previous proof.

## 7.2 Epistemic agents with memory

We think that the system $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ can be pushed further following the idea of assessing more *"intelligent"* actions to the epistemic agents. We recall that in extending $\mathcal{L}^{\mathfrak{S}}_{DES}$ to $\mathcal{L}^{\mathfrak{S}^+}_{DES}$, we considered the possibility that the epistemic agents (and only them) can be seen by the external observer,

being the centers of interest for him. Hence the observer does not see only if the system performed the action $\alpha$, but if the action $\alpha$ was realized by one of the epistemic actions, then the observer will know it.

Such a logic might have interesting applications in causality problems, as we will detail in the next chapter.

Going further, it might be interesting to assume that the epistemic agents, themselves, have memory and they can communicate with the external observer providing information not only about their current actions and states, but also about their history. Thus, the external observer will have some partial information about the past of the system from which it might deduct interesting properties.

Suppose, for example, that our agents are processors running together (possibly with other unknown processors), in a big system. The external observer finds an undesirable result in the current state of the whole system, say a bug. He wants to know if the source of it is one of his processors and thus to fix the problem, or the unknown ones. If his processors (epistemic agents) have memory, he might use some backtrack analysis to understand if the joint work of them can generate the error in any environment. If this is the case, then the problem is originated in his processors; if this is not the case, then the influence of the environment causes it.

In our ongoing work [7] we analyze such a logic. The difference with respect to the logics presented here is that we do not evaluate formulas to processes in contexts, but to traces of computations of agents in contexts.

Hereafter we will outline the main ideas.

**Definition 7.2.1** (Computational traces)**.** Let $\mathcal{M}$ be a context. We define

the set of traces over this context, $\mathcal{T}$, by

$$\mathcal{T} = \{n = (P_0, a_0, P_1, a_1, ...a_{k-1}, P_k) \mid P_i \in \mathcal{M}, a_i \in Act(\mathcal{M}), P_i \xrightarrow{a_i} P_{i+1}\}$$

For each trace $n = (P_0, a_0, P_1, a_1, ...a_{k-1}, P_k)$ we define

$$init(n) \stackrel{def}{=} P_0 \text{ and } fin(n) \stackrel{def}{=} P_k$$

If $n = (P_0, a_0, P_1, a_1, ...a_{k-2}, P_{k-1})$ and $n' = (P_0, a_0, P_1, a_1, ...a_{k-2}, P_{k-1}, a_{k-1}, P_k)$ then we use the notation $n \xrightarrow{a_{k-1}} n'$ and $n \longrightarrow n'$.

Hence, a *computational trace* is a branch of the transition system of $P_0$ that relates $P_0$ with $P_k$. These traces will be used as states to which we will evaluate the formulas of our logic.

Now we define the projection $n_A$ of an epistemic agent $A \in \mathfrak{S}$ on a computational trace $n$. Suppose that

$$n = (P_0, a_0, P_1, a_1, ...a_{k-2}, P_{k-1})$$

and that the agent $A$, in the initial state of the system, can be described by a process $Q_0$ such that $P_0 \equiv Q_0|R$, i.e. the agent $A$ is active. If the system follows the evolution described by the trace $n$, what $A$ *knows* about this is only what $A$ did, i.e. it has its own computational trace $n_A$ that represents what $A$ sees from $n$.

The syntax of such a logic may de defined by the following grammar:

$$\phi := \ 0 \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi|\phi \mid K_A\phi \mid \langle A : \sigma\rangle\phi \mid \ \lhd \phi \mid \ \odot \phi$$

with the semantics

1. $\mathfrak{M}, n \models \top$ always

2. $\mathfrak{M}, n \models 0$ iff $init(n) \equiv 0$
3. $\mathfrak{M}, n \models \phi \wedge \psi$ iff $\mathfrak{M}, n \models \phi$ and $\mathfrak{M}, n \models \psi$
4. $\mathfrak{M}, n \models \neg\phi$ iff $\mathfrak{M}, n \not\models \phi$
5. $\mathfrak{M}, n \models \phi|\psi$ iff $\exists A, B$ such that $init(n) \equiv init(n_A)|init(n_B)$ and $\mathfrak{M}, n_A \models \phi$, $\mathfrak{M}, n_B \models \psi$
6. $\mathfrak{M}, n \models K_A\phi$ iff $n_A \neq \emptyset$ and for all $m \in \mathcal{T}$ such that $m_A = n_A$, we have $\mathfrak{M}, m \models \phi$
7. $\mathfrak{M}, n \models \langle A : \sigma\rangle\phi$ iff $\exists n' \in \mathcal{T}$ such that $n \xrightarrow{A:\sigma} n'$ and $\mathfrak{M}, n' \models \phi$
8. $\mathfrak{M}, n \models \triangleleft\phi$ iff $\forall m \in \mathcal{T}$ such that $m \longrightarrow^* n$, we have $\mathfrak{M}, m \models \phi$
9. $\mathfrak{M}, n \models \odot\phi$ iff $\mathfrak{M}, (init(n)) \models \phi$

The novelty, with respect to the previous logics, is the presence of two temporal operators.

$\mathfrak{M}, n \models \triangleleft\phi$ is the operator $\phi$ *was always satisfied in the past* that allows specification concerning the history of the system in which our agents have been involved.

$\mathfrak{M}, n \models \odot\phi$ means $\phi$ *was satisfied in the initial state*, which allows specification of the moment when an agent was activated and started to register the evolution of the system.

## 7.3 Other alternatives

One of the most interesting subjects to be related with our logics is considering the perspective of calculi with locations. The agents can be also organized in a hierarchy of named locations, such as ambients, that can

be moved as a whole. In this way we approach more complex paradigm, similar to the one studied by the ambient logic [25]. Interesting questions, concerning the decidability problems for such logics, arise from this approach.

Challenging is also the perspective of constructing logics for calculi involving recursive operators. In our previous works we analyzed, independent of the epistemic approach, the perspective of logics for calculi with locations and recursivity. In [52, 49], developing a coalgebraical semantics for Ambient Calculus in terms of Hypersets Theory, we tried to put together Ambient Calculus and modal logic into a mixed equational-coequational approach. We intend to reconsider all these ideas in relation with the dynamic-epistemic approach.

# Chapter 8

# Applications

In this chapter we will put the Dynamic Epistemic Spatial Logics to work on specifying properties for some real systems and thus argue their usefulness.

## 8.1 Security

Consider the scenario of the *e-mail box* receiving messages. *A message* can be a *spam* containing a *virus* that will be installed in our system if we open the attachment, or it can be a useful message that will provide information by opening its attachment. We can describe this scenario using a process calculus as follows.

The *Inbox* can be described as the agent that, being in contact with a message, can perform an *"open"* action (it opens the message), after which it can perform a *"run"* action that refers to the attachment (if any), after that stops.

$$Inb \stackrel{def}{=} open.run.0$$

A message containing harmless information is described as a process that can perform the action $\overline{open}$ (can be opened), then allows the Inbox to run its attachment (can perform $\overline{run}$), after which it reveals information.

$$Msg \stackrel{def}{=} \overline{open}.\overline{run}.Inf$$

A spam message has a structure similar to that of an ordinary message, but after consuming the $\overline{run}$ action it does not reveal information, but instals a virus.

$$Spm \stackrel{def}{=} \overline{open}.\overline{run}.Vrs$$

The intention in introducing the pairs of actions $(run, \overline{run}), (open, \overline{open})$ is to model communication-like complex actions.

Now, using the logic, we can express properties of a system that implies these processes. Suppose that we have two formulas, $i, v$, that describe $Inf$ and $Vrs$ respectively, i.e.

$$Inf \models i \text{ and } Vrs \models v$$

(such formulas do exist as we proved that each process $P$ has a characteristic formula $c_P$ and this formula uniquely describes the process up to structural congruence). Now we can describe the system in which a virus is revealed by

$$Infect \stackrel{def}{=} v|\top$$

while the situation in which information is revealed by

$$Update \stackrel{def}{=} i|\top$$

We can describe the Inbox, using composed actions, by:

$$Inb \models \langle Inb : open\rangle\langle Inb : run\rangle 0.$$

Because in our logic the following holds

$$\vdash \langle P : \sigma\rangle\phi \rightarrow \langle P : \sigma\rangle\top$$

and

$$\vdash \langle P : \sigma \rangle \top \rightarrow K_P \top$$

we can use them and obtain, based on the soundness results, that

$$Inb \models K_{Inb} \langle Inb : open \rangle \langle Inb : run \rangle \top.$$

This means that putting the Inbox in any context we obtain a system that satisfies:

$$Context | Inb \models \langle Inb : open \rangle \langle Inb : run \rangle \top.$$

The use of the epistemic operator allows us to identify the information that characterized the presence of a known subsystem in any context. As our logic allows analysis of smaller systems inside bigger systems and so on, we can hierarchically organize the information available in given situations.

The interaction between the Inbox and a harmless message can be described by

$$Inb | Msg \models \langle Inb : open \rangle \langle Msg : \overline{open} \rangle \langle Inb : run \rangle \langle Msg : \overline{run} \rangle i$$

Further we can use the syntactic characterization of satisfiability and the theorems

$$\vdash \langle P : \sigma \rangle \phi | \psi \rightarrow \langle P : \sigma \rangle (\phi | \psi)$$

and

$$\vdash K_Q \top \rightarrow \phi \text{ implies } K_Q \top \rightarrow K_Q \phi$$

to derive the specification

$$Inb | Msg \models K_{Inb|Msg} \langle Inb : open \rangle \langle Msg : \overline{open} \rangle \langle Inb : run \rangle \langle Msg : \overline{run} \rangle Update$$

But $\vdash \top | K_Q \phi \rightarrow K_Q \phi$, and $\vdash K_Q \phi \rightarrow \phi$, hence

$$Context | Inb | Msg \models \langle Inb : open \rangle \langle Msg : \overline{open} \rangle \langle Inb : run \rangle \langle Msg : \overline{run} \rangle Update$$

Hence we can prove that

$$\langle Inb : open\rangle\langle Msg : \overline{open}\rangle\langle Inb : run\rangle\langle Msg : \overline{run}\rangle Update$$

is a local property of the subsystem $Inb|Msg$, and that any upper-system satisfies it.

Similarly, the interaction between the Inbox and a spam containing a virus can be described by

$$Inb|Spm \models \langle Inb : open\rangle\langle Spm : \overline{open}\rangle\langle Inb : run\rangle\langle Spm : \overline{run}\rangle v$$

wherefrom, as before, we can derive the property

$$Inb|Spm \models K_{Inb|Spm}\langle Inb : open\rangle\langle Spm : \overline{open}\rangle\langle Inb : run\rangle\langle Spm : \overline{run}\rangle Infect$$

that expresses the fact that a system containing the Inbox in parallel with a spam can be infected by a virus, i.e. that:

$$Context|Inb|Spm \models \langle Inb : open\rangle\langle Spm : \overline{open}\rangle\langle Inb : run\rangle\langle Spm : \overline{run}\rangle Infect$$

All this provides a powerful tool to express and argue over the properties of a system such as $Contex|Msg|Spm|Inb$, even if $Context$ is an unknown process, by analyzing its subsystems. So, this system satisfies the properties of $Context|Inb|Spm$ but also of $Context|Inb|Msg$, or of $Context|Inb$, etc.

Suppose that we want to express the property: "*If a system containing Inbox and a spam carrying a virus opens and runs the spam, then the system will be infected*". Such a system has the general form $Context|Inb|Spm$, and because $Context$ represents an unknown process, it could also have the form $Context|Msg|Inb|Spm$. In a logic without dynamic operators for composed actions we could only express about $Context|Msg|Inb|Spm$ properties such as:

$$Context|Msg|Inb|Spm \models \langle open\rangle\langle\overline{open}\rangle\langle run\rangle\langle\overline{run}\rangle Update$$

$$Context|Msg|Inb|Spm \models \langle open\rangle\langle\overline{open}\rangle\langle run\rangle\langle\overline{run}\rangle Infect$$

$$Context|Msg|Inb|Spm \models [open][\overline{open}][run][\overline{run}](Update \vee Infect)$$

or other combinations of this type. We cannot precisely say that what Inbox opens and runs is the spam or just a useful message. But using the dynamic operators for composed actions, we can be more specific in this respect:

$$Context|Msg|Inb|Spm \models$$

$$[Inb : open][Spm : \overline{open}][Inb : run][Spm : \overline{run}]Infect$$

or equivalently

$$Inb|Spm \models K_{Inb|Spm}[Inb : open][Spm : \overline{open}][Inb : run][Spm : \overline{run}]Infect$$

We also have

$$Context|Msg|Inb|Spm \models$$

$$[Inb : open][Spm : \overline{open}][Inb : run][Spm : \overline{run}]\neg Update$$

or equivalently

$$Inb|Spm \models K_{Inb|Spm}[Inb : open][Spm : \overline{open}][Inb : run][Spm : \overline{run}]\neg Update$$

## 8.2 Systems Biology

Recently process algebra has started to be applied in System Biology [19], aiming to model biological systems as computational, concurrent and distributed systems. The hope is to use the computational paradigm of process algebra for handling the complexity of the biological systems. Analyzing biological reality on different levels of abstraction and then combining

the information by means of a well developed theory of systems, is the main goal of this approach.

The next steps to be considered after mapping the biological phenomena in process calculi will be to develop appropriate tools for simulating and model checking properties of these systems.

Because in the case of the biological systems we discuss, almost always, very complex and large systems running in unpredictable environments, it becomes essential to find logics able to specify and model check their properties. The spatial logics without guarantee operator [12] are not very useful in this respect, because they work only with fully described systems. The spatial logics with guarantee operator are expressive enough for capturing contextual facts, but, unfortunately, are undecidable in the context of dynamic or temporal operators [15].

In this sense we propose our logics as an available alternative, being expressive enough to specify contextual properties, but still decidable. Moreover, interpreting biological agents as epistemic agents seems, in some contexts, very natural.

### Immune System performs epistemic actions

*The immune system* is a network of cells, tissues and organs that work together to defend the body against attacks by foreign invaders - *microbes*, *germs*, *bacteria*, *viruses*, *parasites*, etc. It is the immune system's job to keep them out or, failing that, to seek out and destroy them. The immune system functions due to an elaborate and dynamic communications network. Millions of cells, organized into sets and subsets, gather in clouds swarming around a hive and pass information back and forth.

We briefly describe the interaction of such a cell - a macrophage, with a virus [2]. The macrophage is a cell that floats in our system meeting

different cells. When the macrophage encounters cells of an unknown type, it reacts, *suspecting* them of being viruses. This is a *pattern-recognition*-like action that this cell is able to perform. Note the epistemic nature of the macrophage: it *knows* some types of cells and using its knowledge, reacts.

When it encounters an unknown cell, the macrophage engulfs it and tests it to see if it is a dangerous one or not. Viruses will be destroyed and their structures will be learned and transmitted to the whole immune system. In response, the immune system develops some cells specialized in destroying each type of known virus.

The structures of harmless cells met with will be also learned such that, when a macrophage meets this type again, it recognizes it. Notice, again, the epistemic behavior: the cell learns and updates, continuously, the knowledge of the immune system.

We will try now to sketch a possible specification for this interaction and, using our logics, to express and verify some properties.

Denote by $Mphg$ the macrophage, $Vs$ a virus, $C$ an unknown cell, $kC$ a known cell and $P$ a process that cannot be engulfed (does not contain $\overline{engulf}$ active). We introduce the specifications:

$$\begin{gathered} Mphg \stackrel{def}{=} engulf.test.Digest \\ Digest \stackrel{def}{=} \overline{virus}.0|\overline{nonvirus}.0 \\ Vs \stackrel{def}{=} \overline{engulf}.\overline{test}.virus.0 \\ C \stackrel{def}{=} \overline{engulf}.\overline{test}.nonvirus.0 \\ kC \stackrel{def}{=} P \end{gathered}$$

Using our logic we can express the following properties:

$$Mphg|kC \models \neg\langle engulf\rangle\langle\overline{engulf}\rangle\top \wedge \neg\langle\overline{engulf}\rangle\langle engulf\rangle\top$$

that describes the meeting of the macrophage with a known cell emerging without being engulfed. It is useful, for these specifications, to use the definition of communication action in logic, such as

$$\langle \alpha, \overline{\alpha} \rangle \phi \stackrel{def}{=} \langle \alpha \rangle \langle \overline{\alpha} \rangle \phi \vee \langle \overline{\alpha} \rangle \langle \alpha \rangle \phi$$

The property

$$\langle engulf, \overline{engulf} \rangle \langle test, \overline{test} \rangle \langle virus, \overline{virus} \rangle \top$$

expresses the fact that the macrophage that meets a virus will engulf it, will perform a test on it and, as a result of testing, the virus will eventually be destroyed. Starting from the description presented before of $Mphg$ and $Vs$, we can, using our semantics, infer

$$Mphg|Vs \models \langle engulf, \overline{engulf} \rangle \langle test, \overline{test} \rangle \langle virus, \overline{virus} \rangle \top$$

and similarly

$$Mphg|C \models \langle engulf, \overline{engulf} \rangle \langle test, \overline{test} \rangle \langle nonvirus, \overline{nonvirus} \rangle \top$$

that expresses the fact that a non-virus has been met and its structure has been learned. The action $\langle nonvirus, \overline{nonvirus} \rangle$ could be interpreted as the action of learning by the macrophage of a new harmless structure.

All the properties described before are properties concerning small systems where a macrophage meets a harmless cell or a virus. Knowing these small scenarios we compose bigger ones by adding contexts. Using the knowledge operator and the dynamic operators for composed actions, we can logically express and prove situations as follows:

$$Context|Mphg|Vs \models$$

$$K_{Mphg|Vs} \langle Mphg : engulf \rangle \langle Vs : \overline{engulf} \rangle \langle test, \overline{test} \rangle \langle virus, \overline{virus} \rangle \top$$

meaning that any system containing $Mphg|Vs$ as a subsystem satisfies $\langle Mphg : engulf\rangle\langle Vs : \overline{engulf}\rangle\langle test, \overline{test}\rangle\langle virus, \overline{virus}\rangle\top$, and

$$Context|Mphg|C \models$$

$$K_{Mphg|C}\langle Mphg : engulf\rangle\langle C : \overline{engulf}\rangle\langle test, \overline{test}\rangle\langle nonvirus, \overline{nonvirus}\rangle\top$$

meaning that any system containing $Mphg|C$ as subsystem satisfies $\langle Mphg : engulf\rangle\langle C : \overline{engulf}\rangle\langle test, \overline{test}\rangle\langle nonvirus, \overline{nonvirus}\rangle\top$.

Further, more complex specifications can be imagined. For example, we may want to know, given an initial state (possibly implying some unknown context) in which a virus and a macrophage are involved, if in any possible evolution of the system where the virus meets the macrophage, the virus will be destroyed. For this it is enough to prove that the description of our initial state, say $c_S$ (we can use the characteristic formulas to describe it), in any context, we can specify this by $c_S|\top$, proves it, i.e.

$$\vdash c_S|\top \wedge \langle Mphg : engulf\rangle\langle Vs : \overline{engulf}\rangle\top \rightarrow$$

$$[Mphg : engulf][Vs : \overline{engulf}][test, \overline{test}][virus, \overline{virus}]$$

For proving this we either use the axiomatic system and try to deduce it or we use the fact that the system is sound and complete and we try to check the validity for it (which is decidable), i.e.

$$\models c_S|\top \wedge \langle Mphg : engulf\rangle\langle Vs : \overline{engulf}\rangle\top \rightarrow$$

$$[Mphg : engulf][Vs : \overline{engulf}][test, \overline{test}][virus, \overline{virus}]$$

Observe the role of knowledge (learning and common knowledge) in the functioning of the immune system. When a cell learns something, automatically its knowledge becomes common knowledge. This feature of the immune system explains its success in protecting the body from

continuous external attack. On the other hand, when the immune system hits the wrong target or is crippled it can unleash a torrent of diseases, including allergies, arthritis and AIDS.

In abnormal situations, the immune system can mistake self for non-self and launch an attack against the body's own cells or tissues. The result is called an *autoimmune disease*. Some forms of arthritis and diabetes are autoimmune diseases. In other cases, the immune system responds to a seemingly harmless foreign substance such as ragweed pollen. The result is an allergy, and this kind of antigen is called an allergen.

Some types of cancer can also be described as *"epistemic diseases"*, as they are produced by the erroneous management of information on different levels of the biological system.

For this reason we believe that an epistemic logic could help, as a complementary tool, in understanding the functioning of biological systems. Following these ideas we can go further and define:

- *biological information* as the knowledge of a biological agent;
- *a biological agent* as a well-defined subsystem of a given system (it has a well-defined process-like action in a given context).

Biological agents are compositional. Biological information is compositional. Agents can be used to define locations in a system and to describe the flux of information on different layers.

## 8.3 Causality

The notion of causality was proposed in Process Algebra [9, 10, 33, 34, 35] in order to analyze not only processes, but traces in the evolution of a system.

Suppose that we have a complex system composed of many agents running, in parallel, some protocols. Then each state of the system can be described by the parallel composition of different states of these agents. If an external observer registers a state of the system in a given moment, he will not have enough information to compute the previous state of the system.

Consider, for example, the process $P|Q$ that describes the state of a system in a given moment. Suppose that we also know that the system reached this state after it performed an action $\sigma$. All this information is not sufficient to identify the previous state of the system, as any of the processes $\sigma.P|Q$, $P|\sigma.Q$ and $\sigma.(P|Q)$ describe possible previous states.

In this respect developing an epistemic logic for agents with memory could help, as states in our models will be traces in the transition system (see section 7.2). So, knowing the actual process $P|Q$ and the action $\sigma$ we can be in one of the next trace-states: $(\sigma.P|Q, \sigma, P|Q)$, $(P|\sigma.Q, \sigma, P|Q)$ or $(\sigma.(P|Q), \sigma, P|Q)$.

### Performance Analysis and debugging

Suppose that the time cost of running the process involved in the system we analyze is high, and we want to identify the cause for this. Thus we need to identify the subprocess (if any) that originates this situation and maybe replace it with a less expensive one. This could be the case when the processes of our system run on parallel processors. We might be interested to find out on which processor the expensive subprocess runs.

As the logic that handles agents with memory expresses properties of traces in a transition system, we could use it for identifying the undesirable traces and, in this way, identify the cause of the situation, i.e. the processor that generates the delay.

### Diagnosis

Similarly, the logic that handles agents with memory can be used in diagnosis. Consider the situation of a system running many processes in parallel. If, in the final state, we obtain an error in the system, we want to identify, in the history of our system, the origin of this error.

A similar situation happens with a biological system sick with cancer. In such a situation the quantity of information received by immune system increases exponentially and, as a result, the immune system cannot manage the information anymore. Consequently a chaotic behavior is generated that will destroy the organism. In the case of such an illness, no external cause is identified.

If we could understand where, in the flow of information, such a situation became possible, maybe we could better understand this disease. One hope could be an epistemic approach as the problem is intrinsically epistemic.

# Chapter 9

# Conclusions

This thesis introduces a new class of intensional logics for specifying properties of concurrent and distributed systems - the Dynamic Epistemic Spatial Logics. Our logics combines the classic logical operators with spatial operators proposed by spatial logics [13, 14], with dynamic operators similar to those used by the Hennessy-Milner logic [42, 32] and with epistemic operators [43, 37].

For these logics we develop a process semantics based on a finite fragment of CCS that is *the core* of most process calculi. Thus any logical formula will be evaluated, by mean of the satisfiability relation, for a process in a given context, where the contexts are defined as particular classes of processes.

While the use of the classic and spatial logic operators together with dynamic operators is not new, the combination of all these with the epistemic operators is a novelty. Our intuition was to think of processes as epistemic agents and define their knowledge as the sum of properties they satisfy in any state of a given context where they are active. Thus, each process $P$, in a given context $\mathcal{M} \ni P$ knows the spectrum of properties satisfied by any process $P|R \in \mathcal{M}$.

The epistemic operators allow the expression of contextual properties

for our systems that, in the classic spatial logics, were specified by using the guarantee operator. But it has been proved [15] that the guarantee operator in combination with dynamic or temporal operators makes the problems of satisfiability, validity and model checking against process semantics undecidable. This is the main reason for proposing our logics. Even if the epistemic operators cannot express as much as the guarantee operator, they are expressive enough and in combination with the rest of the operators they generate decidable logics.

To prove the finite model property for our logics, we had to introduce a new congruence relation on processes: the structural bisimulation. This relation provides a bisimulation-like definition for the structural congruence. Informally, it is an approximation of the structural congruence bounded by two sizes: the *height* (the depth of the syntactic tree) and the *weight* (the maximum number of bisimilar subprocesses that can be found in a node of the syntactic tree) of a process. The bigger these sizes, the better approximation we obtain. For sizes big enough, we find exactly the structural congruence.

To the best of our knowledge, a similar relation, with which to provide a bisimulation-like description of structural congruence, has not been proposed for process calculi. A conceptually similar congruence was proposed in [16] for analyzing trees of location for the static ambient calculus.

Using this congruence we succeeded in proving the finite model property for all our logics, but also in deriving interesting properties concerning their expressivity. The dynamic epistemic spatial logics distinguish processes up to structural congruence level, as do the other spatial logics. But each formula describes a process not up to structural congruence, but up to the structural bisimulation indexed by its size. Hence two processes that are structurally bisimilar on the size of the formula, cannot be distinguished. Consequently, choosing the right size, we can define characteristic formulas

for our processes. At the best of our knowledge, a similar result has not previously been proved for spatial logics.

In the thesis are three logics developed and a fourth one is outlined.

We begin by analyzing the Dynamic Spatial Logic, the system $\mathcal{L}_{DS}$. Then this system is extended with epistemic operators, obtaining the system $\mathcal{L}^{\mathfrak{S}}_{DES}$, the first Dynamic Epistemic Spatial Logic.

Further we consider composed actions in CCS. This means that we consider transitions of the type $P \xrightarrow{Q:\alpha} P'$, meaning that $P \xrightarrow{\alpha} P'$, but the action $\alpha$ has been performed by the subprocess $Q$ of $P$. By introducing dynamic operators for the composed actions in logic, we extend the system $\mathcal{L}^{\mathfrak{S}}_{DES}$ to the system $\mathcal{L}^{\mathfrak{S}^+}_{DES}$ where more complex properties can be expressed.

For all the logics we propose Hilbert-style axiomatic systems that were proved to be sound and complete with respect to the process semantics. Thus we identified the main axioms and rules that govern the joint work of our operators. For the epistemic systems, the similitude between our axioms and the axioms of the classic epistemic logics is remarkable. Eventually we use our systems to prove meaningful properties for the semantics.

Having sound complete axiomatic systems in the context of decidability, we have powerful tools for specifying and proving properties for the concurrent distributed systems.

The novelty of our logics with respect to spatial logics comes from proposing a way to express contextual properties within the limits of decidability. Thus our epistemic operators replace the guarantee operator, which leads, in spatial logics, to undecidability.

This idea is also new for epistemic logics, because we have an algebraical structure over the class of epistemic agents that allows composed agents to be considered. Thus the processes $P$ and $Q$ are agents with their own knowledge and dynamics, but also $P|Q$, the process running $P$ and $Q$ in

parallel, has its own knowledge and dynamics. Moreover, the knowledge of the composed agent $P|Q$ may be computed from the knowledge of its subsidiary agents $P$ and $Q$. This feature makes it possible to analyze dynamics of knowledge in multi-agent systems where individuals, societies of individuals, societies of societies of individuals, etc., are considered as agents conjointly.

Our intention is to continue studying the advantages of applying epistemic reasoning to multi-agent distributed systems and to try to understand the mathematics of the classical concurrent operators in their interrelations.

# Bibliography


[1] M. Abadi and A.D. Gordon. A calculus for cryptographic protocols. *Information and Computation*, 148(1):1–70, 1999.

[2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland Publishing, Inc., fourth edition, 2002.

[3] A. Baltag. Logics for communication. *Course presented at NASSLLI03. Available at http://www.indiana.edu/ nasslli*.

[4] A. Baltag. A logic for suspicious players: Epistemic actions and belief updates in games. *Bulletin Of Economic Research*, 54(1):1–46, 2002.

[5] A. Baltag and L.S. Moss. Logics for epistemic programs. *Synthese (: Special Section: Knowledge, Rationality and Action).Editors: J. Symons, J. Hintikka. Special Section Editor: W. van der Hoek. Springer Science+Business Media B.V. ISSN: 0039-7857*, 139 (2):165–224, 2004.

[6] A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements. common knowledge and private suspicions. *CWI Technical Report SEN-R9922*, 1999.

[7] Alexandru Baltag and Radu Mardare. Extending the dynamic epistemic spatial logics. *Unpublished manuscript. Available at http://www.dit.unitn.it/ mardare*, 2005.

[8] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal logic*. Cambridge University Press, New York, NY, USA, 2001.

[9] M. Boreale and D. Sangiorgi. A fully abstract semantics for causality in the pi-calculus. *Acta Informatica*, 35:353–400, 1998.

[10] G. Boudol and I. Castellani. A non-interleaving semantics for ccs based on proved transitions. *Fundamanta Informaticae*, 11:433–452, 1988.

[11] S. D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A theory of communicating sequential processes. *J. ACM*, 31(3):560–599, 1984.

[12] Luis Caires. Behavioral and spatial properties in a logic for the pi-calculus. *In Igor Walukiwicz, editor, Proc. of Foundations of Software Science and Computation Structures2004, Lecture Notes in Computer Science, Springer-Verlag*, vol:2987, 2004.

[13] Luis Caires and Luca Cardelli. A spatial logic for concurrency (part ii). *In Proceedings of CONCUR'2002, Lecture Notes in Computer Science, Springer-Verlag*, vol:2421, 2002.

[14] Luis Caires and Luca Cardelli. A spatial logic for concurrency (part i). *Information and Computation*, Vol: 186/2:194–235, November 2003.

[15] Luis Caires and Etienne Lozes. Elimination of quantifiers and decidability in spatial logics for concurrency. *In Proceedings of CONCUR'2004, Lecture Notes in Computer Science, Springer-Verlag*, vol:3170, 2004.

[16] Cristiano Calcagno, Luca Cardelli, and Andrew D. Gordon. Deciding validity in a spatial logic for trees. *In Proceedings of the ACM Workshop on Types in Language Design and Implementation*, pages 62–73, 2003.

[17] L. Cardelli. Mobility and security. *Proceedings of the NATO Advanced Study Institute on Foundations of Secure Computation. IOS Press. ISBN 1 58603 015 9*, 3- 37, 2000.

[18] Luca Cardelli. Brane calculi - interactions of biological membranes. *Proc. BioConcur'03, Electronic Notes in Theoretical Computer Science, Elsevier.*

[19] Luca Cardelli. Bioware languages. *In: Andrew Herbert, Karen Sprck Jones (Eds.): Computer Systems: Theory, Technology, and Applications - A Tribute to Roger Needham, Monographs in Computer Science. Springer*, ISBN 0-387-20170-X.:59–65., 2004.

[20] Luca Cardelli, Giorgio Ghelli, and Andrew D. Gordon. Ambient groups and mobility types. In *IFIP TCS*, pages 333–347, 2000.

[21] Luca Cardelli and Andrew D. Gordon. Mobile ambients. In *Foundations of Software Science and Computation Structures: First International Conference, FOSSACS '98*. Springer-Verlag, Berlin Germany, 1998.

[22] Luca Cardelli and Andrew D. Gordon. Types for mobile ambients. In *Symposium on Principles of Programming Languages*, pages 79–92, 1999.

[23] Luca Cardelli and Andrew D. Gordon. Anytime, anywhere: Modal logics for mobile ambients. *In Proceedings of the 27th ACM Symposium on Principles of Programming Languages*, pages 365–377, 2000.

[24] Luca Cardelli and Andrew D. Gordon. Logical properties of name restriction. *Proceedings of the 5th International Conference on Typed Lambda Calculi and Applications , Springer-Verlag*, 2044 of Lecture Notes in Computer Science:46–60, 2001.

[25] Luca Cardelli and Andrew D. Gordon. Ambient logic. *To appear in Mathematical Structures in Computer Science*, 2003.

[26] Witold Charatonik, A.d. Gordon, and Jean-Marc Talbot. Finite-control mobile ambients. *In D. Metayer, editor, 11th European Symposium on Programming (ESOP 2002)*, 2305 of Lecture Notes in Computer Science. Springer-Verlag, 2002.

[27] Witold Charatonik, Andrew D. Gordon, and Jean-Marc Talbot. Finite-control mobile ambients. In *ESOP '02: Proceedings of the 11th European Symposium on Programming Languages and Systems*, pages 295–313. Springer-Verlag, 2002.

[28] Witold Charatonik and Jean-Marc Talbot. The decidability of model checking mobile ambients. *Proceedings of the 15th Annual Conference of the European Association for Computer Science Logic, Springer-Verlag*, 2142 of Lecture Notes in Computer Science:339–354, 2001.

[29] B. Chellas. *Modal logic. An introduction*, volume Cambridge UP, Cambridge. 1980.

[30] M. Dam. Proof systems for $\pi$-calculus. *In de Queiroz, editor, Logic for Concurrency and Synchronisation, Studies in Logic and Computation. Oxford University Press. To appear.*

[31] M. Dam. Relevance logic and concurrent composition. *In Proceedings of Third Annual Symposium on Logic in Computer Science, Edinburgh, Scotland, July 1988. IEEE Computer Society.*, pages 178–185.

[32] M. Dam. Model checking mobile processes. *Information and Computation*, vol:129(1):35–51, 1996.

[33] Ph. Darondeau and P. Degano. Causal trees. *in G. Ausiello, M. Dezani-Ciancaglini and S. Ronchi Della Rocca, editors, Automata, Languages and Programming*, LNCS 372, Springer:234–248, 1989.

[34] P. Degano, R. De Nicola, and U. Montanari. Partial ordering derivations for ccs. *in L. Budach, editor Fundamantals of Computation Theory*, LNCS 199, Springer:520–533, 1985.

[35] P. Degano and C. Priami. Proved trees. *in W. Kuich, editor Automata, Languages and Programming*, LNCS 623, Springer:629–640, 1992.

[36] E. A. Emerson. *Temporal and Modal Logic. Handbook of Theoretical Computer Science*, volume B: Formal Models and Sematics. 1990.

[37] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[38] M. Gabbay and A. Pitts. A new approach to abstract syntax involving binders. *To appear in Formal Aspects of Computing.*

[39] R. Goldblatt. *Logics of time and computation*, volume CSLI, Stanford. 1987.

[40] Andrew D. Gordon and Luca Cardelli. Equational properties of mobile ambients. In *Foundations of Software Science and Computation Structure*, pages 212–226, 1999.

[41] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[42] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *JACM*, vol: 32(1):137–161, 1985.

[43] Jaako Hintikka. *Knowledge and Belief.* Ithaca, N.Y.: Cornell University Press, 1962.

[44] Daniel Hirschkoff, tienne Lozes, and Davide Sangiorgi. Separability, expressiveness, and decidability in the ambient logic. *Proceedings of the 17th IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press*, pages 423–432, 2002.

[45] C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, 1978.

[46] G. E. Hughes and M. J. Cresswell. *A new introduction to modal logic*, volume Routledge, London. 1996.

[47] W. Groeneveld J. Gerbrandy. Reasoning about information change. *Journal of Logic, Language and Information*, 6:146–169, 1997.

[48] Etienne Lozes. Adjunct elimination in the static ambient logic. *In Proceedings of the 10th International Workshop on Expressiveness in Concurrency, Electronic Notes in Theoretical Computer Science, Elsevier*, vol:96:51–72, 2004.

[49] R. Mardare and C. Priami. Computing the accessibility relation for the ambient calculus. *Technical Report DIT-03-054, Informatica e Telecomunicationi, University of Trento*, 2003.

[50] R. Mardare and C. Priami. A logical approach to security in the context of ambient calculus. *ENTCS*, vol. 99, 2004.

[51] R. Mardare and C. Priami. Logical analysis of biological systems. *Fundamenta Informaticae*, 64::271–285, 2005.

[52] R. Mardare and C. Priami. The principles of ambient calculus revisited. *Technical Report DIT-05-018, Informatica e Telecomunicazioni, University of Trento*, 2005.

[53] R. Mardare, C. Priami, P. Quaglia, and A. Vagin. Model checking biological systems described using ambient calculus. *Proceedings of CMSB04, Lecture Notes in BioInformatics. Berlin: Springer-Verlag*, 3082: 3:85– 10, 2005.

[54] Radu Mardare. A decidable extension of hennessy-milner logic with spatial operators. *Technical Report DIT-06-009, Informatica e Telecomunicationi, University of Trento. Submited*, 2006.

[55] Radu Mardare. Dynamic epistemic spatial logics for conccurency: part 1. *Technical Report DIT-06-010, Informatica e Telecomunicationi, University of Trento. Submited*, 2006.

[56] Radu Mardare. Dynamic epistemic spatial logics for conccurency: part 2. *Technical Report DIT-06-010, Informatica e Telecomunicationi, University of Trento. Submited*, 2006.

[57] R. Milner. *A Calculus of Communicating Systems.* Springer-Verlag New York, Inc., 1982.

[58] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, (25):267–310, 1983.

[59] R. Milner. Process constructors and interpretations. *Information Processing, Proc. IFIP 10th World Computer Congress, Dublin, Ireland*, 1986.

[60] R. Milner. *Communication and concurrency.* Prentice-Hall, Inc., 1989.

[61] R. Milner, J. Parrow, and D. Walker. Modal logics for mobile processes. *Theoretical Computer Science*, vol:114:149–171, 1993.

[62] Robin Milner. *Communicating and Mobile Systems: the Pi-Calculus.* Cambridge University Press, 1999.

[63] Gordon D. Plotkin. A structural approach to operational semantics. *Technical Report FN-19, DAIMI, Department of Computer Science, University of Aarhus, Aarhus, Denmark*, 43, September 1981.

[64] C. Priami and P. Quaglia. Beta Binders for Biological Interactions. In *CMSB '04*, volume 3082 of *LNBI*. Springer, 2005.

[65] C. Priami and P. Quaglia. Operational patterns in beta-binders. In *Transactions on Computational Systems Biology*, volume 3380 of *LNCS*. Springer, 2005.

[66] Aviv Regev, Ekaterina M. Panina, William Silverman, Luca Cardelli, and Ehud Shapiro. Bioambients: an abstraction for biological compartments. *Theor. Comput. Sci.*, 325(1):141–167, 2004.

[67] James Riely and Matthew Hennessy. A typed language for distributed mobile processes. In *Conference Record of POPL 98: The 25TH ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Diego, California*, pages 378–390, New York, NY, 1998.

[68] Davide Sangiorgi and David Walker. *The Pi-calculus. A Theory of Mobile Processes*. Cambridge University Press, 2001.

[69] Colin Stirling. *Modal and temporal properties of processes*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[70] J. F. A. K. van Benthem. Games in dynamic epistemic logic. *Bulletin of Economic Research, Los Altos*, 53(4):219–248, 2001.

[71] J. F. A. K. van Benthem. Logic for information update. *In Proceedings of TARK01, Los Altos*, 2001.

[72] H. van Ditmarsch. Knowledge games. *Bulletin of Economic Research*, 53(4):249–273, 2001.