UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

A DECIDABLE EXTENSION OF HENNESSY-MILNER LOGIC WITH SPATIAL OPERATORS

Radu Mardare and Corrado Priami

January 2006

Technical Report # DIT-06-009

A Decidable Extension of Hennessy-Milner logic with spatial operators*

Radu Mardare[1] and Corrado Priami[1,2]
[1]University of Trento, Italy
[2]Microsoft Research - University of Trento Center
for Computational and Systems Biology

Abstract

In this paper we focus on Dynamic Spatial Logic, the extension of Hennessy-Milner logic with the parallel operator. We develop a sound complete Hilbert-style axiomatic system for it comprehending the behavior of spatial operators in relation with dynamic/temporal ones.
Underpining on a new congruence we define over the class of processes - the structural bisimulation - we prove the finite model property for this logic that provides the decidability for satisfiability, validity and model checking against process semantics.

1 Introduction

Process Algebra [1] models the concurrent distributed systems by abstracting the agents of the system, on the level of their behavior, and using some algebraic calculi and operational semantics [15] describes the evolution of such a system. Inspired by λ-calculus and deeply related with the programming languages, this paradigm succeeds in modelling complex computational scenarios. Further, as the behavior of a concurrent system is, mainly, a succession of affine states in (possibly branching) time, was considered the possibility of applying modal (especially temporal) logics for specifying properties of the systems we modelled.

In studying security problems, for example, we may want to be able to specify systems composed by agents that deal with fresh or secret resources. We may want to express properties such as "*the agent has the key*", "*eventually the agent crosses the firewall*" or "*there is always at most one agent here able to decrypt the message*".

In systems biology [6] we need to handle big complex systems having extreme dimensions and variable environments. We 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 eventually destroyed*", or "*the presence of the protein x will stimulate the reaction X*", etc.

Hennessy-Milner logic [13] is one of the first modal logics that proposes some modal operators, indexed by actions, to describe the behavior of the systems in CCS. The idea was further developed in combination with temporal operators [16] or applied to other calculi [14, 9, 11]. Latter, Mads Dam introduced a tensor that can express properties of modularity in the system [10], i.e. it can identify subsystems of a system. All these logics are characterized by their *extensional nature*, meaning that they cannot distinguish between processes that behave the same, even if these processes are different.

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. 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

*Work partially supported by EU-IST project 016004 SENSORIA

certain location, at some locations or at every location. Thus, *Spatial logics* [3, 2, 8] propose, in addition to the modal temporal operators, some modal spatial operators such as the *parallel operator* $\phi|\psi$ (meaning that the current process can be split into a parallel composition $Q|R$ of a process Q satisfying ϕ and a process R satisfying ψ), and its adjoint - the *guarantee operator* $\phi \triangleright \psi$, or *location operator*[1] $n[\phi]$ (meaning that the current process is an ambient $n[P]$ and the process P satisfies ϕ), etc. 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*, the properties they express being invariant only for simple spatial rearrangements of the system.

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

The decidability of Dynamic Spatial Logic has been anticipated in [4]. Still, on the best of our knowledge, there is no prove in this direction. In this paper we will provide such a prove underpinning on finite model property. In proving the finite model property for our logic, we used a new congruence on processes - *the structural bisimulation*. A conceptually similar congruence has been proposed in [5], 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 we find exactly the structural congruence.

For the logic we propose a complete Hilbert-style axiomatic system, which helps in understanding the basic algebraical behavior of the classical process operators. We prove its soundness and completeness with respect to the process semantics, as usual in spatial logics. Thus, many properties can be syntactically verified and proved. Moreover we have characteristic formulas able to identify a process (agent) up to structural congruence (cloned copies).

2 Hennessy-Milner Logic

Hennessy-Milner logic [13], 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 $Q \models \phi$

[1] This operator is characteristic for Ambient Logic [8], a special spatial logic developed for Ambient Calculus [7].

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 [16]:

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

The associated semantics combines the semantics of Hennessy-Milner logic with the classic semantics of temporal logics [12]:

$P \models AG\phi$ iff for all runs $P \xrightarrow{\mu_1} P_1 \xrightarrow{\mu_2} P_2 \xrightarrow{\mu_3} ...$ and all $i \geq 0$, $P_i \models \phi$
$P \models EF\phi$ iff for some run $P \xrightarrow{\mu_1} P_1 \xrightarrow{\mu_2} P_2 \xrightarrow{\mu_3} ...$ and some i, $P_i \models \phi$
$P \models AF\phi$ iff for all runs $P \xrightarrow{\mu_1} P_1 \xrightarrow{\mu_2} P_2 \xrightarrow{\mu_3} ...$ for some $i \geq 0$, $P_i \models \phi$
$P \models EG\phi$ iff for some run $P \xrightarrow{\mu_1} P_1 \xrightarrow{\mu_2} P_2 \xrightarrow{\mu_3} ...$ 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 ϕ*;
$P \models EF\phi$ means that *some processes reachable from P satisfy ϕ*;
$P \models AF\phi$ means *eventually a process will be reached from P satisfying ϕ*;
$P \models EG\phi$ means *some runs always satisfy ϕ*.

Further, this logic was extended to other process calculi [9, 11, 14].
What is common to all of these is that they satisfy theorem 2.1 proving their extensional nature.

Theorem 2.1. *If $P \sim Q$ and $P \models \phi$ then $Q \models \phi$.*

3 On processes

Definition 3.1. Consider the fragment of CCS generated by the next syntax, where $\mathbb{A}$ is a denumerable set of actions and $\alpha \in \mathbb{A}$:

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

Hereafter this calculus[2] is the object of our paper. We will use α, β to range over $\mathbb{A}$ and we will denote by $\mathfrak{P}$ the class of processes.

We will use this fragment of calculus, further, as semantics for our logic. 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

[2]We can, additionally, consider an involution on $\mathbb{A}$ that associate to each action $\alpha \in \mathbb{A}$ an action $\overline{\alpha} \in \mathbb{A}$, as usual in CCS, and also to take into consideration the silent action τ. But all these represent just syntactic sugar, irrelevant from the point of view of the logic we discuss.

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 with respect to the sizes of the processes involved, we find exactly the structural congruence. A conceptually similar congruence was proposed in [5] 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 recall the definition 3.1 as defining the subcalculus of CCS on which we will focus for the rest of the paper. We will not consider additional features of CCS, 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 silent action τ, but all these can be introduced, in our logic, as derived operators.

Definition 3.2. 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 $P^0 \equiv 0$.

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.*

3.1 Size of a process

Further we propose a definition for the *size of a process*, following a similar idea developed in [5] 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 3.3 (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
  - $P \equiv (\alpha_1.Q_1)^{k_1}|(\alpha_2.Q_2)^{k_2}|...|(\alpha_j.Q_j)^{k_j}$ 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 $w_1 < w_2$.

Remark 3.1. Observe that, by construction, the size of a process is unique up to structural congruence. Moreover, if $[\![P]\!] = (h, w)$ then for any subprocess P' of P we have $[\![P']\!] \leq (h, w)$.

Example 3.1. *We show further the size for some processes:*

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

Definition 3.4 (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[3]. 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).

3.2 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 3.5 (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 3.2. *Consider the processes*

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

We can verify the requirements of the definition 3.5 and decide that $R \approx_2^2 S$. But $R \not\approx_3^2 S$ because on the depth 2 R has an action α (in figure 1 marked with a dashed arrow) while S does not have it (because the height of S is only 2). Also $R \not\approx_2^3 S$ because R contains only 2 (bisimilar) copies of $\beta.0$ while S contains 3 (the extra one is marked with a dashed arrow). 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 α in two ways giving, further, processes in the relation $\approx_0^3$. Indeed

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

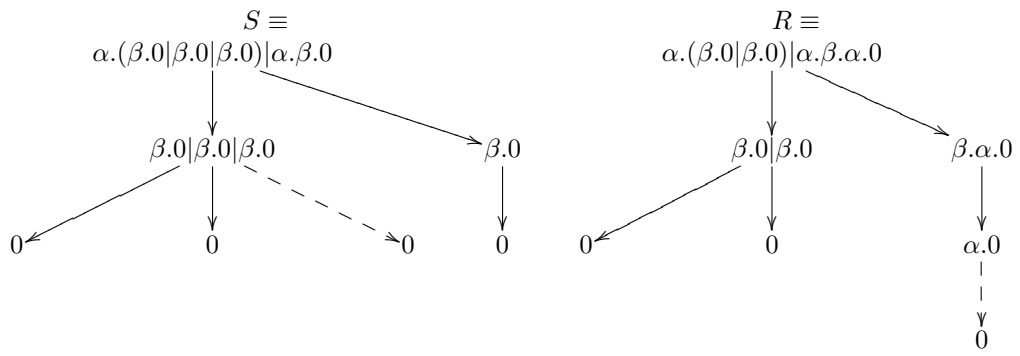

Figure 1: Syntactic trees

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

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 3.1 (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 3.5.

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 3.5, 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 3.5, 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 3.5, 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 3.5, 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 3.2. *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 (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 $P \approx_{h+1}^w Q$, we have $Q \equiv \alpha.Q_1|...|\alpha_i.Q_i|Q'$ with $P_j \approx_h^w Q_j$, 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 $P_j \approx_{h'',w'} Q_j$ for $j = 1..i$. Hence $P \approx_{h''+1}^{w'} Q$, that is, $P \approx_{h'}^{w'} Q$ q.e.d. □

Theorem 3.4 (Congruence). *The following holds:*

1. *if $P \approx_h^w Q$ then $\alpha.P \approx_{h+1}^w \alpha.Q$*
2. *if $P \approx_h^w P'$ and $Q \approx_h^w Q'$ then $P|Q \approx_h^w P'|Q'$*

Proof. 1.: Suppose that $P \approx_h^w Q$. 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 $P \approx_h^w Q$ gives, in the light of definition 3.5, $\alpha.P \approx_{h+1}^w \alpha.Q$.

2.: We prove it by induction on h.
If $h = 0$ then the conclusion is immediate.
For $h + 1$, suppose that $P \approx_{h+1}^w P'$ and $Q \approx_{h+1}^w Q'$; then consider any $i = 1..w$, α 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 $P \approx_{h+1}^w P'$ we have $P' \equiv \alpha.P'_1|...|\alpha.P'_k|P_0$ such that $R_j \approx_h^w P'_j$ for $j = 1..k$.
Similarly, from $Q \approx_{h+1}^w Q'$ we have $Q' \equiv \alpha.Q'_{k+1}|...|\alpha.Q'_i|Q_0$ such that $Rj \approx_h^w Q'_j$ 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 $R_j \approx_h^w P'_j$ for $j = 1..k$ and $R_j \approx_h^w Q'_j$ for $j = (k + 1)..i$, and because a similar argument starting from $P'|Q'$ is possible, we proved that $P|Q \approx_{h+1}^w P'|Q'$. □

Theorem 3.5 (Inversion). *If $P'|P'' \approx_h^{w_1+w_2} Q$ then exists Q', Q'' such that $Q \equiv Q'|Q''$ and $P' \approx_h^{w_1} Q'$, $P'' \approx_h^{w_2} Q''$.*

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 $P'|P'' \approx_{h+1}^{w} Q$.

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 $P_1 \approx_h^w P_2$ then $P_1 \equiv P_2$. Note that $P \approx_{h+1}^w \alpha_1.P_1|\alpha_2.P_1|P_3$. 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 $P \approx_{h+1}^w P_0$.

Now, we can suppose, without loosing generality, that[4]:

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

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:

- if $k'_i + k''_i < w_1 + w_2$ then $P'|P'' \approx_{h+1}^w Q$ 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 $P'|P'' \approx_{h+1}^w Q$ 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 3.4. □

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. $P \approx_h^w Q$ 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 3.6. *If $[\![P]\!] \leq (h, w)$ and $[\![P']\!] \leq (h, w)$ then $P \approx_h^w P'$ 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 3.2.
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, that

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

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

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 $P \approx^w_{h+1} P'$ provides that $P' \equiv \alpha_i.Q''_1|...\alpha_i.Q''_{k_i}|R$ with $Q_i \approx^w_h Q''_j$ 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 3.7. *If $P \approx^w_h Q$ 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 $h > h'$).
The case $h = 1$: we have $h' = 0$ that gives $P \equiv 0$. Further $0 \approx^w_1 Q$ 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 $P \equiv (\alpha_i.P_i)^{k_i}|P^+$, $P \approx^w_h Q$ and $k_i < w$, we obtain that $Q \equiv \alpha_i.R_1|...|\alpha_i.R_{k_i}|R^+$ with $P_i \approx^w_{h-1} R_j$ for any $j = 1..k_i$.
But $P_i \approx^w_{h-1} R_j$ 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 $P \approx^w_h Q$, 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 $P \approx^w_h Q$ 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 $P'_j \approx^w_{h-1} P_i$ 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 3.8. *If $P \equiv R|P'$, $P \approx^w_h Q$ and $[\![R]\!] < (h, w)$ then $Q \equiv R|Q'$.*

Proof. Suppose that $[\![R]\!] = (h', w') < (h, w)$. Because $P \equiv R|P'$ and $P \approx^w_h Q$, using theorem 3.5, we obtain that exists Q_1, Q_2 such that $Q \equiv Q_1|Q_2$ and $R \approx^{w'+1}_h Q_1$ and $P' \approx^{w-(w'+1)}_h Q_2$.
Further, as $R \approx^{w'+1}_h Q_1$ and $[\![R]\!] = (h', w') < (h, w'+1)$ we obtain, by using theorem 3.7, that $Q_1 \equiv R$, hence $Q \equiv R|Q_2$. □

Theorem 3.9. *Let $P \approx^w_h Q$. If $P \equiv \alpha.P'|P''$ then $Q \equiv \alpha.Q'|Q''$ and $P'|P'' \approx^{w-1}_{h-1} Q'|Q''$*

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:

$$P'|P'' \equiv \beta.P_1|...|\beta.P_i|P^\star \tag{3.1}$$

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

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

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

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

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

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

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

  with $R_j \approx_{h-1}^w P_j$ for $j = k+1..i$ and $R' \approx_{h-1}^w P'$. Because $P' \approx_{h-1}^w Q'$ and $\approx_h^w$ 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 $Q' \approx_{h-1}^w P_s$ and as $Q' \approx_{h-1}^w P'$ and $P' \approx_{h-1}^w R'$ we derive $R' \approx_{h-1}^w P_s$ and $Q' \approx_{h-1}^w P'$, 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 (3.1) with $Q|Q'$). □

Theorem 3.10 (Behavioral simulation). *Let $P \approx^w_h Q$. If $P \xrightarrow{\alpha} P'$ then exists a transition $Q \xrightarrow{\alpha} Q'$ such that $P' \approx^{w-1}_{h-1} Q'$.*

Proof. If $P \xrightarrow{\alpha} P'$ then $P \equiv \alpha.R'|R''$ and $P' \equiv R'|R''$. But $P \approx^w_h Q$ gives, using theorem 3.9 that $Q \equiv \alpha.S'|S''$ and $R'|R'' \approx^{w-1}_{h-1} S'|S''$. And because $Q \xrightarrow{\alpha} S'|S''$, we can take $Q' \equiv S'|S''$. □

3.3 Bound pruning processes

In this subsection we prove the bound 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 $P \approx^w_h Q$. 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 given size.

Theorem 3.11 (Bound pruning theorem). *For any process $P \in \mathfrak{P}$ and any (h, w) exists a process $Q \in \mathfrak{P}$ with $P \approx^w_h Q$ and $[\![Q]\!] \leq (h, w)$.*

Proof. We describe the construction[5] of Q by induction on h.

For $h = 0$: we just take $Q \equiv 0$, because $P \approx^w_0 Q$ and $[\![0]\!] = (0, 0)$.

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 $P_i \approx^w_h P'_i$ (by the inductive hypothesis), we obtain, using theorem 3.4, that $\alpha_i.P_i \approx^w_{h+1} \alpha_i.P'_i$ and further $P \approx^w_{h+1} P'$.

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 3.6 (Bound 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 3.11.

Example 3.3. *Consider the process $P \equiv \alpha.(\ \beta.(\gamma.0|\gamma.0|\gamma.0)\ |\ \beta.\gamma.0\ )\ |\ \alpha.\beta.\gamma.0$.*
Observe that $[\![P]\!] = (3, 3)$, hence $P_{(3,3)} \equiv P$. For constructing $P_{(3,2)}$ we have to prune the syntactic tree of P such that to not exist, in any node, more than two bisimilar branches. Hence $P_{(3,2)} = \alpha.(\ \beta.(\gamma.0|\gamma.0)\ |\ \beta.\gamma.0)\ |\ \alpha.\beta.\gamma.0$
If we want to prune P on the size $(3, 1)$, we have to prune its syntactic tree such that, in any node, there are no bisimilar branches. The result is $P_{(3,1)} = \alpha.\beta.\gamma.0$.
For pruning P on the size $(2, 2)$, we have to prune all the nodes on depth 2 and in the new tree we have to let, in any node, a maximum of two bisimilar branches. As a result of these modifications, we obtain $P_{(2,2)} = \alpha.(\beta.0|\beta.0)\ |\ \alpha.\beta.0$. Going further we 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 3.12. *If $P \equiv Q$ then $P_{(h,w)} \equiv Q_{(h,w)}$.*

[5] This construction is not necessarily unique.

Proof. Because a process is unique up to structural congruence, the result can be derived trivially, following the construction in the proof of theorem 3.11. □

Corollary 3.13. $[\![P]\!] \leq (h,w)$ *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 3.6 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 3.1. □

3.4 Substitutions

For the future constructs is also useful to introduce the substitutions of actions in a process.

Definition 3.7 (The set of actions of a process). We define $Act(P) \subset \mathbb{A}$, inductively by:

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

For a set $M \subset \mathfrak{P}$ of processes we define $Act(M) \stackrel{def}{=} \bigcup_{P \in M} Act(P)$.

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

Definition 3.8. Let $A \subset \mathbb{A}$. We define

$$\mathfrak{P}^A_{(h,w)} \stackrel{def}{=} \{P \in \mathfrak{P} \mid Act(P) \subset A,\ [\![P]\!] \leq (h,w)\}$$

Theorem 3.14. *If* $A \subset \mathbb{A}$ *is finite, then* $\mathfrak{P}^A_{(h,w)}$ *is finite*[6].

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

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

We prove this by induction on h.

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

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

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

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

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

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

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

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

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

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

Definition 3.9 (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 σ for sets of processes $M \subset \mathfrak{P}$ by $\sigma(M) \stackrel{def}{=} \{\sigma(P) \mid P \in M\}$.
For short, we will denote, sometimes, $\sigma(P)$ by P^σ and $\sigma(M)$ by M^σ.

Remark 3.2. Observe that $P \equiv Q$ entails $Act(P) = Act(Q)$ and $P^\sigma \equiv Q^\sigma$.

Definition 3.10. Let σ be a substitution. We define the *subject of* σ, $sub(\sigma)$ and the *object of* σ, $obj(\sigma)$, by:

$$sub(\sigma) \stackrel{def}{=} \{\alpha \in \mathbb{A} \mid \sigma(\alpha) \neq \alpha\}$$

$$obj(\sigma) \stackrel{def}{=} \{\beta \in \mathbb{A} \mid \beta \neq \alpha,\ \sigma(\alpha) = \beta\}$$

Theorem 3.15. *If* $sub(\sigma) \cap Act(P) = \emptyset$ *then* $\sigma(P) \equiv P$.

Proof. We prove it by induction on P.
The case $P \equiv 0$**:** by definition, $\sigma(0) \equiv 0$.
The case $P \equiv \alpha.Q$**:** $\sigma(P) \equiv \sigma(\alpha).\sigma(Q)$. But $\alpha \in Act(P)$, and because $Act(P) \cap sub(\sigma) = \emptyset$, we obtain $\alpha \notin sub(\sigma)$, hence $\sigma(\alpha) = \alpha$. But then $\sigma(P) \equiv \alpha.\sigma(Q)$. Further $Act(Q) \subset Act(P)$, i.e. $Act(Q) \cap sub(\sigma) = \emptyset$ and we can apply the inductive hypothesis that provides $\sigma(Q) \equiv Q$, so $\sigma(P) \equiv \alpha.Q$, q.e.d.
The case $P \equiv Q|R$**:** $\sigma(P) \equiv \sigma(Q)|\sigma(R)$. But $Act(Q), Act(R) \subset Act(P)$, hence $Act(Q) \cap sub(\sigma) = Act(R) \cap sub(\sigma) = \emptyset$. Hence we can apply the inductive hypothesis that provides $\sigma(Q) \equiv Q$ and $\sigma(R) \equiv R$, thus $\sigma(P) \equiv Q|R \equiv P$. □

Theorem 3.16. *If* $obj(\sigma) \cap Act(P) = \emptyset$ *then* $\sigma(Q) \equiv P$ *implies* $Q \equiv P$.

Proof. We prove it by induction on P.
If $P \equiv 0$**:** if $Q \not\equiv 0$ then $Q \equiv \alpha.Q'|Q''$, thus $\sigma(Q) \equiv \sigma(\alpha).\sigma(Q')|\sigma(Q'') \not\equiv 0$. Impossible.
If $P \not\equiv 0$**:** Suppose that

$$P \equiv \alpha_1.P_1|...|\alpha_n.P_n$$

and

$$Q \equiv \beta_1.Q_1|...|\beta_m.Q_m$$

Then $\sigma(Q) \equiv \sigma(\beta_1).\sigma(Q_1)|...|\sigma(\beta_m).\sigma(Q_m)$ and

$$\alpha_1.P_1|...|\alpha_n.P_n \equiv \sigma(\beta_1).\sigma(Q_1)|...|\sigma(\beta_m).\sigma(Q_m)$$

But then $m = n$ and for each $i = 1..n$ there exists $j = 1..n$ such that $\alpha_i.P_i \equiv \sigma(\beta_j).\sigma(Q_j)$, thus $\alpha_i = \sigma(\beta_j)$. But from $obj(\sigma) \cap Act(P) = \emptyset$ we derive $\sigma(\beta_j) = \beta_j = \alpha_i$. Further, from $\alpha_i.P_i \equiv \sigma(\beta_j).\sigma(Q_j)$ we infer $P_i \equiv \sigma(Q_j)$, and since $Act(P_i) \subset Act(P)$, we can use the inductive hypothesis and derive $P_i \equiv Q_j$. Thus $P \equiv Q$. □

Theorem 3.17. *If $\sigma(P) \equiv Q|R$ then there exist processes Q', R' such that $P \equiv Q'|R'$, with $\sigma(Q') \equiv Q$ and $\sigma(R') \equiv R$.*

Proof. Suppose that $P \equiv \alpha_1.P_1|...|\alpha_n.P_n$. Then

$$\sigma(P) \equiv \sigma(\alpha_1).\sigma(P_1)|...|\sigma(\alpha_n).\sigma(P_n) \equiv Q|R$$

We can suppose, without loosing generality, that

$$Q \equiv \sigma(\alpha_1).\sigma(P_1)|...|\sigma(\alpha_i).\sigma(P_i)$$

$$R \equiv \sigma(\alpha_{i+1}).\sigma(P_{i+1})|...|\sigma(\alpha_n).\sigma(P_n)$$

Then we can define $Q' \equiv \alpha_1.P_1|...|\alpha_i.P_i$ and $R' \equiv \alpha_{i+1}.P_{i+1}|...|\alpha_n.P_n$. □

Theorem 3.18. *If $P \not\equiv R|Q$ and $obj(\sigma) \cap Act(R) = \emptyset$, then $\sigma(P) \not\equiv R|S$.*

Proof. Suppose that $\sigma(P) \equiv R|S$ for some S. Then, by the theorem 3.17, there exists R', S' such that $P \equiv S'|R'$ and $\sigma(R') \equiv R$, $\sigma(S') \equiv S$. But because $obj(\sigma) \cap Act(R) = \emptyset$ and $\sigma(R') \equiv R$, we derive, applying the theorem 3.16, that $R' \equiv R$, hence $P \equiv R|S'$. But this contradicts the hypothesis of the theorem. So, there is no S such that $\sigma(P) \equiv R|S$. □

4 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 of our system or of a subsystem of our system. Hence if a process belongs to a context then any process obtained by pruning its syntactic tree, in any way[7], should belong to the context, as it might represent a subsystem. For the same reason, the context should be also closed to transitions.

It is useful in this point to define some operations on sets of processes.

Definition 4.1. For any sets of processes $M, N \subset \mathfrak{P}$ and any $\alpha \in \mathbb{A}$ we define:

$\alpha.M \stackrel{def}{=} \{\alpha.P \mid P \in M\}$ $\qquad$ $M|N \stackrel{def}{=} \{P|Q \mid P \in M, Q \in N\}$

As we speak about processes up to structural congruence, the parallel operator on sets of processes will be commutative, associative and will have $\{0\}$ as null.

We associate further to each process P the set $\pi(P)$ of all processes obtained by pruning, in the most general way, the syntactic tree of P.

Definition 4.2. For $P \in \mathfrak{P}$ we define[8] $\pi(P) \subset \mathfrak{P}$ inductively by:

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

We extend the definition of π to sets of processes $M \subset \mathfrak{P}$ by $\pi(M) \stackrel{def}{=} \bigcup_{P \in M} \pi(P)$.

Theorem 4.1. *The next assertions hold:*

1. $P \in \pi(P)$ $\qquad$ 2. $0 \in \pi(P)$ $\qquad$ 3. $P \in \pi(P|Q)$ $\qquad$ 4. $P_{(h,w)} \in \pi(P)$

Proof. 1. We prove it by induction on P

[7] We do not refer here on bound pruning only, but on any possible pruning of the syntactic tree.

[8] We consider also $\pi(P)$ defined up to structural congruence.

- if $P \equiv 0$ then $\pi(P) = \{0\} \ni 0 \equiv P$
- if $P \equiv \alpha.Q$ then $\pi(P) = \{0\} \cup \alpha.\pi(Q)$. But the inductive hypothesis gives $Q \in \pi(Q)$, hence $\alpha.Q \in \alpha.\pi(Q) \subset \pi(P)$.
- if $P \equiv Q|R$ then $\pi(P) = \pi(Q)|\pi(R)$. The inductive hypothesis provide $Q \in \pi(Q)$ and $R \in \pi(R)$, hence $P \equiv Q|R \in \pi(Q)|\pi(R) = \pi(P)$.

2. We prove it by induction on P.
   - if $P \equiv 0$ we have, by definition, $\pi(P) = \{0\} \ni 0$
   - if $P \equiv \alpha.Q$ then $\pi(P) = \{0\} \cup \alpha.\pi(Q) \ni 0$.
   - if $P \equiv Q|R$ then $\pi(P) = \pi(Q)|\pi(R)$. The inductive hypothesis provide $0 \in \pi(Q)$ and $0 \in \pi(R)$, hence $0 \equiv 0|0 \in \pi(Q)|\pi(R) = \pi(P)$.

3. We have $\pi(P|Q) = \pi(P)|\pi(Q)$. But $P \in \pi(P)$ and $0 \in \pi(Q)$, hence $P \equiv P|0 \in \pi(P)|\pi(Q) = \pi(P|Q)$.
4. We prove the theorem by induction on the structure of P.
   - if $P \equiv 0$: we have $P_{(h,w)} \equiv 0 \in \{0\} = \pi(P)$ for any (h, w).
   - if $P \equiv \alpha.Q$: we distinguish two more cases:
     if $w = 0$ then $P_{(h,0)} \equiv 0 \in \pi(P)$
     if $w \neq 0$ then $(\alpha.Q)_{(h,w)} \equiv \alpha.Q_{(h-1,w)}$ by the construction of the adjusted processes. If we apply the inductive hypothesis we obtain that $Q_{(h-1,w)} \in \pi(Q)$, hence $(\alpha.Q)_{(h,w)} \in \alpha.\pi(Q) \subset \pi(P)$.
   - if $P \equiv (\alpha.Q)^k$: we have $P_{(h,w)} \equiv (\alpha.Q_{(h-1,w)})^l$ where $l = min(k, w)$, by the construction of the adjusted processes. The inductive hypothesis gives $Q_{(h-1,w)} \in \pi(Q)$, hence $\alpha.Q_{(h-1,w)} \in \alpha.\pi(Q) \subset \pi(\alpha.Q)$. But because $0 \in \pi(\alpha.Q)$ and
   $$P_{(h,w)} \equiv \underbrace{\alpha.Q_{(h-1,w)}|...|\alpha.Q_{(h-1,w)}}_{l} \,|\, \underbrace{0|...|0}_{k-l}$$
   we obtain
   $$P_{(h,w)} \in \underbrace{\pi(\alpha.Q)|...|\pi(\alpha.Q)}_{k} = \pi(P)$$
   - if $P \equiv (\alpha_1.P_1)^{k_1}|...|(\alpha_n.P_n)^{k_n}$ with $n \geq 2$: we split it in two subprocesses $Q \equiv (\alpha_1.P_1)^{k_1}|...|(\alpha_i.P_i)^{k_i}$ and $R \equiv (\alpha_{i+1}.P_{i+1})^{k_{i+1}}|...|(\alpha_n.P_n)^{k_n}$. By the way we split the process P we will have $P_{(h,w)} \equiv Q_{(h,w)}|R_{(h,w)}$ and using the inductive hypothesis on Q and R we derive $P_{(h,w)} \equiv Q_{(h,w)}|R_{(h,w)} \in \pi(Q)|\pi(R) = \pi(P)$.

□

Theorem 4.2. *1.* $Act(\pi(P)) \subseteq Act(P)$ *2. If* $P \longrightarrow Q$ *then* $Act(Q) \subseteq Act(P)$.

Proof. 1. We prove it by induction on P.
if $P \equiv 0$ **then** $Act(\pi(P)) = Act(\emptyset) = \emptyset \subseteq Act(P)$.
if $P \equiv \alpha.Q$ **then** $Act(\pi(P)) = Act(\{0\} \cup \alpha.\pi(Q)) = Act(\alpha.\pi(Q)) = \{\alpha\} \cup Act(\pi(Q))$. By inductive hypothesis, $Act(\pi(Q)) \subseteq Act(Q)$, hence $Act(\pi(P)) \subseteq \{\alpha\} \cup Act(Q) = Act(P)$.
if $P \equiv Q|R$ **then** $Act(\pi(P)) = Act(\pi(Q)|\pi(R)) = Act(\pi(Q)) \cup Act(\pi(R))$. Using the inductive hypothesis, $Act(\pi(Q)) \subseteq Act(Q)$ and $Act(\pi(R)) \subseteq Act(R)$, hence $Act(\pi(P)) \subseteq Act(Q) \cup Act(R) = Act(Q|R) = Act(P)$.
2. If $P \longrightarrow Q$ then $P \equiv \alpha.Q_1|Q_2$ and $Q \equiv Q_1|Q_2$. Then $Act(Q) = Act(Q_1) \cup Act(Q_2) \subseteq \{\alpha\} \cup Act(Q_1) \cup Act(Q_2) = Act(P)$. □

Theorem 4.3. $\pi(\pi(P)) = \pi(P)$.

Proof. We prove it by induction on P.
The case $P \equiv 0$**:** $\pi(\pi(0)) = \pi(\{0\}) = \pi(0)$
The case $P \equiv \alpha.Q$**:** $\pi(\pi(\alpha.Q)) = \pi(\{0\} \cup \alpha.\pi(Q)) = \pi(0) \cup \pi(\alpha.\pi(Q)) = \{0\} \cup \alpha.\pi(\pi(Q))$. Now we can use the inductive hypothesis and we obtain $\pi(\pi(Q)) = \pi(Q)$. Hence $\pi(\pi(\alpha.Q)) = \{0\} \cup \alpha.\pi(Q) = \pi(\alpha.Q) = \pi(P)$.
The case $P \equiv Q|R$**:** $\pi(\pi(P)) = \pi(\pi(Q|R)) = \pi(\pi(Q)|\pi(R)) = \pi(\pi(Q))|\pi(\pi(R))$. Now we ca apply the inductive hypothesis on Q and R and obtain $\pi(\pi(P)) = \pi(Q)|\pi(R) = \pi(Q|R) = \pi(P)$. □

Theorem 4.4. *If* $Q \in \pi(P)$ *then* $\pi(Q) \subset \pi(P)$.

Proof. $Q \in \pi(P)$ implies $\pi(Q) \subset \pi(\pi(P))$, and applying the theorem 4.3, we obtain $\pi(Q) \subset \pi(P)$. □

Theorem 4.5. *If* σ *is a substitution, then* $\pi(\sigma(P)) = \sigma(\pi(P))$.

Proof. We prove it by induction on P.
The case $P \equiv 0$**:** $\pi(\sigma(P)) = \pi(0) = \{0\} = \sigma(\{0\}) = \sigma(\pi(P))$.
The case $P \equiv \alpha.Q$**:** $\pi(\sigma(P)) = \pi(\sigma(\alpha).\sigma(Q)) = \{0\} \cup \sigma(\alpha).\pi(\sigma(Q))$. But the inductive hypothesis gives $\pi(\sigma(Q)) = \sigma(\pi(Q))$, hence

$$\pi(\sigma(P)) = \{0\} \cup \sigma(\alpha).\sigma(\pi(Q))$$

from the other side, $\sigma(\pi(P)) = \sigma(\{0\} \cup \alpha.\pi(Q)) = \{0\} \cup \sigma(\alpha).\sigma(\pi(Q))$.
The case $P \equiv Q|R$**:** $\pi(\sigma(Q|R)) = \pi(\sigma(Q)|\sigma(R)) = \pi(sigma(Q))|\pi(\sigma(R))$. But the inductive hypothesis gives $\pi(\sigma(Q)) = \sigma(\pi(Q))$ and $\pi(\sigma(R)) = \sigma(\pi(R))$. Hence $\pi(\sigma(P)) = \sigma(\pi(Q))|\sigma(\pi(R)) = \sigma(\pi(Q)|\pi(R)) = \sigma(\pi(P))$. □

These being proved, we can propose the definition of context:

Definition 4.3 (Context). *A context* is a nonempty set $\mathcal{M} \subseteq \mathfrak{P}$ of processes such that

- if $P \in \mathcal{M}$ and $P \longrightarrow P'$ then $P' \in \mathcal{M}$
- if $P \in \mathcal{M}$ then $\pi(P) \subset \mathcal{M}$

Theorem 4.6. *If* $\mathcal{M}$ *is a context and* σ *a substitution, then* $\mathcal{M}^\sigma$ *is a context.*

Proof. Let $P \in \mathcal{M}^\sigma$. Then it exists a process $Q \in \mathcal{M}$ such that $\sigma(Q) \equiv P$. Then $\pi(P) = \pi(\sigma(Q))$, and using theorem 4.5 we derive $\pi(P) = \sigma(\pi(Q))$. But $Q \in \mathcal{M}$ implies $\pi(Q) \subset \mathcal{M}$, thus $\sigma(\pi(Q)) \subset \mathcal{M}^\sigma$. Then $\pi(P) \subset \mathcal{M}^\sigma$.
Let $P \in \mathcal{M}^\sigma$ and $P \longrightarrow P'$. Then it exists $Q \in \mathcal{M}$ such that $\sigma(Q) \equiv P$. Suppose that

$$Q \equiv \alpha_1.Q_1|...|\alpha_k.Q_k$$

then

$$P \equiv \sigma(Q) \equiv \sigma(\alpha_1).\sigma(Q_1)|...|\sigma(\alpha_k).\sigma(Q_k)$$

But then $P \longrightarrow P'$ gives that it exists $i = 1..k$ such that

$$P' \equiv \sigma(\alpha_1).\sigma(Q_1)|...|\sigma(\alpha_{i-1}).\sigma(Q_{i-1}) \mid \sigma(Q_i) \mid \sigma(\alpha_{i+1}).\sigma(Q_{i+1})|...|\sigma(\alpha_k).\sigma(Q_k)$$

and if we define

$$Q' \equiv \alpha_1.Q_1|...|\alpha_{i-1}.Q_{i-1} \mid Q_i \mid \alpha_{i+1}.Q_{i+1}|...|\alpha_k.Q_k$$

we obtain $Q \longrightarrow Q'$ (i.e. $Q' \in \mathcal{M}$) and $\sigma(Q') \equiv P'$. Hence $P' \in \mathcal{M}^\sigma$. □

Observe that, due to the closure clauses in definition 4.3, we can consider the possibility to define systems of generators for a context, as a class of processes that, using the rules in definition 4.3 can generate the full context.

Definition 4.4 (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 $\overline{M} = \mathcal{M}$.

Theorem 4.7. *If $M \in \mathfrak{P}$ is a finite set of processes, then $\overline{M}$ is a finite context.*

Proof. Trivial. □

4.1 Structural bisimulation on contexts

We 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 bound pruning theorem proved for processes.

Definition 4.5 (Structural bisimulation over contexts). Let $\mathcal{M}, \mathcal{N}$ be two contexts. We write $\mathcal{M} \approx_h^w \mathcal{N}$ iff

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

We convey to 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 $\mathcal{M} \approx_h^w \mathcal{N}$.

Theorem 4.8 (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 we obtain $P \approx_{h'}^{w'} Q$. And the same if we start from a process $Q \in N$. These proves that $\mathcal{M} \approx_{h'}^{w'} \mathcal{N}$. □

4.2 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 4.6 (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 4.9. *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 $P \approx_w^h Q$). □

Theorem 4.10. *For any context $\mathcal{M}$ and any size (h, w) we have $Act(\mathcal{M}_{(h,w)}) \subseteq Act(\mathcal{M})$.*

Proof. As $P_{(h,w)} \in \pi(P)$ for any process $P \in \mathcal{M}$ and any (h, w), by theorem 4.1, we obtain, by applying theorem 4.2, $Act(P_{(h,w)}) \subseteq Act(\mathcal{M})$, hence $Act(\{P_{(h,w)} \mid P \in \mathcal{M}\}) \subseteq Act(\mathcal{M})$. Further applying again theorem 4.2, we trivially derive the desired result. □

Definition 4.7. Let $A \subset \mathbb{A}$. We denote by $\mathfrak{M}^A_{(h,w)}$ the set of all contexts generated by systems with the size at most (h, w) and the actions in A:

$$\mathfrak{M}^A_{(h,w)} \stackrel{def}{=} \{\overline{M} \subset \mathfrak{P} \mid Act(M) \subseteq A,\ [\![M]\!] \leq (h, w)\}$$

Theorem 4.11. *If $A \subset \mathbb{A}$ is a finite set of actions, then the following hold:*

1. *If $\mathcal{M} \in \mathfrak{M}^A_{(h,w)}$ then $\mathcal{M}$ is a finite context.*
2. *$\mathfrak{M}^A_{(h,w)}$ is finite.*

Proof. 1.: If $\mathcal{M} \in \mathfrak{M}^A_{(h,w)}$ then $\mathcal{M} = \overline{M}$, $[\![M]\!] \leq (h, w)$ and $Act(M) \subset A$. Thus $M \subset \mathfrak{P}^A_{(h,w)}$. But $\mathfrak{P}^A_{(h,w)}$ is finite, by theorem 3.14. Thus, by theorem 4.7, $\overline{M} = \mathcal{M}$ is a finite context.

2.: As $\mathfrak{P}^A_{(h,w)}$ is finite by theorem 3.14, the set of its subsets is finite, and as all the elements of $\mathfrak{M}^A_{(h,w)}$ are generated by subsets of $\mathfrak{P}^A_{(h,w)}$, we obtain that $\mathfrak{M}^A_{(h,w)}$ is finite. □

Theorem 4.12 (Pruning theorem). *Let $\mathcal{M}$ be a context. Then for any (h, w) there is a context $\mathcal{N} \in \mathfrak{M}^{Act(\mathcal{M})}_{(h,w)}$ such that $\mathcal{M} \approx^w_h \mathcal{N}$.*

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 $N = \{P_{(h,w)} \mid P \in \mathcal{M}\}$. Moreover $[\![N]\!] \leq (h, w)$ and, by theorem 4.10, $Act(\mathcal{M}_{(h,w)}) \subseteq Act(\mathcal{M})$. Hence $\mathcal{N} \in \mathfrak{M}^{Act(\mathcal{M})}_{(h,w)}$. □

4.3 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 [4], 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 [4], but to our knowledge it has not been proved. Also new is the Hilbert-style approach to spatial logics.

4.4 Syntax of Dynamic Spatial Logic

Definition 4.8 (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 4.9 (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. $\langle!\alpha\rangle\psi \stackrel{def}{=} (\langle\alpha\rangle\psi) \wedge 1$

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*".[9]

$\top$ 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 ϕ and the other satisfying ψ.

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 α thus meeting a state described by ϕ"*.

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

The dynamic-like operator $[\alpha]\phi$, the dual operator of $\langle\alpha\rangle\phi$, expresses the situation where either the system cannot perform α, or if the system can perform α then any future state that can be reached by performing α can be described by ϕ.

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 α, which, after consuming α, will satisfy ψ.

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

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

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. *Consider the formula*

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

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

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

4.5 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 4.10 (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)

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

Definition 4.11. 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 4.1. ϕ is satisfiable iff $\neg\phi$ is not a validity, and vice versa, ϕ is a validity iff $\neg\phi$ is not satisfiable.

4.6 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 ϕ 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 ϕ, we can construct a finite class C_ϕ of couples $(\mathcal{M}, P)$ (where $\mathcal{M}$ is a context and $P \in \mathcal{M}$), depending on ϕ, such that if ϕ 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_ϕ is finite, for deciding if ϕ 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_ϕ 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 ϕ. Deeper ϕ is not "sensitive". In other words, we can derive from the structure of ϕ 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 $\mathcal{N}, Q \models \phi$. Similarly the satisfiability relation does not perceive information involving other actions but those that appear in the syntax of the logical formula. Hence if σ is a substitution that replaces an action α by β and both α, β did not appear in the syntax of ϕ, then $\mathcal{M}, P \models \phi$ iff $\mathcal{M}^\sigma, P^\sigma \models \phi$.

Further we will prove that these intuitions are correct, and we will identify C_ϕ.

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

Definition 4.12 (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 $(\!|\phi|\!) = (h, w)$

Definition 4.13 (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 4.13. *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 ϕ.

- **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, 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 $P \approx_{max(h_1,h_2)}^{w_1+w_2} Q$, using theorem 3.5, we obtain $Q \equiv Q_1|Q_2$ with $P_i \approx_{max(h_1,h_2)}^{w_i} Q_i$ for $i = 1, 2$. Further, using theorem 3.3 we obtain $P_i \approx_{h_i}^{w_i} Q_i$.
  Now $\mathcal{M}, P_i \models \phi_i$ and $P_i \approx_{h_i}^{w_i} Q_i$ 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 $P \approx_{1+h}^{1+w} Q$, using theorem 3.10, we obtain that $Q \xrightarrow{\alpha} Q'$ and $P' \approx_h^w Q'$.
  Now from $\mathcal{M}, P' \models \phi'$ and $P' \approx_h^w Q'$, 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 4.13 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.

Theorem 4.14. *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, 3.11, exists the process $P_{(h,w)}$ with $P \approx_h^w P_{(h,w)}$. Then, using lemma 4.13, 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 $\mathcal{M}_{(h,w)}, P_{(h,w)} \models \phi$. □

Definition 4.14 (The set of actions of a formula). We define the set of actions of a formula ϕ, $act(\phi) \subset \mathbb{A}$, inductively by:

1. $act(0) \stackrel{def}{=} \emptyset$
2. $act(\top) \stackrel{def}{=} \emptyset$
3. $act(\phi \wedge \psi) = act(\phi|\psi) \stackrel{def}{=} act(\phi) \cup act(\psi)$
4. $act(\neg\phi) = act(\phi)$
5. $act(K_R\phi) \stackrel{def}{=} Act(R) \cup act(\phi)$
6. $act(\langle\alpha\rangle\phi) \stackrel{def}{=} \{\alpha\} \cup act(\phi)$

The next result states that a formula ϕ does not reflect properties that involves more then the actions in its syntax. Thus if $\mathcal{M}, P \models \phi$ then any substitution σ having the elements of $act(\phi)$ as fix points preserves the satisfaction relation, i.e. $\mathcal{M}^\sigma, P^\sigma \models \phi$.

Theorem 4.15. *If $\mathcal{M}, P \models \phi$ and σ is a substitution with $act(\sigma) \bigcap act(\phi) = \emptyset$ then $\mathcal{M}^\sigma, P^\sigma \models \phi$.*

Proof. We prove, simultaneously, by induction on ϕ, 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 $act(\sigma) \cap act(\phi) = \emptyset$ we derive that $act(\sigma) \cap act(\psi_1) = \emptyset$ and $act(\sigma) \cap act(\psi_2) = \emptyset$. Further, applying the inductive hypothesis, we obtain $\mathcal{M}^\sigma, P^\sigma \models \psi_1$ and $\mathcal{M}^\sigma, P^\sigma \models \psi_2$ that implies $\mathcal{M}^\sigma, P^\sigma \models \psi_1 \wedge \psi_2$.
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, $act(\sigma) \cap act(\psi_1) = \emptyset$ and $act(\sigma) \cap act(\psi_2) = \emptyset$, hence we can apply the inductive hypothesis that entails $\mathcal{M}^\sigma, P^\sigma \not\models \psi_1$ or $\mathcal{M}^\sigma, P^\sigma \not\models \psi_2$. Thus $\mathcal{M}^\sigma, P^\sigma \not\models \psi_1 \wedge \psi_2$.

The case $\phi = \neg\psi$:

1. $\mathcal{M}, P \models \neg\psi$ is equivalent with $\mathcal{M}, P \not\models \psi$ and because $act(\sigma) \cap act(\phi) = \emptyset$ guarantees that $act(\sigma) \cap act(\psi) = \emptyset$, 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 $act(\sigma) \cap act(\phi) = \emptyset$ we have $act(\sigma) \cap act(\psi_1) = \emptyset$ and $act(\sigma) \cap act(\psi_2) = \emptyset$. 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 $act(\sigma) \cap act(\phi) = \emptyset$ guarantees that $act(\sigma) \cap act(\psi_1) = \emptyset$ and $act(\sigma) \cap act(\psi_2) = \emptyset$. 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 any arbitrary decomposition $\sigma(P) \equiv P'|P''$. By theorem 3.17, there exists $P \equiv Q|R$ such that $\sigma(Q) \equiv P'$ and $\sigma(R) \equiv P''$. Thus either $\sigma(\mathcal{M}), P' \not\models \psi_1$ or $\sigma(\mathcal{M}), P'' \not\models \psi_2$. Hence $\sigma(\mathcal{M}), \sigma(P) \not\models \psi_1|\psi_2$.

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 $act(\sigma) \cap act(\langle\gamma\rangle\psi) = \emptyset$ implies $act(\sigma) \cap act(\psi) = \emptyset$. 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 $\sigma(P) \equiv \sigma(\gamma).\sigma(P')|\sigma(P'')$. Now $act(\sigma) \cap act(\langle\gamma\rangle\psi) = \emptyset$ ensures that $\sigma(\gamma) = \gamma$. So $\sigma(P) \equiv \gamma.\sigma(P')|\sigma(P'')$ 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 γ, 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 γ 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 $\gamma \neq \alpha_i$, and $\gamma \notin act(\sigma)$, we can state that $\gamma \neq \sigma(\alpha_i)$, hence $\sigma(P)$ cannot perform a transition by γ. Thus $\sigma(\mathcal{M}), \sigma(P) \not\models \langle\gamma\rangle\psi$.
   If there are transitions of P by γ, and for any such a transition $P \xrightarrow{\gamma} Q$ we have $\mathcal{M}, Q \not\models \psi$: then, because from $act(\sigma) \cap act(\langle\gamma\rangle\psi) = \emptyset$ we can derive $act(\sigma) \cap act(\psi) = \emptyset$, the inductive hypothesis can be applied and we obtain $\sigma(\mathcal{M}), \sigma(Q) \not\models \psi$. But because $\gamma \notin act(\sigma)$ we obtain $\sigma(\gamma) = \gamma$ and $\sigma(P) \xrightarrow{\gamma} \sigma(Q)$. Hence $\sigma(\mathcal{M}), \sigma(P) \not\models \langle\gamma\rangle\psi$.

□

We suppose to have defined on $\mathbb{A}$ a lexicographical order $\ll$. So, for a finite set $A \subset \mathbb{A}$ we can identify a maximal element that is unique. Hence the successor of this element is unique as well. We convey to denote by A_+ the set obtained by adding to A the successor of its maximal element.

Theorem 4.16 (Finite model property).

$$\textit{If } \mathcal{M}, P \models \phi \textit{ then } \exists \mathcal{N} \in \mathfrak{M}_{(\!|\phi|\!)}^{act(\phi)_+} \textit{ and } Q \in \mathcal{N} \textit{ such that } \mathcal{N}, Q \models \phi$$

Proof. Consider the substitution σ that maps all the actions $\alpha \in \mathbb{A} \setminus act(\phi)$ in the successor of the maximum element of $act(\phi)$ (it exists as $act(\phi)$ is finite). Obviously $act(\sigma) \cap act(\phi) = \emptyset$, hence, using theorem 4.15 we obtain $\mathcal{M}^\sigma, P^\sigma \models \phi$. Further we take $\mathcal{N} = \mathcal{M}^\sigma_{(h,w)} \in \mathfrak{M}^{act(\phi)^+}_{(h,w)}$ and $Q = P^\sigma_{(h,w)} \in \mathcal{M}^{act(\phi)^+}_{(h,w)}$, and theorem 4.13 proves the finite model property. □

Because $act(\phi)$ is finite implying $act(\phi)_+$ finite, we apply theorem 4.11 ensuring that $\mathfrak{M}_{(\!|\phi|\!)}^{act(\phi)_+}$ is finite and any context $\mathcal{M} \in \mathfrak{M}_{(\!|\phi|\!)}^{act(\phi)_+}$ is finite as well. Thus we obtain the finite model property for our logic.

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 ϕ as input, decides, in a finite manner, if there exists a process satisfying it in a context; indeed, this procedure may construct C_ϕ in the manner presented before and then browsing it to find such a model - the searching is finite because C_ϕ is finite.
- Validity checking is decidable, because ϕ 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 ϕ then $\mathcal{M}, P \models \phi$ is equivalent with $\mathcal{M}_{(\!|\phi|\!)}, P_{(\!|\phi|\!)} \models \phi$ that requires a finite verification.

Theorem 4.17 (Decidability of $\mathcal{L}_{DS}$). *For $\mathcal{L}_{DS}$ against process semantics, satisfiability, validity and model checking are decidable.*

4.7 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 4.15 (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. $\vdash \top|\bot \to \bot$

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

Axiom D 3. $\vdash \phi|\psi \to \psi|\phi$

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

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

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

Spatial rules

Rule $\mathbf{D}_R$ 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 $\mathbf{D}_R$ 4. *If* $\vdash \bigvee_{Q \in \mathfrak{P}_{(\!|\phi|\!)}^{act(\phi)_+}} c_Q \rightarrow \phi$ *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 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 α, 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, α_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 α, axiom D11 states that a system described by a guarded process can perform one and only one action, the guarding one.

Rule D_R2 is the classic necessity rule used for the dynamic operator.

Rule 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 $\mathfrak{P}_{(\!|\phi|\!)}^{act(\phi)_+}$ is finite (modulo $\equiv$).

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 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. □

5.1 Soundness of the spatial axioms and rules

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

Lemma 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 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 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 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 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 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 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 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 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 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 5.10 (Soundness of rule $\mathbf{D}_R\mathbf{1}$).

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

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, $\mathcal{M}, Q \models \phi \rightarrow \psi$, hence $\mathcal{M}, Q \models \psi$. Then $\mathcal{M}, P \models \psi|\rho$, so $\models \phi|\rho \rightarrow \psi|\rho$. □

5.2 Soundness of the dynamic axioms and rules

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

Lemma 5.11 (Soundness of axiom D7). $\models \langle\alpha\rangle\phi|\psi \rightarrow \langle\alpha\rangle(\phi|\psi)$.

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 5.12 (Soundness of axiom D8).

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

Proof. Let $\mathcal{M}, P \models [\alpha](\phi \rightarrow \psi)$ 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 $\mathcal{M}, P' \models \phi \rightarrow \psi$ and $\mathcal{M}, P' \models \phi$. Hence $\mathcal{M}, P' \models \psi$, i.e. $\mathcal{M}, P \models [\alpha]\psi$. □

Lemma 5.13 (Soundness of axiom D9). $\models 0 \rightarrow [\alpha]\bot$

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 5.14 (Soundness of axiom D10).

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

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 5.15 (Soundness of axiom D11). $\models \langle !\alpha\rangle\phi \rightarrow [\alpha]\phi$

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 5.16 (Soundness of rule $\mathbf{D}_R\mathbf{2}$). *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 5.17 (Soundness of rule $\mathbf{D}_R\mathbf{3}$).

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

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 $\models \phi \rightarrow [\alpha]\phi'$ and $\models \psi \rightarrow [\alpha]\psi'$, 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 α, then $\mathcal{M}, P \models [\alpha]\bot$, and using the soundness of axiom D8 and rule D_R2 we derive

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

  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 α, then $Q|R \xrightarrow{\alpha} Q'|R$ and the transitions of $P \equiv Q|R$ by α 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 α 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 5.18 (Soundness of rule $\mathbf{D}_R$4).

$$\textit{If } \models \bigvee_{Q \in \mathfrak{P}_{(\!|\phi|\!)}^{act(\phi)_+}} c_Q \to \phi \textit{ then } \models \phi$$

Proof. Suppose that $\models \bigvee_{Q \in \mathfrak{P}_{(\!|\phi|\!)}^{act(\phi)_+}} c_Q \to \phi$ 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 4.16, we obtain that $\mathcal{M}_{(\!|\neg\phi|\!)}^{act(\neg\phi)_+}, P_{(\!|\neg\phi|\!)}^{act(\neg\phi)_+} \models \neg\phi$. But $(\!|\phi|\!) = (\!|\neg\phi|\!)$ and $act(\phi) = act(\neg\phi)$, so $\mathcal{M}_{(\!|\phi|\!)}^{act(\phi)_+}, P_{(\!|\phi|\!)}^{act(\phi)_+} \models \neg\phi$.
Further, because $\models \bigvee_{Q \in \mathfrak{P}_{(\!|\phi|\!)}^{act(\phi)_+}} c_Q \to \phi$, we have

$$\mathcal{M}_{(\!|\phi|\!)}^{act(\phi)_+}, P_{(\!|\phi|\!)}^{act(\phi)_+} \models \bigvee_{Q \in \mathfrak{P}_{(\!|\phi|\!)}^{act(\phi)_+}} c_Q \to \phi$$

But $\mathcal{M}_{(\!|\phi|\!)}^{act(\phi)_+}, P_{(\!|\phi|\!)}^{act(\phi)_+} \models c_{P_{(\!|\phi|\!)}^{act(\phi)_+}}$ and we obtain

$$\mathcal{M}_{(\!|\phi|\!)}^{act(\phi)_+}, P_{(\!|\phi|\!)}^{act(\phi)_+} \models \bigvee_{Q \in \mathfrak{P}_{(\!|\phi|\!)}^{act(\phi)_+}} c_Q$$

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

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.

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 6.1. $\vdash \top|\top \leftrightarrow \top$

Proof. Obviously $\vdash \top|\top \to \top$. As $\vdash 0 \to \top$, using rule D_R1, we obtain $\vdash \top|0 \to \top|\top$. Further axiom D2 gives us $\vdash \top \to \top|\top$. □

Theorem 6.2. *If* $\vdash \phi$ *then* $\vdash \theta|\rho \to \phi|\rho$

Proof. Because $\vdash \phi$ implies $\vdash \theta \to \phi$, using rule D_R1 we obtain the result. □

Theorem 6.3. $\vdash \phi|\psi \leftrightarrow \psi|\phi$

Proof. We use axiom D3 in both directions. □

Theorem 6.4. $\vdash (\phi|\psi)|\rho \leftrightarrow \phi|(\psi|\rho)$

Proof. We use axiom D4 and theorem 6.3. □

Theorem 6.5. $\vdash \phi|(\psi \vee \rho) \leftrightarrow (\phi|\psi) \vee (\phi|\rho)$

Proof. $\vdash \psi \to \psi \vee \rho$ so, using rule D_R1, $\vdash \phi|\psi \to \phi|(\psi \vee \rho)$. Similarly, $\vdash \phi|\rho \to \phi|(\psi \vee \rho)$. Hence $\vdash (\phi|\psi) \vee (\phi|\rho) \to \phi|(\psi \vee \rho)$. The other direction is stated by axiom D5. □

Theorem 6.6. $\vdash \phi|(\psi \wedge \rho) \to (\phi|\psi) \wedge (\phi|\rho)$

Proof. Because $\vdash \psi \wedge \rho \to \psi$, by applying rule D_R1, we have $\vdash \phi|(\psi \wedge \rho) \to \phi|\psi$. Similarly $\vdash \phi|(\psi \wedge \rho) \to \phi|\rho$. □

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

Theorem 6.7. *If* $\vdash \phi \to \rho$ *and* $\vdash \psi \to \theta$ *then* $\vdash \phi|\psi \to \rho|\theta$.

Proof. If $\vdash \phi \to \rho$ then rule D_R1 gives us $\vdash \phi|\psi \to \rho|\psi$. If $\vdash \psi \to \theta$, then the same rule gives $\vdash \rho|\psi \to \rho|\theta$. Hence $\vdash \phi|\psi \to \rho|\theta$. □

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

Theorem 6.8. *If for any decomposition* $P \equiv Q|R$ *we have* $\vdash c_Q \to \neg\phi$ *or* $\vdash c_R \to \neg\psi$ *then* $\vdash c_P \to \neg(\phi|\psi)$.

Proof. $\vdash c_Q \to \neg\phi$ is equivalent with $\vdash c_Q \wedge \phi \to \bot$ and because $\vdash c_R \wedge \psi \to \top$, we obtain, by theorem 6.7 $\vdash (c_Q \wedge \phi)|(c_R \wedge \psi) \to \bot|\top$. And using axiom D1, we derive

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

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

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

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

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

But axiom D6 gives

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

hence

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

□

Remark 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 $\mathcal{M}, P \models \phi|\psi$.

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 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 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 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 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 6.11. *If P cannot do any transition by α 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 α 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 6.9, $\vdash (\langle\alpha_i\rangle c_{P_i} \wedge 1) \to (\langle\alpha_i\rangle\top \wedge 1)$. Further, we apply theorem 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 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 α, hence, by theorem 6.11, $\vdash c_P \to [\alpha]\bot$. But
$\vdash \neg\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \to \top$, and using theorem 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 $\vdash c_P \to [\alpha]c_{P'}$. Hence

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

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

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

and

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

We apply rule D_R3 and obtain

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

Using theorem 6.5, we obtain this result equivalent with

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

□

Theorem 6.13. *If $\vdash \bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \to \phi$ then $\vdash c_P \to [\alpha]\phi$*

Proof. If $\vdash \bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \rightarrow \phi$ then rule 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 6.12 gives $\vdash c_P \rightarrow [\alpha]\bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\}$, hence $\vdash c_P \rightarrow [\alpha]\phi$. □

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 7.1. $\mathcal{M}, P \models c_P$.

Proof. It has been proved as theorem 5.7. □

Theorem 7.2. $\mathcal{M}, P \models c_Q$ *iff* $P \equiv Q$.

Proof. It has been proved as theorem 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 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 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 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 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 $\vdash c_P \to 1 \wedge \langle\alpha\rangle\top$.
  But axiom D10 gives $\vdash \langle\alpha\rangle\top \wedge 1 \to \neg\langle\beta\rangle\top$ for any $\beta \neq \alpha$.
  Hence, for any $\beta \neq \alpha$ we have $\vdash c_P \to \neg\langle\beta\rangle\top$.
  - if $Q \equiv 0$ we already proved that $\vdash c_Q \to \neg c_P$ (because $P \not\equiv 0$), so $\vdash c_P \to \neg c_Q$
  - if $Q \equiv \beta.Q'|Q''$ for some $\beta \neq \alpha$, then $\vdash c_Q \to \langle\beta\rangle\top$, hence $\vdash \neg\langle\beta\rangle\top \to \neg c_Q$. But we proved that $\vdash c_P \to \neg\langle\beta\rangle\top$. Hence $\vdash c_P \to \neg c_Q$.
  - if $Q \equiv \alpha.Q_1|...|\alpha.Q_k$ for $k > 1$, then $\vdash c_Q \to \neg 0|\neg 0$ (as $\vdash 0 \to \neg c_{\alpha.Q_1}$ and $\vdash 0 \to \neg c_{\alpha.Q_2|...|\alpha.Q_k}$). Then $\vdash c_Q \to \neg 1$, i.e.
    $\vdash 1 \to \neg c_Q$. But $\vdash c_P \to 1$. Hence $\vdash c_P \to \neg c_Q$.
  - 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 $\vdash c_{Q'} \to \neg c_{P'}$. Further, applying theorem 6.10, we obtain
    $\vdash [\alpha]c_{P'} \to [\alpha]\neg c'_Q$, i.e.
    $\vdash [\alpha]c_{P'} \to \neg\langle\alpha\rangle c_{Q'}$ that gives, because $c_Q = 1 \wedge \langle\alpha\rangle c_{Q'}$,
    $\vdash [\alpha]c_{P'} \to \neg c_Q$.
    Now, using axiom D11, $\vdash 1 \wedge \langle\alpha\rangle c_{P'} \to [\alpha]c_{P'}$, so $\vdash c_P \to [\alpha]c_{P'}$, and, combining it with the previous result, we derive $\vdash c_P \to \neg c_Q$.

□

Theorem 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 6.3, i.e. $\vdash c_P \leftrightarrow c_Q$
- if $P = (R|S)|U$ and $Q = R|(S|U)$ then theorem 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 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 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 ϕ. 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 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 ϕ.

- **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 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 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 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 ψ.
  - **the subcase** $\psi = 0$**:** $\mathcal{M}, P \models \neg 0$ means that $P \not\equiv 0$, and using theorem 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 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 α, then, by theorem 6.11 $\vdash c_P \to [\alpha]\bot$ that implies further $\vdash c_P \to [\alpha]\neg\psi_1$ (because $\vdash \psi_1 \to \top$).
    If P can perform α, 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 $\vdash c_Q \to \neg\psi_1$, i.e.
    $\vdash \bigvee\{c_Q \mid P \xrightarrow{\alpha} Q\} \to \neg\psi_1$.
    Now, using theorem 6.13, we obtain $\vdash c_P \to [\alpha]\neg\psi_1$ q.e.d.

($\Longleftarrow$) Let $\vdash c_P \to \phi$ 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 $\vdash c_P \to \neg\phi$, hence $\vdash c_P \to \bot$.
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 [4], 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 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 7.5 we obtain $\vdash c_P \to \phi$. As $P \in \mathcal{M}'$ and $\vdash c_P \to \phi$ we can apply the same lemma once again and obtain $\mathcal{M}', P \models \phi$ q.e.d. □

Corollary 7.7. *If $\vdash c_P \to c_Q$ then $P \equiv Q$.*

Proof. If $\vdash c_P \to c_Q$ then, by lemma 7.5, $\mathcal{M}, P \models c_Q$, for any context $\mathcal{M} \ni P$, hence, by theorem 5.8 $P \equiv Q$. □

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 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 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 ϕ 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 8.1. *If ϕ 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 ϕ 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 7.5 gives $\vdash c_P \rightarrow \neg\phi$, for any $P \in \mathfrak{P}$. Thus we also have $\vdash c_P \rightarrow \neg\phi$ for any $P \in \mathfrak{P}^{act(\neg\phi)_+}_{(\!|\neg\phi|\!)}$, i.e.

$$\vdash \bigvee\nolimits_{P \in \mathfrak{P}^{act(\neg\phi)_+}_{(\!|\neg\phi|\!)}} c_P \rightarrow \neg\phi.$$

Further, using rule D_R4, we obtain $\vdash \neg\phi$. This contradicts with the hypothesis of $\mathcal{L}_{DS}$-consistency of ϕ.

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

Theorem 8.2 (Completeness). *The $\mathcal{L}_{DS}$ system is complete with respect to process semantics.*

Proof. Suppose that ϕ is a valid formula with respect to process semantics, but ϕ 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 8.1 that $\neg\phi$ is satisfiable with respect to our semantics, contradicting the validity of ϕ.
Hence, if ϕ is valid, then it is $\mathcal{L}_{DS}$-provable. □

Acknowledgements. We thank to Alexandru Baltag for contributing with valuable comments, since the beginning, on the construction of this logic. Thanks also to Luca Cardelli for comments and related discussions. The name *structural bisimulation* was suggested to us by Gordon Plotkin.

References

[1] J. A. Bergstra. *Handbook of Process Algebra*. Elsevier Science Inc., New York, NY, USA, 2001.

[2] 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.

[3] Luis Caires and Luca Cardelli. A spatial logic for concurrency (part i). *Information and Computation*, Vol: 186/2:194–235, November 2003.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Luca Cardelli and Andrew D. Gordon. Ambient logic. *To appear in Mathematical Structures in Computer Science*, 2003.

[9] M. Dam. Proof systems for π-calculus. *In de Queiroz, editor, Logic for Concurrency and Synchronisation, Studies in Logic and Computation. Oxford University Press. To appear.*

[10] 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.

[11] M. Dam. Model checking mobile processes. *Information and Computation*, vol:129(1):35–51, 1996.

[12] E. A. Emerson. *Temporal and Modal Logic. Handbook of Theoretical Computer Science*, volume B: Formal Models and Sematics. 1990.

[13] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *JACM*, vol: 32(1):137–161, 1985.

[14] R. Milner, J. Parrow, and D. Walker. Modal logics for mobile processes. *Theoretical Computer Science*, vol:114:149–171, 1993.

[15] 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.

[16] Colin Stirling. *Modal and temporal properties of processes.* Springer-Verlag New York, Inc., New York, NY, USA, 2001.