UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

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

THE PRINCIPLES OF AMBIENT CALCULUS REVISITED

Radu Mardare and Corrado Priami

2005

Technical Report # DIT-05-018

The Principles of Ambient Calculus Revisited †

Radu Mardare and Corrado Priami,
University of Trento, Italy
Via Sommarive, 14 I-38050 Povo, Trento, Italy
$\{mardare, priami\}$@dit.unitn.it

"All science is the study of patterns.
Mathematics is the science of patterns of patterns."
(Jon Barwise)

Abstract

The paper presents a foundational analysis of the class of ambient calculi focussing on the spatial structures of the processes they can describe. We propose sound correct set-theoretical models for different types of ambient calculi (involving recursion, denumerable parallel composition, etc) by using special types of coalgebras - labelled flat systems of equations of set theory. These models help to understand the spatiality of the ambient processes and provide a set-theoretical description of the structural congruence. Consistently with the model, we extend the classical structural congruence, the extension proving that $P|P|P... \equiv (recX.X|P) \equiv (recX.X|X|...|X|P)$, if X does not appear free in P. In Ambient Calculus the space has a decisive influence in the behavior of a process. Consequently, replication cannot simulate recursion because the processes involving replication have always a finite or denumerable class of active actions, while there are recursive ambient processes having infinite non-denumerable active actions due to a possibly non-wellfounded spatial structure. Still recursion can successfully describe any process involving replication. Thus we propose the most expressive Recursive Ambient Calculus, interpreting replication as recursion. Using the set-theoretical model for this calculus, we construct a propositional temporal logic on top of it able to describe properties of the processes. Our logic works similarly with wellfounded and non-wellfounded processes.

Key words: ambient processes, hypersets, coalgebra, temporal logics.

1 Introduction

The study of communication and mobility has a central role in understanding and operating many realities of a particular interest for today's sciences, from the World-Wide Web to the biological systems. In the direction opened by CCS [15] and CSP [6] some formalisms have been developed to provide suitable formal tools for modelling and analyzing phenomena involving communication and mobility. Interpreting diverse realities like computational environments generates more and more complex paradigms of understanding the space of communication and mobility. In the case of CCS [15] - π-Calculus [16] type of calculi the state of a system is expressed by a flat network of processes connected by *channels*. Instead, in Ambient Calculus [10] type of calculi the state of a system is organized by a *boxes inside boxes* type of structure, each box (ambient) representing a closed spatial collection of running processes and sub-boxes (subambients). Thus the ambient nesting relation generates a tree structure for each state of the ambient-based system, the evolution of the system being just a reorganization in this tree. The π-calculus class of calculi works with a communication space of Leibnizian type: it does not exist "per se", but it is generated by the relative positions of the processes (defined by the channels that connect them). In this paradigm the components of the system are just entities able to exchange information. The communication space after ambient class of calculi

†Work partially supported by the FET project IST-2001-32072 DEGAS under the pro-active initiative on Global Computing.

is of Newtonian type: the space exists independently of the processes as a precondition of their existence. The ambient processes are in space and manifest themselves through this space; they have positions in space and perform movements referring to it. They are not just connected entities, but entities spatially organized, and the spatial positions are the connected ones.

We make here a foundational analysis of ambient type of calculi focussing on the spatial structures of the processes they can describe. Many ambient calculi were developed to model various phenomena and most of them handle the same type of structures, differing only in defining the basic computations with respect to the peculiarities of the phenomena they intend to model. Little attention has been paid to analyze the complexity of the background spatial structures of the processes that such calculi can describe. We analyze the diversity of definable ambient calculi (involving recursion, replication, denumerable parallel composition) from this point of view. We focus on the structures these calculi can describe as models. Special attention is paid to the recursively definable models in connection with different types of ambient calculi, and we try to establish a hierarchy of these calculi from the point of view of expressivity. It is widely believed, for example, that in the Ambient Calculus, as in the π-Calculus, replication is expressive enough to describe any structure that could be described by using recursivity. We will prove, as a consequence of our analysis, that in Ambient Calculus this is not the case and that replication can describe and simulate only a special class of recursive ambient processes, but not all of them.

Our approach is inspired by the work of Peter Aczel [3] and Jon Barwise [5] done in coalgebraical semantics in relation with the calculi for communicating systems and logic. The motivation of Aczel's work, in [3], was to provide a set-theoretical model for Milner's calculus of communicating systems (CCS) [15] exploiting some results in Set Theory without foundation[1]. Barwise [5] presents Aczel's work as originating the concept of *hyperset*. Lately, hypersets were deeply studied, independently and in connection with logic [5, 4] and Mu-Calculus [14]. We reuse these results to fully understand the spatial structures of the ambient processes and their role in the economy of each type of ambient calculus[2].

Resorting to the concept of *hyperset*, we develop a special class of coalgebras - a class of labelled flat systems of equations, named *Ambient Hierarchies* - that generates correct sound models for any ambient calculus. In this way we identify the set theoretical structure of any possible model for each type of calculus. For example the calculus in its classical form [10] has a dual nature concerning the possible structures that can be modelled using replication. We can define replication by $!P \stackrel{def}{=} (recX.X|P)$, with X not free in P, as in [11], but in this case we loose two important properties of replication: $!(P|Q) \equiv !P|!Q$ and $!!P \equiv !P$. Alternatively, we can define replication as a denumerable parallel composition $!P \stackrel{def}{=} P|P|P|...$. If we use the first definition, then ambient processes describe any hereditarily finite set theoretical structure (possibly non-wellfounded) but not the cardinal infinite ones, while the second definition allows to avoid the non-wellfounded structures, but imposes structures with infinite cardinality.

We also analyze the possible extensions to the classical ambient calculus in order to get the maximum expressivity in the context of a finite syntax, and we identify a class of regular recursive processes having this property. We prove that this Recursive Ambient Calculus is strictly more expressive than the classical Ambient Calculus, but also than the other types of recursive ambient calculi proposed in the literature [11, 17]. In [11] such a calculus is studied under the restriction that processes of the type $(recX.X|X)$ are ill-formed, while in [17] the ambients with unbounded depth, like $(recX.m[X])$, are disregarded. Our calculus does not disregard any of these.

Furthermore, the set theoretical model gives a way to extend the structural congruence of classical ambients to the recursive ones by using the patterns of structural unfoldings[3] of an infinite process[4] understood as an infinite tree. This congruence relates classical processes with recursive ones. Such a relation has not been proposed before. The other recursive calculi studied in the literature do not propose any structural congruence rule able to relate a process

[1]Studies in this direction were firstly developed by F. Honsell and M. Forti [13].

[2]We use the syntagm *type of calculus* meaning the type of the algebraical structure of the calculus.

[3]This is a tool used in studying infinite structures, see for example, [4].

[4]By an infinite process we do not refer to a process that can evolve forever, but to a process having an infinite spatial structure, such as $P_1|P_2|P_3|...$ or as $n[n[n[...]]]$

involving recursion with one which do not involve recursion. Or it is widely believed that, for example, $!P = P|P|P|...$ and $(recX.X|P)$ (if X does not appear free in P) have equivalent structures. Using our relation, this equivalence can be proved. Moreover, we also prove that $(recX.X|P|Q) \equiv (recX.X|P)|(recX.X|Q)$ (if X does not occur free in P and Q) and that $(recX.X|(recY.Y|P)) \equiv (recX.X|P)$ (if X, Y do not occur free in P), so that processes involving replication can be successfully described by recursion.

Finally, we prove that the classical Ambient Calculus with replication is strictly less expressive than our Recursive Ambient Calculus: any structure that can be described by a classical ambient process (possibly including replication) can be described by a recursive ambient process, and not vice verse. We prove that replication can describe only a special class of recursive processes while processes like $(recX.m[X])$ have a structure that cannot be modelled by replication. Moreover, we prove that, Milner's conversion from recursion to replication proposed in [16] for π-Calculus can work only with a Leibnizian space of location and not in a Newtonian one. We prove that no method of simulating recursion by replication is possible in ambient calculi because there exist recursive processes having infinite non-denumerable active actions, while the classical processes have always finite or denumerable active actions. Ambient Calculus has a space behind processes which could have unbounded depth, and this property has decisive consequences over the space of computations.

The extension of the structural equivalence proposed here generalizes, via Ambient Hierarchies, the set theoretical bisimulation proposed in [5]. Hence, our approach provides a set theoretical description of the structural congruence.

The most important achievement of this paper is that, using the set theoretical model of our Recursive Ambient Calculus, we are able to construct a simple propositional temporal logic, CTL^*, on top of it able to express properties of processes. We propose this logic as an alternative to the Spatial Logics developed for calculi with locations [7, 9, 8]. We sustain it for its simplicity (we manage the input and the private names on the model level such that no additional operators are required on the logical level), for its expressivity (we can express properties of recursive processes), but also because the temporal logics are well-known and deeply studied [12, 1, 2].

The paper is organized as follows. First we present the basic intuitions that motivate this work, pointing to the major role of the space in the behavior of an ambient process. In section 3 we present the classical Ambient Calculus, as it was introduced in [10], and we analyze some possible extensions of it by adding denumerable parallel composition or recursion, all these extensions being sound with possible interpretations of replication. In the end of the section we identify the algebraical structure for each possible calculus. In section 4 we develop the main tools used to analyze the structures of ambient processes. We present the system ZFA of set theory (Zermelo-Fraenkel with Anti-Foundation Axiom) together with the notion of flat system of equations. The flat systems of equations are special types of coalgebras that describe a structure (possible non-wellfounded) as a system of equations of set theory. Between the flat systems it can be defined a bisimulation relation that acts as the identity for wellfounded structures. In section 5 we introduce the Ambient-Graphs, a special type of flat systems labelled by names and sequences of capabilities imported from ambient calculus. The Ambient-Graphs describe each ambient process revealing its spatial structure. The bisimulation relation between structures defines equivalence classes of Ambient-Graphs - the Ambient-Hierarchies. We prove that special classes of Ambient-Hierarchies provide sound correct models for different ambient calculi. In order to prove this we introduce, in section 6, an algebra of Ambient-Hierarchies able to simulate compositionality, while section 7 adds to the algebra an equivalence relation that will be proved to be a sound correct model for the structural congruence. In section 8 we prove that our models are, indeed, sound correct models for the analyzed types of ambient calculi, by proving that the algebraical structure of each one is isomorphic with the algebraical structure of the corresponding ambient calculus. One of the major points in this paper is presented in section 9. We propose the most expressive recursive calculus that can be defined by using a finite syntax. In the context of this calculus we propose an extension of the structural congruence and we identify recursive processes that cannot be described or simulate by replication. Section 10 enriches the set theoretical model to became a model for the recursive calculus with extended structural congruence. In the section 11 we construct a CTL^* logic in top of our Recursive Ambient Calculus to express properties of the system. Our logic can express also properties of non-wellfounded processes with the same accuracy and simplicity as in the wellfounded case. Eventually, we conclude the paper with final remarks. We added an appendix with the major points in the proofs of the non-trivial results of the paper.

2 The basic intuitions

In this section we propose a few examples to sketch the main intuitions that motivated our work.

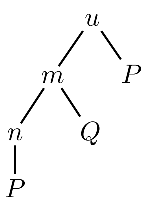
2.1 The space of locations

Consider the ambient process:

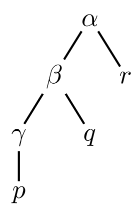
$$u[(\nu n)m[n[open\ m.P]|open\ n.Q]|open\ m.P] \tag{2.1}$$

If we take the occurrences of the ambients (in this case u, m, n) and of the unspecified processes (in this case P and Q) of an ambient process as vertices of a tree, and the ambient nesting relation as defining edges, we could univocally associate to each ambient process a tree. We call this tree *the structure tree associated with the process.* For our example the structural tree is described in fig.1.

$fig.1$



$fig.2$



Such a tree depicts the structural hierarchy of locations for any ambient process. Observe that P has two positions, one inside the ambient u and an other inside n. Hence, if we try to describe this tree by its set of relations, in the table T1, we are faced with undesirable consequences.

$$T1: \begin{cases} m \in u \\ P \in u \\ n \in m \\ Q \in m \\ P \in n \end{cases} \qquad T2: \begin{cases} \beta \in \alpha & \mathcal{L}(\alpha) = u \\ & \mathcal{L}(\beta) = m \\ r \in \alpha & \mathcal{L}(r) = P \\ \gamma \in \beta & \mathcal{L}(\gamma) = n \\ q \in \beta & \mathcal{L}(q) = Q \\ p \in \gamma & \mathcal{L}(p) = P \end{cases} \qquad T3: \begin{cases} \beta \in \alpha & \mathcal{L}(\alpha) = u \\ & \mathcal{L}(\beta) = (\nu n)m \\ r \in \alpha & \mathcal{L}(r) = open\ m.P \\ \gamma \in \beta & \mathcal{L}(\gamma) = n \\ q \in \beta & \mathcal{L}(q) = open\ n.Q \\ p \in \gamma & \mathcal{L}(p) = open\ m.P \end{cases}$$

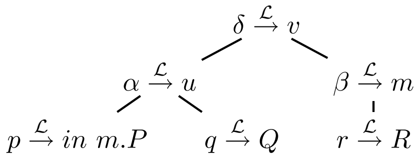
Such a system states that the spatial positions denoted by the ambients n and u are not disjunct (they share P), and this contradicts the basic intuition behind the syntax of Ambient Calculus. Thus the spatial positions refer not directly to processes as entities, but to the occurrences of the processes. Processes are in space but do not define the space, i.e. there are two distinct spatial positions where P can be. In this respect, it seems more correct to describe the spatiality of the ambient process by the tree in fig.2 enriched with a labelling function, $\mathcal{L}$, that associates each spatial position (here denoted by Greek letters) with the process that runs there. We obtain the system in the table T2. Furthermore, we can add also the prefixes to the processes by the same labelling function and we obtain the system in the table T3 that fully and unambiguously describes our ambient process.

In this way any ambient process can be univocally described on top of a set theoretical structure that depicts its spatiality. The systems constructed previously are known in the literature as *labelled flat systems of equations of set theory*, see for example [5]. Further the reductions can be described as a relation between certain types of labelled flat systems. Consider, for example, the next reduction that follows the (In-rule):

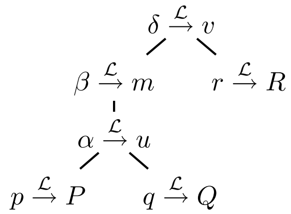
$$v[u[in\ m.P|Q]|m[R]] \longrightarrow v[m[u[P|Q]|R]] \tag{2.2}$$

The first ambient can be described by the tree[5] in fig.3 and the second by the tree in fig.4, and respectively the first by the system S_1 and the second by S_2:

$fig.3$



$fig.4$



$$S_1 : \begin{cases} \alpha \in \delta & \mathcal{L}(\delta) = v \\ & \mathcal{L}(\alpha) = u \\ \beta \in \delta & \mathcal{L}(\beta) = m \\ p \in \alpha & \mathcal{L}(p) = in\ m.P \\ q \in \alpha & \mathcal{L}(q) = Q \\ r \in \beta & \mathcal{L}(r) = R \end{cases} \qquad S_2 : \begin{cases} \beta \in \delta & \mathcal{L}(\delta) = v \\ & \mathcal{L}(\beta) = m \\ \alpha \in \beta & \mathcal{L}(\alpha) = u \\ q \in \alpha & \mathcal{L}(q) = Q \\ p \in \alpha & \mathcal{L}(p) = P \\ r \in \beta & \mathcal{L}(r) = R \end{cases}$$

We will provide a formal definition for these systems (called Ambient Hierarchies) and for their reductions. Then we will prove that, with these definitions, we obtain a sound correct model for Ambient Calculus. We will define also a special type of equivalence relation for them, on top of the classical definition of bisimilar set-theoretical structures, which become a model for the structural congruence of processes. In all this construction special attention will be paid to the behavior of the private names and input names. Hence, we describe any ambient process as a labelled spatial tree having each node labelled by the process which runs in that spatial position. Reductions, with respect to this intuition, are relations defined on the class of these trees. They are just laws defining consistent temporal paths, i.e. possible futures of a given process.

2.2 Temporal inferences

Recalling the ambient process 2.1, we could use the equations of the flat systems that describe the spatial structure of the process as atomical propositions in order to describe it. Taking the Table 3 that describes this process, we can say that the next logical formula describes the spatial architecture of the process:

$$(\beta \in \alpha) \wedge (r \in \alpha) \wedge (\gamma \in \beta) \wedge (q \in \beta) \wedge (p \in \gamma) \tag{2.3}$$

together with the labelling function $\mathcal{L}$. Now it is more clear why we need to name the spatial positions for describing the spatiality and why we did not directly use the processes in order to do this. Defining, in the same way as before, a logical proposition for describing the equations of the table T1 we obtain:

$$(m \in u) \wedge (P \in u) \wedge (n \in m) \wedge (Q \in m) \wedge (P \in n) \tag{2.4}$$

Now, if the processes would describe the spatiality, then if $P \in n$ is true, then $(P \in n) \wedge (P \in u)$ is inconsistent if n and u represents two disjoint positions, and such a situation is undesirable.

Consider now the reduction described by 2.2. The systems S_1 and S_2 can be described by:

$$S_1 : (\alpha \in \delta) \wedge (\beta \in \delta) \wedge (p \in \alpha) \wedge (q \in \alpha) \wedge (r \in \beta) \tag{2.5}$$

$$S_2 : (\alpha \in \beta) \wedge (\beta \in \delta) \wedge (p \in \alpha) \wedge (q \in \alpha) \wedge (r \in \beta) \tag{2.6}$$

Hence, the difference between the two ambient hierarchies is that the second one (the one obtained after *in m* was consumed) looses the prefix *in m* as label of p and changes the truth values of two atomical propositions with respect to the first hierarchy: $\alpha \in \delta$ becomes false

[5]We added, in the same tree, the vertices (the spatial positions) denoted by Greek letters and their labels, by using $\stackrel{\mathcal{L}}{\to}$ to denote the labelling relation.

(it was true in the first ambient hierarchy) and $\alpha \in \beta$ becomes true (it was false in the first one). In this way we can describe any possible reduction of an ambient process. Adding to these atomical propositions temporal operators we obtain a temporal propositional logic that can describe the future of a given ambient hierarchy, hence of a given ambient process.

2.3 A challenge: actual infinite processes versus potential infinite processes

In the context of the intuition described before, any ambient process can be seen as a labelled tree (labelled set), where the labelling function associates to each node a process. Hence any model of an ambient process has a structure of type "*boxes inside boxes*", i.e. a hyperset, see [5, 4]. This observation raises a question: can we have non-wellfounded structures as models for ambient processes?

Consider the ambient process[6]:

$$!m[!in\ m] \tag{2.7}$$

If we apply the reduction rules we obtain

$$
\begin{gathered}
!m[!in\ m] \equiv !m[!in\ m]|m[in\ m|!in\ m] \longrightarrow \\
!m[!in\ m]|m[!in\ m|m[!in\ m\ \underbrace{]]}_{2} \equiv !m[!in\ m]|m[!in\ m]|m[in\ m|!in\ m|m[!in\ m]] \longrightarrow \\
!m[!in\ m]|m[!in\ m|m[!in\ m|m[!in\ m\ \underbrace{]]]}_{3} \longrightarrow \\
!m[!in\ m]|m[!in\ m|m[!in\ m|m[!in\ m|m[!in\ m\ \underbrace{]]]]}_{4} \longrightarrow \\
!m[!in\ m]|m[!in\ m|m[!in\ m|m[!in\ m|m[!in\ m|m[!in\ m\ \underbrace{]]]]]}_{5} \longrightarrow \\
\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
!m[!in\ m]|m[!in\ m|m[!in\ m|m[!in\ m|m[!in\ m|m[!in\ m|\dots\underbrace{]]]]\dots]}_{k}
\end{gathered}
$$

Hence this process can produce unbounded depth processes, i.e. for each $k < \aleph_0$ we can produce a process having the ambient depth larger than k, where we denoted by $\aleph_0$ the first limit ordinal. This being proved can we sustain that the ambient 2.8 can be derived as well?

$$!m[!in\ m]|m[!in\ m|m[!in\ m|m[!in\ m|m[!in\ m|m[!in\ m|\dots\underbrace{]]]]\dots]}_{\aleph_0} \tag{2.8}$$

To obtain 2.8 from 2.7 we should apply the reduction rules denumerable number of times! Observe that 2.8 is not a wellformed ambient calculus formula for the reason that the syntax of our calculus doesn't allow infinite deep ambient nesting. Still, the structure 2.8 can be seen as the limit of the sequence of wellformed ambient processes that can be derived from 2.7. Producing unbounded depth processes is different of producing an infinite depth process. This distinction corresponds to the well-known distinction between the actual and potential infinite. The process 2.8 has an actual infinite structure, while the process 2.7 has only a potential infinite structure, i.e. taking it in its evolution it might have (in an infinite future) the structure 2.8. Still in Ambient Calculus actual infinite structures are accepted. Interpreted as a denumerable parallel composition, the replication has an actual infinite structure.

If we accept recursion in our calculus and we interpret $!P$ as $(recX.X|P)$, 2.7 can be rewritten as:

$$(recX.X|m[(recY.Y|in\ m)]) \tag{2.9}$$

while 2.8 can be rewritten as:

$$(recX.X|m[(recY.Y|in\ m)|X]) \tag{2.10}$$

[6] We thank to Luca Cardelli for proposing us this example

Should the structures as 2.9 be accepted as ambient processes while the structures as 2.10 are not considered "first-class citizens" in Ambient Calculus? What could motivate such a choice? Is this choice meaningful from the point of the set theoretical structure of the models such processes can describe? These questions motivate our interest for the subject of this paper.

3 Reconsidering Ambient Calculus

We briefly recall the Ambient Calculus starting with the syntax of ambient processes as it was introduced in [10]. Hereafter we refer to it as the classical Ambient Calculus in order to be distinguished by other ambient calculi we will discuss.

The syntax

$P, Q, R ::=$	**processes**	$M ::=$	**capabilities**
$(\nu n)P$	restriction	n	name
nil	void	$in\ M$	can enter into M
$P\|Q$	composition	$out\ M$	can exit out of M
$!P$	replication	$open\ M$	can open M
$M[P]$	ambient	$M.M'$	path
$M.P$	capability action	ε	null
$(n).P$	input action		
$\langle M \rangle$	output action		

The structural congruence

$(Refl)\ P \equiv P$
$(Symm)\ P \equiv Q \Rightarrow Q \equiv P$
$(Trans)\ P \equiv Q, Q \equiv R \Rightarrow P \equiv R$
$(Res)\ P \equiv Q \Rightarrow (\nu n)P \equiv (\nu n)Q$
$(Par)\ P \equiv Q \Rightarrow P|R \equiv Q|R$
$(Repl)\ P \equiv Q \Rightarrow !P \equiv !Q$
$(Amb)\ P \equiv Q \Rightarrow n[P] \equiv n[Q]$
$(ResAmb)\ (\nu n)(m[P]) \equiv m[(\nu n)P],\quad n \neq m$
$(ReplPar)\ !(P|Q) \equiv !P|!Q$
$(ReplNil)\ !nil \equiv nil$
$(ReplRepl)\ !!P \equiv !P$
$(ReplCopy)\ !P \equiv P|!P$
$(Act)\ P \equiv Q \Rightarrow M.P \equiv M.Q$
$(Inp)\ P \equiv Q \Rightarrow (n)P \equiv (n)Q$
$(Empty)\ P \equiv \varepsilon.P$
$(Struct)\ (M.M').P \equiv M.M'.P$
$(ResRes)\ (\nu n)(\nu m)P \equiv (\nu m)(\nu n)P$
$(ResNil)\ (\nu n)nil \equiv nil$
$(ResPar)\ (\nu n)(P|Q) \equiv P|(\nu n)Q, n \notin fn(P)$
$(ParNil)\ P|nil \equiv P$
$(Assoc)\ (P|Q)|R \equiv P|(Q|R)$
$(Comm)\ P|Q \equiv Q|P$
$(InpRen)\ (x).P \equiv (y).P(x \leftarrow y)$ if $y \notin fn(P)$
$(ResRen)\ (\nu n)P \equiv (\nu m)P(n \leftarrow m)$ if $m \notin fn(P)$

The reduction rules

(In-Rule) $n[in\ m.P|Q]|m[R] \rightarrow m[n[P|Q]|R]$
(Out-Rule) $m[n[out\ m.P|Q]|R] \rightarrow n[P|Q]|m[R]$
(Open-Rule) $open\ n.P|n[Q] \rightarrow P|Q$
(Comm-Rule) $(n).P|\langle M \rangle \rightarrow P\{n \leftarrow M\}$
(New-Rule) $P \rightarrow Q \Rightarrow (\nu n)P \rightarrow (\nu n)Q$
(Par-Rule) $P \rightarrow Q \Rightarrow P|R \rightarrow Q|R$
(Amb-Rule) $P \rightarrow Q \Rightarrow n[P] \rightarrow n[Q]$
($\equiv$-Rule)$P' \equiv P, P \rightarrow Q, Q \equiv Q' \Rightarrow P' \rightarrow Q'$

We denote by $\rightarrow^+$ the transitive closure of $\rightarrow$.

We assume that ambient programs can include unspecified processes denoted by capital letters P, Q, R, hereafter called *atomical processes*[7].

Further we define some possible extensions of Ambient Calculus which are foundationally meaningful.

3.1 Ambient Calculus with denumerable parallel composition

As a first extension, we add the denumerable parallel composition to the Syntax:

If $(P_i)_{i\in\mathbb{N}}$ is a denumerable set of processes, we accept as well-formed process formula

$$|_{i\in\mathbb{N}} P_i \stackrel{def}{=} P_1|P_2|P_3...$$

[7]This is a necessary requirement in developing models where we have to recognize and distinguish, over time, unspecified processes inside the target process. For instance P is an unspecified process in $n[in\ m.P]$

Additionally, if $(P_i)_{i\in\mathbb{N}}$, $(Q_i)_{i\in\mathbb{N}}$ are denumerable sets of processes we accept the next congruence rule:

$$(GenPar)\ P_i \equiv Q_i \text{ for each } i \in \mathbb{N} \Rightarrow |_{i\in\mathbb{N}}P_i \equiv |_{i\in\mathbb{N}}Q_i.$$

and the reduction rule:

$$(DenPar - Rule)\ P_i \rightarrow Q_i \text{ for each } i \in \mathbb{N} \Rightarrow |_{i\in\mathbb{N}}P_i \rightarrow |_{i\in\mathbb{N}}Q_i.$$

This extension gives a first interpretation for replication as a particular case of denumerable composition, $!P \stackrel{def}{\equiv} P|P|P....$ In this interpretation, the axioms $(GenPar)$, $(ReplPar)$, $(ReplRepl)$ are not independent, the last two can be trivially derived from the first using the commutativity of parallel composition.

3.2 Recursive processes

Following [11, 17] we introduce the Recursive Ambient Calculus by adding to the syntax of the classical Ambient Calculus the following:

$P, Q, R ::=$	**processes**
X	**identifier**
$(recX.P)$	**recursive process**

Consequently, we add the congruence rules:

(Rec) $(recX.P) \equiv P\{X \leftarrow (recX.P)\}$
(RecNull) $(recX.X) \equiv nil$
(Rec$\equiv$) $P \equiv Q \Rightarrow (recX.P) \equiv (recX.Q)$

We call an identifier X that appears in the scope of $recX$ *bound*, otherwise it is *free*. We call a process *closed* if it does not contain any free identifier.

We identify the recursive processes up to renaming the bound identifiers, i.e. we have:

$$(recX.P) = (recY.P\{X \leftarrow Y\}) \text{ if } Y \text{ is not free in } P.$$

As expected, we write $P\{X \leftarrow Q\}$ for the outcome of substituting Q for each free occurrence of X in P, and we assume that any bound identifier is different from any free identifier.

Definition 3.1. We define the recursive depth of a process P, $rdepth(P)$ by:
$rdepth(P) = 0$ if P is a classical process,
$rdepth(P|Q) = max\{rdepth(P), rdepth(Q)\}$,
$rdepth(|_{i\in\mathbb{N}}P_i) = max_{i\in\mathbb{N}}\{rdepth(P_i)\}$,
$rdepth(C.P) = rdepth(n[P]) = rdepth(!P) = rdepth(P)$, where C is any chain of prefixes,
$rdepth(recX.P) = rdepth(P) + 1$.

Definition 3.2. We call a process *regular* iff it is structural congruent with a process having a finite recursive depth.

Hereafter we consider only regular ambient processes, these being the only processes that can be described using a finite syntax. Moreover, with respect to our definition, the recursive processes analyzed in [11, 17] are regular too.

The operational semantics proposed here for the recursive calculus, following [11, 17], do not give the possibility to recognize when a process defined by a recursive definition is congruent with a classical one. Intuitively $!P$ and $(recX.X|P)$ are observationally equivalent [16, 11]. Still $!P \equiv (recX.X|P)$ is undecidable in our operational semantics. Denote by $Q = (recX.X|P)$; if we use the structural congruence rules, we obtain $Q \equiv !P$ only if $Q|P \equiv !P$ (by (Rec)), i.e. only if $Q \equiv !P$. So, the argument is circular! Moreover, $(recX.X|(recY.Y|P)) \equiv (recX.X|P)$ and $(recX.X|P|Q) \equiv (recX.X|P)|(recX.X|Q)$ when X, Y do not appear free in P, Q (which should be the recursive correspondents of $!!P \equiv !P$ and $!(P|Q) \equiv !P|!Q$) are also undecidable in the given operational semantics.

For the moment we accept this structural congruence, but we will reopen this problem in section 9 where, inspired by the model, we will propose an extension of the structural congruence. By using the concept of pattern of structural unfoldings of an infinite tree we will provide operational semantics which will decide the truth values of the previously mentioned statements.

3.3 The Master Ambient

The reduction rule $(Amb - Rule) P \to Q \Rightarrow n[P] \to n[Q]$ allows us to assume the existence of a special ambient, denoted by $\widehat{1}$ (hereafter *the master ambient*), which fulfills the intuition that all the processes are enclosed in a Universe. With this intuition, the master ambient cannot be guarded by any prefix and its name, $\widehat{1}$, cannot be referred by any capability, input, or new name prefix. Moreover, $\widehat{1}$ appears in a process syntax only once, enclosing all the process, to ensure that its structure tree has a unique root. This sustains the intuition that any process, as a whole, occupies a spatial position.

3.4 The Transparent Ambient

The basic idea of spatiality in the case of Ambient Calculus is to consider the processes that are isolated together from the rest of the system as belonging to the same ambient (i.e. to the same spatial location). These processes cannot interact directly with the rest of the system, and can interact inside the ambient only if the ambient is not guarded by some capabilities.

Consider now the processes $\widehat{1}[P|Q|R]$, $\widehat{1}[P|c.(Q|R)]$ and $\widehat{1}[P|c.n[Q|R]]$, where $c \neq \varepsilon$ is a capability that guards the parallel composition of the processes Q and R in the second process.

While in the first process any interaction between P, Q and R is allowed, in the second the interaction between P and Q or R is forbidden, and Q and R cannot interact before c will be consumed. The same regarding the third process. It seems that in the second process, in spite of the fact that P, Q and R are in the same ambient (they share the same spatial position), the capability c determines a closed spatial position that acts as an ambient. It is not an ambient, because it cannot move as a whole, but still, on the spatiality level, the structure of the second process looks more like the structure of the third process than as the first.

We propose, in order to describe the action of this "theoretical ambient" to accept the existence of a special ambient $\widehat{0}$ (named *the transparent ambient*) to describe this spatial anomaly. This ambient name, as $\widehat{1}$, cannot be argument for any capability, input, or new name operator, but, unlike $\widehat{1}$, can be guarded by capabilities. It satisfies the following congruence rule:

$$\text{(Transp): } \widehat{0}[P] \equiv P.$$

With this agreement we will treat $\widehat{1}[P|c.(Q|R)]$ as $\widehat{1}[P|c.\widehat{0}[Q|R]]$, while after consuming the capability c we have $\widehat{1}[P|\widehat{0}[Q|R]] \equiv \widehat{1}[P|Q|R]$.

The action of the transparent nodes is more intuitive in the tree representation. Indeed, the processes $\widehat{1}[S|c.(P|Q)|R]$ (that actually we write as $\widehat{1}[S|c.\widehat{0}[P|Q]|R]$) and $\widehat{1}[S|c.P|Q|R]$ have the structure trees described by figures 5 and 6, respectively.

*fig.*5

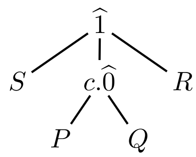
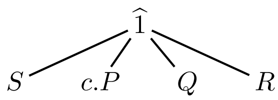
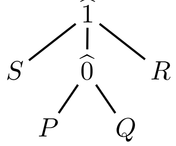
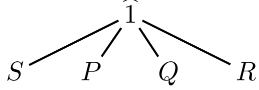


*fig.*6

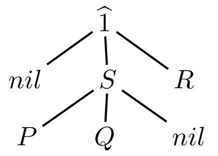


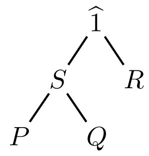
Transparent nodes $\widehat{0}$ guarded by the empty capability (or not guarded) can be dissolved, due to the rule $(Transp)$ $\widehat{0}[P] \equiv P$. In this case the parent of a transparent node becomes a parent for its children, as in the next graphs:





The branches ending with nodes labelled by the empty process *nil*, due to the structural congruence rules, can be cut off. So, the next two structure trees are equivalent.





3.5 The Signatures of the ambient calculi

We denote by Π the class of atomical process names with $nil \in \Pi$. In the classical Ambient Calculus this class contains only the empty process name. Still we find it interesting to accept other uninterpreted processes because they are widely used in applications. Let Λ be the class of ambient names with $\widehat{0}, \widehat{1} \in \Lambda$.

Hereafter we use $\mathfrak{P}$ to denote the class of the classical ambient processes that can be described without involving replication, and $\mathfrak{P}^!$ as the supraclass of $\mathfrak{P}$ containing also the processes involving replication. Classical Ambient Calculus is an algebraical structure $(\mathfrak{P}^!, \equiv, |, m[\ ], M.\ , (\nu n), !, \rightarrow)$, where the symbols have the usual meaning.

We denote by $\mathfrak{P}^\infty$, the ambient processes enriched with denumerable parallel composition. Obviously, $\mathfrak{P}^! \subset \mathfrak{P}^\infty$ if we interpret $!P = P|P|P|....$ This syntax gives the algebraical structure $(\mathfrak{P}^\infty, \equiv, |, |_{i\in\mathbb{N}}, m[\ ], M.\ , (\nu n), !, \rightarrow)$.

We also consider the class of recursive ambient processes that do not involve replication $\mathfrak{P}_{rec}$, and the class of recursive ambient processes involving replication, $\mathfrak{P}^!_{rec}$. As a superclass of the last one we consider the class of the recursive ambient processes involving denumerable parallel composition, and we denote it by $\mathfrak{P}^\infty_{rec}$. All these came with their algebraical structures: $(\mathfrak{P}_{rec}, \equiv, |, M., m[\ ], (\nu n), (recX.\ ), \rightarrow)$, $(\mathfrak{P}^!_{rec}, \equiv, |, M., m[\ ], (\nu n), !, (recX.\ ), \rightarrow)$, and $(\mathfrak{P}^\infty_{rec}, \equiv, |, |_{i\in\mathbb{N}}, M., m[\ ], (\nu n), !, (recX.\ ), \rightarrow)$.

We denote by $\overline{rec}P \stackrel{def}{=} (recX.X|P)$, when there is no free occurrence of X in P. As we discussed before, $\overline{rec}P$ does not model $!P$, because it is not the case that $\overline{rec}(P|Q) \equiv \overline{rec}P|\overline{rec}Q$ and that $\overline{rec}(\overline{rec}P) \equiv \overline{rec}P$.

We denote by *Cap* the class of capabilities taking values over $\Lambda \setminus \{\widehat{0}, \widehat{1}\}$. For reaching more uniformity in the calculus, we also treat $\langle M\rangle$ as a capability considering the process $c.\langle M\rangle$ as being $c.\langle M\rangle.nil$. We refer to capabilities together with new name prefix (νn) as to *prefixes*. We will use C, C_i, C', C'' to refer to chains of prefixes, and c, c_i, c', c'' to refer to individual capabilities. We will also use M when we refer either to a capability or to a new name prefix. Any chain of prefixes has to be suffixed by the empty capability or by an output. For this reason we split *Cap* in $Cap_{\langle\rangle}$ and respectively Cap_ε to refer to the two classes.

When we refer to the chains of prefixes outside the context of Ambient Calculus we will use the notation $\langle c_1, c_2, ...c_k\rangle$ for denoting the chain $c_1.c_2...c_k$. If C_1, C_2 are two chains, we will use the notation $\langle C_1, C_2\rangle$ to denote the catenation of the two, while when we refer to one of them we use the capital letter without angles parenthesis.

Theorem 3.1. $\rightarrow^+$ *is an internal operation with respect to each of the sets* $\mathfrak{P}, \mathfrak{P}^!, \mathfrak{P}^\infty, \mathfrak{P}_{rec}, \mathfrak{P}^\infty_{rec}$.

4 Overview on Set Theory

In this section we present set theoretical tools used to analyze hierarchies. We define the notion of *flat system of equations*. These systems will be identified behind the spatial structure of any ambient process, and will be used to understand the relevance of structural congruence between processes from a spatial point of view.

Our work is based on some results in Set Theory without foundation, first developed by F. Honsell and M. Forti (1983) [13], P. Aczel (1988) [3], and by J.Barwise and L.Moss (1996) [5].

4.1 The system ZFA

We work in Zermelo-Fraenkel system of Set Theory with the Anti-Foundation Axiom[8] ZFA. This system assumes a class $\mho$ of urelements, objects which are not sets (they do not have elements) but can be elements of sets. The urelements together with the empty set $\emptyset$ generates all the sets we work with (sometimes sets of sets). We also assume the Strong Axiom of Plenitude[9] in the system, which ensures the richness of the class $\mho$.

Definition 4.1. A binary relation $\mathfrak{R}$ over a set S is *wellfounded* if there is no infinite sequence $b_0, b_1, b_2, ...$ of elements of S such that $b_{n+1}\mathfrak{R}b_n$ for each $n = 0, 1, 2, ...$ If there is such a sequence, then $\mathfrak{R}$ is said to be *non-wellfounded*.

Informally, the Foundation Axiom (FA) states that the membership relation $\in$ over the universe of Set Theory is a wellfounded one, while the Anti-Foundation Axiom (AFA) states that $\in$ is non-wellfounded. If we accept (FA) we obtain the system ZFC of the classical Set Theory, while if we accept (AFA) in place of (FA) we obtain the system ZFA of Hypersets Theory [3, 5].

The advantage of using the system ZFA is that it allows the existence of non-wellfounded sets (hypersets), and in this way it enriches the classical universe of Set Theory with new entities that can be used to describe circular phenomena (as is the case with phenomena that ask for recursive definitions). For example, the set defined by $x = \{x\}$ (Aczel's set Ω) is a hyperset, reflecting a structure for which $\in$ is non-wellfounded. In classical set theory such entities were rejected in order to avoid the paradoxes. Concerning the economy of the paper, we will not present additional features of the system here. Hereafter, when we refer to sets we mean both wellfounded (classical) sets and non-wellfounded sets (pure hypersets). We end this subsection by recalling some definitions that will be used further.

Definition 4.2. A set a is *transitive* if all the elements of a set b, which is an element of a, also belong to a: $\forall b \in a$ if $c \in b$ then $c \in a$.
The *transitive closure* of a, denoted by $TC(a)$ is the smallest transitive set[10] including a.

Definition 4.3. The *support of a set* a, denoted by $supp(a)$ is $TC(a) \cap \mho$. The elements of $supp(a)$ are the urelements that are *somehow involved* in a.

Definition 4.4. If $a \subseteq \mho$ then $V[a] \overset{def}{=} \{b \mid b \text{ is a set and } supp(b) \subseteq a\}$. $V[a]$ is the class of all sets in which the only urelements that are somehow involved are the urelements of a.

Definition 4.5. Let $A \subseteq \mho$. A set a is *hereditarily finite over* A if every set $b \in TC(\{a\})$ is finite and $supp(a) \subseteq A$.

4.2 The flat systems of equations

Definition 4.6. A *(flat) system of equations* is a tuple $\mathcal{E} = \langle X, A, e\rangle$ consisting of a set $X \subseteq \mho$ of *indeterminates*, a set A of *atoms* disjoint from X, and a function $e : X \to \mathcal{P}(X \cup A)$. We refer to $X \cup A$ as *the domain of* $\mathcal{E}$. For each $v \in X$, the set $b_v \overset{def}{=} e_v \cap X$ is called the set of indeterminates on which v immediately depends. Similarly, the set $c_v \overset{def}{=} e_v \cap A$ is called the set of atoms on which v immediately depends (we wrote e_v for $e(v)$).

Example 4.1. *Consider the set $x = \{\alpha, \{\beta, \{\gamma\}\}\}$. This hierarchical structure can be described by the one-level hierarchies $x = \{\alpha, y\}$, $y = \{\beta, z\}$, $z = \{\gamma\}$ or using a flat system of equations $\mathcal{E} = \langle X, A, e\rangle$ with $X = \{x, y, z\}$, $A = \{\alpha, \beta, \gamma\}$ and $e_x = \{\alpha, y\}$, $e_y = \{\beta, z\}$, $e_z = \{\gamma\}$. The solution of this system of equations describes our set.*

If we define $x = \{\alpha, \{\beta, \{x\}\}\}$, i.e. it is a hyperset (having a recursive definition), it still can be expressed using a finite flat system of equations defined by $e_x = \{\alpha, y\}$, $e_y = \{\beta, z\}$, $e_z = \{x\}$.

[8] This axiom will be clarified later in this section.
[9] This axiom states that anytime we choose an urelement and a set, we can find an urelement different of the chosen one and which is not an element of the chosen set.
[10] The existence of $TC(a)$ is justified by: $TC(a) = \cup\{a, \cup a, \cup\cup a, ...\}$

Definition 4.7. A *solution* to $\mathcal{E}$ is a function s with domain X satisfying $s_x = \{s_y | y \in b_x\} \cup c_x$ for each $x \in X$. The *solution-set* of a flat system $\mathcal{E}$ of equations is the set
$ss(\mathcal{E}) \stackrel{def}{=} \{s_v | v \in X\} = s[X]$ (we wrote $s[X]$ for the image of X by s).

The next theorem allows our further construction ([5], part III, section 6).

Theorem 4.2. *In ZFA each set $a \in V[A]$ is a solution-set of a flat system of equations which has A as the set of atoms (or a subset of A) and any flat system of equations with the atoms in A has a set $a \in V[A]$ (possible non-wellfounded set) as unique solution-set.*

4.3 The bisimulation relation on flat systems

We introduce two relations of bisimulation over systems of equations, extending the one proposed in [5].

Definition 4.8. Let $A, B \in \mho$ be two sets of urelements and $\zeta \subset A \times B$ a relation on them. ζ ia a *proper relation* if the following conditions hold:

1. $\forall a \in A, \exists b \in B$ such that $(a, b) \in \zeta$ and $\forall b \in B, \exists a \in A$ such that $(a, b) \in \zeta$
2. $(a_1, b_1), (a_1, b_2), (a_2, b_1) \in \zeta \Longrightarrow (a_2, b_2) \in \zeta$
3. $(a_1, b_1), (a_2, b_1), (a_2, b_2) \in \zeta \Longrightarrow (a_1, b_2) \in \zeta$

Theorem 4.3 (The characterization of proper relations). *Let $\zeta \subset A \times B$ be a proper relation. There exist a partition $(A_i)_{i \in I}$ of A and a partition $(B_i)_{i \in I}$ of B such that $(a, b) \in \zeta$ iff it exists $i \in I$ with $a \in A_i$, $b \in B_i$.*

Definition 4.9. Let $A, A' \subseteq \mho$ be two sets of urelements and $\zeta \subset A \times A'$ a proper relation on them. *The canonical extension of* ζ is the relation $\overline{\zeta} \subset (A \times A') \cup (V[A] \times V[A'])$ defined by:

1. if $(b, b') \in A \times A'$ then $(b, b') \in \overline{\zeta}$ iff $(b, b') \in \zeta$
2. if $(b, b') \in V[A] \times V[A']$ then $(b, b') \in \overline{\zeta}$ iff
   (a) $\forall c \in b \; \exists c' \in b'$ such that $(c, c') \in \overline{\zeta}$
   (b) $\forall c' \in b' \; \exists c \in b$ such that $(c, c') \in \overline{\zeta}$
   (c) $card(b) = card(b')$

Definition 4.10. Let $\mathcal{E} = \langle X, A, e\rangle$, $\mathcal{E}' = \langle X', A', e'\rangle$ be two flat systems of equations, $A, A' \subset \mho$ and $\zeta \subset A \times A'$ a proper relation. A ζ*-bisimulation relation* between $\mathcal{E}$ and $\mathcal{E}'$ is a relation $\mathfrak{R} \subseteq X \times X'$ such that whenever $x \mathfrak{R} x'$ the following conditions hold:

1. For every $y \in e_x \cap X$ there is an $y' \in e'_{x'} \cap X'$ such that $y \mathfrak{R} y'$
2. For every $y' \in e'_{x'} \cap X'$ there is an $y \in e_x \cap X$ such that $y \mathfrak{R} y'$
3. $(e_x \cap A, e'_{x'} \cap A') \in \overline{\zeta}$

We say that the systems are *substitutive-bisimilar* (for a particular ζ we call them ζ*-bisimilar*), and we write $\mathcal{E} \equiv_S \mathcal{E}'$ if there is a ζ-bisimulation relation between them such that for every $x \in X$ there is an $x' \in X'$ with $x \mathfrak{R} x'$ and vice versa.

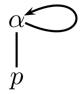
Theorem 4.4. *The relation $\equiv_S$ is an equivalence relation over flat systems of equations.*

The meaning of the previous definition is that two substitutive-bisimilar systems describe the same structure up to the choice of the atoms. Two (non-wellfounded) systems could be substitutive-bisimilar even if not identical, as in the next example.

Example 4.5. *Consider the systems* $\mathcal{E} = \langle X, A, e\rangle$*, with* $X = \{\alpha\}, A = \{p\}, e(\alpha) = \{\alpha, p\}$ *and* $\mathcal{E}' = \langle X', A', e'\rangle$*, with* $X' = \{\alpha, \beta\}, A' = \{p'\}, e(\alpha) = \{\beta, p'\}, e(\beta) = \{\alpha, p'\}$*. Let* $\zeta(p) = p'$ *and* $\mathfrak{R} = \{\langle\alpha, \alpha\rangle, \langle\alpha, \beta\rangle\}$*. It is easy to verify that, in spite of the fact that* $cardX \neq cardX'$ *the two systems are substitutive-bisimilar. Still, if we analyze the graphs describing the structures of the two systems (the set of vertices is the set of indeterminates and atoms of the system, and the edges are defined by the function e) we found the same structure, as it can be seen in the fig.7 and fig.8.*[11]*. Being this recursive behavior, the two graphs describe the same structure up to the choice of atoms.*

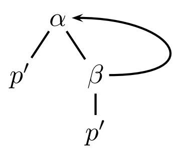
$fig.7$

$\mathcal{E} = \langle X, A, e\rangle$



$fig.8$

$\mathcal{E}' = \langle X', A', e'\rangle$



Theorem 4.6. *Let* $\mathcal{E}, \mathcal{E}'$ *be two flat systems of equations and* ζ *a proper relation on their sets of atoms. Then the two systems are* ζ*-bisimilar iff their solution-sets are in relation* $\overline{\zeta}$*.*

This theorem says that, by extending the map ζ from the sets of atoms of the two systems to the whole universe of set theory with respect to the conditions in definition 4.9, we obtain bisimilar systems iff their solutions correspond each other by $\overline{\zeta}$.

4.4 Regular systems

Definition 4.11. Let $\mathcal{E} = \langle X, A, e\rangle$ be a flat system of equations and let R be the binary relation on $X \cup A$ defined by xRy iff $y \in e_x$. We call the system *wellfounded* if R is wellfounded and *non-wellfounded* else.

Definition 4.12. A *descendent path originated in* $u \in X$ is a sequence of different $u_1, u_2... \in X \setminus \{u\}$ such that uRu_1, u_iRu_{i+1}. *A descendent path from* $u \in X$ *to* $v \in X \cup A$ (not necessarily $u \neq v$) is a finite descendent path $u_1, ...u_k \in X \setminus \{u, v\}$, originated in u, such that u_kRv. If there is a descendent path from u to v, we say that v *is accessible from* u.

Definition 4.13. We call *a loop in* u a descendent path from u to itself. We say that *a loop in* v *is subsequent of a loop in* u if v is accessible from u.

Definition 4.14. We call a flat system *regular* if does not contain any element originating an infinite descendent path and $e(x)$ is finite or denumerable for any $x \in X$.

Remark 4.1. Due to the definition, a descendent path has to contain distinct elements. For this reason a circular path is not a descendent one, hence systems containing loops can be regular. But if in a system there is a denumerable chain of loops each loop having as a successor a subsequent one, the system is not regular for the reason that the denumerable sequence crossing all the nodes originating the loops is an infinite descendent path.

Definition 4.15. A flat system of equations $\mathcal{E} = \langle X, A, e\rangle$ is *rooted* if there exists a unique element $\alpha \in X$ originating descendent paths to any element in $(X \cup A) \setminus \{\alpha\}$. We call α *the root of the system* and we denote such a system by $\mathcal{E} = \langle X, A, e, \alpha\rangle$. A *bisimulation of rooted flat systems* is a bisimulation $\mathfrak{R}$ with the additional condition $\alpha_1\mathfrak{R}\alpha_2$, α_1, α_2 being the roots of the two systems.

The uniqueness condition for α ensures that it exists only one root of the system.

Definition 4.16. If $\mathcal{E} = \langle X, A, e, \alpha\rangle$ is a flat system of equation, a *subsystem* of $\mathcal{E}$ is a flat system $\mathcal{E}' = \langle X', A', e', \alpha'\rangle$ with $X' \subseteq X$, $A' = A$ and $e' = e|_{X'}$. We denote this by $\mathcal{E}' \sqsubseteq \mathcal{E}$.

[11]Hereafter we use arrows in a tree connecting two vertices to point to the presence of a recursive behavior of the structure following the direction indicated by the arrow.

5 Ambient-Graphs and Ambient-Hierarchies

Let Λ be the class of names of the Ambient Calculus, and let $\Lambda_{bn} \stackrel{def}{=} (\Lambda \setminus \{\widehat{0}, \widehat{1}\}) \times \mathbb{N}^+ \times \{bn\}$, and $\Lambda_{pn} \stackrel{def}{=} (\Lambda \setminus \{\widehat{0}, \widehat{1}\}) \times \mathbb{N}^+ \times \{pn\}$ be two indexed labelled copies of $\Lambda \setminus \{\widehat{0}, \widehat{1}\}$, where bn and pn are two labels. We write $n_i^{bn} \stackrel{def}{=} (n, i, bn) \in \Lambda_{bn}$, $n_I^{bn} = \{(n, i, bn) \mid i \in I\}$, and symmetrically, $n_i^{pn} \stackrel{def}{=} (n, i, pn) \in \Lambda_{pn}$, $n_I^{pn} = \{(n, i, pn) \mid i \in I\}$.

We define the sets $\mathcal{M}_{amb} = (\Lambda \cup \Lambda_{bn} \cup \Lambda_{pn}) \times Cap_\varepsilon$, and $\mathcal{M}_{pr} = (\Pi \times Cap_\varepsilon) \cup (\{nil\} \times Cap_{\langle\rangle})$, and we consider $\mathcal{M} = \mathcal{M}_{pr} \cup \mathcal{M}_{amb}$, where the sets of capabilities are built upon the names in $(\Lambda \cup \Lambda_{bn} \cup \Lambda_{pn}) \setminus \{\widehat{1}, \widehat{0}\}$.

We will use the set $\mathcal{M}$ as a set of labels for the trees that depict the spatial structure of the ambient processes. To simplify the presentation we identify three special labels as follows: $\widehat{1} = (\widehat{1}, \varepsilon)$, $\widehat{0} = (\widehat{0}, \varepsilon)$ and $nil = (nil, \varepsilon)$. We have $\widehat{1}, \widehat{0} \in \mathcal{M}_{amb}$ and $nil \in \mathcal{M}_{pr}$. The reason for duplicating Λ by Λ_{bn} and Λ_{pn} is that we intend to use these classes to distinguish the free occurrences of a name $n \in \Lambda$ by its input-bounded occurrences (collected in Λ_{bn}) and, respectively, by its private occurrences (collected in Λ_{pn}).

We say that a name $n \in \Lambda \cup \Lambda_{bn} \cup \Lambda_{pn} \cup \Pi$ is *involved* in $m \in \mathcal{M}$ if m contains an occurrence of n either as a name of an ambient, or as an argument of a prefix.

Definition 5.1 (Decorated Graphs). A *decorated graph* is a couple $G = \langle \mathcal{E}, \mathcal{L} \rangle$ where $\mathcal{E} = \langle X, A, e, \alpha \rangle$ is a regular rooted flat system of equations and $\mathcal{L} : X \cup A \to \mathcal{M}$ is a function with the properties:

$$1.\ \mathcal{L}(X) \subseteq \mathcal{M}_{amb} \quad 2.\ \mathcal{L}(A) \subseteq \mathcal{M}_{pr} \quad 3.\ \mathcal{L}^{-1}(\widehat{1}) = \{\alpha\}$$

A decorated graph contains a structure (described by a flat system of equations) and a labelling function that associates to each node a couple consisting in a name (of an ambient or of an atomical process) together with a list of prefixes. This representation provides a correct model for the Ambient Calculus, but not a sound one. In fact, to provide a model for the structural congruence, we should have the names in Λ_{bn} only occurring as input names and those in Λ_{pn} only occurring as private. But in a decorated graph there is no restriction in this sense.

Definition 5.2. Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ be a decorated graph with $\mathcal{E} = \langle X, A, e, \alpha \rangle, v \in X$ and $\mathcal{L}(v) = (m, \langle C', c, C'' \rangle)$. *The subgraph of G prefixed by c* is the decorated graph $G' = \langle \mathcal{E}', \mathcal{L}' \rangle$ with $\mathcal{E}' = \langle X', A, e', v \rangle \sqsubseteq \mathcal{E}$, defined by the subset $X' \subseteq X$ of the elements accessible in $\mathcal{E}$ from v, having v as root, and the labels defined by $\mathcal{L}'(v) = (m, \langle c, C'' \rangle)$ and $\mathcal{L}' = \mathcal{L}|_{X' \setminus \{v\} \cup A'}$.

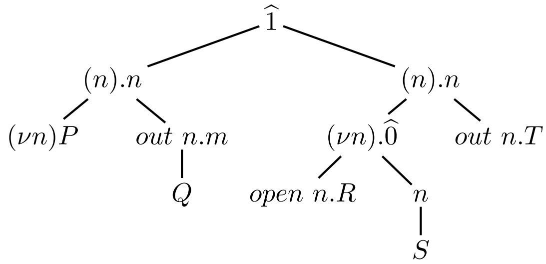
Definition 5.3. Let n be a name involved in the decorated graph G. If there is an occurrence of (n) in G, we call the subgraph prefixed by (n) *the input domain of* n. If there is no occurrence of (n) in G, we say that n is a *free name in* G. If in G there is no free occurrence of n, we say that n is a *bound name*. If there is an occurrence of (νn) in G, we call the subgraph prefixed by (νn) *the private domain of* n. If there is no occurrence of (νn) in G, we say that n is a *public name in* G. If there is no public occurrence of n in G we say that n is a *private name*.

Remark 5.1. There are names which are neither free nor bound (when occurrences of the same name appears both free and bounded by input). There are names which are neither public nor private (when occurrences of the same name appears both free and private). This is why the decorated graphs are not sound models for ambient processes. Hence we introduce the Ambient-Graphs.

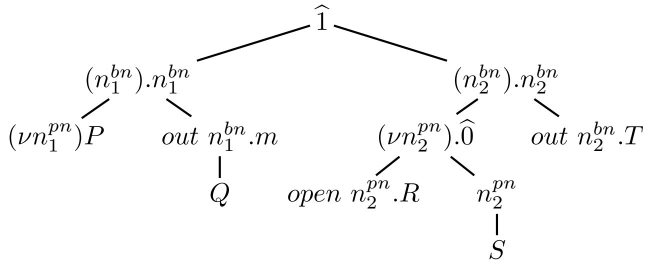
Definition 5.4 (Ambient-Graphs). An *Ambient-Graph* (or A-graph for short) is a decorated graph $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with the property that any name $n \in \Lambda \cup \Lambda_{bn} \cup \Lambda_{pn}$ involved in G is either public and free (and in this case $n \in \Lambda$), either private having only one private domain (in this case $n \in \Lambda_{pn}$), or bound having only one input domain (in this case $n \in \Lambda_{bn}$). We denote by $\mathcal{G}$ the class of A-graphs.

We intend to model the process $\widehat{1}[(n).n[(\nu n)P|out\ n.m[Q]]|(n).n[(\nu n)(open\ n.R|n[S])|out\ n.T]]$ by the ambient-graph in fig.10, and not by the decorated graph described in fig.9 (hereafter we sometimes add only the labels of the trees without the names of the nodes).

$fig.9$



$fig.10$



Definition 5.5. Given an A-graph G, we denote by $fn(G)$ the sets of free and public names, by $bn(G)$ the set of bound names and by $pn(G)$ the set of private names of G. If $n_i^{bn} \in bn(G)$ we define $arity_G^{bn}(n) = \{i \in \mathbb{N}^+ \mid n_i^{bn} \in bn(G)\}$; if $n_i^{pn} \in pn(G)$ we define $arity_G^{pn}(n) = \{i \in \mathbb{N}^+ \mid n_i^{bn} \in pn(G)\}$.

For the A-graph in fig.10 we have $fn(G) = \{m\}, bn(G) = \{n_1^{bn}, n_2^{bn}\}, pn(G) = \{n_1^{pn}, n_2^{pn}\}$ and $arity_G^{bn}(n) = \{1, 2\}, arity_G^{pn}(n) = \{1, 2\}$.

5.1 The structures of the private and input domains

Definition 5.6. Consider an A-graph G and $n_i^{bn}, m_j^{bn} \in \Lambda_{bn}$. We define a relation $\downarrow$ on the set of input names by $n_i^{bn} \downarrow m_j^{bn}$ if the input domain of m_j^{bn} is included in the domain of n_i^{bn}.

Consider now the flat system of equations $\mathcal{N}_G^{bn} = \langle X_{bn}, A_{bn}, e_{bn}\rangle$, where $X_{bn} = \{n_i^{bn} \in bn(G) \mid \exists m_j^{bn} \in bn(G), n_i^{bn} \downarrow m_j^{bn}\}$, $A_{bn} = \{n_i^{bn} \in bn(G) \mid \nexists m_j^{bn} \in bn(G), n_i^{bn} \downarrow m_j^{bn}\}$, and $e_{bn}(n_i^{bn}) = \{m_j^{bn} \mid n_i^{bn} \downarrow m_j^{bn}\}$. We call $\mathcal{N}_G^{bn}$ *the system of bounded names of G.*

Similarly, we define:

Definition 5.7. Consider the A-graph G and $n_i^{pn}, m_j^{pn} \in \Lambda_{pn}$. We define a relation $\downarrow$ on the set of private names by $n_i^{pn} \downarrow m_j^{pn}$ if the private domain of m_j^{pn} is included in the domain of n_i^{pn}. In the case that $(\nu m_1^{pn}), (\nu m_2^{pn}), ...(\nu m_k^{pn})$ are consecutive prefixes in the same label, we assume $m_i^{pn} \downarrow m_j^{pn}$ for any $i \neq j$.

Consider now the flat system of equations $\mathcal{N}_G^{pn} = \langle X_{pn}, A_{pn}, e_{pn}\rangle$, where $X_{pn} = \{n_i^{pn} \in pn(G) \mid \exists m_j^{pn} \in pn(G), n_i^{pn} \downarrow m_j^{pn}\}$, $A_{pn} = \{n_i^{pn} \in pn(G) \mid \nexists m_j^{pn} \in pn(G), n_i^{pn} \downarrow m_j^{pn}\}$, and $e_{pn}(n_i^{pn}) = \{m_j^{pn} \mid n_i^{pn} \downarrow m_j^{pn}\}$. We call $\mathcal{N}_G^{pn}$ *the system of private names of G.*

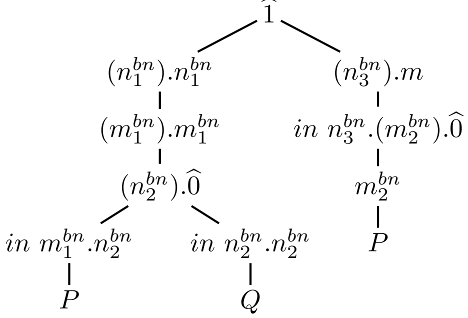
Remark 5.2. Observe that $X_{bn} \cap A_{bn} = \emptyset, X_{bn} \cup A_{bn} = bn(G)$ and $X_{pn} \cap A_{pn} = \emptyset, X_{pn} \cup A_{pn} = pn(G)$.

Consider now the ambient process:

$$\widehat{1}[(n).n[(m).m[(n).\widehat{0}[in\ m.n[P]|in\ n.n[Q]]]|(n).m[in\ n.(m).\widehat{0}[m[P]]]]]]$$

Its structure is described by the ambient-graph described in fig.11.

$fig.11$

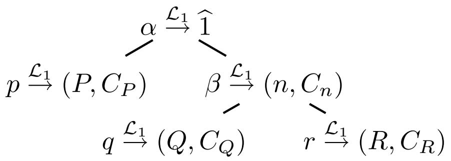


For this A-graph we have $A_{bn} = \{n_2^{bn}, m_2^{bn}\}$, $X_{bn} = \{n_1^{bn}, n_3^{bn}, m_1^{bn}\}$, $e_{bn}(n_1^{bn}) = \{m_1^{bn}\}, e_{bn}(n_3^{bn}) = \{m_2^{bn}\}, e_{bn}(m_1^{bn}) = \{n_2^{bn}\}$.

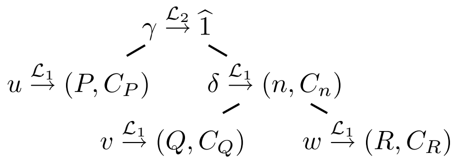
5.2 The congruence relation on A-graphs

In this subsection we will introduce a congruence relation that match the A-graphs describing the same ambient process. The first idea is to ask for two congruent A-graphs to have bisimilar structures and the bisimilar nodes to have identical labels up to the renaming of bound names and private names respectively. For example the A-graphs in figures 12 and 13 models the same process $\widehat{1}[C_P.P|C_n.n[C_Q.Q|C_R.R]]$.

$fig.12$

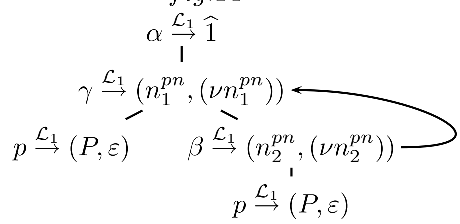


$fig.13$

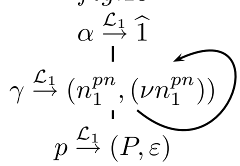


The labels of bisimilar nodes do not have to be identical when private names, or input names are involved. Requiring for a one-to-one correspondence between the sets of private names and between the sets of input names, respectively, is a solution only for well-founded A-graphs. Indeed, consider the A-graphs in figures 14 and 15, where in both situations we have recursive structures.

$fig.14$



$fig.15$



These A-graphs describe the processes $(recX.\ (\nu n)n[P|(\nu n)n[P|X]])$ and $(recX.\ (\nu n)n[P|X])$ which are structurally congruent, due to (Rec), hence the A-graphs should be congruent. In this case a one-to-one correspondence between private (respective bound) names is not a satisfactory requirement because the first A-graph has two private names while the second A-graph has only one. Similar examples can be constructed for input names.

We then ask for two congruent A-graphs to have bisimilar flat systems and bisimilar systems of bound names and of private names, respectively. Indeed, in our example the two systems of private names are ζ-bisimilar if we define $\zeta(n_1) = n_1$ and $\zeta(n_2) = n_1$.

Definition 5.8. Let G, G' be two A-graphs and $\phi \subset (\Lambda \cup \Lambda_{bn} \cup \Lambda_{pn}) \times (\Lambda \cup \Lambda_{bn} \cup \Lambda_{pn})$ a relation between their sets of names that is extended to the instances of the names and of the prefixes[12] by:

1. if $(u, v) \in \phi$ then any instances of these names are in the relation ϕ,
2. if $(u, v) \in \phi \cap (\Lambda \times \Lambda)$ then $(Mu, Mv) \in \phi$,
3. if $(u, v) \in \phi \setminus (\Lambda \times \Lambda)$ and $\nexists w \neq v$ s.t. $(u, w) \in \phi$ then $(Mu, Mv) \in \phi$,
4. if $(u, v), (u, w) \in \phi \setminus (\Lambda \times \Lambda)$ then $(Mu, Mv) \in \phi$ only if this occurrence of Mv is in the subdomain of v and not in the one of w
5. $(\langle M \rangle, \langle M' \rangle) \in \phi$ iff $(M, M') \in \phi$; $(\varepsilon, \varepsilon) \in \phi$; $(\langle c_1, ...c_k \rangle, \langle c'_1, ...c'_k \rangle) \in \phi$ iff $\forall i, (c_i, c'_i) \in \phi$.

A *syntactical extension of* ϕ, denoted by $\overline{\phi}$, is a relation between the sets of labels of G and G' defined by

$$((m, C), (m', C')) \in \overline{\phi} \text{ iff } (m, m'), (C, C') \in \phi$$

Definition 5.9. Let $G = \langle \mathcal{E}, \mathcal{L} \rangle, G' = \langle \mathcal{E}', \mathcal{L}' \rangle$ be two A-graphs with $\mathcal{E} = \langle X, A, e, \alpha \rangle$, and $\mathcal{E}' = \langle X', A', e', \alpha' \rangle$. The two A-graphs are congruent, $G \cong G'$, iff

[12] We denote by Mu a prefix involving the name u

1. $\mathcal{E} \equiv_S \mathcal{E}'$ where the bisimulation $\mathfrak{R}$ is defined by a proper relation ζ of A to A'.

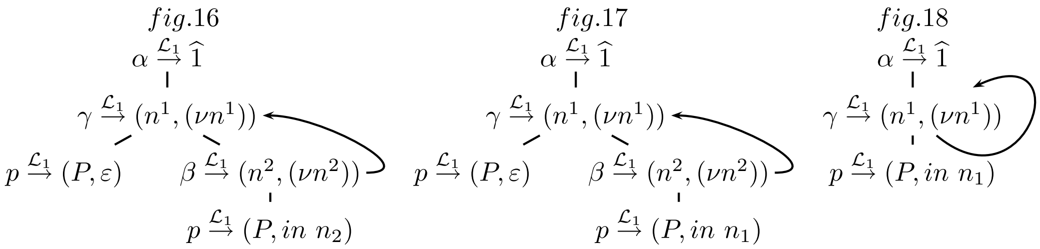
2. $\mathcal{N}_G^{bn} \equiv_S \mathcal{N}_{G'}^{bn}$ by the bisimulation relation defined by $\phi_{bn} \subset bn(G) \times bn(G')$

3. $\mathcal{N}_G^{pn} \equiv_S \mathcal{N}_{G'}^{pn}$ by the bisimulation relation defined by $\phi_{pn} \subset pn(G) \times pn(G')$

4. $\mathcal{L}(x_1)\overline{\phi}\mathcal{L}'(x_2)$ for $x_1 \mathfrak{R} x_2$, where $\phi = \phi_{pn} \cup \phi_{bn} \cup id_{\Lambda \cup \Pi}$

If we reconsider the A-graphs in figures 14 and 15, we observe that $\phi_{pn} = \{(n^1, n^1), (n^2, n^1)\}$ defines a substitutive-bisimulation between the two systems of private names, and defines the relation between the labels of bisimilar A-graph nodes. Indeed, for wellfounded A-graphs (A-graphs with wellfounded structures) the bisimulation of private names became a one-to-one correspondence.

The role of the conditions 3 and 4 in the definition 5.8 is to prevent a situation like in the following examples. The A-graph in fig.16 must be congruent with the one in fig.18 by $\zeta(n_1) = \zeta(n_2) = n_1$, but the same ζ does not have to give us a bisimulation between the A-graphs in fig.17 and in fig.18, due to the fact that the position of $in\ n_1$ in the A-graph 17 does not simulate correctly the action of $in\ n_1$ in the A-graph 18 (being in the domain of (νn_2)).



Theorem 5.1. *The congruence relation over A-graphs is an equivalence relation.*

Definition 5.10. We call the equivalence classes over $\mathcal{G}$ defined by the congruence relation $\cong$ *Ambient-Hierarchies* (or *A-hierarchies* for short). We denote by $\mathcal{A}$ the set of all A-hierarchies. We denote the A-hierarchy that contains the A-graph G by $(\!|G|\!)$.

We intend to use the A-hierarchies as models for ambient processes. Thus we define the A-hierarchy as "the pattern of the structure" up to the choice of the A-graph used to describe it (the A-graph being dependent of the urelements involved in its flat system).

6 An algebra of Ambient-Hierarchies

In this section we define some operations over A-hierarchies that will organize algebraic structures over $\mathcal{A}$. Our intention is to provide for $\mathcal{A}$, algebraical structures isomorphic with those of Ambient Calculi.

Lemma 6.1. *Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ be an A-graph with $\mathcal{E} = \langle X, A, e, \alpha \rangle$.*

1. *For any set S there is an A-graph $G' = \langle \mathcal{E}', \mathcal{L}' \rangle$, $\mathcal{E}' = \langle X', A', e', \alpha' \rangle$ with $G' \in (\!|G|\!)$ (i.e. $G \cong G'$) such that $S \cap (X' \cup A') = \emptyset$.*

2. *Let $S \subseteq \mho$ be a set cardinal-equivalent with X and R a set cardinal-equivalent with A, $S \cap R = \emptyset$. There is an A-graph $G' = \langle \mathcal{E}', \mathcal{L}' \rangle$, $\mathcal{E}' = \langle S, R, e', \alpha' \rangle$ with $G' \in (\!|G|\!)$ (i.e. $G \cong G'$).*

Hereafter we denote by $(n \leftrightarrow m)$ the substitution between the names n and m, and by $(A \leftrightarrow B)$ the bijective substitution between two cardinal equivalent sets of names A and B.

6.1 Atomical A-Hierarchies

The intuition behind the construction of an atomical A-hierarchy is to describe the A-hierarchy for any process of the type $\widehat{1}[C.P]$ where P is an atomical process and C is a chain of prefixes. The atomical A-hierarchies have the same type of spatial structure.

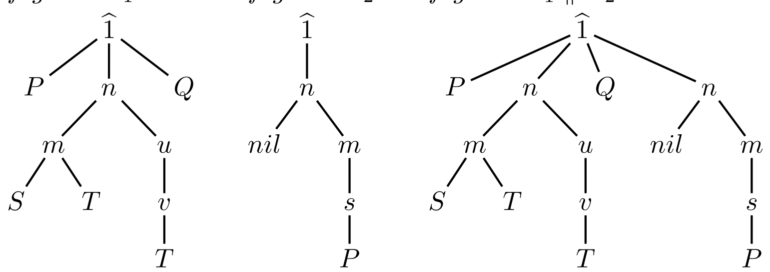
Definition 6.1. We call *atomical A-hierarchy* the A-hierarchy $(\!|G|\!)$ where $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle \{\alpha\}, \{p\}, e, \alpha \rangle$, $e(\alpha) = \{p\}$ and $\mathcal{L}(\alpha) = \widehat{1}$. If $\mathcal{L}(p) = (P, C) \in \mathcal{M}_{pr}$ then we denote *the atomical A-graph* G by $\{C.P\}_p$ and $(\!|G|\!)$ by $(\!|C.P|\!)$. If $C = \varepsilon$ we denote $\{\varepsilon.P\}_p$ by $\{P\}_p$ and its A-hierarchy by $(\!|P|\!)$

Observe that the atomical A-hierarchies are correctly defined, because for all p, p' we have $\{C.P\}_p \cong \{C.P\}_{p'}$.

6.2 Parallel composition of A-Hierarchies

We define the parallel composition of two A-hierarchies. Consider the A-hierarchies in fig. 19 and 20. Then, intuitively, we intend to define the hierarchy described in fig.21 as the parallel composition of them.

$fig.19 : G_1$ $\quad$ $fig.20 : G_2$ $\quad$ $fig.21 : G_1 \parallel G_2$



A special care must be paid for the private and bound names in order to avoid clashing of names.

Definition 6.2. Let $H_1, H_2 \in \mathcal{A}$ and $G_i \in H_i$, $i = 1, 2$. Suppose that $G_i = \langle \mathcal{E}_i, \mathcal{L}_i \rangle$ with $\mathcal{E}_i = \langle X_i, A_i, e_i, \alpha_i \rangle$ and $(X_1 \cup A_1) \cap (X_2 \cup A_2) = \emptyset$[13]. Let $\alpha \in \mho \setminus (X_1 \cup A_1 \cup X_2 \cup A_2)$.
Consider, for each $n \in \Lambda$ such that $n_i^{bn} \in bn(G_2)$, a set $R_n \subset n_I^{bn} \setminus bn(G_1)$ such that $cardR_n = card(arity_{G_2}^{bn}(n))$. Let $R = \cup_n R_n$.
Consider also, for each $n \in \Lambda$ such that $n_i^{pn} \in pn(G_2)$, a set $S_n \subset n_I^{pn} \setminus pn(G_1)$ such that $cardS_n = card(arity_{G_2}^{pn}(n))$. Let $S = \cup_n S_n$.

We construct the A-graph $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ (called *the parallel composition of the A-graphs* G_1 *and* G_2, denoted by $G_1 \parallel G_2$) by:

1. $X = (X_1 \setminus \{\alpha_1\}) \cup (X_2 \setminus \{\alpha_1\}) \cup \{\alpha\}$ and $A = A_1 \cup A_2$,
2. $e : X \to \mathcal{P}(X \cup A)$ by $\quad e(\alpha) = \bigcup_i e_i(\alpha)$ and $e(u) = e_i(u)$ for $u \in X_i \setminus \{\alpha_i\}$
3. $\mathcal{L} : X \cup A \to \mathcal{M}$ by

$$\mathcal{L}(u) = \begin{cases} (pn(G_2) \leftrightarrow S) \circ (bn(G_2) \leftrightarrow R) \circ \mathcal{L}_2(u) & \text{if } u \in X_2 \cup A_2 \\ \mathcal{L}_1(u) & \text{if } u \in X_1 \cup A_1 \end{cases}$$

Remark 6.1. The idea that motivates the first condition in the definition of $\mathcal{L}$ is that we need to replace, in the second process, all the private and input names with unused ones. So, we have $bn(G_1 \parallel G_2) = bn(G_1) \cup R$, and $pn(G_1 \parallel G_2) = pn(G_1) \cup S$. If $pn(G_1) \cap pn(G_2) = \emptyset$ (respectively $bn(G_1) \cap bn(G_2) = \emptyset$) then we can use $S = pn(G_2)$ (respectively $R = bn(G_2)$).

Theorem 6.2. *Suppose that* G_1', G_2', *and* G_1'', G_2'' *are two couples of A-graphs, each couple satisfying the requirements in definition 6.2. If* $G_1' \cong G_1''$ *and* $G_2' \cong G_2''$ *then* $G_1' \parallel G_2' \cong G_1'' \parallel G_2''$.

This result makes possible the following definition.

[13]By Lemma 6.1, for any A-hierarchies H_1, H_2 we can choose such A-graphs.

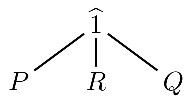
Definition 6.3. Let H_1, H_2 be two A-hierarchies and $G_1 \in H_1, G_2 \in H_2$ be two A-graphs satisfying the conditions in the definition 6.2. We call the A-hierarchy $(\!|G_1 \parallel G_2|\!)$ *the parallel composition of the hierarchies* H_1 *and* H_2 and we denote it by $H_1 \parallel H_2$.

Theorem 6.3. *By construction, the relation* $\parallel$ *on A-hierarchies is associative and commutative. By construction, the relation* $\parallel$ *on A-graphs (satisfying the requirements of the construction), is associative and commutative modulo A-graphs congruence.*

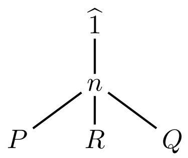
6.3 Root composition

We define here the way in which we can embed a given A-graph in a given location. Suppose that we have the A-graph in fig.22, and we want to embed it in a location labelled by $m = (n, \varepsilon)$. We obtain than the A-graph described in fig. 23.

$fig.22 : G$



$fig.23 : (n, \varepsilon)\lfloor G \rceil$



Definition 6.4. Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ be an A-graph, $\beta \in \mho \setminus (X \cup A)$, and $m = (n, \varepsilon) \in \mathcal{M}_{amb} \setminus \{\widehat{1}\}$ with $n \in \Lambda$. We construct the A-graph $G' = \langle \mathcal{E}', \mathcal{L}' \rangle$ with $\mathcal{E}' = \langle X', A', e', \alpha \rangle$ (called *the* m_β*-rooted A-graph of* G, denoted by $m_\beta \lfloor G \rceil$) by:

1. $X = X \cup \{\beta\}$, and $A' = A$,
2. $e' : X' \to \mathcal{P}(X' \cup A')$ by $e'(\alpha) = \{\beta\}, e'(\beta) = e(\alpha)$, and $e' = e$ on $X \setminus \{\alpha\}$
3. $\mathcal{L}' : X' \cup A' \to \mathcal{M}$ by $\mathcal{L}'(\alpha) = \widehat{1}, \mathcal{L}'(\beta) = m$ and $\mathcal{L}' = \mathcal{L}$ on $(X \setminus \{\alpha\}) \cup A$

Theorem 6.4. *Let* G_1, G_2 *be two A-graphs,* $m = (n, \varepsilon) \in \mathcal{M}_{amb} \setminus \{\widehat{1}\}$ *and* $\beta' \in \mho \setminus (X_1 \cup A_1)$, $\beta'' \in \mho \setminus (X_2 \cup A_2)$. *If* $G_1 \cong G_2$ *then* $m_{\beta'} \lfloor G_1 \rceil \cong m_{\beta''} \lfloor G_2 \rceil$.

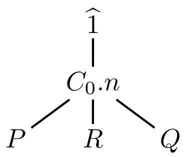
This result makes possible the definition:

Definition 6.5. We call the A-hierarchy associated with the A-graph G' constructed before *the m-rooted hierarchy of* $(\!|G|\!)$. We denote it[14] by $m\lfloor (\!|G|\!) \rceil$.

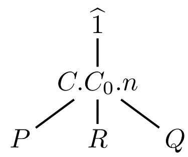
6.4 Capability composition

We are interested as well in the possibility to add a sequence C of capabilities in front of a given A-graph G, as in fig.24,25.

$fig.24 : G$



$fig.25 : C.G$



Definition 6.6. Let $G \in H \in \mathcal{A}$. Suppose that $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ and $e(\alpha) = \{\beta\}$ with $\mathcal{L}(\beta) = (n, C_0) \in \mathcal{M}_{pr} \cup \mathcal{M}_{amb}$. Let $C \in Cap$. We construct the A-graph $G' = \langle \mathcal{E}, \mathcal{L}' \rangle$, denoted by $C.G$, having the same flat system as G but different labels by:

1. If $C = c$ consists of only one capability, then
   (a) if $c = \varepsilon$, we define $\varepsilon.G \stackrel{def}{=} G$
   (b) if $c = \langle M \rangle$, we define $\langle M \rangle.G$ iff $G = \{nil\}_p$, and, in this case, $\langle M \rangle.\{nil\}_p \stackrel{def}{=} \{\langle M \rangle.nil\}_p$

[14] Note that the choice of β is important only for A-graphs, but not for A-hierarchies.

(c) if $c = (n_i^{bn})$, we define $\mathcal{L}'(\beta) = (n_i^{bn} \leftrightarrow n_j^{bn})(n, \langle (n_i^{bn}), C_0 \rangle)$, and $\mathcal{L}' = (n_i^{bn} \leftrightarrow n_j^{bn}) \circ \mathcal{L}$, where $j \in \mathbb{N} \setminus arity_G^{bn}(n)$

(d) if $c \notin \{(n_i^{bn}), \varepsilon, \langle M \rangle\}$, we define $\mathcal{L}'(\beta) = (n, \langle c, C_0 \rangle)$ and $\mathcal{L}' = \mathcal{L}$ on $(X \cup A) \setminus \{\beta\}$

2. If $C = \langle C', C'' \rangle$ then $C.G \overset{def}{=} C'.(C''.G)$

Theorem 6.5. *If G, G' are two A-graphs satisfying the requirements of the definition 6.6, $C \in Cap$ and $G \cong G'$, then $C.G \cong C.G'$.*

This result allows us to propose the next definition.

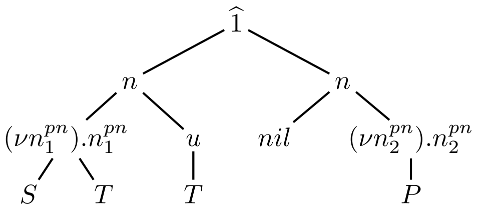
Definition 6.7. Let $G \in H \in \mathcal{A}$, and $C \in Cap$. We call the A-hierarchy $(\!| C.G |\!)$ the capability composition of H with C, and we denote it by $C.H$.

6.5 The private labels

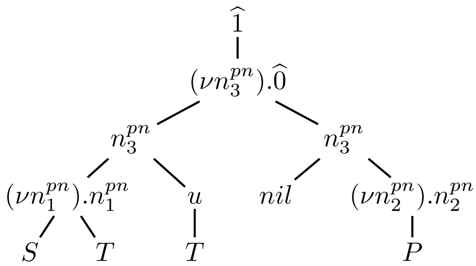
Definition 6.8. Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ be an A-graph with $e(\alpha) = \{\beta\}$, $\mathcal{L}(\beta) = (m, C) \in \mathcal{M}_{pr} \cup \mathcal{M}_{amb}$, and $n \in \Lambda \setminus \{\widehat{1}, \widehat{0}\}$. We denote by $(new\ n)G$ the A-graph $G' = \langle \mathcal{E}, \mathcal{L}' \rangle$ with $\mathcal{E}$ identical with the flat system of G and $\mathcal{L}'$ defined by $\mathcal{L}'(\beta) = (n \leftrightarrow n_i^{pn})(m, \langle (\nu n), C \rangle)$ and $\mathcal{L}' = (n \leftrightarrow n_i^{pn}) \circ \mathcal{L}$ in rest, where $i \notin arity_G^{pn}(n)$.
We extend this definition by $(new\ n)(G_1 \parallel G_2) \overset{def}{=} (new\ n)(\widehat{0}[G_1 \parallel G_2])$

Such a construction is represented in the next example. Having $arity_G^{pn}(n) = \{1, 2\}$, we use the substitution $(n \leftrightarrow n^3)$ to construct $(new\ n).G$.

$fig.26 : G$



$fig.27 : (new\ n)G$



Theorem 6.6. *Let $G_1 \cong G_2$ and $n \in \Lambda$ then $(new\ n)G_1 \cong (new\ n)G_2$.*

This theorem makes possible the definition of the class $(new\ n)(\!| G |\!) \overset{def}{=} (\!| (new\ n)G |\!)$.

Theorem 6.7. $(new\ n)(new\ m)G \cong (new\ m)(new\ n)G$

6.6 Extended Compositions

If H, H_1, H_2 are A-hierarchies with $G \in H, G_1 \in H_1, G_2 \in H_2$ s.t. G_1, G_2 satisfy the requirements of parallel composition, $C \subset Cap$ and $m = (n, \varepsilon) \in \mathcal{M}_{amb} \setminus \{\widehat{1}\}$ and $\beta \in \mho$ satisfying the conditions of ambient composition w.r.t. G, we define:

1. $(n, C)_\beta \lfloor G \rceil \overset{def}{=} C.(n, \varepsilon)_\beta \lfloor G \rceil$; $(n, C) \lfloor H \rceil \overset{def}{=} C.(n, \varepsilon) \lfloor H \rceil$

2. $C.(G_1 \parallel G_2) \overset{def}{=} (\widehat{0}, C)_\beta \lfloor G_1 \parallel G_2 \rceil$; $C.(H_1 \parallel H_2) \overset{def}{=} (\widehat{0}, C) \lfloor H_1 \parallel H_2 \rceil$

6.7 Generalized Parallel composition

Definition 6.9. Let $H_1, H_2, ...H_n, ... \in \mathcal{A}$ be a denumerable class of distinct A-hierarchies. For each i we choose $G_i = \langle \mathcal{E}_i, \mathcal{L}_i \rangle$ with $\mathcal{E}_i = \langle X_i, A_i, e_i, \alpha \rangle$ such that $(X_i \cup A_i) \cap (X_j \cup A_j) = \emptyset$ for $i \neq j$.
Consider two denumerable classes $(R^i)_i \subset \Lambda_{bn}$ and $(S^i)_i \subset \Lambda_{pn}$ satisfying:
$R^1 = bn(G_1)$, $S^1 = pn(G_1)$,
$R^i = \cup_n R_n^i$ where for each $n \in \Lambda$ with $n_j^{bn} \in bn(G_i)$, we define $R_n^i \subset n_I^{bn} \setminus R^{i-1}$ such that

$cardR^i_n = card(arity^{bn}_{G_i}(n))$,
$S^i = \cup_n S^i_n$ where for each $n \in \Lambda$ with $n^{pn}_j \in pn(G_i)$, we define $S^i_n \subset n^{pn}_I \setminus S^{i-1}$ such that $cardS^i_n = card(arity^{pn}_{G_i}(n))$.
We construct the A-graph $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle, \alpha \notin \cup_{i\in\mathbb{N}}(X_i \cup A_i)$ (called *the generalized parallel composition of the class of A-graphs* $(G_i)_{i\in\mathbb{N}}$, denoted $\|_{i\in\mathbb{N}}\, G_i$) by:

1. $X = \cup_{i\in\mathbb{N}}(X_i \setminus \{\alpha_i\})$ and $A = \cup_{i\in\mathbb{N}} A_i$,
2. $e : X \to \mathcal{P}(X \cup A)$ by $\quad e(\alpha) = \cup_{i\in\mathbb{N}} e_i(\alpha)$ and $\quad e(u) = e_i(u)$ if $u \in X_i \setminus \{\alpha_i\}$
3. $\mathcal{L} : X \cup A \to \mathcal{M}$ by $\mathcal{L}(u) = (pn(G_i) \leftrightarrow S^i) \circ (bn(G_i) \leftrightarrow R^i) \circ \mathcal{L}_i(u)$ if $u \in X_i \cup A_i$

Definition 6.10. We call the A-hierarchy $(\!|\|_{i\in\mathbb{N}}\, G_i|\!)$ *the generalized parallel composition of the class* $(H_i)_{i\in\mathbb{N}}$ *of A-hierarchies* and we denote it by $\|_{i\in\mathcal{N}}\, H_i$. Additionally we define

- $C.(\|_{i\in\mathbb{N}}\, G_i) \overset{def}{=} (\widehat{0}, C)_\beta \lfloor \|_{i\in\mathbb{N}}\, G_i \rceil$
- $C.(\|_{i\in\mathbb{N}}\, H_i) \overset{def}{=} (\widehat{0}, C) \lfloor \|_{i\in\mathbb{N}}\, H_i \rceil$

6.8 Recursive A-Hierarchies

Definition 6.11. Let $G = \langle \mathcal{E}_G, \mathcal{L}_G \rangle$ with $\mathcal{E}_G = \langle X_G, A_G, e_G, \alpha_G \rangle$ and $F = \langle \mathcal{E}_F, \mathcal{L}_F \rangle$ with $\mathcal{E}_F = \langle X_F, A_F, e_F, \alpha_F \rangle$ be two A-graphs with disjoint domains and $P \in \Pi$ an atomical process name involved in G. Suppose that $B = \{p_1, , ...p_k\} = \mathcal{L}_G^{-1}(P \times Cap)$, $\mathcal{L}_G(p_i) = (P, C_i)$ for each $i = \overline{1,k}$ and R, S are constructed as in definition 6.2. Consider a set $\Gamma = \{\gamma_1, ..., \gamma_k\} \subset \mho$ disjoint of the domains of the two A-graphs and cardinal equivalent with B. We denote by $G_\Gamma\{P \leftarrow F\}$ the A-graph $E = \langle \mathcal{E}_E, \mathcal{L}_E \rangle$ with $\mathcal{E}_E = \langle X_E, A_E, e_E, \alpha_E \rangle$ obtained by:

1. $X_E = X_G \cup X_F \cup \Gamma$, $A_E = (A_G \setminus B) \cup A_F$, $\alpha_E = \alpha_G$
2. $e_E(\gamma_i) = \{\alpha_F\}, e_E = e_G\{p_i \leftrightarrow \gamma_i\} \cup e_F$ in rest
3. $\mathcal{L}_E(\gamma_i) = (\widehat{0}, C_i), \mathcal{L}_E(\alpha_F) = \widehat{0}$ and $\mathcal{L}_E = \mathcal{L}_G \cup ((pn(F) \leftrightarrow S) \circ (bn(F) \leftrightarrow R) \circ \mathcal{L}_F|_{A_F \cup X_F \setminus \{\alpha_F\}})$ in rest.

The fig 28, 29, 30 describe this construction.

fig.28 : G

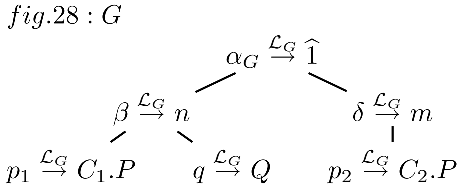


fig.29 : F

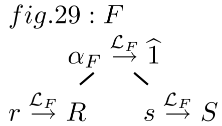
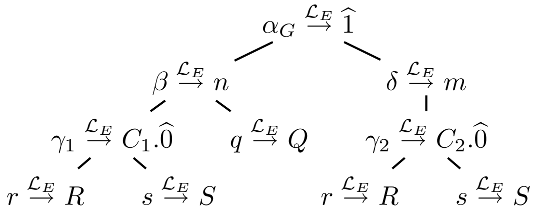


fig.30 : $G_\Gamma\{P \leftarrow F\}$



Theorem 6.8. *Let F, G respectively F', G' be two pairs of A-graphs satisfying the requirements of the previous construction and, additionally, $F \cong F', G \cong G'$. Then $G_\Gamma\{P \leftarrow F\} \cong G'_{\Gamma'}\{P \leftarrow F'\}$.*

This result allows to introduce the next definition.

Definition 6.12. Let H_1, H_2 be two A-hierarchies. We define $H_1\{P \leftarrow H_2\} = (\!|G_\Gamma\{P \leftarrow F\}|\!)$, where $H_1 = (\!|G|\!), H_2 = (\!|F|\!)$.

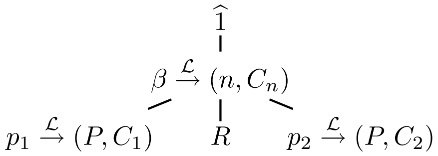
Definition 6.13. Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ be a non-atomical A-graph, $P \in \Pi$ a name involved in it, and $B = \mathcal{L}_1^{-1}(P \times Cap) = \{p_1, ...p_k\} \subseteq A$. Suppose that $\mathcal{L}(p_i) = (P, C_i)$. For a set $\Gamma = \{\gamma_1, ...\gamma_k\} \subset \mho \setminus (X \cup A)$, we define the A-graph $G' = \langle \mathcal{E}', \mathcal{L}' \rangle$ with $\mathcal{E}' = \langle X', A', e', \alpha \rangle$, denoted by $(rec.P)_\Gamma.G$, by:

1. $X' = X \cup \Gamma$, $A' = A \setminus B$
2. $e'(\gamma_i) = (e(\alpha))\{p_j \leftrightarrow \gamma_j\}$ and $e' = e\{p_j \leftrightarrow \gamma_j\}$ in rest
3. $\mathcal{L}'(\gamma_i) = (\widehat{0}, C_i)$ and $\mathcal{L}' = \mathcal{L}$ in rest.

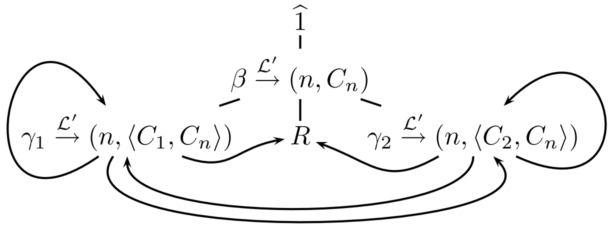
We convey, in addition that $(rec.P)_\Gamma.\{P\}_p \stackrel{def}{=} \{nil\}_p$, that $(rec.P)_\Gamma.\{Q\}_q \stackrel{def}{=} \{Q\}_q$ and that $(rec.P)_\Gamma.G \stackrel{def}{=} G$ if P is not involved in G.

In fig. 31, 32 is represented such a construction.

$fig.31 : G$



$fig.32 : (rec.P)_{\{\gamma_1,\gamma_2\}}.G$



Theorem 6.9. *If Γ and Δ are two cardinal-equivalent sets of urelements fulfilling the conditions of the previous construction, then $(rec.P)_\Gamma.G \cong (rec.P)_\Delta.G$.*

Theorem 6.10. *If $G_1 \cong G_2$ then $(rec.P)_\Gamma.G_1 \cong (rec.P)_\Gamma.G_2$.*

Definition 6.14. We define the A-hierarchy $(rec.P).H = (\!|(rec.P)_\Gamma G|\!)$ where $H = (\!|G|\!)$

Theorem 6.11. *$(rec.P).H \cong H\{P \leftarrow (rec.P).H\}$.*

Corollary 6.12. *$(recP)_\Gamma.G \cong G_\Delta\{P \leftarrow F\}$, where $F \cong (recP)_\Gamma.G$, F and $(recP)_\Gamma.G$ have disjoint domains and Γ, Δ are cardinal equivalent sets.*

6.9 Replication

In this section we consider the replication operator and we provide two distinct construction for it. More precisely, we first handle replication as a generalized parallel composition and then, we consider the recursion operator to implement the replication.

6.9.1 Replication as generalized parallel composition

Definition 6.15. Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ be an A-graph. Consider two denumerable classes $(R^i)_i \subset \Lambda_{bn}$ and $(S^i)_i \subset \Lambda_{pn}$ satisfying:
$R^1 = bn(G)$, $S^1 = pn(G)$,
$R^i = \cup_n R^i_n$ where for each $n \in \Lambda$ with $n^{bn}_j \in bn(G)$, we define $R^i_n \subset n^{bn}_I \setminus R^{i-1}$ such that $cardR^i_n = card(arity^{bn}_G(n))$,
$S^i = \cup_n S^i_n$ where for each $n \in \Lambda$ with $n^{pn}_j \in pn(G)$, we define $S^i_n \subset n^{pn}_I \setminus S^{i-1}$ such that $cardS^i_n = card(arity^{pn}_G(n))$.

Let $(f^i)_{i\in\mathbb{N}}$ be a denumerable class of injective functions $f^i : X \cup A \to \mho$ such that $(X \cup A) \cap f^i(X \cup A) = \emptyset$ and for any $u, v \in X \cup A$ and any $i \neq j$ we have $f^i(u) \neq f^j(v)$.

We call *copies generator of* G the triple $\langle f_i, R^i, S^i \rangle_{i\in\mathbb{N}}$. We call *the i^{th} copy of G generated by this copies generator* the A-graph $G_i = \langle \mathcal{E}_i, \mathcal{L}_i \rangle$ with $\mathcal{E}_i = \langle X_i, A_i, e_i, \alpha_i \rangle$ such that $G \cong G_i$ by the substitution $\zeta = f^i$, bisimulation defined by $x\mathfrak{R}x'$ iff $x' = f^i(x)$, and $\phi = (pn(G) \leftrightarrow S_i) \circ (bn(G) \leftrightarrow R_i) \circ id$.

This definition gives a way to generate, for each $i \in \mathbb{N}$ a copy G_i of a given A-graph G with a disjunct domain. The same is true for any two different G_i and G_j. This mean that the class $(G_i)_{i\in\mathbb{N}}$ satisfies the requirements of the generalized parallel composition.

Hereby we define $G^\infty \stackrel{def}{=} \|_{i\in\mathbb{N}} G_i$. Of course, this definition depends on the copies generator $\langle f_i, R^i, S^i\rangle_{i\in\mathbb{N}}$, but the class $(\!|G^\infty|\!)$ is independent of this. For this reason, if $H = (\!|G|\!)$, we denote $(\!|G^\infty|\!)$ by H^∞.

Theorem 6.13. 1. $G \parallel G^\infty \cong (G^\infty)^\infty \cong G^\infty$,   2. $H \parallel H^\infty = (H^\infty)^\infty = H^\infty$

6.9.2 Replication as recursion

Definition 6.16. Let $G = \langle \mathcal{E}, \mathcal{L}\rangle$ with $\mathcal{E} = \langle X, A, e, \alpha\rangle$ be an A-graph, $P \in \Pi$ a name not involved in G and $p, \gamma \in \mho \setminus (X \cup A), p \neq \gamma$. We define the A-graph $\widehat{rec}G \stackrel{def}{=} (recP)_{\{\gamma\}}.(\{P\}_p \parallel G)$ and, similarly, the A-hierarchy $\widehat{rec}H \stackrel{def}{=} (recP).(\{P\} \parallel H)$.

Theorem 6.14. 1. $G \parallel \widehat{rec}G \cong \widehat{rec}G$,   2. $H \parallel \widehat{rec}H = \widehat{rec}H$

7 The regular representation and equivalence of A-hierarchies

In this section we define an equivalence relation on A-hierarchies to provide a correct model for the structural congruence on ambient processes. Such a definition, as the proofs in the next sections, works inductively on the structures of A-hierarchies. For this purpose, in the first subsection we prove that each A-hierarchy can be decomposed in atomical A-hierarchies.

7.1 The representation theorem

The results of this sub-section show that a complex A-graph (A-hierarchy) can be decomposed in simpler ones, up to the level of atomical A-graphs (A-hierarchies), by using the algebraical structures defined in the previous section.

Definition 7.1. We define the representation of an A-graph by:

1. If $G \cong G_1 \parallel ... \parallel G_k$ we say that G can be represented by $G_1, ...G_k$
2. If $G \cong \|_{i\in\mathbb{N}} G_i$ we say that G can be represented by $(G_i)_{i\in\mathbb{N}}$.
3. If $G \cong m_\beta\lfloor G'\rceil$, or $G \cong C.G'$, or $G \cong (new\ n)G'$ we say that G can be represented by G'
4. If $G \cong (recP)_\Gamma G'$, we say that G can be represented by G'
5. If G can be represented by $G_1, ...G_k$ and if G_1 can be represented by $F_1, ..., F_p$, we say that G can be represented by $F_1, ...F_p, G_2, ...G_k$.

The previous 5 conditions define *the algebraical representation*. The conditions 1,3,5 define *the simple representation*. The conditions 1,2,3,5 define *the denumerable representation*. The conditions 1,3,4,5 define *the recursive representation*.

We extend these representation concepts canonically from A-graphs to A-hierarchies.

Theorem 7.1 (The representation theorem). *Let $G \in H \in \mathcal{A}$.*

1. *G can be algebraically represented by a set of atomical A-graphs; H can be algebraically represented by a set of atomical A-hierarchies and this representation is unique.*
2. *If G is wellfounded, it can be denumerably represented by a set of atomical A-graphs and H can be denumerably represented by a set of atomical A-hierarchies and this representation is unique.*
3. *If G is hereditarily finite, it can be recursively represented by a set of atomical A-graphs and H can be recursively represented by a set of atomical A-hierarchies and this representation is unique.*
4. *If G is wellfounded hereditarily finite, it can be simply represented by a set of atomical A-graphs and H can be simply represented by a set of atomical A-hierarchies and this representation is unique.*

7.2 The equivalence of A-Hierarchies

A-hierarchies are congruence classes of A-graphs that have bisimilar set-theoretical structures. Still, not all the A-graphs intuitively corresponding to structural congruent ambient processes are in the same A-hierarchy. Indeed, A-graphs containing null processes, or transparent ambients as labels have different structures than those in which these nodes are reduced. Moreover, with the rules (ResAmb) and (ResPar) of the structural congruence on ambient processes, the positions of (νn) in a process can vary, hence the configuration in the system of private names of a process is not fixed. For this reasons we define an equivalence relation on A-hierarchies in order to provide a correct model for the structural equivalence on processes.

Definition 7.2. We define the relation $\approx \subset \mathcal{A} \times \mathcal{A}$ by:

1. $\approx$ associates the A-hierarchies:

   (a) $(new\ n)(H_1 \parallel H_2) \approx H_1 \parallel (new\ n)H_2$ if $n \notin fn(H_1)$
   (b) $(new\ n)(m,\varepsilon)\lfloor H \rceil \approx (m,\varepsilon)\lfloor (new\ n)H \rceil$ if $n \neq m$
   (c) $(new\ n)nil \approx nil$

2. $\approx$ is insensitive with respect to the transparent ambient and null atomical process:

   (a) $\widehat{0}\lfloor H \rceil \approx H$ (b) $H \parallel \{nil\} \approx H$ (c) $\{nil\}^\infty \approx \{nil\}$

3. $\approx$ is closed with respect to structural composition, i.e. if $H_1 \approx H_2$ then:

   (a) $(new\ n)H_1 \approx (new\ n)H_2$ (b) $H_1 \parallel H_3 \approx H_2 \parallel H_3$ (c) $n\lfloor H_1 \rceil \approx n\lfloor H_2 \rceil$
   (d) $C.H_1 \approx C.H_2$ (e) $(recP).H_1 \approx (recP).H_2$
   (f) if $H_i \approx H_i'$ for $i \in \mathbb{N}$ then $\parallel_{i\in\mathbb{N}} H_i \approx \parallel_{i\in\mathbb{N}} H_i'$

4. $\approx$ is equivalence-closed:

   (a) $H \approx H$ (b) $H_1 \approx H_2 \Rightarrow H_2 \approx H_1$ (c) $H_1 \approx H_2, H_2 \approx H_3 \Rightarrow H_1 \approx H_3$

8 The isomorphism

So far we have defined two algebraical structures:

- One over the class of A-graphs, $(\mathcal{G}, \cong, \parallel, \parallel_{i\in\mathbb{N}}, m_\beta\lfloor\ \rceil, M.\ , \widehat{rec}, ^\infty, (new\ n), (rec\ X)_\Gamma.)$,
- One over the class of A-hierarchies, $(\mathcal{A}, \approx, \parallel, \parallel_{i\in\mathbb{N}}, m\lfloor\ \rceil, M.\ , \widehat{rec}, ^\infty, (new\ n), (rec\ X).\ )$

Later in this section we will define also a relation $\Longrightarrow$ between A-hierarchies to simulate the reduction relation on ambient processes.

We identify a few classes of important A-hierarchies. We denote by $\mathcal{A}_{wf}$, the class of wellfounded A-hierarchies, by $\mathcal{A}^{fin}$, the class of hereditarily finite A-hierarchies, and by $\mathcal{A}^{fin}_{wf}$, the class of hereditarily finite wellfounded A-hierarchies.

In this section we prove that:

- the algebraical structure of classical Ambient Calculus without replication $(\mathfrak{P}, \equiv, |, m[\ ], M.\ , (\nu n), \rightarrow)$ is isomorphic with the structure of wellfounded hereditarily finite A-hierarchies $(\mathcal{A}^{fin}_{wf}, \approx, \parallel, m\lfloor\ \rceil, M.\ , (new\ n), \Longrightarrow)$,
- the algebraical structure of classical Ambient Calculus enriched with denumerable parallel composition $(\mathfrak{P}^\infty, \equiv, |, |_{i\in\mathbb{N}}, m[\ ], M.\ , !, (\nu n) \rightarrow)$ is isomorphic with the structure of wellfounded A-hierarchies if we interpret the replication of A-hierarchies as denumerable parallel composition $(\mathcal{A}_{wf}, \approx, \parallel, \parallel_{i\in\mathbb{N}}, m\lfloor\ \rceil, M.\ , ^\infty, (new\ n))$,
- the algebraical structure of Recursive Ambient Calculus without replication $(\mathfrak{P}_{rec}, \equiv, |, m[\ ], M.\ , (\nu n), (rec.X\ \ ), \rightarrow)$ is isomorphic with the structure of hereditarily finite A-hierarchies $(\mathcal{A}^{fin}_{wf}, \approx, \parallel, m\lfloor\ \rceil, M.\ , (new\ n), (rec.X\ \ ), \Longrightarrow)$,

- the algebraical structure of the Recursive Ambient Calculus with denumerable parallel composition, $(\mathfrak{P}^{\infty}_{rec}, \equiv, |, |_{i\in\mathbb{N}}, m[\ ], M.\ , !, (\nu n), (rec.X\ ), \rightarrow)$ is isomorphic with the structure of all class of A-hierarchies if we interpret replication of A-hierarchies as denumerable parallel composition $(\mathcal{A}, \approx, \|, \|_{i\in\mathbb{N}}, m\lfloor\ \rceil, M.\ ,^{\infty}, (new\ n), (rec.X\ ), \Longrightarrow)$.

Note that we avoided to interpret the replication of A-hierarchies in the recursive paradigm, because the recursivity is neither distributive, nor idempotent, i.e. we do not have $\widehat{rec}(H_1 \parallel H_2) \approx \widehat{rec}H_1 \parallel \widehat{rec}H_2$ or $\widehat{rec}(\widehat{rec}H) \approx \widehat{rec}H$. Later when we extend the operational semantics of Ambient Calculus in the section 9, we will also extend the relation $\approx$. With this extension, we will prove in the section 10, the isomorphisms are preserved also when replication is interpreted as recursion.

For the moment we focus on the discussed structures. We now define inductively two functions, $[\![\ ]\!]$ that associates to each recursive ambient process P an A-hierarchy $[\![P]\!]$ and $\ulcorner\ \urcorner$ that associates to each A-hierarchy H, a recursive ambient process $\ulcorner H\urcorner$. Then restrictions of these functions to appropriate domains give us the wanted isomorphisms.

Definition 8.1.

$[\![\ ]\!] : \mathfrak{P}^{\infty}_{rec} \rightarrow \mathcal{A}$	$\ulcorner\ \urcorner : \mathcal{A} \rightarrow \mathfrak{P}^{\infty}_{rec}$
1. $[\![P]\!] \stackrel{def}{=} \{P\}$, if P atomical process	1'. $\ulcorner\{P\}\urcorner \stackrel{def}{=} P$
2. $[\![P\|Q]\!] \stackrel{def}{=} [\![P]\!] \parallel [\![Q]\!]$	2'. $\ulcorner H_1 \parallel H_2\urcorner \stackrel{def}{=} \ulcorner H_1\urcorner\|\ulcorner H_2\urcorner$
3. $[\![C.P]\!] \stackrel{def}{=} C.[\![P]\!]$	3'. $\ulcorner C.H\urcorner \stackrel{def}{=} C.\ulcorner H\urcorner$
4. $[\![n[P]]\!] \stackrel{def}{=} (n,\varepsilon)\lfloor[\![P]\!]\rceil$, $n\in\Lambda$	4'. $\ulcorner(n,\varepsilon)\lfloor H\rceil\urcorner \stackrel{def}{=} n[\ulcorner H\urcorner]$, $n\in\Lambda$
5. $[\![(\nu n)P]\!] \stackrel{def}{=} (new\ n)[\![P]\!]$	5'. $\ulcorner(new\ n)H\urcorner \stackrel{def}{=} (\nu n)\ulcorner H\urcorner$
6. $[\![(rec\ X.P)]\!] \stackrel{def}{=} (rec\ X).[\![P]\!]$	6'. $\ulcorner(rec\ X).H\urcorner \stackrel{def}{=} (rec\ X.\ulcorner H\urcorner)$
7. $[\![P_i\|_{i\in\mathbb{N}}]\!] \stackrel{def}{=} [\![P_i]\!] \parallel_{i\in\mathbb{N}}$	7'. $\ulcorner H_i \parallel_{i\in\mathbb{N}}\urcorner \stackrel{def}{=} \ulcorner H_i\urcorner\|_{i\in\mathbb{N}}$
8. $[\![!P]\!] \stackrel{def}{=} [\![P]\!]^{\infty}$	8'. $\ulcorner H^{\infty}\urcorner \stackrel{def}{=} !\ulcorner H\urcorner$

Theorem 8.1. *If $P \in \mathfrak{P}^{\infty}_{rec}$ and $H \in \mathcal{A}$ then $\ulcorner[\![P]\!]\urcorner = P$ and $[\![\ulcorner H\urcorner]\!] = H$.*

Corollary 8.2. *1. $[\![\mathfrak{P}]\!] = \mathcal{A}^{fin}_{wf}$ and $\ulcorner\mathcal{A}^{fin}_{wf}\urcorner = \mathfrak{P}$*
2. $[\![\mathfrak{P}^{\infty}]\!] = \mathcal{A}_{wf}$ and $\ulcorner\mathcal{A}_{wf}\urcorner = \mathfrak{P}^{\infty}$
3. $[\![\mathfrak{P}_{rec}]\!] = \mathcal{A}^{fin}$ and $\ulcorner\mathcal{A}^{fin}\urcorner = \mathfrak{P}_{rec}$
4. $[\![\mathfrak{P}^{\infty}_{rec}]\!] = \mathcal{A}$ and $\ulcorner\mathcal{A}\urcorner = \mathfrak{P}^{\infty}_{rec}$

8.1 The first theorem of isomorphism: The structures

Theorem 8.3. *1. If $P, Q \in \mathfrak{P}^{\infty}_{rec}$ and $P \equiv Q$ then $[\![P]\!] \approx [\![Q]\!]$.*
2. If $H_1, H_2 \in \mathcal{A}$ and $H_1 \approx H_2$ then $\ulcorner H_1\urcorner \equiv \ulcorner H_2\urcorner$.

Hereafter we denote by $\stackrel{iso}{\backsim}$ two isomorphic structures. As a consequence of the definition 8.1 and of the theorems 8.1, 8.2, and 8.3 we obtain the next result:

Theorem 8.4 (The first theorem of isomorphism). *Being the notations proposed before,*
1. The algebraical structure of classical Ambient Calculus without replication is isomorphic with the algebraical structure of wellfounded hereditarily finite A-hierarchies by the function $[\![\ ]\!]$.

$$(\mathfrak{P}, \equiv, |, m[\ ], M.\ , (\nu n)) \stackrel{iso}{\backsim} (\mathcal{A}^{fin}_{wf}, \approx, \|, m\lfloor\ \rceil, M.\ , (new\ n))$$

2. The algebraical structure of Ambient Calculus with denumerable parallel composition is isomorphic with the structure of wellfounded A-hierarchies by the function $[\![\ ]\!]$ if we interpret replication of A-hierarchies by denumerable parallel composition.

$$(\mathfrak{P}^{\infty}, \equiv, |, |_{i\in\mathbb{N}}, m[\ ], M.\ , !, (\nu n)) \stackrel{iso}{\backsim} (\mathcal{A}_{wf}, \approx, \|, \|_{i\in\mathbb{N}}, m\lfloor\ \rceil, M.\ ,^{\infty}, (new\ n)).$$

3. The algebraical structure of Recursive Ambient Calculus without replication is isomorphic with the structure of hereditarily finite A-hierarchies by the function $[\![\ ]\!]$.

$$(\mathfrak{P}_{rec}, \equiv, |, m[\ ], M.\ , (\nu n), (rec.X\ ))\overset{iso}{\backsim}(\mathcal{A}^{fin}, \approx, \|, m\lfloor\ \rceil, M.\ , (new\ n), (rec.P\ )).$$

4. The algebraical structure of Recursive Ambient Calculus with denumerable parallel composition is isomorphic with the structure of A-hierarchies by the function $[\![\ ]\!]$ *if we interpret replication of A-hierarchies by denumerable parallel composition.*

$$(\mathfrak{P}^{\infty}_{rec}, \equiv, |, |_{i\in\mathbb{N}}, m[\ ], M.\ , !, (\nu n), (rec.X\ ))\overset{iso}{\backsim}(\mathcal{A}, \approx, \|, \|_{i\in\mathbb{N}}, m\lfloor\ \rceil, M.\ ,^{\infty}, (new\ n), (rec.P\ )).$$

8.2 The second theorem of isomorphism: The reductions

Before we studied ambient processes as *statical entities*, now we go further to analyze how reductions impact on the structure.

Definition 8.2. Consider the relation $\Longrightarrow \subseteq \mathcal{A} \times \mathcal{A}$ over the class of A-hierarchies by:

1. **(In-Rule)**:$(m,\varepsilon)\lfloor in\ n.H_1 \parallel H_2\rceil \parallel (n,\varepsilon)\lfloor H_3\rceil \Longrightarrow (n,\varepsilon)\lfloor H_3 \parallel (m,\varepsilon)\lfloor H_1 \parallel H_2\rceil\rceil$
2. **(Out-Rule)**:$(n,\varepsilon)\lfloor H_3 \parallel (m,\varepsilon)\lfloor out\ n.H_1 \parallel H_2\rceil\rceil \Longrightarrow (m,\varepsilon)\lfloor H_1 \parallel H_2\rceil \parallel (n,\varepsilon)\lfloor H_3\rceil$
3. **(Open-Rule)**:$open\ n.H_1 \parallel (n,\varepsilon)\lfloor H_2\rceil \Longrightarrow H_1 \parallel H_2$
4. **(Comm-Rule)**:$(x).H \parallel \langle M\rangle.\{nil\} \Longrightarrow H\{x \leftarrow M\}$
5. **(Par-Rule)**:If $H_1 \Longrightarrow H_2$ then $H_1 \parallel H \Longrightarrow H_2 \parallel H$
6. **(DenPar-Rule):**If $H_i \Longrightarrow H_i'$ for each $i \in \mathbb{N}$, then $H_i \parallel_{i\in\mathbb{N}} \Longrightarrow H_i' \parallel_{i\in\mathbb{N}}$
7. **(Amb-Rule)**:If $H_1 \Longrightarrow H_2$ then $m\lfloor H_1\rceil \Longrightarrow m\lfloor H_2\rceil$
8. **(New-Rule):**If $H_1 \Longrightarrow H_2$ then $(new\ n)H_1 \Longrightarrow (new\ n)H_2$
9. **(Closure-Rule)**:If $H_1' \Longrightarrow H_2'$, $H_1' \approx H_1''$ and $H_2' \approx H_2''$ then $H_1'' \Longrightarrow H_2''$

We denote by $\Longrightarrow^{+}$ the transitive closure of $\Longrightarrow$.

Theorem 8.5 (The second theorem of isomorphism). *Being the notations proposed before,*
1. $(\mathfrak{P}, \longrightarrow) \overset{iso}{\backsim} (\mathcal{A}^{fin}_{wf}, \Longrightarrow)$*, the isomorphism being defined by* $[\![\ ]\!]$.
2. $(\mathfrak{P}^{\infty}, \longrightarrow) \overset{iso}{\backsim} (\mathcal{A}_{wf}, \Longrightarrow)$*, the isomorphism being defined by* $[\![\ ]\!]$.
3. $(\mathfrak{P}_{rec}, \longrightarrow) \overset{iso}{\backsim} (\mathcal{A}^{fin}, \Longrightarrow)$*, the isomorphism being defined by* $[\![\ ]\!]$.
4. $(\mathfrak{P}^{\infty}_{rec}, \longrightarrow) \overset{iso}{\backsim} (\mathcal{A}, \Longrightarrow)$*, the isomorphism being defined by* $[\![\ ]\!]$.

The first and second theorems of isomorphism allow us to state the following result.

Theorem 8.6 (The general theorem of isomorphism). *Being the notations proposed,*
1. $(\mathfrak{P}, \equiv, |, m[\ ], M.\ , (\nu n), \longrightarrow) \overset{iso}{\backsim} (\mathcal{A}^{fin}_{wf}, \approx, \|, m\lfloor\ \rceil, M.\ , (new\ n), \Longrightarrow)$
2. $(\mathfrak{P}^{\infty}, \equiv, |, |_{i\in\mathbb{N}}, m[\ ], M.\ , !, (\nu n), \longrightarrow)\overset{iso}{\backsim}(\mathcal{A}_{wf}, \approx, \|, \|_{i\in\mathbb{N}}, m\lfloor\ \rceil, M.\ ,^{\infty}, (new\ n), \Longrightarrow)$.
3. $(\mathfrak{P}_{rec}, \equiv, |, m[\ ], M.\ , (\nu n), (rec.X\ ), \longrightarrow)\overset{iso}{\backsim}(\mathcal{A}^{fin}, \approx, \|, m\lfloor\ \rceil, M.\ , (new\ n), (rec.P\ ), \Longrightarrow)$.
4. $(\mathfrak{P}^{\infty}_{rec}, \equiv, |, |_{i\in\mathbb{N}}, m[\ ], M.\ , !, (\nu n), (rec.X), \longrightarrow)\overset{iso}{\backsim}(\mathcal{A}, \approx, \|, \|_{i\in\mathbb{N}}, m\lfloor\ \rceil, M.\ ,^{\infty}, (new\ n), (rec.P), \Longrightarrow)$

9 Recursive Ambient Calculus

In this section we reconsider the recursive ambient processes as defined in sub-section 3.2. Following the intuition that the recursive ambients are nothing more than non-wellfounded A-hierarchies we will propose an extension of the structural congruence using the patterns of structural unfoldings of processes, a method used in analyzing infinite trees, [4].

9.1 The structural unfolding of an ambient process

For any ambient process we define:

Definition 9.1. Let $P \in \mathfrak{P}^{\infty}$. *The structural depth of* P, $depth(P)$ is defined by:

1. if P is an atomical process $depth(P) \stackrel{def}{=} 0$
2. if $P = \widehat{0}[Q]$ then $depth(P) \stackrel{def}{=} depth(Q)$
3. if $P = n[Q], n \neq \widehat{0}$ then $depth(P) \stackrel{def}{=} depth(Q) + 1$
4. if $P = Q|R$ then $depth(P) \stackrel{def}{=} max(depth(Q), depth(R))$
5. if $P = |_{i\in\mathbb{N}} Q_i$ then $depth(P) \stackrel{def}{=} max(depth(Q_i))$
6. if $P = M.Q$ then $depth(P) \stackrel{def}{=} depth(Q)$
7. if $P = (\nu n)Q$ then $depth(P) \stackrel{def}{=} depth(Q)$
8. if $P =!Q$ then $depth(P) \stackrel{def}{=} depth(Q)$

Remark 9.1. Obviously, if $P \equiv Q$ then $depth(P) = depth(Q)$. Hence, we can consistently add this as a condition on the previous definition in order to extend it to $\mathfrak{P}^{\infty}_{rec}$.

Definition 9.2. For $P \in \mathfrak{P}^{\infty}_{rec}$ we define the structural depth of P by the conditions 1-8 of definition 9.1 together with:

9. if $P \equiv Q$ then $depth(P) \stackrel{def}{=} depth(Q)$
10. if X is a free identifier, then $depth(X) \stackrel{def}{=} 0$

Theorem 9.1. *The depth of a classical ambient process (possibly involving denumerable parallel composition) is finite.*

Theorem 9.2 (Classical Ambients vs Recursive Ambients). $\mathfrak{P} \subsetneq \mathfrak{P}_{rec}$*;* $\quad \mathfrak{P}^{\infty} \subsetneq \mathfrak{P}^{\infty}_{rec}$

Proof. Consider the recursive process $(recX.n[X]) \in \mathfrak{P}_{rec}$ with $n \neq \widehat{0}$. Then $(recX.n[X]) \equiv n[(recX.n[X])]$ by (Rec), so $depth(recX.n[X]) = depth(recX.n[X]) + 1$ by the definition 9.1. Hence $depth(X) = \aleph_0$. This result proves that $\mathfrak{P} \subsetneq \mathfrak{P}_{rec}$. We can prove the other case by using the process $|_{i\in\mathbb{N}} P_i$, where $P_i = (recX.n_i[X])$. □

The above theorem says that replication is not expressive enough for describing all the recursive ambients. We now introduce the pattern of unfoldings.

Definition 9.3. Let $P \in \mathfrak{P}^{\infty}_{rec}$. We define, inductively on finite ordinals, *the denumerable pattern of unfoldings of the process* P as being a sequence $(P^{\alpha})_{\alpha<\aleph_0} \subset \mathfrak{P}^{\infty}$ with the properties:

1. if P is an atomical process then $P^{\alpha} \stackrel{def}{=} P$
2. if $P = c.Q$ then $P^{\alpha} \stackrel{def}{=} c.Q^{\alpha}$
3. if $P = Q|R$ then $P^{\alpha} \stackrel{def}{=} Q^{\alpha}|R^{\alpha}$
4. if $P = |_{i\in\mathbb{N}} Q_i$ then $P^{\alpha} \stackrel{def}{=} |_{i\in\mathbb{N}} Q_i^{\alpha}$
5. if $P = n[Q]$ then $P^0 \stackrel{def}{=} nil$ and for $\alpha > 0$, $P^{\alpha} \stackrel{def}{=} n[Q^{\alpha-1}]$
6. if $P = (\nu n)Q$ then $P^{\alpha} \stackrel{def}{=} (\nu n)Q^{\alpha}$
7. if $P =!Q$ then $P^{\alpha} \stackrel{def}{=} !Q^{\alpha}$
8. if $P = (recX.Q)$ then $P^0 \stackrel{def}{=} nil$, and for $\alpha > 0$ $P^{\alpha} \stackrel{def}{=} Q^{\alpha-1}\{X^{\beta} \leftarrow P^{\beta}\}$ for $\beta \leq \alpha - 1$.

We refer to the process $P^{\alpha} \in \mathfrak{P}^{\infty}$ as *the unfolding of rank* $\alpha < \aleph_0$ *of* P.

Example 9.3. *Consider the next process where* Q, R, S, T *are atomical processes and* C_i *prefixes*
$P = \widehat{1}[C_1.n[C_2.R|C_3.m[C_4.s[C_5.T]|C_6.S]|C_7.m[C_8.u[C_9.v[C_{10}.Q]|C_{11}.T]]|C_{12}.R]]$*, then*
$P^0 = nil$,
$P^1 = \widehat{1}[C_1.nil]$,
$P^2 = \widehat{1}[C_1.n[C_2.R|C_3.nil|C_7.nil|C_{12}.R]]$,
$P^3 = \widehat{1}[C_1.n[C_2.R|C_3.m[C_4.nil|C_6.S]|C_7.m[C_8.nil]|C_{12}.R]]$,
$P^4 = \widehat{1}[C_1.n[C_2.R|C_3.m[C_4.s[C_5.T]|C_6.S]|C_7.m[C_8.u[C_9.nil|C_{11}.T]]|C_{12}.R]]$,
$P^5 = \widehat{1}[C_1.n[C_2.R|C_3.m[C_4.s[C_5.T]|C_6.S]|C_7.m[C_8.u[C_9.v[C_{10}.Q]|C_{11}.T]]|C_{12}.R]]$
and hereafter the sequence is stationary and equal with P.

Consider now the process $P = (recX.m[X])$ then
$P^0 = (recX.m[X])^0 = (m[(recX.m[X])])^0 = nil$
$P^1 = (m[(recX.m[X])])^1 = m[P^0] = m[nil]$
$P^2 = (m[(recX.m[X])])^2 = m[m[P^0]] = m[m[nil]]$
$P^3 = (m[(recX.m[X])])^3 = m[m[m[P^0]]] = m[m[m[nil]]]$, *etc*
the sequence will never be stationary.

Remark 9.2. Observe that a pattern of unfoldings is nothing more than a denumerable sequence of wellfounded ambient processes. We denote by $Seq(\mathfrak{P}^\infty)$ the class of these sequences. Obviously, for a given $P \in \mathfrak{P}^\infty_{rec}$ and a given $\alpha < \aleph_0$, P^α is unique up to identity. In extenso, the pattern of unfoldings of a given process is unique as well. But if we consider two equivalent processes, like P and $Q = \widehat{0}[\widehat{0}[P]]$ then their patterns are not identical anymore. Indeed, if $P^0, P^1, P^2, ...$ is the pattern of P, the pattern of Q is $Q^0 = 0, Q^1 = \widehat{0}[nil] = nil, Q_2 = \widehat{0}[\widehat{0}[P^0]] = P^0, Q^3 = P^1, ...$ In this case, the pattern of Q is "delayed" with respect to the pattern of P. We will define further an equivalence relation over patterns of unfoldings to handle this anomaly.

Corollary 9.4. *For any $P \in \mathfrak{P}^\infty_{rec}$ we have $depth(P^\alpha) \leq \alpha$. If $depth(P) = \alpha < \aleph_0$ then for any $\alpha < \beta < \aleph_0$ we have $P^\beta \equiv P^\alpha$.*

Corollary 9.5. *$(P\{X \leftarrow Q\})^\alpha \equiv P^\alpha\{X^\beta \leftarrow Q^\beta\}$ for all $\beta \leq \alpha$.*

Theorem 9.6. *$(recX.X)^\alpha \equiv nil$*

Corollary 9.7. *If $P \in \mathfrak{P}^\infty$ then $\exists k < \aleph_0, P^k \equiv P$ and $P^{k+s} \equiv P$ for any s.*

9.2 An extension of the operational semantics

With the remark 9.2, we can consider the next definition.

Definition 9.4. The construction of the pattern of unfoldings of a process is a function $u : \mathfrak{P}^\infty_{rec} \to Seq(\mathfrak{P}^\infty)$, and the construction of the α-unfolding is a function $u_\alpha : \mathfrak{P}^\infty_{rec} \to \mathfrak{P}^\infty$.

Definition 9.5. Let $\mathcal{P}' = (P'_i)_{i<\aleph_0}, \mathcal{P}'' = (P''_i)_{i<\aleph_0} \in Seq(\mathfrak{P}^\infty)$, $(\mathcal{P}^i)_{i\in\mathbb{N}} \subseteq Seq(\mathfrak{P}^\infty)$ a denumerable set of sequences, and M a prefix. We define the parallel composition, denumerable parallel composition, ambient composition, and prefixing for sequences of classical ambient processes by:
$| \ : Seq(\mathfrak{P}^\infty) \times Seq(\mathfrak{P}^\infty) \to Seq(\mathfrak{P}^\infty)$, $\mathcal{P}'|\mathcal{P}'' \stackrel{def}{=} (P'_i|P''_i)_{i<\aleph_0}$;
$! \ : Seq(\mathfrak{P}^\infty) \to Seq(\mathfrak{P}^\infty)$, $!\mathcal{P}' \stackrel{def}{=} (!P'_i)_{i<\aleph_0}$
$|_{i\in\mathbb{N}} \ : Seq(\mathfrak{P}^\infty) \times Seq(\mathfrak{P}^\infty) \times Seq(\mathfrak{P}^\infty) \times ... \to Seq(\mathfrak{P}^\infty)$, $|_{i\in\mathbb{N}}\mathcal{P}^i \stackrel{def}{=} (|_{i\in\mathbb{N}}P^i_j)_{j<\aleph_0}$;
$[\,] : Seq(\mathfrak{P}^\infty) \times Seq(\mathfrak{P}^\infty) \to Seq(\mathfrak{P}^\infty)$, $\mathcal{P}'[\mathcal{P}''] \stackrel{def}{=} (P'_i[P''_i])_{i<\aleph_0}$, if $\mathcal{P}'$ is the pattern for a process of the type $n[nil]$;
$M. \ : Seq(\mathfrak{P}^\infty) \to Seq(\mathfrak{P}^\infty)$, $M.\mathcal{P}' \stackrel{def}{=} (M.P'_i)_{i<\aleph_0}$.

Definition 9.6. We call two denumerable sequences of classical ambient processes $\mathcal{P}' = (P'_i)_{i<\aleph_0}, \mathcal{P}'' = (P''_i)_{i<\aleph_0} \in Seq(\mathfrak{P}^\infty)$ equivalent, denoted by $\mathcal{P}' \sim \mathcal{P}''$ if one of the following conditions is satisfied:

1. $P'_1 = ... = P'_k = nil$, and $P'_{k+s} \equiv P''_s$ for any $s < \aleph_0$ or $P''_1 = ... = P''_k = nil$, and $P''_{k+s} \equiv P'_s$ for any $s < \aleph_0$
2. $\mathcal{P}' = \mathcal{Q}'_1|\mathcal{Q}'_2$, $\mathcal{P}'' = \mathcal{Q}''_1|\mathcal{Q}''_2$, $\mathcal{Q}'_1 \sim \mathcal{Q}''_1$ and $\mathcal{Q}'_2 \sim \mathcal{Q}''_2$, where $\mathcal{Q}'_1, \mathcal{Q}''_1, \mathcal{Q}'_2, \mathcal{Q}''_2 \in Seq(\mathfrak{P}^\infty)$
3. $\mathcal{P}' = |_{i\in\mathbb{N}}\mathcal{Q}'_i$, $\mathcal{P}'' = |_{i\in\mathbb{N}}\mathcal{Q}''_i$, $\mathcal{Q}'_i \sim \mathcal{Q}''_i$ for any $i = 1, n$, where $(\mathcal{Q}'_i)_{i\in\mathbb{N}}, (\mathcal{Q}''_i)_{i\in\mathbb{N}} \subset Seq(\mathfrak{P}^\infty)$
4. $\mathcal{P}' = !\mathcal{Q}'$, $\mathcal{P}'' = !\mathcal{Q}''$, and $\mathcal{Q}' \sim \mathcal{Q}''$, where $\mathcal{Q}', \mathcal{Q}'' \in Seq(\mathfrak{P}^\infty)$
5. $\mathcal{P}' = \mathcal{Q}'_1[\mathcal{Q}'_2]$, $\mathcal{P}'' = \mathcal{Q}''_1[\mathcal{Q}''_2]$, $\mathcal{Q}'_1 \sim \mathcal{Q}''_1$ and $\mathcal{Q}'_2 \sim \mathcal{Q}''_2$, where $\mathcal{Q}'_1, \mathcal{Q}''_1, \mathcal{Q}'_2, \mathcal{Q}''_2 \in Seq(\mathfrak{P}^\infty)$
6. $\mathcal{P}' = M.\mathcal{Q}'$, $\mathcal{P}'' = M.\mathcal{Q}''$, and $\mathcal{Q}' \sim \mathcal{Q}''$, where $\mathcal{Q}', \mathcal{Q}'' \in Seq(\mathfrak{P}^\infty)$

Remark 9.3. Since each element P_i of $\mathcal{P}$ is wellfounded, $\mathcal{P}$ must be wellfounded with respect to the relations previously introduced on sequences.

Theorem 9.8. *1.* $u(n[P]) \sim u(n)[u(P)]$
2. $u(P|Q) \sim u(P)|u(Q)$
3. $u(P_i|_{i\in\mathbb{N}}) \sim u(P_i)|_{i\in\mathbb{N}}$
4. $u(!P) \sim !u(P)$
5. $u(M.P) \sim M.u(P)$.

Theorem 9.9. $\sim$ *is an equivalence relation over* $Seq(\mathfrak{P}^\infty)$.

Theorem 9.10. *1. If* $P, Q \in \mathfrak{P}^\infty_{rec}$ *and* $P \equiv Q$ *then* $u(P) \sim u(Q)$.
2. If $P, Q \in \mathfrak{P}^\infty$ *then* $P \equiv Q$ *iff* $u(P) \sim u(Q)$.

Remark 9.4. $\sim$ is a generalization of $\equiv$, for classical ambient processes being equivalent with the structural congruence. The theorem 9.10 suggests using the equivalence between the patterns of unfoldings as a definition for congruence over $\mathfrak{P}^\infty_{rec}$.

Definition 9.7. Let $P, Q \in \mathfrak{P}^\infty_{rec}$. We call the two processes *pattern-structural-congruent* and we write $P \equiv^+ Q$ iff $u(P) \sim u(Q)$, i.e. iff their patterns of unfoldings are equivalent.

This definition gives us the properties of distributivity and idempotence of recursion that could not be derived with the previous rules of structural congruence.

Theorem 9.11 (The distributivity of recursion). *If* P, Q *are processes that do not contain any free occurrence of* X*, then* $(recX.X|P|Q) \equiv^+ (recX.X|P)|(recX.X|Q)$

Proof. Let $R = (recX.X|P|Q)$ then
$R^0 = nil$
$R^1 = (X|P|Q)^0\{X^0 \leftarrow R^0\} \equiv R^0|P^0|Q^0 \equiv P^0|Q^0$
$R^2 = (X|P|Q)^1\{X^1 \leftarrow R^1\} \equiv R^1|P^1|Q^1 \equiv P^0|Q^0|P^1|Q^1$, etc
can be proved by induction that $R^{k+1} \equiv P^0|Q^0|P^1|Q^1|...|P^k|Q^k$.
Let $S = (recX.X|P)$ and $T = (recX.X|Q)$, then
$(S|T)^0 \equiv S^0|T^0 \equiv nil$
$(S|T)^1 \equiv S^1|T^1 \equiv P^0|Q^0$
$(S|T)^2 \equiv S^2|T^2 \equiv P^0|P^1|Q^0|Q^1$, etc
can be proved inductively that $(S|T)^{k+1} \equiv P^0|P^1|...|P^k|Q^0|Q^1|...|Q^k$.
Hence $R^k \equiv (S|T)^k$, so $u(R) \sim u(S|T)$, q.e.d. □

Theorem 9.12 (The idempotence of recursion). *If* X, Y, Z *are not free names in* P *then* $(recX.X|(recY.Y|P)) \equiv^+ (recZ.Z|P)$.

Proof. Let $R = (recX.X|P)$ and $Q = (recY.Y|R)$ and we prove that $R \equiv^+ Q$. We have
$R^0 = nil \equiv nil|nil|nil|...$
$R^1 = (X|P)^0\{X^0 \leftarrow R^0\} \equiv R^0|P^0 \equiv P^0|nil|nil|...$
$R^2 = (X|P)^1\{X^1 \leftarrow R^1\} \equiv R^1|P^1 \equiv P^1|P^0|nil|nil|...$, etc
can be proved by induction that $R^{k+1} \equiv P^k|...|P^1|P^0|nil|nil|....$
$Q^0 = nil \equiv nil|nil|nil|...$
$Q^1 \equiv R^0 = nil \equiv nil|nil|nil|...$
$Q^2 \equiv R^0|R^1 \equiv P^0|nil|nil|nil|...$
$Q^3 \equiv R^0|R^1|R^2 \equiv P^0|(P^0|P^1)|nil|nil|...$
can be proved inductively that $Q^{k+2} = P^0|(P^0|P^1)|(P^0|P^1|P^2)...|(P^0|...|P^k)|nil|nil|...$
If we denote by $\mathcal{Q}$ the sequence of Q^i, by $\mathcal{R}$ the sequence of R^i, and by $\mathcal{P}^i$ the sequence $nil, nil, ...nil, P^0, P^1, ...$ which begins with i of nil then we have just proved that $\mathcal{R} \sim \mathcal{P}^0|\mathcal{P}^1|\mathcal{P}^2|...$ and that $\mathcal{Q} \sim \mathcal{P}^0|(\mathcal{P}^0|\mathcal{P}^1)|(\mathcal{P}^0|\mathcal{P}^1|\mathcal{P}^2)...|(\mathcal{P}^0|...|\mathcal{P}^k)|nil|nil|.....$ But, for each $i < \aleph_0$ we have $u(P) \sim \mathcal{P}^i$, hence $u(!P) \sim !u(P) \sim \mathcal{Q} \sim \mathcal{R}$. q.e.d. □

Now we can prove that $!P$ and $(recX.X|P)$ are (pattern-) structural congruent processes if in P there is no free occurrence of X.

Theorem 9.13. *If in P there is no free occurrence of X then*
1. $(recX.X|P) \equiv^+ !P$ *2.* $(recX.\underbrace{X|X|...|X}_{k\ times}|P) \equiv^+ !P$ *3.* $(recX.!X|P) \equiv^+ !P$

Proof. 1. Let $Q = (recX.X|P)$ then
$Q^0 = (recX.X|P)^0 = nil \equiv^+ nil|nil|nil|...$
$Q^1 = (recX.X|P)^1 = (X|P)^0\{X^0 \leftarrow Q^0\} = (X^0\{X^0 \leftarrow Q^0\})|P^0 = Q^0|P^0 \equiv^+ P^0|nil|nil|...$
$Q^2 = (recX.X|P)^2 = X^1|P^1\{X^1 \leftarrow Q^1\} = Q^1|P^1 \equiv^+ P^1|P^0|nil|nil|...$
$Q^3 = (recX.X|P)^3 = (X^2|P^2)\{X^2 \leftarrow Q^2\} = Q^2|P^2 \equiv^+ P^2|P^1|P^0|nil|nil|...$
In the general case it can be proved inductively on k that $Q^k = P^{k-1}|P^{k-2}|...|P^1|P^0|nil|nil|...$:
Indeed $Q^k = (recX.X|P)^k = (X^{k-1}|P^{k-1})\{X^{k-1} \leftarrow Q^{k-1}\} = Q^{k-1}|P^{k-1}$ where we can use the inductive premises.
If we denote by $\mathcal{Q}$ this sequence and by $\mathcal{P}^i$ the sequence $nil, nil, ...nil, P^0, P^1, ...$ which begins with i of nil then we just proved that $\mathcal{Q} \sim \mathcal{P}^0|\mathcal{P}^1|\mathcal{P}^2|....$ But, for each $i < \aleph_0$ we have $u(P) \sim \mathcal{P}^i$, hence $u(!P) \sim !u(P) \sim \mathcal{Q}$. q.e.d.

2. Let $Q = (recX.\underbrace{X|X|...|X}_{k\ times}|P)$ then

$Q^0 = (recX.\underbrace{X|X|...|X}_{k\ times}|P)^0 = nil \equiv^+ nil|nil|nil|...$

$Q^1 = (recX.\underbrace{X|X|...|X}_{k\ times}|P)^1 = (\underbrace{X|X|...|X}_{k\ times}|P)^0\{X^0 \leftarrow Q^0\} = (\underbrace{X^0|X^0|...|X^0}_{k\ times}\{X^0 \leftarrow Q^0\})|P^0 =$
$\underbrace{Q^0|Q^0|...|Q^0}_{k\ times}|P^0 \equiv^+ P^0|nil|nil|...$

$Q^2 = (recX.\underbrace{X|X|...|X}_{k\ times}|P)^2 = \underbrace{X^1|X^1|...|X^1}_{k\ times}|P^1\{X^1 \leftarrow Q^1\} = \underbrace{Q^1|Q^1|...|Q^1}_{k\ times}|P^1 \equiv^+ P^1|\underbrace{P^0|P^0|...|P^0}_{k\ times}|nil|nil|...$

$Q^3 = (recX.\underbrace{X|X|...|X}_{k\ times}|P)^3 = (\underbrace{X^2|X^2|...|X^2}_{k\ times}|P^2)\{X^2 \leftarrow Q^2\} = \underbrace{Q^2|Q^2|...|Q^2}_{k\ times}|P^2$
$\equiv^+ \underbrace{P^2|P^2|...|P^2}_{2k\ times}|\underbrace{P_1|P_1|...|P_1}_{k\ times}|P^0|nil|nil|...$

In the general case it can be proved inductively on k that
$Q^n = \underbrace{P^{n-1}|P^{n-1}|...|P^{n-1}}_{(n-1)k\ times}|\underbrace{P^{n-2}|P^{n-2}|...|P^{n-2}}_{(n-2)k\ times}|...|\underbrace{P^1|P^1|...|P^1}_{k\ times}|P^0|nil|nil|...$:
If we denote by $\mathcal{Q}$ this sequence and by $\mathcal{P}^i$ the sequence $nil, nil, ...nil, P^0, P^1, ...$ which begins with i of nil then we just proved that $\mathcal{Q} \sim \mathcal{P}^0|\underbrace{\mathcal{P}^1|\mathcal{P}^1|...|\mathcal{P}^1}_{k\ times}|\underbrace{\mathcal{P}^2|\mathcal{P}^2|...|\mathcal{P}^2}_{2k\ times}|....$ But, for each $i < \aleph_0$ we have $u(P) \sim \mathcal{P}^i$, hence $u(!P) \sim !u(P) \sim \mathcal{Q}$ due to the fact that $\aleph_0 \times k = \aleph_0$ for $k < \aleph_0$. q.e.d.

3.For $Q = (recX.!X|P) = (recX.\underbrace{X|X|...}_{\aleph_0}|P)$ we can be proved similarly that

$Q^{k+1} = \underbrace{P^0|P^0|...}_{\aleph_0}|\underbrace{P^1|P^1|...}_{\aleph_0}|...|\underbrace{P^k|P^k|...}_{\aleph_0}$ using the fact that $\aleph_0 \times k = \aleph_0$. Now, using the commutativity of parallel composition we have
$Q^{k+1} = P^0|(P^0|P^1)|(P^0|P^1|P^2)|...(P^0|P^1...|P^k)|(P^0|P^1...|P^k|nil)|(P^0|P^1...|P^k|nil|nil)...$
If we denote by $\mathcal{Q}$ this sequence and by $\mathcal{P}^i$ the sequence $nil, nil, ...nil, P^0, P^1, ...$ which begins with i of nil then we have just proved that $\mathcal{Q} \sim \mathcal{P}^0(\mathcal{P}^0|\mathcal{P}^1)|(\mathcal{P}^0|\mathcal{P}^1|\mathcal{P}^2)|...(\mathcal{P}^0|\mathcal{P}^1...|\mathcal{P}^k)|(\mathcal{P}^0|\mathcal{P}^1...|\mathcal{P}^k|\mathcal{P}^{k+1})|....$
But, for each $i < \aleph_0$ we have $u(P) \sim \mathcal{P}^i$, hence $u(!P) \sim !u(P) \sim \mathcal{Q}$ due to the fact that $\aleph_0 \times \aleph_0 = \aleph_0$. q.e.d. □

Remark 9.5. To remain consistent with the intension of the classical Ambient Calculus, we replace the reduction rule
$(\equiv -Rule):\ P' \equiv P, P \to Q, Q \equiv Q' \Rightarrow P' \to Q'$
with the extended rule:
$(\equiv^+ -Rule):\ P' \equiv^+ P, P \to Q, Q \equiv^+ Q' \Rightarrow P' \to Q'$

9.3 Linear versus nonlinear recursive processes

We use the notation $P \downarrow Q$ to express that $P = n[Q]$ with $n \neq \widehat{0}$ or that $P = M.Q$ with $M \neq \varepsilon$ and we denote by $\downarrow^+$ the transitive closure of it. We use the notation $P \Downarrow Q$ to express that $P = n[Q]$ with $n \neq \widehat{0}$ and we denote by $\Downarrow^+$ the transitive closure of it.

Definition 9.8. Let $(recX.P)$ be a recursive ambient process. We call it *nonlinear* if there is at least one occurrence of X in P such that $P \downarrow^+ X$, otherwise we call it *linear*. We call the process *proper nonlinear* if there is at least one occurrence of X such that $P \Downarrow^+ X$.

With this definition, using the Theorem 9.13 we obtain that always a linear recursive process is (pattern-) structurally congruent with a classical one involving replication, as stated in the next theorem.

Theorem 9.14. *Any linear recursive ambient process is pattern-structural congruent with a classical ambient process (involving replication).*

Theorem 9.15. *If P is a recursive proper nonlinear process, then $depth(P) = \aleph_0$.*

Proof. Let $P = (recX.Q)$ with $Q \Downarrow^+ X$. $depth(P) = depth(Q\{X \leftarrow P\})$ and because $Q \Downarrow^+ X$, we have $depth(P) = depth(P) + k$, where $k > 0$ is the depth where we can find the first occurrence of X. Hence, $depth(P) = \aleph_0$. □

Due to the Theorem 9.1, the depth of a classical ambient process is finite, thus the previous theorem proves that there is no classical ambient process (pattern-) structurally congruent with a proper non-linear recursive process. Hence, the Recursive Ambient Calculus is strictly more expressive than the classical one (even enriched with denumerable parallel composition).

Not only the proper nonlinear processes have structures that cannot be described by classical processes (with replication), but all the nonlinear processes. Consider the process $(recX.c.X|P)$ with $c \neq \varepsilon$. At any level, this process is a parallel composition of P and a complex process guarded by c. It seems impossible to express this process by using denumerable parallel composition and, hence, replication. In order to prove this we will define a new concept of depth of a process that is sensitive to prefixes.

Definition 9.9. Let $P \in \mathfrak{P}^{\infty}_{rec}$. $\overline{depth}(P)$ is defined by:

1. if P is an atomical process $\overline{depth}(P) \stackrel{def}{=} 0$
2. if $P = \widehat{0}[Q]$ then $\overline{depth}(P) \stackrel{def}{=} \overline{depth}(Q)$
3. if $P = n[Q], n \neq \widehat{0}$ then $\overline{depth}(P) \stackrel{def}{=} \overline{depth}(Q) + 1$
4. if $P = Q|R$ then $\overline{depth}(P) \stackrel{def}{=} max(\overline{depth}(Q), \overline{depth}(R))$
5. if $P = |_{i\in\mathbb{N}} Q_i$ then $\overline{depth}(P) \stackrel{def}{=} max_{i\in\mathbb{N}}(\overline{depth}(Q_i))$
6. if $P = M.Q$, $M \neq \varepsilon$ then $\overline{depth}(P) \stackrel{def}{=} \overline{depth}(Q) + 1$
7. if $P = (\nu n)Q$ with $Q \neq nil$ then $\overline{depth}(P) \stackrel{def}{=} \overline{depth}(Q) + 1$, and $\overline{depth}((\nu n)nil) \stackrel{def}{=} nil$
8. if $P = !Q$ then $\overline{depth}(P) \stackrel{def}{=} \overline{depth}(Q)$
9. if $P \equiv Q$ then $\overline{depth}(P) \stackrel{def}{=} \overline{depth}(Q)$
10. if X is a free identifier, then $\overline{depth}(X) \stackrel{def}{=} 0$

Comparing this definition with the definition 9.1 we obtain the following corollary.

Corollary 9.16. *For any process $P \in \mathfrak{P}^{\infty}_{rec}$ $depth(P) \leq \overline{depth}(P)$.*

Theorem 9.17. *If P is a classical ambient process (possibly involving denumerable parallel composition) then $\overline{depth}(P)$ is finite.*

Theorem 9.18. *If P is a nonlinear recursive ambient process, then $\overline{depth}(P) = \aleph_0$.*

Proof. Let $P = (recX.Q)$ with $Q \downarrow^+ X$. $\overline{depth}(P) = \overline{depth}(Q\{X \leftarrow P\})$ and because $Q \downarrow^+ X$, we have $\overline{depth}(P) = \overline{depth}(P) + k$, where $k > 0$ is the depth where we can find the first occurrence of X. Hence, $\overline{depth}(P) = \aleph_0$. □

Concluding:

Corollary 9.19. *1. P is a classical ambient process (possibly a linear recursive one), iff $depth(P) \leq \overline{depth}(P) < \aleph_0$.*
2. P is a nonlinear recursive ambient process iff $\overline{depth}(P) = \aleph_0$.
3. P is a proper nonlinear recursive ambient process iff $depth(P) = \aleph_0$.

Hence the recursive ambient calculus is more expressive than classical ambient calculus (with respect to the models they can describe) because any ambient program (possibly involving replication) can be successfully described by a recursive program, but not any recursive program can be expressed by a classical one (eventually involving replication).

9.4 In Ambient Calculus replication cannot simulate recursion

We prove in this section that replication is not able to simulate recursion even up to a notion of observational behavior (as it happens instead in [16] for the π-Calculus) because more complex models generate more complex behaviors.

Let $P \in \mathfrak{P}^{\infty}_{rec}$ be an ambient process and consider its transition system labelled by the possible actions that P can perform, collected in the set $Act(P)$. Obviously, we consider any two actions that are performed in different locations (ambients) as different. Consider, for example, the process

$$P = u[(\nu n)m[a.n[b.S]|c.Q]|a.T|n[a.Q|b.R]], \ \textit{with}\ a, b, c \in Cap \setminus \{\varepsilon\} \tag{9.1}$$

This process can perform the action a at the location u, the actions a and c at the location m, and the actions a and b at the location denoted by the second ambient named n[15]. The action b, that prefixes the process S inside the first ambient named n, is not active because n is not active due to its prefix a. We distinguish between the different actions a if they can be performed at different locations. If the process P performs the action a at the location u produces a different output than in the case it performs a in m or it performs a in the second ambient n.

If someone would provide an ambient process P' that could simulate the behavior of P, then this process should satisfy the condition $Act(P') = Act(P)$. This is the basic argument in Milner's translation from recursion to replication [16].

Theorem 9.20. *If P is a classical ambient process (possibly involving replication) then its set of active actions is finite or denumerable.*

Proof. We start by observing that each action is performed to a location, being the master ambient or another ambient that is active in our process. The maximum number of possible active processes at each location for a classical ambient processes is $\aleph_0$ (if we accept denumerable parallel composition). We will develop here a maximal deduction in order to determine the maximum number of possible active actions. Hence, we can suppose that inside the master ambient there exist denumerable active processes together with denumerable active ambients. For each of these active ambients we can argue in the same way. So, if this process has the depth 2, then it could have maximum $\aleph_0 \times \aleph_0 = \aleph_0$ active actions, hence denumerable many. Going deeper, to the depth k, the maximum number of possible active actions is $\underbrace{\aleph_0 \times \aleph_0 \times ... \times \aleph_0}_{k\ times}$.
But a classical process has a finite depth, hence $k < \aleph_0$ for any classical process. Thus, the maximum number of active actions is $\underbrace{\aleph_0 \times \aleph_0 \times ... \times \aleph_0}_{k\ times} = \aleph_0$. q.e.d. ☐

The last two theorems allow us to conclude with the following result.

Theorem 9.21. *There exist processes $P \in \mathfrak{P}_{rec}$ having the set of active actions infinite non-denumerable.*

[15] Not necessarily all these actions can be performed because some of them could not fulfill the requirements in the reduction rules. Still this aspect is irrelevant for our point here.

Proof. We prove this theorem by constructing such a process.

Let $P = (recX.n[X]|m[X]|(u)|\langle M\rangle)$. This process can perform an input/output action at any location in its structure tree. But each node of its structure tree contains, as children, two ambients. It grows exponentially having $2^{\aleph_0} = \aleph_1$ nodes. Thus the set of its possible actions has the cardinality $c \geq \aleph_1$, hence it is infinite non-denumerable. q.e.d. □

Theorem 9.22. *There are recursive ambient processes that cannot be simulated by classical ambient processes.*

Proof. As proved in the Theorem 9.21 there are recursive processes that have the set of possible actions infinite non-denumerable. However, conforming with Theorem 9.20, any classical ambient process has the set of possible actions finite or denumerable. Hence the recursive processes identified by the theorem 9.21 cannot be simulated by classical processes. q.e.d. □

Summing up, the Recursive Ambient Calculus is strictly more expressive than the classical one, being able to describe phenomena with much more complex structures, including structures that can have an infinite non-denumerable set of possible actions. Classical Ambient Calculus, even enriched with replication, can describe structures that have finite or denumerable many possible actions.

Because we introduced the denumerable parallel composition only to provide a model for replication, hereafter we consider only calculi without this operator, its action being successfully expressible by recursion.

10 The extended isomorphism

We just proved that $(\mathfrak{P}^!_{rec}, \equiv, !) \overset{iso}{\backsim} (\mathfrak{P}_{rec}, \equiv^+, \overline{rec})$. We propose now an extension of the relation $\approx$ on A-hierarchies by extending the definition 7.2.

Definition 10.1. We define the relation $\approx^* \subset \mathcal{A} \times \mathcal{A}$ by:

1. $\approx^*$ is a supraclass of $\approx$, i.e. $\approx^* \supseteq \approx$
2. the linear recursivity is distributive and idempotent with respect to $\approx^*$:

$$\text{(a) } \widehat{rec}(F \parallel G) \approx^* \widehat{rec}F \parallel \widehat{rec}G, \qquad \text{(b) } \widehat{rec}(\widehat{rec}G) \approx^* \widehat{rec}G$$

3. $\approx^*$ is closed with respect to structural composition and equivalence laws.

We can also extend the isomorphism theorem.

Theorem 10.1 (The extended theorem of isomorphism). *Being the proposed notations,*

$$1.\ (\mathfrak{P}_{rec}, \equiv^+, |, m[\ ], C.\ , !, (rec.X\ ), (\nu n), \longrightarrow) \overset{iso}{\backsim} (\mathcal{A}^{fin}, \approx^*, \|, m\lfloor\ \rceil, C.\ , \widehat{rec}, (rec.X\ ), (new\ n), \Longrightarrow)$$

$$2.\ (\mathfrak{P}^{\infty}_{rec}, \equiv^+, |, |_{i\in\mathbb{N}}, m[\ ], C.\ , !, (rec.X\ ), (\nu n), \longrightarrow) \overset{iso}{\backsim} (\mathcal{A}, \approx^*, \|, \|_{i\in\mathbb{N}}, m\lfloor\ \rceil, C.\ , \widehat{rec}, (rec.X\ ), (new\ n), \Longrightarrow).$$

11 A propositional temporal logic for Recursive Ambient Calculus

$\mathcal{A}^{fin}$ and $\mathcal{A}$ are models for the most expressive ambient calculi. Since Recursive Ambient Calculus without denumerable parallel composition is an extension of the classical one by interpreting the replication in the recursive approach, we will focus further only on this calculus. Hence we will focus only on the classes of hereditarily finite A-hierarchies, $\mathcal{A}^{fin}$, and A-graphs, $\mathcal{G}^{fin}$. For these we develop a CTL^* logic able to predict on the computations. This logic will work also with the Recursive Ambient Calculus because $\mathfrak{P}_{rec} \overset{iso}{\backsim} \mathcal{A}^{fin}$. We sketch here the basic lines of this construction. We mention that following these lines other logics can be developed as well, such as CTL.

11.1 The canonical representatives of a family of A-Hierarchies

We project $\approx^*$ to hereditarily finite A-graphs, following the definition 10.1:

Definition 11.1. Let the relation $\approx^* \subset \mathcal{G}^{fin} \times \mathcal{G}^{fin}$ defined by:

1. $G \approx^* G$
2. $G_1 \approx^* G_2 \Rightarrow G_2 \approx^* G_1$
3. $G_1 \approx^* G_2, G_2 \approx^* G_3 \Rightarrow G_1 \approx^* G_3$
4. $G_1 \approx^* G_2 \Rightarrow (\nu n)G_1 \approx^* (\nu n)G_2$
5. $G_1 \approx^* G_2 \Rightarrow G_1 \parallel G_3 \approx^* G_2 \parallel G_3$
6. $G_1 \approx^* G_2 \Rightarrow \widehat{rec}G_1 \approx^* \widehat{rec}G_2$
7. $G_1 \approx^* G_2 \Rightarrow n_\beta\lfloor G_1 \rceil \approx^* n_\beta\lfloor G_2 \rceil$
8. $(\nu n).m_\beta\lfloor G \rceil \approx^* m_\beta\lfloor(\nu n).G\rceil, \quad n \neq m$
9. $\widehat{rec}(G_1 \parallel G_2) \approx^* \widehat{rec}G_1 \parallel \widehat{rec}G_2$
10. $\widehat{rec}\{nil\}_p \approx^* \{nil\}$
11. $\widehat{rec}(\widehat{rec}G) \approx^* \widehat{rec}G, \widehat{rec}G \approx^* G \parallel \widehat{rec}G$
12. $G_1 \approx^* G_2 \Rightarrow M.G_1 \approx^* M.G_2$
13. $G_1 \approx^* G_2 \Rightarrow (n).G_1 \approx^* (n).G_2$
14. $G \approx^* \varepsilon.G$
15. $(M.M').G \approx^* M.M'.G$
16. $(\nu n)(\nu m)G \approx^* (\nu m)(\nu n)G$
17. $(\nu n)\{nil\}_p \approx^* \{nil\}_p$
18. $(\nu n)G_1 \parallel G_2 \approx^* G_1 \parallel (\nu n)G_2, n \notin fn(P)$
19. $G \parallel \{nil\}_p \approx^* G$
20. $(G_1 \parallel G_2) \parallel G_3 \approx^* G_1 \parallel (G_2 \parallel G_3)$
21. $G_1 \parallel G_2 \approx^* G_2 \parallel G_1$
22. $(x).G \approx^* (y).G(x \leftarrow y)$ if $y \notin fn(G)$
23. $(\nu n).G \approx^* (\nu m).G(n \leftarrow m)$ if $m \notin fn(G)$
24. $\widehat{0}_\beta\lfloor G \rceil \approx^* G$
25. $(recX)_{\{\gamma\}}.X \approx^* \{nil\}$
26. $(recX).G \approx^* G\{X \leftarrow (recX).G\}$
27. $G_1 \approx^* G_2 \Rightarrow (recX).G_1 \approx^* (recX).G_2$

Theorem 11.1. *If $G_1 \approx^* G_2$ then $(\!|G_1|\!) \approx^* (\!|G_2|\!)$. If $H_1 \approx^* H_2$ and $G_1 \in H_1$ then it exists $G_2 \in H_2$ such that $G_1 \approx^* G_2$.*

Similarly, we project the relation $\Longrightarrow$ to $\mathcal{G}^{fin}$, recalling the definition 8.2.

Definition 11.2. We define the relation $\Longrightarrow \subseteq \mathcal{G}^{fin} \times \mathcal{G}^{fin}$ over A-graphs by the following rules:

1. **(In-Rule):**$(m,\varepsilon)_\beta\lfloor in\ n.G_1 \parallel G_2\rceil \parallel (n,\varepsilon)_\delta\lfloor G_3\rceil \Longrightarrow (n,\varepsilon)_\delta\lfloor G_3 \parallel (m,\varepsilon)_\beta\lfloor G_1 \parallel G_2\rceil\rceil$
2. **(Out-Rule):**$(n,\varepsilon)_\beta\lfloor G_3 \parallel (m,\varepsilon)_\gamma\lfloor out\ n.G_1 \parallel G_2\rceil\rceil \Longrightarrow (m,\varepsilon)_\gamma\lfloor G_1 \parallel G_2\rceil \parallel (n,\varepsilon)_\beta\lfloor G_3\rceil$
3. **(Open-Rule):**$open\ n.G_1 \parallel (n,\varepsilon)\lfloor G_2\rceil \Longrightarrow G_1 \parallel G_2$
4. **(Comm-Rule):**[16]:$(x).G \parallel \langle M\rangle.\{nil\}_p \Longrightarrow G\{x \leftarrow M\}$
5. **(Par-Rule):**If $G_1 \Longrightarrow G_2$ then $G_1 \parallel G \Longrightarrow G_2 \parallel G$
6. **(DenPar-Rule):**If $G_i \Longrightarrow G'_i$ for each $i \in \mathbb{N}$, then $\parallel_{i\in\mathbb{N}} G_i \Longrightarrow \parallel_{i\in\mathbb{N}} G'_i$
7. **(Amb-Rule):**If $G_1 \Longrightarrow G_2$ then $m_\beta\lfloor G_1\rceil \Longrightarrow m_\beta\lfloor G_2\rceil$
8. **(New-Rule):**If $G_1 \Longrightarrow G_2$ then $(new\ n)G_1 \Longrightarrow (new\ n)G_2$
9. **(Closure-Rule):**If $G'_1 \Longrightarrow G'_2$, and $G'_1 \approx^* G''_1$, respective $G'_2 \approx^* G''_2$ then $G''_1 \Longrightarrow G''_2$

Let $\Longrightarrow^*$ be the transitive closure of $\Longrightarrow$ over $\mathcal{G}^{fin}$ and respectively over $\mathcal{A}^{fin}$.

As an immediate consequence of the definitions 8.2 and 11.2 we have the next result.

Theorem 11.2. *If $G_1 \Longrightarrow^* G_2$ then $(\!|G_1|\!) \Longrightarrow^* (\!|G_2|\!)$. If $H_1 \Longrightarrow^* H_2$ and $G_1 \in H_1$ then it exists $G_2 \in H_2$ such that $G_1 \Longrightarrow^* G_2$.*

Theorem 11.3. *If $G_1 \Longrightarrow^* G_2$, $G_1 = \langle \mathcal{E}_1, \mathcal{L}_1\rangle$ with $\mathcal{E}_1 = \langle X_1, A_1, e_1, \alpha\rangle$ and $G_2 = \langle \mathcal{E}_2, \mathcal{L}_2\rangle$ with $\mathcal{E}_2 = \langle X_2, A_2, e_2, \alpha\rangle$, then $X_2 \subseteq X_1$ and $A_2 \subseteq A_1$.*

Definition 11.3. Consider the A-graph $G_0 = \langle \mathcal{E}_0, \mathcal{L}_0\rangle$ with $\mathcal{E}_0 = \langle X_0, A_0, e_0, \alpha\rangle$, and $H_0 = (\!|G_0|\!)$. The set $\mathcal{F}_{H_0} = \{H \in \mathcal{A}^{fin} | \ H_0 \Longrightarrow^* H\}$ is *the reductive family of* H_0. Analogously, the set $\mathcal{F}_{G_0} = \{G \in \mathcal{G}^{fin} | \ G_0 \Longrightarrow^* G\}$ is *the reductive family of* G_0.

[16] We denoted by $G\{x \leftarrow M\}$ the A-graph obtained from G by replacing in all its labels any occurrence of x by M.

With this definition, using the representation theorem 7.1 and the theorems 11.1, 11.2 we obtain:

Theorem 11.4. $(\mathcal{F}_{G_0}, \approx^*, \|, m\lfloor\ \rceil, C.\ , \widehat{rec}, (rec.X\ ), (new\ n), \Longrightarrow)$ *is a system of canonical representatives for* $(\mathcal{F}_{(\!|G_0|\!)}, \approx^*, \|, m\lfloor\ \rceil, C.\ , \widehat{rec}, (rec.X\ ), (new\ n), \Longrightarrow)$.

With this result we will construct the logic in top of $\mathcal{F}_{G_0}$ for a given G_0 as initial state.

The logic we intend to construct is a branching propositional temporal logic, CTL^*[17]. The requirements of such a construction [12] are to organize a structure $\mathfrak{M} = (S_0, \Sigma, R, I)$ where S_0 is the initial state of our model, Σ is the class of all possible states in our model, R is the accessibility relation between states, $R \subseteq \Sigma \times \Sigma$, and $I : \Sigma \longrightarrow \mathcal{P}(Ap)$ is a function which associates to each state $S \in \Sigma$ a set of atomical propositions $I(S) \subseteq \mathcal{P}(Ap)$ - the set of true atomical propositions in the state S (Ap will be the class of atomical propositions).

Due to the General Theorem of Isomorphism, we can use A-hierarchies as states. Consider the A-graph $G_0 = \langle \mathcal{E}_0, \mathcal{L}_0 \rangle$ with $\mathcal{E}_0 = \langle X_0, A_0, e_0, \alpha \rangle$, and $H_0 = (\!|G_0|\!)$. Because $\mathcal{F}_{G_0}$ is a system of canonical representatives for $\mathcal{F}_{(\!|G_0|\!)}$ we will use these A-graphs to speak about the A-hierarchies.

We need, as *initial state*, H_0. We take $S_0 = G_0$.

The set of possible states should be the family of H_0, $\mathcal{F}_{H_0}$. We take $\Sigma = \mathcal{F}_{G_0}$.

The accessibility relation between states is generated by $\Longrightarrow$.

We take as *the set of atomical propositions* the set $Ap = \{x \mathfrak{in} y \mid x \in A_0 \cup X_0, y \in X_0\}$.

We take *the interpretation function* $I : \mathcal{F}_{G_0} \to \mathcal{P}(Ap)$ by:

$$I(G) = \{x \mathfrak{in} y \mid x \in e(y), where\ e\ are\ the\ equations\ of\ G\}.$$

11.2 Syntax and Semantics

Following the classic way of introducing CTL^* we define a *fullpath* as an infinite sequence $H_0, H_1, ...$ of states such that $H_i \Longrightarrow H_{i+1}$ for all i[18].

Further, we introduce the syntax of the CTL* logic in the usual way [12]. We inductively define a class of state formulae (formulae which will be true or false of states) and a class of path formulae (true or false of paths), starting from *Ap*. We have the classical logic operators - $\wedge$ and $\neg$ - together with the temporal operators X (*next time*) and $\cup$ (*until*). We also have the path quantifier E (*for some futures*). From these we can derive the temporal operators G (*always*) and F (*sometimes*), and the path quantifier A (*for all futures*). The propositions of this logic can be satisfied by processes, or by sequences of processes (as a computational path). The syntactical rules are the classical ones for CTL* [12].

We now define $\models$ inductively. We write $\mathfrak{M}, H_0 \models p$ to mean that the formula p is true at state H_0 in the model $\mathfrak{M}$, and $\mathfrak{M}, x \models p$ to mean that the path formula p is true for the fullpath x in the structure $\mathfrak{M}$. The rules are:

$\mathfrak{M}, H_0 \models P$ iff $P \in I(H_0)$, where $P \in Ap$

$\mathfrak{M}, H_0 \models p \wedge q$ iff $\mathfrak{M}, H_0 \models p$ and $\mathfrak{M}, H_0 \models q$

$\mathfrak{M}, H_0 \models \neg p$ iff it is not the case that $\mathfrak{M}, H_0 \models p$

$\mathfrak{M}, H_0 \models Ep$ iff $\exists$ fullpath $x = (H_0, H_1, ...)$ in $\mathfrak{M}$ with $\mathfrak{M}, x \models p$

$\mathfrak{M}, H_0 \models Ap$ iff $\forall$ fullpath $x = (H_0, H_1, ...)$ in $\mathfrak{M}$ with $\mathfrak{M}, x \models p$

$\mathfrak{M}, x \models p$ iff $\mathfrak{M}, H_0 \models p$

$\mathfrak{M}, x \models p \wedge q$ iff $\mathfrak{M}, x \models p$ and $\mathfrak{M}, x \models q$

$\mathfrak{M}, x \models \neg p$ iff it is not the case that $\mathfrak{M}, x \models p$

$\mathfrak{M}, x \models p \cup q$ iff $\exists i \left(\mathfrak{M}, x^i \models q \text{ and } \forall j \left(j < i \text{ implies } \mathfrak{M}, x^j \models p\right)\right)$

[17] We choose CTL^* for its expressivity, but a CTL is possible as well.

[18] We use the convention that if $x = (H_0, H_1, ...)$ denotes a fullpath, then x^i denotes the suffix path $(H_i, H_{i+1}, H_{i+2}, ...)$.

$\mathfrak{M}, x \models Xp$ iff $\mathfrak{M}, x^1 \models p$

Definition 11.4. A state formula p (resp. path formula p) is *valid* provided that for every structure $\mathfrak{M}$ and every state H (resp. fullpath x) in $\mathfrak{M}$ we have $\mathfrak{M}, H \models p$ (resp. $\mathfrak{M}, x \models p$). A state formula (resp. path formula) p is *satisfiable* provided that for some structure $\mathfrak{M}$ and some states H (resp. fullpath x) in $\mathfrak{M}$ we have $\mathfrak{M}, H \models p$ (resp. $\mathfrak{M}, x \models p$).

12 Conclusions

We analyzed the algebraical types of ambient calculi focusing on spatial structures of the processes. We developed the Ambient Hierarchies, set theoretical entities that can be algebraically organized to have the same types with ambient calculi. The Ambient Hierarchies (tree structures possibly non-wellfounded) give us the possibility to understand the role of the spatial structures in the economy of each calculus. Thus we obtain a classification of ambient calculi from the expressivity point of view. We identify a Recursive Ambient Calculus as the most expressive formalism in the context of a finite syntax. Using the patterns of unfolding of an infinite tree, we proved that, for these calculi, replication is not expressive enough to model or simulate recursion, assuming that actions occurring in different locations, even if they have the same names, are not behaviorally equivalent. Following the same idea we provide a structural congruence for Recursive Ambient Calculus, stronger than the relations proposed before, that can recognize some equivalences which were undecidable, such as $!P = P|P|P|... \equiv (recX.X|P)$ (if X does not appear free in P), $(recX.X|P|Q) \equiv (recX.X|P)|(recX.X|Q)$ (if X does not occur free in P and Q), or $(recX.X|(recY.Y|P)) \equiv (recX.X|P)$ (if X, Y do not occur free in P). As an application of the Ambient Hierarchies, we construct a temporal logic on top of them resorting on their tree structure. Due to the isomorphism between ambient calculi and algebras of A-hierarchies, this logic can be used as well to express properties of ambient processes.

References

[1] NuSMV: A new symbolic model checker. *http://nusmv.irst.itc.it/*.

[2] The SMV system. *http://www-2.cs.cmu.edu/ modelcheck/smv.html*.

[3] P. Aczel. *Non-Well-Founded Sets.* CLSI Lecture Notes Number 14 Stanford: CSLI Publication, 1988.

[4] A. Baltag. Sts: A structural theory of sets. *Logic Journal of the IGPL*, 7(4):481–515, 1999.

[5] J. Barwise and L. Moss. *Vicious Circles. On the Mathematics of Non-Wellfounded Phenomena.* CLSI Lecture Notes Number 60 Stanford: CSLI Publication, 1996.

[6] S. D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A theory of communicating sequential processes. *J. ACM*, 31(3):560–599, 1984.

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

[8] L. Cardelli and A.D. Gordon. Ambient logic. *Mathematical Structures in Computer Science. To appear. Available at http://www.luca.demon.co.uk/.*

[9] L. Cardelli and A.D. Gordon. Anytime, anywhere. modal logics for mobile ambients. *Proceedings of the 27th ACM Symposium on Principles of Programming Languages*, pages 365–377, 2000.

[10] L. Cardelli and A.D. Gordon. Mobile ambients. *Theoretical Computer Science, Special Issue on Coordination, D. Le Mtayer Editor*, pages 177–213, June 2000.

[11] Witold Charatonik, Andrew D. Gordon, and Jean-Marc Talbot. Finite-control mobile ambients. In *ESOP '02: Proceedings of the 11th European Symposium on Programming Languages and Systems*, pages 295–313. Springer-Verlag, 2002.

[12] E. A. Emerson. Temporal and modal logic. *Handbook of Theoretical Computer Science*, B: Formal Models and Sematics:995–1072, 1990.

[13] F. Honsell and M. Forti. *Set Theory with Free Construction Principles.* Annali Scuola Normale Supeiore di Pisa, 1983.

[14] L.Alberucci and V. Salipante. On modal mu-calculus and non-well-founded set theory. *Journal of Philosophical Logic*, 33, no.4:343–360(18), August 2004.

[15] R. Milner. *Communication and concurrency.* Prentice-Hall, Inc., London, 1989.

[16] Robin Milner. *Communicating and Mobile Systems: the Pi-Calculus.* Cambridge University Press, 1999.

[17] Davide Sangiorgi. Extensionality and intensionality of the ambient logics. In *POPL '01: Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 4–13. ACM Press, 2001.

Appendix: Proofs of the theorems

In this appendix we present the main points in the proofs of the non-trivial results presented in the paper.

Proof of the Theorem 4.3. For the beginning we define the two partitions. Let $\zeta_A \subset A \times A$ defined by $(a_1, a_2) \in \zeta_A$ iff $\exists b \in B$ such that $(a_1, b), (a_2, b) \in \zeta$. We prove that ζ_A is an equivalence relation on A. By the first condition of the definition 4.8, for any $a \in A$ it exists $b \in B$ such that $(a, b) \in \zeta$, hence $(a, a) \in \zeta_A$, so ζ_A is reflexive. The symmetry derives also from the definition of ζ_A. Now we prove the transitivity. Let $(a_1, a_2), (a_2, a_3) \in \zeta_A$. There exist $b_1, b_2 \in B$ such that $(a_1, b_1), (a_2, b_1), (a_2, b_2), (a_3, b_2) \in \zeta$. But from $(a_1, b_1), (a_2, b_1), (a_2, b_2) \in \zeta$ we derive, by the third condition of definition 4.8, that $(a_1, b_2) \in \zeta$, and because $(a_3, b_2) \in \zeta$ we obtain that $(a_1, a_3) \in \zeta_A$. Hence ζ_A is an equivalence relation, so its set of equivalence classes defines a partition of A, let's denote it by $(A_i)_{i \in I}$. Similarly we define a relation $\zeta_B \subset B \times B$ by $(b_1, b_2) \in \zeta_B$ iff it exists $a \in A$ such that $(a, b_1), (a, b_2) \in \zeta$. As before, ζ_B defines a partition $(B_j)_{j \in J}$. We prove now that $I = J$. Consider the relation $r \subset (A_i)_{i \in I} \times (B_j)_{j \in J}$ defined by $(A_i, B_j) \in r$ iff there exist $a \in A_i, b \in B_j$ such that $(a, b) \in \zeta$. We prove that r defines a one-to-one onto relation. Let $A_k \in (A_i)_{i \in I}$ and $a \in A_k$. Due to the first condition of the definition 4.8 it exists $b \in B$ such that $(a, b) \in \zeta$. But it exists $l \in J$ with $b \in B_l$. Hence $(A_k, B_l) \in r$. Suppose now that $(A_k, B_s), (A_k, B_t) \in r$. Then there exists $a \in A_k, b_1 \in B_l, b_2 \in B_t$ such that $(a, b_1), (a, b_2) \in \zeta$. By the definition of ζ_B we obtain $(b_1, b_2) \in \zeta_B$, hence $l = k$. Symmetrically can be proved that if $(A_k, B_l), (A_m, B_l) \in r$ then $m = k$. Hence $I = J$. □

Proof of the Theorem 4.4. We use the fact that $\zeta = id_A$ and $x\mathfrak{R}x'$ iff $x = x'$ satisfies the requirements of bisimulation, to prove that $\equiv_S$ is reflexive. For symmetry we use ζ^{-1} and $\mathfrak{R}^{-1}$ to define inverse bisimulation, $\zeta, \mathfrak{R}$ being those that defines the direct one. For transitivity we use $\zeta_1 \circ \zeta_2$ and $\mathfrak{R}_1 \circ \mathfrak{R}_2$. □

Proof of the Theorem 4.6. The theorem was proved in [5], page 79, theorem 7.1, for the particular case $\zeta = id$, the authors being interested to define the identity of structures as depending of the atoms involved. We follow the same idea for this proof. Let $\mathcal{E} = \langle X, A, e \rangle$ and $\mathcal{E}' = \langle X', A', e' \rangle$. Assume first that $\mathcal{E}$, $\mathcal{E}'$ have the solution-sets s, s' in the relation $\overline{\zeta}$. We define a relation $\mathfrak{R}$ on $X \times X'$ by:

$$x\mathfrak{R}x' \ iff \ (s_x, s'_{x'}) \in \overline{\zeta} \tag{12.1}$$

$\mathfrak{R}$ defines a ζ-bisimulation relation. We verify some conditions. Suppose that $x \in X$, so that $s_x \in ss(\mathcal{E})$. Then there is some $x' \in X'$ such that $(s_x, s'_{x'}) \in \overline{\zeta}$. Thus $x\mathfrak{R}x'$. The converse can be proved similarly. Suppose now that $x\mathfrak{R}x'$ and $y \in e_x \cap X$. Then since $s_y \in s_x$ and $(s_x, s'_{x'}) \in \overline{\zeta}$, we must have some $y' \in e'_{x'}$ such that $(s_y, s'_{y'}) \in \overline{\zeta}$. Thus $y\mathfrak{R}y'$. Once again the converse is similar. Finally if $(s_x, s'_{x'}) \in \overline{\zeta}$ then their sets of urelements must be related by $\overline{\zeta}$. The set of urelements of s_x is $e_x \cap A$, and the set of urelements of $s'_{x'}$ is $e'_{x'} \cap A'$. Therefore $(e_x \cap A, e'_{x'} \cap A') \in \overline{\zeta}$. This concludes that $\mathfrak{R}$ is a ζ-bisimulation.

To prove the converse it is enough to show that $(s_x, s'_{x'}) \in \overline{\zeta}$ if $x\mathfrak{R}x'$. To do this we will construct two additional systems $\mathcal{E}^* = \langle X^*, A, e^* \rangle$ and $\mathcal{E}^{**} = \langle X^*, A', e^{**} \rangle$, where $X^* = \{(x, x') \mid x \in X, x' \in X', x\mathfrak{R}x'\}$ and

$$e^*_{(u,u')} = \{(v, v') \in X^* \mid v \in e_u, v' \in e'_{u'}\} \cup (A \cap e_u) \tag{12.2}$$

$$e^{**}_{(u,u')} = \{(v, v') \in X^* \mid v \in e_u, v' \in e'_{u'}\} \cup (A' \cap e'_{u'}) \tag{12.3}$$

Now, it is trivial to verify that $\mathcal{E}^*$ and $\mathcal{E}^{**}$ are ζ-bisimilar and for any $(u, u') \in X^*$ we have $(s^*_{(u,u')}, s^{**}_{(u,u')}) \in \overline{\zeta}$. It can be proved that s is a solution of $\mathcal{E}^*$ and s' is a solution of $\mathcal{E}'$ respectively (details of these proofs can be found in [5], chapter 7, pages 79-80). Further, using AFA (which is equivalent with the fact that each flat system of equations has a unique solution, by the theorem 4.2) we obtain that s, s' are in relation $\overline{\zeta}$, which means that for all $x\mathfrak{R}x'$ we have $(s_x, s_{x'}) \in \overline{\zeta}$. q.e.d. □

Proof of Theorem 5.1. Observe that the first three conditions of congruence in the definition 5.9 are satisfied due to the fact that the bisimulation relation on systems of equations is an equivalence relation. We have to verify the fourth condition.

Reflexivity: We take $\mathfrak{R} = id_A \cup id_X$ and $\phi_{pn} = id_{pn(G)}$, $\phi_{bn} = id_{bn(G)}$.

Symmetry: suppose that $G_1 \cong G_2$ by $\zeta, \mathfrak{R}$ and ϕ. We define $\zeta' = \zeta^{-1}$, $\mathfrak{R}' = \mathfrak{R}^{-1}$, and $\phi' = \phi^{-1}$. We have $\mathcal{L}_2(x_2)\overline{\phi'}\mathcal{L}_1(x_1)$ iff $\mathcal{L}_1(x_1)\overline{\phi}\mathcal{L}_2(x_2)$ for $x_1\mathfrak{R}x_2$ that is equivalent with $x_2\mathfrak{R}'x_1$.

Transitivity: Suppose that $G_1 \cong G_2$ by $\zeta_1, \mathfrak{R}_1, \phi_1$ and $G_2 \cong G_3$ by $\zeta_2, \mathfrak{R}_2, \phi_2$. We define $\phi = \phi_2 \circ \phi_1$, $\zeta = \zeta_2 \circ \zeta_1$ and $\mathfrak{R} = \mathfrak{R}_2 \circ \mathfrak{R}_1$. Suppose that $x_1\mathfrak{R}x_3$ then $x_2 \in X_2 \cup A_2$ exists such that $x_1\mathfrak{R}_1x_2, x_2\mathfrak{R}_2x_3$. Then $\mathcal{L}_1(x_1)\overline{\phi}_1\mathcal{L}_2(x_2)$ and $\mathcal{L}_2(x_2)\overline{\phi}_2\mathcal{L}_3(x_3)$, so, by the definition of ϕ, $\mathcal{L}_1(x_1)\overline{\phi}\mathcal{L}_3(x_3)$. □

Proof of Lemma 6.1. 1. Consider a bijective function $f : X \cup A \to \mho \setminus S$. Without going too deep into details, we ensure to the reader that such a function does exist due to the Strong Axiom of Plenitude available in ZFC^-, [5]. We take $X' = f(X)$, $A' = f(A)$, $\alpha' = f(\alpha)$, $e' = f \circ e \circ f^{-1}$ and $\mathcal{L}' = \mathcal{L} \circ f^{-1}$. We define $a\zeta a'$ iff $f(a) = a'$ and $x\mathfrak{R}x'$ iff $f(x) = x'$. $\mathfrak{R}$ defines the bisimulation $\mathcal{E}' \equiv_S \mathcal{E}$, $id_{\Lambda_{bn}}$ defines the bisimulation $\mathcal{N}^{bn}_G \equiv_S \mathcal{N}^{bn}_{G'}$, and $id_{\Lambda_{pn}}$ defines the bisimulation $\mathcal{N}^{pn}_G \equiv_S \mathcal{N}^{pn}_{G'}$. We take $\phi = id_{\Lambda\cup\Lambda_{pn}\cup\Lambda_{bn}\cup\Pi}$. If $x\mathfrak{R}x'$ then $f(x) = x'$, so $\mathcal{L}'(x') = (\mathcal{L} \circ f^{-1})(x') = \mathcal{L}(x)$, hence $\mathcal{L}(x)\overline{\phi}\mathcal{L}'(x')$. Thus the two A-graphs are congruent. By construction, $S \cap (X' \cup A') = \emptyset$.

2. S and X, R and A, being cardinal-equivalent and $X \cap A = R \cap S = \emptyset$, there exists a bijective function $f : X \cup A \to S \cup R$ with $f(X) = S$ and $f(A) = R$. If we take $\alpha' = f(\alpha)$, the construction goes on as in the previous case. □

Proof of Theorem 6.2. Let $G'_i = \langle \mathcal{E}'_i, \mathcal{L}'_i \rangle$ with $\mathcal{E}'_i = \langle X'_i, A'_i, e'_i, \alpha'_i \rangle$ and $G''_i = \langle \mathcal{E}''_i, \mathcal{L}''_i \rangle$ with $\mathcal{E}''_i = \langle X''_i, A''_i, e''_i, \alpha''_i \rangle$, $i = 1, 2$, each couple satisfying the requests of the previous construction. Suppose that $\zeta_i \subset A'_i \times A''_i$ are the two proper relations that define the two bisimulation relations $\mathfrak{R}_i \subseteq X'_i \times X''_i$ that give us, together with ϕ_i the congruences stated in the theorem, $i = 1, 2$. Between the sets of atoms of $G'_1 \parallel G'_2$ and $G''_1 \parallel G''_2$ we define the proper relation $\zeta \subset (A'_1 \cup A'_2) \times (A''_1 \cup A''_2)$ by $\zeta \stackrel{def}{=} \zeta_i|_{X_i \cup A_i}$, and the relation $\mathfrak{R} \stackrel{def}{=} \mathfrak{R}_1 \cup \mathfrak{R}_2$. This gives the substitutive-bisimulation between the flat systems of $G'_1 \parallel G'_2$ and $G''_1 \parallel G''_2$. Similarly we obtain $\mathcal{N}^{bn}_{G'_1\parallel G'_2} \equiv_S \mathcal{N}^{bn}_{G''_1\parallel G''_2}$ from $\mathcal{N}^{bn}_{G'_1} \equiv_S \mathcal{N}^{bn}_{G''_1}$, $\mathcal{N}^{bn}_{G'_2} \equiv_S \mathcal{N}^{bn}_{G''_2}$ and $\mathcal{N}^{pn}_{G'_1\parallel G'_2} \equiv_S \mathcal{N}^{pn}_{G''_1\parallel G''_2}$ from $\mathcal{N}^{pn}_{G'_1} \equiv_S \mathcal{N}^{pn}_{G''_1}$, $\mathcal{N}^{pn}_{G'_2} \equiv_S \mathcal{N}^{pn}_{G''_2}$. If $(S' \leftrightarrow pn(G'_2)) \circ (R' \leftrightarrow bn(G'_2)), (S'' \leftrightarrow pn(G''_2)) \circ (R'' \leftrightarrow bn(G''_2))$ are the substitutions used in the constructions of the two parallel composed graphs, then $\phi = (S' \leftrightarrow pn(G'_2)) \circ (R' \leftrightarrow bn(G'_2)) \circ \phi_1 \circ (S'' \leftrightarrow pn(G''_2)) \circ (R'' \leftrightarrow bn(G''_2)) \circ \phi_2$ fulfills the requests of the fourth condition in definition 5.9. With these definitions, it is trivial to verify the fourth condition of the definition 5.9. □

The proof of the Theorem 6.3. We prove only the second part of the theorem, concerning the A-graphs, the first part being a consequence of this one. Let $G_i = \langle \mathcal{E}_i, \mathcal{L}_i \rangle$ with $\mathcal{E}_i = \langle X_i, A_i, e_i, \alpha_i \rangle$, $i = 1, 3$ be three A-graphs satisfying the conditions of parallel composition, i.e. $\cap_i(A_i \cup X_i) = \emptyset$. We prove that $(G_1 \parallel G_2) \parallel G_3 \cong G_1 \parallel (G_2 \parallel G_3)$. Defining $\mathfrak{R} = id_{\cup_i(A_i\cup X_i)}$ we obtain the two flat systems bisimilar. Let R_2, S_2 be the sets that substitute $bn(G_2)$ and respectively $pn(G_2)$ in $G_1 \parallel G_2$, let R_3, S_3 be the similar sets in $(G_1 \parallel G_2) \parallel G_3$, R'_3, S'_3 in $G_2 \parallel G_3$, and R'_2, S'_2 in $G_1 \parallel (G_2 \parallel G_3)$. We use for $G_1 \parallel (G_2 \parallel G_3)$ first $(bn(G_3) \leftrightarrow R'_3)$, then $((bn(G_2) \cup R'_3) \leftrightarrow R'_2)$, i.e. $R'_2 = R''_2 \cup R''_3$, with $R''_2 \cap R''_3 = \emptyset$ and $((bn(G_2) \cup R'_3) \leftrightarrow R'_2) = (bn(G_2) \leftrightarrow R''_2) \circ (R'_3 \leftrightarrow R''_3)$. Now, if we define $\phi_{bn} = (R_2 \leftrightarrow R''_2) \circ (R_3 \leftrightarrow R''_3)$, this defines $\mathcal{N}^{bn}_G \equiv_S \mathcal{N}^{bn}_{G'}$. In a similar way we can prove the bisimulation $\mathcal{N}^{pn}_G \equiv_S \mathcal{N}^{pn}_{G'}$. Moreover, the extension of these definitions to ϕ fulfills the requests of the fourth condition in definition 5.9. Hence, the proof is complete for associativity.

For commutativity, using the same notations for the A-graphs G_1, G_2, we consider R_1 the set used to replace the bound names of G_1 in the construction of $G_2 \parallel G_1$, and R_2 the similar in the construction of $G_1 \parallel G_2$. Then $\phi_{bn} = (bn(G_1) \leftrightarrow R_1) \circ (bn(G_2) \leftrightarrow R_2)$ gives $\mathcal{N}^{bn}_{G_1\parallel G_2} \equiv_S \mathcal{N}^{bn}_{G_2\parallel G_1}$. Similarly, $\mathcal{N}^{pn}_{G_1\parallel G_2} \equiv_S \mathcal{N}^{pn}_{G_2\parallel G_1}$. Now, extending these to ϕ we can trivially verify the fourth condition in the definition 5.9, q.e.d. □

Proof of the Theorem 6.4. Suppose that $\zeta, \mathfrak{R}, \phi_{bn}$ and ϕ_{pn} are those which defines the congruence between G_1 and G_2. We take $\zeta' = \zeta, \mathfrak{R}' = \mathfrak{R} \cup \{(\beta', \beta'')\}, \phi'_{bn} = \phi_{bn}$, and $\phi'_{pn} = \phi_{pn}$.

These fulfill the requirements of the definition 5.9. $\square$

Proof of the Theorem 6.5. It is sufficient to prove it for $C = c$. Suppose that $G \cong G'$ is defined by ζ, $\mathfrak{R}$ and ϕ, that $e(\alpha) = \{\beta\}, e'(\alpha') = \{\beta'\}$ and that $(\beta, \beta') \in \mathfrak{R}$. If $c = \varepsilon$, or $c = \langle M \rangle$ the result is trivial. Suppose that $c \notin \{(n_i^{bn}), \varepsilon, \langle M \rangle\}$ and that $c = Mu, u \in \Lambda$. The first three conditions of definition 5.9 are verified for $c.G, c.G'$ due to the fact that are verified by G, G'. If $(u, u) \in \phi$ then ζ, $\mathfrak{R}$ and ϕ define the wanted bisimulation. If $(u, u) \notin \phi$ we extend it to contain this couple, and with this extension the requirements are fulfilled. If $c = (n_i^{bn})$, suppose that $(n_i^{bn} \leftrightarrow n_j^{bn})$ defines $c.G$ and $(n_i^{bn} \leftrightarrow n_k^{bn})$ defines $c.G'$. In this case we have to prove that we still have $\mathcal{N}_{c.G}^{bn} \equiv_S \mathcal{N}_{c.G'}^{bn}$. If ϕ_{bn} defines $\mathcal{N}_G^{bn} \equiv_S \mathcal{N}_{G'}^{bn}$ then we have to extend it by $\phi'_{bn} = \phi_{bn} \cup (n_j^{bn}, n_k^{bn})$. With this extension all the requirements are fulfilled. $\square$

Proof of Theorem 6.6. Suppose that $G_i = \langle \mathcal{E}_i, \mathcal{L}_i \rangle$ with $\mathcal{E}_i = \langle X_i, A_i, e_i, \alpha \rangle$ for $i \in \{1, 2\}$. Suppose that $\zeta \subset A_1 \times A_2$, $\phi = \phi_{bn} \cup \phi_{pn} \cup id|_{\Lambda \cup \Pi}$ and $\mathfrak{R}$ define the congruence relation between G_1 and G_2 and that $(n_i^{pn} \leftrightarrow n)$, respective $(n_j^{pn} \leftrightarrow n)$ are the substitutions used to define $(new\ n)G_1$ and $(new\ n)G_2$ respectively, where $i \notin arity_{G_1}^{pn}(n)$ and $j \notin arity_{G_2}^{pn}(n)$. We extend $\phi'_{pn} = \phi_{pn} \cup (n_i^{pn}, n_j^{pn})$, and this relation provides $\mathcal{N}_{(new\ n)G_1}^{pn} \equiv_S \mathcal{N}_{(new\ n)G_2}^{pn}$. Updating ϕ and keeping ζ and $\mathfrak{R}$ unchanged we obtain the wanted congruence. $\square$

Proof of Theorem 6.7. Consider that $(n \leftrightarrow n_j^{pn}) \circ (m \leftrightarrow m_i^{pn})$ and $(n \leftrightarrow n_t^{pn}) \circ (m \leftrightarrow m_s^{pn})$ are the substitutions used in constructing the A-hierarchies for $(new\ n)(new\ m)G$ and respectively $(new\ m)(new\ n)G$, where $i, s \notin pn(G)$, $j \notin pn(G) \cup \{i\}, t \notin pn(G) \cup \{s\}$. $\mathfrak{R} = id|_{X \cup A}, \phi_{bn} = id_{bn(G)}$ and $\phi_{pn} = (n_j^{pn} \leftrightarrow n_t^{pn}) \circ (m_i^{pn} \leftrightarrow m_s^{pn})$ defines the congruence. $\square$

Proof of the theorem 6.8. We denote by $E = G\{P \leftarrow F\}$ and by $E' = G'\{P \leftarrow F'\}$. Suppose that $G = \langle \mathcal{E}_G, \mathcal{L}_G \rangle$ with $\mathcal{E}_G = \langle X_G, A_G, e_G, \alpha_G \rangle$, $G' = \langle \mathcal{E}_{G'}, \mathcal{L}_{G'} \rangle$ with $\mathcal{E}_{G'} = \langle X_{G'}, A_{G'}, e_{G'}, \alpha_{G'} \rangle$, $F = \langle \mathcal{E}_F, \mathcal{L}_F \rangle$ with $\mathcal{E}_F = \langle X_F, A_F, e_F, \alpha_F \rangle$, $F' = \langle \mathcal{E}_{F'}, \mathcal{L}_{F'} \rangle$ with $\mathcal{E}_{F'} = \langle X_{F'}, A_{F'}, e_{F'}, \alpha_{F'} \rangle$. Suppose that $G \cong G'$ is defined by $\zeta_G, \mathfrak{R}_G, \phi_{bn}^G$ and ϕ_{pn}^G. Suppose that we have $B = \{p_1, ...p_k\} = \mathcal{L}_G^{-1}(P \times Cap) \subseteq A_G$ and $B' = \{p'_1, ...p'_{k'}\} = \mathcal{L}_{G'}^{-1}(P \times Cap) \subseteq A_{G'}$. Let $\Gamma = \{\gamma_1, ...\gamma_k\}$, $\Gamma' = \{\gamma'_1, ...\gamma'_{k'}\}$ and $r = \{(\gamma_i, \gamma'_{i'}) \in \Gamma \times \Gamma' \mid (p_i, p'_{i'}) \in \zeta_G\}$. Suppose that $F \cong F'$ is defined by $\zeta_F, \mathfrak{R}_F, \phi_{bn}^F$ and ϕ_{pn}^F. $\zeta_E = \zeta_G \cup \zeta_F$, that generates $\mathfrak{R}_E = \mathfrak{R}_G \cup \mathfrak{R}_F \cup r$, defines the bisimulation between the flat systems of E and E'. Suppose that R, S and R', S' are the sets used to rename the bound and private names of F in E and respectively of F' in E'. $\phi_{bn}^E = \phi_{bn}^G \cup ((bn(F) \leftrightarrow R) \circ \phi_{bn}^F \circ (bn(F') \leftrightarrow R')^{-1})$ defines $\mathcal{N}_E^{bn} \equiv_S \mathcal{N}_{E'}^{bn}$, while $\phi_{pn}^E = \phi_{pn}^G \cup ((pn(F) \leftrightarrow S) \circ \phi_{pn}^F \circ (pn(F') \leftrightarrow S')^{-1})$ defines $\mathcal{N}_E^{pn} \equiv_S \mathcal{N}_{E'}^{pn}$.

$\square$

Proof of the Theorem 6.9. If $\Gamma = \{\gamma_1, ...\gamma_k\}, \Delta = \{\delta_1, ...\delta_k\}$ then defining $\zeta = id_{A_G}$, $\mathfrak{R} = \{(\gamma_i, \delta_i) \mid i = \overline{1,k}\} \cup id_{X_G}$, $\phi_{bn} = id_{bn(G)}, \phi_{pn} = id_{pn(G)}$ we obtain the congruence. $\square$

Proof of the Theorem 6.10. If $\mathfrak{R}, \zeta, \phi$ are those that defines $G_1 \cong G_2$, for verifying $(rec.P)_\Gamma.G_1 \cong (rec.P)_\Gamma.G_2$ it is enough to define $\mathfrak{R}' = \mathfrak{R} \cup id_\Gamma$. $\square$

Proof of the Theorem 6.11. Let $H = (\!|G|\!)$ and let $F = (recP)_\Gamma.G$ and $F^* \in (recP).H$ having disjunct domains. We will prove that $F \cong G_\Delta\{P \leftarrow F^*\}$. Suppose that $G = \langle \mathcal{E}_G, \mathcal{L}_G \rangle$ and $\mathcal{E}_G = \langle X_G, A_G, e_G, \alpha_G \rangle$. Let $B = \{p_1, ..., p_k\} \subset A_G$ be the set of those atoms having $\mathcal{L}_G(p_i) = (P, C_i)$, $\Gamma = \{\gamma_1, ..., \gamma_k\}, \Delta = \{\delta_1, ..., \delta_k\}$ as in the previous constructions.

Then $F = \langle \mathcal{E}_F, \mathcal{L}_F \rangle, \mathcal{E}_F = \langle X_F, A_F, e_F, \alpha_F \rangle$ with $X_F = X_G \cup \Gamma, A_F = A_G \setminus B, e_F(\gamma_i) = (e_G(\alpha_G))\{p_i \leftrightarrow \gamma_i\}$ and $e_F = e_G\{p_i \leftrightarrow \gamma_i\}$ in rest, $\mathcal{L}_F(\gamma_i) = (\widehat{0}, C_i)$ and $\mathcal{L}_F = \mathcal{L}_G$ in rest.

Let $F^* = \langle \mathcal{E}_{F^*}, \mathcal{L}_{F^*} \rangle, \mathcal{E}_{F^*} = \langle X_{F^*}, A_{F^*}, e_{F^*}, \alpha_{F^*} \rangle$ be given by one-to-one onto relations $\zeta_F, \phi_{bn}^F, \phi_{pn}^F$ from F such that the two to have different domains and $bn(F^*) \cap bn(F) = pn(F^*) \cap bn(F) = \emptyset$. We refer further to the elements of F^* naming them by the corresponding ones in F marked by $(\ )^*$. For example the root of F^* will be denoted by $(\alpha_F)^*$.

We construct the A-graph $E = G_\Delta\{P \leftarrow F^*\}$. $E = \langle \mathcal{E}_E, \mathcal{L}_E \rangle, \mathcal{E}_E = \langle X_E, A_E, e_E, \alpha_E \rangle$ with $X_E = X_G \cup X_{F^*} \cup \Delta = X_G \cup (X_G)^* \cup (\Gamma)^* \cup \Delta, A_E = (A_G \setminus B) \cup (A_G \setminus B)^*, e_E(\delta_i) = \{(\alpha_G)^*\}$ and $e_E = e_G\{p_i \leftrightarrow \delta_i\} \cup (e_G)^*$ in rest, $\mathcal{L}_E(\delta_i) = (\widehat{0}, C_i), \mathcal{L}_E((\alpha_F)^*) = \widehat{0}$ and $\mathcal{L}_E = \mathcal{L}_G \cup \mathcal{L}^*|_{(A_F \cup X_F \setminus \{\alpha_F\})^*}$ in rest.

We prove that $E \cong F$. Let $\zeta = \{(a, a), ((a)^*, a) \in A_E \times A_F \mid a \in A_G \setminus B\}$ and $\mathfrak{R} = \{(x, x), ((x)^*, x) \mid x \in X_G\} \cup \{((\gamma_i)^*, \gamma_i), ((\alpha_F)^*, \gamma_i) \mid \gamma_i \in \Gamma\} \cup \{(\delta_i, \gamma_i) \mid \delta_i \in \Delta, \gamma_i \in \Gamma\}$, $\phi_{bn} =$

$\{(n_i^{bn}, n_i^{bn}), ((n_i^{bn})^*, n_i^{bn}) \mid n_i^{bn} \in bn(G)\}$ and $\phi_{pn} = \{(n_i^{pn}, n_i^{pn}), ((n_i^{pn})^*, n_i^{pn}) \mid n_i^{pn} \in pn(G)\}$. These relations fulfill the requirements of the definition 5.9. $\square$

Proof of Theorem 6.13. 1. Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ be an A-graph, and let $\langle f_i, R^i, S^i \rangle_{i \geq 1}$ be a copies generator. Let $g : X \cup A \to \mho \setminus \cup_{i \in \mathbb{N}}(X_i \cup A_i)$ be a substitution that generates an A-graph $G' \cong G$ having the bound names and the private names distinct of those in G. Consider $\langle g_i, R^i, S^i \rangle_{i \geq 0}$ defined by $g^0 = g$ and $g^i = f^i$. $\langle f_i, R^i, S^i \rangle_{i \geq 0}$ is a copies generator of G. So, $G' \parallel G^\infty \cong G^\infty$. But $G \cong G'$. So, $G \parallel G^\infty \cong G^\infty$. Further we use the commutativity of $\parallel$.

2. Suppose that $H = (\!|G|\!)$. We proved that $G \parallel G^\infty \cong G^\infty \parallel G \cong G^\infty$, so $(\!|G \parallel G^\infty|\!) = (\!|G^\infty \parallel G|\!) = (\!|G^\infty|\!)$. But $(\!|G \parallel G^\infty|\!) = (\!|G|\!) \parallel (\!|G^\infty|\!) = H \parallel H^\infty$, $(\!|G^\infty \parallel G|\!) = (\!|G^\infty|\!) \parallel (\!|G|\!) = H^\infty \parallel H$, and $(\!|G^\infty|\!) = H^\infty$. $\square$

Proof of Theorem 6.14. 1. $\widehat{rec}G = (recP)_{\{\gamma\}}(\{P\}_p \parallel G) \cong (\{P\}_p \parallel G)_{\{\delta\}}\{P \leftarrow F\}$ where $F \cong \widehat{rec}G$ due to the corollary 6.12. But $(\{P\}_p \parallel G)_{\{\delta\}}\{P \leftarrow F\} \cong F \parallel G \cong \widehat{rec}G \parallel G$ by the theorem 6.8.

2. Suppose that $H = (\!|G|\!)$. We proved that $G \parallel (\widehat{rec}G) \cong (\widehat{rec}G) \parallel G \cong \widehat{rec}G$, so $(\!|G \parallel (\widehat{rec}G)|\!) = (\!|(\widehat{rec}G) \parallel G|\!) = (\!|\widehat{rec}G|\!)$. But $(\!|G \parallel (\widehat{rec}G)|\!) = (\!|G|\!) \parallel (\!|(\widehat{rec}G)|\!) = H \parallel (\widehat{rec}H)$, $(\!|(\widehat{rec}G) \parallel G|\!) = (\!|\widehat{rec}G|\!) \parallel (\!|G|\!) = (\widehat{rec}H) \parallel H$, and $(\!|\widehat{rec}G|\!) = \widehat{rec}H$. $\square$

Proof of the Theorem 7.1. In order to prove this theorem we will introduce and prove first a few lemmas.

Lemma 12.1 (Parallel decomposition). *Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ be an A-graph and $\beta \in e_\alpha$. Let F be an A-graph congruent with the one generated by all the paths in G starting in α and crossing β, and let E be an A-graph congruent with the one generated by all paths in G starting in α and not crossing β, such that F and E have disjunct domains. Then $G \cong F \parallel E$.*

Proof. Suppose that $F = \langle \mathcal{E}_F, \mathcal{L}_F \rangle$ with $\mathcal{E}_F = \langle X_F, A_F, e_F, \alpha_F \rangle$ and $E = \langle \mathcal{E}_E, \mathcal{L}_E \rangle$ with $\mathcal{E}_E = \langle X_E, A_E, e_E, \alpha_E \rangle$. We suppose in addition that $bn(F) \cap bn(E) = pn(F) \cap pn(E) = \emptyset$. Let $G' = F \parallel E$ defined by $G' = \langle \mathcal{E}', \mathcal{L}' \rangle$ with $\mathcal{E}' = \langle X', A', e', \alpha' \rangle$. We prove that $G \cong G'$. Consider the A-graph $G_1 = \langle \mathcal{E}_1, \mathcal{L}_1 \rangle$ with $\mathcal{E}_1 = \langle X_1, A_1, e_1, \alpha \rangle$ generated by all the paths in G initiated in α and crossing β and $G_2 = \langle \mathcal{E}_2, \mathcal{L}_2 \rangle$ with $\mathcal{E}_2 = \langle X_2, A_2, e_2, \alpha \rangle$ generated by all the paths not crossing β. Then $G_1 \cong F$ by $\zeta_F, \mathfrak{R}_F, \phi_{bn}^F$ and ϕ_{pn}^F, and $G_2 \cong E$ by $\zeta_E, \mathfrak{R}_E, \phi_{bn}^E$ and ϕ_{pn}^E. To prove that $\mathcal{E}' \equiv_S \mathcal{E}$, we take $\zeta = \zeta_F \cup \zeta_E$ and $\mathfrak{R} = \mathfrak{R}_F \cup \mathfrak{R}_E$. Taking $\phi_{bn} = \phi_{bn}^F \cup \phi_{bn}^E$ we obtain $\mathcal{N}_G^{bn} \equiv_S \mathcal{N}_{G'}^{bn}$ and taking $\phi_{pn} = \phi_{pn}^F \cup \phi_{pn}^E$ we obtain $\mathcal{N}_G^{pn} \equiv_S \mathcal{N}_{G'}^{pn}$. $\square$

Corollary 12.2. *Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ be an A-graph having $e_\alpha = \{\beta_1, ...\beta_k\}$, and for $i = \overline{1,k}$ let G_i be an A-graph congruent with the one generated by all the paths of G starting in α and crossing β_i, any two such A-graphs having disjoint domain. Then $G \cong G_1 \parallel ... \parallel G_k$.*

Lemma 12.3 (Generalized parallel decomposition). *Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ be an A-graph having $e_\alpha = \{\beta_1, \beta_2, ...\beta_i, ...\}$, and for $i \in \mathbb{N}$ let G_i be an A-graph congruent with the one generated by all the paths of G starting in α and crossing β_i, any two such A-graphs having disjoint domain. Then $G \cong \parallel_{i \in \mathbb{N}} G_i$.*

Proof. Goes similarly with the one of the Theorem 12.1. $\square$

Lemma 12.4 (Ambient decomposition). *If $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ is an A-graph having $e_\alpha = \{\beta\}$ with $\beta \in X$, $\mathcal{L}(\beta) = m = (n, \varepsilon) \in \mathcal{M}_{amb}$, then $G \cong m_\beta \lfloor G' \rceil$ where $G' = \langle \mathcal{E}', \mathcal{L}' \rangle$ with $\mathcal{E}' = \langle X', A', e', \alpha \rangle$, $X' = X \setminus \{\beta\}, A' = A, e'(\alpha) = e(\beta), \mathcal{L}'(\alpha) = \widehat{1}$ and $\mathcal{L}' = \mathcal{L}, e' = e$ on $X \cup A \setminus \{\beta\}$.*

Proof. We construct $m_\beta \lfloor G' \rceil$ starting from G'. Further, taking $\zeta = id_A$, $\mathfrak{R} = id_X$, $\phi_{bn} = id_{bn(G)}$ and $\phi_{pn} = id_{pn(G)}$ we obtain the congruence. $\square$

Lemma 12.5 (The capability decomposition). *Assume that $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ and $e(\alpha) = \{\beta\}$ with $\mathcal{L}(\beta) = (m, \langle c, C \rangle)$. Then $G \cong c.G'$ where $G' = \langle \mathcal{E}, \mathcal{L}' \rangle$ with $\mathcal{L}'(\beta) = (m, C)$ and $\mathcal{L}' = \mathcal{L}$ for the rest of the system.*

Proof. We start from G' and we construct $c.G'$. If $c \neq (n_i^{bn})$ then $G = c.G'$, hence $G \cong c.G$. If $c = (n_i^{bn})$, we construct the A-graph $G'' = C_1.G'$ and we prove that $G'' \cong G$. We choose $j \notin arity_{G'}^{bn}(n)$ and we define $\mathcal{L}''(\beta) = (n_j^{bn} \leftrightarrow n_i^{bn})(m, \langle (n_i^{bn}), C \rangle)$ and $\mathcal{L}'' = (n_j^{bn} \leftrightarrow n_i^{bn}) \circ \mathcal{L}'$ for the rest. We have $\mathcal{E} = \mathcal{E}' = \mathcal{E}''$, hence $\mathcal{E} \equiv_S \mathcal{E}''$ by $\zeta = id_A$ and $\mathfrak{R} = id_X$. $\mathcal{N}_G^{pn} = \mathcal{N}_{G'}^{pn} = \mathcal{N}_{G''}^{pn}$, hence $\mathcal{N}_G^{pn} \equiv_S \mathcal{N}_{G''}^{pn}$ by $\phi_{pn} = id_{pn(G)}$. Now, taking $\phi_{bn} = (n_i^{bn} \leftrightarrow n_j^{bn})$ we obtain also $\mathcal{N}_G^{bn} \equiv_S \mathcal{N}_{G''}^{bn}$, and these functions fulfill the fourth condition of the definition 5.9. □

Corollary 12.6. *Suppose that $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ and $e(\alpha) = \{\beta\}$ with $\mathcal{L}(\beta) = (m, \langle C_1, C_2 \rangle)$. Then $G \cong C_1.G'$ where $G' = \langle \mathcal{E}, \mathcal{L}' \rangle$ with $\mathcal{L}'(\beta) = (m, C_2)$ and $\mathcal{L}' = \mathcal{L}$ for the rest of the system.*

Lemma 12.7 (The private label decomposition). *Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ be an A-graph prefixed by (νn_i^{pn}), i.e. $e(\alpha) = \{\beta\}$ with $\mathcal{L}(\beta) = (m, \langle (\nu n_i^{pn}), C \rangle)$, and $n \in \Lambda \setminus fn(G)$. Let $G' = \langle \mathcal{E}, \mathcal{L}' \rangle$ having the same flat system as G, $\mathcal{L}'(\beta) = (n \leftrightarrow n_i^{pn})(m, C)$ and $\mathcal{L}' = (n \leftrightarrow n_i^{pn}) \circ \mathcal{L}$ in rest. Then $G \cong (new\ n)G'$.*

Proof. We start with G' and construct $(new\ n)G'$. Because $i \notin arity_{G'}^{pn}(n)$ we can use the substitution $(n \leftrightarrow n_i^{pn})$ in our construction. Doing so we obtain $(new\ n)G' = G$, hence $(new\ n)G' \cong G$. □

Lemma 12.8 (The recursive decomposition). *Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ be an A-graph for which there exists a nonempty set $\Gamma = \{\gamma_1, ...\gamma_k\} \subset X$ such that $G \cong G_i$ for each $i = \overline{1,k}$, where G_i is the A-subgraph of G having γ_i as the root. Assume that $\mathcal{L}(\gamma_i) = (\widehat{0}, C_i)$, $P \in \Pi$ is a name not used in the labels of G and $\{p_1, ...p_k\} \subset \mho \setminus (X \cup A)$. Then $G \cong (rec.P)_\Delta.G'$, where $G' = \langle \mathcal{E}', \mathcal{L}' \rangle$ with $\mathcal{E}' = \langle X', A', e', \alpha \rangle$, $X' = X \setminus (\cup_{i=\overline{1,k}} X_i)$, $A' = A \cup \{p_1, ...p_k\}$, $e'(x) = e(x)\{\gamma_i \leftarrow p_i\}$ and $\mathcal{L}'(p_i) = (P, C_i)$.*

Proof. Suppose that $G_i = \langle \mathcal{E}_i, \mathcal{L}_i \rangle$ with $\mathcal{E}_i = \langle X_i, A_i, e_i, \gamma_i \rangle$ and that $G_i \cong G$ is defined by $\zeta_i, \mathfrak{R}_i, \phi_{bn}^i$ and ϕ_{pn}^i. Δ has to satisfy the requirements of the definition 6.13, hence $\Delta = \{\delta_1, ...\delta_k\}$. We construct $F = (rec.P)_\Delta.G$. $X_F = X' \cup \Delta = (X \setminus (\cup_{i=\overline{1,k}} X_i)) \cup \Delta$, $A_F = A' \setminus \{p_1, ...p_k\} = A$, $e_F(\delta_i) = (e(\alpha))\{p_j \leftrightarrow \delta_j\}$, $e_F = e' \circ \{p_j \leftrightarrow \delta_j\}$ in rest, $\mathcal{L}_F(\delta_i) = (\widehat{0}, C_i)$ and $\mathcal{L}_F = \mathcal{L}'$ in rest. We prove that $G \cong F$. Let $\eta = id_{X'} \cup \{(\gamma_i, \delta_i) \mid i = 1, k\}$. We define $\zeta = \cup_{i=1,k} \zeta_i$, $\mathfrak{R} = \eta \cup \mathfrak{R}_i \circ \eta$, $\phi_{bn} = \cup_{i=1,k} \phi_{bn}^i$ and $\phi_{pn} = \cup_{i=1,k} \phi_{pn}^i$. These define the congruence. □

These lemmas being proved, we can prove the representation theorem.

Let $G = \langle \mathcal{E}, \mathcal{L} \rangle$ with $\mathcal{E} = \langle X, A, e, \alpha \rangle$ be an A-graph. Suppose that G is wellfounded. For the wellfounded A-graphs we define the structural depth inductively by:

1. $depth(\{P\}) = 0$
2. $depth(P_1 \parallel ... \parallel P_k) = max(depthP_1, ...depthP_k)$
3. $depth(C.G) = depth((new\ n)G) = depth(G)$
4. $depth(m_\beta \lfloor G \rceil) = depth(G) + 1$

We prove the theorem inductively on the depth of A-graphs. For atomical A-graphs (those having null depth) the proof is trivial. Suppose that the theorem is true for all A-graphs F having the $depth(F) \leq k$, and let G be an A-graph with $depth(G) = k$. If $e(\alpha) = \{\beta_1, ..., \beta_s\}$, the requirements of the theorem 12.1 are fulfilled, hence G can be algebraically represented by $G_1, ...G_s$ where each G_i has the property that its master ambient contains only one child. If $e(\alpha) = \{\beta\}$, because G is not atomical, $\beta \in X$ and $\mathcal{L}(\beta) = (C, n) \in \mathcal{M}_{amb}$. Using the theorems 12.7 and 12.5 a finite number of times (C is a finite sequence of capabilities) we obtain $G = C.G'$ which means that G can be algebraically represented by G'. Now G' satisfies the requirements of the theorem 12.4, hence $G' = (\varepsilon, n)_\beta \lfloor G'' \rceil$, i.e. G' can be algebraically represented by G''. However $depth(G'') = depth(G') - 1 = depth(G) - 1$, hence $depth(G'') = k - 1$ and the inductive hypothesis can be used. So G'' can be algebraically represented by atomical A-graphs; hence if G is wellfounded, it can be algebraically represented by atomical processes.

We analyze now the situation when G is non-wellfounded. For A-graphs we define the circular depth by:

1. If G is not a recursive process $circ(G) = 0$

2. $circ((recP)_\Gamma G') = circ(G') + 1$

Due to the fact that $\mathcal{E}$ is regular, it cannot contain an infinite descendent path. Therefore, as argued in the remark 4.1, it cannot contain an infinite chain of loops, each loop having as successor a subsequent one. In any such chain of loops there exists a final one, hence any A-graph has a finite circular depth. We will prove the theorem by induction on circular depth. For $circ = 0$ we are in the case of wellfounded A-hierarchies (because any non-wellfounded one has to contain loops in order to not contain infinite descendent paths) and for this case we proved the theorem. Suppose that the theorem is true for any A-graph F with $circ(F) \leq k-1$ and let $circ(G) = k$. Hence G contains some subgraphs $G_1, G_2, ...$ each of them containing a loop originated in their root and $circ(G_i) \leq k$. We will focus only on those for which $circ(G_i) = k$, the rest of them being algebraically represented by atomical A-hierarchies due to the inductive assumption. But each such G_i satisfies the requirements of the theorem 12.8 (because they have loops originated in the root). So we can decompose each G_i in $G_i = (recP)_\Gamma F_i$, hence G_i can be algebraically represented by F_i. But $circ(F_i) = circ(G_i) - 1 = k - 1$, so, by the inductive assumption, F_i can be algebraically represented by atomical A-hierarchies. G_i can therefore be represented by atomical A-hierarchies, meaning that G has the same property. The part of the theorem referring to A-hierarchies derives, trivially, from this one. □

Proof of the Theorem 8.1. P, having the recursive depth $rdepth(P)$ finite, can be represented, by using a wellfounded syntax involving the introduced operations, as a combinations of a set of atomical processes. H, due to the representation theorem can be uniquely represented as well on top of a set of atomical A-hierarchies by a wellfounded syntax. For this reason we will prove both parts of the theorems by induction on these syntaxes.

First we prove that $\ulcorner [\![P]\!] \urcorner = P$. If P is an atomical process we have $\ulcorner [\![P]\!] \urcorner = \ulcorner \{P\} \urcorner = P$. Suppose that $\ulcorner [\![P_1]\!] \urcorner = P_1$ and $\ulcorner [\![P_2]\!] \urcorner = P_2$ then $\ulcorner [\![P_1|P_2]\!] \urcorner = \ulcorner [\![P_1]\!] \parallel [\![P_2]\!] \urcorner = \ulcorner [\![P_1]\!] \urcorner | \ulcorner [\![P_2]\!] \urcorner = P_1|P_2$. In the same way goes for denumerable composition.
Suppose that $\ulcorner [\![P]\!] \urcorner = P$, then $\ulcorner [\![C.P]\!] \urcorner = \ulcorner C.[\![P]\!] \urcorner = C. \ulcorner [\![P]\!] \urcorner = C.P$, $\ulcorner [\![n[P]]\!] \urcorner = \ulcorner (n, \varepsilon) \lfloor [\![P]\!] \rceil \urcorner = n[\ulcorner [\![P]\!] \urcorner] = n[P]$, $\ulcorner [\![(\nu n)P]\!] \urcorner = \ulcorner (new\ n)[\![P]\!] \urcorner = (\nu n) \ulcorner [\![P]\!] \urcorner = (\nu n)P$ and $\ulcorner [\![!P]\!] \urcorner = !P$ in both models of replication.

$[\![\ulcorner H \urcorner]\!] = H$ is proved in the same way. □

The proof of the Theorem 8.3. This result is proved by using the next Lemma.

Lemma 12.9. *We have the following properties:*

1. $[\![P]\!] \approx [\![P]\!]$
2. $[\![P]\!] \approx [\![Q]\!] \Rightarrow [\![Q]\!] \approx [\![P]\!]$
3. $[\![P]\!] \approx [\![Q]\!], [\![Q]\!] \approx [\![R]\!] \Rightarrow [\![P]\!] \approx [\![R]\!]$
4. $[\![P]\!] \approx [\![Q]\!] \Rightarrow [\![(\nu n)P]\!] \approx [\![(\nu n)Q]\!]$
5. $[\![P]\!] \approx [\![Q]\!] \Rightarrow [\![P|R]\!] \approx [\![Q|R]\!]$
6. $[\![P]\!] \approx [\![Q]\!] \Rightarrow [\![!P]\!] \approx [\![!Q]\!]$
7. $[\![P]\!] \approx [\![Q]\!] \Rightarrow [\![n[P]]\!] \approx [\![n[Q]]\!]$
8. $[\![(\nu n)(m[P])]\!] \approx [\![m[(\nu n)P]]\!], \quad n \neq m$
9. $[\![!(P|Q)]\!] \approx [\![!P|!Q]\!]$
10. $[\![!nil]\!] \approx [\![nil]\!]$
11. $[\![!!P]\!] \approx [\![!P]\!], [\![!P]\!] \approx [\![P|!P]\!]$
12. $[\![P]\!] \approx [\![Q]\!] \Rightarrow [\![M.P]\!] \approx [\![M.Q]\!]$
13. $[\![P]\!] \approx [\![Q]\!] \Rightarrow [\![(n).P]\!] \approx [\![(n).Q]\!]$
14. $[\![P]\!] \approx [\![\varepsilon.P]\!]$
15. $[\![(M.M').P]\!] \approx [\![M.M'.P]\!]$
16. $[\![(\nu n)(\nu m)P]\!] \approx [\![(\nu m)(\nu n)P]\!]$
17. $[\![(\nu n)nil]\!] \approx [\![nil]\!]$
18. $[\![(\nu n)P|Q]\!] \approx [\![P|(\nu n)Q]\!], n \notin fn(P)$
19. $[\![P|nil]\!] \approx [\![P]\!]$
20. $[\![(P|Q)|R]\!] \approx [\![P|(Q|R)]\!]$
21. $[\![P|Q]\!] \approx [\![Q|P]\!]$
22. $[\![(x).P]\!] \approx [\![(y).P(x \leftarrow y)]\!]$ *if* $y \notin fn(P)$
23. $[\![(\nu n)P]\!] \approx [\![(\nu m)P(n \leftarrow m)]\!]$ *if* $m \notin fn(P)$
24. $[\![\widehat{0}[P]]\!] \approx [\![P]\!]$
25. $[\![(recX.X)]\!] \approx [\![nil]\!]$
26. $[\![(recX.P)]\!] \approx [\![P\{X \leftarrow (recX.P)\}]\!]$
27. $[\![P]\!] \approx [\![Q]\!] \Rightarrow [\![(recX.P)]\!] \approx [\![(recX.Q)]\!]$
28. $\forall i \in \mathbb{N}\ [\![P_i]\!] \approx [\![Q_i]\!] \Rightarrow [\![|_{i \in \mathbb{N}} P_i]\!] \approx [\![|_{i \in \mathbb{N}} Q_i]\!]$

Proof of the Lemma 12.9. The first three properties derive from the fact that $\approx$ is an equivalence relation. The properties 4, 5, 7, 12, 13 and 28 are consequences of the fourth condition of the definition 7.2. The properties 8, 17 and 18 are consequences of the first condition of the definition 7.2. The 14th and 15th properties derive from the fact that $H = \varepsilon.H$ and respectively $(M.M').H = M.M'.H$. The property 16 is a consequence of the theorem 6.7. The properties

19 and 24 derive from the second condition of the definition 7.2. 20 and 21 derive from associativity and respectively commutativity of $\|$. 22 and 23 are consequences of the fact that two A-graphs are congruent if they are identical up to the renaming of input names and respectively of new names. 25 derive from the definition 6.13, 26 is a consequence of the theorem 6.11, and 27 of the theorem 6.10.

Now we prove the properties involving replication. The property 9: $(\llbracket P \rrbracket \parallel \llbracket Q \rrbracket)^\infty \approx \llbracket P \rrbracket^\infty \parallel \llbracket Q \rrbracket^\infty$. We construct a copies generator for $\llbracket P \rrbracket \parallel \llbracket Q \rrbracket$ and we divide it in two: one for nodes of $\llbracket P \rrbracket$ and one for the nodes of $\llbracket Q \rrbracket$.

The property 10: $\llbracket nil \rrbracket^\infty \approx \llbracket nil \rrbracket$ is a consequence of the point (c) of the second condition of the definition 7.2. 11 derives from the theorem 6.13. □

Coming back to the proof of the Theorem 8.3, we have:

1. Is a direct consequence of the lemma 12.9

2. If $H_1 \approx H_2$ then $\ulcorner H_1 \urcorner \equiv \ulcorner H_2 \urcorner$. We prove this inductively on the structure of A-hierarchies using the representation theorem. Hence it is sufficient to prove that:

1. (a) $\ulcorner (new\ n)(H_1 \parallel H_2) \urcorner \equiv \ulcorner H_1 \parallel (new\ n) H_2 \urcorner$ if $n \notin fn(H_1)$
   (b) $\ulcorner (new\ n)(m,\varepsilon)\lfloor H \rceil \urcorner \equiv \ulcorner (m,\varepsilon)\lfloor (new\ n) H \rceil \urcorner$ if $n \neq m$
   (c) $\ulcorner (new\ n) nil \urcorner \equiv \ulcorner nil \urcorner$

2. (a) $\ulcorner \widehat{0}\lfloor H \rceil \urcorner \equiv \ulcorner H \urcorner$ (b) $\ulcorner H \parallel \{nil\} \urcorner \equiv \ulcorner H \urcorner$ (c) $\ulcorner \{nil\}^\infty \urcorner \equiv \ulcorner \{nil\} \urcorner$

3. if $\ulcorner H_1 \urcorner \equiv \ulcorner H_2 \urcorner$ then
   (a) $\ulcorner (new\ n) H_1 \urcorner \equiv \ulcorner (new\ n) H_2 \urcorner$ (b) $\ulcorner H_1 \parallel H_3 \urcorner \equiv \ulcorner H_2 \parallel H_3 \urcorner$ (c) $\ulcorner n\lfloor H_1 \rceil \urcorner \equiv \ulcorner n\lfloor H_2 \rceil \urcorner$
   (d) $\ulcorner C.H_1 \urcorner \equiv \ulcorner C.H_2 \urcorner$ (e) $\ulcorner (recP).H_1 \urcorner \equiv \ulcorner (recP).H_2 \urcorner$
   (f) if $\ulcorner H_i \urcorner \equiv \ulcorner H'_i \urcorner$ for $i \in \mathbb{N}$ then $\ulcorner \|_{i\in\mathbb{N}} H_i \urcorner \equiv \ulcorner \|_{i\in\mathbb{N}} H'_i \urcorner$

4. (a) $\ulcorner H \urcorner \equiv \ulcorner H \urcorner$ (b) $\ulcorner H_1 \urcorner \equiv \ulcorner H_2 \urcorner \Rightarrow \ulcorner H_2 \urcorner \equiv \ulcorner H_1 \urcorner$
   (c) $\ulcorner H_1 \urcorner \equiv \ulcorner H_2 \urcorner, \ulcorner H_2 \urcorner \equiv \ulcorner H_3 \urcorner \Rightarrow \ulcorner H_1 \urcorner \equiv \ulcorner H_3 \urcorner$

   All these can be easily verified using the definition 8.1.

□

The proof of the Theorem 8.5. We use induction on structures as before to prove that if $P \longrightarrow Q$ then $\llbracket P \rrbracket \Longrightarrow \llbracket Q \rrbracket$ and if $H_1 \Longrightarrow H_2$ then $\ulcorner H_1 \urcorner \longrightarrow \ulcorner H_2 \urcorner$. We prove this for each reduction rule. Once this is done, the four points of the theorems can be derived trivially by taking into account the representation theorem for each case.

($\Rightarrow$) If $P \longrightarrow Q$ then $\llbracket P \rrbracket \Longrightarrow \llbracket Q \rrbracket$.
First we prove this for each reduction rule of Ambient Calculus:
(In-Rule): $\llbracket n[in\, m.P|Q]|m[R] \rrbracket \Longrightarrow \llbracket m[n[P|Q]|R] \rrbracket$. We have $\llbracket n[in\, m.P|Q]|m[R] \rrbracket = (n,\varepsilon)\lfloor in\, m.\llbracket P \rrbracket \parallel \llbracket Q \rrbracket \rceil \parallel (m,\varepsilon)\lfloor \llbracket R \rrbracket \rceil$ and $\llbracket m[n[P|Q]|R] \rrbracket = (m,\varepsilon)\lfloor \llbracket R \rrbracket \parallel (n,\varepsilon)\lfloor \llbracket P \rrbracket \parallel \llbracket Q \rrbracket \rceil \rceil$.
But (In-Rule) for A-hierarchies gives us $(n,\varepsilon)\lfloor in\ m.\llbracket P \rrbracket \parallel \llbracket Q \rrbracket \rceil \parallel (m,\varepsilon)\lfloor \llbracket R \rrbracket \rceil \Longrightarrow (m,\varepsilon)\lfloor \llbracket R \rrbracket \parallel (n,\varepsilon)\lfloor \llbracket P \rrbracket \parallel \llbracket Q \rrbracket \rceil \rceil$ q.e.d. We can prove, in the same way, that the rules (Out-Rule), (Open-Rule) and (Comm-Rule) are preserved by $\llbracket\ \rrbracket$.

Further we suppose that for two processes P and Q we have: If $P \longrightarrow Q$ then $\llbracket P \rrbracket \Longrightarrow \llbracket Q \rrbracket$. We prove that (Par-Rule) is conserved. We have $\llbracket P|R \rrbracket = \llbracket P \rrbracket \parallel \llbracket R \rrbracket$ and $\llbracket Q|R \rrbracket = \llbracket Q \rrbracket \parallel \llbracket R \rrbracket$. Using (Par-Rule) over A-Hierarchies we obtain $\llbracket P \rrbracket \parallel \llbracket R \rrbracket \Longrightarrow \llbracket Q \rrbracket \parallel \llbracket R \rrbracket$, so $\llbracket P|R \rrbracket \Longrightarrow \llbracket Q|R \rrbracket$ q.e.d. In the same way we can prove the property for (Amb-Rule), (New-Rule) and (DenPar-Rule).

($\Leftarrow$) If $H_1 \Longrightarrow H_2$ then $\ulcorner H_1 \urcorner \longrightarrow \ulcorner H_2 \urcorner$. The proof goes inductively as for the reversed implication. □

Proof of the Theorem 9.1. If $P \in \mathfrak{P}^\infty$ then $\exists k < \aleph_0$ such that $depth(P) = k$. The syntax of Ambient Calculus allows only wellfounded programs, hence the structure tree of P has a finite depth. Then the length of the longest branch in the syntax tree, if we do not count the transparent nodes, gives us $k < \aleph_0$. □

Proof of the Theorem 9.6. $(recX.X)^{\alpha} = X^{\alpha-1}\{X^{\beta} \leftarrow (recX.X)^{\beta}\}$ for all $\beta \leq \alpha-1$. Hence $(recX.X)^{\alpha} = (recX.X)^{\alpha-1} = ... = (recX.X)^{0} \equiv nil$ □

Proof of the Corollary 9.7. We prove it by induction on the structure of P. If P is an atomical process then the property is true for $k = 0$. Suppose that the property is true for P (it exists k) and we will prove it for $M.P$ and for $n[P]$. Because $(M.P)^{k+s} = M.P^{k+s}$ we obtain that $(M.P)^{k+s} = M.P$ for any $s < \aleph_0$. Because $(n[P])^{k+1} = n[P^k]$ we obtain that the property is true for $n[P]$ as well. Suppose now that the property is true for P and for Q, having $k, l < \aleph_0$. Then $(P|Q)^{max(k,l)} = P^{max(k,l)}|Q^{max(k,l)}$. But $max(k,l) > k, max(k,l) > l$, so we can use the inductive hypothesis. Thus $P|Q$ satisfies the property. In the same way $|_{i\in\mathbb{N}}P_i$ satisfies the property, hence also $!P$. □

Proof of the Theorem 9.8. 1. $u(n[P])$ is $nil, n^1[P^0], n^2[P^1], ..., n^i[P^{i-1}], ...$. But $nil, n^1, n^2, ...n^i, ...$ is the unfolding of $n[\ ]$ and $P^0, P^1, ...P^i, ...$ is the unfolding of P. Then, by definition 9.6, $u(n[P]) \sim u(n)[u(P)]$. The rest of the theorem can be verified in the same way. □

Proof of the Theorem 9.9. It is trivial to prove reflexivity and symmetry. We prove further the transitivity. Suppose we have $\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3 \in Seq(\mathfrak{P}^{\infty})$ with $\mathcal{P}^1 \sim \mathcal{P}^2$ and $\mathcal{P}^2 \sim \mathcal{P}^3$. We prove this property by structural induction.

Suppose that $\mathcal{P}^1 \sim \mathcal{P}^2$ due to the first condition of the definition 9.6, i.e. $P^1_{k+s} \equiv P^2_s$ for any $s \in \mathbb{N}$. If $\mathcal{P}^2 \sim \mathcal{P}^3$ due to the same condition, we have $P^2_s \equiv P^3_{l+s}$ for any s, hence $P^1_{k+s} \equiv P^2_s \equiv P^3_{l+s}$ for any s, so, due to the first condition of the definition 9.6, $\mathcal{P}^1 \sim \mathcal{P}^3$. If $\mathcal{P}^2 \sim \mathcal{P}^3$ due to the second condition of the definition 9.6, then $\mathcal{P}^2 = \mathcal{R}|\mathcal{S}$, $\mathcal{P}^3 = \mathcal{T}|\mathcal{U}$, with $\mathcal{R} \sim \mathcal{T}$ and $\mathcal{S} \sim \mathcal{U}$. This means that $P^2_s \equiv R_s|S_s$, and because $P^1_s \equiv P^2_{k+s}$ we obtain that $P^1_{k+s} \equiv R_s|S_s$ for any s, hence $\mathcal{P}^1 = \mathcal{R}|\mathcal{S}$. Due to the second condition of the definition 9.6 we obtain that $\mathcal{P}^1 \sim \mathcal{P}^3$. In the same way the property can be proved if $\mathcal{P}^2 \sim \mathcal{P}^3$ is due to the other four conditions of the definition 9.6.

The next step is to suppose that $\mathcal{P}^2 \sim \mathcal{P}^3$, due to the first condition of the definition 9.6 and to analyze, one by one, the cases for $\mathcal{P}^1 \sim \mathcal{P}^2$ as before.

For the inductive step, we suppose that $\mathcal{P}^1 \sim \mathcal{P}^2$ and $\mathcal{P}^2 \sim \mathcal{P}^3$ are satisfied by any condition but the first. Using the possible configuration of the processes, it is trivial to verify that the requirements of the theorem are fulfilled iff they are fulfilled for the components of $\mathcal{P}^1, \mathcal{P}^2$, and $\mathcal{P}^3$. We use here also the fact that if one of the couples $\mathcal{P}^1 \sim \mathcal{P}^2$, and $\mathcal{P}^2 \sim \mathcal{P}^3$ is satisfied due to the conditions 2, 3, or 4, then the other must be satisfied by one of the same conditions; if one is satisfied by the condition 5, then the other must be satisfied also by 5; and if the first is due to the condition 6 for $M \neq \varepsilon$, the second has to have the same property. □

Proof of the Theorem 9.10. In order to prove this theorem we prove the next Lemma.

Lemma 12.10. *We have the following properties:*

1. $u(P) \sim u(P)$
2. $u(P) \sim u(Q) \Rightarrow u(Q) \sim u(P)$
3. $u(P) \sim u(Q), u(Q) \sim u(R) \Rightarrow u(P) \sim u(R)$
4. $u(P) \sim u(Q) \Rightarrow u((\nu n)P) \sim u((\nu n)Q)$
5. $u(P) \sim u(Q) \Rightarrow u(P|R) \sim u(Q|R)$
6. $u(P) \sim u(Q) \Rightarrow u(!P) \sim u(!Q)$
7. $u(P) \sim u(Q) \Rightarrow u(n[P]) \sim u(n[Q])$
8. $u((\nu n)m[P]) \sim u(m[(\nu n)P]), \quad n \neq m$
9. $u(!(P|Q)) \sim u(!P|!Q)$
10. $u(!0) \sim u(0)$
11. $u(!!P) \sim u(!P), u(!P) \sim u(P|!P)$
12. $u(P) \sim u(Q) \Rightarrow u(M.P) \sim u(M.Q)$
13. $u(P) \sim u(Q) \Rightarrow u((n).P) \sim u((n).Q)$
14. $u(P) \sim u(\varepsilon.P)$
15. $u((M.M').P) \sim u(M.M'.P)$
16. $u((\nu n)(\nu m)P) \sim u((\nu m)(\nu n)P)$
17. $u((\nu n)0) \sim u(0)$
18. $u((\nu n)P|Q) \sim u(P|(\nu n)Q), n \notin fn(P)$
19. $u(P|0) \sim u(P)$
20. $u((P|Q)|R) \sim u(P|(Q|R))$
21. $u(P|Q) \sim u(Q|P)$
22. $u((x).P) \sim u((y).P(x \leftarrow y))$ *if* $y \notin fn(P)$
23. $u((\nu n)P) \sim u((\nu m)P(n \leftarrow m))$ *if* $m \notin fn(P)$
24. $u(\widehat{0}[P]) \sim u(P)$
25. $u((recX.X)) \sim u(0)$
26. $u((recX.P)) \sim u(P\{X \leftarrow (recX.P)\})$
27. $u(P) \sim u(Q) \Rightarrow u((recX.P)) \sim u((recX.Q))$
28. $\forall i \in \mathbb{N}\ u(P_i) \sim u(Q_i) \Rightarrow u(P_i|_{i\in\mathbb{N}}) \sim u(Q_i|_{i\in\mathbb{N}})$

Proof of the Lemma 12.10. The first three properties derive from the fact that $\sim$ is an equivalence relation. The definition 9.6 combined with the theorem 9.8 gives us the properties 4, 5, 6, 7, 12, 13, and 28. Because $((\nu n)m[P])^{\alpha} \equiv (m[(\nu n)P])^{\alpha}$ if $n \neq m$ for any α, we obtain the property 8, and in the same mode, using the structural congruence of the unfoldings of rank α for any α, we can argue for the properties 9, 10, 11, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25. The property 24 can be proven using the fact that $(\widehat{0}[P])^\alpha \equiv P^{\alpha-1}$.

For proving the properties 26 and 27 we use the Corollary 9.5.

The property 26: $u((recX.P)) \sim u(P\{X \leftarrow (recX.P)\})$. We have $(recX.P)^\alpha = P^{\alpha-1}\{X^\beta \leftarrow (recX.P)^\beta\}$ for $\beta < \alpha$. But $(P\{X \leftarrow (recX.P)\})^{\alpha-1} \equiv P^{\alpha-1}\{X^\beta \leftarrow (recX.P)^\beta\}$ for $\beta < \alpha$ due to the Corollary 9.5. Hence $(recX.P)^\alpha \equiv (P\{X \leftarrow (recX.P)\})^{\alpha-1}$ for any α which proves that $u((recX.P)) \sim u(P\{X \leftarrow (recX.P)\})$.

The property 27: $u(P) \sim u(Q) \Rightarrow u((recX.P)) \sim u((recX.Q))$. It can be proved by a double induction. We take each case of the definition 9.6 that can define $u(P) \sim u(Q)$ and prove that $u((recX.P)) \sim u((recX.Q))$ inductively on the pattern of unfoldings. □

Returning to our theorem:
1. is a consequence of the properties stated in the theorem 12.10.
2. (⇒) If $u(P) \sim u(Q)$ then $P \equiv Q$.
Because $P, Q \in \mathfrak{P}^\infty$, due to the corollary 9.7, $\exists k, l < \aleph_0$ such that for any s $P \equiv P^{k+s}$ and $Q \equiv Q^{l+s}$. Because $u(P) \sim u(Q)$, $\exists m, n < \aleph_0$ such that $P^{m+s} \equiv Q^{n+s}$ for any s. Hence $P \equiv P^{k+l+m+n} \equiv Q^{k+l+m+n} \equiv Q$. □

Proof of the Theorem 9.14. $(recX.X), (recX.P|X), (recX.X|X...|X|P), (recX.!X|P)$ are all possible linear processes. We defined the first as identical with nil, while all the rest are equivalent with $!P$ as proven in theorem 9.13. □

Proof of the Theorem 9.17. As for the theorem 9.1 we use the length of the longest branch of the structure tree of the process (that has to be wellfounded) but now we count also the transparent nodes. □

Proof of the Theorem 11.3. It is trivial to verify this condition, one by one, for each of the 9 rules in the definition 11.2. □