Extensionality of Spatial Observations in Distributed Systems

Luís Caires [1] and Hugo Torres Vieira [2]

CITI / Departamento de Informática, FCT Universidade Nova de Lisboa, Portugal

Abstract

We discuss the tensions between intensionality and extensionality of spatial observations in distributed systems, showing that there are natural models where extensional observational equivalences may be characterized by spatial logics, including the composition and void operators. Our results support the claim that spatial observations do not need to be always considered intensional, even if expressive enough to talk about the structure of systems. For simplicity, our technical development is based on a minimalist process calculus, that already captures the main features of distributed systems, namely local synchronous communication, local computation, asynchronous remote communication, and partial failures.

Introduction

Logical characterizations of concurrent behaviors have been introduced for a long time now. A fundamental result in the field, due to Hennessy and Milner [13], is the characterization of behavioral equivalence in process algebras as indistinguishability with respect to a modal logic. Such results are important not only theoretically, but also because of their influence in the design of practical specification languages for software systems. Hennessy-Milner logic (HML) adds to propositional operators the action modality $\langle\lambda\rangle A$, allowing the logic to observe a grain of behavior: a process satisfies $\langle\lambda\rangle A$ if it satisfies A after performing action λ. HML characterizes behavioral equivalence in the sense that two processes are strongly bisimilar if and only if they satisfy exactly the same formulas.

More recently, spatial logics for concurrency [6,9,4] have been proposed with the aim of specifying distributed behavior and other essential aspects of distributed computing systems. In general terms, these developments reflect a shift of focus in concurrency research, that has been building up from the last decade on, from the study of centralized concurrent systems to the study of general distributed systems.

[1] Luis.Caires@di.fct.unl.pt
[2] htv@di.fct.unl.pt

While centralized processes may be accurately modeled as pure objects of behavior, in distributed systems many interesting phenomena besides pure interaction, such as location dependent behavior, resource usage, and mobility, must be considered.

Present in all spatial logics for concurrency are the composition operator $A \mid B$ and the void operator $\mathbf{0}$ [4]. Intuitively, a system satisfies $A \mid B$ if it can be decomposed in two disjoint subsystems such that one satisfies A and the other satisfies B, while a system satisfies $\mathbf{0}$ if it is the empty system. The guarantee (logical adjunct of the composition operator) $A \triangleright B$, introduced in [9], allows the logic to talk about contextual properties. Namely, a process satisfies $A \triangleright B$ if whenever composed with a system that satisfies A, yields a (possibly larger) system that satisfies B. Decomposition and composition of systems as mentioned here is generally interpreted up to structural congruence, and thus structural congruence seems to play a key role in the semantics of spatial logics.

Observation of features such as spatial separation are frequently considered intensional because they usually induce fine distinctions among processes that are not substantiated by purely behavioral (extensional) observations. According to Sangiorgi [19], "A logic is intensional if it can separate terms on the basis of their internal structure, even though their behaviors are the same". Moreover, in many situations, it turns out that the logical equivalence induced by a spatial logic on processes, is not only strictly finer than behavioral congruence, but coincides with structural congruence [19,5,11,20].

These results contributed to widespread the impression that spatial observations, as those induced by spatial connectives, are intrinsically intensional, imposed extraneously so to increase the power of the observer. For example, Hirschkoff has shown [14] that if the so-called intensional connectives composition and void are removed from a spatial logic for the pi-calculus, while retaining the guarantee, one obtains a logic whose separation power precisely coincides with strong bisimulation and may then be considered extensional. The ability of the spatial connectives to capture structural congruence is also attributed to their ability to count, separate, and express arithmetical constraints, *e.g.*, about the number of subsystems of a given system. The observational power of spatial logics may then sometimes appear a bit arbitrary, in the sense that structural congruence does not have a canonical status among behavioral process equivalences, and is frequently seen just as a technical convenience, with a syntactic flavor, to ease the presentation of a calculus operational semantics.

On the other hand, it has been argued [4,2,3] that the intensional character of logical characterizations of spatiality in distributed computation may be, at least in part, incidental, and does not necessarily reflect the fundamental motivation for introducing spatial logics for concurrency. Ideally, we would like spatial observations, as captured by spatial logics, to reflect natural distinctions and similarities between distributed systems, in a context where spatial location is a relevant observable, in parity with more standard behavioral observables. We expect spatial observations of the sort, captured by spatial logic operators such as composition, to be taken modulo an intended notion of equality of the observable space-time

structure, independently on whether such equality relation is technically defined using a notion of structural congruence. If certain spatial-behavioral observations precisely capture the observable structure of a model in our sense, they would have to be considered extensional, even if able to detect aspects of spatial structure.

In this paper, we pursue the informal discussion started above in technical terms. Namely, we make precise the claim that spatial observations, including structural ones, may be understood as purely extensional in fairly natural models of distributed systems. To discuss the several issues of interest in a simplified setting, we consider a minimal distributed process calculus, obtained by extending the smallest concurrent fragment of CCS with flat anonymous locations. Our model can be seen as a general abstraction of the essence of distributed systems, already featuring all the key ingredients present in distributed process calculi, although in a possibly less refined way. Processes may synchronously communicate locally to a site through standard CCS-like synchronization, and asynchronously communicate at a distance, by means of a migration primitive. We also allow systems to nondeterministically exhibit partial failures, as in [1,12]. Notice that it is not our aim here to propose yet another distributed process calculus, but rather to set up a convenient setting to compare distributed system observational equivalences and their spatial logical characterizations.

Our technical contributions may be summarized as follows. After introducing the process calculus and its reduction semantics, we define observational equivalence by adopting the canonical notion of reduction barbed congruence. Barbed congruence [17] and reduction barbed congruence [16] are currently accepted as the standard approach to define reference behavioral equivalences for general process calculi. After showing some basic properties of reduction barbed congruence in our setting, we define strong bisimulation, an alternative coinductive characterization of observational equivalence, which is shown equivalent to reduction barbed congruence. The interesting aspect of our definition of strong bisimulation is that it contains "intensional" clauses (in the sense of [19]), namely a clause expressing separation, and a clause for observing the empty system. We then use the characterization of reduction barbed congruence in terms of strong bisimulation to identify a spatial logic characterization of both reduction barbed congruence and strong bisimulation: our logic is an extension of HML with the composition and void operators of spatial logic. The same line of development is also carried out for the weak case. In this latter setting, we prove minimality of the logic, thus showing the essential role of all of the logic operators, in particular of the spatial operators, in the intended expressive and separation power. We can verify that in both the strong and weak cases the process equivalences induced by the logics are coarser than structural congruence, and that the presence of the composition and void operators, semantically interpreted in the standard way, do not carry any lack of extensionality (with extensionality interpreted with reference to a standard observational equivalence), even if the logics can express separation and counting constraints on the structure of systems.

1 A Simple Model of Distributed Systems

In this section we present the syntax and operational semantics of our distributed process calculus. Assume given an infinite set Λ of *names*, ranged over by a, b, c.

Definition 1.1 [Actions, Processes and Networks] The sets $\mathcal{A}$ of *actions*, $\mathcal{P}$ of *processes*, and $\mathcal{N}$ of *networks* are given by:

$$\alpha ::= \bar{a} \mid a \mid \tau \qquad P, Q ::= \mathbf{nil} \mid P \mid Q \mid \alpha.P \mid \mathbf{go}.P \qquad N, M ::= \mathbf{0} \mid N \mid M \mid [P]$$

For actions we consider the output $\bar{a}$, the input a and the internal computation τ. For processes, we consider the smallest fragment of CCS featuring some form of concurrency, thus we have inaction **nil**, parallel composition $P \mid Q$, and action prefixing $\alpha.P$. On top of this, we introduce a notion of distribution by locating processes P inside sites of the form $[P]$, and by adding the migration capability $\mathbf{go}.P$ to processes, which since sites are not natively named, allows processes to non-deterministically migrate to other sites. A distributed system is thus represented by a network consisting of a collection of sites spread in space, by means of spatial composition $N \mid M$, which we will abbreviate using $\prod_{j \in J} [P^j]$ for a J-fold collection of sites. $\mathbf{0}$ stands for the empty network. We use $\mathit{fn}(N)$ to denote the set of free names of a network N, defined as usual. The operational semantics of our calculus follows, captured by the relations of structural congruence and reduction.

Definition 1.2 [Structural congruence] *Structural congruence*, noted $\equiv$, is the least congruence on processes and networks such that $(\mathcal{P}, \mathbf{nil}, \mid)$ and $(\mathcal{N}, \mathbf{0}, \mid)$ are commutative monoids, and $P \equiv Q$ implies $[P] \equiv [Q]$.

Definition 1.3 [Reduction] *Reduction*, noted $N \rightarrow M$, is the relation between processes inductively defined as follows

$$[\bar{a}.P \mid a.Q \mid R] \rightarrow [P \mid Q \mid R] \text{ (Red Comm)} \qquad [\tau.P \mid Q] \rightarrow [P \mid Q] \text{ (Red Tau)}$$

$$[\mathbf{go}.P \mid Q] \mid [R] \rightarrow [Q] \mid [P \mid R] \text{ (Red Go)} \qquad [P] \mid N \rightarrow \mathbf{0} \text{ (Red Fail)}$$

$$\frac{N \rightarrow N'}{N \mid M \rightarrow N' \mid M} \text{ (Red Cong)} \qquad \frac{N \equiv N' \rightarrow M' \equiv M}{N \rightarrow M} \text{ (Red Struct)}$$

The rule (Red Comm) specifies interaction between two processes through co-action synchronization locally inside a site, while rule (Red Tau) specifies internal action of a process. Rule (Red Go) specifies that a process prefixed by **go** may migrate to another site. Rule (Red Fail) expresses that any non-empty network may fail, thus modeling fail-stop failure of an arbitrary subsystem.

Our aim now is to define a natural notion of observational equivalence on networks. To that end, we adopt the canonical notion of barbed equivalence, according to which two systems are observationally equivalent if no context can distinguish between them by barb detection. In our case, we restrict to one-hole spatial contexts, as *e.g.*, in [1,12], hence of the form $C\,[\bullet] ::= N \mid \bullet$, for some network N.

We use the standard notion of barb observation [17], even if it assumes in a sense the existence of a global observer, which might be debatable in the context

of distributed systems. Thus a network N exhibits barb a, noted $N\downarrow_a$, if there are P, Q, M such that $N \equiv [a.P \mid Q] \mid M$, hence reflecting the fact that any external observer can get to know that an input is ready via some channel name, at some accessible site. We now define our reference observational equivalence relation.

Definition 1.4 [Strong reduction barbed congruence] *Strong reduction barbed congruence*, noted $\simeq$, is the largest symmetric relation R such that for all $(N, M) \in R$:

For all barbs a, if $N\downarrow_a$ then $M\downarrow_a$	(Barb closed)
If $N \rightarrow N'$ then there is M' s.t. $M \rightarrow M'$ and $(N', M') \in R$	(Reduction closed)
For all contexts $C\,[\bullet]$, $(C\,[N]\,, C\,[M]) \in R$	(Context closed)

We establish some standard properties of strong reduction barbed congruence, such as $\simeq$ is a congruence. Notice that we just consider in this paper, congruences under spatial (static) contexts. As explained above, this does not carry a lack of generality, given the main motivations of our development. Moreover:

Proposition 1.5 *We have* $\equiv \subset \simeq$.

Proof. The proof of $\subseteq$ follows standard lines. To prove that $\equiv$ is strictly included in $\simeq$ we may show that $[a.\mathbf{nil} \mid a.\mathbf{nil}] \simeq [a.a.\mathbf{nil}]$ but $[a.\mathbf{nil} \mid a.\mathbf{nil}] \not\equiv [a.a.\mathbf{nil}]$. □

It follows from the congruence property that strong reduction barbed congruence is closed under composition. In particular for site composition, we have:

Lemma 1.6 *Let P^i and Q^i $(i \in J)$ be collections of processes. If for all $i \in J$ we have $[P^i] \simeq [Q^i]$, then also $\prod_{j\in J}[P^j] \simeq \prod_{j\in J}[Q^j]$.*

Although Definition 1.4 is standard, with reference to the global observation of barbs in networks, observations already leak some relevant information about the distributed structure of systems. Lemma 1.7 states that strong reduction barbed congruent networks always result from an underlying one-one and onto correspondence of strong reduction barbed congruent sites. In particular, we conclude strong reduction barbed congruent networks always have the same number of sites.

Lemma 1.7 *Let M, N be networks such that $N \triangleq \prod_{j\in J}[P^j]$, where P^j $(j \in J)$ is a collection of processes, and $N \simeq M$. Then there is a collection of processes Q^j $(j \in J)$ such that $M \equiv \prod_{j\in J}[Q^j]$ and for all $j \in J$ we have $[P^j] \simeq [Q^j]$.*

Proof. (Sketch, full proof in [7]) We consider a context that holds processes that may migrate and *mark* every site of N with an input on the unique name, and we make sure that every input is located at a different site. Since M behaves the same as N under this context (and using a symmetric reasoning) we obtain that M has $\#J$ sites. We then exploit failures in N that leave only a single site active, being that this behavior must be mimicked by failures in M that also leave just one site up. These singled out sites are strong reduction barbed congruent, hence hold the same unique input, thus ensuring an unique correspondence. We then consider another context that may clean up the marker and all other foreign elements, which then allows us to conclude the sites were originally strong reduction barbed congruent.□

2 Strong Bisimilarity

Since strong reduction barbed congruence relies on universal quantification over all contexts, we now propose a more manageable characterization of observational equivalence. More concretely, we introduce a labeled transition system with the aim of capturing the contextual behavior of the networks, by means of observing process commitments, in turn expressed by transition labels. Building on such labeled transition system, a coinductive definition of bisimilarity is then presented.

The set of transition labels, noted $\mathcal{L}$, is given by $\mathcal{L} \triangleq \{\alpha \mid \alpha \in \mathcal{A}\} \cup \{[a] \mid a \in \Lambda\}$, and ranged over by λ. Transition labels reflect internal computation (τ), and output and input communication ($\bar{a}$ and a). Given the mobile capability of processes we also consider $[a]$ transitions, that will be used to observe migration of processes to the external environment. This turns out to be essential for covering the case of networks with a single site, since the enlargement of the system with a new site gives processes intending to migrate a possible destination. Given these ingredients, we define our labeled transition system as follows.

Definition 2.1 [Commitment] *Commitment*, noted $N \xrightarrow{\lambda} M$, is the relation on processes and labels inductively defined as follows

$$[\bar{a}.P \mid a.Q \mid R] \xrightarrow{\tau} [P \mid Q \mid R] \text{ (Comm)} \qquad [\tau.P \mid Q] \xrightarrow{\tau} [P \mid Q] \text{ (Tau)}$$

$$[\bar{a}.P \mid Q] \xrightarrow{\bar{a}} [P \mid Q] \text{ (Out)} \qquad [a.P \mid Q] \xrightarrow{a} [P \mid Q] \text{ (In)}$$

$$[\mathbf{go}.P \mid Q] \mid [R] \xrightarrow{\tau} [Q] \mid [P \mid R] \text{ (Go)}$$

$$[P] \mid N \xrightarrow{\tau} \mathbf{0} \text{ (Fail)} \qquad N \xrightarrow{[a]} N \mid [a.\mathbf{nil}] \text{ (Grow)}$$

$$\frac{N \xrightarrow{\lambda} N'}{N \mid M \xrightarrow{\lambda} N' \mid M} \text{ (Cong)} \qquad \frac{N \equiv N' \xrightarrow{\lambda} M' \equiv M}{N \xrightarrow{\lambda} M} \text{ (Struct)}$$

As a sanity check, we ensure τ commitments match reductions and inversely. Notice that although *e.g.*, the systems $[\mathbf{nil}] \mid [\mathbf{nil}]$ and $[\tau.\mathbf{nil}]$ have exactly the same commitment graph, they are not observationally equivalent in the light of Lemma 1.7. Thus, in order to properly capture strong reduction barbed congruence, we include in the definition of strong bisimulation two spatial clauses (referred to as "intensional clauses" in [19]). We then have:

Definition 2.2 [Strong Bisimulation] A binary relation $B \subseteq \mathcal{N} \times \mathcal{N}$ is a *strong bisimulation* if and only if it is symmetric and whenever $(N, M) \in B$ then

$$N \equiv N' \mid N'' \Rightarrow \exists M', M'' \,.\, M \equiv M' \mid M'' \wedge (N', M') \in B \wedge (N'', M'') \in B$$

$$N \equiv \mathbf{0} \Rightarrow M \equiv \mathbf{0}$$

$$N \xrightarrow{\lambda} N' \Rightarrow \exists M' \,.\, M \xrightarrow{\lambda} M' \wedge (N', M') \in B$$

We remark that the second clause in Definition 2.2 is subsumed by the third one since only void systems have no possible internal actions (due to failures), however we prefer to include it in the definition for the sake of uniformity with the

corresponding weak version, and thus avoid some extra incidentality. Notice also that the first clause properly distinguishes $[\tau.\mathbf{nil}]$ and $[\mathbf{nil}] \mid [\mathbf{nil}]$, because there is no way to split $[\tau.\mathbf{nil}]$ (up to $\equiv$) in two parts with some transition each. We prove strong bisimulations are equivalence relations closed under union, and define:

Definition 2.3 [Strong bisimilarity] *Strong bisimilarity*, noted $\sim$, is the largest strong bisimulation.

2.1 Full Abstraction

This section is devoted to proving that strong bisimilarity, as defined in Definition 2.3, characterizes strong reduction barbed congruence in a fully abstract way. The proof builds on a series of intermediate technical results.

Lemma 2.4 *Let M be a network and $P^j (j \in J)$ a collection of processes where $\prod_{j\in J}[P^j] \sim M$. Then there is a collection of processes $Q^j (j \in J)$ such that $M \equiv \prod_{j\in J}[Q^j]$ and for all $j \in J$, $[P^j] \sim [Q^j]$.*

Proof. By induction on the size of J, using the separation and emptiness clauses.□

The proof of the main result of this section (Theorem 2.6) is not technically involved, but critically depends on next Lemma 2.5, that expresses a key compositionality principle of our calculus. Notice that the basic building block of systems referred to in the statement of Lemma 2.5 is the process: since we have to take migration into account, it is essential to assure compositionality at the process level. We abbreviate collections of sites such that each one holds a collection of processes.

Lemma 2.5 *Let J be a finite set and I_j, for all $j \in J$, be a finite set. Let P_i^j and Q_i^j be processes such that for all $j \in J$ and $i \in I_j$ we have $\left[P_i^j\right] \sim \left[Q_i^j\right]$. Then*

$$\prod_{j\in J}\left[\prod_{i\in I_j} P_i^j\right] \sim \prod_{j\in J}\left[\prod_{i\in I_j} Q_i^j\right]$$

Proof. (Sketch, full proof in [7]) By coinduction on the definition of strong bisimulation. We sketch the proof for the interesting case of migration.

We exploit the grow transition using a fresh name, in the sense that it does not occur in neither one of the P_i^js and Q_i^js, which creates a possible target for migrations and allows us to isolate migrating processes, since we can decompose and observe the input on the fresh name. Using this technique and since we can establish that the newly created sites are bisimilar, we can be sure to obtain a collection of sites that respects the statement of the Lemma for any choice of target of the migration. Notice that a migration on one side need not be always matched by a migration on the other, because the migrating process can *e.g.*, be inaction, in which case, a migration may be matched by an internal computation step. □

By Lemma 2.4 and Lemma 2.5 we prove strong bisimilarity is a congruence, from which follows, in standard lines, that $\sim \subseteq \simeq$. We then prove $\simeq \subseteq \sim$, using Lemma 1.6 and Lemma 1.7 to address the structural issues. We can then state:

Theorem 2.6 (Full abstraction) *We have $\sim = \simeq$.*

2.2 Logical Characterization of Strong Bisimilarity

In this section, we characterize strong bisimilarity (and thus strong reduction barbed congruence) in logical terms, using a simple spatial logic.

Definition 2.7 [Spatial logic $\mathcal{L}_s$] Formulas are defined by the following syntax:

$$\text{(Formulas) } A, B, C ::= \mathbf{T} \;\Big|\; \neg A \;\Big|\; A \wedge B \;\Big|\; \mathbf{0} \;\Big|\; A \mid B \;\Big|\; \langle\lambda\rangle A$$

Our logic, besides the usual action modality from HML, includes the composition and void operators of spatial logics, interpreted in the standard way. For example, we may express property "network has exactly one site" by the formula $\neg\mathbf{0} \wedge \neg(\neg\mathbf{0} \mid \neg\mathbf{0})$. The semantics of the logic is given by the denotation of the formulas, i.e., a formula denotes the set of networks that satisfy it.

Definition 2.8 [Semantics of $\mathcal{L}_s$] A formula's denotation is inductively given by

$$[\![\mathbf{T}]\!] \triangleq \mathcal{N} \quad [\![\neg A]\!] \triangleq \mathcal{N} \backslash [\![A]\!] \quad [\![A \wedge B]\!] \triangleq [\![A]\!] \cap [\![B]\!] \quad [\![\mathbf{0}]\!] \triangleq \{N \mid N \equiv \mathbf{0}\}$$

$$[\![A \mid B]\!] \triangleq \{N \mid \exists N', N'' \,.\, N \equiv N' \mid N'' \wedge N' \in [\![A]\!] \wedge N'' \in [\![B]\!]\}$$

$$[\![\langle\lambda\rangle A]\!] \triangleq \{N \mid \exists N' \,.\, N \xrightarrow{\lambda} N' \wedge N' \in [\![A]\!]\}$$

We write $N \models A$ to mean $N \in [\![A]\!]$. We say that networks M and N are *logically equivalent* w.r.t. $\mathcal{L}_s$, written $M =_{\mathcal{L}_s} N$, if and only if they satisfy exactly the same formulas of $\mathcal{L}_s$, namely if and only if, for any formula A of $\mathcal{L}_s$, we have $M \models A \iff N \models A$. We now state our logical characterization result.

Theorem 2.9 (Logical Characterization of $\sim$) *We have $\sim = =_{\mathcal{L}_s}$.*

Proof. (Sketch, full proof in [7]) Proof of $\sim \subseteq =_{\mathcal{L}_s}$ follows by a standard induction on the structure of the formulas. We prove $=_{\mathcal{L}_s} \subseteq \sim$ by coinduction on the definition of strong bisimulation, using the witness $R \triangleq \{(N, M) \mid N =_{\mathcal{L}_s} M\}$. Proof of the emptiness clause is immediate. For both the separation and transition clauses we build on the fact that the image set of the transition for the latter and of all possible decompositions for the former is finite (up to structural congruence). We then exploit the finiteness of these finite sets to prove that there is a (logical equivalent) correspondence between at least one of their elements. Otherwise we could collect the finite set of all formulas that distinguish them in a conjunction that must hold for both networks, either after a decomposition or after an action, since they are logically equivalent. We then obtain our bisimilar result by coinduction. □

As a corollary we immediately conclude that $=_{\mathcal{L}_s}$ precisely characterizes $\simeq$. Thus the separation power of our spatial logic coincides with behavioral equivalence, even if it includes the basic structural connectives of composition and void, allowing it to *e.g.*, express arithmetical constraints on the number of sites in a system. We may however ask whether these structural operations are essential to characterize behavioral equivalence, in other words, whether the logic is minimal in some sense. We will give a positive answer to this question in the next section, in the more interesting case of weak behavioral equivalences.

3 Weak Bisimilarity

In this section we refine our previous results by considering a coarser observational equivalence, disregarding internal action, thus we adopt weak reduction barbed congruence as the reference observational equivalence. We denote by $\Rightarrow$ the reflexive-transitive closure of reduction ($\rightarrow$) and state that a network N weakly exhibits a barb a, noted $N\Downarrow_a$, if there is N' such that $N \Rightarrow N'$ and $N'\downarrow_a$. We then have:

Definition 3.1 [Weak reduction barbed congruence] *Weak reduction barbed congruence*, noted $\approxeq$, is the largest symmetric relation R such that for all $(N, M) \in R$:

For all barbs a, if $N\downarrow_a$ then $M\Downarrow_a$	(Barb closed)
If $N \rightarrow N'$ then there is M' s.t. $M \Rightarrow M'$ and $(N', M') \in R$	(Reduction closed)
For all contexts $C\,[\bullet]$, $(C\,[N]\,, C\,[M]) \in R$	(Context closed)

We establish some standard properties of weak reduction barbed congruence, such as $\approxeq$ is a congruence. We relate $\approxeq$ to the strong reduction barbed congruence.

Proposition 3.2 *We have* $\simeq \subset \approxeq$.

Proof. The proof of $\subseteq$ follows standard lines. To prove that $\simeq$ is strictly included in $\approxeq$ we may show that [**go**.**nil**] $\approxeq$ [**nil**] but [**go**.**nil**] $\not\simeq$ [**nil**]. □

Note that from Proposition 3.2 and Proposition 1.5 we immediately conclude $\equiv \subseteq \approxeq$. From the congruence property we obtain that reduction barbed congruence is closed under composition, which in particular for site composition gives us:

Lemma 3.3 *Let* P^i *and* Q^i $(i \in J)$ *be collections of processes. If for all* $i \in J$ *we have* $[P^i] \approxeq [Q^i]$, *then also* $\prod_{j\in J}[P^j] \approxeq \prod_{j\in J}[Q^j]$.

As for the strong case, weak reduction barbed congruence is already able to distinguish systems based on aspects of their structure, for instance, weak reduction barbed congruent networks always have the same number of sites. Also, as stated in Lemma 3.4, weak reduction barbed congruent networks weakly reduce to a one-one and onto correspondence of weakly reduction barbed congruent sites.

Lemma 3.4 *Let* M, N *be networks such that* $N \triangleq \prod_{j\in J}[P^j]$, *where* P^j $(j \in J)$ *is a collection of processes, and* $N \approxeq M$. *Then there is a collection of processes* Q^j $(j \in J)$ *such that* $M \Rightarrow \prod_{j\in J}[Q^j]$ *and for all* $j \in J$ *we have* $[P^j] \approxeq [Q^j]$.

Proof. (Sketch, full proof in [7]) The general idea is similar to that in the proof of Lemma 1.7. However, since now we may only weakly observe a barb, a different trick must be used to make sure that the migration of all the mark-placing processes has already occurred. We thus exploit the failure behavior of the context at a chosen point, avoiding in this way any chance for the migratory processes to postpone their choice of target, thus ensuring an unique correspondence. □

3.1 Weak Bisimilarity

We now propose a coinductive characterization of weak reduction barbed congruence. Weak commitment $\stackrel{\lambda}{\Longrightarrow}$ is the transition relation such that $N \stackrel{\lambda}{\Longrightarrow} N'$ when $N \stackrel{\tau}{\longrightarrow}^* M' \stackrel{\lambda}{\longrightarrow} M'' \stackrel{\tau}{\longrightarrow}^* N'$ and $\lambda \neq \tau$, and $N \stackrel{\tau}{\Longrightarrow} N'$ when $N \stackrel{\tau}{\longrightarrow}^* N'$. Given this we define weak bisimulations by adapting the labeled transition and separation clauses to the weak case.

Definition 3.5 [Weak Bisimulation] A binary relation $B \subseteq \mathcal{N} \times \mathcal{N}$ is a *weak bisimulation* if and only if it is symmetric and whenever $(N, M) \in B$ then

$$N \equiv N' \mid N'' \Rightarrow \exists M', M'' \,.\, M \Rightarrow M' \mid M'' \wedge (N', M') \in B \wedge (N'', M'') \in B$$

$$N \equiv \mathbf{0} \Rightarrow M \equiv \mathbf{0}$$

$$N \stackrel{\lambda}{\longrightarrow} N' \Rightarrow \exists M' \,.\, M \stackrel{\lambda}{\Longrightarrow} M' \wedge (N', M') \in B$$

We can prove that weak bisimulations enjoy usual properties, such as being equivalence relations, and closure under union. We thus define:

Definition 3.6 [Weak bisimilarity] *Weak bisimilarity*, noted $\approx$, is the largest weak bisimulation.

3.2 Full Abstraction

In this section, we prove that weak bisimilarity characterizes weak reduction barbed congruence in a fully abstract way, proof of which builds on the following results.

Lemma 3.7 *Let M be a network and P^j $(j \in J)$ a collection of processes such that $\prod_{j\in J}[P^j] \approx M$. Then there is a collection of processes Q^j $(j \in J)$ such that $M \Rightarrow \prod_{j\in J}[Q^j]$ and for all $j \in J$, $[P^j] \approx [Q^j]$.*

Proof. By induction on the size of J, using the separation and emptiness clauses.□

Lemma 3.8 is the cornerstone for proving full abstraction (Theorem 3.9). As for the strong case we must ensure compositionality at the process level due to process mobile capability, as process migration to sites results in inner site composition.

Lemma 3.8 *Let J be a finite set and I_j, for all $j \in J$, be a finite set. Let P_i^j and Q_i^j be processes such that for all $j \in J$ and $i \in I_j$ we have $\left[P_i^j\right] \approx \left[Q_i^j\right]$. Then*

$$\prod_{j\in J}\left[\prod_{i\in I_j} P_i^j\right] \approx \prod_{j\in J}\left[\prod_{i\in I_j} Q_i^j\right]$$

Proof. By coinduction on the definition of strong bisimulation. The proof follows the lines given for Lemma 2.5, with several adaptations needed for the weak case. Interesting to notice, in the strong case a migration of the inaction process could be mimicked by an internal computation, while here it can be mimicked by the empty sequence of internal actions (we no longer distinguish [**go**.**nil**] from [**nil**]). □

By Lemma 3.7 and Lemma 3.8 we prove that weak bisimilarity is a congruence, after which proof that $\approx \subseteq \cong$ follows in standard lines. To prove $\cong \subseteq \approx$ the

difficulty lies in the spatial clauses, given by Lemma 3.3 and Lemma 3.4. Thus:

Theorem 3.9 (Full abstraction) *We have* $\approx = \cong$.

3.3 Logical Characterization of Weak Bisimilarity

We characterize weak bisimilarity (and thus weak reduction barbed congruence) using the spatial logic $\mathcal{L}_w$.

Definition 3.10 [Spatial Logic $\mathcal{L}_w$] Formulas are defined by the following syntax:

$$\text{(Formulas) } A, B, C ::= \mathbf{T} \;\Big|\; \neg A \;\Big|\; A \wedge B \;\Big|\; \mathbf{0} \;\Big|\; A \Uparrow B \;\Big|\; \langle\!\langle \lambda \rangle\!\rangle A$$

The logic $\mathcal{L}_w$ is obtained from $\mathcal{L}_s$ by adapting the composition operator, now noted $A \Uparrow B$, and the action modality, now noted $\langle\!\langle \lambda \rangle\!\rangle A$, to the weak case as defined in Definition 3.11. We leave the void operator with it's standard interpretation (notice that $N \Rightarrow \mathbf{0}$ is a trivial condition, due to the failure behavior).

Definition 3.11 [Semantics of $A \Uparrow B$ and of $\langle\!\langle \lambda \rangle\!\rangle A$]

$$[\![A \Uparrow B]\!] \triangleq \{N \mid \exists N', N'' \,.\, N \Rightarrow N' \mid N'' \wedge N' \in [\![A]\!] \wedge N'' \in [\![B]\!]\}$$
$$[\![\langle\!\langle \lambda \rangle\!\rangle A]\!] \triangleq \{N \mid \exists N' \,.\, N \stackrel{\lambda}{\Longrightarrow} N' \wedge N' \in [\![A]\!]\}$$

We prove logical characterization of $\approx$, following the lines of Theorem 2.9.

Theorem 3.12 (Logical Characterization of $\approx$) *We have* $\approx = =_{\mathcal{L}_w}$.

As a corollary of Theorem 3.12 we conclude that the separation power of $\mathcal{L}_w$ precisely coincides with weak reduction barbed congruence, even if it includes the spatial operators composition and void. At this point, we may ask, as at the end of Section 2.2, whether the spatial operators are essential to the characterization. We may verify that $\mathbf{T}$ can be expressed as $\langle\!\langle \tau \rangle\!\rangle \mathbf{0}$, and $\langle\!\langle \tau \rangle\!\rangle A$ as $A \Uparrow \mathbf{0}$. Thus let $\mathcal{L}_w^{min}$ be the $(\mathbf{T}, \langle\!\langle \tau \rangle\!\rangle A)$-free fragment of $\mathcal{L}_w$. We may show that $\mathcal{L}_w^{min}$ is as expressive as $\mathcal{L}_w$, and moreover that all of its connectives are essential for its expressiveness.

Theorem 3.13 (Minimality) *The logic $\mathcal{L}_w^{min}$ is minimal. Moreover, the spatial operators are essential to characterize weak reduction barbed congruence.*

Proof. (Sketch, full proof in [7]) We show that any logic obtained from $\mathcal{L}_w^{min}$ by removing each connective is strictly less expressive.

- $(\neg A)$ The $\neg$-free fragment does not distinguish $[\mathbf{nil}] \mid [\mathbf{nil}]$ from $[\mathbf{nil}]$.
- $(A \wedge B)$ In the $\wedge$-free fragment we can no longer express property 1.
- $(\mathbf{0})$ The $\mathbf{0}$-free fragment does not tell $\mathbf{0}$ and $[\mathbf{nil}]$ apart.
- $(A \Uparrow B)$ The $\Uparrow$-free fragment does not separate $[\mathbf{nil}] \mid [\mathbf{nil}]$ from $[\mathbf{nil}]$.
- $(\langle\!\langle \alpha \rangle\!\rangle A, \alpha = \bar{a}, a)$ The $\langle\!\langle \alpha \rangle\!\rangle$-free fragment does not tell $[\alpha.\mathbf{nil}]$ and $[\mathbf{nil}]$ apart.
- $(\langle\!\langle [a] \rangle\!\rangle A)$ The $\langle\!\langle [a] \rangle\!\rangle$-free fragment does not distinguish $[\mathbf{go}.b.\mathbf{nil}]$ from $[\mathbf{nil}]$.

□

4 Concluding Remarks

We have studied observational equivalences in a distributed computation model, having obtained spatial logic characterizations of observational congruence in both the strong and weak cases. Taking as reference semantics for observational congruence the standard reduction barbed congruence, we have derived equivalent characterizations of observational congruences in terms of co-inductively defined bisimilarities. The logics considered are natural extensions of HML with spatial operators, interpreted in the standard way. We have thus shown, in a precise sense, that spatial logics, in particular the structural operators they offer, are not necessarily intensional, and may offer adequate expressive power for logically characterizing distributed behavior. We have also concluded, in the case of the specific process model here considered, that the composition operator $A \mid B$ is essential to capture (extensional) observational equivalence. Intuitively, such structural observations do not violate extensionality because distributed process behavior already has a related observational power, due to migration behavior and failures.

Observational equivalences of distributed systems have been studied extensively in the context of CCS-like models; a comprehensive survey may be found in [10]. However, it seems that logical characterizations have not been much discussed, and the distributed process equivalences proposed were technically defined by means of location or history-sensitive transition systems, where the use of location names plays a key role, both in the dynamic and static cases. Here, we build on a more abstract notion of spatial observation, avoiding the use of location names, and consider a calculus with anonymous sites and migration primitives, in the spirit of more recent proposals of calculi for distribution and mobility [8,18].

Our adoption of the simplest fail-stop failure model was motivated by the belief that it already captures the key consequences of failure, cf., the folklore slogan that in a distributed system one cannot distinguish a failed system from a system that will respond (much) later. The fail-stop model has been frequently adopted in formalizations of failure since [1], even if recent related works prefer to trigger failure by means of an explicit "kill" primitive [12]. Failures play an essential role in our results. However, it is conceivable that other notions of failure, and a different set of spatial behaviors and spatial observations, may lead to results comparable to the ones reported in this paper.

It is interesting to compare our results with those of [14], where an extensional spatial logic (for the π-calculus) is considered. In that work, extensionality is obtained by removing the composition and void operators, while retaining the guarantee, whereas here we obtain extensionality by retaining the composition and void operators, while doing without the guarantee. We believe that the guarantee could be added to our developments, without breaking the results. Then, it would be instructive to see how to capture indirectly the action modalities, as in [15]. It would be certainly important to assess how to extend the general approach presented here to richer models, with name restriction, name passing, and full computational power.

References

[1] R. M. Amadio and S. Prasad. Localities and Failures (Extended Abstract). In P. S. Thiagarajan, editor, *Foundations of Software Technology and Theoretical Computer Science*, volume 880 of *Lecture Notes in Computer Science*, pages 205–216. Springer-Verlag, 1994.

[2] L. Caires. Behavioral and spatial properties in a logic for the pi-calculus. In Igor Walukiwicz, editor, *Proc. of Foundations of Software Science and Computation Structures'2004*, Lecture Notes in Computer Science. Springer Verlag, 2004.

[3] L. Caires. Proof Techniques for Distributed Resources and Behaviors using Spatial Logics. In *(discussion at) Symposium on Trustworthy Global Computing*, 2005.

[4] L. Caires and L. Cardelli. A Spatial Logic for Concurrency (Part I). *Information and Computation*, 186(2):194–235, 2003.

[5] L. Caires and E. Lozes. Elimination of Quantifiers and Undecidability in Spatial Logics for Concurrency. *Theoretical Computer Science*, 10(2), 2006.

[6] L. Caires and L. Monteiro. Verifiable and Executable Specifications of Concurrent Objects in $\mathcal{L}_\pi$. In C. Hankin, editor, *7th European Symp. on Programming (ESOP 1998)*, number 1381 in Lecture Notes in Computer Science, pages 42–56. Springer-Verlag, 1998.

[7] L. Caires and H. T. Vieira. Extensionality of Spatial Observations in Distributed Systems (Draft). Technical Report TR-DI/FCT/UNL-1/2006, DI/FCT Universidade Nova de Lisboa, 2006. http://ctp.di.fct.unl.pt/~htv/pub/extspatial.pdf.

[8] L. Cardelli and A. D. Gordon. Mobile ambients. In M. Nivat, editor, *First International Conference on Foundations of Software Science and Computation Structures (FoSSaCS '98)*, volume 1378 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.

[9] L. Cardelli and A. D. Gordon. Anytime, Anywhere. Modal Logics for Mobile Ambients. In *27th ACM Symp. on Principles of Programming Languages*, pages 365–377. ACM, 2000.

[10] I. Castellani. Process algebras with localities. In J. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*, pages 945–1045. North-Holland, 2001.

[11] G. Conforti, D. Macedonio, and V. Sassone. Spatial Logics for Bigraphs. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 766–778. Springer-Verlag, 2005.

[12] A. Francalanza and M. Hennessy. A Theory of System Behaviour in the Presence of Node and Link Failures. In Martín Abadi and Luca de Alfaro, editors, *CONCUR*, volume 3653 of *Lecture Notes in Computer Science*, pages 368–382. Springer, 2005.

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

[14] D. Hirschkoff. An Extensional Spatial Logic for Mobile Processes. In P. Gardner and N. Yoshida, editors, *CONCUR 2004 15th International Conference*, volume 3170 of *Lecture Notes in Computer Science*, pages 325–339. Springer-Verlag, 2004.

[15] D. Hirschkoff, É. Lozes, and D. Sangiorgi. Minimality Results for the Spatial Logics. In P. K. Pandya and J. Radhakrishnan, editors, *Foundations of Software Technology and Theoretical Computer Science*, volume 2914 of *Lecture Notes in Computer Science*, pages 252–264. Springer-Verlag, 2003.

[16] K. Honda and N. Yoshida. On reduction-based process semantics. *Theoretical Computer Science*, 151(2):437–486, November 1995.

[17] R. Milner and D. Sangiorgi. Barbed bisimulation. In Werner Kuich, editor, *Automata, Languages and Programming, 19th International Colloquium*, volume 623 of *Lecture Notes in Computer Science*, pages 685–695, Vienna, Austria, 13–17 July 1992. Springer-Verlag.

[18] J. Riely and M. Hennessy. Distributed processes and location failures. *Theor. Comput. Sci.*, 266(1-2):693–735, 2001.

[19] D. Sangiorgi. Extensionality and Intensionality of the Ambient Logics. In *28th Annual Symposium on Principles of Programming Languages*, pages 4–13. ACM, 2001.

[20] E. Tuosto and H. T. Vieira. An observational model for spatial logics. *Electronic Notes in Theoretical Computer Science*, 142:229–254, 2006.