
UNIFYING THE LINEAR TIME-BRANCHING TIME SPECTRUM OF STRONG PROCESS SEMANTICS

DAVID DE FRUTOS ESCRIG^a, CARLOS GREGORIO RODRÍGUEZ^b, MIGUEL PALOMINO^c,
AND DAVID ROMERO HERNÁNDEZ^d

^{a,b,c,d} Departamento de Sistemas Informáticos y Computación, Universidad Complutense de Madrid
e-mail address: {defrutos, cgr, miguelpt}@sip.ucm.es, dromeroh@pdi.ucm.es

ABSTRACT. Van Glabbeek’s linear time-branching time spectrum is one of the most relevant work on comparative study on process semantics, in which semantics are partially ordered by their discrimination power. In this paper we bring forward a refinement of this classification and show how the process semantics can be dealt with in a uniform way: based on the very natural concept of constrained simulation we show how we can classify the spectrum in layers; for the families lying in the same layer we show how to obtain in a generic way equational, observational, logical and operational characterizations; relations among layers are also very natural and differences just stem from the constraint imposed on the simulations that rule the layers. Our methodology also shows how to achieve a uniform treatment of semantic preorders and equivalences.

2012 ACM CCS: [Theory of computation]: Formal languages and automata theory—Formalisms—Algebraic language theory; Formal languages and automata theory—Semantics and reasoning—Program semantics; Logic; Models of computation—Concurrency.

Key words and phrases: process semantics, linear time-branching time spectrum, algebraic languages, simulation semantics, linear semantics, constrained simulation, axiomatizations, unification.

^{a,c,d} David de Frutos Escrig, Miguel Palomino and David Romero Hernández were partially supported by the Spanish MEC project DESAFIOS10 TIN2009-14599-C03-01 and the project PROMETIDOS S2009/TIC-1465.

^b Carlos Gregorio Rodríguez was partially supported by the Spanish MEC project ESTuDIo TIN2012-36812-C02-01.



CONTENTS

1. Introduction	2
1.1. Overview of results	4
1.2. Some related work	7
1.3. Paper structure	8
2. Preliminaries	9
3. Equational semantics	12
3.1. A new axiomatization of the most popular semantics	12
3.2. The coarsest semantics in the spectrum	18
4. Observational semantics	21
4.1. Branching general observations	22
4.2. Linear observations and linear time semantics	26
4.3. Deterministic branching observations	31
4.4. Back to branching observations	34
5. Relating the observational and equational frameworks	36
5.1. Semantics coarser than ready simulation	36
5.2. The semantics that are not coarser than ready simulation	39
6. Logical characterization of semantics	41
6.1. A new logical characterization of the most popular semantics	44
6.2. Our new unified logical characterizations of the semantics	49
7. Relating the unified logics and the unified observational model	55
8. On the real diamond structure	59
9. Operational semantics	64
9.1. Local simulations up-to	64
9.2. Operational rules for the linear semantics of processes	66
9.3. Characterizing the semantics corresponding to other constraints	68
9.4. Application: trace deterministic normal forms	69
10. Conclusions and some future work	69
References	70

1. INTRODUCTION

Since the foundational work by Robin Milner [41, 42] and Tony Hoare [32] on process semantics, there has been a multitude of proposals to endow processes with meaning and to define equivalence and preorder relations over them. Among the most relevant work are those of Matthew Hennessy [30], who introduced the testing methodology defining process semantics from test cases, and those of Jan Bergstra and Jan Willen Klop [11], later continued by Jos Baeten and Peter Weijland [10], which were based on an axiomatic approach.

These proposals define algebraic languages for the specification of processes, diverging in subtle details concerning the treatment of non-determinism and parallelism. These aspects are captured by means of certain operators which may (strongly) vary in each particular language.

Focusing on equivalences, it is interesting to note how the pioneering work in this area already established two fundamental notions, bisimulation and traces/failures, that constitute an upper and a lower bound on the natural framework in which other process equivalences can be studied. Hoare—with his characteristic clarity—summarizes the situation in the following paragraph.

CCS makes many distinctions between processes which would be regarded as identical in this book. The reason for this is that CCS is intended to serve as a framework for a family of models, each of which may make more identifications than CCS but cannot make less. To avoid restricting the range of models, CCS makes only those identifications which seem absolutely essential. In the mathematical model of this book [CSP] we have pursued exactly the opposite goal—we have made as many identifications as possible, preserving only the most essential distinctions. [32]

In between these two fundamental notions of equivalence—bisimulation and traces—the last two decades of the 20th century witnessed the surge of a large variety of new equivalences associated to new calculi and process algebras, whose aim was to explore the different needs for expressivity and distinction capabilities in many applications.

The most important taxonomic work on process semantics was carried out by Rob van Glabbeek as part of his doctoral dissertation [54]. In two papers, titled *Linear time-branching time spectrum* [55, 56], he collected the most important of these equivalences establishing, among other results that we will comment on, a classification based on their capability to distinguish processes. The first of the papers concentrate on *strong* semantics, in the sense that they consider each action processes perform as being observable by their environment. The second paper consider the inclusion in the language of a new and invisible action τ ; process semantics considering this internal action are usually called *weak* semantics. Figure 1 shows a slightly expanded version of the spectrum proposed by van Glabbeek for the case of strong semantics in [55]. These strong semantics, that do not consider at all the special role of internal actions τ , are the only ones that we consider in this paper.

This array of semantics is supported by many authors who claim that there is no single “good” definition. Process theory can be applied in a wide spectrum of contexts and situations and the concrete uses will have a decisive influence in the election of what a suitable semantics should be.

The choice of a suitable semantics may depend on the tools an environment has, to distinguish between certain processes. It is conceivable that a concurrency theory is equipped with different semantics, and has the capacity to express equality on different levels. [57]

The possibility to define several and varied semantics can then be considered to be an advantage of the theory, since it allows for the necessary flexibility to reflect different notions of processes and equivalence and preorder relations over them.

Nevertheless, this multiplicity has gone hand in hand in the literature with an individual study of each of the semantics that somehow makes the whole theory less appealing because such a cornucopia can become a handicap both for its study and its practical application. For instance, although most of the semantic notions defined for processes simultaneously induce both a (pre)order and an equivalence,¹ the literature has frequently overlooked the

¹A remarkable exception, however, is the bisimulation notion, for which no non-trivial order relation is known.

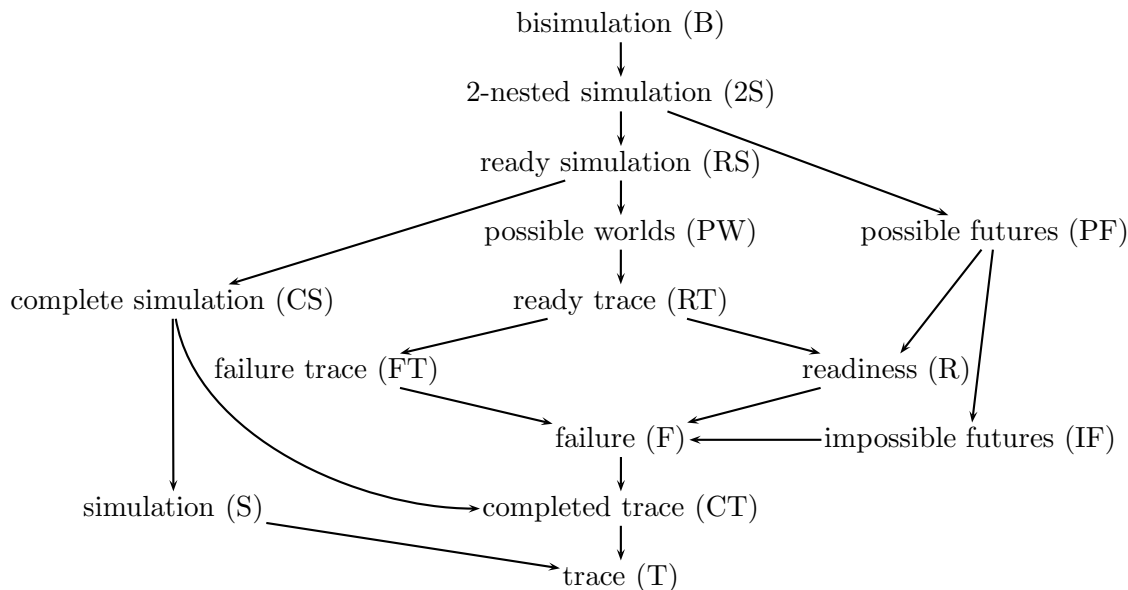


Figure 1: Linear time-branching time spectrum.

fact that these two notions are mutually intertwined, as we will show later. Likewise, the study of the concrete models has been usually undertaken paying little attention to the other semantics or to the relations among them, even though it is well-known that there exist “families” of semantics—such as the linear semantics—which are undoubtedly related.

A unified study of semantics has both methodological and practical implications that have been explored along the last years by the authors of this work, for example in [22, 24, 25, 20, 21], and also in work by important researchers in the area [5, 4, 16, 40]. This research shows that a unified view of process semantics is indeed possible.

This is precisely the main goal we set to reach with our work: we aim to study process semantics in a generic way, making the equivalence and (pre)order relations our object of study in order to find patterns, to identify families, to search for properties among these relations, so that we obtain generic results that need not be proved again and again for each of the semantics.

We aim, in a nutshell, at a unifying view of process semantics that can be used to understand them both jointly and individually and that allows to continue with their theoretical study in a more focused manner, helping to identify those properties a semantics should have for a particular application.

1.1. Overview of results. This paper contains a consolidated and extended presentation of the unification of observational, equational and the logic process semantics published in [20, 19, 47] for strong behavioral semantics.² We take advantage of the joint and larger presentation of the subject to tighten the connections between the different views. Besides, we make the paper mostly self-contained providing proofs for all the results; we also complete the study with new results not included in [20, 19, 47]. We have also completed the

²We comment in Section 10 the work of some of the authors on unification of weak semantics.

revision of the unification of strong process semantics with a section devoted to the unified presentation of the operational semantics.

Next we describe the main results we have obtained. They can be used as a roadmap for reading the paper and to understand the technical details in the following sections.

- One of the most generic results we have proved is the existence of two essential families of semantics: branching semantics and linear semantics. Certainly, this was already hinted by van Glabbeek when he named its spectrum of semantics “linear time-branching time”.

Our results show that the most representative branching semantics have characterizations as simulation semantics. Moreover, every simulation has a natural family of coarser linear semantics associated to it that inherit some of its properties.

In Figure 11 (page 34), branching semantics are located to the left and each of them defines a layer of “induced” linear semantics, to their right. For example, ready simulation is the branching semantics from which the classic diamond of linear semantics composed of failures, readiness, failure trace, and ready trace semantics is generated. These semantics, as we will later see in detail, inherit some of their axiomatic characterizations directly from the ready simulation. In addition, the same layer also contains the possible worlds semantics, which is a deterministic branching semantics.

Even though the axiomatic characterizations contained in Section 3 already show this dependency between linear and branching semantics—see Figure 5 (page 17)—it is in Section 4 where, using techniques from denotational semantics, the relations between the original branching semantics and the induced linear semantics can be fully appreciated. The relationships among the different linear semantics in the same branching layer are also completely specified in that section.

- Equational characterizations reveal in a very concise manner the basic properties of the different semantics. As a result of our research, we have been able to derive a generic axiomatic characterization of the semantics in the spectrum which shows clearly the relationships among them: the uniformity in the definitions of the branching semantics and the different families of linear semantics becomes apparent, as well as the tight relation between each branching semantics and its associated linear semantics. In Section 3 we present all the details related to these axiomatic characterizations.
- Another result that we consider important is that it is indeed possible to establish a clear relationship between the preorder and the equivalence associated to a given semantics. From an axiomatic point of view, in Section 3 we show how these characterizations are closely related. In fact, there are algorithms that allow to easily obtain the axioms for the equivalence from those of the preorder [4, 21], and also the other way around, the axioms for the preorder from those of the equivalence [23, 24].
- We also offer a unifying view of the process semantics based on observational (denotational) semantics, according to which we have classified the process semantics in four categories:
 - bisimulation semantics, which is the finest semantics in the spectrum and the only one that cannot be defined by means of a non-trivial preorder;
 - the simulation semantics (simulation, complete simulation, ready simulation, nested simulation, ...) which are characterized by means of branching observations, that is, labeled trees;
 - the linear semantics (traces, failures, readiness, ...), characterized by linear observations, a degenerated case of branching observations;

- the deterministic branching semantics corresponding to an intermediate class between branching and linear, where observations are deterministic trees. Possible worlds semantics is the only semantics in the original van Glabbeek’s spectrum in this class.

Besides their linear or branching nature, semantics are characterized by a local observation function that generates the local observations at the states. For the linear case there is also the possibility of observing this local information in a partial way and this is how for each local observer, in principle, up to four different semantics can be obtained. In particular, this gives rise to the classic diamond below the ready simulation semantics formed by failures, failure trace, readiness, and ready trace semantics.

The uniform presentation of the process semantics that we offer in Section 4 clarifies the relationships and hierarchies among all the semantics; moreover, it will make possible the development of generic proofs of their common properties.

- We also present a unified view of the logical semantics. Again, the bisimulation semantics, which is characterized by the Hennessy-Milner logic HML [31], is our starting point, and then we aim for the sublogics that characterize each of the semantics in the spectrum. Guided by our main unification goal, we have not tried to obtain the smallest possible sets of formulas, but have veered for the largest sublogics that characterize each of the semantics.

Hence, the finest semantics are characterized by the largest sublogics and in fact we obtained a uniform characterization that informs us about the hierarchy of semantics, by proving that a semantics S_1 is finer than another S_2 if and only if the corresponding logics satisfy $\mathcal{L}_{S_2} \subseteq \mathcal{L}_{S_1}$. Moreover, the classification into branching and linear time semantics is also reflected in the structural definition of each logic. In particular, the branching semantics are characterized by the free use of negation over the formulas that define the corresponding constraint, while the linear semantics at each layer of the spectrum introduce ever more limitations in the subformulas.

- Finally, we also discuss an operational-like presentation of the semantics in the spectrum; more precisely, we consider an evaluation semantics to derive the appropriate data which characterize them. Those data are quite similar to the ones employed for our observational semantics so that it is not them, but the way in which they are derived, that enhances our understanding of the features of each of the semantics and the relationships between them. These presentations somehow generalize the work by Cleaveland and Hennessy [17] on the characterization of the Testing semantics by means of bisimulation. There is also a clear connection with the work by two of the authors of this paper on (Bi)simulation up-to [25].
- A concomitant, but still important result of our work, is of methodological nature: the semantics are amenable to a working methodology that allows for general results that can be applied to families of semantics as well as to yet to be defined semantics. The requirements we impose on these new semantics are relatively mild. An example of this is shown seen in Section 8, where some new process semantics—indeed two new families—smoothly integrate into our general theory. In fact, it was nice to discover that one of them is just the revivals semantics, which has been recently developed by Bill Roscoe [48].
- Each of the characterization frameworks—equational, observational, logical or operational—sheds light on the spectrum in different and complementary ways. This has provided us with different ways to study all the previously known semantics and the relationships between them. That complementary nature also sprang up along our unification work when we discovered one by one all the factors that contribute to the structure of the

extended spectrum. In particular, when considering simple and natural combinations of axioms we found out the new meet semantics in Section 8, while their dual join semantics was discovered in a natural way when considering the observational characterizations. Finally, the semantics of minimal readies (in Section 6.2.2) appeared when investigating the logical framework. While not too important on its own, our unification work has also revealed a mistake in the classic logical characterization of one of the semantics in the original spectrum (see Section 6.2.3): it was the general and systematic approach that guides our uniform characterization that allowed it.

1.2. Some related work. Naturally, the goal of defining a global or general theory of process semantics has been around for a long time and several relevant authors in the field have already paved the way that we now tread.

Despite the methodological differences between Milner’s work, based on bisimulation, and Hoare’s, on denotational semantics, both of them had in common the search for characterizations—logical, axiomatic, observational—that could shed light from different angles on the world of process semantics.

Hennessy introduced the testing methodology to endow processes with semantics, making the notion of equivalence to spring from the application of the interaction principles for processes expressed within the model. Perhaps one of the most important contributions of his work was what he called the “trinity”: processes can be seen as syntactic terms in an algebra, as operational descriptions in labeled transition systems, or as denotational objects in a mathematical model. With our work we have somehow extended this trinity in a generic manner to all the semantics in the extended spectrum.

Van Glabbeek’s work, *the linear time-branching time spectrum* aimed at the comparison of most known semantics—at the time he developed his seminal work—by presenting them within common frameworks that would allow a comparative study of their properties. Besides providing uniform definitions over transition systems, van Glabbeek also proposed to characterize the semantics in terms of logical formulas. The set of modal formulas whose satisfaction equivalence identifies the same processes as the corresponding semantics is defined. Because of the compositional definition of the corresponding sets of formulas, this characterization can be considered to be denotational semantics.

Another characterization provided by van Glabbeek is the axiomatic one, for which he defines the BCCSP language that is used in this work (see Definition 2.1). Twelve of the semantics in the spectrum are characterized by means of sets of axioms over syntactic terms for this language. For most of them—except for bisimulation, that has no associated order—their characterizations are actually twofold: on the one hand, the natural order relation that defines each semantics—Table 1—and, on the other hand, the induced equivalence—Table 2. Many of these characterizations were previously known but, again, their uniform presentation is one of its main merits.

A deep study—individual as well as comparative—of these axiomatizations and the quest for answers to the new questions that arise from this study has been one of the leading forces behind our research. Actually, some of our most relevant results can be combined into a new way of presenting the spectrum—Figure 11 (page 34)—that allows for a better comprehension of the semantics since it clarifies their relative positions within it and shows the existence of “gaps” that correspond to new semantics whose addition to the graph reflects a desirable regularity that makes it clearer. Hoare’s work on the unification of the

study of process algebras [33] was also an important influence. Specially, the relationship between similarity and trace refinement, which we have generalized by establishing the connection between branching time and linear time semantics, and the connection between the denotational, the algebraic, and the operational styles proposed by him and He Jifeng in [34].

As already mentioned, Roscoe has contributed in an independent research effort in parallel with ours to the study of new process semantics by proposing his stable revivals model [48]. He relates his new semantics with other well known linear time semantics and the rediscovery of that semantics in our expanded spectrum gave us the opportunity to present those relationships with a unified and generic light.

There is other relevant work in the area of process theory that has inspired us. The number of contributions is too large to cite all of them here. Anyone interested on finding a more exhaustive list of relevant references may collect them, for instance, from [28, 7, 1, 57, 50]. There the historic evolution of the area and many of the most important contributions to it are reviewed. To them we can add four recent books on process algebras and related subjects [8, 49, 6, 51], presenting different points of view and some of the semantics studied in this paper. Finally, in our Conclusions, we will discuss a bit the work on the generic study and classification of the weak semantics.

1.3. Paper structure. We have structured this paper as follows. Section 2 introduces all the basic definitions and notation to properly follow the developments in the following sections.

In Section 3 we propose alternative characterizations for the axiomatizations of the semantics in the spectrum, both for orders and equivalences. All these axiomatizations are based on just two parametrical axiom skeletons that clearly highlight the relations among the different semantics.

Section 4 presents a unified observational characterization for process semantics. One of the key ideas is that constrained simulations are uniformly characterized by a branching observation plus a local observation function. From the observations of a given constrained simulation, the linear semantics in its layer are uniformly derived.

In Section 5 we prove that the equations we presented in Section 3 are deduced from the observations defined in Section 4 in a general way, without using at all the already known axiomatizations for the semantics.

Section 6 follows the trails of Sections 3 and 4 by introducing a unified logical characterization of process semantics.

In Section 7 we prove that the observational characterizations developed in Section 4 allow for generic proofs for the logical characterizations presented in Section 6. Therefore the “trinity” of equations, observations, and logical formulas is established in a generic way for large families of process semantics.

Section 8 is a practical proof of the applicability of our unification proposal. Some new process semantics, that were not listed in the original linear time-branching time spectrum, are easily accommodated in our framework thus getting the corresponding semantic characterizations that we have presented in previous sections.

In Section 9 we conclude the unified presentation of the semantics in the spectrum by developing an operational characterization which mainly produces the information provided by the observational semantics, but inferred in an operational way, using a (unified) set of SOS-like rules.

Finally, in Section 10 we offer some conclusions and lines for future work.

Acknowledgments. We gratefully acknowledge three anonymous referees for their very thoughtful and detailed comments on a previous version of this work, that have greatly helped us to improve the presentation of this material.

2. PRELIMINARIES

Although the main results in this paper are also valid for infinite processes—as we showed in [22, 25]—in order to simplify the presentation of the concepts, we will mainly consider finite processes generated by the basic process algebra BCCSP which contains only the basic process algebraic operators from CCS [42], and CSP [32], but is sufficiently powerful to express all finite synchronization trees [41]. This language has repeatedly been used in unification work, e.g. [4, 58].

Definition 2.1. Given a set of actions Act , the set $BCCSP(Act)$ of processes is defined by the following BNF-grammar:

$$p ::= \mathbf{0} \mid ap \mid p + q$$

where $a \in Act$; $\mathbf{0}$ represents the process that performs no action; for every action in Act , there is a prefix operator; and $+$ is a choice operator.

The operational semantics for BCCSP terms is defined in Figure 2. As usual, we write $p \xrightarrow{a}$ if there exists a process q such that $p \xrightarrow{a} q$, and $p \xrightarrow{\alpha}$ if $\alpha = a_1 \dots a_n$ and $p \xrightarrow{a_1} p' \xrightarrow{a_2} \dots \xrightarrow{a_n} q$. The initial offer of a process is the set $I(p) = \{a \mid a \in Act \text{ and } p \xrightarrow{a}\}$. This is a simple, but quite important observation function that plays a central role in the definition of the most popular semantics in the linear time-branching time (ltbt) spectrum. We will also denote by I the relation expressing the fact that two processes have the same initial offers: $pIq \Leftrightarrow I(p) = I(q)$.

One way to capture semantics is by means of the equivalence relation induced by it: given a formal semantics $\llbracket \cdot \rrbracket_Z$, we say that processes p and q are equivalent iff they have the same semantics, that is, $p \equiv_Z q \Leftrightarrow \llbracket p \rrbracket_Z = \llbracket q \rrbracket_Z$. These semantics can be defined by means of adequate observational scenarios, or by logical characterizations that induce natural preorders \sqsubseteq_Z whose kernels are the semantic equivalences. We refer to [58] for the original definition and usual notation for all the semantics in the ltbt spectrum that will be discussed throughout the paper.

To properly express equations or inequations within the process language, we introduce variables from any adequate set V , and consider the extended set $BCCCSP(Act, V)$ of terms including variables in V .

Some of the semantics in the spectrum are concrete examples of the general notion of constrained simulation semantics that can be defined in a parameterized way.

Definition 2.2. Given a relation N over BCCSP processes, a relation S_N is an N -constrained simulation if pS_Nq implies:

- for every $a \in Act$, if $p \xrightarrow{a} p'$ then there exists some q' such that $q \xrightarrow{a} q'$ and $p'S_Nq'$, and
- pNq .

We say that process p is N -simulated by process q , or that q N -simulates p , written $p \sqsubseteq_{NS} q$, whenever there exists an N -constrained simulation S_N such that pS_Nq .

$$ap \xrightarrow{a} p \qquad \frac{p \xrightarrow{a} p'}{p + q \xrightarrow{a} p'} \qquad \frac{q \xrightarrow{a} q'}{p + q \xrightarrow{a} q'}$$

Figure 2: Operational semantics for BCCSP terms.

$$\begin{array}{ll} (B_1) & x + y \simeq y + x \\ (B_2) & (x + y) + z \simeq x + (y + z) \end{array} \qquad \begin{array}{ll} (B_3) & x + x \simeq x \\ (B_4) & x + \mathbf{0} \simeq x \end{array}$$

Figure 3: The axiomatization for the (strong) bisimulation equivalence.

We have already studied the constrained simulation semantics in detail in [24], stressing their general properties. In particular, the following constraints were considered:

- the universal relation U relating all processes, which gives rise to the simulation semantics;
- the relation C , which holds for processes p and q when both, or none, are isomorphic to $\mathbf{0}$, and that gives rise to the complete simulation semantics;
- the relation I relating processes with the same initial offer, which is the constraint for ready simulation;
- the relation T , that holds for processes having the same set of traces and gives rise to the trace simulation semantics;
- the relation S , the inverse of the simulation equivalence relation, whose associated constrained simulation is the 2-nested simulation.

Throughout this paper there appear different order relations. We use \sqsubseteq to denote semantic preorders and, for the sake of simplicity, we use the symbol \sqsupseteq to represent the preorder relation \sqsubseteq^{-1} . With \equiv we denote the induced equivalence (that is, $\sqsubseteq \cap \sqsupseteq$). To refer to a specific preorder we shall append the initials of its name as subscripts to the symbol \sqsubseteq (\sqsubseteq_{RS} for ready simulation, \sqsubseteq_F for failures, and so on). A similar convention applies to the kernels of the preorders (\equiv_{RS} , \equiv_F , ...) and to the bisimulation equivalence \equiv_B .

An *inequation* (respectively, an *equation*) over the language BCCSP is a formula of the form $t \preceq u$ (respectively, $t \simeq u$), where $t, u \in \text{BCCSP}(\text{Act}, V)$. An *(in)equational axiom system* is a set of (in)equations over the language BCCSP. An equation $t \simeq u$ is derivable from an equational axiom system E , written $E \vdash t \simeq u$, if it can be proven from the axioms in E using the rules of equational logic (viz. reflexivity, symmetry, transitivity, substitution and closure under BCCSP contexts):

$$t \simeq t \qquad \frac{t \simeq u}{u \simeq t} \qquad \frac{t \simeq u \quad u \simeq v}{t \simeq v} \qquad \frac{t \simeq u}{\sigma(t) \simeq \sigma(u)} \qquad \frac{t \simeq u}{at \simeq au} \qquad \frac{t \simeq u \quad t' \simeq u'}{t + t' \simeq u + u'}$$

where substitutions σ are defined and applied as usual.

For the derivation of an inequation $t \preceq u$ from an inequational axiom system E , the rule for symmetry—that is, the second rule above—is omitted. We write $E \vdash t \preceq u$ if the inequation $t \preceq u$ can be derived from E .

It is well-known that, without loss of generality, one may assume that substitutions happen first in (in)equational proofs, i.e., that the fourth rule may only be used when its premise is one of the (in)equations in E . Moreover, by postulating that for each equation in E its symmetric counterpart is also present in E , one may assume that applications of symmetry happen first in equational proofs, i.e., that the second rule is never used. In the

remainder of this paper, we shall always tacitly assume that equational axiom systems are closed with respect to symmetry. Note that, with this assumption, there is no difference between the rules of inference of equational and inequational logic. In what follows, we shall consider an equation $t \simeq u$ as a shorthand for the pair of inequations $t \preceq u$ and $u \preceq t$.

An inequation $t \preceq u$ is *sound* with respect to a given preorder relation \sqsubseteq , if $t \sqsubseteq u$ holds true. An (in)equational axiom system E is sound with respect to \sqsubseteq if so is each (in)equation in E . An (in)equational axiomatization is called ground-complete if it can prove all the valid (in)equivalences relating terms with no occurrences of variables. As in [58], we abbreviate ground-completeness for completeness because this is the only kind we use along the paper.

Bisimilarity, the strongest of the semantics in the spectrum, can be axiomatized by means of the four simple axioms in Figure 3. These axioms state that the choice operator is commutative, associative and idempotent, having the empty process as identity element. These axioms also justify the use of the notation $\sum_a \sum_i ap_a^i$ for processes, where the commutativity and associativity of the choice operator is used to group together the summands whose initial action is a . We will also write $p|_a$ for the (sub)process we get by projecting all the a -summands of p ; that is, if $p = \sum_a \sum_i ap_a^i$, then $p|_a = \sum_i ap_a^i$.

Besides the semantics in the spectrum, we are interested in a general study that can be applied to any “reasonable” semantics coarser than bisimilarity. Since we will use preorders to characterize these semantics we introduce the following definitions that state the desired properties of those reasonable preorders.

Definition 2.3. A preorder relation \sqsubseteq over processes is a *behavior preorder* if

- it is weaker than bisimilarity, i.e., $p \equiv_B q \Rightarrow p \sqsubseteq q$, and
- it is a precongruence with respect to the prefix and choice operators, i.e., if $p \sqsubseteq q$ then $ap \sqsubseteq aq$ and $p + r \sqsubseteq q + r$.

If \sqsubseteq is actually an equivalence, it is said to be a *behavior equivalence*.

Another way of presenting a semantics is by means of a logical characterization. The Hennessy-Milner logic [31], characterizing the bisimulation semantics is the most popular one.

Definition 2.4 (Hennessy-Milner logic, HML). The set \mathcal{L}_{HM} of Hennessy-Milner logical formulas is defined by: if $\varphi, \varphi_i \in \mathcal{L}_{HM}$ for all $i \in I$ and $a \in Act$, then $\bigwedge_{i \in I} \varphi_i, a\varphi, \neg\varphi \in \mathcal{L}_{HM}$.

The satisfaction relation \models is defined by:

- $p \models a\varphi$ if there exists q such that $p \xrightarrow{a} q$ and $q \models \varphi$;
- $p \models \bigwedge_{i \in I} \varphi_i$ if for all $i \in I : p \models \varphi_i$.
- $p \models \neg\varphi$ if $p \not\models \varphi$.

Note that $\bigwedge_{i \in \emptyset} \varphi_i \in \mathcal{L}_{HM}$, and we have $p \models \bigwedge_{i \in \emptyset} \varphi_i$ for all p . Therefore, in the following we will consider that $\top \in \mathcal{L}_{HM}$, where \top is syntactic sugar for $\bigwedge_{i \in \emptyset} \varphi_i$. The finite version of this logic, \mathcal{L}_{HM}^f , uses binary conjunction \wedge instead of the general conjunction $\bigwedge_{i \in I}$. It is well-known that \mathcal{L}_{HM}^f characterizes the bisimulation semantics between image-finite processes, that are those that do not allow infinite branching for any action $a \in Act$ at any state. Van Glabbeek uses \mathcal{L}_B to refer to \mathcal{L}_{HM} in [58].

3. EQUATIONAL SEMANTICS

On Tables 1 and 2 appear the axiomatic characterizations for the preorders and equivalences in van Glabbeek’s spectrum [58]. For each column, the set of axioms marked with “+” are sound and complete with respect to the preorder or equivalence in the head of that column; axioms marked with “v” are valid but not needed. When studying these tables there are several questions that naturally arise: for every semantics, is there any connection between the axioms defining the preorder and those for the equivalence? Can the axiomatizations of some of these semantics be jointly tackled?

In this section we will develop new axiomatizations for all the semantics in the ltb spectrum that offer a clear answer to the previous questions: even if there was not a systematic procedure that led to produce the axiomatizations of those tables, we can obtain equivalent axiomatizations that do follow a given procedure.

These new axiomatizations are obtained after noticing that every process semantics can be understood as the product of two “design decisions”, decisions that define what we have called the “dynamic” and the “static” basis of the semantics. We will show that, besides B_1 – B_4 , we only need a generic simulation axiom (NS)—Proposition 3.1—which characterizes the family of constrained simulation semantics, to axiomatize the whole class of pure branching semantics. Moreover, to characterize the linear time semantics, we only need to add to the corresponding simulation axiom the adequate instantiation of a generic axiom (ND)—see page 14—for reducing the observability of non-determinism in processes, by means of which we introduce the additional identifications induced by each of the linear semantics.

Also the axiomatizations between orders and equivalences are closely related; in fact, in the case of the linear semantics we could just use an equivalence (ND_{\equiv}) axiom, leaving the order or equivalence aspect to be determined by the use of the order or equivalence axiom of the corresponding branching semantics, see Figure 5.

In order to justify the form of our axiomatizations without leaving the axiomatic framework, in this section we prove our results with separate and ad-hoc proofs for each semantics just comparing the new characterizations with those previously known. This allows us to quickly get the taste of the underlying relations of the process semantics. Once the unified observational characterization of semantics is presented in Section 4, we will provide generic proofs for these results in Section 5 that show the suitability of the new axiomatizations with respect to the observational characterizations of the semantics.

3.1. A new axiomatization of the most popular semantics. We start our study with a very representative and well-known group of semantics in the spectrum, each of which has been developed and used in important work in the area: ready simulation [38, 13], failures [14, 32], readiness [43], ready trace [9] and failure traces [44].

3.1.1. Semantic preorders. As already hinted above, the dynamic part of the semantics is inherited from a simulation preorder. As stated in our Introduction, bisimilarity can be axiomatized by the set of axioms B_1 – B_4 . All the other semantics in the spectrum are coarser than it, and therefore also satisfy these axioms. But due to the fact that bisimulations define equivalence relations and not just preorders, we cannot base on them the characterization of any other interesting semantics. But, plain simulations are somehow defined as half-bisimulations, and can indeed be used as support for the characterizations of

	B	RS	PW	RT	FT	R	F	CS	CT	S	T
$(x + y) + z \simeq x + (y + z)$	+	+	+	+	+	+	+	+	+	+	+
$x + y \simeq y + x$	+	+	+	+	+	+	+	+	+	+	+
$x + 0 \simeq x$	+	+	+	+	+	+	+	+	+	+	+
$x + x \simeq x$	+	+	+	+	+	+	+	+	+	+	+
$ax \preceq ax + ay$		+	+	+	+	+	+	v	v	v	v
$a(bx + by + z) \simeq a(bx + z) + a(by + z)$			+	v	v	v	v		v		v
$I(x) = I(y) \Rightarrow ax + ay \simeq a(x + y)$				+	v	v	v		v		v
$ax + ay \succeq a(x + y)$					+		v		v		v
$a(bx + u) + a(by + v) \succeq a(bx + by + u)$						+	v		v		v
$ax + a(y + z) \succeq a(x + y)$							+		v		v
$ax \preceq ax + y$								+	+	v	v
$a(bx + u) + a(cy + v) \simeq a(bx + cy + u + v)$									+		v
$x \preceq x + y$										+	+
$ax + ay \simeq a(x + y)$											+

Table 1: Axiomatization for the preorders in the linear time-branching time spectrum.

some interesting semantics, such as trace semantics. Nevertheless, plain similarity becomes too weak, and some other finer class of simulations is needed to support the characterization of the interesting semantics listed above. Next we recall the axiomatizations of plain, ready and general constrained similarity.

Proposition 3.1 ([58, 24]).

- (1) *Plain similarity can be axiomatically defined by means of the axiom (S) $x \preceq x + y$, together with the axioms B_1 – B_4 that define bisimilarity.*
- (2) *Ready similarity can be axiomatically defined by means of the conditional axiom (RS) $xIy \Rightarrow x \preceq x + y$, together with B_1 – B_4 . It can also be axiomatized by means of the axiom scheme $ax \preceq ax + ay$, where a represents any arbitrary action.*
- (3) *Whenever N is a behavior preorder, N -similarity can be axiomatically defined by means of the conditional axiom (NS) $N(x, y) \Rightarrow x \preceq x + y$, together with B_1 – B_4 .*

Let us now consider the diamond of semantics coarser than ready similarity in the ltbt spectrum. It consists of the failures, readiness, failure trace, and ready trace semantics. None of them is a simulation semantics, so their classic axiomatizations (see Table 1) contain an additional axiom:

$$\begin{aligned}
\text{Failures:} & \quad (F) & \quad a(x + y) \preceq ax + a(y + w) \\
\text{Readiness:} & \quad (R) & \quad a(bx + by + u) \preceq a(bx + u) + a(by + v) \\
\text{failure trace:} & \quad (FT) & \quad a(x + y) \preceq ax + ay \\
\text{ready trace:} & \quad (RT) & \quad I(x) = I(y) \Rightarrow ax + ay \simeq a(x + y)
\end{aligned}$$

Since we are interested in capturing the reduction of observability of non-determinism, our first candidate for a general axiom covering all cases was (FT), which captures the fact that by delaying the choices we get “smaller” processes. However, since this axiom characterizes the failure trace semantics and this is finer than failure semantics, a more general axiom is needed: axiom (F) became our next proposal because failure semantics is the coarsest of the four semantics. More precisely, we expected to achieve the axiomatization of the four semantics in the diamond by adding the adequate instance of the generic

constrained conditional axiom

$$(ND) \quad M(x, y, w) \Rightarrow a(x + y) \preceq ax + a(y + w).$$

This seemed reasonable since the other semantics in the group are finer than failures and by adding a constraint to (F) we certainly obtain a more restricted axiom that produces a finer preorder. The conjecture turned out to be correct and we found that the semantics in the diamond can be characterized by the following instances:

$$\begin{aligned} (ND^F) \quad M_F(x, y, w) &\iff \text{true} \\ (ND^R) \quad M_R(x, y, w) &\iff I(x) \supseteq I(y) \\ (ND^{FT}) \quad M_{FT}(x, y, w) &\iff I(w) \subseteq I(y) \\ (ND^{RT}) \quad M_{RT}(x, y, w) &\iff I(x) = I(y) \text{ and } I(w) \subseteq I(y) \end{aligned}$$

Since M_F is the universal relation containing all triples of processes, the corresponding instance of the conditional axiom (ND) is clearly equivalent to (F) , and thus adding it to the set $\{B_1\text{-}B_4, (RS)\}$ we obtain a ground-complete axiomatization of \sqsubseteq_F . Let us now prove that the remaining three semantics are also axiomatized by the corresponding instances of the axiom (ND) together with (RS) .

Proposition 3.2.

- (1) *The readiness preorder \sqsubseteq_R is axiomatized by $\{B_1\text{-}B_4, (RS), (ND^R)\}$.*
- (2) *The failure trace preorder \sqsubseteq_{FT} is axiomatized by $\{B_1\text{-}B_4, (RS), (ND^{FT})\}$.*
- (3) *The ready trace preorder is axiomatized by the set $\{B_1\text{-}B_4, (RS), (ND^{RT})\}$.*

Proof.

- (1) Let us show that the set $\{B_1\text{-}B_4, (RS), (ND^R)\}$ is logically equivalent to $\{B_1\text{-}B_4, (RS), (R)\}$. By taking $x = bx' + u$, $y = by'$, and $w = v$ we have that (ND^R) implies (R) . In the other direction, let x and y be arbitrary closed BCCSP terms with $I(y) \subseteq I(x)$: we will prove, by structural induction on y , that $\{B_1\text{-}B_4, (RS), (R)\} \vdash a(x + y) \preceq ax + a(y + w)$, for any term w .
 - For $y = \mathbf{0}$, we have $a(x + y) \simeq ax \preceq ax + a(y + w)$, by application of (RS) .
 - For $y = by' + y''$, it must be $x = bx' + x''$ and taking $v = y'' + w$ in (R) we obtain $a(x + y) = a(bx' + by' + x'' + y'') \preceq a(x + y'') + a(y + w)$. Then we have $I(y'') \subseteq I(x)$ and we can apply the induction hypothesis to get $\{B_1\text{-}B_4, (RS), (R)\} \vdash a(x + y) \preceq ax + a(y + w)$.
- (2) Let us show that the sets $\{B_1\text{-}B_4, (RS), (ND^{FT})\}$ and $\{B_1\text{-}B_4, (RS), (FT)\}$ are logically equivalent. The implication from left to right follows by taking $w = \mathbf{0}$. In the other direction, let w and y with $I(w) \subseteq I(y)$, so that $a(x + y) \preceq ax + ay$ using (FT) and, since $I(y) = I(y + w)$, we have $y \preceq y + w$ using (RS) : hence we conclude, $a(x + y) \preceq ax + a(y + w)$.
- (3) Let us show that the set $\{B_1\text{-}B_4, (RS), (ND^{RT})\}$ is logically equivalent to $\{B_1\text{-}B_4, (RS), (RT)\}$. We first note that $\{B_1\text{-}B_4, (RS), (RT)\}$ is equivalent to $\{B_1\text{-}B_4, (RS), (RT_{\succeq})\}$, where (RT_{\succeq}) is the axiom $M_{RT}(x, y, w) \Rightarrow ax + ay \succeq a(x + y)$. This follows from the fact that, whenever $I(x) = I(y)$, we can use (RS) to get $x \preceq x + y$ and $y \preceq x + y$, and then $ax + ay \preceq a(x + y)$. Now, the implication from left to right follows by taking $w = \mathbf{0}$. From right to left, as above, whenever $I(w) \subseteq I(y)$ we have $y \preceq y + w$ and then, if $I(x) = I(y)$ we have $a(x + y) \preceq ax + ay$, and therefore $a(x + y) \preceq ax + a(y + z)$. □

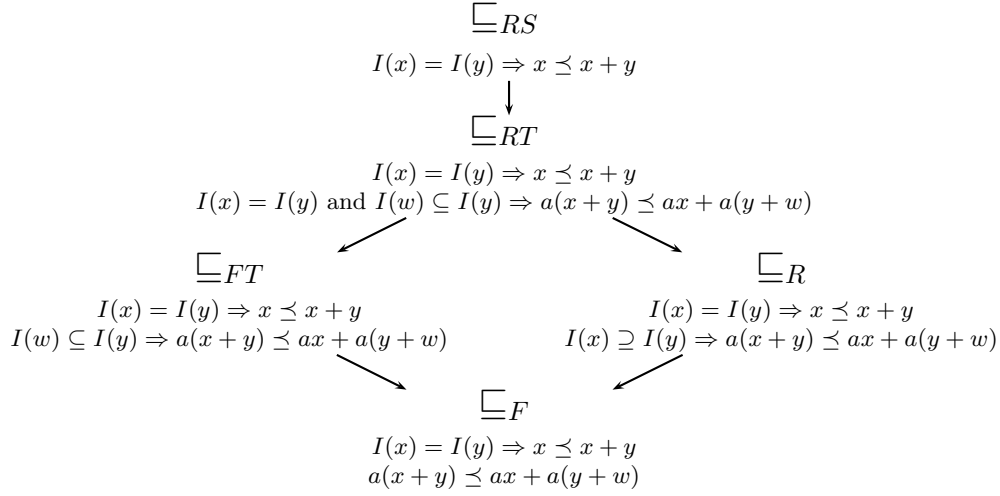


Figure 4: Inclusion relation for the ready simulation preorder and its associated linear semantics.

Figure 4 shows the already known relations between the semantics of the spectrum in the ready simulation layer. However, we want to stress the fact that once the new axiomatizations are proved to be correct, those relations became obvious since the four constraints defined above trivially satisfy $M_{RT}(x, y, w) \Rightarrow M_{FT}(x, y, w) \wedge M_R(x, y, w)$ and $M_{FT}(x, y, w) \vee M_R(x, y, w) \Rightarrow M_F(x, y, w)$. It is even more important that the tight relations and the subtle differences between these semantics clearly stand out by just looking at their axiomatizations.

Certainly, if we compare our new axiomatizations and those in Table 1, the use of conditions in our axioms could be on the grounds that complex conditions could be used to hide the complexity of the semantics. However, the conditions that we have introduced for the alternative axiomatizations of the semantics in the spectrum are very simple. In any case, our main interest was to obtain a uniform presentation of the axiomatizations that could be used to simplify their generic algebraic study.

Corollary 3.3.

- (1) \sqsubseteq_{FT} is axiomatized by the set $\{B_1-B_4, (RS), (ND_0^{FT})\}$, where (ND_0^{FT}) is the instance of (ND^{FT}) where w is $\mathbf{0}$.
- (2) \sqsubseteq_{RT} is axiomatized by $\{B_1-B_4, (RS), (ND_0^{RT})\}$, where (ND_0^{RT}) is the instance of (ND^{RT}) where w is $\mathbf{0}$.

Proof. Note that for the proof of Proposition 3.2 only the case $w = \mathbf{0}$ is needed. □

Even if the simplifications above are possible, we prefer to maintain the general forms of the axioms (ND^{FT}) and (ND^{RT}) to keep all axiomatizations as similar as possible, which will come in handy when proving general properties of the semantics.

Corollary 3.4.

- (1) \sqsubseteq_F can be axiomatized by the axioms $\{B_1-B_4, (ND^F)\}$.
- (2) \sqsubseteq_R can be axiomatized by the axioms $\{B_1-B_4, (ND^R)\}$.

	B	RS	PW	RT	FT	R	F	CS	CT	S	T
$(x + y) + z \simeq x + (y + z)$	+	+	+	+	+	+	+	+	+	+	+
$x + y \simeq y + x$	+	+	+	+	+	+	+	+	+	+	+
$x + 0 \simeq x$	+	+	+	+	+	+	+	+	+	+	+
$x + x \simeq x$	+	+	+	+	+	+	+	+	+	+	+
$I(x) = I(y) \Rightarrow a(x + y) \simeq a(x + y) + ay$			+	v	v	v	v	v	v	v	v
$a(bx + by + z) \simeq a(bx + z) + a(by + z)$			+	v	v	v	v		v		v
$I(x) = I(y) \Rightarrow ax + ay \simeq a(x + y)$				+	+	v	v		v		v
$ax + ay \simeq ax + ay + a(x + y)$					+		v		v		v
$a(bx + u) + a(by + v) \simeq a(bx + by + u) + a(by + v)$						+	+		v		v
$ax + a(y + z) \simeq ax + a(x + y) + a(y + z)$							+		v		v
$a(x + by + z) \simeq a(x + by + z) + a(by + z)$								+	v	v	v
$a(bx + u) + a(cy + v) \simeq a(bx + cy + u + v)$									+		v
$a(x + y) \simeq a(x + y) + ay$										+	v
$ax + ay \simeq a(x + y)$											+

Table 2: Axiomatization for the equivalences in the linear time-branching time spectrum.

Proof. Note that (ND^F) implies (RS) and therefore (ND^R) implies (RS) , by taking $y = \mathbf{0}$ and $w = y$. \square

3.1.2. *Equivalences and their preorders.* Let us now study the equivalences and first of all note that the axiom (ND) controlling the reduction of non-determinism has been presented as an inequational axiom. Certainly, it cannot simply be replaced by the corresponding equation since, in general, it is not true that $ax + ay \simeq a(x + y)$. However, the two dimensions corresponding to (RS) and (ND^Z) that control the “growth” of a process with respect to a preorder \preceq are not orthogonal; for example, $a(x + y) \preceq a(x + y) + ax$ can be derived either by an application of (ND^{FT}) or by one of (RS) . As a consequence of the relation between these two axioms, once (RS) is assumed then the inequational axiom (ND) can be substituted by its (stronger) equational form

$$(ND_{\equiv}) \quad M(x, y, w) \Rightarrow ax + a(y + w) + a(x + y) \simeq ax + a(y + w).$$

As above, we write (ND_{\equiv}^Z) for the concrete instances of this axiom for $Z \in \{F, R, FT, RT\}$.

Proposition 3.5.

(1) *The set $\{B_1\text{--}B_4, (RS), (ND)\}$ is logically equivalent to $\{B_1\text{--}B_4, (RS), (ND_+)\}$, where (ND_+) is the axiom*

$$M(x, y, w) \Rightarrow ax + a(y + w) + a(x + y) \preceq ax + a(y + w).$$

(2) *$\{B_1\text{--}B_4, (RS), (ND_+)\}$ is logically equivalent to $\{B_1\text{--}B_4, (RS), (ND_{\equiv})\}$.*

Proof.

(1) We only need to prove the implication from right to left, since the other follows from \preceq being a precongruence. For that, from (RS) we get $a(x + y) \preceq a(x + y) + ax + a(y + w)$ whence, using (ND_+) , $a(x + y) \preceq ax + a(y + w)$.

(2) We only need to prove that, if $M(x, y, w)$, then

$$\{B_1-B_4, (RS), (ND_+)\} \vdash ax + a(y + w) \preceq ax + a(y + w) + a(x + y),$$

which follows from (RS) . □

This result can be interpreted as saying that the only way to “enlarge” a process is by extending its possible behaviors by means of the “dynamic” simulation axioms; the static rules, (ND) and its variants, instead generate new identifications among processes.

Actually, any complete axiomatization of a preorder that contains the axiom (RS) can be turned into an equivalent axiomatization by replacing every inequality $u \preceq v$ by $u + v \simeq v$.

Proposition 3.6. *Let $Q = \{B_1-B_4, (RS)\} \cup Q'$ be an axiomatization of an order \sqsubseteq such that $\sqsubseteq \subseteq I$. Then, the equational variant of Q , $Q^= = \{B_1-B_4, (RS)\} \cup \{M \Rightarrow u + v \simeq v \mid M \Rightarrow u \preceq v \in Q'\}$ is also an axiomatization of \sqsubseteq .*

Proof. Analogous to the particular case considered in Proposition 3.5 above. For the sake of clarity we have preferred to present the particular case before, because it is easily stated and it corresponds to the most important instance of the general result. □

Finally, to conclude this section we gather in Table 2 axiomatic characterizations for the semantic equivalences that are an alternative to the classic axioms appearing in [58].

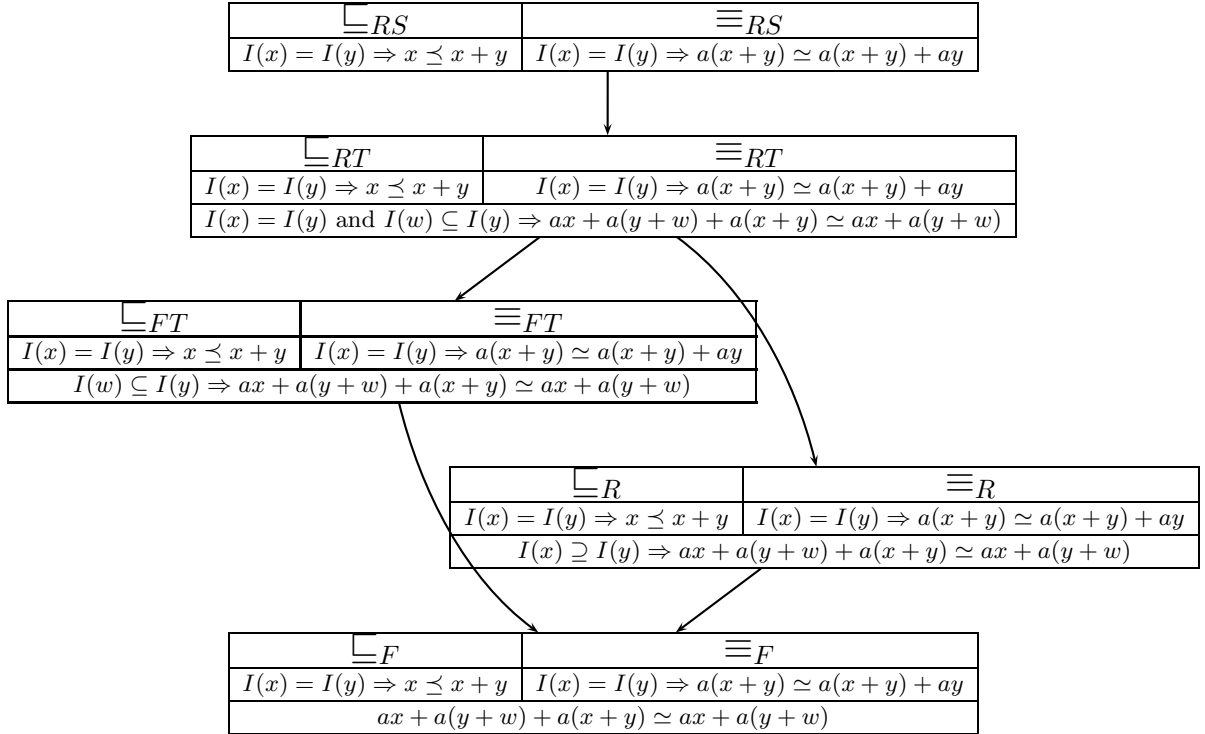


Figure 5: Axioms for the ready simulation layer of semantics.

Following the same ideas that we have already discussed for the preorders, a key point is to find the equations that characterize the simulation equivalence that governs each layer.

As showed in [24], there is a generic axiom that we can use:

$$(NS_{\equiv}) \quad N(x, y) \Rightarrow a(x + y) \simeq a(x + y) + ay.$$

We consider the instantiated equation that characterizes the ready simulation equivalence:

$$(RS_{\equiv}) \quad I(x) = I(y) \Rightarrow a(x + y) \simeq a(x + y) + ay,$$

and the rest of the characterization follows by using the equation (ND_{\equiv}) presented above.

Proposition 3.7.

- (1) *The failure equivalence \equiv_F is axiomatized by $\{B_1\text{-}B_4, (RS_{\equiv}), (ND_{\equiv}^F)\}$.*
- (2) *The readiness equivalence \equiv_R is axiomatized by $\{B_1\text{-}B_4, (RS_{\equiv}), (ND_{\equiv}^R)\}$.*
- (3) *The failure trace equivalence \equiv_{FT} is axiomatized by $\{B_1\text{-}B_4, (RS_{\equiv}), (ND_{\equiv}^{FT})\}$.*
- (4) *The ready trace equivalence \equiv_{RT} is axiomatized by the set $\{B_1\text{-}B_4, (RS_{\equiv}), (ND_{\equiv}^{RT})\}$.*

Proof. To prove these results we can compare the new and old axiomatizations similarly as we did in the proof of Proposition 3.5 or alternatively make use of the “ready to preorder” algorithm thoroughly studied in [4, 18, 21]. \square

The results in this section clarify the entanglement between axiomatizations for preorders and equivalences. For example: for the ready simulation and its associated linear semantics, we just need three axioms (RS) , (RS_{\equiv}) and (ND_{\equiv}) —conveniently instantiated—to characterize the 10 relations (orders and equivalences) involved, as summarized in Figure 5.

3.2. The coarsest semantics in the spectrum. The results in Section 3.1 show the relations between the ready simulation and the linear semantics naturally associated to it. The same phenomenon occurs for other simulations. In this section we focus on the bottom part of the spectrum where lie the simulation semantics coarser than ready simulation: plain and complete simulation, and the semantics coarser than these. For the simulation semantics we obtain the corresponding axiomatizations simply by considering the universal constraint for the case of plain simulations and the complete constraint for complete simulations:

$$\begin{array}{ll} \text{Simulation} & U(x, y) \iff \text{true} \\ \text{Complete simulations} & C(x, y) \iff (x = \mathbf{0} \text{ iff } y = \mathbf{0}) \end{array}$$

Trace and completed trace semantics can be defined by simply adding our axiom (ND^F) to the appropriate instance of

$$(NS) \quad N(x, y) \Rightarrow x \preceq x + y.$$

Proposition 3.8.

- (1) \sqsubseteq_T is axiomatized by the axioms³ $\{B_1\text{-}B_4, (S), (ND^F)\}$.
- (2) \sqsubseteq_{CT} is axiomatized by the axioms $\{B_1\text{-}B_4, (CS), (ND^F)\}$, where (CS) is the instantiation of (NS) taking $C(x, y)$ as $N(x, y)$.

Proof.

³Note that (S) is equivalent to (US) , the instantiation of (NS) with U as N .

- (1) The classic axiomatization of trace semantics is given by $\{B_1-B_4, (S), (T)\}$, where (T) is the axiom $ax + ay \simeq a(x + y)$. Note that $\{B_1-B_4, (S), (T)\}$ is logically equivalent to $\{B_1-B_4, (S), (T_{\sqsubseteq})\}$, where (T_{\sqsubseteq}) is the axiom $a(x + y) \preceq ax + ay$, because (S) can be used to obtain $ax \preceq a(x + y)$ and $ay \preceq a(x + y)$. And it is immediate that (ND^F) implies (T_{\sqsubseteq}) . Also, $\{(S), (T_{\sqsubseteq})\} \vdash a(x + y) \preceq ax + a(y + w)$, since $a(x + y) \preceq ax + ay$ by (T_{\sqsubseteq}) and $ax + ay \preceq ax + a(y + w)$ by (S) .
- (2) Analogous to the previous case once we realize that the classic axiom for completed trace, $(CT) a(bx+u)+a(cy+v) \simeq a(bx+cy+u+v)$, is equivalent to the conditional axiom $C(x, y) \Rightarrow ax+ay \simeq a(x+y)$. This follows because $bx+u$ and $cy+v$ are two independent patterns describing non-null processes and when the condition is instantiated with x and y equal to $\mathbf{0}$ the identity is trivial: $a\mathbf{0} + a\mathbf{0} \simeq a\mathbf{0}$. \square

By an argument analogous to that in Proposition 3.5, we can obtain for \sqsubseteq_T the axiomatization $\{B_1-B_4, (S), (ND_{\sqsubseteq}^F)\}$. Note that although (ND_{\sqsubseteq}^F) is an equation, this axiomatization is not the classic one; obviously, $(T) ax + ay = a(x + y)$ implies (ND_{\sqsubseteq}^F) but the converse is false.

It is easy to check that in the case of trace semantics, the particular instance (ND_0) of the axiom (ND) with w equal to $\mathbf{0}$ is powerful enough to generate the trace preorder. This was certainly not the case when we were under ready simulation, where (ND_0) just generates the failure trace preorder instead of the coarser failures preorder.

It is also interesting to note that for the trace semantics the symmetric version of (ND) ,

$$(ND_{vw}) \quad a(x + y) \preceq a(x + v) + a(y + w),$$

is also valid, so we can take both $\{B_1-B_4, (S), (ND_{vw})\}$ and $\{B_1-B_4, (S), (ND_{vw}^{\equiv})\}$, where

$$(ND_{vw}^{\equiv}) \quad a(x + v) + a(y + w) + a(x + y) \simeq a(x + v) + a(y + w),$$

as alternative axiomatizations of the trace preorder.

Should we expect another diamond of “reasonable” semantics under plain simulation in the spectrum? Were that to be the case, why have we only found the trace semantics?

In order to answer these questions, note that the diamond of semantics under ready simulation was completely governed by the function I , which appears in the constraints of the different instantiations of the axiom (ND) . For plain simulations, however, the trivially true predicate $U(x, y)$ corresponds to the observation function that can see nothing. As a consequence, if we substitute U for I in each of the four constraints of the diamond they all collapse into a single one: trace semantics. Nevertheless, an alternative path can be explored to obtain new semantics: let us keep the different axioms (ND^Z) the way they stand and simply replace (RS) by (S) . Then we obtain the following results:

Proposition 3.9. $\{B_1-B_4, (S), (ND^{FT})\}$ is another axiomatization of trace semantics. Hence, under (S) the failures and the failure trace axioms generate the same preorder, namely the trace preorder.

Proof. $\{B_1-B_4, (S), (ND_0)\}$ is a complete axiomatization of trace preorder, and (ND_0) is a particular case of (ND^{FT}) . \square

The axioms corresponding to readiness and ready trace, however, give rise to two new semantics that we shall name *extended ready* and *extended ready trace* semantics. They are defined by the order obtained by inclusion of the offers of the processes, either just at the end of a trace, or after each action within it: in order to have $p \sqsubseteq_{ER} q$, for each $p \xrightarrow{\alpha} p'$

with $I(p') = R$ we need some $q \xrightarrow{\alpha} q'$ with $I(q') \supseteq R$; the extended ready trace preorder \sqsubseteq_{ERT} is defined analogously, but using ready traces.

Proposition 3.10.

- (1) The set $\{B_1\text{-}B_4, (S), (ND^R)\}$ is an axiomatization of \sqsubseteq_{ER} .
- (2) The set $\{B_1\text{-}B_4, (S), (ND^{RT})\}$ is an axiomatization of \sqsubseteq_{ERT} .

Let us now consider the versions of the axioms (ND^R) , (ND^{FT}) , (ND^{RT}) where the constraint I has been replaced by the completeness condition C defined by $C(x) \iff x = \mathbf{0}$:

$$\begin{aligned} (C\text{-}ND^R) \quad M_{CR}(x, y, w) &\iff (C(x) \text{ implies } C(y)) \\ (C\text{-}ND^{FT}) \quad M_{CFT}(x, y, w) &\iff (C(y) \text{ implies } C(w)) \\ (C\text{-}ND^{RT}) \quad M_{CRT}(x, y, w) &\iff ((C(x) \text{ iff } C(y)) \text{ and } (C(y) \text{ implies } C(w))) \end{aligned}$$

Once again, we simply obtain three alternative axiomatizations of the completed trace semantics.

Proposition 3.11. *The following axiomatizations are equivalent:*

- (1) $\{B_1\text{-}B_4, (CS), (ND^F)\}$.
- (2) $\{B_1\text{-}B_4, (CS), (C\text{-}ND^R)\}$.
- (3) $\{B_1\text{-}B_4, (CS), (C\text{-}ND^{FT})\}$.
- (4) $\{B_1\text{-}B_4, (CS), (C\text{-}ND^{RT})\}$.

Proof. Clearly, (1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (4) and therefore it is enough to prove that (4) \Rightarrow (1). If x and y are not $\mathbf{0}$ we can apply $(C\text{-}ND^{RT})$ to obtain the inequality in (ND^F) . If x is $\mathbf{0}$ but y is not, we need to obtain $ay \preceq a\mathbf{0} + a(y+w)$. By (CS) we have $y \preceq y+w$ and then $ay \preceq a(y+w)$; applying (CS) again, $a(y+w) \preceq a(y+w) + a\mathbf{0}$ and thus $ay \preceq a\mathbf{0} + a(y+w)$. If y is $\mathbf{0}$ but x is not, we need to obtain $ax \preceq ax + aw$, which results from an immediate application of (CS) . Finally, if both x and y are $\mathbf{0}$, $a\mathbf{0} \preceq a\mathbf{0} + aw$. \square

As before, if we consider the original axioms (ND^R) , (ND^{FT}) , and (ND^{RT}) we obtain, together with an alternative axiomatization of the completed trace semantics, two new semantics.

Proposition 3.12. *The set $\{B_1\text{-}B_4, (CS), (ND^{FT})\}$ is logically equivalent to $\{B_1\text{-}B_4, (CS), (ND^F)\}$. Hence, under (CS) , the failures and the failure trace axioms generate the same semantics.*

Proof. It is enough to prove that $(C\text{-}ND^{FT})$ can be derived from $\{B_1\text{-}B_4, (CS), (ND^F)\}$.

- If y is $\mathbf{0}$ we then have w equal to $\mathbf{0}$ and can apply (ND^{FT}) .
- If y is not $\mathbf{0}$ we can apply (ND_0^{FT}) to obtain $a(x+y) \preceq ax + ay$ and then (CS) to conclude that $a(x+y) \preceq ax + a(y+w)$. \square

By contrast, as happened for plain simulations, under (CS) the axioms of the ready semantics generate two slightly different versions of the extended ready and extended ready trace semantics introduced before, that we call *extended complete ready* and *extended complete ready trace* semantics. In order to have $p \sqsubseteq_{ECR} q$, whenever $p \xrightarrow{\alpha} p'$ with $I(p') \neq \emptyset$ we require some $q \xrightarrow{\alpha} q'$ with $I(q') \supseteq I(p')$, but if $I(p') = \emptyset$ then the corresponding q' also has to satisfy $I(q') = \emptyset$. The extended complete ready trace preorder \sqsubseteq_{ECRT} is defined in an analogous way, starting from the ready traces of the processes.

As we did in Section 3.1.2, we can prove that the axioms that characterize trace and completed trace preorders reflect the fact that the order relation is inherited from simulation

and complete simulation, respectively, and that the role of the static rules is to introduce identifications. As stated in Proposition 3.12 above, the only inequation that we use to axiomatize the trace and completed trace orders is (S) , the remaining axioms being equational axioms.

Proposition 3.13.

- (1) $\{B_1-B_4, (S), (ND^F)\}$ is logically equivalent to $\{B_1-B_4, (S), (ND_{\equiv}^F)\}$.
- (2) $\{B_1-B_4, (CS), (ND^F)\}$ is logically equivalent to $\{B_1-B_4, (CS), (ND_{\equiv}^F)\}$.

A similar discussion could have been carried out for trace and completed trace equivalences, and indeed a very natural axiomatization for these relations can be obtained based on the corresponding instantiation of the (NS_{\equiv}) equation:

$$\begin{aligned} (S_{\equiv}) \quad & a(x + y) \simeq a(x + y) + ay \\ (CS_{\equiv}) \quad & C(x, y) \Rightarrow a(x + y) \simeq a(x + y) + ay. \end{aligned}$$

Proposition 3.14.

- (1) The trace equivalence \equiv_T is axiomatized by $\{B_1-B_4, (S_{\equiv}), (ND_{\equiv}^F)\}$.
- (2) The completed trace equivalence \equiv_{CT} is axiomatized by $\{B_1-B_4, (CS_{\equiv}), (ND_{\equiv}^F)\}$.

To conclude this section devoted to the unification of the equational characterizations of process semantics, we present in Figure 6 a condensed view of our new spectrum. This presentation exploits in an expressive way the two dimensions of the picture, which in fact reflects a tridimensional structure. On the lefthand side the constrained simulations and bisimulations appear, totally ordered from top to bottom. Each constrained simulation generates a layer of semantics. Here, we have only detailed the layers corresponding to ready simulation and that of plain simulation. As a matter of fact, the latter degenerates to a single point due to the simplicity of the constraint U governing plain simulations. The naturality of the semantics appearing in this part of the spectrum is illustrated by our generic axiomatization, where a single (constrained) simulation axiom governs all the constrained simulation semantics, whereas adding a single axiom we complete the axiomatizations of each of the linear semantics at the righthand side of the picture.

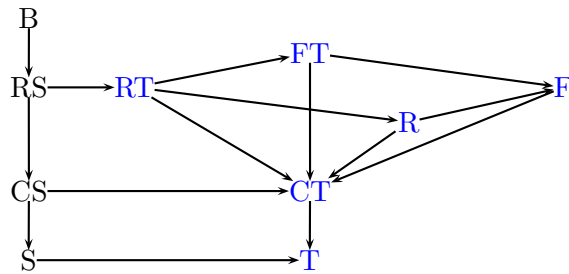


Figure 6: New view of the linear time-branching time spectrum.

4. OBSERVATIONAL SEMANTICS

Along Section 3 we have presented some views of the axiomatizations for process semantics that highlight the common properties and the subtle differences between them; likewise these

views of the axiomatic characterizations point out the similarities between the preorder and the equivalence of a given semantics.

In this section we focus on the characterizations of process semantics based on observations. Indeed, this idea of determining the semantics by means of observations lies deep inside the foundations of process theory.

Our calculus is founded in two central ideas. The first is observation; [...] two systems are indistinguishable if we cannot tell them apart without pulling them apart. We therefore give a formal definition of observation equivalence and investigate its properties. [41]

Imagine there is an observer with a notebook who watches the process and writes down the name of each event as it occurs. [32]

Besides the classical references to Milner and Hoare, this idea of observation pervades the Hennessy’s testing methodology [30] and most of the work on linear semantics. Observations, in spite of the variations in different proposals, constitute a denotational space closely related to the classical developments of semantics based on denotations for programming languages [52].

In this section we will show how most of the semantics can be characterized with one of the two main families of observations:

- Branching general observations, Section 4.1, that are essentially labeled trees, that characterize the simulation semantics: simulation, complete simulation, ready simulation, nested simulation, . . .
- Linear observations, a simplified case of branching observations, Section 4.2, that characterize the linear semantics: traces, failures, readiness, ready trace, . . .

We consider also in Section 4.3 a more exotic kind of observations, deterministic branching observations, which are essentially deterministic trees. Possible worlds semantics is the only semantics appearing in the classical spectrum in this class, although, our general approach will show how this kind of observations define new full families of process semantics.

To develop this observational characterization for process semantics allows us to deepen into the ultimate nature of the similarities and differences between them. Along this section we present a thorough study of the local observation functions that generate the local observations of the states, Figure 10. For the linear case, there is also the possibility of observing this local information in a partial way and this is how for each local observer, in principle, up to four different semantics can be obtained. This fact explains the classic diamond below the ready simulation semantics formed by the failures, failure trace, readiness, and ready trace semantics. Again, the generality of our study makes it exportable to other simulation layers enriching and completing the spectrum of semantics, Figure 11.

Finally, from a methodological point of view, the unification of observational semantics that we present in this section introduces all the technical machinery needed to rewrite the proofs of Section 3 in a generic way, proving that the two unification procedures produce characterizations of the same semantics. We will address this topic in Section 5. Let us now concentrate on the observational semantics.

4.1. Branching general observations. In order to characterize the simulation semantics in an extensional way we need local and branching general observations.

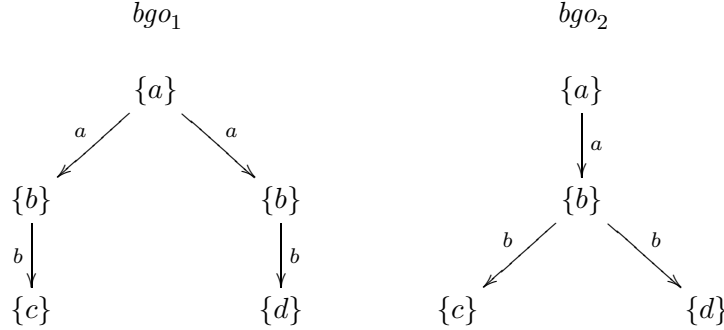


Figure 7: Two branching observations.

Definition 4.1. The sets L_N of *local observations* corresponding to each of the constrained simulations in the spectrum, and $L_N(p)$ of observations associated to a process p , are defined as follows:

- Universal (or Plain) simulation: $L_U = \{\cdot\}$; $L_U(p) = \cdot$.
- Ready simulation: $L_I = \mathcal{P}(Act)$; $L_I(p) = I(p)$.
- Complete simulation: $L_C = Bool$; $L_C(p)$ is *true* if $I(p) = \emptyset$ and *false* otherwise.
- Trace simulation⁴: $L_T = \mathcal{P}(Act^*)$; $L_T(p) = T(p)$, the set of traces of p .
- 2-nested simulation: $L_S = \{\llbracket p \rrbracket_S \mid p \in BCCSP\}$; $L_S(p) = \llbracket p \rrbracket_S$, where $\llbracket p \rrbracket_S$ represents the equivalence class of p with respect to the simulation equivalence.

Definition 4.2.

- (1) A *branching general observation* (bgo for short) of a process is a finite, non-empty tree whose arcs are labeled with actions in Act and whose nodes are labeled with local observations from L_N , for N a constraint; the corresponding set BGO_N is recursively defined as:

- $\langle l, \emptyset \rangle \in BGO_N$ for $l \in L_N$.
- $\langle l, \{(a_i, bgo_i) \mid i \in 1..n\} \rangle \in BGO_N$ for every $n \in \mathbb{N}$, $a_i \in Act$ and $bgo_i \in BGO_N$.

- (2) The set $BGO_N(p)$ of branching general observations of p corresponding to the constraint N is

$$BGO_N(p) = \{\langle L_N(p), S \rangle \mid S \subseteq \{(a, bgo) \mid bgo \in BGO_N(p'), p \xrightarrow{a} p'\}\}.$$

- (3) We write $p \leq_N^b q$ if $BGO_N(p) \subseteq BGO_N(q)$.

In Figure 7 some simple examples of bgo's for $N = I$ are shown. We represent bgo_1 as

$$\langle \{a\}, \{(a, \langle \{b\}, \{(b, \langle \{c\}, \emptyset \rangle)\}), (a, \langle \{b\}, \{(b, \langle \{d\}, \emptyset \rangle)\})\} \rangle \rangle$$

and bgo_2 as

$$\langle \{a\}, \{(a, \langle \{b\}, \{(b, \langle \{c\}, \emptyset \rangle), (b, \langle \{d\}, \emptyset \rangle)\})\} \rangle.$$

We use braces for the set of children of a node, parentheses to represent a branch of the tree as a pair (initial arc, subtree below), and angular brackets to represent each tree as a pair (root, children).

⁴Trace simulations are the only ones in this list that do not appear in [58]. They can be defined as T -simulations, with $T(x, y) ::= T(x) = T(y)$, and the general theory about constrained simulations in [24] applies to them. In particular, they can be axiomatized as stated in Proposition 3.1(3), page 13.

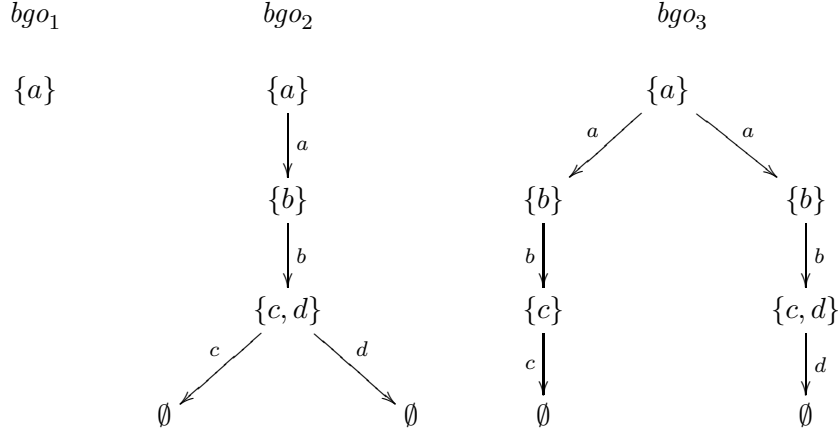


Figure 8: Three branching observations.

Note that the bgo’s of a process p described by its transition system can be generated by inductively applying the clauses defining the set $BGO_N(p)$, even when p is infinite. For instance, if $N = I$ and we consider the process $p ::= c.p$ defining a clock, since $\emptyset \subseteq \{(c, bgo) \mid bgo \in BGO_I(p), p \xrightarrow{c} p\}$, it follows that $\langle \{c\}, \emptyset \rangle \in BGO_I(p)$. But now $\{(c, \langle \{c\}, \emptyset \rangle)\} \subseteq \{(c, bgo) \mid bgo \in BGO_I(p), p \xrightarrow{c} p\}$ and therefore $\langle \{c\}, \{(c, \langle \{c\}, \emptyset \rangle)\} \rangle \in BGO_I(p)$, and so on.

It is clear that the bgo’s of a process have an operational flavor. The nodes of the observations correspond to its states and the arcs to its transitions; this is why we will be able to define the orders associated to the different simulation semantics simply by set inclusion over the sets of bgo’s.

Let us also comment on the fact that in all five cases that we have considered in Definition 4.1, which correspond to the five constrained simulation semantics in the spectrum, the local observation functions L_N define a representation of the equivalence relation N used to define the constrained simulation relations. This means that we have $L_N(p) = L_N(q) \iff pNq$.

Example 4.3. For $N = I$, if $x = b(c + d)$ and $y = bc + bd$, then for $p = a(x + y)$ we have $bgo_k \in BGO_I(p)$ for $k \in \{1, 2, 3\}$, where the bgo’s are depicted in Figure 8. It is easy to check that all of them are also branching observations of $q = a(x + y) + ax$. As a matter of fact, we have $BGO_I(p) = BGO_I(q)$. Note that in order to obtain $bgo_3 \in BGO_I(p)$ we need to combine two different observations of the process $x + y$, which is the only p' such that $a(x + y) \xrightarrow{a} p'$.

In contrast, for $p = a(bc + bd)$ and $q = abc + abd$, $BGO_I(q) \not\subseteq BGO_I(p)$, since for the branching observation bgo_1 in Figure 7 we have $bgo_1 \in BGO_I(q)$ and $bgo_1 \notin BGO_I(p)$. And also, we have $BGO_I(p) \not\subseteq BGO_I(q)$, since for bgo_2 as in Figure 7 we have $bgo_2 \in BGO_I(p)$, but $bgo_2 \notin BGO_I(q)$. The key idea is that we can indeed include in a single bgo two separated computations but we cannot “mix” two different ones, even if the labels both in their initial transitions and in the local observations of the reached nodes were the same. This is why $bgo_2 \notin BGO_I(q)$.

The following simple properties will be immediate consequences of Theorem 4.9 below; we use them here to illustrate the expressive power of each kind of bgo.

Definition 4.4. An axiom $t \preceq u$, respectively $t \simeq u$, is satisfied in a model BGO_N if $BGO_N(t') \subseteq BGO_N(u')$, respectively $BGO_N(t') = BGO_N(u')$, for every possible ground instantiation $t' \preceq u'$ or $t' \simeq u'$ of the axiom.

Proposition 4.5.

- (1) The axiom (S) $x \preceq x + y$ is satisfied in the model BGO_U .
- (2) The axiom (S_{\equiv}) $a(x + y) \simeq a(x + y) + ax$ is satisfied in the model BGO_U .

Proof.

- (1) It is an immediate consequence of the fact that if $p \xrightarrow{a} p'$ then $p + q \xrightarrow{a} p'$, and therefore $\{a \mid p \xrightarrow{a}\} \subseteq \{a \mid p + q \xrightarrow{a}\}$.
- (2) Again, it is a simple exercise to check that $BGO_U(p) \subseteq BGO_U(q)$ implies $BGO_U(ap) \subseteq BGO_U(aq)$, and that if $BGO_U(p), BGO_U(q) \subseteq BGO_U(r)$, then $BGO_U(p+q) \subseteq BGO_U(r)$; in combination with (1), this produces the result. \square

Proposition 4.6. $BGO_I(p) \subseteq BGO_I(p + q)$ iff $I(q) \subseteq I(p)$.

Proof. (\Leftarrow) Since $I(p + q) = I(p)$, the root of the bgo's is the same for both processes and obviously $p + q$ has all the observations of p .

(\Rightarrow) If $I(q) \not\subseteq I(p)$, then $I(p) \neq I(p + q)$ and then no bgo of p is a bgo of $p + q$ because the roots of the observations of both processes are different. \square

The fact, that we now prove, that the observational semantics $BGO_N(p)$ can be defined in a compositional way, is an important property that will simplify the proofs of many of their properties.

Theorem 4.7. Let L be a function used as a local observation function and let us also denote by L the range of L , as done in Definition 4.1. If there exist semantic functions $+^L : L \times L \rightarrow L$ and $a^L : L \rightarrow L$ satisfying $L(ap) = a^L L(p)$ and $L(p + q) = L(p) +^L L(q)$, then:

- $BGO_N(ap) = \{\langle a^L L(p), \{a, bgo\} \mid bgo \in B\} \mid B \subseteq BGO_N(p)\}$.
- $BGO_N(p + q) = \{\langle L(p) +^L L(q), S_1 \cup S_2 \rangle \mid \langle L(p), S_1 \rangle \in BGO_N(p), \langle L(p), S_2 \rangle \in BGO_N(q)\}$.

Proof. The first equality is immediate by definition of $BGO_N(ap)$. As for the second, we only need to realize that $p + q \xrightarrow{a} r$ iff $p \xrightarrow{a} r$ or $q \xrightarrow{a} r$: then, the set of children of the root labeled by $L_N(p + q)$ at any bgo $\in BGO_N(ap)$ correspond to the union of the two sets of children that contain some bgo's of processes p_i such as $p \xrightarrow{a} p_i$ (and then $p + q \xrightarrow{a} p_i$) or q_i such that $q \xrightarrow{a} q_i$ (and then $p + q \xrightarrow{a} q_i$). Note that from the equalities above it follows that $BGO_N(p)$ can be computed compositionally. \square

In particular, $BGO_N(p)$ is compositional for any of the constraints considered in Definition 4.1.

Proposition 4.8. For $N \in \{U, I, C, T, S\}$, L_N can be defined in a compositional way over the terms in $BCCSP$.

Proof. The result for U is obvious since it is a degenerate semantics that identifies all processes. By Theorem 4.7 and Theorem 4.9 below we can conclude that the simulation semantics can indeed be denotationally defined. The result for traces is well-known, while I and C can be easily defined denotationally since $I(ap) = \{a\}$ and $I(p + q) = I(p) \cup I(q)$. \square

Now we show that bgo's characterize N -simulation semantics in all cases.

Theorem 4.9. *For all $N \in \{U, I, C, T, S\}$ and any two processes p and q , $p \sqsubseteq_{NS} q$ iff $p \leq_N^b q$.*

Proof. (\Rightarrow) Let $p = \sum \sum a p_a^i$ and $q = \sum \sum a q_a^j$; if $p \sqsubseteq_{NS} q$, then pNq and therefore $L_N(p) = L_N(q)$. Now we proceed by induction on p . If $p \equiv \mathbf{0}$ the result is trivial. Otherwise, for every $a \in I(p)$ such that $p \xrightarrow{a} p'$ there exists $q \xrightarrow{a} q'$ such that $p' \sqsubseteq_{NS} q'$. By induction hypothesis $BGO_N(p') \subseteq BGO_N(q')$ from where, by the definition of $BGO_N(p)$, it follows that $BGO_N(p) \subseteq BGO_N(q)$.

(\Leftarrow) Let us show that the relation $R = \{(p, q) \mid BGO_N(p) \subseteq BGO_N(q)\}$ is an N -simulation. If $(p, q) \in R$, then $L_N(p) = L_N(q)$ because $\langle L_N(p), \emptyset \rangle \in BGO_N(q)$ and thus pNq . Now, for each $p \xrightarrow{a} p'$ we have $\{\langle L_N(p), \{(a, bgo)\} \rangle \mid bgo \in BGO_N(p')\} \subseteq BGO(q)$ and therefore there must exist some $q \xrightarrow{a} q'$ such that $BGO_N(p') \subseteq BGO_N(q')$, so that $(p', q') \in R$. \square

Note that for this result to hold it is only required that the local observation function L_N satisfies pNq iff $L_N(p) = L_N(q)$. That is, L_N must compute a concrete representative of the equivalence class defined by N and this stresses again the interest of using behavior equivalences N as constraints for the definition of constrained simulations. Let us recall that, in principle, any behavior preorder could be used as such a constraint. For instance, the predicate I_{\supseteq} defined by $pI_{\supseteq}q$ iff $I(q) \subseteq I(p)$ could be used to define I_{\supseteq} -simulations (which in fact coincide with I -simulations). But from $I(q) \subseteq I(p)$ we cannot conclude that $L_N(p) = L_N(q)$ and, hence, either a more complicated characterization of \sqsubseteq_{NS} in terms of bgo's or an additional argument to show that $p \sqsubseteq_{I_{\supseteq}} q$ implies $I(p) \subseteq I(q)$ would be needed. And although this is obvious for a constraint as simple as I , or even T or S , it could be far from trivial for other, more complex constraints: therefore, it is always advisable to consider equivalence behaviors as constraints.

Corollary 4.10. *For any constraint N that is a behavior equivalence, whenever we have as local observation function L_N the quotient function $L_N(p) = \llbracket p \rrbracket_N$ or any concrete representation of it satisfying $L_N(p) = L_N(q) \iff pNq$, then $p \sqsubseteq_{NS} q$ iff $BGO_N(p) \subseteq BGO_N(q)$.*

The results above bring forward the fact that despite the resemblance between the bgo's of a process and its computation tree, the possibility of mixing several computations in a single branching observation makes it possible to identify non-bisimilar processes by their sets of branching observations.

4.2. Linear observations and linear time semantics. We introduce the linear observations of a process as a particular (degenerate) case of branching observations: those with a linear structure.

Definition 4.11.

- (1) The set $LGON$ of *linear general observations* (lgo for short) for a local observer L_N is the subset of BGO_N defined as:
 - $\langle l, \emptyset \rangle \in LGON$ for each $l \in L_N$.
 - $\langle l, \{(a, lgo)\} \rangle$, whenever $a \in A$ and $lgo \in LGON$.
- (2) The set of *linear general observations* of a process p with respect to the local observer L_N is $LGON(p) = BGO_N(p) \cap LGON$.

Since lgo's are linear they can be presented as traces, avoiding the sets of descendants in the bgo's. Therefore, we will consider them as elements of the set $L_N \times (Act \times L_N)^*$.

It is also clear that the set of linear observations can be defined recursively without resorting to branching observations.

Definition 4.12. The set $LGO_N(p)$ of linear general observations of a process p is recursively defined by

$$LGO_N(p) = \{\langle L_N(p) \rangle\} \cup \{\langle L_N(p), a \rangle \circ lgo \mid p \xrightarrow{a} p', lgo \in LGO_N(p')\}.$$

We can also compute $LGO_N(p)$ in a compositional way.

Proposition 4.13. Let L be a local observation function such that there exist semantic functions $+^L : L_N \times L_N \rightarrow L_N$ and $a^L : L_N \rightarrow L_N$ satisfying $L(ap) = a^L L(p)$ and $L(p+q) = L(p) +^L L(q)$. Then:

- $LGO_N(ap) = \{\langle a^L L(p) \rangle\} \cup \{\langle a^L L(p), a \rangle \circ LGO_N(p)\}$.
- $LGO_N(p+q) = \{\langle L(p) +^L L(q) \rangle \circ t \mid \langle L(p) \rangle \circ t \in LGO_N(p) \text{ or } \langle L(q) \rangle \circ t \in LGO_N(q)\}$.

Proof. Just like that of Theorem 4.7. □

Obviously, for $N = U$ we have that LGO_U is isomorphic to Act^* and thus $LGO_U(p) = Traces(p)$. By contrast, for $N = I$, $LGO_I(p)$ is the set of ready traces of p , $ReadyTraces(p)$.

Set inclusion of linear observations with respect to a local observer L_N gives us the preorder defining the corresponding semantics.

Definition 4.14. A process p is less than or equal to q with respect to the linear observations generated by L_N , denoted $p \leq_N^l q$, if $LGO_N(p) \subseteq LGO_N(q)$. We will denote the corresponding equivalence by $=_N^l$.

Proposition 4.15. (1) $\leq_U^l = \sqsubseteq_T$; (2) $\leq_I^l = \sqsubseteq_{RT}$; (3) $\leq_C^l = \sqsubseteq_{CT}$.

Proof. It is trivial, since $LGO_U(p) = Traces(p)$, $LGO_I(p) = ReadyTraces(p)$, and $LGO_C(p) = \{(false, a_1) \circ \dots \circ (false, a_n, true), (false, a_1) \circ \dots \circ (false, a_i, false) \mid a_1 \dots a_n \in CompleteTraces(p), i < n\}$. □

Proposition 4.16. For $N \in \{U, C, I, T, S\}$, if $p \sqsubseteq_{NS} q$ then $p \leq_N^l q$, but the converse is false in general.

Proof. The implication follows from Theorem 4.9 and the fact that lgo's are just a particular case of bgo's. To see that the converse is false in general consider $N = U$; we have $\sqsubseteq_{US} = \sqsubseteq_S$ and $\leq_U^l = \sqsubseteq_T$, and it is well-known that $\sqsubseteq_S \not\subseteq \sqsubseteq_T$ since, for instance, $a(b+c) \not\sqsubseteq_S ab+ac$, but $a(b+c) =_T ab+ac$. □

Therefore, by means of linear observations and set inclusion we can characterize the orders that define some of the semantics in the spectrum which are not simulation semantics. However, there are still some other semantics for which a different way of treating the linear observations is needed. We need to introduce some identifications in the corresponding domain LGO_N to obtain their characterizations.

Definition 4.17. For $\mathcal{T}, \mathcal{T}' \subseteq LGO_I$ we define the orders $\leq_I^{l\supseteq}$, \leq_I^{lf} , and $\leq_I^{lf\supseteq}$ by:

- $\mathcal{T} \leq_I^{l\supseteq} \mathcal{T}' \iff$ for all $X_0 a_1 X_1 \dots X_n \in \mathcal{T}$
there is some $Y_0 a_1 Y_1 \dots Y_n \in \mathcal{T}'$ with $X_i \supseteq Y_i$, for all $i \in 0..n$.

- $\mathcal{T} \leq_I^{lf} \mathcal{T}' \iff$ for all $X_0a_1X_1 \dots X_n \in \mathcal{T}$
there is some $Y_0a_1Y_1 \dots Y_n \in \mathcal{T}'$ with $X_n = Y_n$.
- $\mathcal{T} \leq_I^{lf\supseteq} \mathcal{T}' \iff$ for all $X_0a_1X_1 \dots X_n \in \mathcal{T}$
there is some $Y_0a_1Y_1 \dots Y_n \in \mathcal{T}'$ with $X_n \supseteq Y_n$.

Then, for each $\delta \in \{\supseteq, f, f\supseteq\}$ we write $p \leq_I^{\delta} q$ if $LGO_I(p) \leq_I^{\delta} LGO_I(q)$.

Since the definition of \leq_I^{lf} ignores all the intermediate ready sets X_i with $i < n$ and requires the final ready sets to coincide, it is obvious that it defines the readiness preorder. Let us now prove that the two semantics based on failures are also characterized by our preorders $\leq_I^{lf\supseteq}$ and \leq_I^{\supseteq} .

Proposition 4.18. *The preorder $\leq_I^{lf\supseteq}$ generates the failures preorder and \leq_I^{\supseteq} generates the failure trace preorder.*

Proof. The proof is based on the definition of initial failures of a process: we say that p rejects X if and only if $X \cap I(p) = \emptyset$. Then, $\langle \alpha, X \rangle$ is a failure of p if and only if $p \xrightarrow{\alpha} p'$ and p' rejects X . Using lgo's, for $\alpha = a_1 \dots a_n$, $\langle \alpha, X \rangle$ is a failure of p iff there exists $X_0a_1 \dots X_n \in \mathcal{T}$ such that $X_n \cap X = \emptyset$. Thus:

$$\begin{aligned}
p \sqsubseteq_F p' &\iff Failures(p) \subseteq Failures(p') \\
&\iff \langle \alpha, X \rangle \in Failures(p') \text{ for all } \langle \alpha, X \rangle \in Failures(p) \\
&\iff X_0a_1 \dots X_n \in LGO_I(p) \text{ with } X_n \cap X = \emptyset \text{ implies that there exists} \\
&\quad Y_0a_1 \dots Y_n \in LGO_I(p') \text{ with } Y_n \cap X = \emptyset,
\end{aligned}$$

and then $p \leq_I^{lf\supseteq} p'$ implies $p \sqsubseteq_F p'$.

Conversely, assume that $p \sqsubseteq_F p'$ and recall that $p \leq_I^{lf\supseteq} p'$ iff for all $t = X_0a_1 \dots X_n \in LGO_I(p)$ there exists $Y_0a_1 \dots Y_n \in LGO_I(p')$ such that $X_n \supseteq Y_n$. For each set X , let us denote by X^c its complement. If $t \in LGO_I(p)$, we have $\langle \alpha, X_n^c \rangle \in Failures(p)$ and therefore $\langle \alpha, X_n^c \rangle \in Failures(p')$, which implies that there exists $p' \xrightarrow{\alpha} p''$ such that $I(p'') \cap X_n^c = \emptyset$. This means that there is some $t' = Y_0a_1 \dots a_n I(p'') \in LGO_I(p')$ with $I(p'') \subseteq X_n$, and therefore we can conclude that $p \leq_I^{lf\supseteq} p'$.

The proof for failure trace is very similar and we omit it. \square

As a matter of fact, the characterization of failures by means of the reverse inclusion of offerings is not a great discovery at all: for instance, the same idea can be found in the definition of acceptance trees [29]. However, our sets of linear observations produce quite a nice characterization and allow us to forget about the notion of failures and consider instead reverse inclusion of offerings. But the most important property of our characterizations in terms of different orders on the set LGO_I is that they can be generalized to other local observation functions.

Definition 4.19. For $\mathcal{T}, \mathcal{T}' \subseteq LGO_N$ we define the orders \leq_N^{\supseteq} , \leq_N^{lf} , and $\leq_N^{lf\supseteq}$ by:

- $\mathcal{T} \leq_N^{\supseteq} \mathcal{T}' \iff$ for all $X_0a_1X_1 \dots X_n \in \mathcal{T}$
there is some $Y_0a_1Y_1 \dots Y_n \in \mathcal{T}'$ with $X_i \supseteq Y_i$ for all $i \in 0..n$.
- $\mathcal{T} \leq_N^{lf} \mathcal{T}' \iff$ for all $X_0a_1X_1 \dots X_n \in \mathcal{T}$
there is some $Y_0a_1Y_1 \dots Y_n \in \mathcal{T}'$ with $X_n = Y_n$.
- $\mathcal{T} \leq_N^{lf\supseteq} \mathcal{T}' \iff$ for all $X_0a_1X_1 \dots X_n \in \mathcal{T}$
there is some $Y_0a_1Y_1 \dots Y_n \in \mathcal{T}'$ with $X_n \supseteq Y_n$.

Then, for each $\delta \in \{\supseteq, f, f\supseteq\}$ we write $p \leq_N^{l\delta} q$ if $LGO_N(p) \leq_N^{l\delta} LGO_N(q)$.

By abuse of notation, we have used the superset inclusion symbol \supseteq in the definitions above for any N . That is indeed the right interpretation for the cases $N = I, T$; however, for $N = U, C$ the superset inclusions degenerate to equalities while for $N = S$ it should be interpreted as $\llbracket p \rrbracket_S \geq_S \llbracket q \rrbracket_S$. Then, with the right notation we could have used such an inequality $\llbracket p \rrbracket_N \geq_N \llbracket q \rrbracket_N$ in all the cases.

When defining an observational semantics one expects the order between processes to be plain set inclusion as is the case, for instance, for the classic definition of failures semantics. Fortunately, it is easy to obtain such a characterization for the three semantics considered above by means of some suitable closure operators.

Definition 4.20. For $\mathcal{T} \subseteq LGO_N$, the following three closures are defined:

- $\overline{\mathcal{T}}^\supseteq = \{X_0a_1X_1 \dots a_nX_n \mid \text{there is some } Y_0a_1Y_1 \dots a_nY_n \in \mathcal{T} \text{ with } X_i \supseteq Y_i \text{ for } i \in 0..n\}$.
- $\overline{\mathcal{T}}^f = \{X_0a_1X_1 \dots a_nX_n \mid \text{there is some } Y_0a_1Y_1 \dots a_nX_n \in \mathcal{T}\}$.
- $\overline{\mathcal{T}}^{f\supseteq} = \{X_0a_1X_1 \dots a_nX_n \mid \text{there is some } Y_0a_1Y_1 \dots a_nY_n \in \mathcal{T} \text{ with } X_n \supseteq Y_n\}$.

Proposition 4.21. *All the operators in Definition 4.20 are indeed closures: if $\delta \in \{\supseteq, f, f\supseteq\}$ and $\mathcal{T}, \mathcal{T}' \subseteq LGO_N$, then $\mathcal{T} \subseteq \overline{\mathcal{T}}^\delta$ and $\overline{\overline{\mathcal{T}}^\delta} = \overline{\mathcal{T}}^\delta$; also, if $\mathcal{T} \subseteq \mathcal{T}'$ then $\overline{\mathcal{T}}^\delta \subseteq \overline{\mathcal{T}'}^\delta$.*

Proof. The first and third conditions are immediate from the definitions. As for the second, let $X_0a_1X_1 \dots a_nX_n \in \overline{\overline{\mathcal{T}}^f}$. Then, there exists $Y_0a_1Y_1 \dots a_nX_n \in \overline{\mathcal{T}}^f$ and thus there exists $Z_0a_1Z_1 \dots a_nX_n \in \mathcal{T}$, which implies $X_0a_1X_1 \dots a_nX_n \in \overline{\mathcal{T}}^f$; the inclusion in the other direction follows from monotonicity. Analogously for the other two operators. \square

Proposition 4.22. *For all $\delta \in \{\supseteq, f, f\supseteq\}$, $\mathcal{T} \leq_N^{l\delta} \mathcal{T}'$ iff $\overline{\mathcal{T}}^\delta \subseteq \overline{\mathcal{T}'}^\delta$.*

Proof. It is easy but tedious, so only the case $\delta = f\supseteq$ is presented in detail. Assume $\mathcal{T} \leq_N^{lf\supseteq} \mathcal{T}'$: for all $t = X_0a_1X_1 \dots a_nX_n \in \mathcal{T}$ there exists $Y_0a_1Y_1 \dots a_nY_n \in \mathcal{T}'$ with $X_n \supseteq Y_n$ and hence $t \in \overline{\mathcal{T}'}^{f\supseteq}$ and $\mathcal{T} \subseteq \overline{\mathcal{T}'}^{f\supseteq}$; $\overline{\mathcal{T}}^{f\supseteq} \subseteq \overline{\mathcal{T}'}^{f\supseteq}$ follows because of the properties of closures.

Conversely, from $\overline{\mathcal{T}}^{f\supseteq} \subseteq \overline{\mathcal{T}'}^{f\supseteq}$ it follows that $\mathcal{T} \subseteq \overline{\mathcal{T}'}^{f\supseteq}$ and thus for all $X_0a_1X_1 \dots a_nX_n \in \mathcal{T}$ there exists $Y_0a_1Y_1 \dots a_nY_n \in \mathcal{T}'$ with $X_n \supseteq Y_n$: therefore $\mathcal{T} \leq_N^{lf\supseteq} \mathcal{T}'$. \square

Definition 4.23. For each $\delta \in \{\supseteq, f, f\supseteq\}$, $p \in BCCSP$, and N a constraint, we define

$$LGO_N^\delta(p) = \overline{LGO_N(p)}^\delta.$$

Let us see which of the semantics in the spectrum are characterized by the orders $\leq_N^{l\delta}$ defined above.

Proposition 4.24. *For $N = U$ we have $\leq_U^l = \leq_U^{l\supseteq} = \leq_U^{lf} = \leq_U^{lf\supseteq} = \sqsubseteq_T$. As a consequence, the only semantics coarser than plain simulation that can be characterized by means of linear observations using L_U is the trace semantics.*

Proof. The first three equalities are obvious since U provides useless (empty) local information ($L_U = \{\cdot\}$). The last equality was proved in Proposition 4.15(1). \square

Proposition 4.25. *For $N = C$ we have $\leq_C^l = \leq_C^{l\supseteq} = \leq_C^{lf} = \leq_C^{lf\supseteq} = \sqsubseteq_{CT}$. As a consequence, the only semantics coarser than complete simulation that can be characterized by means of linear observations using L_C is the completed trace semantics.*

Proof. Note that the local information at the intermediate steps of traces in LGO_C has to be false, since it corresponds to non-terminated states; thus, only the final states provide real information. Since in this case \supseteq corresponds to Boolean equality, the first three equalities follow; the fourth was proved in Proposition 4.15(3). \square

Proposition 4.26. *For $N = I$, $\leq_I^{lf\supseteq}$ characterizes the failures semantics, \leq_I^{lf} the readiness semantics, $\leq_I^{l\supseteq}$ the failure trace semantics, and \leq_I^l the ready trace semantics. Therefore, the possible worlds semantics is the only semantics in the lbtt spectrum coarser than ready simulation that cannot be characterized using lgo_I 's.*

Proof. We have already proved (Propositions 4.15 and 4.18) the four characterizations, while \sqsubseteq_{PW} cannot be characterized using lgo_I 's because all the information available in our lgo_I 's was needed to capture the ready trace semantic and it is well-known that the possible worlds semantics is strictly finer. \square

As we will see in Section 4.3, the possible worlds semantics is the only deterministic branching semantics in the spectrum and will require the use of the deterministic branching observations introduced there to be characterized in an observational way. This is not the case, however, for the possible futures semantics (already discussed in [58]), and the impossible futures semantics [59].

Definition 4.27.

- (1) The impossible futures semantics is defined as: $p \sqsubseteq_{IF} q$ if for all $S \subseteq \mathcal{P}(Act^*)$, if $p \xrightarrow{\alpha} p'$ with $T(p') \cap S = \emptyset$ then there exists $q \xrightarrow{\alpha} q'$ with $T(q') \cap S = \emptyset$.
- (2) The possible futures semantics is defined as: $p \sqsubseteq_{PF} q$ if $p \xrightarrow{\alpha} p'$ then there exists $q \xrightarrow{\alpha} q'$ with $T(q') = T(p')$.

Proposition 4.28.

- (1) \leq_T^{lf} is the possible futures preorder.
- (2) $\leq_T^{lf\supseteq}$ is the impossible futures preorder.

Proof.

- (1) Obvious.
- (2) Assume that $p \leq_T^{lf\supseteq} q$. Then $p \xrightarrow{\alpha} p'$, with $\alpha = a_1 \dots a_n$, implies $q \xrightarrow{\alpha} q'$ with $T(q') \subseteq T(p')$. Therefore, if $p \xrightarrow{\alpha} p'$ with $T(p') \cap X = \emptyset$ then $q \xrightarrow{\alpha} q'$ with $T(q') \cap X = \emptyset$ which implies $p \sqsubseteq_{IF} p'$.

Conversely, if $p \sqsubseteq_{IF} q$, $t = X_0 a_0 X_1 \dots X_n \in LGO_T(p)$ and $p \xrightarrow{\alpha} p'$ with $\alpha = a_1 \dots a_n$, obviously we have $T(p') \cap T(p')^c = \emptyset$, where $T(p')^c$ just represent the complement of the set $T(p')$. Now applying the definition of \sqsubseteq_{IF} , we have some $q \xrightarrow{\alpha} q'$ with $T(q') \cap T(p')^c = \emptyset$. Hence, there exists $t' = X'_0 a_0 X'_1 \dots X'_n \in LGO_T(q)$ with $T(q') \subseteq T(p')$, which implies $p \leq_T^{lf\supseteq} q$. \square

As a matter of fact, the possible futures semantics is just below the 2-nested simulation semantics in the spectrum only because the trace simulation semantics is missing there.

At this point we are ready to present our first two “missing links”, which arise through the remaining two orders: \leq_T^l and $\leq_T^{l\triangleright}$.

Definition 4.29. The *possible futures trace semantics* is defined by lgo_T 's related by \leq_T^l and the *impossible futures trace semantics* is defined by $\leq_T^{l\triangleright}$.

Let us complete this part of the new extended spectrum by introducing the diamond generated by lgo_S 's. This produces four new semantics coarser than 2-nested semantics. For instance, for the case of failures we obtain the following definition.

Definition 4.30. The *extended simulation failures* of a process p are defined as

$$\text{ExtSimFailures}(p) = \{\langle \alpha, p'' \rangle \mid \alpha \in A^*, p \xrightarrow{\alpha} p', p' \sqsubseteq_S p''\}.$$

The *simulation failures* of a process p are defined as $\text{SimFailures}(p) = \{\langle \alpha, B \rangle \mid p \xrightarrow{\alpha} p', B \cap \text{BGO}_U(p') = \emptyset\}$. We write $p \sqsubseteq_{SF} q$ iff $\text{SimFailures}(p) \subseteq \text{SimFailures}(q)$.

It can be proved that the inclusion $\text{SimFailures}(p) \subseteq \text{SimFailures}(q)$ holds if and only if $\text{ExtSimFailures}(p) \subseteq \text{ExtSimFailures}(q)$. Thus, simulation failures are essentially defined by translating the characterization of ordinary failures with the closure of readiness.

Proposition 4.31. $\leq_S^{lf\triangleright} = \sqsubseteq_{SF}$.

Proof. Analogous to the characterization of \sqsubseteq_F in terms of $\leq_I^{lf\triangleright}$. □

4.3. Deterministic branching observations.

Definition 4.32.

- (1) We say that a bgo is *deterministic* if the set of children $\{(a_i, \text{bgo}_i)\}$ of every node satisfies $a_i \neq a_j$ whenever $i \neq j$. We denote with dBGO_N the set of deterministic observations in BGO_N .
- (2) The set of deterministic branching observations (dbgo for short) of a process p is $\text{dBGO}_N(p) = \text{BGO}_N(p) \cap \text{dBGO}_N$.
- (3) We write $p \leq_N^{db} q$ if $\text{dBGO}_N(p) \subseteq \text{dBGO}_N(q)$.

Like the linear observations, the set $\text{dBGO}_N(p)$ can be defined recursively and the corresponding semantics, compositionally.

Example 4.33. For the two processes $p = a(bc + bd)$ and $q = abc + abd$ we have that both deterministic observations in Figure 9 belong to $\text{dBGO}_I(p)$ and $\text{dBGO}_I(q)$. Indeed, that must be the case since it is easy to check that $\text{dBGO}_I(p) = \text{dBGO}_I(q)$.

In order to prove that dbgo's for the constraint I characterize the possible worlds semantics we first recall the definition of that semantics in [58].

Definition 4.34. A deterministic process p is a possible world of a process q if $p \sqsubseteq_{RS} q$. The set of possible worlds of p is denoted by $\text{PW}(p)$. We define the order $p \sqsubseteq_{PW} q$ iff $\text{PW}(p) \subseteq \text{PW}(q)$.

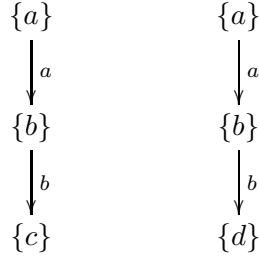


Figure 9: Deterministic branching observations.

When defining the possible worlds of a process we have to solve all the non-deterministic choices in it, each choice leading to one of its possible worlds. The same idea supports the selection of *dbgo*'s to characterize this semantics: the non-deterministic branching observations in $BGO_N(p)$ are not present in $dBGO_N(p)$, where we have instead all the possible deterministic subtrees of every branching observation.

In our proof below we will relate the *dbgo*'s in $dBGO_I(p)$ and the possible worlds in $PW(p)$. When necessary, we will consider observations in $dBGO_I(p)$ as processes in BCCSP by removing the information from their nodes; by abuse of notation we will also denote with *dbgo* the process obtained after such a removal. Also, we call *complete* those observations that, for every node labeled by an offering A , have a branch labeled by each of the actions in A .

Definition 4.35. The set of *complete deterministic* branching observations for the local observation function L_I is the set $cdBGO_I \subseteq dBGO_I$ recursively defined as:

- $\langle \emptyset, \emptyset \rangle \in cdBGO_I$.
- $\langle A, \{(a, cdbgo_a) \mid a \in A\} \rangle \in cdBGO_I$ for every $a \in A$ and $cdbgo_a \in cdBGO_I$.

For each $p \in BCCSP$ we define its set of complete deterministic branching observations $cdBGO_I(p) = dBGO_I(p) \cap cdBGO_I$.

We also associate to a deterministic process q its universal (complete deterministic) branching observation.

Definition 4.36. For a deterministic process p , its *universal deterministic* branching observation $cdbgo(p)$ is:

- $cdbgo(\mathbf{0}) = \langle \emptyset, \emptyset \rangle$.
- $cdbgo(\sum_{a \in A} ap_a) = \langle A, \{(a, cdbgo(p_a)) \mid a \in A\} \rangle$.

The following result is now immediate.

Proposition 4.37. For every $p \in BCCSP$, $cdbgo(p) \in cdBGO_I(p)$.

Lemma 4.38. For every $q \in PW(p)$, $cdbgo(q) \in cdBGO_I(p)$.

Proof. By structural induction on q :

- If q is $\mathbf{0}$, then $p \equiv \mathbf{0}$ and $\langle \emptyset, \emptyset \rangle \in cdBGO_I(\mathbf{0})$.
- If q is $\sum aq_a$, since $q \in PW(p)$ we have $q \sqsubseteq_{RS} p$. This implies $I(q) = I(p)$ and that, for all $a \in A$, there exists $p \xrightarrow{a} p_a$, $q_a \sqsubseteq_{RS} p_a$, so that $q_a \in PW(p_a)$. By induction hypothesis, $cdbgo(q_a) \in cdBGO_I(p)$. Now, by definition, $cdbgo(q) = \langle A, \{(a, cdbgo(q_a)) \mid a \in A\} \rangle$

and, from $p \xrightarrow{a} p_a$ and $I(p) = I(q)$, we conclude $cdbgo(q) \in dBGO_I(p)$ and therefore $cdbgo(q) \in cdBGO_I(p)$. \square

Lemma 4.39. *For every process q such that $cdbgo(q) \in cdBGO_I(p)$ we have $q \sqsubseteq_{RS} p$ and therefore $q \in PW(p)$.*

Proof. We will prove that the set $S = \{(q, p) \mid cdbgo(q) \in cdBGO_I(p)\}$ is a ready simulation. Obviously, for $(q, p) \in S$ it is $I(q) = I(p)$ and, if $q \xrightarrow{a} q_a$, there exists $p \xrightarrow{a} p_a$ with $cdbgo(q_a) \in cdBGO_I(p_a)$, which shows that $(q_a, p_a) \in S$ and that S is a ready simulation. \square

Theorem 4.40. *For all processes $p_1, p_2 \in BCCSP$, $p_1 \sqsubseteq_{PW} p_2$ iff $p_1 \leq_I^{db} p_2$.*

Proof. (\Leftarrow) For $q \in PW(p_1)$, by Lemma 4.38 we have $cdbgo(q) \in cdBGO_I(p_1)$ and therefore $cdbgo(q) \in cdBGO_I(p_2)$. Now, by Lemma 4.39, $q \sqsubseteq_{RS} p_2$ and thus $q \in PW(p_2)$.

(\Rightarrow) Let $dbgo \in dBGO_I(p_1)$: by definition of $dBGO_I(p_1)$ it is clear that we can extend $dbgo$ into some $dbgo' \in cdBGO_I(p_1)$. Now, by Lemma 4.39, $dbgo' \sqsubseteq_{RS} p_1$ (taking $dbgo'$ as a deterministic process). Therefore, $dbgo' \in PW(p_1)$ and thus $dbgo' \in PW(p_2)$ and, by Lemma 4.38, $cdbgo(dbgo') = dbgo' \in cdBGO_I(p_2)$: hence $dbgo \in dBGO_I(p_2)$ as required. \square

Remark 4.41. If we consider infinite processes, then our characterization of \sqsubseteq_{PW} by means of \leq_I^{db} only works if we restrict ourselves to image-finite processes. We will continue the discussion on this part when studying the logical characterization of this semantics at Section 6.

Let us briefly consider the remaining new semantics definable by means of deterministic branching observations. It is clear that in all cases the corresponding orders verify $\leq_N^b \subseteq \leq_N^{db} \subseteq \leq_N^l$, so that the associated semantics will be situated between the corresponding semantics defined by branching observations in BGO_N and linear observations in LGO_N , as is the case for the possible worlds semantics, located between the ready simulation semantics and the ready trace semantics.

Admittedly, most of these semantics are rather strange and this is probably the reason why, as far as we know, they have not been previously considered. However, the simplest of them all, that corresponding to $N = U$, has properties similar to the possible worlds semantics and, in fact, can be defined by simply removing from its definition the “ R ” in the condition $q \sqsubseteq_{RS} p$. Hence, we can regard as possible worlds those deterministic implementations where we offer just a part of the action offered by the given process.

Definition 4.42. The partial possible worlds of a process p are those deterministic processes that verify $q \sqsubseteq_S p$. We denote with $PW_U(p)$ the set of partial possible worlds of a process p and define $p \sqsubseteq_{UPW} q$ if $PW_U(p) \subseteq PW_U(q)$.

Proposition 4.43. *For all processes $p_1, p_2 \in BCCSP$, $p_1 \sqsubseteq_{UPW} p_2$ iff $p_1 \leq_U^{db} p_2$.*

Proof. Similar to Theorem 4.40, simplified by the fact that all $dbgo$ in $PW_U(p)$ satisfy $dbgo \sqsubseteq_S p$. \square

Example 4.44. We have $a \sqsubseteq_{UPW} a + b$ since $\langle \cdot, \{(a, \emptyset)\} \rangle \in dBGO_U(a + b)$. By contrast, for $p = ab + ac$ and $q = a(b + c)$ we have $p \sqsubseteq_{UPW} q$ but $q \not\sqsubseteq_{UPW} p$ because $\langle \cdot, \{(a, \langle \cdot, \{(b, \langle \cdot, \emptyset \rangle), (c, \langle \cdot, \emptyset \rangle)\})\}) \rangle \in dBGO_U(q) - dBGO_U(p)$.

Analogously, for any other constraint N we could define the N -possible worlds order \sqsubseteq_{NPW} using \sqsubseteq_{NS} instead of \sqsubseteq_S at Definition 4.42. However, it is easy to see that when N is fine enough, e.g. $N = T$, this order would become totally wrong. Instead, we can

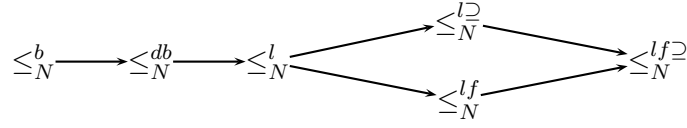


Figure 10: Basic layer in the linear time-branching time spectrum.

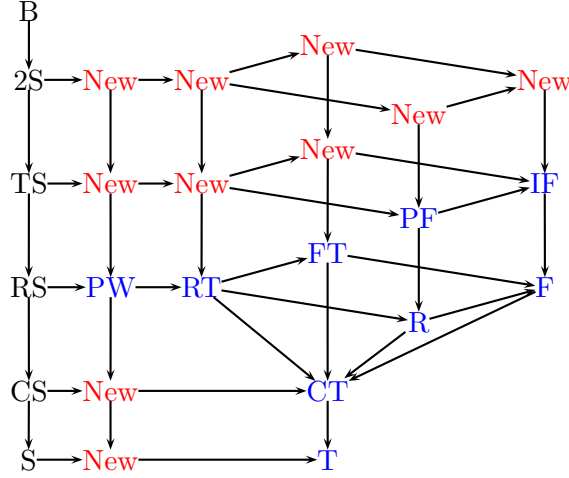


Figure 11: Semantics in the new linear time-branching time spectrum.

still consider the observations in $dBGO_N$ and by means of them we define the “reasonable” deterministic branching semantics, for any layer in the spectrum.

The extended spectrum can now be depicted as in Figures 11 and 10.

4.4. Back to branching observations. The orders $\leq_N^{l\delta}$ with $\delta \in \{\supseteq, f, f\supseteq\}$ that characterize some of the linear semantics studied in Section 4.2, restricted in several ways the use of the local information, when characterizing those semantics. The same scheme can be generalized to the branching observations. This way, for each constraint N we would obtain three new branching semantics based on bgo ’s in BGO_N which, together with the original N -simulation semantics, would constitute a diamond of branching semantics at a higher layer in our extended lbt spectrum. The introduction of these new semantics also offers a clearer view of the spectrum, with two main levels of branching and linear semantics and an intermediate one of deterministic branching semantics. Although this provides the means for obtaining a host of new semantics, it is also true that most of them are bizarre, in sharp contrast with the fact that the corresponding orders gave rise to interesting semantics when applied to linear observations.

To illustrate the comments above, next we consider in some detail the case $N = I$, which corresponds to the most interesting semantics.

Definition 4.45. For $bgo, bgo' \in BGO_I$ we define:

- $bgo \leq_I^{\supseteq} bgo' \iff (bgo = \langle A_1, S_1 \rangle \text{ and } bgo' = \langle A_2, S_2 \rangle \text{ and } A_1 \supseteq A_2 \text{ and } S_1 = \{(a_i, bgo_i) \mid i \in I\} \text{ and } S_2 = \{(a_i, bgo'_i) \mid i \in I\} \text{ and for all } i \in I (bgo_i \leq_I^{\supseteq} bgo'_i)).$

- $bgo \leq_I^f bgo' \iff (bgo = \langle A_1, \emptyset \rangle \text{ and } bgo' = \langle A_1, \emptyset \rangle) \text{ or}$
 $(bgo = \langle A_1, S_1 \rangle \text{ and } bgo' = \langle A_2, S_2 \rangle \text{ and}$
 $S_1 = \{(a_i, bgo_i) \mid i \in I\} \text{ and } S_2 = \{(a_i, bgo'_i) \mid i \in I\} \text{ and}$
for all $i \in I (bgo_i \leq_I^f bgo'_i)$.
- $bgo \leq_I^{f\supseteq} bgo' \iff (bgo = \langle A_1, \emptyset \rangle \text{ and } bgo' = \langle A_2, \emptyset \rangle \text{ and } A_1 \supseteq A_2) \text{ or}$
 $(bgo = \langle A_1, S_1 \rangle \text{ and } bgo' = \langle A_2, S_2 \rangle \text{ and}$
 $S_1 = \{(a_i, bgo_i) \mid i \in I\} \text{ and } S_2 = \{(a_i, bgo'_i) \mid i \in I\} \text{ and}$
for all $i \in I (bgo_i \leq_I^{f\supseteq} bgo'_i)$.

Definition 4.46. For $\mathcal{B}, \mathcal{B}' \subseteq BGO_I$ and $\delta \in \{\supseteq, f, f\supseteq\}$, we define the orders $\leq_I^{b\delta}$ by:

- $\mathcal{B} \leq_I^{b\delta} \mathcal{B}' \iff$ for all $bgo \in \mathcal{B}$ there exists $bgo' \in \mathcal{B}'$ with $bgo \leq_I^\delta bgo'$.

Then, we write $p \leq_I^{b\delta} q$ if $BGO_I(p) \leq_I^{b\delta} BGO_I(q)$.

It is somewhat surprising to discover that $\leq_I^{b\supseteq} = \leq_I^b$, since this was not the case for their linear “projections” $\leq_I^{l\supseteq}$ and \leq_I^l .

Proposition 4.47. For all processes $p_1, p_2 \in BCCSP$, $p_1 \leq_I^{b\supseteq} p_2$ iff $p_1 \leq_I^b p_2$.

Proof. Assume that $p_1 \leq_I^{b\supseteq} p_2$ and let $bgo \in BGO_I(p_1)$: it is clear that it can be extended into a complete $cbgo \in BGO_I(p_1)$. Then, there exists some $cbgo' \in BGO_I(p_2)$ with $cbgo \leq_I^\supseteq cbgo'$ and, since $cbgo$ is complete, $cbgo = cbgo'$ and hence $bgo \in BGO_I(p_2)$. The other implication is trivial. \square

Example 4.48. For $p_1 = a(b + c)$ and $p_2 = ab + ac$, $p_1 \leq_I^{l\supseteq} p_2$ but $p_1 \not\leq_I^l p_2$. However, $p_1 \not\leq_I^{b\supseteq} p_2$ since for $bgo = \langle \{a\}, (a, \langle \{b, c\}, \{(b, \emptyset), (c, \emptyset)\}) \rangle \rangle \in BGO_I(p_1)$ there is no $bgo' \in BGO_I(p_2)$ with $bgo \leq_I^{l\supseteq} bgo'$.

By contrast, the branching semantics defined by \leq_I^{bf} and $\leq_I^{bf\supseteq}$ are indeed new.

Example 4.49. For the processes p and q in Figure 12, $p \leq_I^{bf} q$ but $p \not\leq_I^b q$.

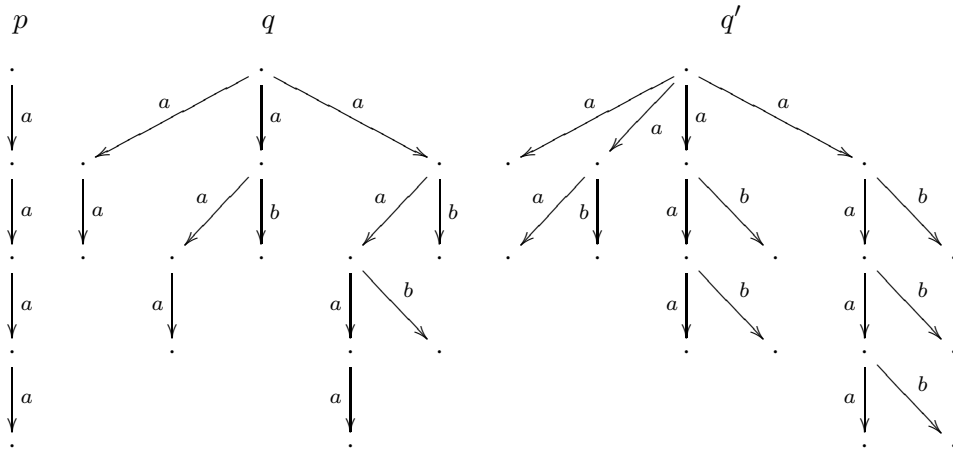


Figure 12: Three processes.

This example shows that it is quite difficult to characterize this semantics as a simulation one. Furthermore, we conjecture that it is not finitely axiomatizable in the classic way (that means using only unconditional axioms). As a matter of fact, we were also unable to find any conditional axiomatization, what we interpret as a “proof” of the fact that these new branching semantics are quite strange.

Definition 4.50. We say that $R \subseteq BGO_I \times BCCSP$ is a *final-ready simulation* when:

- $(\langle A, \emptyset \rangle, q) \in R$ implies $I(q) = A$.
- $(\langle A, \{(a_i, bgo_i)\} \rangle, q) \in R$ implies that for all $i \in I$ there exists $q \xrightarrow{a_i} q_i$ such that $(bgo_i, q_i) \in R$.

We say that p is final-ready simulated by q when for all $bgo \in BGO_I(p)$ there exists a final-ready simulation R with $(bgo, q) \in R$, and write $p \sqsubseteq_{fRS} q$.

Theorem 4.51. For all $p, q \in BCCSP$, $p \sqsubseteq_{fRS} q$ iff $p \leq_I^{bf} q$.

Example 4.52. It is easy to check that for p and q' as in Figure 12 we have $p \leq_I^{bf\supseteq} q'$ but $p \not\leq_I^{bf} q'$.

Final failure simulations are defined exactly like final-ready simulations but substituting $I(q) \subseteq Act$ for $I(q) = A$ in the first clause, giving rise to the order \sqsubseteq_{fFS} between processes.

Theorem 4.53. For all $p, q \in BCCSP$, $p \sqsubseteq_{fFS} q$ iff $p \leq_I^{bf\supseteq} q$.

As previously noted, these are certainly bizarre semantics but we believe it is interesting to indicate their existence because, by analogy to the linear case, their definitions in terms of branching observations look quite natural. However, it also seems that when dealing with branching observations the introduction of any kind of asymmetry in the treatment of local observations produces quite involved semantics.

5. RELATING THE OBSERVATIONAL AND EQUATIONAL FRAMEWORKS

In this section we tie up all loose ends and show how our unification theory is fully self-contained. Namely, we prove the results on axiomatic characterizations in Section 3 from the observational semantics developed in Section 4: we show how the equations are deduced from the observations in a general way without resorting to the already existing axiomatizations.

One of the key points of this section is to illustrate how the particular proofs needed in Section 3 for every one of the semantics can be replaced by a generic proof that stands for a whole family of semantics. In fact, we will show in Section 8 that the same proof is still valid for the new semantics suggested in Roscoe’s work.

5.1. Semantics coarser than ready simulation. Let us now see how, from this uniform definition of the linear semantics, the proofs of the correctness and completeness of the corresponding axiomatizations can be derived in a uniform way avoiding the case analyses of Sections 3.1 and 3.2. Although this could be done generically, with $N \in \{U, C, I, T\}$, we prefer to start with the particular case $N = I$, which corresponds to the most popular semantics already studied in Section 3.1.

To start with, we show how the axiomatizations can be synthesized from the observational characterizations. Our general axiom (*ND*) for the reduction of non-determinism

specifies the hypothesis $M(x, y, w)$ under which the process $ax + a(y + w)$ can be (syntactically) expanded by adding a new summand $a(x + y)$ without changing its semantics. Then, let us compare the two sides of our general axiom. Since $I(ax + a(y + w)) = I(ax) = I(a(y + w)) = \{a\}$, we have

$$\begin{aligned} LGO_I(ax + a(y + w)) &= LGO_I(ax) \cup LGO_I(a(y + w)), \\ LGO_I(a(y + w)) &= \{\{a\}\} \cup \\ &\quad \{\langle \{a\}, a, I(y + w) \rangle \circ S \mid \langle \{a\}, a, I(y) \rangle \circ S \in LGO_I(ay) \vee \\ &\quad \langle \{a\}, a, I(w) \rangle \circ S \in LGO_I(aw)\}. \end{aligned}$$

Notice then that the observations of $a(y + w)$ are exactly those of $ay + aw$ simply replacing $I(y)$ or $I(w)$, respectively, by $I(y + w) = I(y) \cup I(w)$. Analogously,

$$\begin{aligned} LGO_I(a(x + y)) &= \{\{a\}\} \cup \\ &\quad \{\langle \{a\}, a, I(x + y) \rangle \circ S \mid \langle \{a\}, a, I(x) \rangle \circ S \in LGO_I(ax) \vee \\ &\quad \langle \{a\}, a, I(y) \rangle \circ S \in LGO_I(ay)\}. \end{aligned}$$

Now, in order to get the adequate condition $M_Z(x, y, w)$ for each of the semantics, let us examine the formulas that define the preorders \leq_I^Y :

- \leq_I^l . To have $LGO_I(a(x + y)) \subseteq LGO_I(ax) \cup LGO_I(a(y + w))$ it is enough to require $\{\langle \{a\}, a, I(x) \cup I(y) \rangle \circ S \mid \langle I(x) \rangle \circ S \in LGO_I(x)\} \subseteq \{\langle \{a\}, a, I(x) \rangle \circ S \mid \langle I(x) \rangle \circ S \in LGO_I(x)\}$ and $\{\langle \{a\}, a, I(x) \cup I(y) \rangle \circ S \mid \langle I(y) \rangle \circ S \in LGO_I(y)\} \subseteq \{\langle \{a\}, a, I(y) \cup I(w) \rangle \circ S \mid \langle I(y) \rangle \circ S \in LGO_I(y)\}$. Thus, a first proposal for M_{RT} would be

$$I(y) \subseteq I(x) \wedge I(x) = I(y) \cup I(w).$$

However, due to the fact that this axiom will be used in combination with (RS) , the following, more restrictive but simpler form, can be used instead:

$$M_{RT}(x, y, w) \iff I(x) = I(y) \wedge I(w) \subseteq I(y).$$

Clearly, this form is stronger than the condition synthesized above. Reciprocally, $a(x + y) \preceq ax + a(y + w)$ can be proved from the assumptions $I(y) \subseteq I(x)$ and $I(x) = I(y) \cup I(w)$ using (RS) first to get $a(x + y) \preceq a(x + y + w)$, and then (ND) instantiated with M_{RT} to obtain $a(x + y + w) \preceq ax + a(y + w)$.

- $\leq_I^{l\supseteq}$. We need the inclusion $\overline{LGO_I(a(x + y))}^{\supseteq} \subseteq \overline{LGO_I(ax + a(y + w))}^{\supseteq}$ to hold. Since $I(x) \cup I(y) \supseteq I(x)$, the general observations in $a(x + y)$ that arise from x will be also in $\overline{LGO_I(ax)}^{l\supseteq}$. For those that arise from y , it is required that $I(x) \cup I(y) \supseteq I(y) \cup I(w)$. Once again, (RS) can be used to simplify this condition into the simpler

$$M_{FT}(x, y, w) \iff I(w) \subseteq I(y).$$

The less restrictive variant of the axiom can be derived from the stronger one and (RS) as follows. Taking $w = \mathbf{0}$, since $I(\mathbf{0}) \subseteq I(y)$ we obtain $a(x + y) \preceq ax + ay$ from $(ND)^{FT}$; in particular, $a(x + y + w) \preceq ax + a(y + w)$. Also, by (RS) , $x + y \preceq (x + y) + (x + y + w)$, from where it follows $a(x + y) \preceq a(x + y + w)$.

- \leq_I^{lf} . We consider the inclusion $\overline{LGO_I(a(x + y))}^f \subseteq \overline{LGO_I(ax + a(y + w))}^f$. We only have to consider the lgo $\langle \{a\}, a, I(x) \cup I(y) \rangle$ in $\overline{LGO_I(a(x + y))}^f$ and show that it also belongs to $\overline{LGO_I(ax + a(y + w))}^f$, since all lgo's of length greater than 1 start with the prefix $\langle \{a\}, a \rangle$. For that, either $I(x) \cup I(y) = I(x)$ or $I(x) \cup I(y) = I(y) \cup I(w)$, that is,

$I(y) \subseteq I(x)$ or $I(x) \cup I(y) = I(y) \cup I(w)$. Again, we can remove the second condition and define

$$M_R(x, y, w) \iff I(y) \subseteq I(x)$$

since, whenever $I(x) \cup I(y) = I(y) \cup I(w)$, $a(x + y + w) \preceq ax + a(y + w)$ can be obtained by taking $x = y + w$, $y = x$, and $w = \mathbf{0}$, and then by applying (RS) we conclude $a(x + y) \preceq ax + a(y + w)$.

- $\leq_I^{lf \supseteq}$. An argument analogous to the previous one leads us to check that $I(x) \cup I(y) \supseteq I(x)$ or $I(x) \cup I(y) \supseteq I(y) \cup I(w)$, and the first is certainly true.

In order to prove the completeness of our axiomatizations we introduce the following notions of head normal forms.

Definition 5.1. For $p = \sum_{a \in X_0} \sum_{i \in I_a} ap_a^i$ and $Z \in \{F, R, FT, RT\}$, its Z -head normal form $hnf^Z(p)$ is:

- For $a \in X_0$, $i \in I_a$, and $X_1 \subseteq \bigcup_{i \in I_a} I(p_a^i)$ such that $I(p_a^i) \subseteq X_1$, we define $hnf^Z(p, a, i, X_1) = a(p_a^i + \sum \{p_a^j |_{X_1} \mid j \neq i, M_Z(p_a^i, p_a^j |_{X_1}, p_a^j |_{\overline{X_1}})\})$.
- $hnf^Z(p) = p + \sum_{a \in X_0} \sum_{i \in I_a} \sum_{X_1 \subseteq \bigcup I(p_a^i)} hnf^Z(p, a, i, X_1)$.

It is clear that several redundancies arise in this definition: for example, if $Z = RT$ then $hnf^Z(p, a, i, X_1) = hnf^Z(p, a, i, I(p_a^i))$, so that the argument X_1 would not be needed in this case. We prefer to maintain the generic definition in order to allow a homogeneous treatment of all the semantics.

Proposition 5.2. For $Z \in \{RT, FT, R, F\}$, $\{B_1\text{--}B_4, (RS), (ND^Z)\} \vdash hnf^Z(p) \preceq p$.

Proof. Let $p = \sum_{a \in X_0} \sum_{i \in I_a} ap_a^i$.

Considering the definition of $hnf^Z(p, a, i, X_1)$, let us consider an enumeration of the set of j 's contributing to it: If $J_i = \{j \neq i \mid M_Z(p_a^i, p_a^j |_{X_1}, p_a^j |_{\overline{X_1}})\}$, we take $\{j_k \mid k = 1 \dots |J_i|\} = J_i$.

Then we can prove by induction on l that for all $0 \leq l \leq |J_i|$ we have $\{B_1\text{--}B_4, (RS), (ND^Z)\} \vdash a(p_a^i + \sum_{h=1}^l p_a^{j_h} |_{X_1}) \preceq ap_a^i + \sum_{h=1}^l ap_a^{j_h}$.

The case of $l = 0$ is trivial. Assuming the result for l , we prove the result for $l + 1$. From $M_Z(p_a^i, p_a^{j_{l+1}} |_{X_1}, p_a^{j_{l+1}} |_{\overline{X_1}})$ we can infer $M_Z(p_a^i + \sum_{h=1}^l p_a^{j_h} |_{X_1}, p_a^{j_{l+1}} |_{X_1}, p_a^{j_{l+1}} |_{\overline{X_1}})$ so that we can derive $\vdash a(p_a^i + \sum_{h=1}^{l+1} p_a^{j_h} |_{X_1}) \preceq a(p_a^i + \sum_{h=1}^l p_a^{j_h}) + ap_a^{j_{l+1}}$; and applying the i.h. we conclude $\{B_1\text{--}B_4, (RS), (ND^Z)\} \vdash a(p_a^i + \sum_{h=1}^{l+1} p_a^{j_h} |_{X_1} |_{X_1}) \preceq ap_a^i + \sum_{h=1}^{l+1} ap_a^{j_h}$.

From this we immediately obtain $\{B_1\text{--}B_4, (RS), (ND^Z)\} \vdash hnf^Z(p, a, i, X_1) \preceq p|_a$. Finally, adding all these inequalities we conclude $\{B_1\text{--}B_4, (RS), (ND^Z)\} \vdash hnf^Z(p) \preceq p$. \square

Let us define $l(F) = lf \supseteq$, $l(FT) = l \supseteq$, $l(R) = lf$, and $l(RT) = l$. In order to apply structural induction to prove the completeness of the axiomatizations we need to show that, whenever $p = \sum_{a \in X_0} \sum_{i \in I_a} ap_a^i$ and $p \leq_I^{l(Z)} q$, there is a summand ah_a^k of $hnf^Z(q)$ such that $p_a^i \leq_I^{l(Z)} h_a^k$ for each $a \in Act$, $i \in I_a$.

Proposition 5.3. Let $Z \in \{F, FT, R, RT\}$, and let $p = \sum_{a \in X_0} \sum_{i \in I_a} ap_a^i$, and $q = \sum_{a \in X_0} \sum_{j \in J_a} aq_a^j$. If $p \leq_I^{l(Z)} q$ then there exists a summand ah_a^k of $hnf^Z(q)$ such that $p_a^i \leq_I^{l(Z)} h_a^k$.

Proof. Using Definition 4.19 and Proposition 4.22, we need to show that there exists $\overline{LGO_I(p_a^i)}^{l(Z)} \subseteq \overline{LGO_I(h_a^k)}^{l(Z)}$ but, due to the fact that $\overline{(-)}^{l(Z)}$ is a closure operator (Proposition 4.21), it is enough to prove that $LGO_I(p_a^i) \subseteq \overline{LGO_I(h_a^k)}^{l(Z)}$. For $\langle I(p_a^i) \rangle \in LGO_I(p_a^i)$, since $p \leq_I^{l(Z)} q$ there is some q_a^j such that $\langle I(p_a^i) \rangle \in LGO_I(q_a^j)^{l(Z)}$; we then consider $hnf^Z(q, a, j, I(p_a^i)) = ah_a^k$.

If $t \in LGO_I(p_a^i)$ then $\langle I(p), a \rangle \circ t \in LGO_I(p) \subseteq \overline{LGO_I(q)}^{l(Z)}$ and there exists j_t such that $t \in \overline{LGO_I(q_a^{j_t})}^{l(Z)}$. In addition, $M_Z(q_a^j, q_a^{j_t} |_{I(p_a^i)}, q_a^{j_t} |_{\overline{I(p_a^i)}})$:

- If $Z = RT$, then $t \in LGO_I(q_a^{j_t})$ and therefore $I(q_a^{j_t}) = I(p_a^i) = I(q_a^j)$. Hence, condition $M_{RT}(q_a^j, q_a^{j_t} |_{I(p_a^i)}, \mathbf{0})$ holds and therefore $M_{RT}(q_a^j, q_a^{j_t} |_{I(p_a^i)}, q_a^{j_t} |_{\overline{I(p_a^i)}})$.
- If $Z = FT$, from $t \in \overline{LGO_I(q_a^{j_t})}^{\supseteq}$ it follows that $I(q_a^{j_t}) \subseteq I(p_a^i)$ and therefore $I(q_a^{j_t} |_{\overline{I(p_a^i)}}) = \emptyset \subseteq I(q_a^{j_t} |_{I(p_a^i)})$. Hence, $M_{FT}(q_a^j, q_a^{j_t} |_{I(p_a^i)}, q_a^{j_t} |_{\overline{I(p_a^i)}})$.
- If $Z = R$, from $\langle I(p_a^i) \rangle \in \overline{LGO_I(q_a^j)}^f$ we have that $I(p_a^i) = I(q_a^j)$ and thus $I(q_a^{j_t} |_{I(p_a^i)}) \subseteq I(q_a^j)$ and $M_R(q_a^j, q_a^{j_t} |_{I(p_a^i)}, q_a^{j_t} |_{\overline{I(p_a^i)}})$.
- For $Z = F$ it is trivial since $M_F(x, y, w)$ is always true.

Therefore $q_a^{j_t}$ is one of the summands of h_a^k and, since $t \in \overline{LGO_I(q_a^{j_t})}^{l(Z)}$, we have $p_a^i \leq_I^{l(Z)} h_a^k$. \square

Theorem 5.4 (Soundness and completeness). *For $Z \in \{RT, FT, R, F\}$:*

$$p \leq_I^{l(Z)} q \text{ iff } \{B_1\text{--}B_4, (RS), (ND^Z)\} \vdash p \preceq q.$$

Proof. (Soundness) The axiomatizations are sound because of the way they have been derived.

(Completeness) By structural induction on p .

- Let p be $\mathbf{0}$. As usual, we can consider terms up to bisimulation since $B_1\text{--}B_4$ are equations needed for all the semantics. If $p \leq_I^{l(Z)} q$, then q must be $\mathbf{0}$ (or bisimilar to $\mathbf{0}$) because the set of local observations of $\mathbf{0}$ is empty and cannot contain any observations (see Definition 4.19.)
- If $p = \sum_{a \in X_0} \sum_{i \in I_a} ap_a^i$ then, by Proposition 5.3, $p \leq_I^{l(Z)} q$ implies that there exists a summand ah_a^k of $hnf^Z(q)$ such that $p_a^i \leq_I^{l(Z)} h_a^k$. By induction hypothesis, $\{B_1\text{--}B_4, (RS), (ND^Z)\} \vdash p_a^i \preceq h_a^k$ and therefore $\{B_1\text{--}B_4, (RS), (ND^Z)\} \vdash ap_a^i \preceq ah_a^k$; adding all these inequalities and using (RS) , which is allowed because $I(p) = I(q)$, it follows that $\{B_1\text{--}B_4, (RS), (ND^Z)\} \vdash p \preceq hnf^Z(q)$ and, by Proposition 5.2, $\{B_1\text{--}B_4, (RS), (ND^Z)\} \vdash p \preceq q$. \square

5.2. The semantics that are not coarser than ready simulation. Once we have a clear picture of the semantics that are coarser than ready simulation, it is time to consider the rest of the semantics in the spectrum. Let us start with the possible futures and the impossible futures semantics. Recall that we have shown that they can be described by LGO_T observations so that they are defined by $\leq_T^{lf \supseteq}$ and \leq_T^{lf} , respectively.

We introduce the T -versions of our (ND^Z) axioms: all of them are instances of our general axiom for the reduction of non-determinism and therefore are defined by the adequate

constraint $M_Z^T(x, y, w)$. As expected, they are obtained by substituting every occurrence of I in $M_Z(x, y, w)$ by the observer T defining the traces of processes.

Definition 5.5. The constraints M_Z^T that characterize the semantics coarser than T -simulation semantics are:

$$\begin{aligned} (T\text{-}ND^F) \quad & M_F^T(x, y, w) \iff \text{true} \\ (T\text{-}ND^R) \quad & M_R^T(x, y, w) \iff T(x) \supseteq T(y) \\ (T\text{-}ND^{FT}) \quad & M_{FT}^T(x, y, w) \iff T(w) \subseteq T(y) \\ (T\text{-}ND^{RT}) \quad & M_{RT}^T(x, y, w) \iff T(x) = T(y) \text{ and } T(w) \subseteq T(y) \end{aligned}$$

As indicated in Section 4.2 (Definition 4.29), the semantics associated to the last two conditions do not appear in the lbtbt spectrum and, as far as we know, they have not been previously studied nor even defined.

Using the same arguments as in Section 5.1, we can prove that $\leq_T^{l(Z)}$ satisfies the axiom $(T\text{-}ND^Z)$ for $Z \in \{RT, FT\}$.

Proposition 5.6. $M_Z^T(x, y, w)$ implies $T(a(x+y)) = T(ax+a(y+w))$ for $Z \in \{RT, FT\}$. However, this is not the case for $Z \in \{R, F\}$

Proof. M_{RT}^T implies M_{FT}^T , and therefore $T(y+w) = T(y)$, which leads to $T(ax+a(y+w)) = T(a(x+y))$. Neither M_R^T nor M_F^T refer to w and therefore, in general, $T(ax+a(y+w)) \neq T(a(x+y))$ in those cases. \square

Note that when proving the correctness of the corresponding axiom (ND^Z) for $\leq_I^{l(Z)}$ we had $I(a(x+y)) = \{a\} = I(ax+a(y+w))$ in all cases. Now, $T(a(x+y)) = T(ax+a(y+w))$ only under the constraints corresponding to the finer semantics \sqsubseteq_{FT} and \sqsubseteq_{RT} . The properties of the prefixes appearing in all the terms in both sides of the axiom (ND) are not used anymore in the proofs in Section 5.1, so they can be transferred to the T -semantics, thus proving the correctness of $(T\text{-}ND^Z)$ for both $\leq_T^{l(RT)}$ and $\leq_T^{l(FT)}$.

The introduction of the equational version (ND_{\equiv}) of the axiom (ND) now becomes crucial in order to preserve the generality of our unifying work. We saw that under (RS) these two axioms were equivalent. However, when observing the set of traces $T(x)$ of any process, instead of just the initial offer $I(x)$ we need to consider T -simulations, that are constrained by the condition $T(x, y) \iff T(x) = T(y)$; under the corresponding axiom (TS) , things turn out to be different.

Proposition 5.7. $T(a(x+y)+ax+a(y+w)) = T(ax) \cup T(ay) \cup T(aw) = T(ax+a(y+w))$.

As a consequence, for $(T\text{-}ND_{+}^Z)$ and $(T\text{-}ND_{\equiv})$ we can apply the same arguments used in Section 5.1 to show that (ND^Z) was satisfied by $\leq_I^{l(Z)}$.

Proposition 5.8. For $Z \in \{RT, FT, R, F\}$, the preorder $\leq_T^{l(Z)}$ satisfies the axiom $(T\text{-}ND_{+}^Z)$ and also $(T\text{-}ND_{\equiv})$.

Proof. To show that $\leq_T^{l(Z)}$ satisfies $(T\text{-}ND_{+}^Z)$ we just need to apply Proposition 5.7 and follow the line of thought in the second bullet on page 37, substituting the observer T for I . For the other axiom, from $T(a(x+y)) \subseteq T(ax+a(y+w))$ it follows that $\{(TS)\} \vdash ax+a(y+w) \preceq (ax+a(y+w)) + a(x+y)$. \square

Notice that for $Z \in \{RT, FT\}$ we can also obtain the correctness of $(T\text{-}ND_{\equiv}^Z)$ from that of $(T\text{-}ND^Z)$ and vice versa, as a consequence of the following fact.

Proposition 5.9. *The axiomatization $\{B_1\text{-}B_4, (TS), (T\text{-}ND^Z)\}$ is equivalent to the axiomatization $\{B_1\text{-}B_4, (TS), (T\text{-}ND_{\equiv}^Z)\}$ for $Z \in \{RT, FT\}$.*

Proof. Let us first show that $\{B_1\text{-}B_4, (TS), (T\text{-}ND^Z)\}$ is equivalent to $\{B_1\text{-}B_4, (TS), (T\text{-}ND_{+}^Z)\}$. This holds because $(T\text{-}ND^Z)$ implies $(T\text{-}ND_{+}^Z)$ and, since $T(w) \subseteq T(y)$ implies $T(a(x+y)) = T(ax + a(y+w))$ and then we have $\{B_1\text{-}B_4, (TS)\} \vdash a(x+y) \preceq a(x+y) + (ax + a(y+w))$.

To prove $\{B_1\text{-}B_4, (TS), (T\text{-}ND_{+}^Z)\}$ equivalent to $\{B_1\text{-}B_4, (TS), (T\text{-}ND_{\equiv}^Z)\}$ we only need to show that $\{B_1\text{-}B_4, (TS), (T\text{-}ND_{+}^Z)\} \vdash (M_Z^T(x, y, w) \Rightarrow ax + a(y+w) \preceq ax + a(y+w) + a(x+y))$, but we have that for $Z \in \{RT, FT\}$, $M_Z^T(x, y, w)$ implies $T(w) \subseteq T(y)$, so that $T(a(x+y)) = T(ax + a(y+w))$ and therefore $\{(TS)\} \vdash ax + a(y+w) \preceq (ax + a(y+w)) + a(x+y)$. \square

The important fact about the obtained sets of correct axioms for the semantics $\leq_T^{l(Z)}$ is that, although our proofs of completeness for the axiomatizations $\{B_1\text{-}B_4, (RS), (ND^Z)\}$ considered the inequational axioms (ND^Z) , they are also valid for the axiomatizations $\{B_1\text{-}B_4, (RS), (ND_{\equiv}^Z)\}$.

The steps in the procedure that leads to the completeness of $\{B_1\text{-}B_4, (RS), (ND^Z)\}$ can be adapted by substituting each reference to the observer I by T , thus obtaining a proof of the completeness of $\{B_1\text{-}B_4, (RS), (T\text{-}ND_{\equiv}^Z)\}$ for $\leq_T^{l(Z)}$. However, the notion of head normal form for $N = I$ uses the fact that the summands $\text{hnf}^Z(q, a, i, X_1)$ can be defined in terms of the offers $X_1 \subseteq \mathcal{P}(\text{Act})$, which correspond to the values produced by the observer I . For an arbitrary N , a more general definition of hnf's, valid for every observer, is needed.

Definition 5.10. For $p = \sum_{a \in X_0} \sum_{i \in I_a} ap_a^i$, its totally expanded Z -head normal form $\text{tehnf}_N^Z(p)$ is that given by:

- For $a \in X_0$, $i \in I_a$, and $K_a \subseteq I_a$ we consider a decomposition $p_a^k = p_a^{k_1} + p_a^{k_2}$ such that $M_Z^N(p_a^i, p_a^{k_1}, p_a^{k_2})$. Then, $\text{tehnf}_N^Z(p, a, i, \langle (p_a^{k_1}, p_a^{k_2}) \rangle_{k \in K_a}) = a(p_a^i + \sum_{k \in K_a} p_a^{k_1})$.
- $\text{tehnf}_N^Z(p) = \sum \text{tehnf}_N^Z(p, a, i, \langle (p_a^{k_1}, p_a^{k_2}) \rangle_{k \in K_a})$.

It is clear that for $K'_a \subseteq K_a$, or any decomposition $p_a^k = p_a^{k_3} + (p_a^{k_4} + p_a^{k_2})$ with $p_a^{k_1} = p_a^{k_3} + p_a^{k_4}$, the corresponding $\text{tehnf}_N^Z(\dots)$ is a subterm of $\text{tehnf}_N^Z(p, a, i, \langle (p_a^{k_1}, p_a^{k_2}) \rangle_{k \in K_a})$ and thus contributes nothing to the expanded normal form. This is the reason why we preferred the more compact definition of $\text{hnf}^Z(p)$ for semantics coarser than ready simulation.

Theorem 5.11. *For $Z \in \{RT, FT, R, F\}$, $\{B_1\text{-}B_4, (TS), (T\text{-}ND_{\equiv}^Z)\} \vdash p \preceq q$ if and only if $p \leq_T^{l(Z)} q$.*

6. LOGICAL CHARACTERIZATION OF SEMANTICS

The third and a very natural alternative to associate a semantics to processes lies in the logical framework. This is indeed quite a natural way to do it. We have a language to express properties of processes and a way to check whether a process satisfies a formula of the language: then, two processes are equivalent with respect to this semantics if and only if they satisfy the same set of formulas. In fact, the semantics can also be defined in terms of

Formulas \ Semantics (Z)	T	S	CT	CS	F	FT	R	RT	PW	RS	PF	2S	B
$\top \in \mathcal{L}_Z$	•	ν	•	ν	•	•	•	•	ν	ν	ν	ν	ν
$\mathbf{0} \in \mathcal{L}_Z$			•	•	ν	ν	ν	ν	ν	ν	ν	ν	ν
$\varphi \in \mathcal{L}_Z, a \in Act \Rightarrow a\varphi \in \mathcal{L}_Z$	•	•	•	•	•	•	•	•	ν	•	•	•	•
$X \subseteq Act \Rightarrow \tilde{X} \in \mathcal{L}_Z$					•	ν	ν	ν	ν	ν	ν	ν	ν
$X \subseteq Act \Rightarrow X \in \mathcal{L}_Z$							•	ν	•	•	ν	ν	ν
$\varphi \in \mathcal{L}_Z, X \subseteq Act \Rightarrow \tilde{X}\varphi \in \mathcal{L}_Z$						•		ν	ν	ν		ν	ν
$\varphi \in \mathcal{L}_Z, X \subseteq Act \Rightarrow X\varphi \in \mathcal{L}_Z$								•	ν	ν		ν	ν
$\varphi_i \in \mathcal{L}_Z \forall i \in I \Rightarrow \bigwedge_{i \in I} \varphi_i \in \mathcal{L}_Z$		•		•						•		•	•
$X \subseteq Act, \varphi_a \in \mathcal{L}_{PW} \forall a \in X \Rightarrow \bigwedge_{a \in X} a\varphi_a \in \mathcal{L}_Z$									•	ν		ν	ν
$\varphi_i, \varphi_j \in \mathcal{L}_T \forall i \in I \forall j \in J \Rightarrow \bigwedge_{i \in I} \varphi_i \wedge \bigwedge_{j \in J} \neg\varphi_j \in \mathcal{L}_Z$											•	ν	ν
$\varphi \in \mathcal{L}_S \Rightarrow \neg\varphi \in \mathcal{L}_Z$												•	ν
$\varphi \in \mathcal{L}_Z \Rightarrow \neg\varphi \in \mathcal{L}_Z$													•

Table 3: Van Glabbeek’s logical characterizations for the semantics in the ltbt spectrum.

the induced preorder that indicates whether a process satisfies more formulas than another one.

Each subset \mathcal{L} of \mathcal{L}_{HM} induces a semantics as stated in the following definition.

Definition 6.1. Any subset \mathcal{L} of \mathcal{L}_{HM} induces a logical semantics for processes, given by the preorder $\sqsubseteq_{\mathcal{L}}$: $p \sqsubseteq_{\mathcal{L}} q$ whenever for all $\varphi \in \mathcal{L}$, if $p \models \varphi$ then $q \models \varphi$. We say that \mathcal{L} and \mathcal{L}' are equivalent, and we write $\mathcal{L} \sim \mathcal{L}'$, if they induce the same semantics, that is $\sqsubseteq_{\mathcal{L}} = \sqsubseteq_{\mathcal{L}'}$.

Let us start with a look at Table 3, which contains the logical characterization of each of the semantics in van Glabbeek’s spectrum. \mathcal{L}_Z with $Z \in \{T, CT, F, FT, R, RT, PF, S, CS, RS, 2S, PW, B\}$ denotes each of the logics; the dots indicate the clauses needed to obtain the corresponding languages; and the boxes marked with ν correspond to rules that could be added to \mathcal{L}_Z , but would only introduce redundant formulas. The following constructs, which appear in the table but are not in \mathcal{L}_{HM} , can be obtained as syntactic sugar:

$$\begin{aligned} \tilde{X} &:= \bigwedge_{a \in X} \neg a\top & \tilde{X}\varphi' &:= \tilde{X} \wedge \varphi' & 0 &:= \widetilde{Act} \\ \varphi_1 \wedge \varphi_2 &:= \bigwedge_{i \in \{1,2\}} \varphi_i & X &:= \bigwedge_{a \in X} a\top \wedge \bigwedge_{a \notin X} \neg a\top & X\varphi' &:= X \wedge \varphi' & \tilde{a} &:= \neg a\top \end{aligned}$$

Disjunction does not appear in \mathcal{L}_{HM} and therefore neither in any of the logics \mathcal{L}_Z characterizing the semantics in the ltbt spectrum. It is probably folklore that it can be added in all cases without affecting the expressive power of each of these logics, but since we have not found a clear statement in this direction in any of our references, next we establish the result and comment on its proof.

Proposition 6.2. Let us define \mathcal{L}_Z^{\vee} , with $Z \in \{T, CT, F, FT, R, RT, PF, S, CS, RS, 2S, PW, B\}$, by adding the clause $\bigvee_{i \in I} \varphi_i \in \mathcal{L}_Z^{\vee}$ if $\varphi_i \in \mathcal{L}_Z^{\vee}$ for all $i \in I$ to the clauses that define \mathcal{L}_Z ,

replacing \mathcal{L}_Z by \mathcal{L}_Z^\vee in the other clauses, and making $p \models \bigvee \varphi_i$ iff there exists $i \in I$ with $p \models \sigma_i$. Then, $\mathcal{L}_Z^\vee \sim \mathcal{L}_Z$.

Proof. It is interesting to observe that even if the result is valid for all the semantics, the reason behind is not the same as for bisimulation. In that case, we only need to apply the De Morgan laws to get the “definition” of \bigvee as a combination of \neg and \bigwedge . However, for the rest of the semantics we do not have negation as “constructor”, but \bigvee distributes over \bigwedge and the prefix operator (that is $\bigvee a\varphi_i = a\bigvee \varphi_i$), while negation is never applied to a formula $\varphi' \in \mathcal{L}_Z^\vee$. Therefore, by floating to the top any \bigvee , using those distribution laws, a formula in \mathcal{L}_Z^\vee becomes equivalent to a disjunction of formulas within the corresponding language \mathcal{L}_Z , and the equivalence of both logics follows. \square

As we will see in this section, each of our logics is defined by a set of rules and, as usual, only the formulas that can be obtained by finite application of these rules belong to the logics. One important feature of our approach is that instead of focusing on small sets of formulas characterizing each of the semantics, we somewhat follow the opposite approach by including all the formulas, from a certain family, that are preserved by each of the semantics. This choice has many interesting side effects. In particular, we will not need to look for adequate formulas reflecting the characteristics of each of the semantics, but instead pick up from our “repository” of possible formulas those that are preserved by the current semantics. Thus, we characterize each of the semantics by means of the formulas that “see” the kind of observations that define it. As a consequence, we know whether a semantics is coarser than another by checking whether the logic characterizing the former is included in the logic characterizing the latter. Moreover, by using a larger logic we may find a formula expressing some property that is preserved by the corresponding semantics, while if we settle on a smaller logic we might need a collection of formulas to express a simple property.

Formally speaking, for each semantics defined by a preorder \prec we have a language $\mathcal{L} \subseteq \mathcal{L}_{HM}$ characterizing it: $\varphi \in \mathcal{L}$ iff $((p \prec q \wedge p \models \varphi) \text{ implies } q \models \varphi)$. However, it is not easy (nor specially illustrative) to capture the whole set of formulas characterizing the semantics. Instead, we will consider sufficiently large families defined in a simple way that provide natural characterizations of the different semantics and show the relationship between them so that, as stated above, whenever a semantics is finer than another, the logic characterizing the first will contain that for the latter.

As will become clear when we introduce our new logical characterizations, Table 3 readily presents the features that allow us to classify the semantics in the spectrum in four categories:

- Bisimulation semantics, characterized by HML, that is closed under negation (\neg), so that the preorder defined is an equivalence (the bisimulation). The remaining semantics are defined by non-trivial preorders, i.e., the preorders are not equivalences and their logical characterizations are, of course, not closed under negation.
- Simulation semantics (S, CS, RS, ...), characterized by branching observations, which will be reflected by the unrestricted use of the operator \bigwedge in the formulas.
- Linear semantics (T, F, R, ...), characterized by linear observations. We will get them by severely restricting the use of \bigwedge and the use of the negation.
- Deterministic branching semantics, corresponding to an intermediate class between branching and linear semantics, where determinism appears restricting the use of the operator \bigwedge

in combination with the prefix operator. The only semantics in this class in the classical spectrum is PW.

As already happened in Sections 3 and 4, our unified logical semantics will provide an enlarged spectrum—Figure 11. In particular, we will show the logical characterization of revivals semantics, introduced by Roscoe in [48] and already axiomatized in [19].

6.1. A new logical characterization of the most popular semantics. Again, we start with the best known classical semantics, that is, those at the layer of ready simulation in the spectrum. All of them use in some way the set of formulas $\mathcal{L}_I = \{a\top \mid a \in Act\}$ that characterizes the initial offers of a process. In Section 6.2 we will present the logics for the rest of the semantics in a unified way, remarking how they are obtained similarly to those in this section but working from the set \mathcal{L}_N of formulas associated to the corresponding constraint N .

We will prove the equivalence between each of our logics and the corresponding logical characterization defined by van Glabbeek, thus checking that our new logical characterizations are indeed correct. But one of the intended goals of our unification was to obtain direct and natural proofs. This will be illustrated in Section 7 by showing the equivalence between each of our logical semantics and the corresponding observational semantics of Section 4. This will provide a new, single proof of their correctness without having to resort to the characterizations defined by van Glabbeek.

Definition 6.3. *Ready simulation semantics.* We define the set of formulas \mathcal{L}'_{RS} for ready simulation semantics by:

- If $\sigma \in \mathcal{L}_I$ then $\sigma \in \mathcal{L}'_{RS}$;
- if $\sigma \in \mathcal{L}_I$ then $\neg\sigma \in \mathcal{L}'_{RS}$;
- if $\varphi_i \in \mathcal{L}'_{RS}$ for all $i \in I$ then $\bigwedge_{i \in I} \varphi_i \in \mathcal{L}'_{RS}$;
- if $\varphi \in \mathcal{L}'_{RS}$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_{RS}$.

Ready trace semantics. We define the set of formulas \mathcal{L}'_{RT} for ready trace semantics by:

- $\top \in \mathcal{L}'_{RT}$;
- if $\varphi \in \mathcal{L}'_{RT}$ and $X_1, X_2 \subseteq \mathcal{L}'_I$ then $(\bigwedge_{a \in X_1} a\top \wedge \bigwedge_{b \in X_2} \neg b\top) \wedge \varphi \in \mathcal{L}'_{RT}$;
- if $\varphi \in \mathcal{L}'_{RT}$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_{RT}$.

Failure trace semantics. We define the set of formulas \mathcal{L}'_{FT} for failure trace semantics by:

- $\top \in \mathcal{L}'_{FT}$;
- if $\varphi \in \mathcal{L}'_{FT}$ and $X_1 \subseteq \mathcal{L}'_I$ then $(\bigwedge_{a \in X_1} \neg a\top) \wedge \varphi \in \mathcal{L}'_{FT}$;
- if $\varphi \in \mathcal{L}'_{FT}$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_{FT}$.

Readiness semantics. We define the set of formulas \mathcal{L}'_R for readiness semantics by:

- $\top \in \mathcal{L}'_R$;
- if $X_1 \subseteq \mathcal{L}'_I$ and $X_2 \subseteq \mathcal{L}'_I$ then $(\bigwedge_{a \in X_1} a\top \wedge \bigwedge_{b \in X_2} \neg b\top) \in \mathcal{L}'_R$;
- if $\varphi \in \mathcal{L}'_R$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_R$.

Failures semantics. We define the set of formulas \mathcal{L}'_F for failures semantics by:

- $\top \in \mathcal{L}'_F$;
- if $X_1 \subseteq \mathcal{L}'_I$ then $(\bigwedge_{a \in X_1} \neg a\top) \in \mathcal{L}'_F$;
- if $\varphi \in \mathcal{L}'_F$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_F$.

It is immediate that $\mathcal{L}'_{RS} \subseteq \mathcal{L}_B$ and hence ready simulation semantics is coarser than bisimulation equivalence. We also have $\mathcal{L}'_F \subseteq \mathcal{L}'_R$, $\mathcal{L}'_F \subseteq \mathcal{L}'_{FT}$, $\mathcal{L}'_R \subseteq \mathcal{L}'_{RT}$, $\mathcal{L}'_{FT} \subseteq \mathcal{L}'_{RT}$, and $\mathcal{L}'_{RT} \subseteq \mathcal{L}'_{RS}$, which can be interpreted in a similar way. Let us now focus our attention on the third rule of the definition of \mathcal{L}'_{RS} : the unrestricted use of conjunction corresponds to the branching nature of the semantics. Moreover, the two first rules allow to fix the set of offers of a process as I -simulations impose. By contrast, the linear semantics only allow the use of conjunction to join those simple formulas that fix the set of offers along a computation (in the case of the readies-based semantics), or their over-approximations (obtained by means of the negated formulas $\neg a\top$, in the case of the failures-based semantics). Finally, notice how these simple formulas can only be checked at the corresponding final state, for the two simpler coarser semantics.

Now, for $Z \in \{RS, RT, FT, R, F\}$, each of the logics \mathcal{L}'_Z is a superset of the corresponding logic \mathcal{L}_Z defined in Table 3. To be precise, for FT and F we need to remove the syntactic sugar used by van Glabbeek as stated below.

Remark 6.4. We have used in Section 4.2 X^c to denote the complementary of a set, because previously in Definition 4.20 we used the classic over line notation to refer to closures of sets $\mathcal{T} \subseteq LGO_N$. However, since we will not need those closure operators anymore we prefer to used the classic notation referring the complement of a set X by \overline{X} .

Proposition 6.5.

- (1) $\mathcal{L}_{RS} \subsetneq \mathcal{L}'_{RS}$.
- (2) $\mathcal{L}_{RT} \subsetneq \mathcal{L}'_{RT}$.
- (3) $\mathcal{L}'_{FT} \supseteq \text{desugared}(\mathcal{L}_{FT})$, where the *desugaring function* removes the syntactic sugar used in \mathcal{L}_{FT} .
- (4) $\mathcal{L}_R \subsetneq \mathcal{L}'_R$.
- (5) $\mathcal{L}'_F \supseteq \text{desugared}(\mathcal{L}_F)$, where the *desugaring function* removes the syntactic sugar used in \mathcal{L}_F .

Proof. Recall the definition of \mathcal{L}_Z in Table 3.

- (1) To prove that $\mathcal{L}_{RS} \subseteq \mathcal{L}'_{RS}$, it is sufficient to show that each formula $\varphi_X = \bigwedge_{a \in X} a\top \wedge \bigwedge_{b \notin X} \neg b\top$ corresponding to $X \subseteq Act$ belongs to \mathcal{L}'_{RS} . Both $a\top$ and $\neg b\top$ are in \mathcal{L}'_{RS} and the combination of these formulas with the operator \wedge is also in the set \mathcal{L}'_{RS} . For the inclusion to be proper, it is sufficient to notice that the formula $\neg b\top$ belongs to \mathcal{L}'_{RS} but not to the set \mathcal{L}_{RS} .
- (2) To prove that $\mathcal{L}_{RT} \subseteq \mathcal{L}'_{RT}$ it is sufficient to show that for every $X \subseteq Act$ and any $\varphi \in \mathcal{L}_{RT}$, the formula $(\bigwedge_{a \in X} a\top \wedge \bigwedge_{b \notin X} \neg b\top) \wedge \varphi$ belongs to \mathcal{L}'_{RT} . Note that $b \notin X$ is equivalent to $b \in \overline{X}$, so taking $X_1 = X$ and $X_2 = \overline{X}$ we have that the considered formula belongs to \mathcal{L}'_{RT} . To prove that $\mathcal{L}_{RT} \subsetneq \mathcal{L}'_{RT}$, it is sufficient to note that $(\neg b\top) \wedge \varphi$ belongs to \mathcal{L}'_{RT} , by taking $X_1 = \emptyset$ and $X_2 = \{b\}$, but it does not belong to \mathcal{L}_{RS} .
- (3) In this case the result is trivial, since the definitions of \mathcal{L}_{FT} and \mathcal{L}'_{FT} are almost the same, once the syntactic sugar is removed. The only difference is that $\perp \in \mathcal{L}'_{FT}$, which obviously does not affect the inclusion.
- (4) To prove that $\mathcal{L}_R \subseteq \mathcal{L}'_R$, it is sufficient to show that for every $X \subseteq Act$ the formula $\bigwedge_{a \in X} a\top \wedge \bigwedge_{b \notin X} \neg b\top$ belongs to \mathcal{L}'_R . Note that the condition $b \notin X$ is equivalent to $b \in \overline{X}$, so taking $X_1 = X$ and $X_2 = \overline{X}$ we have that the considered formula belongs to \mathcal{L}'_R . To check that $\mathcal{L}_R \subsetneq \mathcal{L}'_R$, it is sufficient to note that the formula $\neg b\top$ belongs to \mathcal{L}'_R by taking $X_1 = \emptyset$ and $X_2 = \{b\}$, while it does not belong to \mathcal{L}_R .

(5) Analogous to 3. □

As stated earlier, in order to obtain more natural characterizations, our logics typically contain large sets of formulas. This is why in most cases our logics contain those proposed by van Glabbeek. In order to prove the equivalence between ours and his, we have to show that our additional formulas are in fact redundant and could be safely removed.

Proposition 6.6. (1) $\mathcal{L}_{RS} \sim \mathcal{L}'_{RS}$; (2) $\mathcal{L}_{RT} \sim \mathcal{L}'_{RT}$; (3) $\mathcal{L}_{FT} \sim \mathcal{L}'_{FT}$; (4) $\mathcal{L}_R \sim \mathcal{L}'_R$; and (5) $\mathcal{L}_F \sim \mathcal{L}'_F$.

Proof.

- (1) Any conjunction and negation of formulas in \mathcal{L}_I can be obtained as the disjunction of the formulas X describing all the “compatible” offers. These are those including the positive and negative information in the corresponding conjunction, i.e., $a\top \sim \bigvee_{a \in X} X$; $\neg a\top \sim \bigvee_{a \notin X} X$. Then, by applying Proposition 6.2, we obtain $\mathcal{L}'_{RS} \sim \mathcal{L}_{RS}$.
- (2) We have shown that the formulas in \mathcal{L}_{RT} are particular cases of the formulas in \mathcal{L}'_{RT} : those that completely define the offers at the states along a computation (when we apply the second clause in the definition of \mathcal{L}'_{RT} with $X_2 = \overline{X_1}$). In contrast, our more general formulas $(\bigwedge_{a\top \in X_1} a\top \wedge \bigwedge_{b\top \in X_2} \neg b\top) \wedge \varphi$, where $\varphi \in \mathcal{L}'_{RT}$, could provide us with some partial information, combining both positive information $a\top \in X_1$ and negative information $b\top \in X_2$, which tells us that we are in an arbitrary state X satisfying $X_1 \subseteq X \subseteq \overline{X_2}$. But we can replace these formulas by the disjunction of all the formulas describing any of these possible offers X . By repeating this procedure at each level of the formula, we finally obtain a disjunction of formulas in \mathcal{L}_{RT} . To conclude, it is enough to apply Proposition 6.2.
- (3) We know $\perp = \neg\top = \bigvee_{i \in \emptyset} \varphi_i$, and applying Proposition 6.2 we get the equivalence.
- (4) Note that van Glabbeek allowed in \mathcal{L}_R only “normal form” formulas from \mathcal{L}'_R , which can give us information about the offers at the final state in a computation (when we apply the second clause in the definition of \mathcal{L}'_R) or simply define these computations by means of the prefix operator (when we apply the third clause in the definition of \mathcal{L}'_R). However, our more general formulas $(\bigwedge_{a\top \in X_1} a\top \wedge \bigwedge_{b\top \in X_2} \neg b\top)$ can also provide us with some partial information about the final state, which could be both positive $a\top \in X_1$ and negative $b\top \in X_2$. In the (allowed) case $X_1 \cap X_2 \neq \emptyset$ we have that the formula is unsatisfiable. Otherwise, we are offering the actions a corresponding to formulas $a\top$ in any $X \subseteq \mathcal{L}_I$ that satisfies $X_1 \subseteq X$ and $X \subseteq \overline{X_2}$, and we can replace again the corresponding formula by a disjunction of formulas in \mathcal{L}_R .
- (5) Analogous to 3. □

In the following, when we consider a logic \mathcal{L}_Z and the index Z refers to some concrete semantics, as is the case with RS , RT , FT , R , and F above, by abuse of notation we will simply write \sqsubseteq'_Z instead of $\sqsubseteq_{\mathcal{L}'_Z}$ for the preorder induced by the logic \mathcal{L}'_Z .

Theorem 6.7.

- (1) The logical semantics \sqsubseteq'_{RS} induced by the logic \mathcal{L}'_{RS} is equivalent to the observational branching semantics defined by \leq^b_I , generated by the set of branching general observations BGO_I .
- (2) For $Z \in \{F, FT, R, RT\}$, the logical semantics \sqsubseteq'_Z induced by the logic \mathcal{L}'_Z is equivalent to the observational linear semantics $\leq^{l(Z)}_I$ in Definitions 4.14 and 4.19.

Proof. It is a consequence of Proposition 6.6 and the results by van Glabbeek collected in Table 3, Theorem 4.9, and Proposition 4.18.

(1) We have already checked that our formulas are equivalent to van Glabbeek's: $\mathcal{L}'_{RS} \sim \mathcal{L}_{RS}$. It is easy to show that once we have eliminated the unsatisfiable formulas in \mathcal{L}'_{RS} (those that simultaneously make two different offers, or perform an action that was not included in the corresponding offer) the remaining formulas in \mathcal{L}'_{RS} admit a normal form in the language $\mathcal{N}(\mathcal{L}_{RS})$, which we define as follows:

- if $X \subseteq Act$, $\{a_i \mid i \in I\} \subseteq X$, and $\varphi_i \in \mathcal{N}(\mathcal{L}_{RS})$, then $(\bigwedge_{b \in X} b\top \wedge \bigwedge_{b \notin X} \neg b\top) \wedge \bigwedge_{i \in I} a_i \varphi_i \in \mathcal{N}(\mathcal{L}_{RS})$;
- if $\{a_i \mid i \in I\} \subseteq Act$ and $\varphi_i \in \mathcal{N}(\mathcal{L}_{RS})$ then $\bigwedge_{i \in I} a_i \varphi_i \in \mathcal{N}(\mathcal{L}_{RS})$.

Within this set, consider the subset of formulas $\mathcal{CN}(\mathcal{L}_{RS})$ which can be generated using the first clause in the above definition. We can establish an isomorphism between $\mathcal{CN}(\mathcal{L}_{RS})$ and the set of possible branching general observations BGO_I . Moreover, it is easy to prove that if for every formula $\varphi \in \mathcal{CN}(\mathcal{L}_{RS})$ we define $bg\varphi$ as the corresponding observation, then $\varphi \models p$ iff $bg\varphi \in BGO_I(p)$, from which it immediately follows that $\mathcal{CN}(\mathcal{L}_{RS})$ characterizes the ready simulation semantics defined via BGO_I .

Now, to conclude the proof it is sufficient to show that $\mathcal{N}(\mathcal{L}_{RS})$ and $\mathcal{CN}(\mathcal{L}_{RS})$ are equivalent. Note that whenever we use the second clause in the definition of $\mathcal{N}(\mathcal{L}_{RS})$, we are ignoring the possibility of specifying the offer X at the state we are. As a consequence, the offer could be any satisfying $\{a_i \mid i \in I\} \subseteq X$, for the corresponding set $\{a_i \mid i \in I\}$. Then we can complete the associated formula $\bigwedge_{i \in I} a_i \varphi_i$ by adding the disjunction $\bigvee_{\{a_i \mid i \in I\} \subseteq X} (\bigwedge_{b \in X} b\top \wedge \bigwedge_{b \notin X} \neg b\top)$. Floating all the disjunctions away we obtain a disjunction of formulas in $\mathcal{N}(\mathcal{L}_{RS})$, which ends the proof.

(2) • If $Z = RT$, we know that $\mathcal{L}'_{RT} \sim \mathcal{L}_{RT}$. It is easy to show that eliminating all the unsatisfiable formulas (those that simultaneously offer two different sets of actions, or perform an action a that is not included in the corresponding offer X) the rest of the formulas in \mathcal{L}'_{RT} admit a normal form in the language $\mathcal{N}(\mathcal{L}_{RT})$, which we define as follows:

- if $X \subseteq Act$ then $(\bigwedge_{b \in X} b\top \wedge \bigwedge_{b \notin X} \neg b\top) \in \mathcal{N}(\mathcal{L}_{RT})$;
- if $X \subseteq Act$, $a \in X$, and $\varphi \in \mathcal{N}(\mathcal{L}_{RT})$ then $(\bigwedge_{b \in X} b\top \wedge \bigwedge_{b \notin X} \neg b\top) \wedge a\varphi \in \mathcal{N}(\mathcal{L}_{RT})$;
- $\top \in \mathcal{N}(\mathcal{L}_{RT})$;
- if $a \in Act$ and $\varphi \in \mathcal{N}(\mathcal{L}_{RT})$ then $a\varphi \in \mathcal{N}(\mathcal{L}_{RT})$.

As we did for the case of ready simulation, we could define the corresponding language of complete formulas $\mathcal{CN}(\mathcal{L}_{RT})$. The formulas in \mathcal{L}'_{RT} that we obtained in the proof of Proposition 6.6, for the case of RT , are indeed in $\mathcal{CN}(\mathcal{L}_{RT})$ because any subformula gives us some partial information about the offers at the corresponding state, which in the worst case could be empty. Therefore, when we translate this information into the language \mathcal{L}'_{RT} we obtain a disjunction between complete formulas in $\mathcal{CN}(\mathcal{L}_{RT})$. We can easily establish the isomorphism between $\mathcal{CN}(\mathcal{L}_{RT})$ and the domain LGO_I , and then prove that for every formula $\varphi \in \mathcal{CN}(\mathcal{L}_{RT})$, if we define $lgo\varphi$ as the corresponding observation, we have $\varphi \models p$ iff $lgo\varphi \in LGO_I(p)$. From here it follows that $\mathcal{CN}(\mathcal{L}_{RT})$ characterizes the ready simulation semantics defined via LGO_I . To conclude the proof we need to show that $\mathcal{N}(\mathcal{L}_{RT})$ and $\mathcal{CN}(\mathcal{L}_{RT})$ are equivalent, which is analogous to $\mathcal{N}(\mathcal{L}_{RS})$ and $\mathcal{CN}(\mathcal{L}_{RS})$ above.

- $Z = FT$. (\Rightarrow) Let p and q be such that $p \sqsubseteq'_{FT} q$: we will show that $p \leq_I^{\perp} q$. Given an observation $X_0 a_1 X_1 \dots a_n X_n \in LGO_I(p)$, we have a failure trace $\overline{X_0 a_1 X_1 \dots a_n X_n}$

for the process p . Now, we consider the formulas $\varphi_n = \bigwedge_{a \in \overline{X}} \neg a \top$, $\varphi_i = \bigwedge_{a \in \overline{X_i}} \neg a \top \wedge a_{i+1} \varphi_{i+1}$ with $i \in 0..n-1$, and we have that $p \models \varphi_0$. Therefore $q \models \varphi_0$, which means that $\overline{X_0 a_1 \overline{X_1} \dots a_n \overline{X_n}}$ is a failure trace of q . Then, there is some $Y_0 a_1 Y_2 \dots a_n Y_n \in LGO_I(p)$ with $Y_i \cap \overline{X_i} = \emptyset$ for all $i = 0..n$ or, equivalently, $X_i \supseteq Y_i$ for all $i = 0..n$. As a result, $LGO_I(p) \leq_I^{l\supseteq} LGO_I(q)$, which means $p \leq_I^{l\supseteq} q$.

(\Leftarrow) Let us suppose that for all $X_0 a_1 X_1 \dots a_n X_n \in LGO_I(p)$ there exists $Y_0 a_1 Y_1 \dots a_n Y_n \in LGO_I(q)$ such that $X_i \supseteq Y_i$ for all $i = 0..n$; we want to show that if $p \models \varphi$ then $q \models \varphi$, for all $\varphi \in \mathcal{L}'_{FT}$. If $p \models \varphi$, we can decompose φ by means of a sequence of formulas, taking $\varphi = \varphi_n$, $\varphi_i = \bigwedge_{a \in X_i^0} \neg a \top \wedge a_i \varphi_{i-1}$ for $i \in 1..n$ and $\varphi_0 = \bigwedge_{a \in X_2^0} \neg a \top$. Therefore, $X_n a_n X_{n-1} \dots a_1 X_0$ is a failure trace for the process p , so there exists $Z_n a_n Z_{n-1} \dots a_1 Z_0 \in LGO_I(p)$ with $Z_i \cap X_i = \emptyset$, and using that $p \leq_I^{l\supseteq} q$, there exists some $Y_n a_n Y_{n-1} \dots a_1 Y_0 \in LGO_I(q)$ with $Y_i \subseteq Z_i$, so that $Y_i \cap X_i = \emptyset$ and then we get $q \models \varphi_n$.

- If $Z = R$, using the result in the proof of Proposition 6.6 for the case of R it is enough to show the result for the set of “normal form” formulas $\mathcal{N}(\mathcal{L}_{\mathcal{R}})$ defined by:
 - if $X \subseteq Act$ then $(\bigwedge_{b \in X} b \top \wedge \bigwedge_{b \notin X} \neg b \top) \in \mathcal{N}(\mathcal{L}_{\mathcal{R}})$;
 - $\top \in \mathcal{N}(\mathcal{L}_{\mathcal{R}})$;
 - $a \in Act$ and $\varphi \in \mathcal{N}(\mathcal{L}_{\mathcal{R}})$ then $\varphi \in \mathcal{N}(\mathcal{L}_{\mathcal{R}})$.

(\Rightarrow) Let p and q be such that $p \sqsubseteq'_R q$: we will show $p \leq_I^{lf} q$. Given an observation $X_0 a_1 X_1 \dots a_n X_n \in LGO_I(p)$, it corresponds to the readiness information $a_1 \dots a_n X_n$ of p . Now, we consider the formulas $\varphi_n = \bigwedge_{a \in X} a \top \wedge \bigwedge_{a \notin X} \neg a \top$; $\varphi_{i-1} = a_i \varphi_i$ with $i \in 1 \dots n-1$, and we have that $p \models \varphi_0$. Therefore $q \models \varphi_0$, and $a_1 \dots a_n X_n$ is a readiness information of q and, as a consequence, there is an observation $Y_0 a_1 Y_2 \dots a_n Y_n \in LGO_I(q)$ with $Y_n = X_n$, proving $p \leq_I^{lf} q$.

(\Leftarrow) Let us suppose that for all $X_0 a_1 X_1 \dots a_n X_n \in LGO_I(p)$ there exists some $Y_0 a_1 Y_1 \dots a_n Y_n \in LGO_I(q)$ such that $X_n = Y_n$. We want to show that if $p \models \varphi$ then $q \models \varphi$ for all $\varphi \in \mathcal{CN}(\mathcal{L}_{\mathcal{R}})$. If $p \models \varphi$, we can decompose φ taking $\varphi = \varphi_n$, $\varphi_i = a_i \varphi_{i-1}$, for all $i \in 1..n$, and $\varphi_0 = \bigwedge_{a \in X_0} a \top \wedge \bigwedge_{a \notin X_0} \neg a \top$. Then we have that $a_n a_{n-1} \dots a_1 X_0$ is a readiness information of p , so there exists some $Z_n a_n Z_{n-1} \dots a_1 X_0 \in LGO_I(p)$, and some $Y_n a_n Y_{n-1} \dots a_1 Y_0 \in LGO_I(q)$ with $Y_0 = X_0$, from which we conclude that $q \models \varphi_n$.

- $Z = F$. (\Rightarrow) Let p and q be such that $p \sqsubseteq'_F q$: we will show $p \leq_I^{lf\supseteq} q$. Given an observation $X_0 a_1 X_1 \dots a_n X_n \in LGO_I(p)$, it generates a (maximal) failure $a_1 \dots a_n \overline{X_n}$ of the process p . Now, we consider the formulas $\varphi_n = \bigwedge_{a \in \overline{X}} \neg a \top$; $\varphi_{i+1} = a_{i+1} \varphi_i$ with $i \in 0..n-1$, and we have that $p \models \varphi_0$. Therefore, $q \models \varphi_0$, so $a_1 \dots a_n \overline{X_n}$ is a failure information of q , and there is some $Y_0 a_1 Y_2 \dots a_n Y_n \in LGO_I(q)$ with $Y_n \cap \overline{X_n} = \emptyset$, or equivalently $X_n \supseteq Y_n$, proving that $p \leq_I^{lf\supseteq} q$.

(\Leftarrow) Let us suppose that for all $X_0 a_1 X_1 \dots a_n X_n \in LGO_I(p)$ there exists some $Y_0 a_1 Y_1 \dots a_n Y_n \in LGO_I(q)$ such that $X_n \supseteq Y_n$. We want to show that if $p \models \varphi$ then $q \models \varphi$ for all $\varphi \in \mathcal{L}'_F$. If $p \models \varphi$, we can decompose φ taking $\varphi = \varphi_n$, $\varphi_i = a_i \varphi_{i-1}$, with $i \in 1..n$, and $\varphi_0 = \bigwedge_{a \in X_0} \neg a \top$. From $p \models \varphi$ we infer that $a_n a_{n-1} \dots a_1 X_0$ is a failure information of the process p , so there exists $Z_n a_n Z_{n-1} \dots a_1 Z_0 \in LGO_I(p)$ with $Z_0 \cap X_0 = \emptyset$, and then there is some $Y_n a_n Y_{n-1} \dots a_1 Y_0 \in LGO_I(q)$ with $Y_n \subseteq Z_n$, so that $Y_n \cap X_n = \emptyset$, obtaining $q \models \varphi_n$.

□

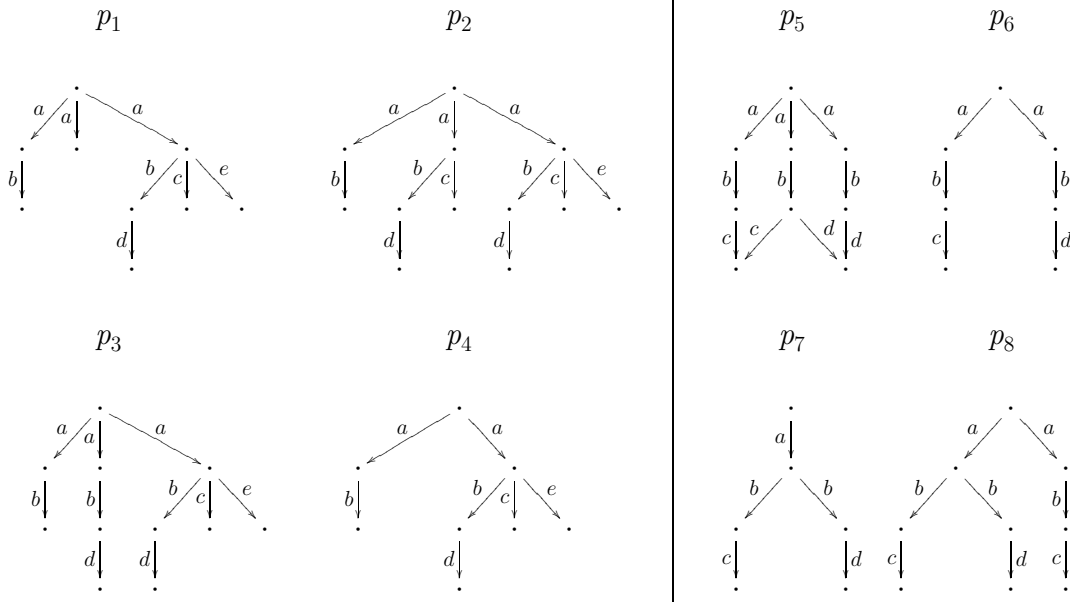


Figure 13: A simple example to show the strength of the different logics

Example 6.8. Figure 13 shows a collection of examples to illustrate the differences between the semantics in the RS layer of the spectrum. All the following equivalences can be checked by taking any arbitrary formula from the logic defining each of the semantics. For readability, we omit the last \top in all subformulas. Besides, \sim_X (resp. \approx_X), where X is a set of indexes, represents any \sim_Z (resp. \approx_Z), with $Z \in X$.

- $p_1 \not\sqsubseteq'_F p_2$ and $p_1 \not\sqsubseteq'_{\{R, FT, RT, RS\}} p_2$ because $p_1 \models a(\neg b \wedge \neg c)$, but p_2 does not satisfy it.
- $p_2 \sim_F p_3$, but $p_2 \not\sqsubseteq'_{\{R, FT\}} p_3$ and thus $p_2 \not\sqsubseteq'_{\{RT, RS\}} p_3$, since p_2 satisfies $a(\neg e \wedge c)$ but p_3 does not.
- $p_3 \sim_{\{F, R\}} p_4$, but $p_3 \not\sqsubseteq'_{FT} p_4$ and thus $p_3 \not\sqsubseteq'_{\{RT, RS\}} p_4$, because p_3 satisfies $a(\neg c \wedge b(\neg e \wedge d))$ but p_4 does not.
- $p_5 \sim_{\{F, FT\}} p_6$, but $p_5 \not\sqsubseteq'_R p_6$ and thus $p_5 \not\sqsubseteq'_{\{RT, RS\}} p_6$, since p_5 satisfies $ab(c \wedge d)$ but p_6 does not.
- $p_6 \sim_{\{F, R, RT, FT\}} p_7$ but $p_7 \not\sqsubseteq'_{RS} p_6$, because p_7 satisfies $a(bc \wedge bd)$ but p_6 does not.
- $p_7 \sim_{\{F, R, RT, FT, RS\}} p_8$.

6.2. Our new unified logical characterizations of the semantics. Inspired by the semantics studied in Section 6.1, next we define the general format for the logics characterizing each of the semantics in the spectrum. We start by enlarging the spectrum yet a bit more.

Definition 6.9.

- (1) *Universal semantics.* We define the set \mathcal{L}'_U of universal formulas that characterize the trivial semantics that identifies all the processes by $\mathcal{L}'_U = \{\top\}$.
- (2) *Complete semantics.* We define the set \mathcal{L}'_C of complete formulas characterizing the semantics that only distinguishes the terminated processes from the non-terminated ones by $\mathcal{L}'_C = \{\top, \neg 0\}$.

- (3) *Initial offer semantics.* We define the set \mathcal{L}'_I of initial offer formulas characterizing the semantics that only observers the set of initial actions of a process by $\mathcal{L}'_I = \{\top, \neg 0\} \cup \{a\top \mid a \in \text{Act}\}$.

In the definition above the subformula $\neg 0$ is just syntactic sugar for the formula $\neg(\bigwedge_{a \in \text{Act}} \neg a\top)$. Therefore, once again all these new logics are sublogics of \mathcal{L}_{HM} and, as a result, we do not need to define their semantics.

Note that \mathcal{L}'_I is a bit larger than the logic \mathcal{L}_I from Section 6.1. Once again, this is so in order to get a more uniform presentation of our logics: $\neg 0$ is indeed redundant. By including it we immediately obtain that the complete semantics is coarser than the initial offer semantics, because $\mathcal{L}'_C \subseteq \mathcal{L}'_I$. Based on this result we will also obtain that the complete simulation is coarser than the ready simulation. Certainly, $\neg 0$ is redundant in \mathcal{L}'_I (but not in \mathcal{L}'_C !), because by means of it we can only distinguish a process that cannot execute any action from any other that can execute someone. But using the corresponding $a\top$ formula we can also get that.

6.2.1. *The simulation semantics.* As repeatedly noted, the family of simulation semantics constitute the spine of the new spectrum. All of them are defined in a homogeneous way thanks to the notion of constrained simulation from [24]. Next we present their logical characterization.

Definition 6.10. Given a set of formulas \mathcal{L}'_N defining a semantics N , we define the set of formulas \mathcal{L}'_{NS} that characterizes the N -constrained simulation semantics by:

- If $\sigma \in \mathcal{L}'_N$ then $\sigma \in \mathcal{L}'_{NS}$;
- if $\sigma \in \mathcal{L}'_N$ then $\neg\sigma \in \mathcal{L}'_{NS}$;
- if $\varphi_i \in \mathcal{L}'_{NS}$ for all $i \in I$ then $\bigwedge_{i \in I} \varphi_i \in \mathcal{L}'_{NS}$;
- if $\varphi \in \mathcal{L}'_{NS}$ and $a \in \text{Act}$ then $a\varphi \in \mathcal{L}'_{NS}$.

Taking $N \in \{U, C, I\}$ we obtain \mathcal{L}'_{US} , \mathcal{L}'_{CS} and \mathcal{L}'_{IS} , that we rewrite as \mathcal{L}'_S and \mathcal{L}'_{RS} in the first and last cases to emphasize the classic notation for simulation semantics. From \mathcal{L}'_S we obtain \mathcal{L}'_{SS} , that we will denote as \mathcal{L}'_{2S} . To complete the collection of simulation semantics in the spectrum we need \mathcal{L}'_{TS} , that will be based on \mathcal{L}'_T , to be defined in the next section.

The definition above differs from the particular case of ready simulation in Definition 6.3 in the two first rules, by means of which we impose that the process will traverse states which are in the corresponding N -equivalence class all along the tree of computations checked by a formula in \mathcal{L}'_{NS} . Note that the combination of positive and negated formulas allows us to shape each of these classes. Next we state the equivalence between our logics for the simulation semantics and those by van Glabbeek's recalled in Table 3.

Proposition 6.11. (1) $\mathcal{L}'_S \sim \mathcal{L}_S$; (2) $\mathcal{L}'_{CS} \sim \mathcal{L}_{CS}$; and (3) $\mathcal{L}'_{2S} \sim \mathcal{L}_{2S}$.

Proof.

- (1) The clauses defining \mathcal{L}'_S and \mathcal{L}_S produce the same set of formulas. The first two clauses in \mathcal{L}'_S only add the two trivial formulas \top and $\neg\top$ because in $\mathcal{L}'_U = \{\top\}$.
- (2) Again, the sets of formulas produced by \mathcal{L}'_{CS} and \mathcal{L}_{CS} are the same because the two first clauses of \mathcal{L}'_{CS} can only generate \top , $\neg\top$, 0 and $\neg 0$ from $\mathcal{L}'_C = \{\top, \neg\top\}$. 0 is needed to reflect the second clause in the definition of \mathcal{L}_{CS} , while $\neg 0 \equiv \bigvee_{a \in \text{Act}} a\top$ so that any formula containing $\neg 0$ can be rewritten into a disjunction of formulas in \mathcal{L}_{CS} .

- (3) Once again, the sets generated by \mathcal{L}'_{2S} and \mathcal{L}_{2S} are the same. The clause “if $\sigma \in \mathcal{L}'_S$ then $\sigma \in \mathcal{L}'_{2S}$ ” in \mathcal{L}'_{2S} does not generate any new formulas because $\mathcal{L}_S \subseteq \mathcal{L}_{2S}$ (the formulas in \mathcal{L}_S are exactly those that can be created using only the last two clauses in the definition of \mathcal{L}_{2S}).

□

Remark 6.12. We can use both positive formulas in \mathcal{L}'_C and their negations for defining \mathcal{L}'_{CS} due to the fact that C -constrained simulation can be built from the equivalence relation defined by C as constraint. However, we could also use \sqsubseteq_C as a constraint and then remove the clause “if $\sigma \in \mathcal{L}'_C$ then $\sigma \in \mathcal{L}'_{SC}$ ”, which generates $\neg 0 \in \mathcal{L}_{SC}$. The other clause, which generates $0 \in \mathcal{L}'_{SC}$, is crucial and cannot be removed from the definition. These two facts also concur in the definition of the other simulation semantics in the extended spectrum, for which we also present a logical characterization including the two clauses above.

6.2.2. *Logical characterization of the linear semantics.* We start by defining the closure operators by means of which we express the extent to which conjunction and negation can be used in the logical characterizations of each of the linear semantics.

Definition 6.13. Given a logical set \mathcal{L}'_N with $N \in \{U, C, I, T, S\}$, we define:

- (1) Its *symmetric closure* $\mathcal{L}'_{N^{\equiv}}$ by: if $\sigma \in \mathcal{L}'_N$ then $\sigma \in \mathcal{L}'_{N^{\equiv}}$ and $\neg\sigma \in \mathcal{L}'_{N^{\equiv}}$; if $\sigma_i \in \mathcal{L}'_{N^{\equiv}}$ for all $i \in I$ then $\bigwedge_{i \in I} \sigma_i \in \mathcal{L}'_{N^{\equiv}}$.
- (2) Its *negative closure* $\mathcal{L}'_{N^{\neg}}$ by: if $\sigma \in \mathcal{L}'_N$ then $\neg\sigma \in \mathcal{L}'_{N^{\neg}}$; if $\sigma_i \in \mathcal{L}'_{N^{\neg}}$ for all $i \in I$ then $\bigwedge_{i \in I} \sigma_i \in \mathcal{L}'_{N^{\neg}}$.
- (3) Its *positive closure* $\mathcal{L}'_{N^{\vee}}$ by: if $\sigma \in \mathcal{L}'_N$ then $\sigma \in \mathcal{L}'_{N^{\vee}}$; if $\sigma_i \in \mathcal{L}'_{N^{\vee}}$ for all $i \in I$ then $\bigwedge_{i \in I} \sigma_i \in \mathcal{L}'_{N^{\vee}}$.

Remark 6.14. Obviously these closures make sense for any given logic \mathcal{L} , but we prefer to restrict our attention to \mathcal{L}'_N since it will be enough for our goal and gives rise to a simpler notation.

Whenever we have a bag of “good” properties (such as \mathcal{L}'_N above), to assert by means of a single formula which is the subset of properties that a certain element satisfies it is not enough to assert that it satisfies each of them: we also need to assert that it does not satisfy any of the rest. This is why we need formulas in the symmetric closure. By contrast, if the only available formulas belong to the negative (resp. positive) closure, we can only assert that the element has at most (resp. at least) the enumerated properties. Next we present the unified logics for all the linear semantics in the spectrum.

Definition 6.15. Inspired by the orders \leq^l_N , $\leq^{l\supset}_N$, \leq^{lf}_N , and $\leq^{lf\supset}_N$, we define the set of formulas $\mathcal{L}'_{\leq^l_N}$, $\mathcal{L}'_{\leq^{l\supset}_N}$, $\mathcal{L}'_{\leq^{lf}_N}$, and $\mathcal{L}'_{\leq^{lf\supset}_N}$, respectively, by means of the rules:

- (1) • $\top \in \mathcal{L}'_{\leq^l_N}$;
 • if $\varphi \in \mathcal{L}'_{\leq^l_N}$ and $\sigma \in \mathcal{L}'_{N^{\equiv}}$ then $\sigma \wedge \varphi \in \mathcal{L}'_{\leq^l_N}$;
 • if $\varphi \in \mathcal{L}'_{\leq^l_N}$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_{\leq^l_N}$.
- (2) • $\top \in \mathcal{L}'_{\leq^{l\supset}_N}$;
 • if $\varphi \in \mathcal{L}'_{\leq^{l\supset}_N}$ and $\sigma \in \mathcal{L}'_{N^{\neg}}$ then $\sigma \wedge \varphi \in \mathcal{L}'_{\leq^{l\supset}_N}$;

- if $\varphi \in \mathcal{L}'_{\leq_N^{I\supseteq}}$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_{\leq_N^{I\supseteq}}$.
- (3)
- $\top \in \mathcal{L}'_{\leq_N^{I\subseteq}}$;
 - if $\sigma \in \mathcal{L}'_{\leq_N^{\equiv}}$ then $\sigma \in \mathcal{L}'_{\leq_N^{I\subseteq}}$;
 - if $\varphi \in \mathcal{L}'_{\leq_N^{I\subseteq}}$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_{\leq_N^{I\subseteq}}$.
- (4)
- $\top \in \mathcal{L}'_{\leq_N^{I\supseteq}}$;
 - if $\sigma \in \mathcal{L}'_{\leq_N^{\neg}}$ then $\sigma \in \mathcal{L}'_{\leq_N^{I\supseteq}}$;
 - if $\varphi \in \mathcal{L}'_{\leq_N^{I\supseteq}}$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_{\leq_N^{I\supseteq}}$.

Note that for the coarsest semantics (i.e. those corresponding to plain refusals and plain readiness when $N = I$) we only check for N at the “end” of the formula because there are no conjunctions in the corresponding languages $\mathcal{L}'_{\leq_N^{I\subseteq}}$ and $\mathcal{L}'_{\leq_N^{I\supseteq}}$, except for those stemming from the corresponding closures $\mathcal{L}'_{\leq_N^{\equiv}}$ and $\mathcal{L}'_{\leq_N^{\neg}}$. The other two logics do introduce additional conjunctions that allow to observe N along the computations.

We have used the negative and symmetric closures for the “failures-based” and “readies-based” semantics, and we can use the positive closure to define two new semantics that have not been considered earlier in this paper, nor elsewhere as far as we know. For that we need to observe partial offers along a computation, or just at its end, where X is a partial offer of p if $X \subseteq I(p)$. It is clear the duality with respect to the failures semantics, where F is a failure of p if $I(p) \subseteq \overline{F}$. We can introduce these two new semantics at each layer of the spectrum through the corresponding partial offers for each $N \in \{U, C, I, T, S\}$.

Definition 6.16.

- (1) The semantics of *partial offer traces* for the constraint N is that defined by the logic $\mathcal{L}'_{\leq_N^{I\subseteq}}$ with:
- $\top \in \mathcal{L}'_{\leq_N^{I\subseteq}}$;
 - if $\varphi \in \mathcal{L}'_{\leq_N^{I\subseteq}}$ and $\sigma \in \mathcal{L}'_{\leq_N^{\vee}}$ then $\sigma \wedge \varphi \in \mathcal{L}'_{\leq_N^{I\subseteq}}$;
 - if $\varphi \in \mathcal{L}'_{\leq_N^{I\subseteq}}$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_{\leq_N^{I\subseteq}}$.
- (2) The semantics of *partial offers* for the constraint N is that defined by the logic $\mathcal{L}'_{\leq_N^{I\subseteq}}$ with:
- $\top \in \mathcal{L}'_{\leq_N^{I\subseteq}}$;
 - if $\sigma \in \mathcal{L}'_{\leq_N^{\vee}}$ then $\sigma \in \mathcal{L}'_{\leq_N^{I\subseteq}}$;
 - if $\varphi \in \mathcal{L}'_{\leq_N^{I\subseteq}}$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_{\leq_N^{I\subseteq}}$.

Duality between failures and partial offers causes the picture of the complete layer of linear semantics for each N to become two diamonds that share the side corresponding to the readies-based semantics. Now, recalling Theorem 6.7.

Proposition 6.17.

- (1) \mathcal{L}'_F and $\mathcal{L}'_{\leq_I^{I\subseteq}}$ are incomparable: $p \leq_I^{I\supseteq} q$ does not imply $p \leq_I^{I\subseteq} q$ and $p \leq_I^{I\subseteq} q$ does not imply $p \leq_I^{I\supseteq} q$.

- (2) \mathcal{L}'_{FT} and $\mathcal{L}'_{\leq_I^{\subseteq}}$ are incomparable: $p \leq_I^{\supseteq} q$ does not imply $p \leq_I^{\subseteq} q$ and $p \leq_I^{\subseteq} q$ does not imply $p \leq_I^{\supseteq} q$.

Proof. In fact, we have a stronger result by combining these two statements: if we consider $p = ab + ac$, $q = a(b + c)$, and $r = p + q$, then $p =_I^{\supseteq} r$ but $r \not\leq_I^{\subseteq} p$, and $q =_I^{\subseteq} r$ but $r \not\leq_I^{\supseteq} q$. \square

Similar counterexamples exist for $N \in \{T, S\}$. However, for $N \in \{U, C\}$, which produce the trace and the completed trace semantics, respectively, it is easy to prove that the six logics of the layer are equivalent.

Proposition 6.18.

- (1) $\mathcal{L}'_{\leq_U^{\supseteq}} = \mathcal{L}'_{\leq_U^{\subseteq}} = \mathcal{L}'_{\leq_U^{\supseteq}} = \mathcal{L}'_{\leq_U^{\subseteq}} = \mathcal{L}'_{\leq_U^{\supseteq}} = \mathcal{L}'_{\leq_U^{\subseteq}} = \mathcal{L}_T$
(2) $\mathcal{L}'_{\leq_C^{\supseteq}} = \mathcal{L}'_{\leq_C^{\subseteq}} = \mathcal{L}'_{\leq_C^{\supseteq}} = \mathcal{L}'_{\leq_C^{\subseteq}} = \mathcal{L}'_{\leq_C^{\supseteq}} = \mathcal{L}'_{\leq_C^{\subseteq}} = \mathcal{L}_{CT}$.

Proof.

- (1) Trivial, since the sets of clauses defining $\mathcal{L}'_{\leq_U^{\supseteq}}$ and \mathcal{L}_T are almost the same. Note that the clause “if $\sigma \in \mathcal{L}_U^{\subseteq}$ then $\sigma \in \mathcal{L}'_{\leq_U^{\supseteq}}$ ” does not give rise to new formulas because $\mathcal{L}_U^{\subseteq} = \{\top\}$.
(2) Note that the sets of clauses defining $\mathcal{L}'_{\leq_C^{\supseteq}}$ and \mathcal{L}_{CT} are the same but for the clause “if $\sigma \in \mathcal{L}_C^{\supseteq}$ then $\sigma \in \mathcal{L}'_{\leq_C^{\supseteq}}$ ”. On the one hand, this causes $\neg\top \in \mathcal{L}'_{\leq_C^{\supseteq}}$ (which adds nothing) because $\top \in \mathcal{L}_C^{\supseteq}$ and thus $\neg\top \in \mathcal{L}_C^{\supseteq}$. On the other hand, we also have $0 \in \mathcal{L}'_{\leq_C^{\supseteq}}$ because $\neg 0 \in \mathcal{L}_C^{\supseteq}$ and then $\neg\neg 0 \in \mathcal{L}_C^{\supseteq}$. \square

Corollary 6.19. $\mathcal{L}'_{\leq_U^{\supseteq}} \sim \mathcal{L}_T$ and $\mathcal{L}'_{\leq_C^{\supseteq}} \sim \mathcal{L}_{CT}$.

An interesting result illustrating the generality of our characterizations concerns one of the finest semantics in the classic spectrum: possible futures. Possible futures is located in Figure 1 below 2-nested simulation because the more accurate trace simulation semantics was not yet included in the spectrum; this is corrected in the spectrum in Figure 11. Indeed, for $N = T$ we have the following result.

Proposition 6.20. $\mathcal{L}'_{\leq_T^{\supseteq}} = \mathcal{L}_{PF}$.

Proof. Trivial, since the sets of clauses defining $\mathcal{L}'_{\leq_T^{\supseteq}}$ and \mathcal{L}_{PF} are almost the same: our definition includes the clause “ $\top \in \mathcal{L}'_{\leq_T^{\supseteq}}$ ”, which does not appear explicitly in that of \mathcal{L}_{PF} because it corresponds to the conjunction of an empty set of formulas. \square

Corollary 6.21. $\mathcal{L}'_{\leq_T^{\supseteq}} \sim \mathcal{L}_{PF}$.

6.2.3. *Logical characterization of the deterministic branching semantics.* Now we consider the deterministic branching semantics. In the classic spectrum the only such semantics is possible worlds but, as we pointed out before, there is one such semantics at each layer of the extended spectrum.

Formulas \ Constraints (\mathcal{N})	U	C	I	T	S	B
$\top \in \mathcal{L}'_{\mathcal{N}}$	•	•	•	•	ν	ν
$\neg\top = \perp \in \mathcal{L}'_{\mathcal{N}}$	ν	ν	ν	ν	ν	ν
$\neg 0 \in \mathcal{L}'_{\mathcal{N}}$		•	•	ν	ν	ν
$a \in Act \Rightarrow a\top \in \mathcal{L}'_{\mathcal{N}}$			•	ν	ν	ν
$\varphi \in \mathcal{L}'_{\mathcal{N}}, a \in Act \Rightarrow$ $a\varphi \in \mathcal{L}'_{\mathcal{N}}$				•	•	•
$\varphi_i \in \mathcal{L}'_{\mathcal{N}} \forall i \in I \Rightarrow$ $\bigwedge_{i \in I} \varphi_i \in \mathcal{L}'_{\mathcal{N}}$					•	•
$\varphi \in \mathcal{L}'_{\mathcal{N}} \Rightarrow$ $\neg\varphi \in \mathcal{L}'_{\mathcal{N}}$						•

Table 4: Logical characterizations of the semantics used as constraints.

In order to capture determinism we need to consider conjunctive formulas to express the desired branching, but only when it corresponds to a choice between different actions. This leads us to the following scheme:

$$\text{if } X \subseteq Act \text{ and } \varphi_a \in \mathcal{L}_{D_N} \text{ for all } a \in X, \text{ then } \bigwedge_{a \in X} a\varphi_a \in \mathcal{L}_{D_N}.$$

Definition 6.22. For each $N \in \{U, C, I, T, S\}$, we define the formulas of \mathcal{L}'_{D_N} by:

- $\top \in \mathcal{L}'_{D_N}$;
- if $\varphi \in \mathcal{L}'_{D_N}$ and $\sigma \in \mathcal{L}_N^{\equiv}$ then $\sigma \wedge \varphi \in \mathcal{L}'_{D_N}$;
- if $X \subseteq Act$ and $\varphi_a \in \mathcal{L}'_{D_N}$ for all $a \in X$ then $\bigwedge_{a \in X} a\varphi_a \in \mathcal{L}'_{D_N}$.

For $N = I$ we obtain the unified logical characterization of the possible worlds semantics.

Proposition 6.23. $\mathcal{L}'_{D_I} \supseteq \mathcal{L}_{PW}$.

Proof. Analogous to the case of ready simulation semantics. \square

Proposition 6.24. $\mathcal{L}'_{D_I} \approx \mathcal{L}_{PW}$.

Proof. This is a consequence of the fact that the original logical characterization of the possible worlds semantics, \mathcal{L}_{PW} , was wrong. For instance, taking $p = abc + a(bc + d) + ab$ and $q = a(bc + d) + ab$ then $p \not\equiv_{PW} q$ but $p \sim_{\mathcal{L}_{PW}} q$, since \mathcal{L}_{PW} cannot “observe” the intermediate offer that makes the possible world abc different from those of q . By contrast, the formula $\varphi = a(\neg d \wedge bc) \in \mathcal{L}'_{D_I}$ is enough to distinguish p and q , since $p \models \varphi$ and $q \not\models \varphi$. \square

We postpone to Section 7 the proof of the equivalence between our observational and logical characterizations of the possible worlds semantics. As a consequence of this correspondence, we have that a logical characterization only works in the infinite case if we restrict ourselves to image-finite processes.

In Tables 4 and 5 we present our results in a three-dimensional way. Table 5 shows the rules defining the logics characterizing each of the semantics at each layer of the spectrum. On top of it also appears, as example, the classic notation for the corresponding semantics represented when $N = I$. Table 4 contains the logics that characterize the constraint governing each of these layers. There are two semantics that are included in both tables, in order to emphasize their double role as “main” and “auxiliary” semantics. However they are disguised under different names: this is the case of $T = \leq_U^I$ (in fact, it is also equal to the other three linear U -semantics) and $S = US$.

Semantics (\mathcal{Y}_N) Formulas	$\leq_N^{lf\supseteq}$	\leq_N^{lf}	$\leq_N^{l\supseteq}$	\leq_N^l	D_N	NS	$N \in \{U, C, I, T, S\}$ when $N = I$
	F	R	FT	RT	PW	RS	
$\top \in \mathcal{L}'_{\mathcal{Y}_N}$	•	•	•	•	•	ν	
$\varphi \in \mathcal{L}'_{\mathcal{Y}_N}, a \in Act \Rightarrow$ $a\varphi \in \mathcal{L}'_{\mathcal{Y}_N}$	•	•	•	•	ν	•	
$\varphi \in \mathcal{L}_N^- \Rightarrow$ $\varphi \in \mathcal{L}'_{\mathcal{Y}_N}$	•	ν	ν	ν	ν	ν	
$\varphi \in \mathcal{L}_N^{\equiv} \Rightarrow$ $\varphi \in \mathcal{L}'_{\mathcal{Y}_N}$		•		ν	ν	ν	
$\varphi \in \mathcal{L}'_{\mathcal{Y}_N}, \sigma \in \mathcal{L}_N^- \Rightarrow$ $\sigma \wedge \varphi \in \mathcal{L}'_{\mathcal{Y}_N}$			•	ν	ν	ν	
$\varphi \in \mathcal{L}'_{\mathcal{Y}_N}, \sigma \in \mathcal{L}_N^{\equiv} \Rightarrow$ $\sigma \wedge \varphi \in \mathcal{L}'_{\mathcal{Y}_N}$				•	•	ν	
$X \subseteq Act, \varphi_a \in \mathcal{L}'_{\mathcal{Y}_N} \forall a \in X \Rightarrow$ $\bigwedge_{a \in X} a\varphi_a \in \mathcal{L}'_{\mathcal{Y}_N}$					•	ν	
$\varphi_i \in \mathcal{L}'_{\mathcal{Y}_N} \forall i \in I \Rightarrow$ $\bigwedge_{i \in I} \varphi_i \in \mathcal{L}'_{\mathcal{Y}_N}$						•	
$\varphi \in \mathcal{L}_N \Rightarrow$ $\varphi \in \mathcal{L}'_{\mathcal{Y}_N}$						•	
$\varphi \in \mathcal{L}_N \Rightarrow$ $\neg\varphi \in \mathcal{L}'_{\mathcal{Y}_N}$						•	

Table 5: Our new logical characterizations for the semantics at each level of the spectrum.

7. RELATING THE UNIFIED LOGICS AND THE UNIFIED OBSERVATIONAL MODEL

In this section we will relate our unified logical characterizations and the unified observational semantics. As indicated in Section 2, we have to restrict ourselves to image-finite processes; as a byproduct, the finite parts of each of the corresponding languages, that are obtained by intersection with \mathcal{L}_{HM}^f , provide us with a pure finite logical characterization of the semantics. However, it is convenient in the first part of this Section to consider still the full (infinitary) logic characterizing each of the semantics.

Definition 7.1 (Normal formulas $\mathcal{N}(\mathcal{L})$).

- (1) Given a set of formulas \mathcal{L} whose outermost operator is not conjunction, the set $\mathcal{N}(\mathcal{L})$ of induced *normal formulas* is defined by:
 - $\top \in \mathcal{N}(\mathcal{L})$;
 - if $\Gamma_1, \Gamma_2 \subseteq \mathcal{L}, \{a_i \mid i \in I\} \subseteq Act$, and $\varphi_i \in \mathcal{N}(\mathcal{L})$, then $(\bigwedge_{\sigma \in \Gamma_1} \sigma \wedge \bigwedge_{\sigma \in \Gamma_2} \neg\sigma) \wedge \bigwedge_{i \in I} a_i\varphi_i \in \mathcal{N}(\mathcal{L})$.
- (2) For each $N \in \{U, C, I, T, S\}$ and each $\mathcal{Y}_N \in \{NS, \leq_N^l, \leq_N^{l\supseteq}, \leq_N^{lf}, \leq_N^{lf\supseteq}, \leq_N^{l\subseteq}, \leq_N^{lf\subseteq}, D_N\}$ in the spectrum, we define the set of normal formulas $\mathcal{N}_{\mathcal{Y}_N}(\mathcal{L}''_N) \subseteq \mathcal{L}'_{\mathcal{Y}_N}$ as $\mathcal{N}_{\mathcal{Y}_N}(\mathcal{L}''_N) = \mathcal{N}(\mathcal{L}''_N) \cap \mathcal{L}'_{\mathcal{Y}_N}$, where \mathcal{L}''_N is the set of formulas in \mathcal{L}'_N whose outermost operator is not conjunction.

Remark 7.2. The clause in Definition 7.1.1 is more involved than it appears. Initially, we can apply it with $I = \emptyset$ to obtain the first (non-trivial) normal formulas and then recursively to obtain more complex normal formulas; note that the two first subformulas stem always from the original set \mathcal{L} . By abuse of notation, when some of the elements in our normal formulas do not appear in the corresponding set $\mathcal{L}'_{\mathcal{Y}_N}$, we assume that these formulas have been extended by conjunction with \top using the fact that $\bigwedge_{\sigma \in \emptyset} \sigma$ is another syntactic form to express \top .

Also note that infinite conjunction is allowed in the two first subformulas. As a consequence, if we consider the tree-like form of these (possibly infinitary) formulas they could have infinite depth. However, if we define the normal depth of formulas in $\mathcal{N}(\mathcal{L}_{\mathcal{N}})$ as that obtained by counting the recursive nesting in the application of Definition 7.1, then any normal formula has finite normal depth, and the set they form can be explored by structural induction.

Theorem 7.3. *Each set of normal formulas $\mathcal{N}_{\mathcal{Y}_{\mathcal{N}}}(\mathcal{L}''_{\mathcal{N}})$ associated to the semantics in the spectrum is equivalent to the full set of formulas $\mathcal{L}'_{\mathcal{Y}_{\mathcal{N}}}$.*

Proof. By structural induction, all the formulas in $\mathcal{L}'_{\mathcal{Y}_{\mathcal{N}}}$ admit a normal formula in the sense of Definition 7.1, that is obtained by gathering the subformulas and applying Proposition 6.2. \square

Definition 7.4. The set of *complete normal* formulas $\mathcal{CN}(\mathcal{L})$ (resp., the set of complete normal formulas associated to each semantics in the spectrum, $\mathcal{CN}_{\mathcal{Y}_{\mathcal{N}}}(\mathcal{L}''_{\mathcal{N}})$) is the set of normal formulas (resp., the set of normal formulas associated to each semantics in the spectrum) for which the rule in Definition 7.1 is applied with $\Gamma_2 = \overline{\Gamma_1}$.

Now we prove that infinite conjunction in Definition 7.1 can be approximated by finite conjunction.

Theorem 7.5. *If we restrict ourselves to image-finite processes, for each denumerable set of formulas \mathcal{L} , any complete normal formula $\varphi \in \mathcal{CN}(\mathcal{L})$ can be approximated by a set of finite normal formulas $\{\varphi^k \mid k \in \mathbb{N}\}$ that only use finite conjunction, that is, $p \models \varphi$ iff $p \models \varphi^k$ for all $k \in \mathbb{N}$.*

Proof. We define the sequence φ^k by structural induction on the normal depth of φ :

- $\varphi = (\bigwedge_{\sigma \in \Gamma_1} \sigma \wedge \bigwedge_{\sigma \in \overline{\Gamma_1}} \neg \sigma)$. We consider a fixed enumeration of the set $\mathcal{L} = \{\sigma_n \mid n \in \mathbb{N}\}$, and define $\mathcal{L}^{\leq n} = \{\sigma_j \in \mathcal{L} \mid j \leq n\}$. Then, for each $k \in \mathbb{N}$:

$$\varphi^k = \bigwedge_{\sigma \in \Gamma_1 \cap \mathcal{L}^{\leq k}} \sigma \wedge \bigwedge_{\sigma \in \overline{\Gamma_1} \cap \mathcal{L}^{\leq k}} \neg \sigma.$$

We have $p \models \varphi \Leftrightarrow (p \models \sigma \forall \sigma \in \Gamma_1 \text{ and } p \not\models \sigma \forall \sigma \notin \Gamma_1)$ and $p \models \varphi^k \forall k \in \mathbb{N} \Leftrightarrow (p \models \sigma \forall \sigma \in \Gamma_1 \cap \mathcal{L}^{\leq k} \text{ and } p \not\models \sigma \forall \sigma \in \overline{\Gamma_1} \cap \mathcal{L}^{\leq k})$ and the result follows from a the equality

$$\Gamma_1 = \Gamma_1 \cap \mathcal{L} = \Gamma_1 \cap \left(\bigcup_{k \in \mathbb{N}} \mathcal{L}^{\leq k} \right).$$

- $\varphi = (\bigwedge_{\sigma \in \Gamma_1} \sigma \wedge \bigwedge_{\sigma \in \overline{\Gamma_1}} \neg \sigma) \wedge \bigwedge_{i \in I} a_i \varphi_i$. By structural induction we can assume that the result is true for any subformula φ_i . Then we define $\varphi^k = \bigwedge_{\sigma \in \Gamma_1 \cap \mathcal{L}^{\leq k}} \sigma \wedge \bigwedge_{\sigma \in \overline{\Gamma_1} \cap \mathcal{L}^{\leq k}} \neg \sigma \wedge \bigwedge_{i \in I} a_i \varphi_i^k$. Now, if we decompose φ as $\varphi_I \wedge \varphi_{II}$ (taking $\varphi_{II} = \bigwedge_{i \in I} a_i \varphi_i$, and analogously for the set of approximations) we have that $p \models \varphi^k$ iff $p \models \varphi_I^k$ and $p \models \varphi_{II}^k$. If $p \models \varphi^k$ then $p \models \varphi_I^k$ for all $k \in \mathbb{N}$ and arguing as in the base case above we conclude that $p \models \varphi_I$. Any image-finite process p can be decomposed as $p = \sum_{a_i \in Act} \sum_{j=1}^{m_i} a_i^j p_i^j$, and we have $p \models \varphi_{II}^k$ iff for all i there exists j with $a_i = a_i^j$ and $p_i^j \models \varphi_i^k$. Then, if $p \models \varphi_{II}^k$ for all $k \in \mathbb{N}$, for each i there exists some $j \in 1..m_i$ such that $p_i^j \models \varphi_i^k$ for infinitely many k , but this means that $p_i^j \models \varphi_i^k$ for all $k \in \mathbb{N}$ and then, by the induction hypothesis, $p_i^j \models \varphi_i$ thus getting $p \models \varphi$. \square

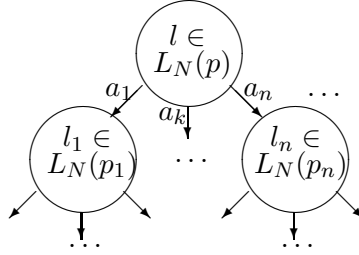


Figure 14: A branching observation.

Definition 7.6. For each $N \in \{U, C, I, T, S\}$ and each $\mathcal{Y}_N \in \{NS, \leq_N^l, \leq_N^{\lceil}, \leq_N^f, \leq_N^{f\lceil}, \leq_N^{\lceil}, \leq_N^{f\lceil}, D_N\}$ in the spectrum, we define the finite logic for the semantics $\mathcal{L}_{\mathcal{Y}_N}^f$ as $\mathcal{L}_{\mathcal{Y}_N}^f \cap \mathcal{L}_{HM}^f$.

Corollary 7.7. For each $N \in \{U, C, I, T, S\}$ and each $\mathcal{Y}_N \in \{NS, \leq_N^l, \leq_N^{\lceil}, \leq_N^f, \leq_N^{f\lceil}, \leq_N^{\lceil}, \leq_N^{f\lceil}, D_N\}$ in the spectrum, if we restrict ourselves to the set of image-finite processes we have $\mathcal{L}_{\mathcal{Y}_N}^f \sim \mathcal{L}'_{\mathcal{Y}_N}$.

Proof. We only need to apply Theorem 7.5. The only non trivial case is when $N = S$, where we have to apply twice the Theorem, using also the fact that $CN(\mathcal{L}_S'') \sim CN(\mathcal{L}_S^f)$, because $\mathcal{L}_S'' \sim \mathcal{L}_S^f$. \square

Theorem 7.8. For each $N \in \{U, C, I, T, S\}$ and each $\mathcal{Y}_N \in \{NS, \leq_N^l, \leq_N^{\lceil}, \leq_N^f, \leq_N^{f\lceil}, \leq_N^{\lceil}, \leq_N^{f\lceil}, D_N\}$ in the spectrum there exists a correspondence between the set of complete normal formulas $CN_{\mathcal{Y}_N}(\mathcal{L}_N'')$ and the corresponding domain of observations ΩGO_N with $\Omega \in \{B, L\}$. This correspondence \leftrightarrow satisfies that $\varphi \leftrightarrow \theta$ implies that $(p \models \varphi \text{ iff } \theta \in \Omega GO_N(p))$. Moreover:

- (1) The set of complete normal formulas $CN_{NS}(\mathcal{L}_N'')$ (resp. $CN_{\mathcal{D}_N}(\mathcal{L}_N'')$) and the domain of branching general observations BGO_N (resp. $dBGO_N$) are isomorphic, that is, \leftrightarrow is one to one.
- (2) The set of complete normal formulas $CN_{\leq_N^l}(\mathcal{L}_N'')$, $CN_{\leq_N^{\lceil}}(\mathcal{L}_N'')$ and the domain of linear general observations LGO_N are isomorphic, that is, \leftrightarrow is one to one.
- (3) The set of complete normal formulas $CN_{\leq_N^f}(\mathcal{L}_N'')$ (resp. $CN_{\leq_N^{f\lceil}}(\mathcal{L}_N'')$) and the quotient domain $LGO_N / \simeq_{\leq_N^f}^{lf}$ (resp. $LGO_N / \simeq_{\leq_N^{f\lceil}}^{lf\lceil}$) are isomorphic, that is, \leftrightarrow^{-1} is injective and $\varphi \leftrightarrow \theta$ iff $\theta \simeq_{\leq_N^{f\lceil}}^{lf\lceil} \varphi$, for some adequate θ_φ .

Proof.

- (1) As can be seen in Figure 14, a branching observation is a labeled tree whose nodes are local observations and whose arcs are labeled by actions.

The general form of any complete normal formula in $CN_{NS}(\mathcal{L}_N)$ is $(\bigwedge_{\sigma \in \Gamma} \sigma \wedge \bigwedge_{\sigma \notin \Gamma} \neg \sigma) \wedge \bigwedge_{i \in I} a_i \varphi_i$, with $\varphi_i \in CN_{NS}(\mathcal{L}_N)$ for all $i \in I$. Since the language \mathcal{L}_N' characterizes the semantics used to get the local observations, we can associate to each complete formula $(\bigwedge_{\sigma \in \Gamma} \sigma \wedge \bigwedge_{\sigma \notin \Gamma} \neg \sigma)$ the corresponding local observation $l \in L_N$. Then, by structural induction, we obtain the observation associated to each formula $\varphi_i \in CN_{NS}(\mathcal{L}_N)$, thus getting the branching general observation BGO_N associated to the given formula. It is easy to see that this correspondence is indeed a bijection.

The case for $\mathcal{CN}_{\mathcal{D}_N}(\mathcal{L}_N)$ is analogous, but now it is not allowed to have repeated actions in the arcs leaving any node of an observation; this is obviously reflected in the form of the formulas in the corresponding language.

- (2) The case for $\mathcal{CN}_{\leq_N^l}(\mathcal{L}_N)$ is similar to the previous one, but now the obtained (degenerated) tree is just a single branch corresponding to a lgo in LGO_N .

For $\mathcal{CN}_{\leq_N^{l\supseteq}}(\mathcal{L}_N)$ the general form of a complete normal formula is $\varphi = (\top \wedge \bigwedge_{\sigma \notin \Gamma} \neg \sigma) \wedge a\varphi'$, with $\varphi' \in \mathcal{CN}_{\leq_N^{l\supseteq}}(\mathcal{L}_N)$. If we close the set Γ by derivability obtaining Γ' and then consider its complement $\overline{\Gamma'}$, we can consider the local observation l that satisfies all the formulas in $\overline{\Gamma'}$ and none in Γ' . The linear general observation lgo corresponding to φ is then recursively defined as $\langle l, \{(a, lgo')\} \rangle$ where lgo' is the linear general observation corresponding to φ' .

To proceed in the opposite direction, we just need to take as Γ the complement of the set of formulas in \mathcal{L}'_N satisfied by the local observation l at the root of the given LGO_N , and then proceed in a recursive way.

- (3) In this case, the general form of a complete normal formula in $\mathcal{CN}_{\leq_N^{lf}}(\mathcal{L}_N)$ is $\varphi = \top \wedge a_1(\dots(\top \wedge a_{n-1}(\top \wedge a_n(\bigwedge_{\sigma \in \Gamma} \sigma \wedge \bigwedge_{\sigma \notin \Gamma} \neg \sigma) \dots))$. Now we establish a correspondence between the set of local observations L_N and the sets $\Gamma \subseteq \mathcal{L}_N$ as done in cases (1) and (2) above, and then define the correspondence \leftrightarrow by ignoring the values of all the intermediate local observations in the considered lgo , keeping only the local observation at the end.

For $\mathcal{CN}_{\leq_N^{lf\supseteq}}(\mathcal{L}_N)$ we just need to apply the same procedure above combined with the ideas along the proof for $\mathcal{CN}_{\leq_N^{l\supseteq}}(\mathcal{L}_N)$. \square

Remark 7.9. It came as a surprise to notice that the lgo 's in LGO_N are in a bijective relation both with the complete normal formulas in $\mathcal{N}_{\leq_N^l}(\mathcal{L}_N)$ and those in $\mathcal{N}_{\leq_N^{l\supseteq}}(\mathcal{L}_N)$, so let us consider the case $N = I$ to explain this fact. A cnf in $\mathcal{N}_{\leq_I^l}(\mathcal{L}_I)$ specifies the corresponding local observation $I(p) \subseteq \mathcal{P}(Act)$ by means of a formula $(\bigwedge_{\sigma \in \Gamma} \sigma \wedge \bigwedge_{\sigma \notin \Gamma} \neg \sigma)$, where the formulas in Γ are just the elements of the corresponding set $I(p)$ while those in $\overline{\Gamma}$ correspond to its complement. When considering the failure trace semantics, the formulas in $\mathcal{N}_{\leq_I^{l\supseteq}}(\mathcal{L}_I)$ only contain the part $\bigwedge_{\sigma \notin \Gamma} \neg \sigma$ corresponding to the complement $\overline{I(p)}$. Since in this case the sets of lgo 's could be assumed to be closed with respect to the order $\leq_N^{l\supseteq}$ in Definition 4.19, soundness is retained after “assuming” that any formula $\bigwedge_{\sigma \notin \Gamma} \neg \sigma$ “generates” the observation associated to Γ , even though some of the formulas $\sigma \in \Gamma$ may not be satisfied when the corresponding $I(p)$ is smaller. But for the failures and failure trace semantics we can proceed by closing the set of offers upwards with respect to \subseteq and no new failure is introduced.

Theorem 7.10. *For each $N \in \{U, C, I, T, S\}$ and each $\mathcal{Y}_N \in \{NS, \leq_N^l, \leq_N^{l\supseteq}, \leq_N^{lf}, \leq_N^{lf\supseteq}, \leq_N^{lf\subseteq}, \leq_N^{lf\subseteq}, D_N\}$ in the spectrum, if we restrict ourselves to image-finite processes, the logical semantics $\sqsubseteq_{\mathcal{Y}_N}^f$ induced by the logic $\mathcal{L}_{\mathcal{Y}_N}^f$, is equivalent to the corresponding observational semantics in Definitions 4.2, 4.11 and 4.32. In order to unify our notation, here we will denote by GO_N the corresponding semantic domain.*

Proof. By Theorem 7.3, $\mathcal{L}'_{\mathcal{Y}_N} \sim \mathcal{N}_{\mathcal{Y}_N}(\mathcal{L}_N)$, and from Theorem 7.8 we get the isomorphism between the set $\mathcal{CN}_{\mathcal{Y}_N}(\mathcal{L}_N)$ and the corresponding set of general observations GO_N .

To finish the proof, we just need to show that $\mathcal{N}_{\mathcal{Y}_{\mathcal{N}}}(\mathcal{L}_{\mathcal{N}})$ and $\mathcal{CN}_{\mathcal{Y}_{\mathcal{N}}}(\mathcal{L}_{\mathcal{N}})$ are equivalent. Any consistent formula in $\mathcal{N}_{\mathcal{Y}_{\mathcal{N}}}(\mathcal{L}_{\mathcal{N}})$ ($\Gamma_1 \cap \Gamma_2 = \emptyset$) provides only some partial information about the states in a computation, so that the concrete values of these states are characterized by a set Γ with $\Gamma_1 \subseteq \Gamma \subseteq \overline{\Gamma_2}$. Therefore, we can replace Γ_1 and Γ_2 by Γ and $\overline{\Gamma}$, respectively, adding the disjunction over all the possible values of Γ , to characterize the set of processes specified by the formula. Now it is enough to float the disjunction up to obtain a disjunction of formulas in $\mathcal{CN}_{\mathcal{Y}_{\mathcal{N}}}(\mathcal{L}_{\mathcal{N}})$, and applying Proposition 6.2 we get the equivalence between the two sets of formulas. Finally, we only need to apply Corollary 7.7 to conclude. \square

Corollary 7.11.

- (1) *The unified logical semantics in Definition 6.10 is equivalent to the N-simulation semantics.*
- (2) *The unified logical semantics in Definition 6.15.1 is equivalent to the N-ready trace semantics.*
- (3) *The unified logical semantics in Definition 6.15.2 is equivalent to the N-failure trace semantics.*
- (4) *The unified logical semantics in Definition 6.15.3 is equivalent to the N-readiness semantics.*
- (5) *The unified logical semantics in Definition 6.15.4 is equivalent to the N-failure semantics.*
- (6) *The unified logical semantics in Definition 6.22 is equivalent to the N-deterministic branched semantics.*

Moreover, if we restrict ourselves to image-finite processes we have also an equivalence with the corresponding finite logical semantics.

Proof. Since it was proved in Section 4 that any observational semantics characterizes the corresponding (classical) semantics in the (extended) ltbt spectrum, the desired equivalence between our (unified) logical characterizations and the classical semantics is an immediate corollary. \square

8. ON THE REAL DIAMOND STRUCTURE

This section is a practical proof of the suitability of our unification work. Some recently proposed semantics that were not in the original ltbt spectrum are nicely included in our extended spectrum, which shows why and how the old spectrum has to be expanded. Our unified approach immediately absorbs these new semantics and the results about the different characterizations are easily extended to cover them. We warmly thank Roscoe for pointing out to us his work on the stable revivals semantics [46, 48], where an endeavor for an adequate presentation of the notion of responsiveness for a CSP-like language is made. (Responsiveness had been previously studied by Fournet et al. in [27] for CCS, under the name of stuck-freeness.)

When faced with the diamond shape of the collection of linear semantics that are associated to each simulation semantics in the extended spectrum, it would be natural to expect it to reflect the structure of a lattice. Then, failure semantics would be the greatest lower bound of the readiness and failure trace semantics, while ready trace semantics would be the corresponding lowest upper bound. However, both intuitions turn out to be wrong

and a new semantics finer than failures and another one coarser than ready trace can be found: together with readiness and failure trace, they do constitute a lattice.

Let us first consider the case of the lowest upper bound. We postulate that the axiomatization of the associated semantics is obtained by instantiating our general axiom with the conjunction of the two conditions M_R and M_{FT} :

$$M_{R\wedge FT}(x, y, w) \iff I(x) \supseteq I(y) \text{ and } I(w) \subseteq I(y).$$

We denote with $\sqsubseteq_{R\wedge FT}$ the order axiomatized by the corresponding axiom ($ND^{R\wedge FT}$).

Definition 8.1. The *readiness and failure trace semantics*, or *join semantics* $R \wedge FT$, is that defined by the order $\sqsubseteq_{R\wedge FT}$ generated by the set of axioms $\{B_1\text{--}B_4, (RS), (ND^{R\wedge FT})\}$.

Proposition 8.2. *The ready trace semantics is strictly finer than the readiness and failure trace semantics.*

Proof. $\sqsubseteq_{RT} \subseteq \sqsubseteq_{R\wedge FT}$ is an immediate consequence of Proposition 3.2 and the fact that condition M_{RT} implies both M_R and M_{FT} , and hence also $M_{R\wedge FT}$. To show that $\sqsubseteq_{RT} \not\subseteq \sqsubseteq_{R\wedge FT}$, let us take $w = \mathbf{0}$, $y = b$, and $x = bB' + c$; then we have:

$$\underbrace{a(bB + bB' + c)}_p \sqsubseteq_{R\wedge FT} \underbrace{a(bB' + c) + abB}_q$$

but, if $I(B) \neq I(B')$,

$$a(bB + bB' + c) \not\sqsubseteq_{RT} a(bB' + c) + abB$$

because $\{a\}a\{b, c\}bI(B) \in \text{ReadyTraces}(p) \setminus \text{ReadyTraces}(q)$. \square

It is clear that the readiness and failure trace semantics is finer than both the readiness and the failure trace semantics; to show that it is actually the coarsest upper bound we need to prove that $\sqsubseteq_{R\wedge FT} = \sqsubseteq_R \cap \sqsubseteq_{FT}$. Even if the axiom ($ND^{R\wedge FT}$) was created with this goal in mind, this cannot be easily shown using only algebraic arguments. Instead, it is trivial to obtain the observational characterization of the desired semantics by gathering together the failure trace and the ready observations. Based on Definition 4.19, we can define the corresponding order $\leq_N^{l\supseteq \wedge f}$ by taking

$$\mathcal{T} \leq_N^{l\supseteq \wedge f} \mathcal{T}' \iff \mathcal{T} \leq_N^{l\supseteq} \mathcal{T}' \text{ and } \mathcal{T} \leq_N^{lf} \mathcal{T}'.$$

A direct characterization can be obtained as follows. We combine both kinds of observations into a single family of decorated traces that we call *failure trace with final ready sets*, by considering failure sets all along the trace except at the end of it, where we introduce the corresponding ready set.

Definition 8.3. We define the order $\leq_N^{l\supseteq \wedge f}$ by

$$\mathcal{T} \leq_N^{l\supseteq \wedge f} \mathcal{T}' \iff \text{for all } X_0 a_1 \dots X_n \in \mathcal{T} \text{ there is some } Y_0 a_1 \dots Y_n \in \mathcal{T}' \\ \text{with } X_n = Y_n \text{ and } X_i \supseteq Y_i, \text{ for } i \in 0..n-1.$$

Proposition 8.4. *The semantics defined by the order $\sqsubseteq_{R\wedge FT}$ coincides with that defined by $\leq_I^{l\supseteq \wedge f}$ and is thus the lowest upper bound of the readiness and failure trace semantics.*

Proof. Similar to that of Theorem 5.4. \square

Let us finally consider the logical characterization of this semantics. It is clear that the conjunction of two semantics should be characterized in a logical way by simply considering the union of the logics that characterize both semantics (although there could possibly be a more compact presentation).

Definition 8.5. We define the set of formulas $\mathcal{L}'_{\leq_I \sqsupset \wedge f}$ as that generated by the clauses:

- $\top \in \mathcal{L}'_{\leq_I \sqsupset \wedge f}$;
- if $\varphi \in \mathcal{L}'_{\leq_I \sqsupset \wedge f}$ and $\sigma \in \mathcal{L}'_I$ then $\sigma \wedge \varphi \in \mathcal{L}'_{\leq_I \sqsupset \wedge f}$;
- if $\sigma \in \mathcal{L}'_I$ then $\sigma \in \mathcal{L}'_{\leq_I \sqsupset \wedge f}$;
- if $\varphi \in \mathcal{L}'_{\leq_I \sqsupset \wedge f}$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_{\leq_I \sqsupset \wedge f}$.

Proposition 8.6. *The logical semantics $\sqsubseteq'_{\leq_I \sqsupset \wedge f}$ induced by the logic $\mathcal{L}'_{\leq_I \sqsupset \wedge f}$ is equivalent to the observational semantics defined by $\leq_I^{\sqsupset \wedge f}$.*

Proof. We just need to check that $\mathcal{L}'_{\leq_I^{\sqsupset \wedge f}} = \mathcal{L}'_{\leq_I^{\sqsupset}} \cup \mathcal{L}'_{\leq_I^{\wedge f}}$, which is immediate. \square

By replacing the I above by the generic N , we get the definitions and results for the general case.

The axiomatic characterization of the greatest lower bound of the readiness and failure trace semantics is much simpler: we simply put together the axioms for the orders defining both semantics.

Definition 8.7. The *meet semantics* $R \vee FT$ is that defined by the order $\sqsubseteq_{R \vee FT}$ generated by the set of axioms $\{B_1-B_4, (RS), (ND^R), (ND^{FT})\}$.

If we define $M_{R \vee FT}$ as $M_R \vee M_{FT}$, that is, $M_{R \vee FT}(x, y, w)$ holds if $I(x) \supseteq I(y)$ or $I(w) \subseteq I(y)$, we have the following characterization of $\sqsubseteq_{R \vee FT}$.

Proposition 8.8. *The order $\sqsubseteq_{R \vee FT}$ is that generated by the set of axioms $\{B_1-B_4, (RS), (ND^{R \vee FT})\}$, where $(ND^{R \vee FT})$ is the instantiation of the generic axiom (ND) with the condition $M_{R \vee FT}$.*

Proposition 8.9. *The semantics defined by the order $\sqsubseteq_{R \vee FT}$ is the finest semantics that is coarser than both the readiness and the failure trace semantics.*

Proof. Obvious since any semantics coarser than the readiness semantics has to satisfy $\{B_1-B_4, (RS), (ND^R)\}$, any one coarser than failure trace must satisfy $\{B_1-B_4, (RS), (ND^{FT})\}$, and $M_{R \vee FT}$ is equivalent to $M_R \vee M_{FT}$. \square

Once again, the semantics defined by $\sqsubseteq^{R \vee FT}$ is not included in the ltbt spectrum and neither in our extended one; in particular, it is different from the failures semantics. To prove this fact we make essential use of the notion of *revival*, as defined by Reed, Roscoe, and Sinclair [46]. Revivals are sequences $a_1, \dots, a_n(X, a)$ where a_1, \dots, a_n is a trace of the corresponding process after which the action a is offered, but the set of actions X is refused.

Proposition 8.10. *The meet semantics $R \vee FT$ is strictly finer than failure semantics.*

Proof. The inclusion $\sqsubseteq^{R \vee FT} \subsetneq \sqsubseteq^F$ is obvious since failures semantics is coarser than both the readiness and the failure trace semantics. To show that the inclusion is strict, note that any two processes related by $\sqsubseteq^{R \vee FT}$ do not only have the same failures but also the same revivals. This is indeed the case since all the axioms $u \preceq v$ in $\{B_1-B_4, (RS), (ND^R), (ND^{FT})\}$

preserve the revivals, which means $Revivals(\sigma(u)) \subseteq Revivals(\sigma(v))$ for every ground substitution σ , and the revivals order is a precongruence for the operators in BCCSP. For instance, for (ND^{FT}) we need to prove that $Revivals(\sigma(a(x+y))) \subseteq Revivals(\sigma(ax)) \cup Revivals(\sigma(a(y+w)))$ whenever $I(\sigma(w)) \subseteq I(\sigma(x))$. It is clear that the only non-trivial case occurs when $a(X,b) \in Revivals(\sigma(a(x+y)))$; then we have $(X,b) \in Revivals(\sigma(x+y))$ so that $X \in Failures(\sigma(x)) \cap Failures(\sigma(y))$ and $b \in I(\sigma(X))$ or $b \in I(\sigma(y))$. In the first case, $a(X,b) \in Revivals(\sigma(ax))$ whereas, in the second, $X \in Failures(\sigma(x+y))$ and therefore $a(X,b) \in Revivals(\sigma(a(x+y)))$. The case for (ND^R) is simpler. Once we know that \sqsubseteq^{RVFT} preserves the revivals we only need to observe that the revivals cannot be obtained from the failures of a process. In particular, we have $ab \sqsubseteq^F a + a(b+c)$, but $a(\{c\}, b) \in Revivals(ab) \setminus Revivals(a + a(b+c))$. \square

Next we present the characterization of the revivals semantics in terms of our observational framework.

Definition 8.11. We define the order $\leq_N^{I\supseteq V^f}$ by

$$\mathcal{T} \leq_N^{I\supseteq V^f} \mathcal{T}' \iff \text{for all } X_0 a_1 \dots X_n \in \mathcal{T} \\ \text{there is } \{Y_0 a_1 Y_1 \dots Y_n^j \mid j \in J\} \subseteq \mathcal{T}' \text{ such that } X_n = \bigcup_{j \in J} Y_n^j.$$

Proposition 8.12. For all $p, q \in BCCSP$, $Revivals(p) \subseteq Revivals(q)$ if and only if $LGO_I(p) \leq_I^{I\supseteq V^f} LGO_I(q)$.

Proof. Note that $\leq_I^{I\supseteq V^f}$ can be equivalently defined as

$$\mathcal{T} \leq_I^{I\supseteq V^f} \mathcal{T}' \iff \text{for all } X_0 a_1 \dots X_n \in \mathcal{T} \text{ and for all } a \in X_n \\ \text{there is } Y_0 a_1 \dots Y_n \in \mathcal{T}' \text{ with } a \in Y_n \text{ and } Y_n \subseteq X_n.$$

Now, since $a_1 \dots a_n(X, a) \in Revivals(p)$ if and only if there exists $X_0 a_1 \dots X_n \in LGO_I(p)$ such that $a \in X_n$ and $X_n \cap X = \emptyset$, we obtain the desired characterization. \square

Definition 8.13. Given $\mathcal{T} \subseteq LGO_N$, $\overline{\mathcal{T}}^{\supseteq V^f}$ is defined as

$$\overline{\mathcal{T}}^{\supseteq V^f} = \{X_0 a_1 \dots X_n \mid \text{there is } \{Y_0 a_1 \dots Y_n^j \mid j \in J\} \subseteq \mathcal{T} \text{ with } X_n = \bigcup_{j \in J} Y_n^j\}.$$

This clearly indicates that $\leq_I^{I\supseteq V^f}$ is in between $\leq_I^{lf\supseteq}$, defining the failures semantics, and \leq_I^{lf} , defining readiness semantics. This is useful for the proof of the axiomatic characterization of the revivals semantics.

Theorem 8.14. The revivals semantics defined by $\sqsubseteq_I^{I\supseteq V^f}$ is axiomatized by $\{B_1-B_4, (RS), (ND^{RVFT})\}$.

Proof sketch. It is quite similar to that of Theorem 5.4 for the case of failures semantics and, hence, also similar to the characterization of that semantics by means of acceptance trees [30] (and where the closure of the set of offers with respect to both union and convex closure is a critical argument), and this is why we only sketch it. In connection to that, recall that the application of the particular case of (ND) corresponding to (ND^{FT}) allowed us to join arbitrary states after the same trace, while that corresponding to (ND^R) allowed us to obtain a common continuation after the same action at any state reachable by the same trace. All this can be done now using (ND^{RVFT}) ; however, we cannot add to an arbitrary state an action offered at another state reachable by the same trace since to do that we needed the unlimited strength of axiom (ND) . \square

Note that for the join semantics $R \wedge FT$ the logical approach was the most direct way of defining it, whereas its equational characterization needed more care. For the meet semantics $R \vee FT$, the situation is just the opposite. As we have seen, $R \vee FT$ is axiomatized by putting together the axioms for R and those for FT ; in contrast, the logic characterizing $R \vee FT$ is obtained by cleverly selecting the common part of the logics characterizing both R and FT . If we had defined the logical semantics by considering all the formulas from HML that are preserved by each semantics, then we could take the intersection of these sets as the logical semantics of any meet semantics. Since we defined our logical semantics by considering only a “basis” that generates the corresponding full set, we cannot simply take their intersection.

Definition 8.15. We define the set of formulas $\mathcal{L}'_{\leq_I \mathcal{D} \vee f}$ as that generated by the clauses:

- $\top \in \mathcal{L}'_{\leq_I \mathcal{D} \vee f}$;
- if $\sigma, \sigma_j \in \mathcal{L}'_I$ for all $j \in J$ then $(\sigma \wedge \bigwedge_{j \in J} \neg \sigma_j \top) \in \mathcal{L}'_{\leq_I \mathcal{D} \vee f}$;
- if $\varphi \in \mathcal{L}'_{\leq_I \mathcal{D} \vee f}$ and $a \in Act$ then $a\varphi \in \mathcal{L}'_{\leq_I \mathcal{D} \vee f}$.

Note that in the second clause of this definition we have relaxed the condition in the definition of \mathcal{L}'_R by considering an arbitrary failure (that defined by the set J), but only a positive offer (the action appearing in σ). This is how the revivals semantics becomes slightly finer than the failures semantics.

Proposition 8.16. *The logical semantics $\sqsubseteq'_{\leq_I \mathcal{D} \vee f}$ induced by the logic $\mathcal{L}'_{\leq_I \mathcal{D} \vee f}$ is equivalent to the observational semantics defined by $\leq_I^{\mathcal{D} \vee f}$.*

Proof. In this case we have taken $\mathcal{L}'_{\leq_I \mathcal{D} \vee f} = \mathcal{L}'_{\leq_I \mathcal{D}} \cap \mathcal{L}'_{\leq_I f}$. Then, to prove that it defines $R \vee FT$ it is enough to check that $p \not\sqsubseteq'_{\leq_I \mathcal{D} \vee f} q$ implies that there exists φ in $\mathcal{L}'_{\leq_I \mathcal{D} \vee f}$ such that $p \models \varphi$ and $q \not\models \varphi$, which is almost immediate. \square

Again, by replacing the I above by the generic N , we get the definitions and results for the general case.

We can generalize most of the results obtained for the refusal semantics when $N = I$ to any reasonable local observation function such as T or S , once we interpret \subseteq as the corresponding order and $=$ as the induced equivalence. However, in order to define the adequate observational characterization of the revivals semantics for a local observation (or constraint) N , we should look for the adequate “elements” of the universe of observations. This leads us to traces when N is T , but it is not so clear how to define those “elements” for a non-extensional semantics such as that obtained when N is S .

Let us conclude this section with a look at the picture in Figure 15 showing the real structure of the full (bidimensional!) diamond, that should be included in all the upper levels of the extended ltbt spectrum.

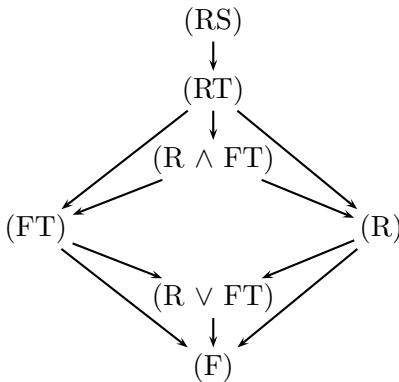


Figure 15: The real diamond below ready simulation.

9. OPERATIONAL SEMANTICS

In this section we explain how to develop the semantics in the spectrum in an operational way. Certainly, this presentation could be argued to be ad-hoc at times since some “high-level” conditions are required in the SOS-like rules for some of the semantics. Moreover, the style of presentation at this Section is certainly less precise and detailed than in the previous ones. However, we believe it still provides some additional insight on the common properties of the semantics and also establishes a connection with our previous work on (bi)simulations up-to [23, 25] as a way to get coinductive characterizations of any “reasonable” process semantics.

Structural operational semantics was introduced by G. Plotkin in 1981, even though his seminal work was not published in a journal until 2004 [45]. In Section 2 we already presented a basic operational semantics for our processes as a starting point for the definition of all the semantics in the spectrum: a small-step semantics that collects the (atomic) actions executed by the processes into the corresponding transition system. By contrast, all the operational semantics in this section will be big-step semantics which directly return the adequate semantic values defining each of the semantics. They are generated by means of SOS-like rules that obtain these values in a compositional way. An extensive presentation of structural operational semantics covering all its variants can be found in [39].

9.1. Local simulations up-to. In order to characterize all the reasonable behavior preorders in a coinductive way we need to generalize constrained N -simulations (Definition 2.2) with N -simulations up-to an order \sqsubseteq .

Definition 9.1. Let \sqsubseteq be a behavior preorder and N a relation over processes. We say that a binary relation S over processes is an N -simulation up-to \sqsubseteq if $S \subseteq N$ and S is a simulation up-to \sqsubseteq . Or equivalently, in a coinductive way, whenever we have pSq we also have:

- for every a , if $p \xrightarrow{a} p'_a$ then there exist q', q'_a such that $q \sqsupseteq q' \xrightarrow{a} q'_a$ and $p'_a Sq'_a$;
- pNq .

We say that process p is N -simulated up-to \sqsubseteq by process q , or that process q N -simulates process p up-to \sqsubseteq , written $p \stackrel{N}{\sqsubseteq} q$, if there exists an N -simulation up-to \sqsubseteq , S , such that pSq .

We often just write $\stackrel{N}{\sqsubseteq}$, instead of $\stackrel{N}{\sqsubseteq}_{\sqsubseteq}$, when the behavior preorder is clear from the context.

We proved in [23] that all the preorders defining the semantics in the ltbt spectrum can be characterized as N -simulations up-to the corresponding equivalence relation \equiv , where N is the constraint defining the coarsest simulation semantics finer than the given semantics. For instance, the result for the semantics between failures semantics and ready simulation was the following.

Theorem 9.2 ([23]). *For every behavior preorder \sqsubseteq satisfying the axiom (RS) and $\sqsubseteq \subseteq I$, we have $p \sqsubseteq q$ if and only if $p \stackrel{I}{\sqsubseteq} q$.*

Table 6 shows the constraints defining the adequate constrained simulation order finer than each of the semantics in the linear time-branching time spectrum. Obviously, they coincide with the layer of the extended spectrum at which each semantics appear.

	T	S	CT	CS	F	R	FT	RT	PW	RS	PF	2N
C_O	U	U	C	C	I	I	I	I	I	I	V	W
	$pUq \iff true$								$pVq \iff p \equiv_T q$			
	$pCq \iff (p = \mathbf{0} \text{ iff } q = \mathbf{0})$								$pWq \iff p \equiv_S q$			
	$pIq \iff I(p) = I(q)$											

Table 6: Constraints for the semantics in the ltbt spectrum.

Note that Theorem 9.2 is more subtle than it could appear: it characterizes a given preorder with a constrained simulation up to the preorder itself (Definition 9.1). Therefore, there are several semantics that share the same constraint. This characterization is indeed rather technical and the key point is that it allows to express any behavior preorder in a simulation-like fashion. We have used this characterization to prove many useful statements in our previous work⁵ and we will use it again several times in the current Section.

In our proof of the completeness of the axiomatizations for the linear semantics in the spectrum in Section 5 we used a notion of normal form which, roughly, was defined by applying repeatedly to any term p the axiom (ND_{\equiv}) from right to left, for as long as possible. Propositions 5.2 and 5.3 were then the key results to complete the proof, and also lie behind the intuition for introducing now the notion of *local I-simulation up-to*.

Definition 9.3. For $Z \in \{F, R, FT, RT\}$ and $p = \sum_a \sum_i a p_a^i$, whenever we have a pair of indices i, j and a decomposition $p_a^j = r_a^j + s_a^j$ with $M_Z(p_a^j, r_a^j, s_a^j)$ we say that p is 1-locally Z -equivalent to $q = p + a(p_a^i + r_a^j)$, and we write $p \equiv_Z^{l1} q$. We say that p and q are locally Z -equivalent when they are related by the reflexive and transitive closure of \equiv_Z^{l1} , and then we write $p \equiv_Z^l q$.

For $Z \in \{F, R, FT, RT\}$ we refer to the I -simulations up-to \equiv_Z^l as local I -simulations up-to \equiv_Z . We say that process p is locally I -simulated up-to \equiv_Z by process q , or that

⁵For instance in [24] (Theorem 10) we provided an axiomatization for any behavior preorder starting from the equations of the corresponding equivalence.

process q locally I -simulates process p up-to \equiv_Z , written $p \sqsubseteq_{\equiv_Z}^I q$, if there exists a local I -simulation up-to \equiv_Z , S , such that pSq .

Local I -simulations up-to are enough to characterize the linear semantics in $\{F, R, FT, RT\}$. Note that we cannot get a local notion of bisimulation up-to equivalent to our unrestricted notion of bisimulation up-to.

Proposition 9.4. *For $Z \in \{F, R, FT, RT\}$ we have $p \sqsubseteq_Z q$ if and only if $p \sqsubseteq_{\equiv_Z}^I q$.*

Proof. The implication from right to left is an immediate consequence of Theorem 9.2. For the other, note that $\{(p, q) \mid p \sqsubseteq_Z q\}$ is a local I -simulation up-to \equiv_Z . Indeed, for any $p \xrightarrow{a} p_a^i$ we have $q \equiv_Z^l hnf^Z(q)$ and taking $hnf^Z(q) = \sum_a \sum_i ah_a^j$ there exists some j such that $hnf^Z(q) \xrightarrow{a} h_a^j$ and $p_a^i \sqsubseteq_Z h_a^j$. \square

Example 9.5. Let us consider the processes $p = abc + abd$ and $q = a(bc + bd)$. We have $p \equiv_F q$ and we can check that $p \sqsubseteq_{\equiv_F}^I q$ since $p \sqsubseteq_{RS} q$. In order to prove that we also have $q \sqsubseteq_{\equiv_F}^I p$, we apply \equiv_F^l to p to obtain $p \equiv_F^l p + q$ and then we obtain $q \sqsubseteq_{RS} p$.

By contrast, if we wanted to apply our bisimulation up-to characterization to prove directly that $p \equiv_F q$ then we would have to turn q into $q + p$ in order to simulate the transition $p \xrightarrow{a} bc$. This would correspond to the local application of (ND_{\equiv}^F) combined with that of

$$(RS_{\equiv}) \quad I(x) = I(y) \implies a(x + y) \simeq a(x + y) + ax.$$

But if we replace the action a by a larger prefix $a_1 \dots a_n$ then we should also modify the process $q' = a_1 \dots a_n(bc + bd)$ in a non-local way in order to obtain $q'' = q' + p'$, so that we could suitably simulate the transition $p' = a_1 \dots a_n bc + a_1 \dots a_n bd \xrightarrow{a_1} a_2 \dots a_n bc$. Certainly, this is not necessary when checking $p' \equiv_F q'$ by means of local simulations up-to.

The coinductive characterization of the semantics by means of simulations up-to has at least two important advantages over that of using bisimulations up-to. First, we can characterize the orders defining the semantics and not just the induced equivalences; and second, we can use a local variant of the up-to mechanism so that we only need to rely on the equivalence relation \equiv_Z^l for the up-to part.

9.2. Operational rules for the linear semantics of processes. In Section 9.1 we have introduced and proved some results that establish the framework using which we achieve our goal: to define for each of the classic linear semantics an operational semantics over BCCSP terms in such a way that we can use constrained simulations to characterize the considered semantics. For instance, if we consider the case of the failures preorder \sqsubseteq_F , we are going to define a new operational semantics for BCCSP terms $(\mathcal{P}, Act, \Rightarrow_F)$ such that $p \sqsubseteq_F q$ if and only if q ready simulates p in $(\mathcal{P}, Act, \Rightarrow_F)$.

Next we will concentrate first on the diamond of linear semantics coarser than ready simulation. All these semantics are based on the observation of the initial set of actions of each process, that can be obtained by application of the SOS-like rules in Figure 16.

The rules in Figure 17 define the transition relation \implies_Z that induces the operational semantics to characterize each of the Z -semantics. The transition relation \longleftrightarrow_Z is an auxiliary relation that captures the iterated application of the axiom (ND_{\equiv}^Z) . Rules (RF) and (TR) define reflexivity and transitivity of the relation \longleftrightarrow_Z . Finally, the rule (CL)

$$\mathbf{0} \longrightarrow_I \emptyset \qquad ap \longrightarrow_I \{a\} \qquad \frac{p \longrightarrow_I A \quad q \longrightarrow_I B}{p + q \longrightarrow_I A \cup B}$$

Figure 16: Rules that compute the set of initial actions of a process.

$$\begin{aligned} \text{(ND)} \quad & \frac{p \longrightarrow_I A_p \quad q \longrightarrow_I A_q \quad r \longrightarrow_I A_r \quad M_Z(A_p, A_q, A_r)}{ap + a(q + r) + s \longleftrightarrow_Z ap + a(q + r) + a(p + q) + s} \\ \text{(RF)} \quad & p \longleftrightarrow_Z p \qquad \qquad \qquad \text{(TR)} \quad \frac{p \longleftrightarrow_Z q \quad q \longleftrightarrow_Z r}{p \longleftrightarrow_Z r} \\ \text{(CL)} \quad & \frac{p \longleftrightarrow_Z p' \quad p' \xrightarrow{a} q}{p \xrightarrow{a} q} \end{aligned}$$

Figure 17: Operational semantics characterizing the linear semantics.

combines the auxiliary relation \longleftrightarrow_Z and the original operational transition relation \longrightarrow (see Figure 2), to define the new labeled transitions \Longrightarrow_Z .

Definition 9.6. For $Z \in \{F, R, FT, RT\}$, the operational semantics for BCCSP terms is given by the labeled transition system $(\mathcal{P}, \text{Act}, \Longrightarrow_Z)$ where the transition relation \Longrightarrow_Z is defined by the rules in Figure 17.

By abuse of notation, we have written $M_Z(A_p, A_q, A_r)$ to express that we check $M_Z(p, q, r)$ using the initials computed by \longrightarrow_I .

The relation \Longrightarrow_Z has some interesting properties. First, it is an extension of the original transition system.

Proposition 9.7. For $Z \in \{F, R, FT, RT\}$, p and q BCCSP processes, and α a sequence of actions in Act , we have that $p \xrightarrow{\alpha} q$ implies $p \Longrightarrow_Z q$.

Although usually some new transitions appear, the set of initial actions of any process always remains the same.

Corollary 9.8. For $Z \in \{F, R, FT, RT\}$ and for any BCCSP process p , we have $I^\rightarrow(p) = I^{\Longrightarrow_Z}(p)$.

It is also clear that, for any $Z \in \{F, R, FT, RT\}$, the auxiliary relation \longleftrightarrow_Z preserves the equivalence \equiv_Z because the rule (ND) corresponds to the application of axiom $(I\text{-}ND_{\equiv}^Z)$, which is sound with respect to \equiv_Z^I .

Proposition 9.9. For $Z \in \{F, R, FT, RT\}$ and any two BCCSP processes p and q , we have $p \longleftrightarrow_Z q$ implies $p \equiv_Z q$.

Now we prove the main theorem in this section, that asserts that for each of the semantics in the considered diamond we can define the corresponding operational semantics as stated in Figure 17.

Theorem 9.10. For $Z \in \{F, R, FT, RT\}$ and any two BCCSP processes p and q , we have

$$p \sqsubseteq_Z q \iff p \sqsubseteq_{RS}^{\Longrightarrow_Z} q.$$

Proof. We will apply our characterization of the orders \sqsubseteq_Z by means of local I -simulations up-to at Proposition 9.4 to show that $p \sqsubseteq_{RS}^{\Rightarrow Z} q$ implies $p \sqsubseteq_{\cong_Z^I}^I q$. This is because any ready simulation over the transition system \Rightarrow_Z is also a local I -simulation up-to \cong_Z . Indeed, if R is a ready simulation over the transition system \Rightarrow_Z , and pRq , then whenever we have $p \xrightarrow{a} p'$ we also have $p \xrightarrow{a}_Z p'$, and therefore there is some $q \xrightarrow{a}_Z q'$ with $p'Rq'$. By definition of the transition system \Rightarrow_Z , there is some process q'' such that $q \longleftrightarrow_Z q''$ and $q'' \xrightarrow{a} q'$. Then we also have $q \cong_Z^I q''$, and thus R is indeed a local I -simulation up-to \cong_Z .

To prove that $p \sqsubseteq_{\cong_Z^I}^I q$ implies $p \sqsubseteq_{RS}^{\Rightarrow Z} q$, we will check that the relation $\sqsubseteq_{\cong_Z^I}^N$ is a ready simulation over the transition relation \Rightarrow_Z . If $p \sqsubseteq_{\cong_Z^I}^N q$, whenever $p \xrightarrow{a}_Z p'$ we have some process p'' such that $p \longleftrightarrow_Z p''$ and $p'' \xrightarrow{a} p'$. Then we also have $p \cong_Z p''$, and so $p'' \sqsubseteq_{\cong_Z^I}^N q$. From $p'' \xrightarrow{a} p'$ we now obtain that there are processes q' and q'' such that $q \cong_Z^I q''$, $q'' \xrightarrow{a} q'$, and therefore we also have $q \longleftrightarrow_Z q''$, thus concluding the proof. \square

As a consequence of our negative results at the end of Section 9.1, it is not possible to obtain an operational semantics locally defined from that which characterizes the linear semantics by means of bisimilarity. However, this can be done if we use mutual similarity instead of bisimulation.

Certainly, the fact that the characterizations in terms of bisimilarity cannot be defined in a local way is related to the fact that the transition systems generated by application of the algorithm in [17] are larger than those generated by our local transformation here. Unfortunately, it is true that our presentation does not magically lead (at least at the theoretical level) to more efficient algorithms to decide the equivalences with respect to the linear semantics (which are known to be quite hard to decide). Obviously, this is related to the fact that simulation is harder than bisimulation [37]. Even so, these are just theoretical worst case bounds, and it is nice to know that in practice we can apply a local transformation to generate the transition systems characterizing those semantics by means of the simulation orders, that in many concrete cases will not be too difficult to decide.

9.3. Characterizing the semantics corresponding to other constraints. Let us start by considering the case of the universal constraint U . As discussed in Section 3.2, if we use U in the condition M_Z it is clear that all the semantics in the corresponding diamond collapse into a single one: trace semantics. It is immediate to realize that the transition system to characterize it in terms of plain simulations is the same transition system \Rightarrow_F that we use to characterize the failures semantics by means of ready simulations.

Theorem 9.11. *The trace preorder \sqsubseteq_T coincides with the simulation order on the transition system \Rightarrow_F , that is, $p \sqsubseteq_T q$ iff $p \sqsubseteq_S^{\Rightarrow F} q$.*

Even if this coincidence is a simple fact that reflects the relation between traces and failures semantics, it contributes to clarify it. In plain words, failures semantics is just traces semantics enriched by the observation of initials, so that the plain simulation order that implies the trace order becomes the ready simulation order.

For other, finer observers such as T we can also characterize the corresponding semantic orders, such as possible and impossible futures, in terms of local simulations up-to. We can use that result to justify that the corresponding transition systems \Rightarrow_Z^T would characterize the semantic orders \sqsubseteq_Z^T in terms of T -simulations that preserve the set of traces of the

simulated process. In this case the corresponding operational characterization has to include rules for the computation of the set of traces $T(p)$ and this cannot certainly be done for infinite processes. But out of the computation of these sets, the rest of the rules for the generation of the corresponding transition systems \Longrightarrow_Z^T are also valid, and their local character is still present.

9.4. Application: trace deterministic normal forms. As a simple application we present the example used by Klin in [36], that we already used in [22] to illustrate our coinductive characterization of the behavior preorders by means of our bisimulations up-to.

Definition 9.12. For any process $p = \sum_a \sum_i a p_a^i$ the *deterministic form* of p is defined as $Det(p) = \sum_a a Det(\sum_i p_a^i)$.

We wish to prove that p and $Det(p)$ are trace equivalent. We will do it by proving that they are simulation equivalent over the transition system \Longrightarrow_F .

Proposition 9.13. *For any process p we have $p \sqsubseteq_F Det(p)$.*

Proof. We will prove that $p \sqsubseteq_S^{\Longrightarrow_F} Det(p)$ by showing that $R = \{(p, Det(p+q)) \mid p, q \text{ processes}\}$ is a simulation for the transition system \Longrightarrow_F . For $q = \sum_a \sum_j a q_a^j$ we have $Det(p+q) = \sum_a Det(\sum_i p_a^i + \sum_j q_a^j)$. Then, for any $p \xrightarrow{a}_F p'$ we have $p = p_a^i + \sum_k r_a^k$, for some index i and $p_a^k = r_a^k + s_a^k$ a decomposition of any of the rest of the summands of p . We have $Det(p+q) \xrightarrow{a} Det(\sum_i a p_a^i + \sum_j a q_a^j) = Det((p_a^i + \sum_k r_a^k) + (\sum_k r_a^k + \sum_j q_a^j))$, so that we also have $Det(p+q) \xrightarrow{a}_F Det((p_a^i + \sum_k r_a^k) + (\sum_k r_a^k + \sum_j q_a^j))$, with $(p_a^i + \sum_k r_a^k, Det((p_a^i + \sum_k r_a^k) + (\sum_k r_a^k + \sum_j q_a^j))) \in R$. \square

Proposition 9.14. *For any process p we have $Det(p) \sqsubseteq_F p$.*

Proof. We will prove that $Det(p) \sqsubseteq_S^{\Longrightarrow_F} p$ by showing that $R = \{(Det(p), p)\}$ is a simulation for the transition system \Longrightarrow_F . Since $Det(p)$ is deterministic for each $a \in Act$ there is a unique transition $Det(p) \xrightarrow{a}_F Det(\sum_i p_a^i)$. By applying the definition of \xrightarrow{a}_F we have $p \xrightarrow{a}_F \sum_i p_a^i$, and clearly we have $(Det(\sum_i p_a^i), \sum_i p_a^i) \in R$. \square

Although this is a very simple example, it is interesting to compare the proof above with that in [22]. This proof is simpler and more natural, mainly because the proof obligations to check bisimulations forced us to remove the sub-terms that were not in the chosen transition when we had to simulate it. This is not necessary for any of the two simulations that are needed to check mutual simulation, as done above. Obviously, this is also related to the impossibility to obtain a notion of local bisimulation up-to characterizing the equivalence under any of the linear semantics.

10. CONCLUSIONS AND SOME FUTURE WORK

Throughout this paper we have provided a global outline of process semantics from different points of view, each of which reveals some of the key ingredients for a more uniform comprehension of those semantics. We have noted that the family consisting of the simulation semantics—constrained simulations, in its generalized version—plays an essential role in the class of process semantics, becoming the cornerstone for sorting and classifying the remaining semantics.

From a framework in which, based on observational trees, denotational semantics are assigned—Section 4—we have been able to prove that the spectrum of process semantics can be structured by means of layers that are induced by the simulations. Each layer is dominated by a simulation semantics that determines the finest distinction available for that layer. The remaining semantic families are also described by abstracting or simplifying the observations needed for the corresponding layer. In particular, below each constrained simulation there appear the corresponding versions for each of the classic linear semantics—failures, readiness, failure trace, and ready trace—and, as we saw in Section 8, other semantics are also explained within our framework.

This observational characterization allowed us to offer a new insight into the axiomatic characterization of the semantics—Sections 3 and 5—revealing a uniformity lacking in all previous studies. To characterize any of the orders that define a process semantics, we have proved that it is enough to use two parametric axioms: one of the required axioms is that for the generalized simulation of the corresponding layer while the other, when it is present, has to do with the reduction of non-determinism that is carried out in each semantics.

Analogously, in Sections 6 and 7 we showed how to characterize process semantics by means of sets of Hennessy-Milner logic formulas out of their observational characterization, and finally we have also discussed a unified operational presentation of the semantics in the extended spectrum.

One of the more obvious lines for future work would be to consider those semantics that allow for an inner, non-visible action, known as *weak semantics*. Actually, some promising results have already been obtained that make clear the regularity and generality present in the domain of weak semantics. In particular, in [16] it is proved that it is possible to apply to weak semantics the algorithm to obtain axiomatic characterizations of semantic equivalences from the axioms for corresponding order [21]. And [2, 3] provides a detailed study of the axiomatization of weak simulation semantics.

Let us also cite here the recent work by Anti Valmari [53], where he presents the full catalogue of (weak) linear-time congruences for finite state systems. Certainly, it is interesting to limit somehow the class of “reasonable” semantics for processes, but this has not been so much the intention of our work in this paper. In fact, it is interesting to note that the results in the paper referenced above limit the set of semantics to explore in a quite personal way: for instance, the semantics of failure traces and that of ready traces are not included in the category, because Valmari (implicitly) considers that they are not “linear-time enough”.

Another interesting approach consists in the use of coalgebras—following the work, among others, of Jesse Hughes and Bart Jacobs [35]—where powerful categorical techniques allow to connect the idea of simulation with that of bisimulation, which is central in the coalgebraic setting. More concretely, these techniques were successfully used in [26] to relate classic and probabilistic bisimulation.

REFERENCES

- [1] Luca Aceto. Some of my favourite results in classic process algebra. In *In Bulletin of the EATCS*, pages 89–108, 2003.
- [2] Luca Aceto, David de Frutos-Escrig, Carlos Gregorio-Rodríguez, and Anna Ingólfssdóttir. Axiomatizing weak ready simulation semantics over bccsp . In Cerone and Pihlajasaari [15], pages 7–24.
- [3] Luca Aceto, David de Frutos-Escrig, Carlos Gregorio-Rodríguez, and Anna Ingólfssdóttir. The equational theory of weak complete simulation semantics over bccsp . In Bieliková et al. [12], pages 141–152.

- [4] Luca Aceto, Wan Fokkink, and Anna Ingólfssdóttir. Ready to preorder: get your BCCSP axiomatization for free! In *Algebra and Coalgebra in Computer Science, Second International Conference, CALCO 2007*, volume 4624 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2007.
- [5] Luca Aceto, Wan Fokkink, Anna Ingólfssdóttir, and Bas Luttik. Finite equational bases in process algebra: Results and open questions. In *Processes, Terms and Cycles*, volume 3838 of *Lecture Notes in Computer Science*, pages 338–367. Springer, 2005.
- [6] Luca Aceto, Anna Ingólfssdóttir, Kim G. Larsen, and Jiri Srba. *Reactive Systems: Modelling, Specification and Verification*. Cambridge University Press, 2007.
- [7] Jos C. M. Baeten. A brief history of process algebra. *Theoretical Computer Science*, 335(2-3):131–146, 2005.
- [8] Jos C. M. Baeten, Twan Basten, and Michel A. Reniers. *Process Algebra: Equational Theories of Communicating Processes (Cambridge Tracts in Theoretical Computer Science)*. Cambridge University Press, 2009.
- [9] Jos C. M. Baeten, Jan A. Bergstra, and Jan Willem Klop. Ready-trace semantics for concrete process algebra with the priority operator. *The Computer Journal*, 30(6):498–506, 1987.
- [10] Jos C.M. Baeten and W.P. Weijland. *Process Algebra*. Cambridge Tracts in Computer Science. Cambridge University Press, 1990.
- [11] Jan A. Bergstra and Jan Willem Klop. Process algebra for synchronous communication. *Information and Control*, 60(1-3):109–137, 1984.
- [12] Mária Bielíková, Gerhard Friedrich, Georg Gottlob, Stefan Katzenbeisser, and György Turán, editors. *SOFSEM 2012: Theory and Practice of Computer Science - 38th Conference on Current Trends in Theory and Practice of Computer Science, pindlerov Mlýn, Czech Republic, January 21-27, 2012. Proceedings*, volume 7147 of *Lecture Notes in Computer Science*. Springer, 2012.
- [13] Bard Bloom, Sorin Istrail, and Albert R. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, 1995.
- [14] Stephen D. Brookes, C.A.R. Hoare, and A. William Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, 1984.
- [15] Antonio Cerone and Pekka Pihlajasaari, editors. *Theoretical Aspects of Computing - ICTAC 2011 - 8th International Colloquium, Johannesburg, South Africa, August 31 - September 2, 2011. Proceedings*, volume 6916 of *Lecture Notes in Computer Science*. Springer, 2011.
- [16] Taolue Chen, Wan Fokkink, and Rob J. van Glabbeek. Ready to preorder: The case of weak process semantics. *Information Processing Letters*, 109(2):104–111, 2008.
- [17] Rance Cleaveland and Matthew Hennessy. Testing equivalence as a bisimulation equivalence. *Formal Asp. Comput.*, 5(1):1–20, 1993.
- [18] David de Frutos-Escrig, Carlos Gregorio-Rodríguez, and Miguel Palomino. Coinductive characterisations reveal nice relations between preorders and equivalences. In *First International Conference on Foundations of Informatics, Computing and Software (FICS 2008)*, volume 212 of *Electronic Notes in Theoretical Computer Science*, pages 149–162. Elsevier, 2008.
- [19] David de Frutos-Escrig, Carlos Gregorio-Rodríguez, and Miguel Palomino. On the unification of process semantics: Equational semantics. *Electronic Notes in Theoretical Computer Science*, 249:243–267, 2009.
- [20] David de Frutos-Escrig, Carlos Gregorio-Rodríguez, and Miguel Palomino. On the unification of process semantics: Observational semantics. In *SOFSEM 2009: Theory and Practice of Computer Science, 35th Conference on Current Trends in Theory and Practice of Computer Science*, volume 5404 of *Lecture Notes in Computer Science*, pages 279–290. Springer, 2009.
- [21] David de Frutos Escrig, Carlos Gregorio-Rodríguez, and Miguel Palomino. Ready to preorder: an algebraic and general proof. *Journal of Logic and Algebraic Programming*, 78(7):539–551, 2009. doi:10.1016/j.jlap.2008.09.001.
- [22] David de Frutos-Escrig and Carlos Gregorio-Rodríguez. Bisimulations up-to for the linear time-branching time spectrum. In *CONCUR 2005 - Concurrency Theory, 16th International Conference*, volume 3653 of *Lecture Notes in Computer Science*, pages 278–292. Springer, 2005.
- [23] David de Frutos-Escrig and Carlos Gregorio-Rodríguez. Simulations up-to and canonical preorders (extended abstract). In *Structural Operational Semantics SOS 2007*, volume 192 of *Electronic Notes in Theoretical Computer Science*, pages 13–28. Elsevier, 2007.

- [24] David de Frutos-Escrig and Carlos Gregorio-Rodríguez. Universal coinductive characterizations of process semantics. In *5th IFIP International Conference on Theoretical Computer Science*, volume 273 of *IFIP*, pages 397–412. Springer, 2008.
- [25] David de Frutos-Escrig and Carlos Gregorio-Rodríguez. (Bi)simulations up-to characterise process semantics. *Information and Computation*, 207(2):146–170, 2009.
- [26] David de Frutos-Escrig, Miguel Palomino, and Ignacio Fábregas. Multiset bisimulations as a common framework for ordinary and probabilistic bisimulations. In *Formal Techniques for Networked and Distributed Systems - FORTE 2008, 28th IFIP WG 6.1 International Conference*, volume 5048 of *Lecture Notes in Computer Science*, pages 283–298. Springer, 2008.
- [27] Cédric Fournet, Tony Hoare, Sriram K. Rajamani, and Jakob Rehof. Stuck-free conformance. In Rajeev Alur and Doron A. Peled, editors, *Computer Aided Verification. 16th International Conference, CAV 2004, Boston, MA, USA, July 13–17, 2004. Proceedings*, volume 3114 of *Lecture Notes in Computer Science*, pages 242–254. Springer, 2004.
- [28] Per Brinch Hansen. *The Origins of Concurrent Programming: From Semaphores to Remote Procedure Calls*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.
- [29] Matthew Hennessy. Acceptance trees. *Journal of the ACM*, 32(4):896–928, 1985.
- [30] Matthew Hennessy. *Algebraic Theory of Processes*. MIT Press, 1988.
- [31] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32:137–161, 1985.
- [32] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
- [33] Tony Hoare. *Process Algebra: A Unifying Approach*, volume 3525 of *Lecture Notes in Computer Science*, pages 36–60. Springer Berlin / Heidelberg, 2005.
- [34] Tony Hoare and He Jifeng. *Unifying Theories of Programming*. Prentice Hall, 1998.
- [35] Jesse Hughes and Bart Jacobs. Simulations in coalgebra. *Theoretical Computer Science*, 327(1–2):71–108, 2004.
- [36] Bartek Klin. A coalgebraic approach to process equivalence and a coinductive principle for traces. In *CMCS'04: 7th International Workshop on Coalgebraic Methods in Computer Science*, volume 106 of *Electronic Notes in Theoretical Computer Science*, pages 201–218. Elsevier, 2004.
- [37] Antonín Kucera and Richard Mayr. Why is simulation harder than bisimulation? In *CONCUR 2002 - Concurrency Theory, 13th International Conference, Proceedings*, volume 2421 of *Lecture Notes in Computer Science*, pages 594–610. Springer, 2002.
- [38] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing (preliminary report). In *Principles of Programming Languages, 16th ACM SIGACT-SIGPLAN Symposium - POPL '89*, pages 344–352. ACM Press, 1989.
- [39] Chris Verhoef Luca Aceto, Wan Fokkink. *Handbook of Process Algebra*, chapter 3, Structural operational semantics, pages 197–292. Elsevier, 2001.
- [40] Gerald Lüttgen and Walter Vogler. Ready simulation for concurrency: It's logical! *Information and Computation*, 208(7):845–867, 2010.
- [41] Robin Milner. *A Calculus of Communicating Systems*. LNCS 92. Springer, 1980.
- [42] Robin Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [43] Ernst R. Olderog and C.A.R. Hoare. Specification-oriented semantics for communicating processes. *Acta Informatica*, 23(1):9–66, 1986.
- [44] Iain Phillips. Refusal testing. *Theoretical Computer Science*, 50(3):241–284, 1987.
- [45] Gordon D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:17–139, 2004.
- [46] Joy N. Reed, A. William Roscoe, and Jane E. Sinclair. Responsiveness and stable revivals. *Formal Aspects of Computing*, 19:303–319, 2007.
- [47] David Romero-Hernández and David de Frutos-Escrig. On the unification of process semantics: Logical semantics. In *Structural Operational Semantics, SOS'11*, volume 62 of *EPTCS*, pages 47–61, 2011.
- [48] A. William Roscoe. Revivals, stuckness and the hierarchy of CSP models. *Journal of Logic and Algebraic Programming*, 78(3):163–190, 2009.
- [49] A. William Roscoe. *Understanding Concurrent Systems (Texts in Computer Science)*. Springer, 2010.
- [50] Davide Sangiorgi. On the origins of bisimulation and coinduction. *ACM Trans. Program. Lang. Syst.*, 31(4):41 pages, 2009.
- [51] Davide Sangiorgi. *Introduction to Bisimulation and Coinduction*. Cambridge University Press, 2012.

- [52] Dana Scott and Christopher Strachey. Towards a mathematical semantics for computer languages. Programming Research Group Technical Monograph PRG-6, Oxford Univ. Computing Lab., 1971.
- [53] Anti Valmari. All linear-time congruences for finite ltss and familiar operators. In *Application of Concurrency to System Design (ACSD), 2012 12th International Conference on*, pages 12–21. IEEE, 2012.
- [54] Rob J. van Glabbeek. *Comparative Concurrency Semantics and Refinement of Actions*. PhD thesis, Free University, Amsterdam, 1990. Second edition available as *CWI tract 109*, CWI, Amsterdam 1996.
- [55] Rob J. van Glabbeek. The linear time-branching time spectrum. In *CONCUR '90 Theories of Concurrency: Unification and Extension*, number 458 in *Lecture Notes in Computer Science*, pages 278–297. Springer-Verlag, 1990.
- [56] Rob J. van Glabbeek. The linear time - branching time spectrum II. In *CONCUR '93 - Concurrency Theory, 5th International Conference*, volume 715 of *Lecture Notes in Computer Science*, pages 66–81. Springer, 1993.
- [57] Rob J. van Glabbeek. Notes on the methodology of ccs and csp. *Theoretical Computer Science*, 177(2):329–349, 1997.
- [58] Rob J. van Glabbeek. *Handbook of Process Algebra*, chapter 1, The Linear Time – Branching Time Spectrum I: The Semantics of Concrete, Sequential Processes, pages 3–99. Elsevier, 2001.
- [59] Marc Voorhoeve and Sjouke Mauw. Impossible futures and determinism. *Information Processing Letters*, 80(1):51–58, 2001.