Relating Reasoning Methodologies in Linear Logic and Process Algebra

Yuxin Deng **Robert J. Simmons**
Iliano Cervesato

December 2011
CMU-CS-11-145
CMU-CS-QTR-111

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

The authors can be reached at yuxindeng@sjtu.edu.cn, rjsimmon@cs.cmu.edu, and iliano@cmu.edu.

Abstract

We show that the proof-theoretic notion of logical preorder coincides with the process-theoretic notion of contextual preorder for a CCS-like process calculus obtained from the formula-as-process interpretation of a fragment of linear logic. The argument makes use of other standard notions in process algebra, namely simulation and labeled transition systems. This result establishes a connection between an approach to reason about process specifications, the contextual preorder, and a method to reason about logic specifications, the logical preorder.

Support for this research was provided by the Qatar National Research Fund under NPRP grant 09-1107-1-168, by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the Carnegie Mellon Portugal Program under Grant NGN-44, and by an X10 Innovation Award from IBM. Yuxin Deng would also like to acknowledge the support of the Natural Science Foundation of China under grant 61173033 held by Shanghai Jiao Tong University.

Keywords: Linear logic; process algebra; meta-reasoning; logical preorder; contextual preorder; simulation; labeled transition systems; equivalence; induction; coinduction.

Contents

1 Introduction **1**

2 First-Order Intuitionistic Linear Logic **2**
2.1 Metatheory of linear logic 3
2.2 The logical preorder 4
2.3 Relating cut, identity, and the logical preorder 6
2.4 Re-characterizing the logical preorder 7

3 Process Interpretation and the Contextual Preorder **8**
3.1 The contextual preorder 10
3.2 Properties of the contextual preorder 13

4 Labeled Transitions and the Simulation Preorder **19**
4.1 Labeled transitions 19
4.2 Examples 21
4.3 Properties of labeled transitions 22
4.4 Properties of the simulation preorder 23
4.5 Equivalence of the contextual and simulation preorders 27
4.6 Equivalence of the logical and simulation preorders 28

5 Exponentials **31**
5.1 Process interpretation and contextual preorder 32
5.2 Logical preorder and contextual preorder 33

6 Concluding remarks **39**

References **40**

1 Introduction

By now, execution-preserving relationships between (fragments of) linear logic and (fragments of) process algebras are well-established (see [5] for an overview). Abramsky observed early on that linear cut elimination resembles reduction in CCS and the π-calculus [21], thereby identifying processes with (some) linear proofs and establishing the *process-as-term* interpretation [1]. The alternative *process-as-formula* encoding, pioneered by Miller around the same time [20], maps process constructors to logical connectives and quantifiers, with the effect of relating reductions in process algebra with proof steps, in the same way that logic programming achieves computation via proof search. Specifically, it describes the state of an evolving concurrent system as a linear logic context. Transitions between such process states are therefore modeled as transitions between linear logic contexts. As a member of a context, a formula stands for an individual process in the process state. On the right-hand side of an intuitionistic derivability judgment, it is a specification that a process state can satisfy. The process-as-formula interpretation has been used extensively in a multitude of domains, in particular in the fields of programming languages [4, 5, 20] and security [3]. For example, [5] developed it into a logically-based rewriting formalism that subsumes and integrates both process-based (e.g., the π-calculus) and transition-based languages (e.g., Petri nets) for specifying concurrent systems.

Not as well established is the relationship between the rich set of notions and techniques used to reason about process specifications and the equally rich set of techniques used to reason about (linear) logic. Indeed, a majority of investigations have attempted to reduce some of the behavioral notions that are commonplace in process algebra to derivability within logic. For example, Miller identified a fragment of linear logic that could be used to observe traces in his logical encoding of the π-calculus, thereby obtaining a language that corresponds to the Hennessy-Milner modal logic, which characterizes observational equivalence [20]. A similar characterization was made in [17], where a sequent $\Gamma \vdash \Delta$ in a classical logic augmented with constraints was seen as process state Γ passing test Δ. Extensions of linear logic were shown to better capture other behavioral relations: for example, adding definitions allows expressing simulation as the derivability of a linear implication [19], but falls short of bisimulation, for which a nominal logic is instead an adequate formalism [26].

This body of work embeds approaches for reasoning *about* process specifications (e.g., bisimulation or various forms of testing) into methods for reasoning *with* logic (mainly derivability). Little investigation has targeted notions used to reason about logic (e.g., proof-theoretic definitions of equivalence). More generally, techniques and tools developed for each formalism rarely cross over — and may very well be rediscovered in due time. Tellingly, process-based definitions and proofs are often coinductive in nature, while techniques based on proof-theory are generally inductive.

This report outlines one such relationship — between the inductive methods used to reason about logic and the coinductive methods used to reason about process calculi. On the linear logic side, we focus on the inductively-defined notion of logical preorder (obtained by forsaking symmetry from derivability-based logical equivalence). On the process-algebraic side, we consider an extensional behavioral relation adapted from the standard coinductive notion of contextual preorder. We prove that, for two fragments of linear logic and matching process calculi, these notions

$$\frac{}{\Gamma; a \vdash a}\ init \qquad \frac{\Gamma, A; \Delta, A \vdash C}{\Gamma, A; \Delta \vdash C}\ clone$$

$$\frac{}{\Gamma; \cdot \vdash \mathbf{1}}\ \mathbf{1}R \qquad \frac{\Gamma; \Delta \vdash C}{\Gamma; \Delta, \mathbf{1} \vdash C}\ \mathbf{1}L$$

$$\frac{\Gamma; \Delta_1 \vdash A \quad \Gamma; \Delta_2 \vdash B}{\Gamma; \Delta_1, \Delta_2 \vdash A \otimes B}\ \otimes R \qquad \frac{\Gamma; \Delta, A, B \vdash C}{\Gamma; \Delta, A \otimes B \vdash C}\ \otimes L$$

$$\frac{\Gamma; \Delta, A \vdash B}{\Gamma; \Delta \vdash A \multimap B}\ \multimap R \qquad \frac{\Gamma; \Delta_1 \vdash A \quad \Gamma; \Delta_2, B \vdash C}{\Gamma; \Delta_1, \Delta_2, A \multimap B \vdash C}\ \multimap L$$

$$\frac{}{\Gamma; \Delta \vdash \top}\ \top R \qquad \textit{(no rule } \top L\textit{)}$$

$$\frac{\Gamma; \Delta \vdash A \quad \Gamma; \Delta \vdash B}{\Gamma; \Delta \vdash A \,\&\, B}\ \&R \qquad \frac{\Gamma; \Delta, A_i \vdash C}{\Gamma; \Delta, A_1 \,\&\, A_2 \vdash C}\ \&L_i$$

$$\frac{\Gamma; \cdot \vdash A}{\Gamma; \cdot \vdash\, !A}\ !R \qquad \frac{\Gamma, A; \Delta \vdash C}{\Gamma; \Delta, !A \vdash C}\ !L$$

Figure 1: Dual intuitionistic linear logic

coincide. Our proofs rely on other standard process algebraic notions as stepping stones, namely simulation and labeled transition systems.

The rest of the report is organized as follows. In Section 2, we briefly review the general fragment of linear logic we are focusing on and define the logical preorder. Then, in Section 3, we recall the standard process-as-formula interpretation for a sublanguage without exponential and define the notion of contextual preorder. In Section 4, we prove their equivalence through the intermediary of a simulation preorder defined on the basis of a labeled transition system. In Section 5, we extend our results and proof techniques to accommodate exponentials. We conclude in Section 6 by anticipating future development enabled by this work.

2 First-Order Intuitionistic Linear Logic

The starting point for our investigation will be intuitionistic linear logic [12] *sans* disjunction $A \oplus B$ and its unit $\mathbf{0}$. The formulas of this fragment of propositional intuitionistic linear logic are defined by the following grammar (a is an atomic formula):

$$A, B, C \ ::= \ a \mid \mathbf{1} \mid A \otimes B \mid A \multimap B \mid \top \mid A \,\&\, B \mid\, !A$$

Intuitionistic derivability for this fragment is expressed by means of sequents of the form

$$\Gamma; \Delta \vdash A$$

where the *unrestricted context* Γ and the *linear context* Δ are multisets of formulas. We define contexts by means of the following grammar:

$$\Gamma, \Delta \quad ::= \quad \cdot \mid \Delta, A$$

where "$\cdot$" represents the empty context, and "," denotes the context extension operation: Δ, A is the context obtained by adding the formula A to the context Δ. As usual, we will tacitly treat "," as an associative and commutative context union operator "Δ_1, Δ_2" with the unit "$\cdot$", which indeed allows us to think of contexts as multisets.

Derivability in intuitionistic linear logic is defined by the inference rules in Figure 1, which follow a standard presentation of propositional intuitionistic linear logic, called Dual Intuitionistic Linear Logic (or DILL) [2, 5]. A sequent $\Gamma; \Delta \vdash A$ is derivable if there is a valid derivation with $\Gamma; \Delta \vdash A$ as its root. A DILL sequent $\Gamma; \Delta \vdash A$ corresponds to $!\Gamma, \Delta \vdash A$ in Girard's original presentation [12]. In the first part of this report, we will be working with the sublanguage of linear logic where the unrestricted context Γ is empty and there are no occurrences of the formula $!A$. We will then abbreviate the purely linear sequent $\cdot; \Delta \vdash A$ as $\Delta \vdash A$.

The remainder of the section is as follows: in Section 2.1 we discuss the fundamental metatheoretic results for dual intuitionistic linear logic — the core of the "proof theorist's toolkit." In Section 2.2, we introduce the *logical preorder*, a natural semantic relationship between contexts that will be our connection to the exploration of the "process calculist's toolkit" in future sections. In Section 2.3, we reveal the nice connection between these two approaches. In Section 2.4, we give a purely inductive characterization of the logical preorder.

2.1 Metatheory of linear logic

Dual intuitionistic linear logic is sound (there is no closed proof of a contradiction) and complete (we can use an assumption of A to prove A). This is proved using Gentzen's methodology, the aforementioned standard toolkit of a working proof theorist [11, 22]. The relevant theorems are these:

Proposition 2.1 (Weakening)**.** *If* $\Gamma; \Delta \vdash A$*, then* $\Gamma, \Gamma'; \Delta \vdash A$ *for any* Γ'.

Proof. By induction on the structure of the given derivation. □

Proposition 2.2 (Identity)**.** $\Gamma; A \vdash A$ *for all formulas* A *and unrestricted contexts* Γ.

Proof. By induction on the structure of the formula A. □

This proposition entails that the following "identity" rule is admissible:

$$\frac{}{\Gamma; A \vdash A}\ id$$

Proposition 2.3 (Cut admissibility)**.** *If* $\Gamma; \Delta \vdash A$ *and* $\Gamma'; \Delta', A \vdash C$*, then* $\Gamma', \Gamma; \Delta', \Delta \vdash C$.

Proof. We generalize the induction hypothesis by proving the statement of the theorem together with the statement "*if* $\Gamma; \cdot \vdash A$ *and* $\Gamma', A; \Delta' \vdash C$*, then* $\Gamma', \Gamma; \Delta' \vdash C$." Then the proof proceeds by mutual, lexicographic induction, first on the structure of the formula A, and second by induction on the structure of the two given derivations. □

Proposition 2.3 is called the "cut admissibility" theorem because the proof indicates that these two "cut" rules are admissible:

$$\frac{\Gamma; \Delta \vdash A \quad \Gamma'; \Delta', A \vdash C}{\Gamma', \Gamma; \Delta', \Delta \vdash C}\ cut \qquad \frac{\Gamma; \cdot \vdash A \quad \Gamma', A; \Delta' \vdash C}{\Gamma', \Gamma; \Delta' \vdash C}\ cut!$$

From the proof theory perspective, Propositions 2.2 and 2.3 are canonical and establish fundamental properties of the logic; any system of rules not validating the identity and cut admissibility theorems can hardly be called a logic. One relevant point about the *proofs* of Propositions 2.2 and 2.3 is that they are quite modular with respect various sublanguages of linear logic. Essentially any syntactic restriction of the language of formulas, and certainly every restriction considered in this report, preserves the validity of the identity and cut properties. They also hold for larger fragments of linear logic, for example with additive disjunction or quantifiers (this latter case requires upgrading the derivability judgment with a context Σ of first-order variables and therefore the statements of these properties).

2.2 The logical preorder

The study of process calculi is primarily concerned with the relationships between different processes, but the only judgment that we have so far for linear logic is the derivability judgment $\Gamma; \Delta \vdash A$, which expresses the judgment that the process state $(\Gamma; \Delta)$ satisfies the formula or specification A.

In the formulation of logics for process calculi, a central judgment is $P \models \phi$, which states that the process P satisfies some formula ϕ. This leads to a natural definition of a *logical preorder*, where $P \preceq_l Q$ if $P \models \phi$ implies $Q \models \phi$ for all ϕ; in other words, if the set of formulas satisfied by the former is included in the set of formulas satisfied by the latter. See e.g. [14] for an example in classical process algebra and [8, 6] in probabilistic process algebra. Our goal is to give a definition of $P \preceq_l Q$ where P and Q are process states.

If we choose to look at the derivability judgment $\Gamma; \Delta \vdash A$ as analogous to the judgment $P \models \phi$, then it gives us an obvious way of relating process states: a specific process state $(\Gamma_{specific}; \Delta_{specific})$ is generalized by a general process state $(\Gamma_{general}; \Delta_{general})$ if all formulas A satisfied by the specific process are also satisfied by the general process. In other words, we can define a preorder $(\Gamma_{specific}; \Delta_{specific}) \preceq_l (\Gamma_{general}; \Delta_{general})$, which we will also call the logical preorder, which holds if $\Gamma_{specific}; \Delta_{specific} \vdash C$ implies $\Gamma_{general}; \Delta_{general} \vdash C$ for all C.

This is an intuitively reasonable definition, but, as we explain below, that definition requires us to assume some specific properties of the logic. By giving a slightly more general (but ultimately equivalent) definition of the logical preorder, we can avoid making *a priori* assumptions about the relationship between logical derivability and the logical preorder.

Definition 2.4 (Logical preorder)**.** *The* logical preorder *is the smallest relation $\preceq_l$ on states such that $(\Gamma_1; \Delta_1) \preceq_l (\Gamma_2; \Delta_2)$ if, for all Γ', Δ', and C, we have that $\Gamma', \Gamma_1; \Delta', \Delta_1 \vdash C$ implies $\Gamma', \Gamma_2; \Delta', \Delta_2 \vdash C$.*

The only difference between our informal motivation and Definition 2.4 is that, in the latter, the specific and general process states must both satisfy the same formula C after being *extended* with the same unrestricted context Γ' and linear context Δ'.

Because the $\multimap R$ and $!L$ rules are invertible (for each rule, the conclusion implies the premise), this definition is equivalent to the definition that does not use extended contexts. This is because, if $\Gamma' = A_1, \ldots, A_n$ and $\Delta' = B_1, \ldots, B_m$, then the judgment $\Gamma', \Gamma; \Delta', \Delta \vdash C$ can be derived if and only if the judgment $\Gamma; \Delta \vdash !A_1 \multimap \ldots \multimap !A_n \multimap B_1 \multimap \ldots \multimap B_m \multimap C$ can be derived. However, the proof of invertibility relies on the metatheory of linear logic as presented in the previous section. The more "contextual" version of Definition 2.4 allows us to talk about the relationship between the derivability judgment and the logical preorder without baking in any assumptions about invertibility.

Altogether, we are taking the view, common in practice, that the context part of the sequent $(\Gamma; \Delta)$ represents the state of some system component and that the consequent A corresponds to some property satisfied by this system. Then, the logical preorder compares specifications on the basis of the properties they satisfy, possibly after the components they describe are plugged into a larger system.

Theorem 2.5. *The logical preorder $\preceq_l$ is a preorder.*

Proof. We need to show that $\preceq_l$ is reflexive and transitive.

Reflexivity: Expanding the definition, the reflexivity statement, $(\Gamma; \Delta) \preceq_l (\Gamma; \Delta)$ for all A assumes the form "*for all Γ', Δ' and A if $\Gamma, \Gamma'; \Delta, \Delta' \vdash A$, then $\Gamma, \Gamma'; \Delta, \Delta' \vdash A$*". This holds trivially.

Transitivity: We want to prove that if $(\Delta_1; \Gamma_1) \preceq_l (\Delta_2; \Gamma_2)$ and $(\Delta_2; \Gamma_2) \preceq_l (\Delta_3; \Gamma_3)$ then $(\Delta_1; \Gamma_1) \preceq_l (\Delta_3; \Gamma_3)$. By the definition of $\preceq_l$, we know that for any Γ', Δ' and A', if $\Gamma_1, \Gamma'; \Delta_1, \Delta' \vdash A'$ then $\Gamma_2, \Gamma'; \Delta_2, \Delta' \vdash A'$. Similarly, for any Γ'', Δ'' and A'', if $\Gamma_2, \Gamma''; \Delta_2, \Delta'' \vdash A''$ then $\Gamma_3, \Gamma''; \Delta_3, \Delta'' \vdash A''$. If we choose Γ'', Δ'' and A'' to be precisely Γ', Δ' and A' respectively, we can chain these two implications, obtaining that for any Γ', Δ' and A', if $\Gamma_1, \Gamma'; \Delta_1, \Delta' \vdash A'$ then $\Gamma_3, \Gamma'; \Delta_3, \Delta' \vdash A'$. This is however exactly the definition of logical preorder: therefore $(\Delta_1; \Gamma_1) \preceq_l (\Delta_3; \Gamma_3)$.

Therefore $\preceq_l$ is indeed a preorder. □

Observe that this proof is independent of the actual rules defining derivability. Indeed, it follows only from the definition of this relation and the fact that informal implication is itself reflexive and transitive. This means that $\preceq_l$ is a preorder for any fragment of linear logic we may care to study.

2.3 Relating cut, identity, and the logical preorder

We have now defined two judgments. The derivability judgment of linear logic, written as $\Gamma; \Delta \vdash A$, declares that the process state $(\Gamma; \Delta)$ meets the specification set by A. The logical preorder $(\Gamma_1; \Delta_1) \preceq_l (\Gamma_2; \Delta_2)$, says that the process state $(\Gamma_1; \Delta_1)$ can act as a specific instance of the more general process state $(\Gamma_2; \Delta_2)$. Recall from the introduction that linear logic formulas have two natures: they are both 1) the specifications that a process state can satisfy and 2) the atomic constituents of a process state. By the convention that $(\cdot; \Delta)$ can be written as Δ, we can also think of the formula A as synonymous with the singleton process state $(\cdot; A)$.

The derivability judgment $\Gamma; \Delta \vdash A$ says that A is one specification that the process state $(\Gamma; \Delta)$ satisfies; of course it may satisfy many other specifications as well — interpreted as a specification, the formula A is specific, and the process state $(\Gamma; \Delta)$ is general. We relate a specific and general process states with the logical preorder: $(\Gamma_{specific}; \Delta_{specific}) \preceq_l (\Gamma_{general}; \Delta_{general})$. This suggests that, if the two natures of a formula are in harmony, we can expect that $\Gamma; \Delta \vdash A$ exactly when $(\cdot; A) \preceq_l (\Gamma; \Delta)$, a suggestion that is captured by the following proposition:

Proposition 2.6 (Harmony for the logical preorder)**.** $\Gamma; \Delta \vdash A$ *if and only if* $A \preceq_l (\Gamma; \Delta)$.

Proof. This will be a simple corollary of Theorem 2.7 below — read on. □

Proposition 2.6 should be seen as a sort of sanity check that the logical preorder's notion of "more specific" and "more general" makes sense relative to the derivability judgment's notion. A result that initially surprised us is that this sanity check is exactly equivalent to the classic sanity checks of the proof theorists: identity and cut admissibility.

Theorem 2.7. *Proposition 2.6 holds if and only if Propositions 2.2 and 2.3 hold.*

Proof. This theorem establishes the equivalence between two propositions; both directions can be established independently.

Assuming harmony, prove identity and cut. For the identity theorem, we must show $A \vdash A$ for some arbitrary formula A. By harmony, it suffices to show $A \preceq_l A$, and $\preceq_l$ is reflexive (Theorem 2.5).

For the cut admissibility theorem, we are given $\Gamma; \Delta \vdash A$ and $\Gamma'; \Delta', A \vdash C$ and must prove $\Gamma', \Gamma; \Delta', \Delta \vdash C$. By harmony and the first given derivation, we have $A \preceq_l (\Gamma; \Delta)$. Expanding Definition 2.4, this means that, for all $\overline{\Gamma'}$, $\overline{\Delta'}$, and $\overline{C}$, we have that $\overline{\Gamma'}; \overline{\Delta'}, A \vdash \overline{C}$ implies $\overline{\Gamma'}, \Gamma; \overline{\Delta'}, \Delta \vdash \overline{C}$. So by letting $\overline{\Gamma'} = \Gamma'$, $\overline{\Delta'} = \Delta'$, and $\overline{C} = C$ and applying the second given derivation $\Gamma'; \Delta', A \vdash C$, we get $\Gamma', \Gamma; \Delta', \Delta \vdash C$, which was what we needed to prove.

Assuming identity and cut, prove harmony. For the forward implication, we are given $\Gamma; \Delta \vdash A$ and we need to prove $A \preceq_l (\Gamma; \Delta)$. Expanding Definition 2.4, this means that, for an arbitrary Γ', Δ', and C, we are given $\Gamma'; \Delta', A \vdash C$ and need to prove $\Gamma', \Gamma; \Delta', \Delta \vdash C$. The statement of cut admissibility, Proposition 2.3, is that if $\Gamma; \Delta \vdash A$ and $\Gamma'; \Delta', A \vdash C$, then $\Gamma', \Gamma; \Delta', \Delta \vdash C$. We have the two premises, and the conclusion is what we needed to prove.

For the reverse implication, we are given $A \preceq_l (\Gamma; \Delta)$ and we need to prove $\Gamma; \Delta \vdash A$. Expanding Definition 2.4, this means that, for all $\overline{\Gamma'}$, $\overline{\Delta'}$, and $\overline{C}$, we have that $\overline{\Gamma'}; \overline{\Delta'}, A \vdash \overline{C}$ implies $\overline{\Gamma'}, \Gamma; \overline{\Delta'}, \Delta \vdash \overline{C}$. So by letting $\overline{\Gamma'} = \cdot$, $\overline{\Delta'} = \cdot$, and $\overline{C} = A$, we have that $A \vdash A$ implies $\Gamma; \Delta \vdash A$. The conclusion is what we needed to show, and the premise is exactly the statement of identity, Proposition 2.2, so we are done. □

It is critical to note that Theorem 2.7 holds *entirely independently* of the actual definition of the derivability judgment $\Gamma; \Delta \vdash A$. This means, in particular, that it holds independently of the actual validity of Propositions 2.2 and 2.3 as they were presented here, and that it holds for any alternative definition of the derivability judgment that we might present. Furthermore, because Propositions 2.2 and 2.3 do, of course, hold for dual intuitionistic linear logic, Proposition 2.6 holds as a simple corollary of Theorem 2.7.

The proof theorist's sanity checks are motivated by arguments that are somewhat philosophical. In Girard's *Proofs and Types*, cut and identity are motivated by an observation that hypotheses and conclusions should have equivalent epistemic strength [13]. Martin Löf's *judgmental methodology* gives a weaker sanity check, *local soundness* [18], which was augmented by Pfenning and Davies with a sanity check of *local completeness* [23]. Local soundness and completeness take the verificationist viewpoint that the meaning of a logical connective is given by its introduction rules in a natural deduction presentation of the logic. This means that the elimination rules must be justified as neither too strong (soundness) nor too weak (completeness) relative to the introduction rules. The surprising (at least, initially, to us) equivalence of Proposition 2.6 to the critical sanity checks of sequent calculi suggests that the process state interpretation of linear logic can actually be treated as fundamental, that is, as a philosophical organizing principle for sequent calculus presentations of logic.

2.4 Re-characterizing the logical preorder

In the previous section, we have argued for the canonicity of the logical preorder by motivating harmony and showing that harmony is equivalent to the canonical properties of cut and identity. However, it is not obvious, given our discussion so far that it is easy or even possible to prove interesting properties of the logical preorder. In this section, we show that there is an alternate characterization of the logical preorder for DILL directly in terms of the existing derivability judgment $\Gamma; \Delta \vdash A$. This re-characterization makes it more obvious that the logical preorder is a fundamentally inductive concept.

This re-characterization depends on the auxiliary concepts of the *tensorial product* of a linear context Δ, written $\bigotimes \Delta$, and *exponential linearization* of an unrestricted context Γ, written $!\Gamma$, which are defined as follows:

$$\begin{cases} \bigotimes(\cdot) & = \mathbf{1} \\ \bigotimes(\Delta, A) & = (\bigotimes \Delta) \otimes A \end{cases} \qquad \begin{cases} !(\cdot) & = \cdot \\ !(\Gamma, A) & = (!\Gamma), !A \end{cases}$$

The main result of this section is given by the following theorem.

Theorem 2.8. $(\Gamma_1; \Delta_1) \preceq_l (\Gamma_2; \Delta_2)$ *iff* $\Gamma_2; \Delta_2 \vdash \bigotimes \Delta_1 \otimes \bigotimes !\Gamma_1$.

Proof. ($\Rightarrow$) Assume that $(\Gamma_1; \Delta_1) \preceq_l (\Gamma_2; \Delta_2)$. Then, by definition, for every Γ, Δ and A such that $\Gamma_1, \Gamma; \Delta_1, \Delta \vdash A$ is derivable, there is a derivation of $\Gamma_2, \Gamma; \Delta_2, \Delta \vdash A$. Take $\Gamma = \Delta = \cdot$ and $A = \bigotimes \Delta_1 \otimes \bigotimes !\Gamma_1$. If we can produce a derivation of $\Gamma_1; \Delta_1 \vdash \bigotimes \Delta_1 \otimes \bigotimes !\Gamma_1$, then our desired result follows. Here is a schematic derivation of this sequent:

$$\dfrac{\dfrac{}{\Gamma_1; \Delta_1 \vdash \bigotimes \Delta_1}\otimes R, \mathbf{1}R, id \quad \dfrac{\cdots \quad \dfrac{\dfrac{\dfrac{}{\Gamma_1; C_i \vdash C_i}\, id}{\Gamma_1; \cdot \vdash C_i}\, clone}{\Gamma_1; \cdot \vdash !C_i}\, !R \quad \cdots}{\Gamma_1; \cdot \vdash \bigotimes !\Gamma_1}\otimes R, \mathbf{1}R}{\Gamma_1; \Delta_1 \vdash \bigotimes \Delta_1 \otimes \bigotimes !\Gamma_1}\otimes R$$

Here, we assume that Γ_1 expands to $C_1, \ldots, C_i, \ldots, C_n$ for an appropriate $n \geq 0$.

($\Leftarrow$) Assume now that $\Gamma_2; \Delta_2 \vdash \bigotimes \Delta_1 \otimes \bigotimes !\Gamma_1$ with derivation $\mathcal{D}$. To show that $\Delta_1 \preceq_l \Delta_2$, we need to show that, given Γ, Δ, A and a derivation $\mathcal{D}_1$ of $\Gamma_1, \Gamma; \Delta_1, \Delta \vdash A$, we can construct a derivation of $\Gamma_2, \Gamma; \Delta_2, \Delta \vdash A$.

To do so, we start by weakening $\mathcal{D}$ with Γ and $\mathcal{D}_1$ with Γ_2 by means of Proposition 2.1. Let $\mathcal{D}'$ and $\mathcal{D}'_1$ be the resulting derivations of $\Gamma_2, \Gamma; \Delta_2 \vdash \bigotimes \Delta_1 \otimes \bigotimes !\Gamma_1$ and $\Gamma_2, \Gamma_1, \Gamma; \Delta_1, \Delta \vdash A$ respectively. We then build the desired derivation schematically as follows:

$$\dfrac{\mathcal{D}' \qquad \dfrac{\mathcal{D}'_1}{\dfrac{\Gamma_2, \Gamma_1, \Gamma; \Delta_1, \Delta \vdash A}{\Gamma_2, \Gamma; \Delta_1, !\Gamma_1, \Delta \vdash A}\, !L}}{\dfrac{\Gamma_2, \Gamma; \Delta_2 \vdash \bigotimes \Delta_1 \otimes \bigotimes !\Gamma_1 \qquad \Gamma_2, \Gamma; \bigotimes \Delta_1 \otimes \bigotimes !\Gamma_1, \Delta \vdash A}{\Gamma_2, \Gamma; \Delta_2, \Delta \vdash A}\, cut}\otimes L, \mathbf{1}L$$

□

This replaces an extensional test that considers arbitrary contexts and goal formulas with an existential test that only requires exhibiting a single derivation. This makes it evident that the logical preorder is a semi-decidable relation. It is interesting to specialize this result to the fragment of our language without exponentials nor permanent contexts. We obtain:

Corollary 2.9. $\Delta_1 \preceq_l \Delta_2$ *iff* $\Delta_2 \vdash \bigotimes \Delta_1$.

For this language fragment, it is decidable whether $\Delta \vdash A$ has a derivation. Therefore, the logical preorder relation is decidable too: given Δ_1 and Δ_2, it is decidable whether $\Delta_1 \preceq_l \Delta_2$.

3 Process Interpretation and the Contextual Preorder

The remainder of this paper will explore the relationship between the logical preorder just introduced and a second relation, the contextual preorder, that emerges from trying to directly understand transitions on linear logic process states. We will do so gradually. Indeed, this section and the next will concentrate on a fragment of intuitionistic linear logic as presented in Section 2. This language is given by the following grammar:

$$\textit{Formulas} \qquad A, B, C \;::=\; a \mid \mathbf{1} \mid A \otimes B \mid a \multimap B \mid \top \mid A \,\&\, B$$

This fragment is purely linear (there is no exponential $!A$) and the antecedents of linear implications are required to be atomic. Our derivability judgment will always have an empty unrestricted context. Therefore, we will write it simply $\Delta \vdash A$. As a consequence, the definition of logical preorder simplifies to "$\Delta_1 \preceq_l \Delta_2$ iff, for all Δ and A, if $\Delta_1, \Delta \vdash A$ then $\Delta_2, \Delta \vdash A$.

This section is written from a process calculus perspective; we will forget, for now, that we presented a sequent calculus in Figure 1 that assigns meaning to the propositions of this fragment of linear logic. In that story, the meaning of propositions is given by the definition of derivability $\Delta \vdash A$ which treats A as a specification that the process state Δ satisfies. In this story, we will try to understand propositions directly by describing their behavior as constituents of a process state directly. The result is somewhat analogous to a fragment of CCS [21] with CSP-style internal choice [15]:

a	atomic process that sends a
$\mathbf{1}$	null process
$A \otimes B$	process that forks into processes A and B
$a \multimap B$	process that receives a and continues as B
$\top$	stuck process
$A \,\&\, B$	process that can behave either as A or as B

According to this interpretation, a process state Δ is a system of interacting processes represented by formulas. Then, the empty context "$\cdot$" is interpreted as a null process while the context constructor "," is a top-level parallel composition. This structure, which makes contexts commutative monoids, is interpreted as imposing a structural equivalence among the corresponding systems of processes. We write this equivalence as $\equiv$ when stressing this interpretation. It is the least relation satisfying the following equations:

$$\begin{array}{rcl} \Delta, \cdot & \equiv & \Delta \\ \Delta_1, \Delta_2 & \equiv & \Delta_2, \Delta_1 \\ \Delta_1, (\Delta_2, \Delta_3) & \equiv & (\Delta_1, \Delta_2), \Delta_3 \end{array}$$

We will always consider contexts modulo this structural equality, and therefore treat equivalent contexts as syntactically identical.

The analogy with CCS above motivates the reduction relation between process states defined in Figure 2. A formula $A \otimes B$ (parallel composition) transitions to the two formulas A and B in parallel (rule $\rightsquigarrow\otimes$), for instance, and a formula $A \,\&\, B$ (choice) either transitions to A or to B (rules $\rightsquigarrow\&_1$ and $\rightsquigarrow\&_2$). The rule corresponding to implication is also worth noting: a formula $a \multimap B$ can interact with an atomic formula a to produce the formula B; we think of the atomic formula a as sending a message asynchronously and $a \multimap B$ as receiving that message.

Proof theory interlude A proof theorist might recognize the strong relationship between the rules in Figure 2 and the left rules from the sequent calculus presentation in Figure 1. This relationship has been studied in details in [5]. It is made explicit in the current setting by the following proposition:

Proposition 3.1. *If $\Delta \rightsquigarrow \Delta'$ and $\Delta' \vdash C$, then $\Delta \vdash C$.*

$$
\begin{array}{rcll}
(\Delta, \mathbf{1}) & \rightsquigarrow & \Delta & (\rightsquigarrow \mathbf{1}) \\
(\Delta, A \otimes B) & \rightsquigarrow & (\Delta, A, B) & (\rightsquigarrow \otimes) \\
(\Delta, A \,\&\, B) & \rightsquigarrow & (\Delta, A) & (\rightsquigarrow \&_1) \\
(\Delta, A \,\&\, B) & \rightsquigarrow & (\Delta, B) & (\rightsquigarrow \&_2) \\
(\Delta, a, a \multimap B) & \rightsquigarrow & (\Delta, B) & (\rightsquigarrow \multimap) \\
\multicolumn{4}{c}{\textit{(No rule for } \top\textit{)}}
\end{array}
$$

Figure 2: The transition formulation of a fragment of linear logic

Proof. This proof proceeds by case analysis on the reduction $\Delta \rightsquigarrow \Delta'$. Most of the cases are straightforward, we give two. If we have a reduction by $\rightsquigarrow \&_2$, then we have $\Delta, A \vdash C$ and must prove $\Delta, A \,\&\, B \vdash C$, which we can do by rule $\&_{L2}$. Alternatively, if we have a reduction by $\rightsquigarrow \multimap$, then we have $\Delta \vdash B$ and must prove $\Delta, a, a \multimap B \vdash C$. We can prove $a \vdash a$ by *init*, and therefore the result follows by $\multimap_L$. □

This theorem can also be understood as demonstrating the admissibility of the following rules:

$$\frac{\Delta \rightsquigarrow \Delta' \quad \Delta' \vdash A}{\Delta \vdash A}\ \mathit{transition}$$

Let $\rightsquigarrow^*$ be the reflexive and transitive closure of $\rightsquigarrow$. Then an easy induction proves a variant of Proposition 3.1 that strengthens $\rightsquigarrow$ to $\rightsquigarrow^*$. It yields the following admissible rule.

$$\frac{\Delta \rightsquigarrow^* \Delta' \quad \Delta' \vdash A}{\Delta \vdash A}\ \mathit{transition}^*$$

Another statement that we could prove is that, given the $\mathit{transition}$ rule or the $\mathit{transition}^*$ rule, all the left rules of the sequent calculus are admissible. Such a proof would be straightforward; the only interesting case is proving $\multimap L$ admissible. However, we will not discuss this property any further in this report. (This concludes our proof theory interlude.)

We will now define the contextual preorder in Section 3.1 and explore some of its properties in Section 3.2.

3.1 The contextual preorder

As we once again forget that we know anything about the proof-theoretic notion of derivability, we turn to the following question: what does it mean for one process state to behave like another process state?[1] Any relation $\mathcal{R}$ which declares $\Delta_1 \ \mathcal{R} \ \Delta_2$ when Δ_1's behavior can be imitated by

[1]The following discussion is intended to be a gentle introduction to some core ideas in process calculus. We want to stress that, with the exception of partition preservation, this section discusses standard and elementary concepts in the process calculus literature.

Δ_2 will be called a *behavioral preorder*,[2] and we will describe a set of desiderata for what makes a good behavioral preorder.

The most fundamental thing a process can do is to be observed; observations are called "barbs" in the language of the process calculists. We'll start with the idea that we can observe only the presence of an atomic proposition in a context. Therefore, if we want to claim (Δ_1, a) is imitated by Δ_2 (that is, if $(\Delta_1, a)\ \mathcal{R}\ \Delta_2$ for some behavioral preorder $\mathcal{R}$), then we should be able to observe the presence of a in Δ_2. But it is not quite right to require that $\Delta_2 \equiv (\Delta_2', a)$; we need to give the process state Δ_2 some time to compute before it is forced to cough up an observation. For this reason we introduce the auxiliary notation $\Delta \Downarrow_a$, which says that $\Delta \rightsquigarrow^* (\Delta', a)$ for some Δ'. This auxiliary notion lets us define our first desiderata: **behavioral preorders should be barb-preserving**. That is, if we say that (Δ_1, a) can be imitated by Δ_2, then it had better be the case that $\Delta_2 \Downarrow_a$.

The next two desiderata do not require any auxiliary concepts. If a process Δ_1 is imitated by Δ_2, and $\Delta_1 \rightsquigarrow \Delta_1'$, then we should expect Δ_2 to be able to similarly evolve to a state that imitates Δ_1'. In other words, **behavioral preorders should be reduction-closed**. The third desiderata has to do with surrounding contexts. If Δ_1 is imitated by Δ_2, then if we put both Δ_1 and Δ_2 in parallel with some other process state Δ', then (Δ_2, Δ') should still imitate (Δ_1, Δ'). In other words, **behavioral preorders should be compositional**.

These three desiderata — barb-preservation, reduction-closure, and compositionality — are standard ideas in what we have been calling the "process calculist's toolkit," and it would be possible to define behavioral preorders entirely in terms of these three desiderata. To foreshadow the developments of the next section, we will eventually want to show that the logical preorder $\Delta_1 \preceq_l \Delta_2$ is *sound* — that it is a behavioral preorder according to these desiderata. This would be provable: the logical preorder *is* barb-preserving, reduction-closed, and compositional. However, the logical preorder is *incomplete* with respect to these three desiderata. Here is a sketch of the reason why: there is a barb-preserving, reduction-closed, and compositional behavioral preorder which would say that the process state $(a \multimap \mathbf{1}, b \multimap \mathbf{1})$ is imitated by the process state $(a \multimap b \multimap \mathbf{1})$.[3] However, $(a \multimap \mathbf{1}, b \multimap \mathbf{1}) \not\preceq_l (a \multimap b \multimap \mathbf{1})$ because $a \multimap \mathbf{1}, b \multimap \mathbf{1} \vdash (a \multimap \mathbf{1}) \otimes (b \multimap \mathbf{1})$ but $a \multimap b \multimap \mathbf{1} \nvdash (a \multimap \mathbf{1}) \otimes (b \multimap \mathbf{1})$.

The logical preorder is, to a degree, fixed and canonical due to its relationship with the standard metatheory of the sequent calculus, so if we want the logical preorder to be complete with respect to our desiderata, we're going to have to add additional desiderata. The culprit for incompleteness, as we identified it, was the derivation rule for $A \otimes B$, which requires us to *partition* a process state into two parts and observe those parts independently. The nullary version of this is $\mathbf{1}$, the

[2] That is a convenient misnomer: while the largest behavioral preorder (which we will name the *contextual preorder*) will turn out to be a true preorder, we will not stipulate that every behavioral preorder is a proper preorder, and indeed many of them are not reflexive — the empty relation will satisfy all of our desiderata for being a behavioral "preorder."

[3] The proof that there is a barb-preserving, reduction-closed, and compositional relation $\mathcal{R}$ such that $(a \multimap \mathbf{1}, b \multimap \mathbf{1})\ \mathcal{R}\ (a \multimap b \multimap \mathbf{1})$ is complex. The simplest way we know how to establish this is to define a labeled transition system which is equivalent to the largest barb-preserving, reduction-closed, and compositional relation. We can then show that, according to this labeled transition system, $(a \multimap \mathbf{1}, b \multimap \mathbf{1})$ is related to $(a \multimap b \multimap \mathbf{1})$. This is essentially the same development we will perform in Section 4 for $\preceq_c$, the largest barb-preserving, reduction-closed, compositional, *and partition-preserving* relation that we are about to define.

unit of $\otimes$, which requires us to split a process state into zero pieces; that is only possible if the process state's linear context is empty. Motivated by this possibility, we added a fourth desiderata, that **behavioral preorders should be partition-preserving**. In the binary case, this means that, if $(\Delta_{1a}, \Delta_{1b})$ is imitated by Δ_2, then it must be the case that $\Delta_2 \rightsquigarrow^* (\Delta_{2a}, \Delta_{2b})$ where Δ_{1a} is imitated by Δ_{2a} and Δ_{1b} is imitated by Δ_{2b}. In the nullary case, this means that, if $\cdot$ is imitated by Δ, then it must be the case that $\Delta_2 \rightsquigarrow^* \cdot$.[4]

Formally, these desiderata are captured by the following definition. In that definition, we use the more traditional notation from process calculus and say that $\Delta \downarrow_a$ if $\Delta = (\Delta', a)$ for some Δ'.

Definition 3.2. *Let $\mathcal{R}$ be a binary relation over states. We say that $\mathcal{R}$ is*

- barb-preserving *if, whenever $\Delta_1 \mathcal{R} \Delta_2$ and $\Delta_1 \downarrow_a$ for any atomic proposition a, we have that $\Delta_2 \Downarrow_a$.*
- reduction-closed *if $\Delta_1 \mathcal{R} \Delta_2$ implies that whenever $\Delta_1 \rightsquigarrow \Delta_1'$, there exists Δ_2' such that $\Delta_2 \rightsquigarrow^* \Delta_2'$ and $\Delta_1' \mathcal{R} \Delta_2'$.*
- compositional *if $\Delta_1 \mathcal{R} \Delta_2$ implies $(\Delta', \Delta_1) \mathcal{R} (\Delta', \Delta_2)$ for all Δ'.*
- partition-preserving *if $\Delta_1 \mathcal{R} \Delta_2$ implies that*
  1. *if $\Delta_1 \equiv \cdot$, then $\Delta_2 \rightsquigarrow^* \cdot$,*
  2. *for all Δ_{1a} and Δ_{1b}, if $\Delta_1 \equiv (\Delta_{1a}, \Delta_{1b})$ then there exist Δ_{2a} and Δ_{2b} such that $\Delta_2 \rightsquigarrow^* (\Delta_{2a}, \Delta_{2b})$ and furthermore $\Delta_{1a} \mathcal{R} \Delta_{2a}$ and $\Delta_{1b} \mathcal{R} \Delta_{2b}$.*

These desiderata are useful in letting us conclude that one processes *does not* imitate another. Barb-preservation lets us conclude that a is not imitated by (b, b, b), and reduction-closure furthermore lets us conclude that $(b, b \multimap a)$ is not imitated by (b, b, b). Partition-preservation lets us conclude that (a, a) is not imitated by a. Compositionality lets us conclude that $(a \multimap b, b \multimap c)$ is not imitated by $(a \multimap c)$, because if we put both process states in parallel with the process state b, then we would be able to step, in the former case, to $(a \multimap b, c)$ and then observe that $(a \multimap b, c) \downarrow_c$; the process $(a \multimap c, b)$ is unable to keep up.

Concluding that $(a \multimap \mathbf{1}, b \multimap \mathbf{1})$ is not imitated by $(a \multimap b \multimap \mathbf{1})$ — as is required by our goal of completeness relative to the logical preorder — requires compositionality and partition-preservation. If $(a \multimap \mathbf{1}, b \multimap \mathbf{1})$ is imitated by $(a \multimap b \multimap \mathbf{1})$, then by partition preservation (and the fact that $(a \multimap b \multimap \mathbf{1})$ can make no transitions), we must be able to split $(a \multimap b \multimap \mathbf{1})$ into two parts, Δ and Δ' so that $a \multimap \mathbf{1}$ is imitated by Δ and $b \multimap \mathbf{1}$ is imitated by Δ'. That necessarily means that $\Delta = \cdot$ and $\Delta' = (a \multimap b \multimap \mathbf{1})$ or vice-versa. Because of compositionality, if $(a \multimap \mathbf{1})$ were imitated by $\cdot$ then $(a, a \multimap \mathbf{1})$ would be imitated by a, but $(a, a \multimap \mathbf{1}) \rightsquigarrow^* \cdot$ and $a \not\rightsquigarrow^* \cdot$. Therefore, $(a \multimap \mathbf{1})$ cannot be imitated by $\cdot$, and by a similar argument $(b \multimap \mathbf{1})$ cannot be imitated by $\cdot$. This furthermore refutes the proposition that $(a \multimap \mathbf{1}, b \multimap \mathbf{1})$ is imitated by $(a \multimap b \multimap \mathbf{1})$, as we had hoped.

[4] A concept reminiscent of this notion of partition-preservation have recently been given in probabilistic process algebras [7]; the similar concept goes by the name Markov bisimulation in that setting.

Our desiderata allow us to define the most generous behavioral preorder by coinduction; we call this relation the *contextual preorder*. The definition has an innocent-until-proven-guilty flavor: unless there is some reason, arising from the desiderata, that one process cannot imitate another, then the contextual preorder declares the two process states to be in relation. This coinductive definition is standard from process algebra (modulo our additional requirement of partition-preservation).

Definition 3.3 (Contextual preorder)**.** *The* contextual preorder*, denoted by $\preceq_c$, is the largest relation over process states which is barb-preserving, reduction-closed, compositional, and partition-preserving.*

Contextual equivalence, which is the symmetric closure of the contextual preorder, has been widely studied in concurrency theory, though its appearance in linear logic seems to be new. It is also known as *reduction barbed congruence* and used in a variety of process calculi [16, 24, 10, 7].

3.2 Properties of the contextual preorder

Having defined the contextual preorder as the largest barb-preserving, reduction-closed, compositional, and partition-preserving binary relation over process states, we will close out this section by proving a few technical lemmas.

We start with three small "multistep" lemmas: the first lets us act like reduction closure was defined exclusively in terms of $\rightsquigarrow^*$, the second lets us act like barb preservation was defined exclusively in terms of $\Downarrow_a$, and the third lets us do something similar for partition preservation. These are used later on, and also help us prove that $\preceq_c$ is actually a preorder. Theorem 3.7 establishes that $\preceq_c$ is actually a preorder. We do not use this as a technical fact anywhere, but we are calling it the contextual *preorder*, and after we've spoken inaccurately about "behavioral preorders," you should be distrustful of us. Finally, Lemma 3.9 is a technical lemma about observations that we need later on in the proof of Theorem 4.12, and the atom renaming lemma (Lemma 3.8) is needed to prove this technical lemma.

Lemma 3.4 (Multistep reduction closure)**.** *Suppose $\Delta_1 \preceq_c \Delta_2$. If $\Delta_1 \rightsquigarrow^* \Delta_1'$, then there exists a Δ_2' such that $\Delta_2 \rightsquigarrow^* \Delta_2'$ and $\Delta_1' \preceq_c \Delta_2'$.*

Proof. We proceed by induction on the number of steps in $\Delta_1 \rightsquigarrow^* \Delta_1'$.

- Suppose $\Delta_1 \rightsquigarrow^* \Delta_1'$ in zero steps (that is, $\Delta_1 \equiv \Delta_1'$). Then $\Delta_2 \rightsquigarrow^* \Delta_2$ in zero steps and $\Delta_1' \preceq_c \Delta_2$ by assumption.
- Suppose $\Delta_1 \rightsquigarrow \Delta_1'' \rightsquigarrow^* \Delta_1'$. Since $\Delta_1 \preceq_c \Delta_2$, there exists some Δ_2'' such that $\Delta_2 \rightsquigarrow^* \Delta_2''$ and $\Delta_1'' \preceq_c \Delta_2''$. The induction hypothesis then implies the existence of some Δ_2' such that $\Delta_2'' \rightsquigarrow^* \Delta_2'$ and $\Delta_1' \preceq_c \Delta_2'$. Since the relation $\rightsquigarrow^*$ is transitive, we have $\Delta_2 \rightsquigarrow^* \Delta_2'$ as required. □

Lemma 3.5 (Multistep barb preservation)**.** *Suppose $\Delta_1 \preceq_c \Delta_2$. If $\Delta_1 \Downarrow_a$ then $\Delta_2 \Downarrow_a$.*

Proof. If $\Delta_1 \Downarrow_a$, then there exists Δ_1' with $\Delta_1 \rightsquigarrow^* \Delta_1'$ and $\Delta_1' \downarrow_a$. By multistep reduction closure (Lemma 3.4), there exists Δ_2' such that $\Delta_2 \rightsquigarrow^* \Delta_2'$ and $\Delta_1' \preceq_c \Delta_2'$. The latter and $\Delta_1' \downarrow_a$ together imply $\Delta_2' \Downarrow_a$, i.e. there exists some Δ_2'' such that $\Delta_2' \rightsquigarrow^* \Delta_2''$ and $\Delta_2'' \downarrow_a$. The transitivity of $\rightsquigarrow^*$ then yields $\Delta_2 \rightsquigarrow^* \Delta_2''$. It follows that $\Delta_2 \Downarrow_a$. □

Lemma 3.6 (Multistep partition preservation)**.** *Suppose* $\Delta_1 \preceq_c \Delta_2$. *If* $\Delta_1 \rightsquigarrow^* \cdot$, *then* $\Delta_2 \rightsquigarrow^* \cdot$, *and if* $\Delta_1 \rightsquigarrow^* (\Delta_{1a}, \Delta_{1b})$, *then there exist* Δ_{2a} *and* Δ_{2b} *such that* $\Delta_2 \rightsquigarrow^* (\Delta_{2a}, \Delta_{2b})$, $\Delta_{1a} \preceq_c \Delta_{2a}$, *and* $\Delta_{1b} \preceq_c \Delta_{2b}$.

Proof. In the first case, we assume $\Delta_1 \rightsquigarrow^* \cdot$ and have by multistep reduction closure (Lemma 3.4) that $\Delta_2 \rightsquigarrow^* \Delta_2'$ such that $\cdot \preceq_c \Delta_2'$. Then, by partition preservation, we have that $\Delta_2' \rightsquigarrow^* \cdot$. The result then follows by the transitivity of $\rightsquigarrow^*$.

In the second case, we assume $\Delta_1 \rightsquigarrow^* (\Delta_{1a}, \Delta_{1b})$ and have by multistep reduction closure (Lemma 3.4) that $\Delta_2 \rightsquigarrow^* \Delta_2'$ such that $(\Delta_{1a}, \Delta_{1b}) \preceq_c \Delta_2'$. Then, by partition preservation, we have that $\Delta_2' \rightsquigarrow^* (\Delta_{2a}, \Delta_{2b})$ such that $\Delta_{1a} \preceq_c \Delta_{2a}$ and $\Delta_{1b} \preceq_c \Delta_{2b}$. The result then follows by the transitivity of $\rightsquigarrow^*$. □

The following three proofs, of Theorem 3.7, Lemma 3.8, and Lemma 3.9, proceed by coinduction. We prove a property $\mathcal{P}$ of the contextual preorder $\preceq_c$ by defining some relation $\mathcal{R}$ in such a way that $\preceq_c$ satisfies $\mathcal{P}$ if $\mathcal{R} \subseteq \preceq_c$. Then we show that $\mathcal{R}$ is barb-preserving, reduction-closed, compositional, and partition-preserving, which establishes that $\mathcal{R} \subseteq \preceq_c$.

Theorem 3.7. $\preceq_c$ *is a preorder.*

Proof. A preorder is a reflexive and transitive relation. It is straightforward to show that the identity relation, $\mathcal{R}_{id}$ such that $\Delta \; \mathcal{R}_{id} \; \Delta$ for all Δ, is barb-preserving, reduction-closed, compositional, and partition-preserving. Since $\preceq_c$ is the largest relation with these properties, we have that $\mathcal{R}_{id} \subseteq \preceq_c$ and therefore $\preceq_c$ is reflexive.

It remains to be shown that $\preceq_c$ is transitive. Consider the relation

$$\mathcal{R} \;:=\; \{(\Delta_1, \Delta_3) \mid \text{ there is some } \Delta_2 \text{ with } \Delta_1 \preceq_c \Delta_2 \text{ and } \Delta_2 \preceq_c \Delta_3\}.$$

It will suffice to show that $\mathcal{R} \subseteq \preceq_c$, because in that case, given $\Delta_1 \preceq_c \Delta_2$ and $\Delta_2 \preceq_c \Delta_3$, we have $\Delta_1 \; \mathcal{R} \; \Delta_3$ and consequently $\Delta_1 \preceq_c \Delta_3$. We can show that $\mathcal{R} \subseteq \preceq_c$ by showing that $\mathcal{R}$ is barb-preserving, reduction-closed, compositional, and partition-preserving. Suppose $\Delta_1 \; \mathcal{R} \; \Delta_3$, that is, $\Delta_1 \preceq_c \Delta_2$ and $\Delta_2 \preceq_c \Delta_3$ for some Δ_2.

Barb-preserving We assume $\Delta_1 \downarrow_a$ for some arbitrary a and must show that $\Delta_3 \Downarrow_a$.

(1)	$\Delta_1 \downarrow_a$	by assumption
(2)	$\Delta_2 \Downarrow_a$	by definition of barb-preserving on $\Delta_1 \preceq_c \Delta_2$ and (1)
(3)	$\Delta_3 \Downarrow_a$	by multistep barb preservation (Lemma 3.5) on $\Delta_2 \preceq_c \Delta_3$ and (2)

Reduction-closed We assume $\Delta_1 \rightsquigarrow \Delta_1'$ for some arbitrary Δ_1' and must show that there exists Δ_3' such that $\Delta_3 \rightsquigarrow^* \Delta_3'$ and $\Delta_1' \; \mathcal{R} \; \Delta_3'$.

(1) $\Delta_1 \rightsquigarrow \Delta_1'$ by assumption
(2) $\Delta_2 \rightsquigarrow^* \Delta_2'$ by definition of reduction-closed on $\Delta_1 \preceq_c \Delta_2$ and (1)
(3) $\Delta_1' \preceq_c \Delta_2'$ *(same)*
(4) $\Delta_3 \rightsquigarrow^* \Delta_3'$ by multistep reduction closure (Lemma 3.4) on $\Delta_2 \preceq_c \Delta_3$ and (2)
(5) $\Delta_2' \preceq_c \Delta_3'$ *(same)*
(6) $\Delta_1' \mathcal{R} \Delta_3'$ by definition of $\mathcal{R}$ on (3) and (5)

Compositional We must show $(\Delta, \Delta_1) \mathcal{R} (\Delta, \Delta_3)$ for arbitrary Δ.

(1) $(\Delta, \Delta_1) \preceq_c (\Delta, \Delta_2)$ by definition of compositional on $\Delta_1 \preceq_c \Delta_2$
(2) $(\Delta, \Delta_2) \preceq_c (\Delta, \Delta_3)$ by definition of compositional on $\Delta_2 \preceq_c \Delta_3$
(3) $(\Delta, \Delta_1) \mathcal{R} (\Delta, \Delta_3)$ by definition of $\mathcal{R}$ on (1) and (2)

Partition-preserving We first assume $\Delta_1 \equiv \cdot$ and must show that $\Delta_3 \rightsquigarrow^* \cdot$.

(1) $\Delta_1 \equiv \cdot$ by assumption
(2) $\Delta_2 \rightsquigarrow^* \cdot$ by definition of partition-preserving on $\Delta_1 \preceq_c \Delta_2$
(3) $\Delta_3 \rightsquigarrow^* \cdot$ by multistep partition preservation (Lemma 3.6) on (2)

We next assume $\Delta_1 \equiv (\Delta_{1a}, \Delta_{1b})$ and must show that $\Delta_3 \rightsquigarrow^* (\Delta_{3a}, \Delta_{3b})$ such that $\Delta_{1a} \mathcal{R} \Delta_{3a}$ and $\Delta_{1b} \mathcal{R} \Delta_{3b}$.

(1) $\Delta_1 \equiv (\Delta_{1a}, \Delta_{1b})$ by assumption
(2) $\Delta_2 \rightsquigarrow^* (\Delta_{2a}, \Delta_{2b})$ by definition of partition-preserving on $\Delta_1 \preceq_c \Delta_2$ and (1)
(3) $\Delta_{1a} \preceq_c \Delta_{2a}$ *(same)*
(4) $\Delta_{1b} \preceq_c \Delta_{2b}$ *(same)*
(5) $\Delta_3 \rightsquigarrow^* (\Delta_{3a}, \Delta_{3b})$ by multistep partition preservation (Lemma 3.6) on (2)
(6) $\Delta_{2a} \preceq_c \Delta_{3a}$ *(same)*
(7) $\Delta_{2b} \preceq_c \Delta_{3b}$ *(same)*
(8) $\Delta_{1a} \mathcal{R} \Delta_{3a}$ by definition of $\mathcal{R}$ on (3) and (6)
(9) $\Delta_{1b} \mathcal{R} \Delta_{3b}$ by definition of $\mathcal{R}$ on (4) and (7)

This suffices to show that $\mathcal{R} \subseteq \preceq_c$, which concludes the proof. □

Lemma 3.8 (Atom renaming)**.** *If ρ is a bijective function substituting atomic propositions for atomic propositions, then $\rho\Delta_1 \preceq_c \rho\Delta_2$ implies $\Delta_1 \preceq_c \Delta_2$*

Proof. We will use throughout this proof two extra lemmas. The first, the *transition renaming property*, is that $\Delta \rightsquigarrow \Delta'$ iff $\rho\Delta \rightsquigarrow \rho\Delta'$. This is shown by case analysis on the given reduction. The second, the *multistep transition renaming property*, is that $\Delta \rightsquigarrow^* \Delta'$ iff $\rho\Delta \rightsquigarrow^* \rho\Delta'$. This is shown by induction on the structure of the given reduction and the transition renaming property.

Consider the relation

$$\mathcal{R} \;:=\; \{(\Delta_1, \Delta_2) \mid \rho\Delta_1 \preceq_c \rho\Delta_2\}.$$

It will suffice to show that $\mathcal{R} \subseteq \preceq_c$, because in that case, given $\rho\Delta_1 \preceq_c \rho\Delta_2$, we have $\Delta_1 \mathrel{\mathcal{R}} \Delta_2$, which will in turn imply $\Delta_1 \preceq_c \Delta_2$. We can show $\mathcal{R} \subseteq \preceq_c$ by showing that $\mathcal{R}$ is barb-preserving, reduction-closed, compositional, and partition-preserving. Suppose $\Delta_1 \mathrel{\mathcal{R}} \Delta_2$, that is, $\rho\Delta_1 \preceq_c \rho\Delta_2$.

Barb-preserving We assume $\Delta_1 \downarrow_a$ for some arbitrary a and must show that $\Delta_2 \Downarrow_a$.

(1)	$\Delta_1 \downarrow_a$	by assumption
(2)	$\rho\Delta_1 \downarrow_{\rho a}$	by definition of $\downarrow_a$ and (1)
(3)	$\rho\Delta_2 \Downarrow_{\rho a}$	by definition of barb-preserving on $\rho\Delta_1 \preceq_c \rho\Delta_2$ and (2)
(4)	$\Delta_2 \Downarrow_a$	by definition of $\Downarrow_a$ and (3)

Reduction-closed We assume $\Delta_1 \rightsquigarrow \Delta_1'$ for some arbitrary Δ_1' and must show that there exists Δ_2' such that $\Delta_2 \rightsquigarrow^* \Delta_2'$ and $\Delta_1' \mathrel{\mathcal{R}} \Delta_2'$. We let Δ_2' be $\rho^{-1}\Delta^*$; the renaming ρ is invertible by virtue of being total and bijective.

(1)	$\Delta_1 \rightsquigarrow \Delta_1'$	by assumption
(2)	$\rho\Delta_1 \rightsquigarrow \rho\Delta_1'$	by transition renaming property on (1)
(3)	$\rho\Delta_2 \rightsquigarrow^* \Delta^*$	by definition of reduction-closure on $\rho\Delta_1 \preceq_c \rho\Delta_2$ and (2)
(4)	$\rho\Delta_1' \preceq_c \Delta^*$	*(same)*
(5)	$\Delta_1' \mathrel{\mathcal{R}} \Delta_2'$	by definition of $\mathcal{R}$ and Δ_2' on (4)

Compositional We must show $(\Delta, \Delta_1) \mathrel{\mathcal{R}} (\Delta, \Delta_3)$ for arbitrary Δ.

(1)	$(\rho\Delta, \rho\Delta_1) \preceq_c (\rho\Delta, \rho\Delta_2)$	by definition of compositionality on $\Delta_1 \preceq_c \Delta_2$
(2)	$(\Delta, \Delta_1) \mathrel{\mathcal{R}} (\Delta, \Delta_2)$	by definition of $\mathcal{R}$ on (1)

Partition-preserving We first assume $\Delta_1 \equiv \cdot$ and must show that $\Delta_2 \rightsquigarrow^* \cdot$.

(1)	$\Delta_1 \equiv \rho\Delta_1 \equiv \cdot$	by assumption
(2)	$\rho\Delta_2 \rightsquigarrow^* \cdot$	by definition of partition-preservation on $\Delta_1 \preceq_c \Delta_2$
(3)	$\Delta_2 \rightsquigarrow^* \cdot$	by multistep transition renaming property on (2)

We next assume $\Delta_1 \equiv (\Delta_{1a}, \Delta_{1b})$ and must show that $\Delta_2 \rightsquigarrow^* (\Delta_{2a}, \Delta_{2b})$ such that $\Delta_{1a} \mathrel{\mathcal{R}} \Delta_{2a}$ and $\Delta_{1b} \mathrel{\mathcal{R}} \Delta_{2b}$. We let Δ_{2a} be $\rho^{-1}\Delta_a^*$ and let Δ_{2b} be $\rho^{-1}\Delta_b^*$.

(1)	$\Delta_1 \equiv (\Delta_{1a}, \Delta_{1b})$	by assumption
(2)	$\rho\Delta_1 \equiv (\rho\Delta_{1a}, \rho\Delta_{1b})$	by (1)
(3)	$\rho\Delta_2 \rightsquigarrow^* (\Delta_a^*, \Delta_b^*)$	by definition of partition-preservation on $\rho\Delta_1 \preceq_c \rho\Delta_2$ and (1)
(4)	$\rho\Delta_{1a} \preceq_c \Delta_a^*$	*(same)*
(5)	$\rho\Delta_{1b} \preceq_c \Delta_b^*$	*(same)*
(6)	$\Delta_2 \rightsquigarrow^* (\Delta_{2a}, \Delta_{2b})$	by multistep transition renaming property on (3)
(7)	$\Delta_{1a} \mathrel{\mathcal{R}} \Delta_{2a}$	by definition of $\mathcal{R}$ on (4)
(8)	$\Delta_{1b} \mathrel{\mathcal{R}} \Delta_{2b}$	by definition of $\mathcal{R}$ on (5)

This suffices to show that $\mathcal{R} \subseteq \preceq_c$, which concludes the proof. $\square$

Lemma 3.9 (Fresh atom removal)**.** *If* $(\Delta_1, a) \preceq_c (\Delta_2, a)$*, where* a *occurs neither in* Δ_1 *nor in* Δ_2*, then* $\Delta_1 \preceq_c \Delta_2$*.*

Proof. Consider the relation

$$\mathcal{R} \;:=\; \{(\Delta_1, \Delta_2) \mid \text{there exists } a \notin (\Delta_1 \cup \Delta_2) \text{ with } (\Delta_1, a) \preceq_c (\Delta_2, a)\}.$$

It will suffice to show that $\mathcal{R} \subseteq \preceq_c$, because in that case, given $(\Delta_1, a) \preceq_c (\Delta_2, a)$ for some a that occurs neither in Δ_1 or Δ_2, we will know that $\Delta_1 \;\mathcal{R}\; \Delta_2$, which will in turn imply $\Delta_1 \preceq_c \Delta_2$. We can show $\mathcal{R}$ is barb-preserving, reduction-closed, compositional, and partition-preserving. Suppose $\Delta_1 \;\mathcal{R}\; \Delta_2$, that is that $(\Delta_1, a) \preceq_c (\Delta_2, a)$ for some arbitrary a such that $a \notin \Delta_1$ and $a \notin \Delta_2$.

Barb-preserving We assume $\Delta_1 \downarrow_b$ for some arbitrary b and must show that $\Delta_2 \Downarrow_b$. It cannot be the case that $\Delta_1 \downarrow_a$, so we have $a \neq b$. Pick another fresh atomic proposition $c \notin (\Delta_1 \cup \Delta_2)$.

(1) $\Delta_1 \downarrow_b$ — by assumption
(2) $\Delta_1 \equiv \Delta_1', b$ — by definition of $\downarrow_b$ and (1)
(3) $(b \multimap a \multimap c, \Delta_1, a) \preceq_c (b \multimap a \multimap c, \Delta_2, a)$ — by definition of compositionality on $(\Delta_1, a) \preceq_c (\Delta_2, a)$
(4) $(b \multimap a \multimap c, \Delta_1, a) \rightsquigarrow (a \multimap c, \Delta_1', a)$ — by rule $\rightsquigarrow\!\multimap$
(5) $(a \multimap c, \Delta_1', a) \rightsquigarrow (\Delta_1', c)$ — by rule $\rightsquigarrow\!\multimap$
(6) $(b \multimap a \multimap c, \Delta_1, a) \Downarrow_c$ — by definition of $\Downarrow_c$, (4), and (5)
(7) $(b \multimap a \multimap c, \Delta_2, a) \Downarrow_c$ — by multistep barb preservation (Lemma 3.5) on (3) and (6)
(8) $(b \multimap a \multimap c, \Delta_2, a) \rightsquigarrow^* (\Delta', c)$ — by definition of $\Downarrow_c$ on (7)

By induction on the structure of (8), $b \multimap a \multimap c$ must be consumed to produce c, meaning that $\Delta_2 \rightsquigarrow^* (\Delta_2', b)$ and, consequently, $\Delta_2 \Downarrow_b$.

Reduction-closed We assume $\Delta_1 \rightsquigarrow \Delta_1'$ for some arbitrary Δ_1' and must show that there exists Δ_3' such that $\Delta_2 \rightsquigarrow^* \Delta_2'$ and $\Delta_1' \;\mathcal{R}\; \Delta_2'$.

(1) $\Delta_1 \rightsquigarrow \Delta_1'$ — by assumption
(2) $(\Delta_1, a) \rightsquigarrow (\Delta_1', a)$ — consequence of (1)
(3) $a \notin \Delta_1'$ — *(same)*
(4) $(\Delta_2, a) \rightsquigarrow^* \Delta^*$ — by definition of reduction-closure on $(\Delta_1, a) \preceq_c (\Delta_2, a)$ and (2)
(5) $(\Delta_1', a) \preceq_c \Delta^*$ — *(same)*
(6) $\Delta^* \equiv (\Delta_2', a)$ — by induction on the structure of (4)
(7) $a \notin \Delta_2'$ — *(same)*
(8) $(\Delta_1', a) \preceq_c (\Delta_2', a)$ — by rewriting (3) using (5)
(9) $\Delta_1' \;\mathcal{R}\; \Delta_2'$ — by definition of $\mathcal{R}$ on (8), (3), and (7)

Compositional We must show $(\Delta, \Delta_1)\ \mathcal{R}\ (\Delta, \Delta_2)$ for arbitrary Δ. Pick another fresh atomic proposition $b \notin (\Delta_1 \cup \Delta_2 \cup \Delta)$.

(1)	$(\Delta_1, b) \preceq_c (\Delta_2, b)$	by atom renaming (Lemma 3.8) on $(\Delta_1, a) \preceq_c (\Delta_2, a)$
(2)	$(\Delta, \Delta_1, b) \preceq_c (\Delta, \Delta_2, b)$	by definition of compositional on (1)
(3)	$(\Delta, \Delta_1)\ \mathcal{R}\ (\Delta, \Delta_2)$	by definition of $\mathcal{R}$ on (2)

Partition-preserving We first assume $\Delta_1 \equiv \cdot$ and must show that $\Delta_2 \rightsquigarrow^* \cdot$.

(1)	$a \preceq_c (\Delta_2, a)$	by $(\Delta_1, a) \preceq_c (\Delta_2, a)$, $\Delta_1 \equiv \cdot$
(2)	$(a, a \multimap \mathbf{1}) \preceq_c (\Delta_2, a, a \multimap 1)$	by definition of compositionality on (1)
(3)	$(a, a \multimap \mathbf{1}) \rightsquigarrow^* \cdot$	by rule $\rightsquigarrow\!\multimap$, rule $\rightsquigarrow\mathbf{1}$, and transitivity of $\rightsquigarrow^*$
(4)	$(\Delta_2, a, a \multimap \mathbf{1}) \rightsquigarrow^* \cdot$	by multistep partition preservation on (2) and (3)
(5)	$\Delta_2 \rightsquigarrow^* \cdot$	by induction on the structure of (4)

We next assume $\Delta_1 \equiv (\Delta_{1a}, \Delta_{1b})$ and must show that $\Delta_2 \rightsquigarrow^* (\Delta_{2a}, \Delta_{2b})$ such that $\Delta_{1a}\ \mathcal{R}\ \Delta_{2a}$ and $\Delta_{1b}\ \mathcal{R}\ \Delta_{2b}$. Pick another fresh atomic proposition $b \notin (\Delta_1 \cup \Delta_2)$.

(1)	$(\Delta_{1a}, \Delta_{1b}, a, b) \preceq_c (\Delta_2, a, b)$	by definition of compositional on $(\Delta_1, a) \preceq_c (\Delta_2, a)$
(2)	$(\Delta_2, a, b) \rightsquigarrow^* (\Delta'_{2a}, \Delta'_{2b})$	by definition of partition-preserving on (1)
(3)	$(\Delta_{1a}, a) \preceq_c \Delta'_{2a}$	*(same)*
(4)	$(\Delta_{1b}, b) \preceq_c \Delta'_{2b}$	*(same)*
(5)	$\Delta'_{2a} \Downarrow_a$	by definition of barb-preserving on (3)
(6)	$\Delta'_{2a} \equiv (\Delta_{2a}, a)$	by induction on the reduction steps in (5)
(7)	$a \notin \Delta_{2a}$	by induction on the structure of (2)
(8)	$\Delta_{1a}\ \mathcal{R}\ \Delta_{2a}$	by definition of $\mathcal{R}$ on (3), (6), (7)
(9)	$\Delta'_{2b} \Downarrow_b$	by definition of barb-preserving on (4)
(10)	$\Delta'_{2b} \equiv (\Delta_{2b}, b)$	by induction on the reduction steps in (9)
(11)	$b \notin \Delta_{2b}$	by induction on the structure of (2)
(12)	$\Delta_{1b}\ \mathcal{R}\ \Delta_{2b}$	by definition of $\mathcal{R}$ on (4), (10), (11)

This suffices to show that $\mathcal{R} \subseteq \preceq_c$, which concludes the proof. □

Note that Lemma 3.9 is invalid if a is not fresh with respect to Δ_1 and Δ_2. For example, we have

$$a \preceq_c (a \multimap a, a) \quad \text{but} \quad (\cdot) \not\preceq_c (a \multimap a)$$

This will be easy to check when using the results in Section 4.5.

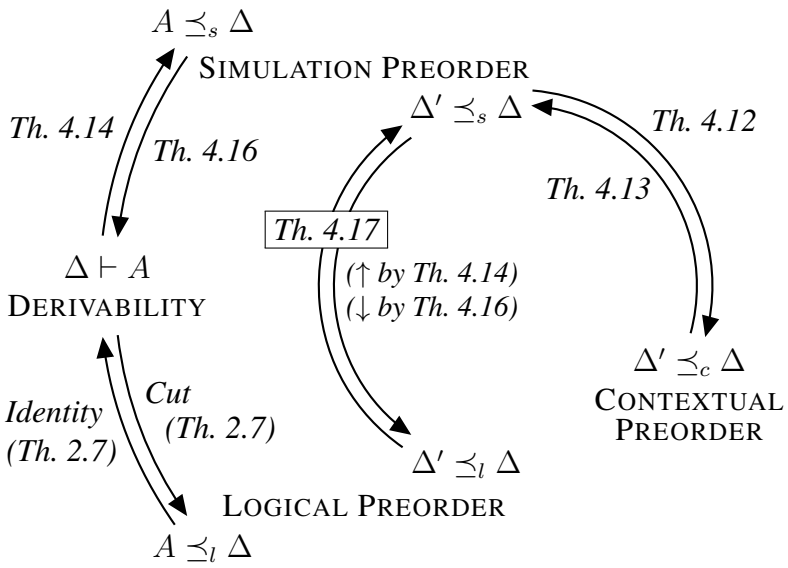

Figure 3: Visual Summary of Equivalence Proofs in Sections 3 and 4

4 Labeled Transitions and the Simulation Preorder

We will now show that, for the restricted fragment of linear logic given in Section 3, the logical preorder and the contextual preorder coincide. The key idea is to coinductively define a third preorder, the *simulation preorder*, that acts as a stepping stone between the two contexts. It is hard to use the contextual preorder to show that one process actually imitates another; the simulation preorder will be more useful for this purpose.

The overall structure of the relevant proofs up to and including this section is shown in Figure 4. In Sections 4.1–4.3, we will present our stepping stones: a labeled transition system and the simulation preorder, and in Section 4.4 we will prove some properties of this preorder. Then, in Section 4.5 we show that the simulation and contextual preorders coincide, and in Section 4.6 we show that the simulation and logical preorders coincide; it is an obvious corollary that the logical and contextual preorders coincide.

4.1 Labeled transitions

We will characterize the contextual preorder as a coinductively defined relation. For that purpose, we give a labeled transition semantics for states. Labels, or actions, are defined by the following grammar:

$$\begin{array}{lrcl} \text{Non-receive actions} & \alpha & ::= & \tau \mid !a \\ \text{Generic actions} & \beta & ::= & \alpha \mid ?a \end{array}$$

We distinguish "non-receive" labels, denoted α, as either the silent action τ or a label $!a$ representing a send action. Generic labels β extend them with receive actions $?a$.

$$\frac{}{(\Delta, a) \xrightarrow{!a} \Delta}\ lts! \qquad \frac{}{(\Delta, a \multimap B) \xrightarrow{?a} (\Delta, B)}\ lts?$$

$$\frac{}{(\Delta, \mathbf{1}) \xrightarrow{\tau} \Delta}\ lts\mathbf{1} \qquad \frac{}{(\Delta, A \otimes B) \xrightarrow{\tau} (\Delta, A, B)}\ lts\otimes$$

$$\frac{}{(\Delta, A \,\&\, B) \xrightarrow{\tau} (\Delta, A)}\ lts\,\&_1 \qquad \frac{}{(\Delta, A \,\&\, B) \xrightarrow{\tau} (\Delta, B)}\ lts\,\&_2$$

(No rule for $\top$) $\qquad \dfrac{\Delta_1 \xrightarrow{!a} \Delta_1' \quad \Delta_2 \xrightarrow{?a} \Delta_2'}{(\Delta_1, \Delta_2) \xrightarrow{\tau} (\Delta_1', \Delta_2')}\ lts!?$

Figure 4: Labeled Transition System

The labeled transition semantics for our states, written using the judgment $\Delta \xrightarrow{\beta} \Delta'$, is defined by the rules in Figure 4. Since $\top$ is a process that is stuck, it has no action to perform. We write $\stackrel{\tau}{\Longrightarrow}$ for the reflexive transitive closure of $\xrightarrow{\tau}$, and $\Delta \stackrel{\beta}{\Longrightarrow} \Delta'$ for $\Delta \stackrel{\tau}{\Longrightarrow}\xrightarrow{\beta}\stackrel{\tau}{\Longrightarrow} \Delta'$, if $\beta \neq \tau$. Note that τ transitions correspond to reductions, as expressed by Lemma 4.1:

Lemma 4.1. *$\Delta_1 \xrightarrow{\tau} \Delta_2$ if and only if $\Delta_1 \rightsquigarrow \Delta_2$, and $\Delta_1 \stackrel{\tau}{\Longrightarrow} \Delta_2$ if and only if $\Delta_1 \rightsquigarrow^* \Delta_2$*

Proof. The proof of the first statement is by case analysis on the τ transition rules in Figure 4 in one direction and by case analysis on the reduction rules in Figure 2 in the other.

The second statement follows from the first by induction on the number of steps taken. □

Now we can use the labeled transition system to define a simulation relation; simulation is defined by coinduction and echoes the definition of the contextual preorder in many ways. However, it critically lacks the compositionality requirement that appears in the definition of the contextual preorder.

Definition 4.2 (Simulation)**.** *A relation $\mathcal{R}$ between two processes represented as Δ_1 and Δ_2 is a* simulation *if $\Delta_1\ \mathcal{R}\ \Delta_2$ implies*

1. *if $\Delta_1 \equiv \cdot$, then $\Delta_2 \stackrel{\tau}{\Longrightarrow} \cdot$.*
2. *if $\Delta_1 \equiv (\Delta_1', \Delta_1'')$, then $\Delta_2 \stackrel{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$ for some Δ_2', Δ_2'' such that $\Delta_1'\ \mathcal{R}\ \Delta_2'$ and $\Delta_1''\ \mathcal{R}\ \Delta_2''$.*
3. *whenever $\Delta_1 \xrightarrow{\alpha} \Delta_1'$, there exists Δ_2' such that $\Delta_2 \stackrel{\alpha}{\Longrightarrow} \Delta_2'$ and $\Delta_1'\ \mathcal{R}\ \Delta_2'$.*
4. *whenever $\Delta_1 \xrightarrow{?a} \Delta_1'$, there exists Δ_2' such that $(\Delta_2, a) \stackrel{\tau}{\Longrightarrow} \Delta_2'$ and $\Delta_1'\ \mathcal{R}\ \Delta_2'$.*

We write $\Delta_1 \preceq_s \Delta_2$ if there is some simulation $\mathcal{R}$ such that $\Delta_1\ \mathcal{R}\ \Delta_2$.

4.2 Examples

The contextual preorder was good at showing that one process was *not* imitated by another, but the simulation preorder is useful for showing that a process state *is* simulated by another. We will give a few examples, most of which implicitly utilize the following property:

Remark 4.3. *Given a formula A and a context Δ, to check if $A \preceq_s \Delta$ holds, there is no need to consider clause (2) in Definition 4.2 because it holds vacuously.*

Proof. We are given that $A \equiv (\Delta_1', \Delta_1'')$ and we have to pick a Δ_2' and a Δ_2'' such that $\Delta_2 \stackrel{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$, $\Delta_1' \preceq_s \Delta_2'$, and $\Delta_1'' \preceq_s \Delta_2''$. Without loss of generality we can say that $\Delta_1' \equiv A$ and $\Delta_1'' \equiv \cdot$ — the other case, where $\Delta_1' \equiv \cdot$ and $\Delta_1'' \equiv A$, is symmetric. We pick Δ_2' to be Δ_2, pick Δ_2'' to be $\cdot$, and we must show

- $\Delta_2 \stackrel{\tau}{\Longrightarrow} (\Delta_2, \cdot)$ — this is immediate from the reflexivity of $\stackrel{\tau}{\Longrightarrow}$ and the definition of contextual equivalence.
- $A \preceq_s \Delta_1$ — this is what we initially set out to prove, so it follows immediately from the coinduction hypothesis.
- $\cdot \preceq_s \cdot$ — this follows from condition 1, none of the other conditions are applicable. □

Given the remark above, we can show that $\top \preceq_s \Delta$ for any Δ, because $\top$ is neither empty nor able to perform any actions. Therefore, conditions 1, 3, and 4 are met vacuously. However, $\mathbf{1} \not\preceq_s \top$, because $\top$ cannot be reduced to $\cdot$ and $\mathbf{1}$ can (condition 1).

As another example, we have that $(a \multimap a) \preceq_s (\cdot)$. Ignoring condition 2 as before, the only possible transition for $(a \multimap a)$ is $(a \multimap a) \stackrel{?a}{\longrightarrow} a$. We match this action, according to condition 4, by letting $a \stackrel{\tau}{\Longrightarrow} a$; we then must show $a \preceq_s a$, which again follows by reflexivity (Lemma 4.8, which we will be proving momentarily).

Finally, let us show that $(a \multimap b \multimap A) \preceq_s (b \multimap a \multimap A)$. The only transition possible for the (purportedly) simulated process state is $(a \multimap b \multimap A) \stackrel{?a}{\longrightarrow} (b \multimap A)$, which means that we can proceed by showing that $(b \multimap a \multimap A, a) \stackrel{\tau}{\Longrightarrow} (b \multimap a \multimap A, a)$ (immediate from the reflexivity of $\stackrel{\tau}{\Longrightarrow}$) and that $(b \multimap A) \preceq_s (b \multimap a \multimap A, a)$. To prove this, we observe that the only transition possible for the (purportedly) simulated process is $(b \multimap A) \stackrel{?b}{\longrightarrow} A$, which means that we can proceed by showing that $(b \multimap a \multimap A, a, b) \stackrel{\tau}{\Longrightarrow} A$ (which can be done in two steps) and that $A \preceq_s A$, which again follows from reflexivity (Lemma 4.8 again).

Another way of looking at this last example is that, in the case where A is $\mathbf{1}$, we have proved that the binary relation

$$\{(a \multimap b \multimap \mathbf{1}, b \multimap a \multimap \mathbf{1}), \quad (b \multimap \mathbf{1}, (b \multimap a \multimap \mathbf{1}, a)), \quad (\mathbf{1}, \mathbf{1}), \quad (\cdot, \cdot)\}$$

is a simulation. The simulation also works in the other direction — $(b \multimap a \multimap A) \preceq_s (a \multimap b \multimap A)$. Again in the case where A is $\mathbf{1}$, this is the same as proving that the binary relation

$$\{(b \multimap a \multimap \mathbf{1}, a \multimap b \multimap \mathbf{1}), \quad (a \multimap \mathbf{1}, (a \multimap b \multimap \mathbf{1}, b)), \quad (\mathbf{1}, \mathbf{1}), \quad (\cdot, \cdot)\}$$

is a simulation.

However, the two process states are *not* bisimilar according to the usual definition of bisimilarity: there is no *single* relation that simultaneously establishes that $b \multimap a \multimap \mathbf{1}$ is simulated by $a \multimap b \multimap \mathbf{1}$ and, if we flip the relation around, establishes that $a \multimap b \multimap \mathbf{1}$ is simulated by $b \multimap a \multimap \mathbf{1}$. To see why, recall the argument that it had to be the case that $(b \multimap \mathbf{1}, (b \multimap a \multimap \mathbf{1}, a))$ the first of the two simulation relations above. But these two process states are *not* similar in both directions: $b \multimap \mathbf{1} \preceq_s (b \multimap a \multimap \mathbf{1}, a)$ and $(b \multimap a \multimap \mathbf{1}, a) \not\preceq_s b \multimap \mathbf{1}$, because $(b \multimap a \multimap \mathbf{1}, a) \xrightarrow{!a} b \multimap a \multimap \mathbf{1}$ while $b \multimap \mathbf{1} \stackrel{!a}{\not\Longrightarrow}$ (condition 3). Thus there is no way to construct a bisimulation.

4.3 Properties of labeled transitions

Towards the ultimate end of proving that the largest simulation, $\preceq_s$, is actually a preorder, we will need a few facts about labeled transitions.

Lemma 4.4 (Compositionality of labeled transitions)**.**

1. *If* $\Delta_1 \xrightarrow{\beta} \Delta_2$*, then* $(\Delta, \Delta_1) \xrightarrow{\beta} (\Delta, \Delta_2)$.
2. *If* $\Delta_1 \stackrel{\beta}{\Longrightarrow} \Delta_2$*, then* $(\Delta, \Delta_1) \stackrel{\beta}{\Longrightarrow} (\Delta, \Delta_2)$.

Proof. To prove the first statement, we proceed by induction on the derivation of $\Delta_1 \xrightarrow{\beta} \Delta_2$. There are 7 cases according to Figure 4. All of them are immediate, except the case of $lts!?$, which we expand. Suppose $\Delta_1 \equiv (\Delta_1', \Delta_1'')$, $\Delta_2 \equiv (\Delta_2', \Delta_2'')$, $\Delta_1' \xrightarrow{!a} \Delta_2'$ and $\Delta_1'' \xrightarrow{?a} \Delta_2''$. By induction, we have $(\Delta, \Delta_1') \xrightarrow{!a} (\Delta, \Delta_2')$ for any Δ. Using rule $lts!?$ on this derivation and $\Delta_1'' \xrightarrow{?a} \Delta_2''$, we obtain $(\Delta, \Delta_1', \Delta_1'') \xrightarrow{\tau} (\Delta, \Delta_2', \Delta_2'')$, i.e. $(\Delta, \Delta_1) \xrightarrow{\tau} (\Delta, \Delta_2)$.

The second statement follows from the first by induction on the number of steps taken. □

Lemma 4.5 (Partitioning)**.** *If* $(\Delta_1, \Delta_2) \xrightarrow{\beta} \Delta^*$ *then we must be in one of the following four cases:*

1. $\Delta_1 \xrightarrow{\beta} \Delta_1'$ *and* $\Delta^* \equiv (\Delta_1', \Delta_2)$*;*
2. $\Delta_2 \xrightarrow{\beta} \Delta_2'$ *and* $\Delta^* \equiv (\Delta_1, \Delta_2')$*;*
3. $\Delta_1 \xrightarrow{?a} \Delta_1'$ *and* $\Delta_2 \xrightarrow{!a} \Delta_2'$ *for some* a*, such that* β *is* τ *and* $\Delta^* \equiv (\Delta_1', \Delta_2')$*;*
4. $\Delta_1 \xrightarrow{!a} \Delta_1'$ *and* $\Delta_2 \xrightarrow{?a} \Delta_2'$ *for some* a*, such that* β *is* τ *and* $\Delta^* \equiv (\Delta_1', \Delta_2')$.

Proof. There are three possibilities, depending on the forms of β.

- $\beta = !a$ for some a. Then the last rule used to derive the transition $(\Delta_1, \Delta_2) \xrightarrow{\beta} \Delta^*$ must be $lts!$ in Figure 4. So a is a member either in Δ_1 or in Δ_2. Correspondingly, we are in case 1 or 2.
- $\beta = ?a$ for some a. Then the last rule used to derive the transition $(\Delta_1, \Delta_2) \xrightarrow{\beta} \Delta^*$ must be $lts?$. So $a \multimap B$ is a member either in Δ_1 or in Δ_2. Correspondingly, we are in case 1 or 2.

- $\beta = \tau$. If the last rule used to derive the transition $(\Delta_1, \Delta_2) \xrightarrow{\beta} \Delta^*$ is not $lts!?$, then the transition is given rise by a particular formula, and we are in case 1 or 2. If $lts!?$ is the last rule used, there is an input action and an output action happening at the same time. Either both of them come from Δ_1, or both of them come from Δ_2, or one action from Δ_1 and the other from Δ_2. Consequently we are in one of the four cases above. □

In the next lemma, we write $\overset{\tau}{\Longleftarrow}$ for the reciprocal of $\overset{\tau}{\Longrightarrow}$.

Lemma 4.6. *$\overset{\tau}{\Longleftarrow}$ is a simulation (and, consequently, $\Delta_1 \overset{\tau}{\Longrightarrow} \Delta_2$ implies $\Delta_2 \preceq_s \Delta_1$).*

Proof. We will show that the four conditions in Definition 4.2 are satisfied by $\overset{\tau}{\Longleftarrow}$, which proves that the relation is a simulation; the corollary follows immediately by virtue of $\preceq_s$ being the largest simulation. Suppose $\Delta_2 \overset{\tau}{\Longrightarrow} \Delta_1$.

1. If $\Delta_2 \equiv \cdot$, then obviously $\Delta_1 \overset{\tau}{\Longrightarrow} \Delta_2 \equiv \cdot$.

2. If $\Delta_2 \equiv \Delta_2', \Delta_2''$, then $\Delta_1 \overset{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$. Taking zero steps we have $\Delta_2' \overset{\tau}{\Longleftarrow} \Delta_2'$ and $\Delta_2'' \overset{\tau}{\Longleftarrow} \Delta_2''$ as required.

3. If $\Delta_2 \xrightarrow{\alpha} \Delta_2'$, then $\Delta_1 \overset{\tau}{\Longrightarrow} \Delta_2 \xrightarrow{\alpha} \Delta_2'$ and therefore $\Delta_2 \overset{\alpha}{\Longrightarrow} \Delta_2'$. Taking zero steps we have $\Delta_2' \overset{\tau}{\Longleftarrow} \Delta_2'$ as required.

4. Now suppose $\Delta_2 \xrightarrow{?a} \Delta_2'$. Then there are Δ_2'' and A such that $\Delta_2 \equiv (\Delta_2'', a \multimap A)$ and $\Delta_2' \equiv (\Delta_2'', A)$. From (Δ_1, a) we have the following matching transition:

$$\begin{array}{rcll} (\Delta_1, a) & \overset{\tau}{\Longrightarrow} & (\Delta_2, a) & \text{by compositionality (Lemma 4.4)} \\ & \equiv & (\Delta_2'', a \multimap A, a) & \\ & \xrightarrow{\tau} & (\Delta_2'', A) & \text{by rule } lts!? \text{ in Figure 4} \\ & \equiv & \Delta_2' & \end{array}$$

Taking zero steps, we have $\Delta_2' \overset{\tau}{\Longleftarrow} \Delta_2'$ as required. □

4.4 Properties of the simulation preorder

It was relatively simple to prove that the contextual preorder was, in fact, a preorder. It is a bit more difficult to show that the simulation preorder is, in fact, a preorder; our goal in this section is to prove Theorem 4.11, that the simulation preorder $\preceq_s$ is a proper preorder.

The structure of this section mirrors the structure of Section 3.2 (Properties of the contextual preorder). First we will prove a technical lemma that lets us act as if simulation was defined exclusively in terms of $\overset{\beta}{\Longrightarrow}$ rather than $\xrightarrow{\beta}$, and then we prove that simulation is reflexive (Lemma 4.8), compositional (Lemma 4.9), and transitive (Lemma 4.10), from which Theorem 4.11 is an immediate result.

Lemma 4.7. *If $\Delta_1 \preceq_s \Delta_2$ then*

1. *whenever* $\Delta_1 \stackrel{\tau}{\Longrightarrow} \cdot$, *then* $\Delta_2 \stackrel{\tau}{\Longrightarrow} \cdot$;

2. *whenever* $\Delta_1 \stackrel{\tau}{\Longrightarrow} (\Delta_1', \Delta_1'')$, *there exist* Δ_2' *and* Δ_2'' *such that* $\Delta_2 \stackrel{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$ *and* $\Delta_1' \preceq_s \Delta_2'$ *and furthermore* $\Delta_1'' \preceq_s \Delta_2''$;

3. *whenever* $\Delta_1 \stackrel{\alpha}{\Longrightarrow} \Delta_1'$, *there exists* Δ_2' *such that* $\Delta_2 \stackrel{\alpha}{\Longrightarrow} \Delta_2'$ *and* $\Delta_1' \preceq_s \Delta_2'$.

Proof. We first prove a particular case of the third statement:

(1) If $\Delta_1 \stackrel{\tau}{\Longrightarrow} \Delta_1'$ then there exists some Δ_2' such that $\Delta_2 \stackrel{\tau}{\Longrightarrow} \Delta_2'$ and $\Delta_1' \preceq_s \Delta_2'$.

We proceed by induction on the length of the transition $\Delta_1 \stackrel{\tau}{\Longrightarrow} \Delta_1'$.

- If $\Delta_1 \equiv \Delta_1'$, then $\Delta_2 \stackrel{\tau}{\Longrightarrow} \Delta_2$ by the reflexivity of $\stackrel{\tau}{\Longrightarrow}$, and $\Delta_1 \preceq_s \Delta_2$ by assumption.
- Suppose $\Delta_1 \stackrel{\tau}{\Longrightarrow} \Delta_1'' \stackrel{\tau}{\longrightarrow} \Delta_1'$ for some Δ_1''. Since $\Delta_1 \preceq_s \Delta_2$, by the induction hypothesis there exists Δ_2'' such that $\Delta_2 \stackrel{\tau}{\Longrightarrow} \Delta_2''$ and $\Delta_1'' \preceq_s \Delta_2''$. The latter implies the existence of some Δ_2' such that $\Delta_2'' \stackrel{\tau}{\Longrightarrow} \Delta_2'$ and $\Delta_1' \preceq_s \Delta_2'$. The transitivity of $\stackrel{\tau}{\Longrightarrow}$ entails that $\Delta_2 \stackrel{\tau}{\Longrightarrow} \Delta_2'$.

We are now ready to prove the lemma.

1. Suppose $\Delta_1 \preceq_s \Delta_2$ and $\Delta_1 \stackrel{\tau}{\Longrightarrow} \cdot$. By (1), there exists Δ_2' such that $\Delta_2 \stackrel{\tau}{\Longrightarrow} \Delta_2'$ and $\cdot \preceq_s \Delta_2'$. By Definition 4.2, we have $\Delta_2' \stackrel{\tau}{\Longrightarrow} \cdot$. By the transitivity of $\stackrel{\tau}{\Longrightarrow}$ entails $\Delta_2 \stackrel{\tau}{\Longrightarrow} \cdot$.

2. Suppose $\Delta_1 \preceq_s \Delta_2$ and $\Delta_1 \stackrel{\tau}{\Longrightarrow} (\Delta_1', \Delta_1'')$. By (1), there exists Δ_2''' such that $\Delta_2 \stackrel{\tau}{\Longrightarrow} \Delta_2'''$ and $(\Delta_1', \Delta_1'') \preceq_s \Delta_2'''$. By Definition 4.2, there exist Δ_2' and Δ_2'' such that $\Delta_2''' \stackrel{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$, $\Delta_1' \preceq_s \Delta_2'$ and $\Delta_1'' \preceq_s \Delta_2''$. By the transitivity of $\stackrel{\tau}{\Longrightarrow}$, we have $\Delta_2 \stackrel{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$.

3. Suppose $\Delta_1 \stackrel{\alpha}{\Longrightarrow} \Delta_1'$ where $\alpha \neq \tau$. Then there are Δ_{11} and Δ_{12} with $\Delta_1 \stackrel{\tau}{\Longrightarrow} \Delta_{11} \stackrel{\alpha}{\longrightarrow} \Delta_{12} \stackrel{\tau}{\Longrightarrow} \Delta_1'$. By (1), there is some Δ_{21} such that $\Delta_2 \stackrel{\tau}{\Longrightarrow} \Delta_{21}$ and $\Delta_{11} \preceq_s \Delta_{21}$. By Definition 4.2, there is Δ_{22} such that $\Delta_{21} \stackrel{\alpha}{\Longrightarrow} \Delta_{22}$ and $\Delta_{12} \preceq_s \Delta_{22}$. By (1) again, there is Δ_2' with $\Delta_{22} \stackrel{\tau}{\Longrightarrow} \Delta_2'$ with $\Delta_1' \preceq_s \Delta_2'$. Note that we also have $\Delta_2 \stackrel{\alpha}{\Longrightarrow} \Delta_2'$.

□

Lemma 4.8 (Reflexivity of $\preceq_s$)**.** *For all contexts* Δ*, we have* $\Delta \preceq_s \Delta$.

Proof. Consider the identity relation $\mathcal{R}_{id}$. It will suffice to show that $\mathcal{R}_{id} \subseteq \preceq_s$, because $\Delta \; \mathcal{R}_{id} \; \Delta$ always holds, which will in turn imply $\Delta \preceq_s \Delta$. We can show that $\mathcal{R}_{id}$ meets the four criteria for simulation. Suppose $\Delta \; \mathcal{R}_{id}$ for some Δ.

1. Assume that $\Delta \equiv \cdot$. Then, in particular, $\Delta \stackrel{\tau}{\Longrightarrow} \cdot$.

2. Assume that $\Delta \equiv (\Delta', \Delta'')$. Then, in particular, $\Delta \stackrel{\tau}{\Longrightarrow} (\Delta', \Delta'')$, and we also have $\Delta' \; \mathcal{R}_{id} \; \Delta'$ and $\Delta'' \; \mathcal{R}_{id} \; \Delta''$.

3. Assume that $\Delta \stackrel{\alpha}{\longrightarrow} \Delta'$. Then, in particular, $\Delta \stackrel{\alpha}{\Longrightarrow} \Delta'$ and we have $\Delta' \; \mathcal{R}_{id} \; \Delta'$.

4. Assume that $\Delta \xrightarrow{?a} \Delta'$, which entails that $\Delta \equiv (a \multimap B, \Delta'')$ and $\Delta' \equiv (B, \Delta'')$. Because $\Delta' \ \mathcal{R}_{id} \ \Delta'$ (it is the identity relation!), it is sufficient to show that $(\Delta, a) \stackrel{\tau}{\Longrightarrow} \Delta'$, i.e., that $(a, a \multimap B, \Delta'') \stackrel{\tau}{\Longrightarrow} (B, \Delta'')$. However, we know that $a \xrightarrow{!a} \cdot$ and $(a \multimap B, \Delta'') \xrightarrow{?a} (B, \Delta'')$ by the rules $lts!$ and $lts?$ in Figure 4, respectively. We can now combine them using rule $lts!?$ in the desired reduction for $(a, a \multimap B, \Delta'') \xrightarrow{\tau} (B, \Delta'')$.

□

Proposition 4.9 (Compositionality of $\preceq_s$)**.** *If* $\Delta_1 \preceq_s \Delta_2$*, then* $(\Delta, \Delta_1) \preceq_s (\Delta, \Delta_2)$.

Proof. Consider the relation

$$\mathcal{R} \ := \ \{((\Delta, \Delta_1), (\Delta, \Delta_2)) \mid \Delta_1 \preceq_s \Delta_2\}.$$

It will suffice to show that $\mathcal{R} \subseteq \preceq_s$, because in that case, given $\Delta_1 \preceq_s \Delta_2$, we have that $(\Delta, \Delta_1) \ \mathcal{R} \ (\Delta, \Delta_2)$ for any Δ, which will in turn imply $(\Delta, \Delta_1) \preceq_s (\Delta, \Delta_2)$. We can show that $\mathcal{R}_{id}$ meets the four criteria for simulation. Suppose $(\Delta, \Delta_1) \ \mathcal{R} \ (\Delta, \Delta_2)$, which means that we also have $\Delta_1 \preceq_s \Delta_2$.

1. Let us show that if $(\Delta, \Delta_1) \ \mathcal{R} \ (\Delta, \Delta_2)$ with $(\Delta, \Delta_1) \equiv \cdot$, then $(\Delta, \Delta_2) \stackrel{\tau}{\Longrightarrow} \cdot$.

   If $(\Delta, \Delta_1) \equiv \cdot$, then $\Delta \equiv \cdot$. Moreover, by definition of $\mathcal{R}$, we have that $\cdot \preceq_s \Delta_2$. Now, because $\preceq_s$ is a simulation, we have that $\Delta_2 \stackrel{\tau}{\Longrightarrow} \cdot$. Since $(\Delta, \Delta_2) \equiv \Delta_2$, we conclude that $(\Delta, \Delta_2) \stackrel{\tau}{\Longrightarrow} \cdot$, as desired.

2. Let us prove that if $(\Delta, \Delta_1) \equiv (\Delta_1', \Delta_1'')$, then $(\Delta, \Delta_2) \stackrel{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$ such that $\Delta_1' \ \mathcal{R} \ \Delta_2'$ and $\Delta_1'' \ \mathcal{R} \ \Delta_2''$.

   If (Δ, Δ_1) can be decomposed into (Δ_1', Δ_1'') for some Δ_1' and Δ_1'', we need to find some Δ_2' and Δ_2'' such that $(\Delta, \Delta_2) \stackrel{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$ with $\Delta_1' \ \mathcal{R} \ \Delta_2'$ and $\Delta_1'' \ \mathcal{R} \ \Delta_2''$. Without loss of generality, assume that we have the decomposition of (Δ, Δ_1) with $\Delta = (\Delta^a, \Delta^b)$ and $\Delta_1 = (\Delta_1^a, \Delta_1^b)$ such that $\Delta_1' = (\Delta^a, \Delta_1^a)$ and $\Delta_1'' = (\Delta^b, \Delta_1^b)$. Since $\Delta_1 \preceq_s \Delta_2$, there exists some transition $\Delta_2 \stackrel{\tau}{\Longrightarrow} (\Delta_2^a, \Delta_2^b)$ such that $\Delta_1^a \preceq_s \Delta_2^a$ and $\Delta_1^b \preceq_s \Delta_2^b$. It follows by compositionality (Lemma 4.4) that

   $$(\Delta, \Delta_2) \stackrel{\tau}{\Longrightarrow} (\Delta^a, \Delta^b, \Delta_2^a, \Delta_2^b) \equiv (\Delta^a, \Delta_2^a, \Delta^b, \Delta_2^b)$$

   Let $\Delta_2' = (\Delta^a, \Delta_2^a)$ and $\Delta_2'' = (\Delta^b, \Delta_2^b)$. We observe that $\Delta_1' \ \mathcal{R} \ \Delta_2'$ and $\Delta_1'' \ \mathcal{R} \ \Delta_2''$, as required.

3. Let us show that if $(\Delta, \Delta_1) \xrightarrow{\alpha} \Delta_1'$ then there is Δ_2' such that $(\Delta, \Delta_2) \stackrel{\alpha}{\Longrightarrow} \Delta_2'$ and $\Delta_1' \ \mathcal{R} \ \Delta_2'$, and if $(\Delta, \Delta_1) \xrightarrow{?a} \Delta_1'$ then there is Δ_2' such that $(\Delta, \Delta_2, a) \stackrel{\tau}{\Longrightarrow} \Delta_2'$ and $\Delta_1' \ \mathcal{R} \ \Delta_2'$. It is convenient to prove both these parts of Definition 4.2 together.

   Assume that $(\Delta, \Delta_1) \xrightarrow{\beta} \Delta^*$. There are four cases, according to Lemma 4.5.

   (a) $(\Delta, \Delta_1) \xrightarrow{\beta} (\Delta', \Delta_1)$ because of the transition $\Delta \xrightarrow{\beta} \Delta'$. If $\beta = \alpha$, then by Lemma 4.4 (1) we also have $(\Delta, \Delta_2) \xrightarrow{\alpha} (\Delta', \Delta_2)$ and clearly $(\Delta', \Delta_1) \ \mathcal{R} \ (\Delta', \Delta_2)$. If $\beta = ?a$, then $(\Delta, a) \xrightarrow{\tau} \Delta'$ and thus $(\Delta, \Delta_2, a) \xrightarrow{\tau} (\Delta', \Delta_2)$. Again, we have $(\Delta', \Delta_1) \ \mathcal{R} \ (\Delta', \Delta_2)$.

(b) If $(\Delta, \Delta_1) \xrightarrow{\beta} (\Delta, \Delta_1')$ because of the transition $\Delta_1 \xrightarrow{\beta} \Delta_1'$, since $\Delta_1 \preceq_s \Delta_2$ there are two possibilities. If $\beta = \alpha$, then there is a matching transition $\Delta_2 \stackrel{\alpha}{\Longrightarrow} \Delta_2'$ and $\Delta_1' \preceq_s \Delta_2'$. It follows that $(\Delta, \Delta_1') \stackrel{\alpha}{\Longrightarrow} (\Delta, \Delta_2')$ by Lemma 4.4 (2) and $(\Delta, \Delta_1') \mathcal{R} (\Delta, \Delta_2')$. If $\beta = ?a$, then $\Delta_2, a \stackrel{\tau}{\Longrightarrow} \Delta_2'$ for some Δ_2' with $\Delta_1' \preceq_s \Delta_2'$. We also have $(\Delta, \Delta_2, a) \stackrel{\tau}{\Longrightarrow} (\Delta, \Delta_2')$ by Lemma 4.4 (2) and $(\Delta, \Delta_1') \mathcal{R} (\Delta, \Delta_2')$.

(c) If $(\Delta, \Delta_1) \xrightarrow{\tau} (\Delta', \Delta_1')$ because of the transitions $\Delta \xrightarrow{?a} \Delta'$ and $\Delta_1 \xrightarrow{!a} \Delta_1'$, since $\Delta_1 \preceq_s \Delta_2$ there is a transition $\Delta_2 \stackrel{!a}{\Longrightarrow} \Delta_2'$ with $\Delta_1' \preceq_s \Delta_2'$. It follows that $(\Delta, \Delta_2) \stackrel{\tau}{\Longrightarrow} (\Delta', \Delta_2')$ and we have $(\Delta', \Delta_1') \mathcal{R} (\Delta', \Delta_2')$.

(d) If $(\Delta, \Delta_1) \xrightarrow{\tau} (\Delta', \Delta_1')$ because of the transitions $\Delta \xrightarrow{!a} \Delta'$ and $\Delta_1 \xrightarrow{?a} \Delta_1'$, then this can be simulated by a transition from (Δ, Δ_2). The reason is as follows. In order for Δ to enable the transition $\Delta \xrightarrow{!a} \Delta'$, it must be the case that $\Delta \equiv \Delta', a$. Since $\Delta_1 \preceq_s \Delta_2$ we know that $(\Delta_2, a) \stackrel{\tau}{\Longrightarrow} \Delta_2'$ for some Δ_2' with $\Delta_1' \preceq_s \Delta_2'$. Therefore, we obtain that $(\Delta, \Delta_2) \equiv (\Delta', a, \Delta_2) \stackrel{\tau}{\Longrightarrow} (\Delta', \Delta_2')$ and $(\Delta', \Delta_1') \mathcal{R} (\Delta', \Delta_2')$.

In summary, we have verified that $\mathcal{R}$ is a simulation. □

Lemma 4.10 (Transitivity of $\preceq_s$)**.** *If* $\Delta_1 \preceq_s \Delta_2$ *and* $\Delta_2 \preceq_s \Delta_3$*, then* $\Delta_1 \preceq_s \Delta_3$.

Proof. Consider the relation

$$\mathcal{R} \;:=\; \{(\Delta_1, \Delta_3) \mid \text{there exists } \Delta_2 \text{ with } \Delta_1 \preceq_s \Delta_2 \text{ and } \Delta_2 \preceq_s \Delta_3\}$$

It will suffice to show that $\mathcal{R} \subseteq \preceq_s$, because in that case, given $\Delta_1 \preceq_s \Delta_2$ and $\Delta_2 \preceq_s \Delta_3$, we will know that $\Delta_1 \mathcal{R} \Delta_3$, which will in turn imply $\Delta_1 \preceq_s \Delta_3$. We can show that $\mathcal{R}$ meets the four criteria for simulation. Suppose $\Delta_1 \mathcal{R} \Delta_3$, that is $\Delta_1 \preceq_s \Delta_2$ and $\Delta_2 \preceq_s \Delta_3$ for some Δ_2.

1. If $\Delta_1 \equiv \cdot$, then $\Delta_2 \stackrel{\tau}{\Longrightarrow} \cdot$. By Lemma 4.7, we have $\Delta_3 \stackrel{\tau}{\Longrightarrow} \cdot$.

2. If $\Delta_1 \equiv (\Delta_1', \Delta_1'')$, then $\Delta_2 \stackrel{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$ for some Δ_2' and Δ_2'' such that $\Delta_1' \preceq_s \Delta_2'$ and $\Delta_1'' \preceq_s \Delta_2''$. By Lemma 4.7, there exist Δ_3' and Δ_3'' such that $\Delta_3 \stackrel{\tau}{\Longrightarrow} (\Delta_3', \Delta_3'')$, $\Delta_2' \preceq_s \Delta_3'$ and $\Delta_2'' \preceq_s \Delta_3''$. Therefore, $\Delta_1' \mathcal{R} \Delta_3'$ and $\Delta_1'' \mathcal{R} \Delta_3''$.

3. If $\Delta_1 \xrightarrow{\alpha} \Delta_1'$, there exists Δ_2' such that $\Delta_2 \stackrel{\alpha}{\Longrightarrow} \Delta_2'$ and $\Delta_1' \preceq_s \Delta_2'$. By Lemma 4.7, there exist Δ_3' such that $\Delta_3 \stackrel{\alpha}{\Longrightarrow} \Delta_3'$ and $\Delta_2' \preceq_s \Delta_3'$, thus $\Delta_1' \mathcal{R} \Delta_3'$.

4. If $\Delta_1 \xrightarrow{?a} \Delta_1'$, there exists Δ_2' such that $(\Delta_2, a) \stackrel{\tau}{\Longrightarrow} \Delta_2'$ and $\Delta_1' \preceq_s \Delta_2'$. By Proposition 4.9 we have $(\Delta_2, a) \preceq_s (\Delta_3, a)$. By Lemma 4.7, there exists Δ_3' with $(\Delta_3, a) \stackrel{\tau}{\Longrightarrow} \Delta_3'$ and $\Delta_2' \preceq_s \Delta_3'$. It follows that $\Delta_1' \mathcal{R} \Delta_3'$. □

Theorem 4.11. $\preceq_s$ *is a preorder.*

Proof. $\preceq_s$ is reflexive (Lemma 4.8) and transitive (Lemma 4.10). □

4.5 Equivalence of the contextual and simulation preorders

We are now in a position to fill in the rightmost portion of the proof map (Figure 4) from the beginning of this section. In this section, we show the equivalence of the largest simulation $\preceq_s$ and the contextual preorder. With the compositionality of $\preceq_s$ (Lemma 4.9) at hand, the soundness proof is straightforward. For the completeness proof, we crucially rely on fresh atom removal (Lemma 3.9).

Theorem 4.12 (Soundness)**.** *If* $\Delta_1 \preceq_s \Delta_2$*, then* $\Delta_1 \preceq_c \Delta_2$.

Proof. We show that all aspects of the definition of the contextual preorder are satisfied.

Barb-preserving We show that $\preceq_s$ is barb-preserving. Suppose $\Delta_1 \preceq_s \Delta_2$ and $\Delta_1 \downarrow_a$. Then $\Delta_1 \equiv (\Delta_1', a)$ for some Δ_1', thus $\Delta_1 \xrightarrow{!a} \Delta_1'$. Since $\Delta_1 \preceq_s \Delta_2$ there exists Δ_2' such that $\Delta_2 \stackrel{!a}{\Longrightarrow} \Delta_2'$, i.e. $\Delta_2 \stackrel{\tau}{\Longrightarrow} \Delta_2'' \xrightarrow{!a} \Delta_2'$. Note that Δ_2'' must be the form (Δ_2''', a) for some Δ_2'''. It follows that $\Delta_2 \Downarrow_a$.

Compositional By Lemma 4.9 $\preceq_s$ is compositional.

Reduction-closed If $\Delta_1 \rightsquigarrow \Delta_1'$, thus $\Delta_1 \xrightarrow{\tau} \Delta_1'$ by Lemma 4.1. By the third condition of Definition 4.2 there exists Δ_2' such that $\Delta_2 \stackrel{\tau}{\Longrightarrow} \Delta_2'$ and $\Delta_1' \preceq_s \Delta_2'$. By Lemma 4.1 again, we have $\Delta_2 \rightsquigarrow^* \Delta_2'$.

Partition-preserving

If $\Delta_1 \equiv \cdot$, by the first condition of Definition 4.2 we have $\Delta_2 \stackrel{\tau}{\Longrightarrow} \cdot$. By Lemma 4.1 this means $\Delta_2 \rightsquigarrow^* \cdot$.

If $\Delta_1 \equiv (\Delta_1', \Delta_1'')$, by the second condition of Definition 4.2 there are Δ_2' and Δ_2'' with $\Delta_2 \stackrel{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$, $\Delta_1' \preceq_s \Delta_2'$ and $\Delta_1'' \preceq_s \Delta_2''$. By Lemma 4.1 this means $\Delta_2 \rightsquigarrow^* (\Delta_2', \Delta_2'')$.

□

Theorem 4.13 (Completeness)**.** *If* $\Delta_1 \preceq_c \Delta_2$ *then* $\Delta_1 \preceq_s \Delta_2$.

Proof. Assume that $\Delta_1 \preceq_c \Delta_2$. We need to show that $\preceq_c$ is a simulation. By Definition 4.2, this decomposes into four parts.

1. Let us show that if $\Delta_1 \preceq_c \Delta_2$ and $\Delta_1 \equiv \cdot$ then $\Delta_2 \stackrel{\tau}{\Longrightarrow} \cdot$. Suppose $\Delta_1 \preceq_c \Delta_2$ and $\Delta_1 \equiv \cdot$, then $\Delta_2 \rightsquigarrow^* \cdot$ as $\preceq_c$ is reduction closed. By Lemma 4.1 we have $\Delta_2 \stackrel{\tau}{\Longrightarrow} \cdot$.

2. Let us show that if $\Delta_1 \preceq_c \Delta_2$ and $\Delta_1 \equiv (\Delta_1', \Delta_2')$ then $\Delta_2 \stackrel{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$ for some Δ_2' and Δ_2'' such that $\Delta_1' \preceq_c \Delta_2'$ and $\Delta_1'' \preceq_c \Delta_2''$.

   Suppose $\Delta_1 \preceq_c \Delta_2$ and $\Delta_1 \equiv (\Delta_1', \Delta_1'')$, then $\Delta_2 \rightsquigarrow^* (\Delta_2', \Delta_2'')$ such that $\Delta_1' \preceq_c \Delta_2'$ and $\Delta_1'' \preceq_c \Delta_2''$ as $\preceq_c$ is reduction-closed. By Lemma 4.1 we have $\Delta_2 \stackrel{\tau}{\Longrightarrow} (\Delta_2', \Delta_2'')$.

3. Let us show that if $\Delta_1 \preceq_c \Delta_2$ and $\Delta_1 \xrightarrow{\alpha} \Delta_1'$ then there exists Δ_2' such that $\Delta_2 \stackrel{\alpha}{\Longrightarrow} \Delta_2'$ and $\Delta_1' \preceq_c \Delta_2'$. Suppose $\Delta_1 \preceq_c \Delta_2$ and $\Delta_1 \xrightarrow{\alpha} \Delta_1'$.

- $\alpha \equiv \tau$. By Lemma 4.1 this means $\Delta_1 \rightsquigarrow \Delta_1'$. Since $\preceq_c$ is reduction-closed, there exists Δ_2' such that $\Delta_2 \rightsquigarrow^* \Delta_2'$ and $\Delta_1' \preceq_c \Delta_2'$. By Lemma 4.1 again, we have $\Delta_2 \stackrel{\tau}{\Longrightarrow} \Delta_2'$.
- $\alpha \equiv !a$ for some a. Note that Δ_1 must be in the form (Δ_1', a). Since $\preceq_c$ is partition-preserving, there exist Δ_2' and Δ_a such that

  $$\Delta_2 \rightsquigarrow^* (\Delta_2', \Delta_a) \tag{2}$$

  with $\Delta_1' \preceq_c \Delta_2'$ and $a \preceq_c \Delta_a$. Then $(a \multimap \mathbf{1}, a) \preceq_c (a \multimap \mathbf{1}, \Delta_a)$ by the compositionality of $\preceq_c$. Since $(a \multimap \mathbf{1}, a) \rightsquigarrow^* \cdot$, by Lemma 3.6 we have $(a \multimap \mathbf{1}, \Delta_a) \rightsquigarrow^* \cdot$. Then there exists some Δ_a' such that $\Delta_a \rightsquigarrow^* (\Delta_a', a)$ and $\Delta_a' \rightsquigarrow^* \cdot$, thus

  $$\Delta_a \rightsquigarrow^* a \tag{3}$$

  by transitivity. It follows from (2) and (3) that $\Delta_2 \rightsquigarrow^* (\Delta_2', a)$. By Lemma 4.1 this means $\Delta_2 \stackrel{!a}{\Longrightarrow} \Delta_2'$, which is the desired transition.

4. Let us show that if $\Delta_1 \preceq_c \Delta_2$ and $\Delta_1 \stackrel{?a}{\longrightarrow} \Delta_1'$ then there exists Δ_2' such that $(\Delta_2, a) \stackrel{\tau}{\Longrightarrow} \Delta_2'$ and $\Delta_1' \preceq_c \Delta_2'$.

   Suppose $\Delta_1 \preceq_c \Delta_2$ and $\Delta_1 \stackrel{?a}{\longrightarrow} \Delta_1'$. Let b be a new atomic proposition. Then $(\Delta_1, a, b) \stackrel{\tau}{\longrightarrow} (\Delta_1', b)$, thus $(\Delta_1, a, b) \rightsquigarrow (\Delta_1', b)$ by Lemma 4.1. Since $\Delta_1 \preceq_c \Delta_2$ we know $(\Delta_1, a, b) \preceq_c (\Delta_2, a, b)$ by the compositionality of $\preceq_c$. So there exists Δ^* such that $(\Delta_2, a, b) \rightsquigarrow^* \Delta^*$ and $(\Delta_1', b) \preceq_c \Delta^*$. Note that Δ^* must be in the form (Δ^{**}, b) because b does not occur in Δ_2 and there is no communication between b and Δ_2. It follows that $(\Delta_2, a) \rightsquigarrow^* \Delta^{**}$. By Lemma 4.1 this means $(\Delta_2, a) \stackrel{\tau}{\Longrightarrow} \Delta^{**}$. By Lemma 3.9 and $(\Delta_1', b) \preceq_c (\Delta^{**}, b)$ we know that $\Delta_1' \preceq_c \Delta^{**}$. □

4.6 Equivalence of the logical and simulation preorders

We will now start to fill in the remaining portions of the proof map (Figure 4) from the beginning of this section. First, we prove the soundness and completeness of *derivability* relative to simulation, and then we use this to prove the soundness and completeness of the contextual preorder relative to the logical preorder.

Theorem 4.14. *If $\Delta \vdash A$, then $A \preceq_s \Delta$.*

Proof. We proceed by rule induction, where the rules are given in Figure 1.

- (rule $\top R$) If $\Delta \vdash \top$ then we have $\top \preceq_s \Delta$ vacuously, as $\top$ is a nonempty process state that can make no transitions.
- (rule $\mathbf{1}R$) If $\cdot \vdash \mathbf{1}$ then it is trivial to see that $\mathbf{1} \preceq_s \cdot$.
- (rule $init$) If $a \vdash a$ then $a \preceq_s a$ follows from the reflexivity of $\preceq_s$.

- (rule $\multimap R$) Suppose $\Delta \vdash a \multimap A$ is derived from $\Delta, a \vdash A$. By induction, we have

  (4) $$A \preceq_s (\Delta, a)$$

  The only transition from $a \multimap A$ is $(a \multimap A) \xrightarrow{?a} A$. It is matched by the trivial transition $(\Delta, a) \stackrel{\tau}{\Longrightarrow} (\Delta, a)$ in view of (4).

- (rule $\multimap L$) Suppose $(\Delta_1, \Delta_2, a \multimap A) \vdash B$ is derived from $\Delta_1 \vdash a$ and $(\Delta_2, A) \vdash B$. By induction, we have

  (5) $$a \preceq_s \Delta_1 \quad \text{and} \quad B \preceq_s (\Delta_2, A)$$

  By the first part of (5) we know that there is some Δ_1' such that $\Delta_1 \stackrel{!a}{\Longrightarrow} \Delta_1'$ and $\cdot \preceq_s \Delta_1'$. It is easy to see that $\Delta_1 \stackrel{\tau}{\Longrightarrow} (\Delta_1', a)$ and $\Delta_1' \stackrel{\tau}{\Longrightarrow} \cdot$. Then

  $$\begin{array}{rcl} (\Delta_1, \Delta_2, a \multimap A) & \stackrel{\tau}{\Longrightarrow} & (\Delta_1', a, \Delta_2, a \multimap A) \\ & \stackrel{\tau}{\longrightarrow} & (\Delta_1', \Delta_2, A) \\ & \stackrel{\tau}{\Longrightarrow} & (\Delta_2, A) \end{array}$$

  In other words, we have $(\Delta_1, \Delta_2, a \multimap A) \stackrel{\tau}{\Longrightarrow} (\Delta_2, A)$. By Lemma 4.6 it follows that $(\Delta_2, A) \preceq_s (\Delta_1, \Delta_2, a \multimap A)$. By transitivity ($\preceq_s$ is a preorder, Theorem 4.11), we can combine this with the second part of (5), yielding

  $$B \preceq_s (\Delta_1, \Delta_2, a \multimap A).$$

- (rule $\otimes R$) Suppose $(\Delta_1, \Delta_2) \vdash A \otimes B$ is derived from $\Delta_1 \vdash A$ and $\Delta_2 \vdash B$. By induction, we have

  (6) $$A \preceq_s \Delta_1 \quad \text{and} \quad B \preceq_s \Delta_2$$

  Now the only transition from $A \otimes B$ is $A \otimes B \xrightarrow{\tau} (A, B)$. It can be matched by the trivial transition $(\Delta_1, \Delta_2) \stackrel{\tau}{\Longrightarrow} (\Delta_1, \Delta_2)$ because the by compositionality of $\preceq_s$ (Lemma 4.9 and (6) we know that

  $$(A, B) \preceq_s (\Delta_1, B) \preceq_s (\Delta_1, \Delta_2).$$

  Now it is immediate that $(A, B) \preceq_s (\Delta_1, \Delta_2)$ by transitivity ($\preceq_s$ is a preorder, Theorem 4.11).

- (rule $\otimes L$) Suppose $(\Delta, A \otimes B) \vdash C$ is derived from $(\Delta, A, B) \vdash C$. By induction we have

  (7) $$C \preceq_s (\Delta, A, B)$$

  Since $(\Delta, A \otimes B) \xrightarrow{\tau} (\Delta, A, B)$, we apply Lemma 4.6 and obtain

  (8) $$(\Delta, A, B) \preceq_s (\Delta, A \otimes B)$$

  By (7), (8) and the transitivity of $\preceq_s$, we have $C \preceq_s (\Delta, A \otimes B)$.

- (rules $\mathbf{1}L$, $\&L_1$ and $\&L_2$) Similar.
- (rule $\&R$) Suppose $\Delta \vdash (A \,\&\, B)$ is derived from $\Delta \vdash A$ and $\Delta \vdash B$. By induction we have

  (9) $$A \preceq_s \Delta \quad \text{and} \quad B \preceq_s \Delta$$

  The only transitions from $A \,\&\, B$ are $(A \,\&\, B) \xrightarrow{\tau} A$ and $(A \,\&\, B) \xrightarrow{\tau} B$. Both of them can be matched by the trivial transition $\Delta \stackrel{\tau}{\Longrightarrow} \Delta$ in view of (9).

□

Proposition 4.15. *If $\Delta_1 \stackrel{\tau}{\Longrightarrow} \Delta_2$ and $\Delta_2 \vdash A$, then $\Delta_1 \vdash A$.*

Proof. Immediate by Proposition 3.1 and Lemma 4.1. □

Theorem 4.16. *If $A \preceq_s \Delta$, then $\Delta \vdash A$.*

Proof. We proceed by induction on the structure of A.

- $A \equiv \top$. By rule $\top R$ we have $\Delta \vdash \top$.
- $A \equiv \mathbf{1}$. Then we also have $A \equiv \cdot$. Since $A \preceq_s \Delta$, it must be the case that $\Delta \stackrel{\tau}{\Longrightarrow} \cdot$. By rule $\mathbf{1}R$, we have $\cdot \vdash \mathbf{1}$. By Proposition 4.15 it follows that $\Delta \vdash \mathbf{1}$.
- $A \equiv a$. Since $A \preceq_s \Delta$ and $A \xrightarrow{!a} \cdot$, there is Δ' such that

  (10) $$\Delta \stackrel{!a}{\Longrightarrow} \Delta' \qquad \text{and} \qquad \cdot \preceq_s \Delta'.$$

  From the first part of (10), we obtain $\Delta \stackrel{\tau}{\Longrightarrow} (\Delta'', a)$ for some Δ'' with $\Delta'' \stackrel{\tau}{\Longrightarrow} \Delta'$. From the second part, we have $\Delta' \stackrel{\tau}{\Longrightarrow} \cdot$. Combining them together yields $\Delta \stackrel{\tau}{\Longrightarrow} a$. By rule $init$ we can infer $a \vdash a$. Then it follows from Proposition 4.15 that $\Delta \vdash a$.
- $A \equiv a \multimap A'$. Since $A \preceq_s \Delta$ and $A \xrightarrow{?a} A'$, there is Δ' such that $(\Delta, a) \stackrel{\tau}{\Longrightarrow} \Delta'$ and $A' \preceq_s \Delta'$. By induction, we know that $\Delta' \vdash A'$. By Proposition 4.15 it follows that $(\Delta, a) \vdash A'$. Now use rule $\multimap R$ we obtain $\Delta \vdash (a \multimap A')$.
- $A \equiv A_1 \,\&\, A_2$. Since $A \preceq_s \Delta$ and $A \xrightarrow{\tau} A_1$, there is Δ_1 such that $\Delta \stackrel{\tau}{\Longrightarrow} \Delta_1$ and $A_1 \preceq_s \Delta_1$. By induction, we have $\Delta_1 \vdash A_1$. By Proposition 4.15 it follows that $\Delta \vdash A_1$. By similar argument, we see that $\Delta \vdash A_2$. Hence, it follows from rule $\&R$ that $\Delta \vdash (A_1 \,\&\, A_2)$.
- $A \equiv A_1 \otimes A_2$. Since $A \preceq_s \Delta$ and $A \xrightarrow{\tau} (A_1, A_2)$, we apply Lemma 4.7 and derive some transition $\Delta \stackrel{\tau}{\Longrightarrow} (\Delta_1, \Delta_2)$ such that $A_1 \preceq_s \Delta_1$ and $A_2 \preceq_s \Delta_2$. By induction, we obtain $\Delta_1 \vdash A_1$ and $\Delta_2 \vdash A_2$. It follows from rule $\otimes R$ that $\Delta_1, \Delta_2 \vdash A_1 \otimes A_2$, that is $\Delta \vdash A$.

Therefore $\Delta \vdash A$ for all A and Δ. □

The next important property is obtained mostly by applying Theorems 4.14 and 4.16.

Theorem 4.17. $\Delta_1 \preceq_l \Delta_2$ *if and only if* $\Delta_1 \preceq_s \Delta_2$.

Proof. ($\Rightarrow$) Suppose $\Delta_1 \preceq_l \Delta_2$. It is trivial to see that $\Delta_1 \vdash \bigotimes \Delta_1$. By the definition of logical preorder, it follows that $\Delta_2 \vdash \bigotimes \Delta_1$. By Theorem 4.14 we have

$$\bigotimes \Delta_1 \ \preceq_s \ \Delta_2. \tag{11}$$

Considering the formula $\bigotimes \Delta_1$ as a context, we have $\bigotimes \Delta_1 \stackrel{\tau}{\Longrightarrow} \Delta_1$ according to our reduction semantics. By Lemma 4.6, it follows that

$$\Delta_1 \ \preceq_s \ \bigotimes \Delta_1. \tag{12}$$

By combining (11) and (12), we obtain that $\Delta_1 \preceq_s \Delta_2$ because $\preceq_s$ is transitive by ($\preceq_s$ is a preorder, Theorem 4.11).

($\Leftarrow$) Suppose that $\Delta_1 \preceq_s \Delta_2$. For any Δ and A, assume that $(\Delta, \Delta_1) \vdash A$. By Theorem 4.14 we have

$$A \preceq_s (\Delta, \Delta_1). \tag{13}$$

Since $\Delta_1 \preceq_s \Delta_2$ and $\preceq_s$ is compositional (Lemma 4.9), we obtain

$$(\Delta, \Delta_1) \ \preceq_s \ (\Delta, \Delta_2). \tag{14}$$

By (13), (14) and the transitivity of $\preceq_s$, we see that $A \preceq_s (\Delta, \Delta_2)$. Then Theorem 4.16 yields $(\Delta, \Delta_2) \vdash A$. Therefore, we have shown that $\Delta_1 \preceq_l \Delta_2$.

This concludes the proof of this result. □

Finally, we arrive at the main result of the section.

Corollary 4.18 (Soundness and completeness)**.** $\Delta_1 \preceq_l \Delta_2$ *if and only if* $\Delta_1 \preceq_c \Delta_2$.

Proof. By Theorems 4.12 and 4.13 we know that $\preceq_c$ coincides with $\preceq_s$. Theorem 4.17 tells us that $\preceq_s$ coincides with $\preceq_l$. Hence, the required result follows. □

5 Exponentials

In this section, we extend the investigation by adding the exponential modality "!" from intuitionistic linear logic, which will closely correspond to the replication operator of the π-calculus. Our extension refers to the propositional linear language seen in Sections 3–4. Specifically, the language we will be working on is:

$$\textit{Formulas} \qquad A, B, C \ ::= \ a \mid \mathbf{1} \mid A \otimes B \mid a \multimap B \mid \top \mid A \,\&\, B \mid !A$$

Observe that this language still limits the antecedent of linear implications to be an atomic proposition — this is a common restriction when investigating fragments of linear logic that correspond to CCS-like process algebras [3, 4, 5]

The structure of this section is similar to our development for the language without exponentials: we first present the extended language and a notion of states in our process interpretation in Section 5.1, then we connect contextual preorder with logical preorder by making use of simulation in Section 5.2.

5.1 Process interpretation and contextual preorder

The π-calculus reading of the language with exponentials is extended by interpreting the exponential ! as the replication operator !.

$$\begin{array}{ll} \ldots & \ldots \\ !A & \text{any number of copies of process } A \end{array}$$

The structural equivalences seen in Section 3 are updated by adding an inert unrestricted context Γ and with the following rules:

$$\begin{array}{rcl} (\Gamma, \cdot;\ \Delta) & \equiv & (\Gamma;\ \Delta) \\ (\Gamma_1, \Gamma_2;\ \Delta) & \equiv & (\Gamma_2, \Gamma_1;\ \Delta) \\ (\Gamma_1, (\Gamma_2, \Gamma_3);\ \Delta) & \equiv & ((\Gamma_1, \Gamma_2), \Gamma_3;\ \Delta) \\ (\Gamma, A, A;\ \Delta) & \equiv & (\Gamma, A;\ \Delta) \end{array}$$

These rule entail that the unrestricted context behaves like a set.

The reductions in Figure 2 are upgraded with an inert context Γ, and the following two reductions are added:

$$\begin{array}{rcll} (\Gamma, A;\ \Delta) & \rightsquigarrow & (\Gamma, A;\ \Delta, A) & (\rightsquigarrow \text{clone}) \\ (\Gamma;\ \Delta, !A) & \rightsquigarrow & (\Gamma, A;\ \Delta) & (\rightsquigarrow !) \end{array}$$

The *composition* of two states $(\Gamma_1; \Delta_1)$ and $(\Gamma_2; \Delta_2)$, written $((\Gamma_1; \Delta_1), (\Gamma_2; \Delta_2))$, is defined as the state $((\Gamma_1, \Gamma_2); (\Delta_1, \Delta_2))$. Recall that unrestricted contexts are set so that Γ_1, Γ_2 may collapse identical formulas occurring in both Γ_1 and Γ_2 (while linear contexts can contain duplicates).

A *partition* of a state $(\Gamma; \Delta)$ is any pair of states $(\Gamma_1; \Delta_1)$ and $(\Gamma_2; \Delta_2)$ such that $(\Gamma; \Delta) = ((\Gamma_1; \Delta_1), (\Gamma_2; \Delta_2))$.

We write $(\Gamma; \Delta)\downarrow_a$ whenever $a \in \Delta$, and $\Delta \Downarrow_a$ whenever $(\Gamma; \Delta) \rightsquigarrow^* (\Gamma'; \Delta')$ for some $(\Gamma'; \Delta')$ with $(\Gamma'; \Delta')\downarrow_a$. The definition of contextual preorder given in Definition 3.3 now takes the following form.

Definition 5.1 (Contextual preorder)**.** *Let $\mathcal{R}$ be a binary relation over states. We say that $\mathcal{R}$ is*

- barb-preserving *if, whenever* $(\Gamma_1; \Delta_1)\ \mathcal{R}\ (\Gamma_2; \Delta_2)$ *and* $(\Gamma_1; \Delta_1)\downarrow_a$*, we have that* $(\Gamma_2; \Delta_2)\Downarrow_a$ *for any* a*.*
- reduction-closed *if* $(\Gamma_1; \Delta_1)\ \mathcal{R}\ (\Gamma_2; \Delta_2)$ *and* $(\Gamma_1; \Delta_1) \rightsquigarrow (\Gamma_1'; \Delta_1')$ *implies* $(\Gamma_2; \Delta_2) \rightsquigarrow^* (\Gamma_2'; \Delta_2')$ *and* $(\Gamma_1'; \Delta_1')\ \mathcal{R}\ (\Gamma_2'; \Delta_2')$ *for some* $(\Gamma_2'; \Delta_2')$*.*
- compositional *if* $(\Gamma_1; \Delta_1)\ \mathcal{R}\ (\Gamma_2; \Delta_2)$ *implies* $((\Gamma_1; \Delta_1), (\Gamma; \Delta))\ \mathcal{R}\ ((\Gamma_2; \Delta_2), (\Gamma; \Delta))$ *for all* $(\Gamma; \Delta)$*.*
- partition-preserving *if* $(\Gamma_1; \Delta_1)\ \mathcal{R}\ (\Gamma_2; \Delta_2)$ *implies that*
  1. *if* $\Delta_1 = \cdot$*, then* $(\Gamma_2; \Delta_2) \rightsquigarrow^* (\Gamma_2'; \cdot)$ *and* $(\Gamma_1; \cdot)\ \mathcal{R}\ (\Gamma_2'; \cdot)$*,*
  2. *for all* $(\Gamma_1'; \Delta_1')$ *and* $(\Gamma_1''; \Delta_1'')$*, if* $(\Gamma_1; \Delta_1) = ((\Gamma_1'; \Delta_1'), (\Gamma_1''; \Delta_1''))$ *then there exists* $(\Gamma_2'; \Delta_2')$ *and* $(\Gamma_2''; \Delta_2'')$ *such that* $(\Gamma_2; \Delta_2) \rightsquigarrow^* ((\Gamma_2'; \Delta_2'), (\Gamma_2''; \Delta_2''))$ *and furthermore* $(\Gamma_1'; \Delta_1')\ \mathcal{R}\ (\Gamma_2'; \Delta_2')$ *and* $(\Gamma_1''; \Delta_1'')\ \mathcal{R}\ (\Gamma_2''; \Delta_2'')$*,*

$$\frac{}{(\Gamma;\Delta,a)\xrightarrow{!a}(\Gamma;\Delta)}\ lts! \qquad \frac{}{(\Gamma;\Delta,a\multimap B)\xrightarrow{?a}(\Gamma;\Delta,B)}\ lts?$$

$$\frac{}{(\Gamma;\Delta,\mathbf{1})\xrightarrow{\tau}(\Gamma;\Delta)}\ lts\mathbf{1} \qquad \frac{}{(\Gamma;\Delta,A\otimes B)\xrightarrow{\tau}(\Gamma;\Delta,A,B)}\ lts\otimes$$

$$\frac{}{(\Gamma;\Delta,A\,\&\,B)\xrightarrow{\tau}(\Gamma;\Delta,A)}\ lts\,\&_1 \qquad \frac{}{(\Gamma;\Delta,A\,\&\,B)\xrightarrow{\tau}(\Gamma;\Delta,B)}\ lts\,\&_2$$

$$\frac{}{(\Gamma;\Delta,!A)\xrightarrow{\tau}(\Gamma,A;\Delta)}\ lts!A \qquad \text{(No rule for } \top\text{)}$$

$$\frac{}{(\Gamma,A;\Delta)\xrightarrow{\tau}(\Gamma,A;\Delta,A)}\ ltsClone$$

$$\frac{(\Gamma_1;\Delta_1)\xrightarrow{!a}(\Gamma_1';\Delta_1') \quad (\Gamma_2;\Delta_2)\xrightarrow{?a}(\Gamma_2';\Delta_2')}{(\Gamma_1,\Gamma_2;\Delta_1,\Delta_2)\xrightarrow{\tau}(\Gamma_1',\Gamma_2';\Delta_1',\Delta_2')}\ lts!?$$

Figure 5: Labeled Transition System with Exponentials

The contextual preorder*, denoted by $\preceq_c$, is the largest relation over processes which is barb-preserving, reduction-closed, compositional and partition-preserving.*

Observe that this definition is structurally identical to our original notion of contextual preorder (Definitions 3.2–3.3). Indeed, we have simply expressed the notions of composing and partitioning states as explicit operations while our original definition relied on context composition, which is what these notion specialize to when we only have linear contexts. The present definition appears to be quite robust and we have used it in extensions of this work to larger languages.

5.2 Logical preorder and contextual preorder

The labeled transition semantics for our language with banged formulas is given in Figure 5. The following is an updated version of Lemma 4.1.

Lemma 5.2. $(\Gamma_1,\Delta_1)\stackrel{\tau}{\Longrightarrow}(\Gamma_2,\Delta_2)$ *if and only if* $(\Gamma_1,\Delta_1)\rightsquigarrow^*(\Gamma_2,\Delta_2)$.

In the new semantics our definition of simulation is in the following form.

Definition 5.3 (Simulation)**.** *A relation $\mathcal{R}$ between two processes represented as $(\Gamma_1;\Delta_1)$ and $(\Gamma_2;\Delta_2)$ is a* simulation *if $(\Gamma_1;\Delta_1)\ \mathcal{R}\ (\Gamma_2;\Delta_2)$ implies*

1. *if $(\Gamma_1;\Delta_1)\equiv(\Gamma_1';\cdot)$ then $(\Gamma_2;\Delta_2)\stackrel{\tau}{\Longrightarrow}(\Gamma_2';\cdot)$ and $(\Gamma_1';\cdot)\ \mathcal{R}\ (\Gamma_2';\cdot)$.*
2. *if $(\Gamma_1;\Delta_1)\equiv((\Gamma_1';\Delta_1'),(\Gamma_1'';\Delta_1''))$ then $(\Gamma_2;\Delta_2)\stackrel{\tau}{\Longrightarrow}((\Gamma_2';\Delta_2'),(\Gamma_2'';\Delta_2''))$ for some $(\Gamma_2';\Delta_2')$ and $(\Gamma_2'';\Delta_2'')$ such that $(\Gamma_1';\Delta_1')\ \mathcal{R}\ (\Gamma_2';\Delta_2')$ and $(\Gamma_1'';\Delta_1'')\ \mathcal{R}\ (\Gamma_2'';\Delta_2'')$.*

3. *whenever* $(\Gamma_1;\Delta_1) \xrightarrow{\alpha} (\Gamma_1';\Delta_1')$*, there exists* $(\Gamma_2';\Delta_2')$ *such that* $(\Gamma_2;\Delta_2) \stackrel{\alpha}{\Longrightarrow} (\Gamma_2';\Delta_2')$ *and* $(\Gamma_1';\Delta_1')\ \mathcal{R}\ (\Gamma_2';\Delta_2')$.

4. *whenever* $(\Gamma_1;\Delta_1) \xrightarrow{?a} (\Gamma_1';\Delta_1')$*, there exists* $(\Gamma_2';\Delta_2')$ *such that* $(\Gamma_2;\ \Delta_2, a) \stackrel{\tau}{\Longrightarrow} (\Gamma_2';\Delta_2')$ *and* $(\Gamma_1';\Delta_1')\ \mathcal{R}\ (\Gamma_2';\Delta_2')$.

We write $(\Gamma_1;\Delta_1) \preceq_s (\Gamma_2;\Delta_2)$ *if there is some simulation* $\mathcal{R}$ *with* $(\Gamma_1;\Delta_1)\ \mathcal{R}\ (\Gamma_2;\Delta_2)$.

Example 5.4. *We have mentioned before that* $!A$ *intuitively represents any number of copies of* A*. Then it is natural to identify* $!!A$ *and* $!A$*, as in some presentations of the* π*-calculus. For instance, we have that*

$$(\cdot;!!a) \preceq_s (\cdot;!a) \qquad \textit{and} \qquad (\cdot;!a) \preceq_s (\cdot;!!a). \tag{15}$$

To prove the first inequality, consider the two sets

$$\begin{array}{rcl} S_1 & = & \{(\cdot;!!a)\} \cup \{(!a;(!a)^n) \mid n \geq 0\} \cup \{(!a, a;\ (!a)^n, a^m) \mid n \geq 0,\ m \geq 0\} \\ S_2 & = & \{(\cdot;!a)\} \cup \{a; a^n) \mid n \geq 0\} \end{array}$$

where we write A^0 *for "*$\cdot$*" and* A^n *for* n *copies of* A *where* $n > 0$*. Let* $\mathcal{R}$ *be* $S_1 \times S_2$*, the Cartesian product of* S_1 *and* S_2*. It can be checked that* $\mathcal{R}$ *is a simulation relation. In the same way, one can see that* $S_2 \times S_1$ *is also a simulation relation, which implies the second inequality in (15).*

By adapting the proof of Proposition 4.9 to the case with exponentials, we have the compositionality of $\preceq_s$.

Proposition 5.5. *If* $(\Gamma_1;\Delta_1) \preceq_s (\Gamma_2,\Delta_2)$ *then* $(\Gamma_1,\Gamma;\Delta_1,\Delta) \preceq_s (\Gamma_2,\Gamma;\Delta_2,\Delta)$ *for any process state* (Γ,Δ).

Similar to Theorems 4.12 and 4.13, it can be shown that the following coincidence result holds.

Theorem 5.6. $(\Gamma_1;\Delta_1) \preceq_c (\Gamma_2;\Delta_2)$ *if and only if* $(\Gamma_1;\Delta_1) \preceq_s (\Gamma_2;\Delta_2)$.

The rest of this subsection is devoted to showing the coincidence of $\preceq_l$ and $\preceq_s$, by following the schema in Section 4.6. We first need two technical lemmas whose proofs are simple and thus omitted.

Lemma 5.7 (Weakening)**.** $(\Gamma;\Delta) \preceq_s ((\Gamma,\Gamma');\Delta)$ *for any* Γ'.

Lemma 5.8. *If* $(\Gamma_1;\Delta_1) \stackrel{\tau}{\Longrightarrow} (\Gamma_2;\Delta_2)$ *then* $(\Gamma_2;\Delta_2) \preceq_s (\Gamma_1;\Delta_1)$.

We are now in a position to connect simulation with provability. First, we state a result akin to Theorem 4.14.

Theorem 5.9. *If* $\Gamma;\Delta \vdash A$ *then* $(\Gamma;A) \preceq_s (\Gamma;\Delta)$.

Proof. As in Theorem 4.14, we proceed by rule induction. Here we consider three new rules.

- (rule clone) Suppose $\Gamma, B; \Delta \vdash A$ is derived from $\Gamma, B; \Delta, B \vdash A$. By induction, we have

  (16) $$(\Gamma, B;\ A) \preceq_s (\Gamma, B;\ \Delta, B).$$

  From $(\Gamma, B; \Delta)$ we have the transition $(\Gamma, B; \Delta) \xrightarrow{\tau} (\Gamma, B;\ \Delta, B)$. By Lemma 5.8, we know that

  (17) $$(\Gamma, B;\ \Delta, B) \preceq_s (\Gamma, B; \Delta).$$

  Combining (16), (17), and the transitivity of similarity, we obtain $(\Gamma, B;\ A) \preceq_s (\Gamma, B;\ \Delta)$.

- (rule !L) Suppose $\Gamma; \Delta, !B \vdash A$ is derived from $\Gamma, B; \Delta \vdash A$. By induction, we have

  (18) $$(\Gamma, B;\ A) \preceq_s (\Gamma, B;\ \Delta).$$

  From $(\Gamma;\ \Delta, !B)$ we have the transition $(\Gamma;\ \Delta, !B) \xrightarrow{\tau} (\Gamma, B; \Delta)$. By Lemma 5.8, we know that

  (19) $$(\Gamma, B;\ \Delta) \preceq_s (\Gamma;\ \Delta, !B).$$

  By Lemma 5.7 we have

  (20) $$(\Gamma;\ A) \preceq_s (\Gamma, B;\ A).$$

  Combining (18) - (20), and the transitivity of similarity, we obtain $(\Gamma;\ A) \preceq_s (\Gamma;\ \Delta, !B)$.

- (rule !R) Suppose $\Gamma; \cdot \vdash !A$ is derived from $\Gamma; \cdot \vdash A$. By induction we have

  (21) $$(\Gamma; A) \preceq_s (\Gamma; \cdot).$$

  We now construct a relation $\mathcal{R}$ as follows.

  $$\begin{array}{rcl} \mathcal{R} & = & \{((\Gamma; \Delta, !A),\ (\Gamma; \Delta)) \mid \text{for any } \Delta\} \\ & & \cup\{((\Gamma, A; \Delta),\ (\Gamma'; \Delta')) \mid \text{for any } \Delta, \Delta' \text{ and } \Gamma' \text{with } (\Gamma; \Delta) \preceq_s (\Gamma'; \Delta')\} \\ & & \cup \preceq_s \end{array}$$

  We show that $\mathcal{R}$ is a simulation, thus $\mathcal{R} \subseteq \preceq_s$. Since $(\Gamma;\ !A)\ \mathcal{R}\ (\Gamma; \cdot)$, it follows that $(\Gamma;\ !A) \preceq_s (\Gamma; \cdot)$.

  To see that $\mathcal{R}$ is a simulation, we pick any pair of states from $\mathcal{R}$. It suffices to consider the elements from the first two subsets of $\mathcal{R}$:

  - The two states are $(\Gamma;\ \Delta, !A)$ and $(\Gamma; \Delta)$ respectively. Let us consider any transition from the first state.
    - The transition is $(\Gamma;\ \Delta, !A) \xrightarrow{\tau} (\Gamma, A;\ \Delta)$. This is matched up by the trivial transition $(\Gamma; \Delta) \stackrel{\tau}{\Longrightarrow} (\Gamma; \Delta)$ because $(\Gamma; \Delta) \preceq_s (\Gamma; \Delta)$ and thus we have $(\Gamma, A; \Delta)\ \mathcal{R}\ (\Gamma; \Delta)$.

* The transition is $(\Gamma; \Delta, !A) \xrightarrow{\alpha} (\Gamma'; \Delta', !A)$ because of $(\Gamma; \Delta) \xrightarrow{\alpha} (\Gamma'; \Delta')$. Then the latter transition can match up the former because $(\Gamma'; \Delta', !A)\ \mathcal{R}\ (\Gamma'; \Delta')$.
* The transition is $(\Gamma; \Delta, !A) \xrightarrow{?a} (\Gamma; \Delta', !A)$ because of $(\Gamma; \Delta) \xrightarrow{?a} (\Gamma; \Delta')$. Then we have $(\Gamma; \Delta, a) \xrightarrow{\tau} (\Gamma; \Delta')$, which is a matching transition because we have $(\Gamma; \Delta', !A)\ \mathcal{R}\ (\Gamma; \Delta')$.
* If $(\Gamma; \Delta, !A)$ can be split as $((\Gamma_1; \Delta_1), (\Gamma_2; \Delta_2))$, then $!A$ occurs in either Δ_1 or Δ_2. Without loss of generality, we assume that $!A$ occurs in Δ_1. That is, there is some Δ_1' such that $\Delta_1 \equiv \Delta_1', !A$. Then $(\Gamma; \Delta) \equiv ((\Gamma_1; \Delta_1'), (\Gamma_2; \Delta_2))$. It is easy to see that $(\Gamma_1; \Delta_1)\ \mathcal{R}\ (\Gamma_1; \Delta_1')$ and $(\Gamma_2; \Delta_2)\ \mathcal{R}\ (\Gamma_2; \Delta_2)$.

– The two states are $(\Gamma, A; \Delta)$ and $(\Gamma'; \Delta')$ respectively with

$$(\Gamma; \Delta) \preceq_s (\Gamma'; \Delta'). \tag{22}$$

Let us consider any transition from the first state.

* If $\Delta \equiv \cdot$, then $(\Gamma; \cdot) \preceq_s (\Gamma'; \Delta')$. So there exists some Γ'' such that $(\Gamma'; \Delta') \stackrel{\tau}{\Longrightarrow} (\Gamma''; \cdot)$ and $(\Gamma; \cdot) \preceq_s (\Gamma''; \cdot)$. It follows that $(\Gamma, A; \cdot)\ \mathcal{R}\ (\Gamma''; \cdot)$ as required.
* The transition is $(\Gamma, A; \Delta) \xrightarrow{\tau} (\Gamma, A; \Delta, A)$. We argue that it is matched up by the trivial transition $(\Gamma'; \Delta') \stackrel{\tau}{\Longrightarrow} (\Gamma'; \Delta')$. By (21) and the compositionality of $\preceq_s$, we obtain

$$(\Gamma; \Delta, A) \preceq_s (\Gamma; \Delta). \tag{23}$$

  By (22) and (23), together with the transitivity of similarity, it can be seen that $(\Gamma; \Delta, A) \preceq_s (\Gamma'; \Delta')$, which implies $(\Gamma, A; \Delta, A)\ \mathcal{R}\ (\Gamma'; \Delta')$.
* The transition is $(\Gamma, A; \Delta) \xrightarrow{\alpha} (\Gamma, A; \Delta'')$ because of $(\Gamma; \Delta) \xrightarrow{\alpha} (\Gamma; \Delta'')$. By (22) there exist some Γ''', Δ''' such that $(\Gamma'; \Delta') \stackrel{\alpha}{\Longrightarrow} (\Gamma'''; \Delta''')$ and $(\Gamma; \Delta'') \preceq_s (\Gamma'''; \Delta''')$. Therefore, $(\Gamma, A; \Delta'')\ \mathcal{R}\ (\Gamma'''; \Delta''')$ and we have found the matching transition from $(\Gamma'; \Delta')$.
* The transition is $(\Gamma, A; \Delta) \xrightarrow{?a} (\Gamma, A; \Delta'')$ because of $(\Gamma; \Delta) \xrightarrow{?a} (\Gamma; \Delta'')$. By (22) there exist some Γ''', Δ''' such that $(\Gamma'; \Delta', a) \stackrel{\alpha}{\Longrightarrow} (\Gamma'''; \Delta''')$ and $(\Gamma; \Delta'') \preceq_s (\Gamma'''; \Delta''')$. Therefore, $(\Gamma, A; \Delta'')\ \mathcal{R}\ (\Gamma'''; \Delta''')$ and we have found the matching transition from $(\Gamma'; \Delta', a)$.
* If $(\Gamma, A; \Delta)$ can be split as $((\Gamma_1; \Delta_1), (\Gamma_2; \Delta_2))$, then A occurs in either Γ_1 or Γ_2. Without loss of generality, we assume that A occurs in Γ_1. That is, there is some Γ_1' such that $\Gamma_1 \equiv \Gamma_1', A$. Then $(\Gamma; \Delta) \equiv ((\Gamma_1'; \Delta_1), (\Gamma_2; \Delta_2))$. By (22) we have the transition $(\Gamma'; \Delta') \stackrel{\tau}{\Longrightarrow} ((\Gamma_3; \Delta_3), (\Gamma_4; \Delta_4))$ for some $(\Gamma_3; \Delta_3)$ and $(\Gamma_4; \Delta_4)$ such that $(\Gamma_1'; \Delta_1) \preceq_s (\Gamma_3; \Delta_3)$ and $(\Gamma_2; \Delta_2) \preceq_s (\Gamma_4; \Delta_4)$. It follows that $(\Gamma_1; \Delta_1)\ \mathcal{R}\ (\Gamma_3; \Delta_3)$ and $(\Gamma_2; \Delta_2)\ \mathcal{R}\ (\Gamma_4; \Delta_4)$.

□

Corollary 5.10. *If* $\Gamma;\Delta \vdash A$*, then* $(\cdot;A) \preceq_s (\Gamma;\Delta)$.

Proof. By Lemma 5.7, Theorem 5.9, and the transitivity of $\preceq_s$. □

Proposition 5.11. *If* $(\Gamma_1;\Delta_1) \stackrel{\tau}{\Longrightarrow} (\Gamma_2;\Delta_2)$ *and* $\Gamma_2;\Delta_2 \vdash A$ *then* $\Gamma_1;\Delta_1 \vdash A$.

Proof. Similar to the proof of Proposition 4.15. We now have two more cases:

- (rule lts!A) Suppose $(\Gamma;\Delta,!A) \stackrel{\tau}{\longrightarrow} (\Gamma,A;\Delta)$ and $\Gamma,A;\Delta \vdash B$. By rule !L, we infer that $\Gamma;\Delta,!A \vdash B$.
- (rule ltsClone) Suppose $(\Gamma,A;\Delta) \stackrel{\tau}{\longrightarrow} (\Gamma,A;\Delta,A)$ and $\Gamma,A;\Delta,A \vdash B$. By rule clone, we infer that $\Gamma,A;\Delta \vdash B$. □

Our next goal is to prove Theorem 5.16, the coincidence of logical preorder with simulation. For that purpose, a series of intermediate results are in order.

Theorem 5.12. *If* $(\Gamma_1;A) \preceq_s (\Gamma_2;\Delta)$ *then* $\Gamma_2;\Delta \vdash A$.

Proof. As in Theorem 4.16, the proof is by induction on the structure of A. We now have one more case.

- $A \equiv !A'$. By rule lts!A we have the transition $(\Gamma_1;A) \stackrel{\tau}{\longrightarrow} (\Gamma_1,A';\cdot)$. Since $(\Gamma_1,A) \preceq_s (\Gamma_2;\Delta)$ there is some Γ_2' such that $(\Gamma_2;\Delta) \stackrel{\tau}{\Longrightarrow} (\Gamma_2';\cdot)$ and

  $$(\Gamma_1,A';\cdot) \preceq_s (\Gamma_2';\cdot). \tag{24}$$

  From $(\Gamma_1,A';\cdot)$ we have the transition $(\Gamma_1,A';\cdot) \stackrel{\tau}{\longrightarrow} (\Gamma_1,A';A')$ by rule ltsClone. By Lemma 5.8 we have

  $$(\Gamma_1,A';A') \preceq_s (\Gamma_1,A';\cdot) \tag{25}$$

  It follows from (24), (25), and the transitivity of similarity that

  $$(\Gamma_1,A';A') \preceq_s (\Gamma_2';\cdot). \tag{26}$$

  Now by induction hypothesis, we obtain $\Gamma_2';\cdot \vdash A'$ because A' has a smaller structure than A. By rule !R we infer that $\Gamma_2';\cdot \vdash A$. Using Proposition 5.11 we conclude that $\Gamma_2;\Delta \vdash A$. □

We now have the counterpart of Theorem 4.17.

Theorem 5.13. $(\Gamma;\Delta_1) \preceq_l (\Gamma;\Delta_2)$ *if and only if* $(\Gamma;\Delta_1) \preceq_s (\Gamma;\Delta_2)$.

Proof. ($\Rightarrow$) Suppose $(\Gamma;\Delta_1) \preceq_l (\Gamma;\Delta_2)$. It is trivial to see that $\Gamma;\Delta_1 \vdash \bigotimes\Delta_1$. By the definition of logical preorder, it follows that $\Gamma;\Delta_2 \vdash \bigotimes\Delta_1$. By Theorem 5.9 we have

$$(\Gamma;\bigotimes\Delta_1) \preceq_s (\Gamma;\Delta_2). \tag{27}$$

According to our reduction semantics, we have $(\Gamma;\bigotimes\Delta_1) \stackrel{\tau}{\Longrightarrow} (\Gamma;\Delta_1)$. By Lemma 5.8, it follows that

$$(\Gamma;\Delta_1) \preceq_s (\Gamma;\bigotimes\Delta_1). \tag{28}$$

By combining (27) and (28), we obtain that $(\Gamma;\Delta_1) \preceq_s (\Gamma;\Delta_2)$ because $\preceq_s$ is transitive.

($\Leftarrow$) Suppose that $(\Gamma;\Delta_1) \preceq_s (\Gamma;\Delta_2)$. For any $\Gamma';\ \Delta$ and A, assume that $(\Gamma',\Gamma;\Delta,\Delta_1) \vdash A$. By Theorem 5.9 we have

$$(\Gamma',\Gamma;A) \preceq_s (\Gamma',\Gamma;\Delta,\Delta_1). \tag{29}$$

Since $(\Gamma;\Delta_1) \preceq_s (\Gamma;\Delta_2)$ and $\preceq_s$ is compositional, we obtain

$$(\Gamma',\Gamma;\Delta,\Delta_1) \ \preceq_s \ (\Gamma',\Gamma;\Delta,\Delta_2). \tag{30}$$

By (29), (30) and the transitivity of $\preceq_s$, we see that $(\Gamma',\Gamma;A) \ \preceq_s \ (\Gamma',\Gamma;\Delta,\Delta_2)$. Then Theorem 5.12 yields $\Gamma',\Gamma;\Delta,\Delta_2 \vdash A$. Therefore, we have shown that $(\Gamma;\Delta_1) \preceq_l (\Gamma;\Delta_2)$. □

In Theorem 5.13 we compare two states with exactly the same unrestricted resource Γ. The theorem can be relaxed so that the two states can have different unrestricted resources. In order to prove that result, we first need two lemmas.

Lemma 5.14. $(\Gamma;\Delta) \preceq_l (\cdot;\ !\Gamma,\Delta)$ *and* $(\cdot;\ !\Gamma;\Delta) \preceq_l (\Gamma;\Delta)$.

Proof. For any Γ',Δ', it follows from rule lts!A that $(\Gamma';\Delta',!\Gamma,\Delta) \stackrel{\tau}{\Longrightarrow} (\Gamma',\Gamma;\Delta',\Delta)$. By Proposition 5.11, if $\Gamma',\Gamma;\Delta',\Delta \ \vdash \ A$ then $\Gamma';\Delta',!\Gamma,\Delta \ \vdash \ A$, for any formula A. In other words, $(\Gamma;\Delta) \preceq_l (\cdot;\ !\Gamma,\Delta)$.

Suppose $\Gamma';\Delta',!\Gamma,\Delta \ \vdash \ A$ for any Γ',Δ' and A. By rule induction on the derivation of $\Gamma';\Delta',!\Gamma,\Delta \vdash A$ it can be shown that $\Gamma',\Gamma;\Delta',\Delta \vdash A$, thus $(\cdot;\ !\Gamma,\Delta) \preceq_l (\Gamma;\Delta)$. □

Lemma 5.15. $(\Gamma;\Delta) \preceq_s (\cdot;!\Gamma,\Delta)$ *and* $(\cdot;!\Gamma,\Delta) \preceq_s (\Gamma;\Delta)$.

Proof. Since $(\cdot;\ !\Gamma,\Delta) \stackrel{\tau}{\Longrightarrow} (\Gamma;\Delta)$, we apply Lemma 5.8 and conclude that $(\Gamma,\Delta) \preceq_s (\cdot;\ !\Gamma,\Delta)$.

To show that $(\cdot;!\Gamma,\Delta) \preceq_s (\Gamma;\Delta)$, we let $\mathcal{R}$ be the relation that relates any state $(\Gamma;!A_1,...,!A_n,\Delta)$ with the state $(\Gamma,A_1,...,A_n;\Delta)$. The relation $\mathcal{R}$ is a simulation. Consider any transition from $(\Gamma;!A_1,...,!A_n,\Delta)$.

- If $(\Gamma;!A_1,...,!A_n,\Delta) \stackrel{\alpha}{\longrightarrow} (\Gamma';!A_1,...,!A_n,\Delta')$ because of $(\Gamma;\Delta) \stackrel{\alpha}{\longrightarrow} (\Gamma';\Delta')$, the transition can be matched up by $(\Gamma,A_1,...,A_n;\Delta) \stackrel{\alpha}{\longrightarrow} (\Gamma',A_1,...,A_n;\Delta')$.
- If $(\Gamma;!A_1,...,!A_n;\Delta) \stackrel{\tau}{\longrightarrow} (\Gamma,A_1;!A_2,...,!A_n,\Delta)$ then the transition can be matched up by the trivial transition $(\Gamma,A_1,...,A_n;\Delta) \stackrel{\tau}{\Longrightarrow} (\Gamma,A_1,...,A_n;\Delta)$.
- If $(\Gamma;!A_1,...,!A_n,\Delta)$ performs an input action, it must be given by an input action from Δ. Obviously, this can be mimicked by $(\Gamma,A_1,...,A_n;\Delta)$.
- It is easy to see that for any splitting of $(\Gamma;!A_1,...,!A_n,\Delta)$ there is a corresponding splitting of $(\Gamma,A_1,...,A_n;\Delta)$.

We have shown that $\mathcal{R}$ is a simulation. Therefore, $(\Gamma;!A_1,...,!A_n,\Delta) \preceq_s (\Gamma,A_1,...,A_n;\Delta)$, and as a special case $(\cdot;!\Gamma,\Delta) \preceq_s (\Gamma;\Delta)$. □

Theorem 5.16. $(\Gamma_1;\Delta_1) \preceq_l (\Gamma_2;\Delta_2)$ *if and only if* $(\Gamma_1;\Delta_1) \preceq_s (\Gamma_2;\Delta_2)$.

Proof. Suppose $(\Gamma_1; \Delta_1) \preceq_l (\Gamma_2; \Delta_2)$. By Lemma 5.14 we infer that

$$(\cdot; !\Gamma_1, \Delta_1) \preceq_l (\Gamma_1; \Delta_1) \preceq_l (\Gamma_2; \Delta_2) \preceq_l (\cdot; !\Gamma_2, \Delta_2).$$

Since $\preceq_l$ is a preorder, its transitivity gives $(\cdot; !\Gamma_1, \Delta_1) \preceq_l (\cdot; !\Gamma_2, \Delta_2)$. By Theorem 5.13, we have $(\cdot; !\Gamma_1, \Delta_1) \preceq_s (\cdot; !\Gamma_2, \Delta_2)$. Then by Lemma 5.15 we infer that

$$(\Gamma_1; \Delta_1) \preceq_s (\cdot; !\Gamma_1, \Delta_1) \preceq_s (\cdot; !\Gamma_2, \Delta_2) \preceq_s (\Gamma_2; \Delta_2).$$

By the transitivity of $\preceq_s$, we obtain that $(\Gamma_1; \Delta_1) \preceq_s (\Gamma_2; \Delta_2)$.

In a similar manner, we can show that $(\Gamma_1; \Delta_1) \preceq_s (\Gamma_2; \Delta_2)$ implies $(\Gamma_1; \Delta_1) \preceq_l (\Gamma_2; \Delta_2)$. □

With Theorem 5.16 we can slightly generalize Theorem 5.12.

Corollary 5.17. *If* $(\Gamma_1; A) \preceq_s (\Gamma_2; \Delta)$ *then* $\Gamma_2; \Delta \vdash A \otimes \bigotimes !\Gamma_1$.

Proof. Suppose $(\Gamma_1; A) \preceq_s (\Gamma_2; \Delta)$. By Theorem 5.16 this means that

$$(\Gamma_1; A) \preceq_l (\Gamma_2; \Delta) \tag{31}$$

By Theorem 2.8, we have that $\Gamma_1; A \vdash A \otimes \bigotimes !\Gamma_1$. Now, by applying the definition of logical equivalence, (31) yields $\Gamma_2; \Delta \vdash A \otimes \bigotimes !\Gamma_1$. □

From Theorems 5.6 and 5.16, we obtain the main result of this subsection.

Corollary 5.18. $(\Gamma_1; \Delta_1) \preceq_l (\Gamma_2; \Delta_2)$ *if and only if* $(\Gamma_1; \Delta_1) \preceq_c (\Gamma_2; \Delta_2)$.

6 Concluding remarks

In this report, we have shown that the proof-theoretic notion of logical preorder coincides with an extensional behavioral relation adapted from the process-theoretic notion of contextual preorder [7]. The former is defined exclusively in terms of traditional derivability, and the latter is defined in terms of a CCS-like process algebra inspired by the formula-as-process interpretation of a fragment of linear logic. In order to establish the connection, a key ingredient is to introduce a coinductively defined simulation as a stepping stone. It is interesting to see that coinduction, a central proof technique in process algebras, is playing an important role in this study of linear logic. This topic definitely deserves further investigation so that useful ideas developed in one field can benefit the other, and vice versa.

We have started expanding the results in this report by examining general implication (i.e., formulas of the form $A \multimap B$ rather than $a \multimap B$) and the usual quantifiers. While special cases are naturally interpreted into constructs found in the join calculus [9] and the π-calculus [21, 25], the resulting language appears to extend well beyond them. If successful, this effort may lead to more expressive process algebras. We are also interested in understanding better the interplay of the proof techniques used in the present work. This may develop into an approach to employ coinduction effectively in logical frameworks so as to facilitate formal reasoning and verification of concurrent systems.

References

[1] Samson Abramsky. Proofs as processes. *Theoretical Computer Science*, 135:5–9, 1994.

[2] Andrew Barber. Dual intuitionistic linear logic. Technical Report ECS-LFCS-96-347, Laboratory for Foundations of Computer Sciences, University if Edinburgh, 1996.

[3] Iliano Cervesato, Nancy Durgin, Max I. Kanovich, and Andre Scedrov. Interpreting Strands in Linear Logic. In H. Veith, N. Heintze, and E. Clark, editors, *2000 Workshop on Formal Methods and Computer Security — FMCS'00*, Chicago, IL, 2000.

[4] Iliano Cervesato, Frank Pfenning, David Walker, and Kevin Watkins. A concurrent logical framework II: Examples and applications. Technical Report CMU-CS-2002-002, Department of Computer Science, Carnegie Mellon University, March 2002. Revised May 2003.

[5] Iliano Cervesato and Andre Scedrov. Relating state-based and process-based concurrency through linear logic. *Information and Computation*, 207:1044–1077, 2009.

[6] Yuxin Deng and Wenjie Du. Logical, metric, and algorithmic characterisations of probabilistic bisimulation. Technical Report CMU-CS-11-110, Carnegie Mellon University, March 2011.

[7] Yuxin Deng and Matthew Hennessy. On the semantics of markov automata. In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP'11)*, volume 6756 of *Lecture Notes in Computer Science*, pages 307–318. Springer, 2011.

[8] Yuxin Deng, Rob van Glabbeek, Matthew Hennessy, Carroll Morgan, and Chenyi Zhang. Characterising testing preorders for finite probabilistic processes. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science*, pages 313–325. IEEE Computer Society, 2007.

[9] C. Fournet and G. Gonthier. The join calculus: a language for distributed mobile programming, 2000. Applied Semantics Summer School — APPSEM'00, Caminha. available from `http://research.microsoft.com/~fournet`.

[10] Cédric Fournet and Georges Gonthier. A hierarchy of equivalences for asynchronous calculi. *Journal of Logic and Algebraic Programming*, 63(1):131–173, 2005.

[11] Gerhard Gentzen. Untersuchungen über das logische schließen. i. *Mathematische Zeitschrift*, 39(1):176–210, 1935.

[12] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[13] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types*. Cambridge University Press, 1989.

[14] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

[15] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[16] Kohei Honda and Mario Tokoro. On asynchronous communication semantics. In P. Wegner M. Tokoro, O. Nierstrasz, editor, *Proceedings of the ECOOP '91 Workshop on Object-Based Concurrent Computing*, volume 612 of *LNCS 612*. Springer-Verlag, 1992.

[17] Patrick Lincoln and Vijay Saraswat. Proofs as concurrent processes: A logical interpretation for concurrent constraint programming. Technical report, Systems Sciences Laboratory, Xerox PARC, November 1991.

[18] Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. *Nordic Journal of Philosophical Logic*, 1(1):11–60, 1996.

[19] Raymond McDowell, Dale Miller, and Catuscia Palamidessi. Encoding transition systems in sequent calculus. *Theoretical Computer Science*, 294(3):411–437, 2003.

[20] Dale Miller. The π-calculus as a theory in linear logic: Preliminary results. In E. Lamma and P.Mello, editors, *Proceedings of the Workshop on Extensions to Logic Programming — ELP'92*, pages 242–265. Springer-Verlag LNCS 660, 1992.

[21] Robin Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[22] Frank Pfenning. Structural cut elimination I. intuitionistic and classical logic. *Information and Computation*, 157(1/2):84–141, 2000.

[23] Frank Pfenning and Rowan Davies. A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11(4):511–540, 2001. Notes to an invited talk at the *Workshop on Intuitionistic Modal Logics and Applications (IMLA'99)*, Trento, Italy, July 1999.

[24] Julian Rathke and Pawel Sobocinski. Deriving structural labelled transitions for mobile ambients. In *Proceedings of the 19th International Conference on Concurrency Theory*, volume 5201 of *Lecture Notes in Computer Science*, pages 462–476. Springer, 2008.

[25] D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.

[26] Alwen Tiu and Dale Miller. A proof search specification of the π-calculus. In *3rd Workshop on the Foundations of Global Ubiquitous Computing*, volume 138 of *Electronic Notes in Theoretical Computer Science*, pages 79–101, September 2004.