



An intensionally fully-abstract sheaf model for (expanded version)

Eberhart, Clovis; Hirschowitz, Tom; Seiller, Thomas

Published in:
Logical Methods in Computer Science

DOI:
[10.23638/LMCS-13\(4:9\)2017](https://doi.org/10.23638/LMCS-13(4:9)2017)

Publication date:
2017

Document version
Publisher's PDF, also known as Version of record

Document license:
[CC BY-ND](https://creativecommons.org/licenses/by-nd/4.0/)

Citation for published version (APA):
Eberhart, C., Hirschowitz, T., & Seiller, T. (2017). An intensionally fully-abstract sheaf model for (expanded version). *Logical Methods in Computer Science*, 13(4), [9]. [https://doi.org/10.23638/LMCS-13\(4:9\)2017](https://doi.org/10.23638/LMCS-13(4:9)2017)

AN INTENSIONALLY FULLY-ABSTRACT SHEAF MODEL FOR π

(EXPANDED VERSION)

CLOVIS EBERHART^a, TOM HIRSCHOWITZ^b, AND THOMAS SEILLER^c

^{a,b} Univ. Savoie Mont Blanc, CNRS, LAMA, F-73000 Chambéry, France

^c Department of Computer Science, University of Copenhagen, Copenhagen, Denmark and CNRS, UMR 7030, Laboratoire d'Informatique de Paris Nord Université Paris 13, Sorbonne Paris Cité, F-93430 Villetaneuse, France

ABSTRACT. Following previous work on CCS, we propose a compositional model for the π -calculus in which processes are interpreted as sheaves on certain simple sites. Such sheaves are a concurrent form of innocent strategies, in the sense of Hyland-Ong/Nickau game semantics. We define an analogue of fair testing equivalence in the model and show that our interpretation is intensionally fully abstract for it. That is, the interpretation preserves and reflects fair testing equivalence; and furthermore, any innocent strategy is fair testing equivalent to the interpretation of some process. The central part of our work is the construction of our sites, relying on a combinatorial presentation of π -calculus traces in the spirit of string diagrams.

CONTENTS

1. Introduction	2
1.1. Causal models and beyond	2
1.2. Traces and naive concurrent strategies	3
1.3. Innocence as a sheaf condition	4
1.4. Main result	5
1.5. Contributions	7
1.6. Related work	7
1.7. Plan	8
2. Notation and preliminaries	9
2.1. Basic notation and labelled transition systems	9
2.2. A π -calculus	10
2.3. Fair testing equivalence	11
2.4. Playgrounds	14

Key words and phrases: Programming languages; categorical semantics; presheaf semantics; game semantics; concurrency; process algebra.

The authors acknowledge support from French ANR projet blancs PiCoq ANR 2010 BLAN 0305 01 and Récré ANR-11-BS02-0010.

3.	A pseudo double category of traces	17
3.1.	String diagrams	18
3.2.	From string diagrams to actions	20
3.3.	From actions to traces	22
3.4.	The main double category	23
4.	Codomain is a fibration	25
4.1.	A factorisation system	25
4.2.	A first ‘fibred’ double category	26
4.3.	Restriction of seeds	27
4.4.	Opliftings	29
4.5.	Restriction of actions	32
4.6.	Restriction of traces	34
5.	A playground for π	37
5.1.	A candidate playground	38
5.2.	Correctness criterion	40
5.3.	A playground	45
6.	A sheaf model	47
6.1.	Strategies and behaviours	48
6.2.	Decomposing behaviours	50
6.3.	Interpretation of π	52
6.4.	Semantic fair testing	53
7.	Intensional full abstraction	54
7.1.	A first graph with testing for behaviours	54
7.2.	A further graph with testing for behaviours	60
7.3.	Intensional full abstraction	62
7.4.	Generalisation	64
8.	Conclusion and future work	65
	References	65
	Appendix A. Proof of Lemma 7.37	70

1. INTRODUCTION

1.1. **Causal models and beyond.** Operational semantics of programming languages standardly model the execution of programs as paths in a certain labelled transition system (LTS). Under this interpretation, different possible interleavings of parallel actions yield different paths. Verification on LTSS thus incurs a well-known state explosion problem. Similarly, causality between various actions, visible in the syntax, is lost in the LTS, thus making, e.g., error diagnostics difficult [32].

Causal models, originally designed for Petri nets [63] and Milner’s CCS [72], intend to remedy both problems, but have yet to be applied to full-scale programming languages. They have recently been extended in two different directions: (1) by Crafa et al. [18] to Milner’s π -calculus, and (2) by Melliès [57] to Girard’s linear logic. The former extension accounts for the subtle interaction of channel creation with synchronisation in π , a significant technical achievement, 30 years after the first causal semantics for CCS. The latter is the first causal

model for functional languages (inspired by Hyland-Ong’s and Nickau’s *games* models for PCF [62, 42]). An important challenge is now the search for a causal model of full-fledged languages with both concurrent and functional features. Winskel and collaborators are currently working in this direction, using extensions of Melliès’s approach [67, 73, 14].

In previous work [40, 38, 39], we have proposed a causal model for CCS based on a different approach. We here push this approach further by applying it to the π -calculus.

1.2. Traces and naive concurrent strategies. In standard causal models, execution traces essentially consist of partially ordered sets of atomic ‘events’. Our approach relies on a new notion of trace, which we now briefly sketch. There is first a (straightforward) notion of *position*, which is essentially a finite hypergraph whose nodes are thought of as *agents*, and whose hyperedges between nodes x_1, \dots, x_n are thought of as communication channels shared by x_1, \dots, x_n . There is then a notion of *atomic action* from one position to another. The collection of atomic actions is thought of as a ‘rule of the game’. Examples of atomic actions are: an agent creates a new, private communication channel; an agent forks into two new agents connected to the same channels; an agent sends some channel a over some channel b to some other agent. We finally have a notion of trace which allows several atomic actions to occur, in a way that only retains some minimal causality information between them. We here mean, e.g., information such as: ‘such agent outputs on such channel only after having created such other channel’.

The main purpose of our notion of trace is to interpret π -calculus processes as some kind of strategies over them. Most naively, a strategy on some position X is a prefix-closed set of ‘accepted’ traces from X . But what should prefix mean in our setting? Well, we may view traces with initial position X and final position Y as morphisms $Y \dashrightarrow X$. Sequential composition of traces, denoted by \bullet , yields an analogue of prefix ordering, defined by $t \leq t \bullet w$. Strategies as prefix-closed sets of traces however fail to suit our needs on three counts. First, such naive strategies may not be stable under isomorphism of traces; second, they are bound to model coarse behavioural equivalences, at least as coarse as may testing equivalence (a.k.a. trace equivalence); and third, they permit undesirable interaction between players. Let us examine these issues in more detail.

The first, easy one is that there is an obvious notion of isomorphism between traces, under which strategies should be closed. The second problem is more serious: until now, these too naive strategies are not concurrent enough to adequately model CCS or the π -calculus.

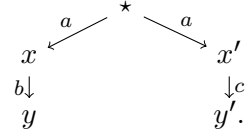
Example 1.1 (Milner’s coffee machines). Consider the CCS processes $P = (a.b + a.c)$ and $Q = a.(b + c)$. The process P has two ways of inputting on a and then, depending on the chosen way, inputs on either b or c . The process Q inputs on a and then has both possibilities of inputting on b or c . They hence exhibit significantly different interactive behaviour. Both processes, however, accept exactly the same traces (in the standard sense), namely $\{\epsilon, a, ab, ac\}$, where ϵ denotes the empty trace.

Thus, taking strategies to be prefix-closed sets of traces would prevent us from directly modelling any reasonably fine behavioural equivalence on processes. Inspired by *presheaf models* [45], we remedy both problems at once by passing from prefix-closed sets of traces to presheaves (of finite sets) on traces. Indeed, in the simple case where traces on X form a mere poset $\mathbb{T}(X)$ by prefix ordering, a prefix-closed set of traces is nothing but a contravariant functor from $\mathbb{T}(X)$ to the ordinal 2, viewed as a category. The latter has two objects 0 and

1 and just one non-trivial morphism $0 \rightarrow 1$. The idea is that a functor $S: \mathbb{T}(X)^{op} \rightarrow 2$ maps any trace to 1 when it is accepted, and to 0 otherwise. Furthermore, if $t \leq t'$, i.e., t is a prefix of t' , then we have a morphism $t \rightarrow t'$ which should be mapped by S to some morphism $S(t') \rightarrow S(t)$. If t' is accepted then $S(t') = 1$, so this has to be a morphism $1 \rightarrow S(t)$. Because there are no morphisms $1 \rightarrow 0$, this entails $S(t) = 1$, hence prefix-closedness of the corresponding strategy. Now in the case where traces form a proper category $\mathbb{T}(X)$, whose morphisms encompass both prefix ordering and isomorphism between traces, considering functors $\mathbb{T}(X)^{op} \rightarrow 2$ retains prefix-closedness and solves our first problem: for any $t \cong t'$, functoriality imposes $S(t) \cong S(t')$. Our second problem is then solved by replacing such functors with presheaves, i.e., functors $\mathbb{T}(X)^{op} \rightarrow \mathbf{Set}$.

Example 1.2. In Example 1.1, the two ways that P has to accept inputting on a may be reflected by mapping the trace a to some two-element set. More precisely, P may be modelled by the presheaf S defined on the left and pictured on the right:

- $S(\epsilon) = \{\star\}$,
- $S(a) = \{x, x'\}$,
- $S(ab) = \{y\}$,
- $S(ac) = \{y'\}$,
- S empty otherwise,
- $S(\epsilon \hookrightarrow a) = \{x \mapsto \star, x' \mapsto \star\}$,
- $S(a \hookrightarrow ab) = \{y \mapsto x\}$,
- $S(a \hookrightarrow ac) = \{y' \mapsto x'\}$,



Presheaves thus may ‘accept a trace in several ways’: the trace a is here accepted in two ways, x and x' . The process Q is of course modelled by equating x and x' .

As it turns out, we actually only need *finitely many* ways of accepting each trace. Thus, we arrive at a first sensible notion of strategy given by presheaves of finite sets, i.e., functors $\mathbb{T}(X)^{op} \rightarrow \mathbf{set}$, where \mathbf{set} denotes the category with as objects all finite subsets of \mathbb{N} , with all maps between them. We call them (*naive*) *strategies* in the sequel. (Please note that \mathbf{set} is equivalent to the category of all finite sets.)

Notation 1.3. For any \mathcal{C} , let $\widetilde{\mathcal{C}}$ denote the category of presheaves of finite sets over \mathcal{C} .

1.3. Innocence as a sheaf condition. The third problem evoked above is that functors $\mathbb{T}(X)^{op} \rightarrow \mathbf{set}$ allow some undesirable behaviours. Intuitively, in π just as in CCS, agents should not have any control over the routing of messages.

Example 1.4. Consider a position X with three agents x, y , and z sharing a communication channel a , and a strategy S accepting (1) the trace where x outputs on a , (2) the trace where y inputs on a , and (3) the trace where z inputs on a . Then, both synchronisations should be accepted by S . However, one easily constructs a naive strategy in which one is refused (see Example 6.2).

In order to rectify this deficiency, we enrich strategies with ‘local’ information. The idea is that a strategy should not only accept or refuse traces on the whole position X , but also traces on all possible subpositions of X . Moreover, this local information should fit together coherently.

Example 1.5. Consider the position X of Example 1.4. Any strategy on X should now in particular include independent strategies for each of the three agents x, y , and z . Coherence means that in order for a trace to be accepted, it should be enough for it to be ‘locally accepted’, i.e., at every stage in the trace, each agent should approve what they see of the

next action. E.g., if the next action is a synchronisation $x \rightarrow y$ with x outputting and y inputting on some channel a , then all that is required for the synchronisation to be accepted is that x accepts to output and y accepts to input. Consequently, if some other agent z also accepts to input on a at this stage, then the synchronisation $x \rightarrow z$ is also accepted.

We call this putative coherence condition *innocence* by analogy with Hyland and Ong’s notion [42]. In order to formalise it, we first extend our category of traces $\mathbb{T}(X)$ on X with new objects representing traces on subpositions of X . We also add new morphisms, which are about ‘locality’. Indeed, standardly, plays form a poset for the prefix ordering, but here we want to enrich this, e.g., by embedding traces on subpositions of X into traces on X .

Example 1.6. Consider a position X with two agents x_1 and x_2 . There is a trace t on X in which both agents fork. Consider now the subposition Y of X consisting solely of x_1 and the trace t' on Y in which x_1 merely forks. There is a morphism $t' \rightarrow t$ in our new category.

This extended category, \mathbb{T}_X , yields an intermediate notion of strategy, given by functors $\mathbb{T}_X^{op} \rightarrow \mathbf{set}$. Among the new objects, we have in particular traces on just one agent of X obtained by sequentially composing atomic actions whose final position again consists of one agent. We call this particular kind of trace a *view*. Views are the most ‘local’ kind of objects in \mathbb{T}_X . They form a full subcategory \mathbb{V}_X of \mathbb{T}_X .

Example 1.7. If X merely consists of an agent x linked to n communication channels, consider the atomic action given by x forking into two new agents, say x_1 and x_2 . This action, viewed as an object of \mathbb{T}_X has three subobjects which are views: (1) the ‘identity’ view, in which nothing happens, (2) π_n^l , which represents the left-hand branch (to x_1), (3) and π_n^r , which represents the right-hand branch (to x_2).

The inclusion $\mathbb{V}_X \hookrightarrow \mathbb{T}_X$ induces a simple Grothendieck topology [54] on \mathbb{T}_X , which amounts to decreeing that any trace is covered by its views. We finally call any $S: \mathbb{T}_X^{op} \rightarrow \mathbf{set}$ *innocent* precisely when it is a *sheaf* for this Grothendieck topology. In particular, giving an innocent presheaf on \mathbb{T}_X is equivalent (up to isomorphism) to separately giving an innocent presheaf for each agent of X , which rules out the undesirable behaviour described in Example 1.4.

Sheaves on \mathbb{T}_X form a category \mathbf{S}_X , which is small thanks to our use of \mathbf{set} instead of \mathbf{Set} . They furthermore map back to naive strategies, i.e., presheaves on $\mathbb{T}(X)$, by forgetting the local information. Finally, because the considered topology is particularly simple, sheaves are equivalent to presheaves on views, i.e., $\mathbf{S}_X \simeq \widetilde{\mathbb{V}}_X$ (recalling Notation 1.3). In summary, we have three categories of strategies: *naive* strategies are presheaves on the ‘global’ category of traces $\mathbb{T}(X)$, *innocent* strategies \mathbf{S}_X are sheaves on the extended category of traces \mathbb{T}_X , and so-called *behaviours* \mathbf{B}_X are presheaves on the category of views \mathbb{V}_X . The last two are equivalent, and we furthermore have an adjunction $\widetilde{\mathbb{T}(X)} \overset{\perp}{\rightleftarrows} \mathbf{S}_X$.

We use both sides of the equivalence: behaviours directly lead to our compositional interpretation $\llbracket - \rrbracket: Pi \rightarrow \mathbf{S}$ of π -calculus processes, and innocent strategies are used below as the basis for our semantic definition of fair testing equivalence.

1.4. Main result. What should we do in order to demonstrate adequacy of our model? By definition, causal models expose some intensional information. Hence, equality is generally much finer than any reasonable behavioural equivalence, so we should not base our main result on it. On the other hand, causal models are supposed to be ‘compositional’, i.e., to

come equipped with an interpretation of syntactic operations in the model. A natural thing to do is thus to choose some behavioural equivalence defined from syntactic operations, use compositionality to transpose it to the model, and prove that the two coincide. More precisely, the considered equivalence induces by quotienting two ‘extensional collapses’, one syntactic and the other semantic, and we want to prove that the translation induces a bijection between both extensional collapses. Following [3], we call this *intensional full abstraction* for the considered equivalence. In fact, all behavioural equivalences end up relying on some notion of observation, which we will also need to transpose to the model.

We here focus on so-called *testing equivalences* [20, 61, 9, 66], which are defined in two stages. First, one chooses a ‘mode of interaction’. That is, one defines what the relevant *tests* are for a given process and specifies how the two should interact. Typically, tests for P are other processes T with the same free communication channels as P , and interaction is just parallel composition $P | T$. This part will be easy to transpose to the model by compositionality. The second stage amounts to choosing when $P | T$ is *successful*. E.g., in may testing equivalence $P | T$ is successful just when there exists a transition $(P | T) \xRightarrow{\heartsuit} P'$ (that is, a \heartsuit transition, possibly surrounded by silent transitions), where \heartsuit is some ‘tick’ action fixed in advance. In must testing equivalence, success is when all maximal (possibly infinite) transition sequences contain at least one \heartsuit transition. In fair testing equivalence, one requires that all silent sequences $(P | T) \Rightarrow P'$ extend to some sequence $P' \Rightarrow P'' \xrightarrow{\heartsuit} P'''$ ending with a \heartsuit transition. These ideas transpose to the model by observing whether a given trace contains a \heartsuit action. In this paper, we focus on fair testing equivalence, i.e., we prove (Theorem 6.25) that our model is intensionally fully-abstract for fair testing equivalence. We finally show (Section 7.4) that the result generalises to a wide range of testing equivalences, obtained by varying the notion of success.

In order to fix intuitions, let us quickly motivate must and fair testing, using *barbed congruence* [69] as a standard starting point. Barbed congruence equates processes P and Q , roughly, when for all contexts C , $C[P]$ and $C[Q]$ are weakly bisimilar w.r.t. *reduction* (i.e., only τ -actions are allowed), and furthermore they have the same interaction capabilities at all stages. Barbed congruence is sometimes perceived as too discriminating w.r.t. guarded choice. Consider, e.g., the following CCS processes.

$$P_1 = \begin{array}{c} \bullet \xrightarrow{\tau} \bullet \xrightarrow{\tau} \bullet \xrightarrow{\tau} \bullet \\ \tau \searrow \quad a \searrow \quad b \searrow \\ \bullet \quad \bullet \quad \bullet \end{array} \quad P_2 = \begin{array}{c} \bullet \xrightarrow{\tau} \bullet \xrightarrow{\tau} \bullet \xrightarrow{\tau} \bullet \\ \tau \searrow \quad b \searrow \quad a \searrow \\ \bullet \quad \bullet \quad \bullet \end{array}$$

Both processes may disable both actions a and b , the only difference being that P_1 disables a *before* disabling b . Barbed congruence distinguishes P_1 from P_2 (consider the trivial context $C = \square$), which some view as a deficiency.

Another possibility would be *must testing* equivalence [20]. Recall that P *must pass* a test process R iff all maximal executions of $P | R$ perform, at some point, the ‘tick’ action \heartsuit . Then, P and Q are must testing equivalent iff they must pass the same tests. Must testing equivalence does equate P_1 and P_2 above, but is sometimes perceived as too discriminating w.r.t. divergence. E.g., consider $Q_1 = !\tau | a$ and $Q_2 = a$. Perhaps surprisingly, Q_1 and Q_2 are *not* must testing equivalent. Indeed, Q_2 must pass the test $\bar{a}.\heartsuit$, but Q_1 does not, due to an infinite, silent reduction sequence.

Fair testing equivalence was originally introduced (for CCS-like calculi) to rectify both the deficiency of barbed congruence w.r.t. choice and that of must testing equivalence w.r.t. divergence. The idea is that two processes are equivalent when they *should* pass the same

tests. A process P should pass the test T iff their parallel composition $P | T$ never loses the ability of performing the special ‘tick’ action \heartsuit , after any \heartsuit -free reduction sequence. Fair testing equivalence thus equates P_1 and P_2 above, as well as Q_1 and Q_2 . Cacciagrano et al. [13] provide an excellent survey of fair testing for π .

Example 1.8 ([13]). The π -calculus features a well-known encoding of internal choice using channel creation and parallel composition. Mixing this with replication leads to intriguing examples of fair testing. Consider the following subtly different encodings of $!(b \oplus c)$, where \oplus denotes internal choice and $!$ denotes replication: let $R_1 = !\nu a.(\bar{a} | a.b | a.c)$ and $R_2 = \nu a.!(\bar{a} | a.b | a.c)$. These are clearly fair testing equivalent. However, each encoding has an execution that always makes, say, the left choice, and Cacciagrano et al. argue that for R_1 this execution is fair, as the involved channel is different each time. They use similar examples to argue that fair testing is in fact too coarse, and instead propose alternative notions (which lie beyond the scope of this paper).

1.5. Contributions. Since this paper follows the same approach as previous work on CCS [40, 38, 39], we should explain in which sense extending the approach to π is more than an easy application.

A first contribution comes from the fact that, in order to even define composition in our category of traces, we need to show that traces form the total category of a *fibration* [43] over positions. In previous work, this was done in an *ad hoc* way. We here introduce a more satisfactory approach based on *factorisation systems* [52, 29].

A second significant contribution is prompted by the interplay between synchronisation and private channels in π , which is notoriously subtle to handle. And indeed, our proof method for CCS fails miserably on π . One reason for this, we think, is that our notion of trace for π , though simple and natural, is not ‘modular’ enough, in the sense that a trace contains strictly more information than the collection of all ‘local’ information accessible to agents (i.e., of all of its views, in the above sense). Thus, adapting our proof technique from CCS would have required us to define a much more complex but more modular notion of trace. Instead, we here take a somewhat rougher route.

Finally, our proof now applies not only to fair testing equivalence, but also to a whole class of testing equivalences.

1.6. Related work. Beyond the obviously closely related, already mentioned work of Winskel et al., we should mention other causal models for π [11, 59, 26, 16, 7, 21, 15, 10, 18, 12], as well as interleaving models [28, 27, 70, 15, 60, 36] and the early approach [44] based on Girard’s Geometry of Interaction. All of these models are based on some LTS for π . Instead, ours is rather based on *reduction rules*. The subtleties usually showing up in LTSS, related to mixing synchronisation and private channels, do resurface in our proof of intensional full abstraction, but not in the definition of our model. Indeed, it merely goes by describing the ‘rule of the game’ in π , and applying the general framework of *playgrounds* [39].

Another general framework relating operational and denotational descriptions of programs is Kleene coalgebra [6], which is mainly designed for automata theory. Playgrounds may be viewed as adapting ideas from Kleene coalgebra to the process-algebraic setting.

We should also mention Laird’s games model of (a fragment of) π [47], which accounts for *trace* (a.k.a. *may testing*) equivalence. Standard game models view strategies as *sets* of traces (with well-formedness conditions), so, as we have seen, lend themselves better to

modelling trace equivalence. In a non-deterministic, yet not concurrent setting, Harmer and McCusker [35] resort to an explicit action for divergence, which allows them to recover a finer behavioural equivalence. We feel that the presheaf-based approach is more general.

Furthermore, recent work by Tsukada and Ong [71] adapts and extends some ideas of [40, 38] to nondeterministic, simply-typed λ -calculus. In particular, they show that innocent strategies as sheaves are compatible with the *hiding* operation of standard game semantics. Eberhart and Hirschowitz further establish [22] a formal link between Tsukada and Ong’s notion of innocence and ours: they construct a model of nondeterministic, simply-typed λ -calculus in our style, and then a morphism of Grothendieck sites, which entails that both models are equivalent.

Let us moreover mention less closely related work: Girard’s *ludics* [31], Melliès’s reworking of game semantics [56, 57], the part of it rediscovered by Levy [51] with a different presentation, Melliès’s game semantics in string diagrams [58], Harmer et al.’s categorical combinatorics of innocence [35], McCusker et al.’s graphical foundation for schedules [55], and Winskel’s strategies as profunctors [73]. Finally, Hildebrandt’s work [37] also uses sheaves, though as a tool to correctly handle infinite behaviour, as opposed to their use here to force reactions of agents to depend only on their views.

1.7. Plan. In previous work [39], we have defined an algebraic notion called *playground*, which provides a sufficient framework for sheaf-based innocence to make sense. Namely, it organises positions, atomic actions, and traces into a *pseudo double category* [24, 25, 33, 34, 49, 30] with additional structure. Any playground \mathbb{D} automatically gives rise, among other things, to

- categories of innocent strategies S_X on each position X , organised into a pseudo double functor from \mathbb{D}^{op} to small categories;
- a simple, yet complete syntax for innocent strategies, together with an LTS $S_{\mathbb{D}}$ for them over an alphabet built from atomic actions.

After introducing some notation, the considered variant of π -calculus, fair testing equivalence (Section 2), and recalling the notion of playground, we construct a playground \mathbb{D} for the π -calculus in Sections 3 to 5. This is a lot of work, and not all aspects of playgrounds are used in defining the model and proving the main result. The reason we devote so much energy to it is that playgrounds provide a really helpful setting, in fact a calculus, for reasoning about positions, traces, views and the various notions of strategies. The underlying pseudo double category is constructed in Section 3. The main playground axiom, asserting that a certain functor is a Grothendieck fibration, is established in Section 4. Finally, the remaining axioms are proved in Section 5.

We then continue in Section 6 by applying results from [39] to define our sheaf model and semantic fair testing equivalence, as well as our translation $\llbracket - \rrbracket$ of π . We then state the main result (Theorem 6.25). In Section 7, after introducing the basic notion of definite residual, we reduce our main theorem to an analogous statement about an LTS S for strategies (derived from $S_{\mathbb{D}}$). The advantage of the latter statement is that it lies entirely in the realm of LTSS. We then define a further, more syntactic LTS \mathcal{M} which we prove equivalent to S , thus further reducing the main result to an analogous statement about \mathcal{M} . We finally prove the latter, which entails the main result.

2. NOTATION AND PRELIMINARIES

We start in this section with a few reminders. In Section 2.1, after fixing some basic notation, we recall LTSS. We use a slightly more general, graph-based notion than the standard, relation-based one. In Section 2.2, we introduce the considered π -calculus, which is mostly standard except that (1) we use a presentation in the style of Berry and Boudol's *chemical abstract machine* [5], and (2) we consider infinite terms, thus sparing us the need for recursion or replication constructs. We then go on and recall fair testing equivalence for π in Section 2.3. In fact, because we will also need to define fair testing for other LTSS, we introduce a general framework in which it makes sense, called *graphs with testing*. We further provide sufficient conditions for a relation between the vertices of two graphs with testing to preserve and reflect fair testing equivalence (Lemma 2.22 and Corollary 2.25). Finally, in Section 2.4, we recall and briefly explain the definition of playgrounds.

2.1. Basic notation and labelled transition systems. First of all, we adopt the notation of [39, Section 2], with the slight modification that **set** now denotes the category with finite subsets of \mathbb{N} as objects, and all maps as morphisms. (This category is equivalent to what we used in [39], but slightly easier to work with for our purposes.) For all $n \in \mathbb{N}$, we often abuse notation and let n denote the finite set $\{1, \dots, n\}$. We denote by $\widehat{\mathbb{C}}$ the category of presheaves on \mathbb{C} , and by $\widetilde{\mathbb{C}}$ the category of presheaves of finite sets, i.e., of contravariant functors to **set**. For any category \mathcal{C} , let \mathcal{C}_f denote the full subcategory of *finitely presentable* objects [4], or *finite* objects for short. In the only case where we'll use this, \mathcal{C} will be a presheaf category $[\mathbb{C}^{op}, \mathbf{set}]$ and furthermore due to the special form of \mathbb{C} , finite presentability of $F \in \widehat{\mathbb{C}}$ will be equivalent to the category of elements of F being finite, and further equivalent to the set of elements of F being finite, i.e., $\sum_{c \in \text{ob}(C)} F(c)$ is finite.

To recall some bare minimum: we often confuse objects C of a category \mathbb{C} with the corresponding representable presheaves $y_C \in \widehat{\mathbb{C}}$. **Gph** denotes the category of reflexive graphs, and all our graphs are reflexive so we often omit mentioning it. We think of morphisms $p: G \rightarrow A$ in **Gph** as LTSS over the *alphabet* A , except that for reasons specific to playgrounds our convention is that a transition from x to y is represented as an edge $x \leftarrow y$. Using graphs as alphabets generalises the standard approach based on sets of labels: indeed, in order to model any set of labels, take for A the graph with one vertex and one endo-edge for each label. The extra generality is useful, e.g., to add some typing information on labels. Finally, using graphs as alphabets provides us with standard tools for transporting LTSS across morphisms (by pullback, resp. post-composition).

For any graph G , G^* denotes the graph with the same vertices and all paths between them; on the other hand, $\text{fc}(G)$ denotes the free category on G , i.e., the category with the same vertices and identity-free paths between them. Both $(-)^*$ and fc extend to functors, i.e., act on morphisms. We often silently coerce $\text{fc}(G)$ into a reflexive graph, and denote by $\widetilde{}$ the obvious morphism $G^* \rightarrow \text{fc}(G)$.

For any graph $p: G \rightarrow A$ over A , $x, y \in \text{ob}(G)$, and edge $a: p(y) \rightarrow p(x)$ in A , we denote by $x \xleftarrow{a} y$ the existence of an edge $e: y \rightarrow x$ in G such that $p(e) = a$. When $a = \text{id}$, we just write $x \leftarrow y$. We denote strong bisimilarity over A by \sim_A .

For any graph $p: G \rightarrow A$ over A , $x, y \in \text{ob}(G)$, and path $\rho: p(y) \rightarrow p(x)$ in A^* , we denote by $x \xleftarrow{\rho} y$ the existence of a path $r: y \rightarrow x$ in G^* such that $\widetilde{p^*(r)} = \rho$. When ρ is the empty path we just write $x \leftarrow y$. We denote weak bisimilarity over A by \approx_A .

2.2. A π -calculus. We now present our variant of π , which features a chemical abstract machine presentation and infinite terms. Also, we keep track of the channels known to the considered process, i.e., we work with a ‘natural deduction’ presentation of terms.

Processes are infinite terms coinductively generated by the grammar

$$\frac{\gamma \vdash_{\mathbb{g}} P_1 \quad \dots \quad \gamma \vdash_{\mathbb{g}} P_n}{\gamma \vdash \sum_{i \in \mathbb{N}} P_i} \quad \frac{\gamma \vdash P \quad \gamma \vdash Q}{\gamma \vdash P | Q}$$

$$\frac{\gamma, a \vdash P}{\gamma \vdash_{\mathbb{g}} \nu a.P} \quad \frac{\gamma \vdash P}{\gamma \vdash_{\mathbb{g}} \heartsuit.P} \quad \frac{\gamma \vdash P}{\gamma \vdash_{\mathbb{g}} \tau.P} \quad \frac{a \in \gamma \quad \gamma, b \vdash P}{\gamma \vdash_{\mathbb{g}} a(b).P} \quad \frac{a, b \in \gamma \quad \gamma \vdash P}{\gamma \vdash_{\mathbb{g}} \bar{a}(b).P},$$

where

- we have two judgements, \vdash for *processes* and $\vdash_{\mathbb{g}}$ for *guarded processes*;
- γ ranges over finite sets of natural numbers, and
- γ, a is defined iff $a \notin \gamma$ and then denotes $\gamma \uplus \{a\}$.

Notation 2.1. Let P_i be the set of all such (non-guarded) processes. Let P_{i_γ} denote the set of processes $\gamma \vdash P$.

As usual, a is bound in $\nu a.P$ and b is bound in $a(b).P$. In the following, processes are considered equivalent up to renaming of bound channels. Capture-avoiding substitution extends the assignment $\gamma \mapsto P_{i_\gamma}$ to a functor $\text{set} \rightarrow \text{Set}$ mapping $\sigma: \gamma \rightarrow \gamma'$ to $P \mapsto P[\sigma]$.

Let us now describe the dynamics of our π -calculus. They are slightly unusual, in that they are presented in the style of the chemical abstract machine. In particular, there are silent transitions for ‘heating’ both parallel composition and name creation. A further slight peculiarity, which we adopt for its convenience in the chemical abstract machine presentation, is that name creation is a guard. E.g., we have some processes of the form $(\nu a.P) + b(x).Q$. This is hardly significant. E.g., the previous process is strongly bisimilar to $(\tau.\nu a.P) + b(x).Q$ in more standard settings, and our results are about equivalences coarser than weak bisimilarity anyway.

Notation 2.2. For any $\gamma \vdash_{\mathbb{g}} P$, $\gamma \vdash Q$ of the form $\sum_{i \in n} Q_i$, and injection $h: n \hookrightarrow n+1$, we denote by $P +_h Q$ the sum $\sum_{j \in n+1} P_j$, where $P_{h(i)} = Q_i$ for all $i \in n$ and $P_k = P$, for k the unique element of $(n+1) \setminus \text{Im}(h)$.

Definition 2.3. Let $-^\circ$ denote the finite multiset monad on sets.

Definition 2.4. A *configuration* is an element of $\text{Conf} = \sum_{\gamma \in \mathcal{P}_f(\mathbb{N})} P_{i_\gamma}^\circ$.

Notation 2.5. Configurations (γ, S) will be denoted by $\langle \gamma \| S \rangle$, and we will use list syntax $[P_1, \dots, P_n]$ for multisets, sometimes dropping brackets, e.g., as in $\langle \gamma \| P_1, \dots, P_n \rangle$. We sometimes resort to a hopefully clear ‘multiset comprehension’ notation $[P \mid \varphi(P)]$. We use \cup for multiset union and $x :: M = [x] \cup M$.

Just as P_i , Conf extends to a functor $\text{set} \rightarrow \text{Set}$ by capture-avoiding substitution.

Let us now extend Conf to an LTS over the alphabet $\{\heartsuit, \tau\}$. This means that we need to construct a graph morphism $\text{Conf} \rightarrow \Sigma$, where Σ denotes

$$\heartsuit \left(\begin{array}{c} \bigcirc \\ \bigcirc \end{array} \right) \tau,$$

τ being the chosen identity edge.

This is done in Figure 1, omitting identity edges. There, we let R and R' range over processes of the form $\sum_{i \in n} P_i$. The last rule makes sense because each transition as in the premise implicitly comes with an inclusion $h: \gamma_1 \hookrightarrow \gamma_2$, and the second occurrence of S is implicitly $S[h]$.

$$\begin{array}{c}
 \frac{}{\langle \gamma \| P \mid Q \rangle \stackrel{\tau}{\leftarrow} \langle \gamma \| P, Q \rangle} \quad \frac{}{\langle \gamma \| \tau.P +_h R \rangle \stackrel{\tau}{\leftarrow} \langle \gamma \| P \rangle} \quad \frac{}{\langle \gamma \| \heartsuit.P +_h R \rangle \stackrel{\heartsuit}{\leftarrow} \langle \gamma \| P \rangle} \\
 \\
 \frac{}{\langle \gamma \| \nu a.P +_h R \rangle \stackrel{\tau}{\leftarrow} \langle \gamma, a \| P \rangle} \quad \frac{}{\langle \gamma \| a(b).P +_h R, \bar{a}(c).Q +_{h'} R' \rangle \stackrel{\tau}{\leftarrow} \langle \gamma \| P[b \mapsto c], Q \rangle} \\
 \\
 \frac{\langle \gamma_1 \| S_1 \rangle \stackrel{\alpha}{\leftarrow} \langle \gamma_2 \| S_2 \rangle}{\langle \gamma_1 \| S \cup S_1 \rangle \stackrel{\alpha}{\leftarrow} \langle \gamma_2 \| S \cup S_2 \rangle} \quad (\alpha \in \{\tau, \heartsuit\})
 \end{array}$$

Figure 1: Reduction rules for Conf

2.3. Fair testing equivalence. Let us now define fair testing equivalence for π , together with our general framework of graphs with testing. Graphs with testing are essentially De Nicola and Hennessy's original framework [20], adapted to our graph-based presentation of LTSS. We derive a few results about general graphs with testing, notably sufficient conditions for a relation between two graphs with testing to preserve and reflect fair testing equivalence.

Remark 2.6. In [39], an abstract framework was defined for studying fair testing equivalence and its relationship with weak bisimilarity. We won't use this here, and instead work in a simpler setting.

We first cover π -calculus, and then generalise. The starting point is that we need to be able to test processes against other processes, and more generally configurations against configurations. Because configurations carry their sets of free channels, it makes sense to consider a partial parallel composition operation:

Definition 2.7. For any $\langle \gamma \| S \rangle, \langle \gamma' \| S' \rangle \in \text{Conf}$, let $\langle \gamma \| S \rangle @ \langle \gamma' \| S' \rangle$ denote $\langle \gamma \| S \cup S' \rangle$ if $\gamma = \gamma'$ and be undefined otherwise. Let furthermore $\epsilon_\gamma = \langle \gamma \| \rangle$.

Lemma 2.8. *The domain of @, i.e., the set of pairs (C, C') such that $C @ C'$, is an equivalence relation.*

Let us denote by \succ_{Conf} this equivalence relation.

Here is the standard definition of fair testing equivalence:

Definition 2.9. Let \perp^{Conf} denote the set of configurations C such that for all $C \leftarrow C'$, there exists $C' \stackrel{\heartsuit}{\leftarrow} C''$.

Any two configurations C and C' are *fair testing equivalent* iff $C \succ_{\text{Conf}} C'$ and for all $D \succ_{\text{Conf}} C$, $(C @ D) \in \perp^{\text{Conf}}$ iff $(C' @ D) \in \perp^{\text{Conf}}$.

Let us now abstract away from this definition. For this, it would make sense to start from a partial, parallel composition map. However, in the model, the corresponding map will involve a pushout of positions which is only determined up to isomorphism. We thus

generalise from partial maps to relations, but we need to impose some conditions. What matters is the use we will make of parallel composition for testing. Intuitively, we will check that the parallel composition $C@C'$ belongs to some given *pole*, which is closed under weak bisimilarity over Σ . This choice is perhaps slightly arbitrary, but it encompasses all known testing equivalences. Finally, to gain just a little more generality, we will use the fact that weak bisimilarity over Σ coincides with strong bisimilarity over $\text{fc}(\Sigma)$, and work with the latter (Definition 7.40). For LTSS over $\text{fc}(\Sigma)$ obtained by applying fc to ones over Σ , this is equivalent to the previous setting. The extra generality is useful for describing LTSS which are not free categories. E.g., in Section 7.1, we introduce the graph with testing \mathcal{C} , whose transitions are traces, which compose in a non-free way. We thus start from the following notion, and then define fair testing equivalence.

Definition 2.10. A *graph with testing* is a graph G together with a morphism $p: G \rightarrow \text{fc}(\Sigma)$ and a relation $R: (\text{ob}(G))^2 \dashrightarrow \text{ob}(G)$ whose domain is an equivalence relation and which is partially functional up to strong bisimilarity over $\text{fc}(\Sigma)$.

The domain being an equivalence relation more precisely means that the set $\{(x, y) \mid \exists z. (x, y) R z\}$ forms an equivalence relation over $\text{ob}(G)$.

Partial functionality up to strong bisimilarity means that if $(x, y) R z$ and $(x, y) R z'$, then $z \sim_{\text{fc}(\Sigma)} z'$.

Notation 2.11. The relation is called the *testing* relation, and we denote it by $|_G$, i.e., $(x, y) R z$ is denoted by $z \in (x |_G y)$. Furthermore, its domain is denoted by \circlearrowright_G . We use $|$ and \circlearrowright when there is no ambiguity. Since $|$ is partially functional up to strong bisimilarity, for any $(x, y) R z$, as long as what we say about z is invariant under strong bisimilarity, then it also holds for any other z' such that $(x, y) R z'$. In such cases, we implicitly make some global choice of z and consider $|$ as partially functional.

Example 2.12. Figure 1 defines a morphism $\text{p}^{\text{Conf}}: \text{Conf} \rightarrow \Sigma$. Because, $@$ is a partial map, it induces a partially functional relation $\text{ob}(\text{Conf})^2 \dashrightarrow \text{ob}(\text{Conf})$, whose domain is an equivalence relation. Because partially functional implies partially functional up to strong bisimilarity, we have:

Proposition 2.13. *The morphism $\text{fc}(\text{p}^{\text{Conf}}): \text{fc}(\text{Conf}) \rightarrow \text{fc}(\Sigma)$, with $@$ as testing relation, forms a graph with testing.*

We may now mimick the standard definition of fair testing equivalence in the abstract setting:

Definition 2.14. For any graph with testing $p: G \rightarrow \text{fc}(\Sigma)$, let \perp^G denote the set of objects x such that for all $x \leftarrow y$, there exists $y \overset{\heartsuit}{\leftarrow} z$.

Any two objects x and y are *fair testing equivalent* iff $x \circlearrowright_G y$ and for all $z \circlearrowright_G x$, $(x |_G z) \in \perp^G$ iff $(y |_G z) \in \perp^G$.

Remark 2.15. Because we are working over $\text{fc}(\Sigma)$, if G has the shape $\text{fc}(G')$, then single transitions like $x \leftarrow y$ denote arbitrary paths of silent transitions. And analogously $x \overset{\heartsuit}{\leftarrow} y$ denotes any path with all edges silent except exactly one.

Notation 2.16. We denote fair testing equivalence in G by \sim_f^G . Given x , any z such that $x \circlearrowright z$ is called a *test* for x , and x *passes* the test iff $(x |_G z) \in \perp^G$.

Example 2.17. Definition 2.14 instantiates to Definition 2.9.

In fact, the construction of the graph with testing $\text{fc}(\mathbf{p}^{\text{Conf}})$ from \mathbf{p}^{Conf} is easily generalised:

Lemma 2.18. *For any morphism $p^G: G \rightarrow \Sigma$ and relation $R: (\text{ob}(G))^2 \dashrightarrow \text{ob}(G)$ whose domain is an equivalence relation, R equips $\text{fc}(p^G)$ with testing structure iff it is partially functional up to weak bisimilarity over Σ .*

Proof. R is a strong bisimulation for $\text{fc}(G)$ iff it is a weak bisimulation for G . \square

Definition 2.19. A graph with testing is *free* iff it is of the form $\text{fc}(p^G)$.

Our $\text{fc}(\mathbf{p}^{\text{Conf}})$ is thus a free graph with testing.

Let us conclude this section with a sufficient condition for a relation between the vertices of two graphs with testing to be adequate for fair testing equivalence, and a natural specialisation to free graphs with testing.

Definition 2.20. A relation $R: \text{ob}(G) \dashrightarrow \text{ob}(H)$ between the vertex sets of two graphs with testing $p^G: G \rightarrow \text{fc}(\Sigma)$ and $p^H: H \rightarrow \text{fc}(\Sigma)$ is *fair* iff

- $x R y$ and $x' R y'$ implies $(x \circ_G x') \Leftrightarrow (y \circ_H y')$;
- R is total and surjective, i.e.,
 - for all $x \in G$, there exists $y \in H$ such that $x R y$, and
 - for all $y \in H$, there exists $x \in G$ such that $x R y$;
- $x R y$ implies $x \sim_{\text{fc}(\Sigma)} y$;
- if $x R y$, $x' R y'$, and $x \circ_G x'$, then there exist $u \in (x \mid_G x')$ and $v \in (y \mid_H y')$ such that $u R v$.

Lemma 2.21. *A relation $R: \text{ob}(G) \dashrightarrow \text{ob}(H)$ between the vertex sets of two graphs with testing $p^G: G \rightarrow \text{fc}(\Sigma)$ and $p^H: H \rightarrow \text{fc}(\Sigma)$ is fair iff its converse R^\dagger is, where R^\dagger is defined by $(y R^\dagger x) \Leftrightarrow (x R y)$.*

Proof. Easy. \square

Lemma 2.22. *For any fair relation $R: \text{ob}(G) \dashrightarrow \text{ob}(H)$, if $x R y$ and $x' R y'$, then $(x \sim_f^G x') \Leftrightarrow (y \sim_f^H y')$.*

For proving this lemma, we need:

Lemma 2.23. *For all graphs with testing $p^G: G \rightarrow \text{fc}(\Sigma)$ and $p^H: H \rightarrow \text{fc}(\Sigma)$, $x \in G$ and $y \in H$, if $x \sim_{\text{fc}(\Sigma)} y$, then $(x \in \perp^G) \Leftrightarrow (y \in \perp^H)$.*

Proof. Assume $x \in \perp^G$ and consider any transition $y \leftarrow y'$ in H . By bisimilarity, $x \leftarrow x' \sim_{\text{fc}(\Sigma)} y'$. By hypothesis, we find $x' \overset{\heartsuit}{\leftarrow} x''$, so by bisimilarity, $y' \overset{\heartsuit}{\leftarrow} y''$. Thus, $y \in \perp^H$, which entails the result by symmetry. \square

Proof of Lemma 2.22. Consider any such x, y, x' , and y' . By Lemma 2.21, it suffices to show one direction of the desired equivalence. So let us assume that $x \sim_f^G x'$. Then $x \circ_G x'$, hence also $y \circ_H y'$ by fairness of R . By symmetry, it again suffices to check one direction of the desired implication. Consider thus any $t \circ_H y$ such that $(y \mid t) \in \perp^H$. By surjectivity of R , we find $s \in G$ such that $s R t$. By fairness again, we find $u \in (x \mid s)$ and $v \in (y \mid t)$ such that $u R v$, so $(x \mid s) \sim_{\text{fc}(\Sigma)} u \sim_{\text{fc}(\Sigma)} v \sim_{\text{fc}(\Sigma)} (y \mid t)$, and hence by the previous lemma $(x \mid s) \in \perp^G$. Because $x \sim_f^G x'$, this entails $(x' \mid s) \in \perp^G$, hence by a similar argument $(y' \mid t) \in \perp^H$, as desired. \square

Definition 2.24. A relation $R: \text{ob}(G) \dashrightarrow \text{ob}(H)$ between the vertex sets of two free graphs with testing respectively generated by $p^G: G \rightarrow \Sigma$ and $p^H: H \rightarrow \Sigma$ is *weakly fair* iff it satisfies the conditions of Definition 2.20, except for the third one, which is replaced by: $x R y$ implies $x \approx_\Sigma y$.

Corollary 2.25. For any weakly fair relation $R: \text{ob}(G) \dashrightarrow \text{ob}(H)$, if $x R y$ and $x' R y'$, then $(x \sim_f^{\text{fc}(G)} x') \Leftrightarrow (y \sim_f^{\text{fc}(H)} y')$.

Proof. Because weak bisimilarity over Σ is the same as strong bisimilarity over $\text{fc}(\Sigma)$, being weakly fair is the same as being fair for the generated graphs with testing. \square

2.4. Playgrounds. To conclude this preliminary section, let us recall the axioms for playgrounds [39]. Some constructions and results are developed from these axioms in op. cit. Some of the main ideas are reviewed and reworked in Sections 6 and 7.

Let us start with a brief recap of pseudo double categories. A pseudo double category \mathbb{D} consists of a set $\text{ob}(\mathbb{D})$ of *objects*, shared by a ‘horizontal’ category \mathbb{D}_h and a ‘vertical’ bicategory \mathbb{D}_v . Since we won’t consider strict double categories, we’ll often omit the word ‘pseudo’. Following Paré [64], \mathbb{D}_h , being a mere category, has standard notation (normal arrows, \circ for composition, *id* for identities), while the bicategory \mathbb{D}_v earns fancier notation (\dashrightarrow for arrows, \bullet for composition, id^\bullet for identities). \mathbb{D} is furthermore equipped with a set of *double cells* α , which have vertical, resp. horizontal, domain and codomain, denoted by $\text{dom}_v(\alpha)$, $\text{cod}_v(\alpha)$, $\text{dom}_h(\alpha)$, and $\text{cod}_h(\alpha)$. The horizontal domain and codomain of a double cell are vertical morphisms, while the vertical domain and codomain are horizontal morphisms. E.g., for α in the diagram below, we have $u = \text{dom}_h(\alpha)$, $u' = \text{cod}_h(\alpha)$, $h = \text{dom}_v(\alpha)$, and $h' = \text{cod}_v(\alpha)$. \mathbb{D} is furthermore equipped with operations for composing double cells: \circ composes them along a common vertical morphism, \bullet composes along horizontal morphisms. Both vertical compositions (of morphisms and double cells) may only be associative up to coherent isomorphism. The full axiomatisation is given by Garner [30], and we here only mention the *interchange law*, which says that the two ways of parsing the above diagram coincide: $(\beta' \circ \beta) \bullet (\alpha' \circ \alpha) = (\beta' \bullet \alpha') \circ (\beta \bullet \alpha)$.

$$\begin{array}{ccccc}
 X & \xrightarrow{h} & X' & \xrightarrow{k} & X'' \\
 \downarrow u & \nearrow \alpha & \downarrow u' & \nearrow \alpha' & \downarrow u'' \\
 Y & \xrightarrow{h'} & Y' & \xrightarrow{k'} & Y'' \\
 \downarrow v & \nearrow \beta & \downarrow v' & \nearrow \beta' & \downarrow v'' \\
 Z & \xrightarrow{h''} & Z' & \xrightarrow{k''} & Z''
 \end{array}$$

For any (pseudo) double category \mathbb{D} , we denote by \mathbb{D}_H the category with vertical morphisms as objects and double cells as morphisms, and by \mathbb{D}_V the bicategory with horizontal morphisms as objects and double cells as morphisms. Domain and codomain maps form functors $\text{dom}_v, \text{cod}_v: \mathbb{D}_H \rightarrow \mathbb{D}_h$ and pseudofunctors $\text{dom}_h, \text{cod}_h: \mathbb{D}_V \rightarrow \mathbb{D}_v$. We will refer to dom_v and cod_v simply as *dom* and *cod*, reserving subscripts for dom_h and cod_h .

- (P4) (Fibration, continued) Restrictions of actions (resp. full actions) to individuals either are actions (resp. full actions), or have length 0.
- (P5) (Views) For any action $M: Y \twoheadrightarrow X$, and $y: d \rightarrow Y$ with $d \in \mathbb{I}$, there exists a cell

$$\begin{array}{ccc} d & \xrightarrow{y} & Y \\ v^{y,M} \downarrow & \overset{\alpha^{y,M}}{=} \Rightarrow & \downarrow M \\ d^{y,M} & \dashrightarrow_{y^M} & X, \end{array}$$

where $v^{y,M}$ either is a basic action or has length 0, which is unique up to canonical isomorphism, i.e., for any $y': d' \rightarrow X$, $v': d \twoheadrightarrow d'$, and $\alpha': v' \rightarrow M$ with $\text{dom}(\alpha') = y$ and $\text{cod}(\alpha') = y'$, we have $y' = y^M$ and there exists a unique isomorphism $\beta: v \rightarrow v'$ making the diagram

$$\begin{array}{ccccc} & & d & \xrightarrow{y} & Y \\ & & \uparrow \alpha^{y,M} & & \downarrow M \\ v^{y,M} & \bullet & & & \\ & \searrow \beta & v' & \xrightarrow{y^M} & X \\ d^{y,M} & \bullet & & & \\ & \swarrow & & & \\ & & d' & \xrightarrow{y'} & \end{array}$$

commute.

- (P6) (Left decomposition) Any double cell

$$\begin{array}{ccc} \begin{array}{ccc} A & \xrightarrow{h} & X \\ \downarrow u & \nearrow \alpha & \downarrow w_1 \\ B & \xrightarrow{k} & Z \end{array} & \text{decomposes as} & \begin{array}{ccc} A & \xrightarrow{h} & X \\ \downarrow u_1 & \nearrow \alpha_1 & \downarrow w_1 \\ C & \xrightarrow{l} & Y \\ \downarrow u_2 & \nearrow \alpha_2 & \downarrow w_2 \\ B & \xrightarrow{k} & Z \end{array} \end{array}$$

with α_3 an isomorphism, in an essentially unique way.

- (P7) (Right decomposition) Any double cell as in the center below, where b is a basic action and M is an action, decomposes in exactly one of the forms on the left and right:

$$\begin{array}{ccc} \begin{array}{ccc} A & \xrightarrow{\quad} & X \\ \downarrow & \nearrow \alpha_1 & \downarrow \\ B & \xrightarrow{\quad} & Y \\ \downarrow & \nearrow \alpha_2 & \downarrow \\ C & \xrightarrow{\quad} & Z \end{array} & \rightsquigarrow & \begin{array}{ccc} A & \xrightarrow{h} & X \\ \downarrow w & \nearrow \alpha & \downarrow u \\ B & & Y \\ \downarrow b & & \downarrow M \\ C & \xrightarrow{k} & Z \end{array} & \rightsquigarrow & \begin{array}{ccc} A & \xrightarrow{\quad} & X \\ \downarrow & \nearrow \alpha_1 & \downarrow \\ B & & Y \\ \downarrow & \nearrow \alpha_2 & \downarrow \\ C & \xrightarrow{\quad} & Z \end{array} \end{array}$$

- (P8) (Finiteness) For any object X , the comma category \mathbb{I}/X (taken in \mathbb{D}_h) is finite.
- (P9) (Basic vs. full) For all $d \in \mathbb{I}$ and actions $M: X \twoheadrightarrow d$, $M': X' \twoheadrightarrow d$, and $b: d' \twoheadrightarrow d$ with M and M' full and b basic, if there exist cells $M \leftarrow b \rightarrow M'$ then $M \cong M'$.

Intuitively, the objects of \mathbb{D} are configurations, or positions, in the game. The considered games are multi-party, so it makes sense to consider embeddings of positions: this is intended to be described by the horizontal category \mathbb{D}_h . The vertical category \mathbb{D}_v is that of traces,

or plays: morphisms $u: Y \dashrightarrow X$ model plays from the initial position X , to the final one, Y . Finally, cells model embeddings of plays, preserving initial and final position. E.g., this could model embedding the part of a play involving a particular agent.

Individuals are intended to model agents in a position, with a role similar to that of representable presheaves among general ones. Typically, the hom-set $\mathbb{D}_h(d, X)$, with $d \in \mathbb{I}$, models the set of agent slots of type d in the position X . For example, in the playground we construct below for π , individuals bear the number of channels that an agent is connected to, so that a morphism $[n] \rightarrow X$ amounts to an n -ary agent in X , i.e., an agent connected to n channels. Accordingly, the object $[n]$ models a position with just one n -ary agent.

Actions model moves in the game, and atomicity (P3) notably says that any play decomposes into moves. The distinction between *basic* and *full* actions has to do with innocence. The two notions are more or less dual: basic actions are as thin as possible, while full ones are as wide as possible. Intuitively, an action is full when it cannot embed into a larger one (unless possibly some agents are added), while it is basic when it cannot embed any smaller one. As we will see, in our playground for π , we have an action for forking, which describes how an agent x may fork into two, say x_1 and x_2 . This action is full, and it has two basic sub-actions, which respectively model the passage from x to x_1 and to x_2 . For another example, we also have an action for inputting on some given channel: it is obviously basic, but in fact also full, because the only way to embed it into a wider action is to add an agent and do a synchronisation. In view of this, it should be natural that basic actions have individuals as their domain and codomain (P2). As alluded to in (P5), views will be defined as composites of basic actions. Axiom (P5) intuitively enforces existence of one sub-action for each agent in the final position of any action. Extending this to general plays will yield an operation analogous to taking the view of a play in standard game semantics. Axiom (P9) requires that these basic sub-actions may not be shared among different full actions.

Both (P6) and (P7) are decomposition axioms. The former says that a decomposition of a play reflects essentially uniquely onto any subplay. The latter essentially says that basic actions are strictly sequential: if any play of the form $b \bullet w$ with b basic embeds into some other play, then the image of w should occur after that of b . This is expressed in a slightly convoluted way by saying that if the latter play decomposes as $M \bullet u$, then

- either b maps to M , in which case w should map to u ,
- or b maps to U , in which case w should also map to u .

This should make most of the axioms rather intuitive: the others are technical, which means that they emerged from our attempts to make things work out, but that we are not yet able to explain them satisfactorily.

3. A PSEUDO DOUBLE CATEGORY OF TRACES

In this section, we introduce our notion of trace, which is based on certain combinatorial objects, close in spirit to string diagrams. We first define these string diagrams, and then use them to define traces. Positions are special, hypergraph-like string diagrams whose vertices represent agents and whose hyperedges represent channels. A perhaps surprising point is that actions are not just a binary relation between positions, because we not only want to say *when* there is an action from one position to another, but also *how* this action is performed. This will be implemented by viewing actions from X to Y as *cospan*s $Y \rightarrow M \leftarrow X$ in a certain category $\tilde{\mathcal{C}}_f$, whose objects we call higher-dimensional string diagrams for lack of a better term. The idea is that X and Y respectively are the initial and final positions, and

that M describes how one goes from X to Y . By combining such actions (by pushout), we get a bicategory \mathbb{D}_v of positions and traces. Finally, we recast \mathbb{D}_v as the vertical bicategory of a pseudo double category \mathbb{D} .

3.1. String diagrams. The category $\widetilde{\mathbb{C}}_f$ will be a category of finite presheaves over a base category, \mathbb{C} . Let us motivate the definition of \mathbb{C} by recalling that (directed, multi) graphs may be seen as presheaves over the category with two objects \star and $[1]$, and two non-identity morphisms $s, t: \star \rightarrow [1]$. Any such presheaf G represents the graph with vertices in $G(\star)$ and edges in $G[1]$, the source and target of any $e \in G[1]$ being respectively $G(s)(e)$ and $G(t)(e)$, or $e \cdot s$ and $e \cdot t$ for short. A way to visualise how such presheaves represent graphs is to compute their *categories of elements* [54]. Recall that the category of elements $\text{el}G$ for a presheaf G over \mathbb{C} has as objects pairs (c, x) with $c \in \mathbb{C}$ and $x \in G(c)$, and as morphisms $(c, x) \rightarrow (d, y)$ all morphisms $f: c \rightarrow d$ in \mathbb{C} such that $y \cdot f = x$. This category admits a canonical functor π_G to \mathbb{C} , and G is the colimit of the composite $\text{el}G \xrightarrow{\pi_G} \mathbb{C} \xrightarrow{y} \widehat{\mathbb{C}}$ with the Yoneda embedding. E.g., the category of elements for $y_{[1]}$ is the poset $(\star, s) \xrightarrow{s} ([1], id_{[1]}) \xleftarrow{t} (\star, t)$, which could be pictured as $\bullet \longrightarrow \blacktriangle \longleftarrow \bullet$, where dots represent vertices, the triangle represents the edge, and links materialise the graph of $G(s)$ and $G(t)$, the convention being that t connects to the apex of the triangle. We thus recover some graphical intuition.

Let us give the formal definition of \mathbb{C} for reference. We advise to skip it on a first reading, as we then attempt to provide some intuition.

Definition 3.1. Let $G_{\mathbb{C}}$ be the graph with, for all n, m , with $a, b \in n$ and $c, d \in m$:

- vertices $\star, [n], \pi_n^l, \pi_n^r, \pi_n, \nu_n, \heartsuit_n, \tau_n, \iota_{n,a}, o_{n,a,b}$, and $\tau_{n,a,m,c,d}$;
- edges $s_1, \dots, s_n: \star \rightarrow [n]$, plus, $\forall v \in \{\pi_n^l, \pi_n^r, \heartsuit_n, \tau_n, o_{n,a,b}\}$, edges $s, t: [n] \rightarrow v$;
- edges $[n] \xrightarrow{t} \nu_n \xleftarrow{s} [n+1]$ and $[n] \xrightarrow{t} \iota_{n,a} \xleftarrow{s} [n+1]$;
- edges $\pi_n^l \xrightarrow{l} \pi_n \xleftarrow{r} \pi_n^r$ and $\iota_{n,a} \xrightarrow{\rho} \tau_{n,a,m,c,d} \xleftarrow{\epsilon} o_{m,c,d}$.

Let \mathbb{C} be the free category on $G_{\mathbb{C}}$, modulo the equations

$$s \circ s_i = t \circ s_i \quad l \circ t = r \circ t \quad \rho \circ t \circ s_a = \epsilon \circ t \circ s_c \quad \rho \circ s \circ s_{n+1} = \epsilon \circ s \circ s_d.$$

The first equation should be understood in $\mathbb{C}(\star, v)$ for all $n \in \mathbb{N}$, $i \in n$, and $v \in \cup_{a,b \in n} \{\pi_n^l, \pi_n^r, \heartsuit_n, \tau_n, \iota_{n,a}, o_{n,a,b}, \nu_n\}$. (This is rather elliptic: if v has the shape $\iota_{n,a}$ or ν_n , $s \circ s_i$ is really $\star \xrightarrow{s_i} [n+1] \xrightarrow{s} v$.) The second equation should be understood in $\mathbb{C}([n], \pi_n)$ for all n , and the last two in $\mathbb{C}(\star, \tau_{n,a,m,c,d})$, for all $n, m, a \in n$, and $c, d \in m$.

Our category of string diagrams is the category of finite presheaves $\widetilde{\mathbb{C}}_f$. A presheaf X over \mathbb{C} is a kind of higher-dimensional graph whose components are typed by objects of \mathbb{C} :

- $X(\star)$ is the set of vertices, or *channels*;
- $X[n]$ is the set of *agents* connected to n channels (which are given by $X(s_i)$);
- $X(\iota_{n,a})$ is the set of *input actions* by some n -ary agent on its a th channel;
- $X(o_{m,c,d})$ is the set of *output actions* by some m -ary agent on its c th channel of its d th channel;
- $X(\tau_{n,a,m,c,d})$ is the set of *synchronisations* between some input and some output on a common channel;
- $X(\pi_n)$ is the set of *forking actions* by some n -ary agent;
- and similarly for $X(\pi_n^l)$, $X(\pi_n^r)$, $X(\nu_n)$, $X(\heartsuit_n)$, and $X(\tau_n)$.

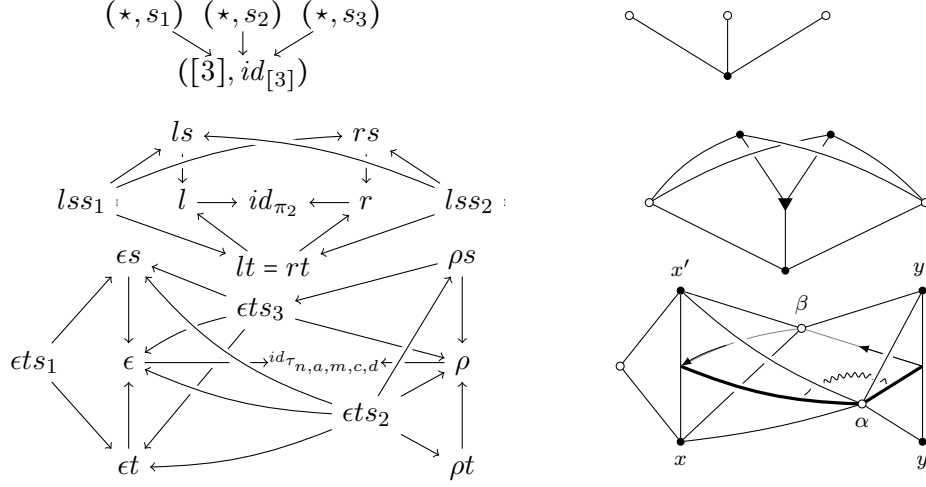


Figure 2: Categories of elements for $[3]$, π_2 , and $\tau_{1,1,3,2,3}$, with graphical representation

To see these intuitions at work, let us compute a few categories of elements. Let us start with an easy one, that of $[3] \in \mathbb{C}$ (recalling that we implicitly identify any $c \in \mathbb{C}$ with y_c). An easy computation shows that it is the poset pictured in the top left part of Figure 2. We think of it as a position with one agent ($[3], id_{[3]}$) connected to three channels, and draw it as in the top right part, where the bullet represents the agent, and circles represent channels.

Definition 3.2. *Positions* are finite presheaves empty except perhaps on \star and $[n]$ s.

Let us organise positions into a category, by designing a notion of morphism. We may equip the objects of \mathbb{C} with a *dimension*: \star has dimension 0, any $[n]$ has dimension 1, all of $\tau_n, \pi_n^l, \pi_n^r, \heartsuit_n, \iota_{n,a}, \circ_{n,c,d}, \nu_n$ have dimension 2, π_n has dimension 3, $\tau_{n,i,m,j,k}$ has dimension 4.

Definition 3.3. We accordingly define the *dimension* of a presheaf X on \mathbb{C} to be the lowest $n \in \mathbb{N}$ such that for any $m \in \mathbb{C}$ of dimension strictly greater than n , $X(m) = \emptyset$.

A position is thus equivalently a finite presheaf in $[\mathbb{C}^{op}, \text{set}]$ of dimension at most 1. An *interface* is one of dimension 0.

Definition 3.4. A map in $\widehat{\mathbb{C}}$ is *1-injective* iff it is injective in all strictly positive dimensions.

A *morphism of positions* is a 1-injective morphism in $\widehat{\mathbb{C}}$. The intuition for a morphism $X \rightarrow Y$ between positions is thus that X embeds into Y , possibly identifying some channels.

Definition 3.5. Positions and morphisms between them form a category \mathbb{D}_h .

Returning to our explanation of \mathbb{C} through categories of elements, let us consider that of π_2 . It is the poset generated by the left-hand graph in the second row of Figure 2 (omitting base objects for conciseness). We think of it as a binary agent (lt) forking into two agents (ls and rs), and draw it as on the right. The equation $lt = rt$ ensures that π_2^l and π_2^r are performed by the same agent. The graphical convention is that a black triangle stands for the presence of id_{π_2} , l , and r . Below, we represent just l as a white triangle with only a left-hand branch, and symmetrically for r . Furthermore, in all our pictures, time flows ‘upwards’.

Another category of elements, characteristic of the π -calculus, is the one for synchronisation $\tau_{n,a,m,c,d}$. The case $(n, a, m, c, d) = (1, 1, 3, 2, 3)$ is the poset generated by the graph at

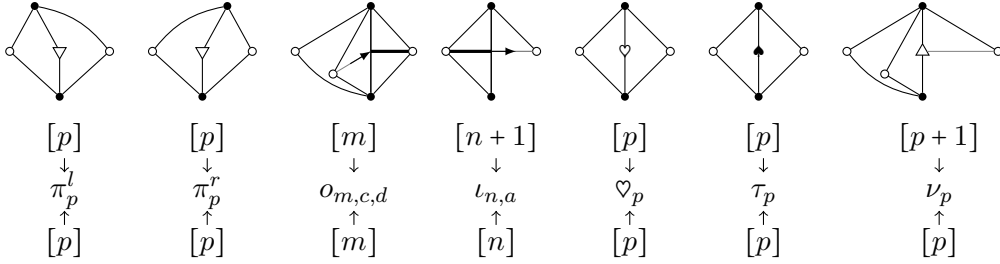


Figure 3: Pictures and corresponding cospans for π_p^l , π_p^r , $o_{m,c,d}$, $l_{n,a}$, \heartsuit_p , τ_p , and ν_p

the bottom left of Figure 2, which we will draw as on the right. The left-hand ternary agent x outputs its 3rd channel, here β , on its 2nd channel, here α . The right-hand unary agent y receives the sent channel on its unique channel, here α . Both agents have two occurrences, one before and one after the action, respectively marked as x/x' and y/y' . Both x and x' are ternary here, while y is unary and y' , having gained knowledge of β , is binary. There are actually three actions here, in the sense that there are three higher-dimensional elements. The first is the output action ϵ from x to x' , graphically represented as the middle point of \rightarrow (intended to evoke the point where β enters channel α). The second is the input action ρ from y to y' , graphically represented as the middle point of \leftarrow (where β exits channel α). The third action is the synchronisation itself, which ‘glues’ the other two together, as represented by the squiggly line.

We leave the computation of other categories of elements as an exercise to the reader. The remaining string diagrams are depicted in the top row of Figure 3, for $p = 2$ and $(n, a, m, c, d) = (1, 1, 3, 2, 3)$. The first two are *views*, in the game semantical sense, of the fork action π_2 explained above. The next two, $o_{m,c,d}$ (for ‘output’) and $l_{n,a}$ (for ‘input’), respectively are views for the sender and receiver in a synchronisation action. The τ_p action is a silent, dummy action standard in π -calculus. The \heartsuit_p action is the special ‘tick’ action used for defining fair testing equivalence. The last one, ν_p , is a channel creation action.

3.2. From string diagrams to actions. In the previous section, we have defined our category of string diagrams as $\tilde{\mathbb{C}}_f$, and provided some intuition on its objects. The next step is to construct a bicategory whose objects are positions, and whose morphisms represent traces. We start in this section by defining in which sense higher-dimensional objects of \mathbb{C} represent actions, and continue in the next one by explaining how to compose actions to form traces. Actions are defined in two stages: *seeds*, first, give their local form, their global form being given by embedding into bigger positions.

To start with, until now, our string diagrams contain no information about the ‘flow of time’, although we mentioned it informally in the previous section. To add this information, for each string diagram M representing an action, we define its initial and final positions, say X and Y , and view the whole action as a cospan $Y \xrightarrow{s} M \xleftarrow{t} X$. We have taken care, in drawing our pictures before, of placing initial positions at the bottom, and final positions at the top. So, e.g., the initial and final positions for the example synchronisation of Figure 2 are as follows.



They map into (the representable presheaf over) $\tau_{1,1,3,2,3}$, yielding the cospan

$$Y \xrightarrow{s} \tau_{1,1,3,2,3} \xleftarrow{t} X.$$

We leave it to the reader to define, based on the above pictures, the expected cospans for forking and synchronisation

$$\begin{array}{ccc} [p] \mid [p] & & [m]_{c,d} \mid_{a,n+1} [n+1] \\ \downarrow & & \downarrow \\ \pi_p & & \tau_{n,a,m,c,d} \\ \uparrow & & \uparrow \\ [p] & & [m]_c \mid_a [n] \end{array}$$

plus the remaining ones specified in the bottom row of Figure 3. Initial positions are at the bottom, and we use:

Notation 3.6. We denote by $[m]_{a_1, \dots, a_p} \mid_{c_1, \dots, c_p} [n]$ the position consisting of an m -ary agent x and an n -ary agent y , quotiented by the equations $x \cdot s_{a_k} = y \cdot s_{c_k}$ for all $k \in p$. When both lists are empty, by convention, $m = n$ and the agents share all channels in order.

Definition 3.7. These cospans are called *seeds*.

We now define actions from seeds by embedding the latter into bigger positions. E.g., we allow a fork action to occur in a position with more than one agent.

Definition 3.8. The *interface* I_F of a presheaf $F \in \widehat{\mathbb{C}}$ is $F(\star) \cdot \star$, the $F(\star)$ -fold coproduct of \star with itself, or in other words the position consisting solely of F 's channels. The interface of a seed $Y \xrightarrow{s} M \xleftarrow{t} X$ is I_X .

Since channels occurring in the initial position remain in the final one, we have for each seed a cone from I_X to the seed. For any morphism of positions $I_X \rightarrow Z$, pushing the cone along $I_X \rightarrow Z$ using the universal property of pushout as in

$$\begin{array}{ccccc} & & Y & \longrightarrow & Y' \\ & \nearrow & \downarrow & & \downarrow \\ & & M & \longrightarrow & M' \\ I_X & \longrightarrow & \uparrow & \longrightarrow & \uparrow \\ & \searrow & X & \longrightarrow & X' \end{array} \quad (3.1)$$

yields a new cospan, say $Y' \rightarrow M' \leftarrow X'$.

Definition 3.9. Let *actions* be all such pushouts of seeds.

Intuitively, taking pushouts glues string diagrams together. Let us do a few examples.

Example 3.10. The seed $[2] \mid [2] \xrightarrow{[ls,rs]} \pi_2 \xleftarrow{lt} [2]$ has as interface the presheaf $I_{[2]} = \star + \star$, consisting of two channels, say a and b . Consider the position $[2] + \star$ consisting of an agent y connected to two channels b' and c , plus an additional channel a' . Further consider the map $h: I_{[2]} \rightarrow [2] + \star$ defined by $a \mapsto a'$ and $b \mapsto b'$. The pushout

$$\begin{array}{ccc} I_{[2]} & \longrightarrow & [2] + \star \\ \downarrow & & \downarrow \\ \pi_2 & \longrightarrow & M' \end{array} \quad \text{is} \quad \begin{array}{c} \begin{array}{ccccccc} & x_1 & & x_2 & & & \\ & \bullet & & \bullet & & & \\ & \nearrow & & \nearrow & & & \\ & & \downarrow & & & & \\ & & \bullet & & & & \\ & \searrow & & \searrow & & & \\ a=a' & & & & b=b' & & y & & c \end{array} \end{array}$$



Figure 4: Example traces

The meaning of such an action is that x forks while y is passive.

Example 3.11. Because we push along *initial* channels, the interface of a seed may not contain all involved channels. E.g., in an input action (not part of any synchronisation), the received channel cannot be part of the initial position.

3.3. From actions to traces. Having defined actions, we now define their composition to yield our bicategory \mathbb{D}_v of positions and traces. Consider $\text{Cospan}(\tilde{\mathbb{C}}_f)$, the bicategory which has as objects all finite presheaves on \mathbb{C} , as morphisms $X \rightarrow Y$ all cospans $X \rightarrow U \leftarrow Y$, and obvious 2-cells. Composition is given by pushout, and hence is not strictly associative.

Notation 3.12. By convention, the initial position is the *target* of the morphism in $\text{Cospan}(\tilde{\mathbb{C}}_f)$. We denote morphisms in $\text{Cospan}(\tilde{\mathbb{C}}_f)$ with special arrows $Y \dashrightarrow X$; composition and identities are denoted with \bullet and id^\bullet , which matches the notation of pseudo double categories (Section 2.4).

Definition 3.13. A *trace* is any cospan in $\tilde{\mathbb{C}}_f$ which is isomorphic to some finite, possibly empty composite of actions in $\text{Cospan}(\tilde{\mathbb{C}}_f)$. Let \mathbb{D}_v denote the subbicategory of $\text{Cospan}(\tilde{\mathbb{C}}_f)$ obtained by restricting to positions, traces, and 1-injective 2-cells.

Thus, arrows $X \rightarrow Y$ in \mathbb{D}_h denote embeddings of X into Y (up to identification of channels), whereas arrows $Y \dashrightarrow X$ in \mathbb{D}_v denote traces with X initial and Y final. Intuitively, composition in \mathbb{D}_v glues string diagrams on top of each other, which yields a truly concurrent notion of trace: the only information retained in a trace about the order of occurrence of actions is their causal dependencies.

Example 3.14. Composing the action of Example 3.10 with a forking action by y yields the first string diagram of Figure 4, which shows that the ordering between remote actions is irrelevant. To illustrate how composition retains causal dependencies between actions, consider the second string diagram. It is unfolded for readability: one should identify both framed nodes, resp. both circled ones. In the initial position, there are channels a, b , and c , and three agents $x(a, b)$, $y(b)$, and $z(a, c)$ (channels known to each agent are in parentheses). In a first action, x sends a on b , and y receives it. In a second action, z sends c on a , and the avatar y' of y receives it. The second action is enabled by the first, by which y gains knowledge of a .

Before going on to construct the base double category for our playground, let us observe the following two basic facts about traces.

Lemma 3.15. For any trace $Y \xrightarrow{s} U \xleftarrow{t} X$, s and t are monos.

Proof. We proceed by induction on the number of actions involved in any decomposition of U . The base case is trivial. For the induction step, because composition of cospans is by pushout, the result follows from the induction hypothesis, stability of monos under pushout and composition, and the fact that the result holds for actions. The latter in turn follows from the fact that monos are stable under pushout (again!), the pushout lemma, and the fact that the result holds for seeds, which holds by case inspection. \square

Lemma 3.16. *For any trace $Y \xrightarrow{s} U \xleftarrow{t} X$, s_* is an isomorphism.*

Proof. Similar to the previous proof. \square

The intuition behind the last lemma is that no channel is forgotten during the play.

3.4. The main double category. At last, we define the base double category \mathbb{D} of our playground for the π -calculus. It is a sub-double category of a double category of cospans in $\widehat{\mathbb{C}}$.

Consider the double category \mathbb{D}^0 with

- positions as objects,
- horizontal morphisms $X \rightarrow Y$ given by all natural transformations $h: X \rightarrow Y$,
- vertical morphisms $X \dashrightarrow Y$ given by cospans $X \xrightarrow{s} U \xleftarrow{t} Y$ in $\widehat{\mathbb{C}}$,
- and double cells $U \rightarrow V$ given by commuting diagrams

$$\begin{array}{ccc}
 X' & \xrightarrow{k} & Y' \\
 s_U \downarrow & & \downarrow s_V \\
 U & \xrightarrow{l} & V \\
 t_U \uparrow & & \uparrow t_V \\
 X & \xrightarrow{h} & Y.
 \end{array} \tag{3.2}$$

Definition 3.17. Let \mathbb{D} denote the sub-double category of \mathbb{D}^0 obtained by restricting

- vertical morphisms to traces,
- horizontal morphisms to 1-injective maps,
- double cells to diagrams (3.2) in which k, l , and h are 1-injective.

Proposition 3.18. \mathbb{D} indeed forms a sub-double category of \mathbb{D}^0 , i.e., is closed under all composition operations.

Proof. The only non-obvious point is that double cells in \mathbb{D} are closed under vertical composition, and in particular that the middle component of the composite is 1-injective. This follows from Lemma 3.20 and Corollary 3.22 below. \square

Definition 3.19. Let \mathcal{V}_0 denote the set of ‘ t ’-legs (i.e., lower legs) of seeds.

Lemma 3.20. *For any morphism (3.2) in \mathbb{D}_H^0 , if U and V are traces, then the upper square is a pullback.*

Proof. For any trace $Y \xrightarrow{s} P \xleftarrow{t} X$ and $n \in \mathbb{N}$, $Y[n]$ consists of all elements of $P[n]$ which are not in the image of (the action of) any map in \mathcal{V}_0 .

Now, consider any double cell as in (3.2). Because s_V is monic, $U \times_V Y'$ may be chosen to be just $l^{-1}(Y') \subseteq U$. By Lemma 3.15 and standard cancellation properties of monos,

the mediating arrow $X' \rightarrow U \times_V Y'$ is mono. To show that it is epi, we proceed pointwise. Over \star , the result follows from s_U and s_V being isomorphisms (Lemma 3.16). Over $[n]$, if $x \in (U \times_V Y')[n]$ then $x \in U[n]$, and $l(x) \in V[n]$ is not in the image of (the action of) any $t \in \mathcal{V}_0$. But if there existed y such that $y \cdot t = x$, then by naturality we would have $l(y) \cdot t = l(x)$, contradicting the latter. Any natural transformation being both epi and mono is an isomorphism, hence the result. \square

Lemma 3.21. *In Set, consider any cube*

$$\begin{array}{ccccc}
 I & \xrightarrow{\quad} & B & & \\
 \downarrow & \lrcorner & \downarrow & \searrow & \\
 & & A & \xrightarrow{\quad} & C \\
 \downarrow & & \downarrow & & \downarrow f \\
 I' & \xrightarrow{\quad} & B' & & \\
 \downarrow & \lrcorner & \downarrow & \searrow & \\
 & & A' & \xrightarrow{\quad} & C'
 \end{array}$$

with the marked pushouts and pullback, and with all arrows mono except perhaps f . Then, f is also mono and the front square is also a pullback.

Proof. Any such cube is naturally isomorphic to some cube of the shape

$$\begin{array}{ccccc}
 I & \xrightarrow{\quad} & I + R & & \\
 \downarrow \text{inj}_1 & \lrcorner & \downarrow & \searrow & \\
 & & I + Y & \xrightarrow{\quad} & I + Y + R \\
 \downarrow & & \downarrow I+h & & \downarrow \text{inj}_1+k \\
 I + X & \xrightarrow{\quad} & I + X + S & & \\
 \downarrow \text{inj}_1 & \lrcorner & \downarrow & \searrow & \\
 & & I + X + Z & \xrightarrow{\quad} & I + X + Z + S \\
 & & & & \downarrow f=I+h+k
 \end{array}$$

the only non-trivial point being that the map $I + R \rightarrow I + X + S$ has the given shape. But this is because we know that its pullback along $I + X \rightarrow I + X + S$ is inj_1 , so the image of R has to lie in S . \square

Corollary 3.22. *Consider any cube*

$$\begin{array}{ccccc}
 X & \xrightarrow{\quad} & B & & \\
 \downarrow & \lrcorner & \downarrow & \searrow & \\
 & & A & \xrightarrow{\quad} & C \\
 \downarrow & & \downarrow & & \downarrow f \\
 X' & \xrightarrow{\quad} & B' & & \\
 \downarrow & \lrcorner & \downarrow & \searrow & \\
 & & A' & \xrightarrow{\quad} & C'
 \end{array}$$

in $\widehat{\mathbb{C}}$ in which all arrows except perhaps f are 1-injective, and the marked squares are pushouts, resp. pullbacks. Then f is also 1-injective.

Proof. We proceed pointwise. On any object C of dimension > 0 , we obtain a diagram in sets for which the lemma applies. \square

4. CODOMAIN IS A FIBRATION

In this section, we prove that the double category \mathbb{D} of traces constructed in the previous section satisfies the primary axiom for playgrounds, namely that the codomain functor $\mathbb{D}_H \rightarrow \mathbb{D}_h$ is a fibration. We proceed as follows. We first define a (strong) factorisation system on $\widehat{\mathbb{C}}$ (Section 4.1), from which we derive in Section 4.2 an intermediate sub-double category $\mathbb{D} \hookrightarrow \mathbb{D}^1 \hookrightarrow \mathbb{D}^0$. We further show that by the properties of factorisation, the codomain functor $\mathbb{D}_H^1 \rightarrow \mathbb{D}_h^1$ is a fibration. Finally, we want to show that $\mathbb{D}_H \rightarrow \mathbb{D}_h$ is a fibration by proving that traces are stable under relevant cartesian lifting in \mathbb{D}^1 , i.e., cartesian liftings of any trace along any morphism in \mathbb{D}_h are again in \mathbb{D}_H . We first check this for seeds, by case analysis, in Section 4.3. In order to generalise to actions $M: Y \twoheadrightarrow X$, the basic idea is to

- (1) decompose X as a pushout of X_0 , where the generating seed M_0 takes place, and Z , which is passive;
- (2) decompose the morphism along which we want to restrict, say $h: X' \rightarrow X$, accordingly, say as $h_0: X'_0 \rightarrow X_0$ and $h_Z: Z' \rightarrow Z$;
- (3) restrict M_0 along h_0 to obtain P'_0 ;
- (4) recompose a trace P' from P'_0 and Z' ;
- (5) check that P' admits a cartesian morphism to M .

Step (4) is non-trivial, so we devote Section 4.4 to it. It works as a kind of formal opposite to restriction, as we essentially lift P'_0 along $h_0: X'_0 \rightarrow X_0$. We call this an *oplifting* of P'_0 along h_0 , by analogy with lifting in opfibrations. However, opliftings do not enjoy the relevant universal property (opcartesianness, which is dual to cartesianness). Instead, we find that opliftings are in fact cartesian! In Section 4.5, we use opliftings to show that actions are stable under restriction, following the above plan. Finally, we extend the result to arbitrary traces in Section 4.6.

4.1. A factorisation system. Let us start by defining the (strong) factorisation system $(\mathcal{V}, \mathcal{H})$ on $\widehat{\mathbb{C}}$, on which the intermediate sub-double category \mathbb{D}^1 will be based. The idea is that all three components of cartesian morphisms in \mathbb{D}_H are in \mathcal{H} , while t -legs of vertical morphisms are in \mathcal{V} . The cartesian lifting of any $V: Y' \twoheadrightarrow Y$ as in (3.2) along any $h: X \rightarrow Y$ is then given by factoring $t_V \circ h$ as $l \circ t_U$ with $t_U \in \mathcal{V}$ and $l \in \mathcal{H}$ to obtain

$$\begin{array}{ccc}
 X' & \xrightarrow{k} & Y' \\
 s_U \downarrow \lrcorner & & \downarrow s_V \\
 \tilde{U} & \xrightarrow{l} & V \\
 t_U \uparrow & & \uparrow t_V \\
 X & \xrightarrow{h} & Y,
 \end{array} \tag{4.1}$$

where the upper square is a pullback.

We recall from Definition 3.19 that \mathcal{V}_0 denotes the set of ‘ t ’-legs (i.e., lower legs) of seeds. Following Bousfield’s [8] construction of ‘cofibrantly generated’ factorisation systems, we define $\mathcal{H} = \mathcal{V}_0^\perp$ to be the class of maps f such that for any $t \in \mathcal{V}_0$ and commuting square $(u, v): t \rightarrow f$ in $\widehat{\mathbb{C}}^\rightarrow$, there exists a unique filler h making the following diagram commute:

$$\begin{array}{ccc}
X & \xrightarrow{u} & X' \\
t \downarrow & \dashrightarrow h & \downarrow f \\
Y & \xrightarrow{v} & Y'.
\end{array}$$

In this situation, one says that f is *right-orthogonal* to t , and t is *left-orthogonal* to f , which is denoted by $t \perp f$.

We finally define $\mathcal{V} = {}^\perp\mathcal{H}$ to consist of all maps which are left-orthogonal to any map in \mathcal{H} . Of course, we have $\mathcal{V}_0 \subseteq \mathcal{V}$. The following is an application of [8, Theorem 4.1]:

Proposition 4.1. *The pair $(\mathcal{V}, \mathcal{H})$ forms a factorisation system.*

What does that mean? Here is a modern definition [29]:

Definition 4.2. The classes of maps \mathcal{V} and \mathcal{H} form a factorisation system iff $\mathcal{V} = {}^\perp\mathcal{H}$, $\mathcal{V}^\perp = \mathcal{H}$, and any arrow factors as $h \circ v$ with $h \in \mathcal{H}$ and $v \in \mathcal{V}$.

In the case where $\mathcal{H} = \mathcal{V}_0^\perp$ and $\mathcal{V} = {}^\perp\mathcal{H}$, Bousfield proves that any map in $\widehat{\mathbb{C}}$ admits a factorisation using a transfinite construction (a so-called *small object* argument). But here we will only need factorisations of particular morphisms, which we will actually be able to calculate by hand. Bousfield's results include:

Lemma 4.3. *\mathcal{V} is stable under pushout and composition, contains all isomorphisms, and enjoys the right cancellation property, i.e., if $v \in \mathcal{V}$ and $fv \in \mathcal{V}$, then $f \in \mathcal{V}$.*

\mathcal{H} is stable under pullback and composition, contains all isomorphisms, and enjoys the left cancellation property, i.e., if $h \in \mathcal{H}$ and $hf \in \mathcal{H}$, then $f \in \mathcal{H}$.

Remark 4.4. Stability under pushout is ambiguous here: we mean that for any pushout

$$\begin{array}{ccc}
X & \longrightarrow & Y \\
v \downarrow & & \downarrow v' \\
Z & \longrightarrow & T,
\end{array}$$

if $v \in \mathcal{V}$, then $v' \in \mathcal{V}$. Stability under pullback is defined dually.

4.2. A first ‘fibred’ double category. We now make concrete the idea evoked in the previous section, of using our factorisation system to obtain a codomain fibration. Consider the sub-double category \mathbb{D}^1 of \mathbb{D}^0 obtained by restricting vertical morphisms to cospans $X \xrightarrow{s} U \xleftarrow{t} Y$ with $t \in \mathcal{V}$. Its vertical morphisms are stable under composition and contain identities by Lemma 4.3, i.e.:

Lemma 4.5. *\mathbb{D}^1 forms a sub-double category of \mathbb{D}^0 .*

Lemma 4.6. *Traces are in \mathbb{D}_v^1 , i.e., we have $\mathbb{D} \subseteq \mathbb{D}^1$.*

Proof. By Lemma 4.3. □

The main interest of introducing \mathbb{D}^1 is:

Lemma 4.7. *The codomain functor $\text{cod}: \mathbb{D}_H^1 \rightarrow \mathbb{D}_h^1$ is a fibration in which a double cell (3.2) is cartesian iff $l \in \mathcal{H}$ and the upper square is a pullback.*

Proof. Let us show that the lifting candidate computed in (4.1) is cartesian. Indeed, consider any double cell (4.1), and any morphism from some vertical morphism $Z' \rightarrow W \leftarrow Z$ to V whose bottom component factors through $h: X \rightarrow Y$. By unique lifting in $(\mathcal{V}, \mathcal{H})$, we obtain a unique dashed arrow making

$$\begin{array}{ccccc}
 Z & \longrightarrow & X & \longrightarrow & U \\
 \downarrow & \searrow & \downarrow & \nearrow & \downarrow \\
 & & Y & & \\
 \downarrow & \dashrightarrow & & \dashrightarrow & \downarrow \\
 W & \longrightarrow & & \longrightarrow & V
 \end{array}$$

commute. We finally obtain the desired arrow $Z' \rightarrow X'$ by universal property of pullback. Conversely, any cartesian double cell, being isomorphic to such a lifting, satisfies the conditions. \square

As a final observation, let us record:

Lemma 4.8. *Any morphism in \mathbb{D}_h is automatically in \mathcal{H} .*

Proof. Indeed, consider any $h: X \rightarrow Y$ in \mathbb{D}_h . There cannot be any commuting square

$$\begin{array}{ccc}
 X_0 & \xrightarrow{u} & X \\
 t \downarrow & & \downarrow h \\
 U & \xrightarrow{v} & Y
 \end{array}$$

with $t \in \mathcal{V}_0$, because U is a representable presheaf of dimension > 1 and Y has dimension ≤ 1 , so there cannot be any $v: U \rightarrow Y$. \square

4.3. Restriction of seeds. We now show that restrictions of seeds (in the sense of \mathbb{D}^1) are traces.

Lemma 4.9. *Consider any diagram $X' \xrightarrow{h} X \xrightarrow{t} M$, where $t \in \mathcal{V}_0$ and $h \in \mathbb{D}_h(X', X)$. Its factorisation $X' \xrightarrow{t'} M' \xrightarrow{h'} M$ with $t' \in \mathcal{V}$ and $h' \in \mathcal{H}$ is such that h' is 1-injective and the obtained restriction is a trace of length at most 2. If X' is an individual, i.e., a position of shape $[n]$, then it is isomorphic to some seed. If X' is an interface, then the restriction is an equivalence (in \mathbb{D}_v).*

Proof. We proceed by case analysis. In each case, one has to check that h' is 1-injective, that X' individual implies t' seed, that X' interface implies that t' is an isomorphism, and that the upper leg of the obtained cospan is as expected: this is routine so we mention it here once and for all.

Let us first treat the case where $M = y_c$, for c not of the shape $\tau_{n,i,m,j,k}$. Then, we have $X = [n]$ for some n . If $id_c \in \text{Im}(th)$, then $X' \cong [n] + I$ for some interface I (since h is 1-injective). Consider the diagram

$$\begin{array}{ccc} M + I & \xrightarrow{[id, tk]} & M \\ t + id_I \uparrow & & \uparrow t \\ [n] + I & \xrightarrow{h=[id, k]} & [n]. \end{array}$$

The map $t + id_I$ is in \mathcal{V} by Lemma 4.3, so we just have to prove that $[id, tk]$ is in \mathcal{V}_0^\perp , which is a simple verification.

If now $id_c \notin \text{Im}(h)$, then X' is an interface, and the relevant factorisation is

$$\begin{array}{ccc} X' & \xrightarrow{t \circ h} & M \\ id \uparrow & & \uparrow t \\ X' & \xrightarrow{h} & [n], \end{array} \quad (4.2)$$

because $t \circ h$ is easily checked to be in \mathcal{V}_0^\perp .

The case of $\tau_{n,i,m,j,k}$ is a bit more complicated. Here, t is actually $t_0 = [\rho t, \epsilon t]$. First of all, if X' is an interface, then we obtain a factorisation analogous to (4.2). Consider now the case where $\text{Im}(h)$ contains both agents of $[n]_i |_j [m]$. Let x denote the n -ary one and y denote the m -ary one (in X'). If $x \cdot s_i = y \cdot s_j$, then $X' = ([n]_i |_j [m]) + I$ for some interface I and the required factorisation is easily seen to be

$$\begin{array}{ccc} \tau_{n,i,m,j,k} + I & \xrightarrow{[id, t_0 k]} & \tau_{n,i,m,j,k} \\ t_0 + id_I \uparrow & & \uparrow t_0 \\ ([n]_i |_j [m]) + I & \xrightarrow{h=[id, k]} & [n]_i |_j [m]. \end{array}$$

Consider now the case where X' still contains both agents but $x \cdot s_i \neq y \cdot s_j$. Then $X' = [n] + [m] + I$ for some interface I , and the required factorisation is

$$\begin{array}{ccc} \iota_{n,i} + o_{m,j,k} + I & \xrightarrow{[\rho, \epsilon, t_0 k]} & \tau_{n,i,m,j,k} \\ t + id_I \uparrow & & \uparrow t_0 \\ [n] + [m] + I & \xrightarrow{[x, y, k]} & [n]_i |_j [m]. \end{array}$$

The only non-trivial point here is to show that $[\rho, \epsilon, t_0 k]$ is in \mathcal{V}_0^\perp , which easily reduces to showing that there is no commuting square

$$\begin{array}{ccc} [n]_i |_j [m] & \xrightarrow{u} & \iota_{n,i} + o_{m,j,k} + I \\ t_0 \downarrow & & \downarrow [\rho, \epsilon, t_0 k] \\ \tau_{n,i,m,j,k} & \xrightarrow{v} & \tau_{n,i,m,j,k}, \end{array}$$

which is true because there is no such u .

The cases where X' only contains one agent of $[n]_i |_j [m]$ are similar to the latter case: if it contains x then the factorisation is through $\iota_{n,i}$, and otherwise it is through $o_{m,j,k}$. \square

4.4. Opliftings. We now aim at extending Lemma 4.9 from seeds to actions. Consider an arbitrary action $Y \rightarrow M \leftarrow X$, obtained by pushing some seed $Y_0 \rightarrow M_0 \leftarrow X_0$ along $I_{X_0} \rightarrow Z$. Consider now a morphism $h: X' \rightarrow X$ in \mathbb{D}_h , along which we wish to restrict M . As explained at the beginning of Section 4, our strategy is to consider the pullback

$$\begin{array}{ccc} X'_0 & \longrightarrow & X_0 \\ \downarrow & \lrcorner & \downarrow \\ X' & \longrightarrow & X. \end{array}$$

Indeed, the part of X really concerned by M is the image of X_0 , so we would like to first restrict M_0 to X'_0 using Lemma 4.9, and then extend it to X' . The present section is devoted to constructing the second step, the extension of $(M_0)|_{X'_0}$ to X' , which we call an *oplifting* of $(M_0)|_{X'_0}$ along $X'_0 \rightarrow X'$.

In the general case, we want to compute the oplifting of $Y \rightarrow U \leftarrow X$ along $h: X \rightarrow X'$. Because h is 1-injective, we can complete the solid part of

$$\begin{array}{ccc} I_X & \dashrightarrow & Z \\ \sqsubseteq \int & & \int \sqsubseteq \\ X & \xrightarrow{h} & X' \end{array}$$

into a pushout. Indeed, we take $Z(\star) = X'(\star)$, $Z[n] = X'[n] \setminus \text{Im}(h_{[n]})$ for all n , and $I_X \rightarrow Z$ is uniquely determined by h_* . In passing, we have:

Lemma 4.10. *This pushout is uniquely determined up to canonical isomorphism by h alone.*

Then, we observe that, because U is a trace, $I_X \rightarrow X \rightarrow U$ factors through $Y \rightarrow U$.

Definition 4.11. Let the *oplifting* of $U: Y \rightarrow X$ along $h: X \rightarrow X'$ be the cospan obtained as in (3.1) by pushing U along $I_X \rightarrow Z$.

All horizontal maps are 1-injective by construction, and we have:

Lemma 4.12. *The obtained cospan $Y' \rightarrow U' \leftarrow X'$ is a trace.*

Proof. We start by showing that U' is an action if U is. So assume U is obtained by pushing a seed $Y_0 \rightarrow M_0 \leftarrow X_0$ along some $I_{X_0} \rightarrow Z_0$. Then, $Z_0 \rightarrow X$ is surjective on \star because $I_{X_0} \rightarrow X_0$ is and epis are stable under pushout in presheaf categories. Thus, $I_X \rightarrow X$ factors through $Z_0 \rightarrow X$. Let Z'' denote the pushout

$$\begin{array}{ccc} I_X & \longrightarrow & Z_0 \\ \downarrow & \lrcorner & \downarrow \\ Z & \longrightarrow & Z''. \end{array}$$

By the pushout lemma, $Y' \rightarrow U' \leftarrow X'$ is isomorphic to the cospan obtained by pushing $Y \rightarrow U \leftarrow X$ along $Z_0 \rightarrow Z''$. By the pushout lemma again, it is isomorphic to the cospan obtained by pushing $Y_0 \rightarrow M_0 \leftarrow X_0$ along $I_{X_0} \rightarrow Z_0 \rightarrow Z''$. Thus, it is indeed an action.

We now prove the general case by induction on $|U|$. This is trivial if U is isomorphic to an identity. If now U is a composite $Z \xrightarrow{V} Y \xrightarrow{M} X$, then we compute the oplifting of M along h to obtain a double cell, say

$$\begin{array}{ccc} Y & \xrightarrow{h'} & Y' \\ M \downarrow & \xrightarrow{\alpha_M} & \downarrow M' \\ X & \xrightarrow{h} & X', \end{array}$$

by pushing M along some morphism $I_X \rightarrow Z_1$ making

$$\begin{array}{ccc} I_X & \longrightarrow & Z_1 \\ \downarrow & \searrow & \downarrow \\ X & \xrightarrow{h} & X' \end{array}$$

into a pushout.

The crucial insight is then that by computing the following pushout Z_2 and applying its universal property, we obtain a diagram

$$\begin{array}{ccccc} I_X & \longrightarrow & I_Y & \longrightarrow & Y \\ \downarrow & & \downarrow & & \downarrow \\ Z_1 & \longrightarrow & Z_2 & \longrightarrow & Y' \end{array} \quad (4.3)$$

whose exterior is a pushout by construction. So by the pushout lemma the right-hand square is again a pushout, which by Lemma 4.10 is the unique pushout along which to push V to compute the oplifting of V along h' , say

$$\begin{array}{ccc} Z & \xrightarrow{h''} & Z' \\ V \downarrow & \xrightarrow{\alpha_V} & \downarrow V' \\ Y & \xrightarrow{h'} & Y'. \end{array}$$

By induction hypothesis, V' is again a trace. But by the pushout lemma again, α_V is also what we obtain by pushing V along the exterior rectangle of (4.3). A bunch of applications of the pushout lemma finally yields that $\alpha_M \bullet \alpha_V$ is the oplifting of $M \bullet V$ along h , so $M' \bullet V'$ is a trace by construction. \square

Although opliftings have an opcartesian flavour, they are in fact not opcartesian in general, and moreover opcartesian liftings do not exist in general.

Example 4.13. Consider the oplifting of the seed $\iota_{1,1}$ along $[1] \hookrightarrow [1]_1|_1[1]$, say $\iota_{1,1} \uparrow_1|_1[1]$, whose final position is $[2]_1|_1[1]$. To see that it is not opcartesian, consider the diagram

$$\begin{array}{ccccc} & & & & [2]_{1,2}|_{1,1}[1] \\ & & & \nearrow & \downarrow \\ [2] & \longrightarrow & [2]_1|_1[1] & \xrightarrow{\quad} & \tau_{1,1,1,1,1} \\ \downarrow & & \downarrow & \nearrow & \uparrow \\ \iota_{1,1} & \longrightarrow & \iota_{1,1} \uparrow_1|_1[1] & \xrightarrow{\quad} & [1]_1|_1[1] \\ \uparrow & & \uparrow & \nearrow & \downarrow \\ [1] & \longrightarrow & [1]_1|_1[1] & \xrightarrow{\quad} & [1]_1|_1[1] \end{array}$$

In this case, opcartesianness would mean finding dashed arrows rendering the diagram commutative. There is indeed a unique arrow $\iota_{1,1,1}|_1[1] \rightarrow \tau_{1,1,1,1,1}$ making the corresponding square commute, but then no arrow $[2]_{1|1}[1] \rightarrow [2]_{1,2|1,1}[1]$ fits. Indeed, there is only one arrow of the given type, and it does not make the relevant square

$$\begin{array}{ccc} [2]_{1|1}[1] & \longrightarrow & [2]_{1,2|1,1}[1] \\ \downarrow & & \downarrow \\ \iota_{1,1,1}|_1[1] & \longrightarrow & \tau_{1,1,1,1,1} \end{array}$$

commute, because one side (down; right) of the square maps the unary player to $\rho \circ t$, while the other (right; down) maps it to $\rho \circ s$. This shows that $\iota_{1,1,1}|_1[1]$ is not an opcartesian lifting of $\iota_{1,1}$. But in fact, $\iota_{1,1,1}|_1[1]$ and $\tau_{1,1,1,1,1}$ are two liftings of $\iota_{1,1}$ along $[1] \hookrightarrow [1]_{1|1}[1]$. Thus, any opcartesian lifting should have a final position mapping both to $[2]_{1|1}[1]$ and $[2]_{1,2|1,1}[1]$, hence containing just one, binary player: no trace can meet this requirement.

However, even though they are not opcartesian, opliftings are in fact cartesian. Let us now show this, starting with a few preliminary results.

Definition 4.14. Let $\text{Agents}(X) = \sum_{n \in \mathbb{N}} X[n]$ denote the set of agents of any position X .

Lemma 4.15. For any seed $Y \rightarrow C \leftarrow X$, the morphism

$$\sum_{(n,x) \in \text{Agents}(X)} [n] \xrightarrow{[x]_{(n,x)}} X$$

is epi.

Proof. This is trivial except in dimension 0, where it holds by case inspection. \square

Corollary 4.16. For all arrows as in

$$\begin{array}{ccc} X & \begin{array}{c} \xrightarrow{h} \\ \xrightarrow{h'} \end{array} & U \\ t \downarrow & & \downarrow f \\ C & \xrightarrow{g} & U' \end{array}$$

in $\widehat{\mathcal{C}}$ such that f is 1-injective, $t \in \mathcal{V}_0$, and $fh = gt = fh'$, we have $h = h'$.

Proof. We construct the diagram

$$\begin{array}{ccc} \sum_{(n,x) \in \text{Agents}(X)} [n] & \begin{array}{c} \xrightarrow{he} \\ \xrightarrow{h'e} \end{array} & U \\ e \downarrow & & \parallel \\ X & \begin{array}{c} \xrightarrow{h} \\ \xrightarrow{h'} \end{array} & U \\ t \downarrow & & \downarrow f \\ C & \xrightarrow{g} & U' \end{array}$$

and observe that $h'e = he$ by 1-injectivity of f , hence $h = h'$ because e is epi by Lemma 4.15. \square

Lemma 4.17. *Opliftings of traces are cartesian.*

Proof. Consider any oplifting $U \xrightarrow{f} U'$ of $Y \rightarrow U \leftarrow X$ along, say $X \rightarrow X'$. By Lemma 4.7, it is enough to show that the middle arrow $U \rightarrow U'$ is in \mathcal{H} and that its upper square is a pullback. The latter follows from Lemma 3.20. So we just have to show that any square

$$\begin{array}{ccc} Z & \xrightarrow{u} & U \\ t \downarrow & & \downarrow \\ C & \xrightarrow{v} & U' \end{array}$$

with $t \in \mathcal{V}_0$ admits a unique lifting $C \rightarrow U$. By the Yoneda lemma, v amounts to an element of $U'(C)$ of dimension > 1 , but by construction U and U' have exactly the same such elements. This yields a candidate lifting, say k , which is unique by 1-injectivity and makes the bottom triangle commute by construction. The top one finally commutes by Corollary 4.16 with $h = u$ and $h' = kt$. \square

4.5. Restriction of actions. Let us now extend Lemma 4.9 from seeds to actions, following the strategy sketched at the beginning of Section 4.

Lemma 4.18. *For any action $Y \xrightarrow{s} M \xleftarrow{t} X$ and $h \in \mathbb{D}_h(X', X)$, the factorisation $X' \xrightarrow{t'} P' \xrightarrow{h'} M$ of t with $t' \in \mathcal{V}$ and $h' \in \mathcal{H}$ is such that h' is 1-injective and the obtained restriction is a trace of length at most 2. If X' is an individual then it is either a seed or an equivalence; if X' is an interface then it is an equivalence.*

Proof. Consider any action $Y \rightarrow M \leftarrow X$ obtained by pushing the following seed-with-interface along $I \rightarrow Z$:

$$\begin{array}{ccccc} & & I & & \\ & \swarrow & \downarrow & \searrow & \\ Y_0 & \longrightarrow & M_0 & \longleftarrow & X_0. \end{array}$$

By Lemma 4.17, the morphism $M_0 \rightarrow M$ is cartesian.

Consider the pullback of the bottom square below along $h: X' \rightarrow X$ to obtain the top square

$$\begin{array}{ccccc} I' & \xrightarrow{\quad} & Z' & & \\ \downarrow & \searrow & \downarrow & \searrow & \\ & X_0' & \xrightarrow{\quad} & X' & \\ \downarrow & \downarrow & \downarrow & \downarrow & \\ I & \xrightarrow{\quad} & Z & & \\ \downarrow & \searrow & \downarrow & \searrow & \\ & X_0 & \xrightarrow{\quad} & X & \end{array}$$

which, because presheaf categories are *adhesive* [46] and $I \rightarrow X_0$ is mono, is again a pushout.

Furthermore, consider the front face, which is a pullback in \mathbb{D}_h . By Lemma 4.9, restricting M_0 along $X_0' \rightarrow X_0$ yields a trace, say $Y_0' \rightarrow P_0' \leftarrow X_0'$ with a morphism to $Y_0 \rightarrow M_0 \leftarrow X_0$ in

\mathbb{D}_H . Since it is a trace, $I' \hookrightarrow X'_0 \hookrightarrow M'_0$ factors through $Y'_0 \hookrightarrow M'_0$. Pushing $Y'_0 \rightarrow P'_0 \leftarrow X'_0$ along $I' \rightarrow Z'$, we obtain a trace $Y' \rightarrow P' \leftarrow X'$ (we may choose X' as initial position because the top square above is a pushout) with a morphism from $Y'_0 \rightarrow P'_0 \leftarrow X'_0$ in \mathbb{D}_H , which is an oplifting, hence cartesian (by Lemma 4.17). Then, by universal property of pushout we obtain a unique morphism $f: P' \rightarrow M$ making the following cube commute:

$$\begin{array}{ccccc}
 I' & \xrightarrow{\quad} & Z' & & \\
 \downarrow & \searrow & \downarrow & \searrow & \\
 & P'_0 & \xrightarrow{\quad} & P' & \\
 \downarrow & \downarrow & \downarrow & \downarrow & \\
 I & \xrightarrow{\quad} & Z & & \\
 \downarrow & \searrow & \downarrow & \searrow & \\
 & M_0 & \xrightarrow{\quad} & M & \\
 & & & & \downarrow f
 \end{array}$$

By Corollary 3.22, f is 1-injective, which entails that the induced morphism of traces is in \mathbb{D}_H .

We now need to show that $P' \rightarrow M$ is cartesian, which by Lemmas 3.20 and 4.7 amounts to showing that its middle arrow $f: P' \rightarrow M$ is in \mathcal{H} . To this end, consider any morphism $t: Z'' \rightarrow C$ in \mathcal{V}_0 and morphism $(u, v): t \rightarrow f$ in $\widehat{\mathcal{C}}^{\rightarrow}$. First of all, because $M_0 \rightarrow M$ is identity in dimensions > 1 , the morphism v uniquely factors through $M_0 \rightarrow M$. Furthermore, in all cases where $f_0: P'_0 \rightarrow M_0$ is identity in dimensions > 1 , the Yoneda lemma entails that $C \rightarrow M_0$ uniquely factors through P'_0 . This yields a diagram

$$\begin{array}{ccccc}
 & & u & & \\
 & & \curvearrowright & & \\
 Z'' & \xrightarrow{\quad l \quad} & P'_0 & \xrightarrow{\quad g \quad} & P' \\
 \downarrow & \searrow k & \downarrow f_0 & & \downarrow f \\
 C & \xrightarrow{\quad} & M_0 & \xrightarrow{\quad} & M \\
 & & \curvearrowleft & & \\
 & & v & &
 \end{array}$$

which commutes except perhaps for the upper part marked ‘?’ . But the latter also commutes by Corollary 4.16 with $h = u$ and $h' = gl$. We thus obtain a lifting, which is unique by 1-injectivity of f .

So what happens when is f_0 non-identity in dimensions > 1 ? By inspection of the proof of Lemma 4.9, except for the easy case where $P'_0 \cong X'_0$, this is when $M_0 = \tau_{n,i,m,j,k}$ for some n, m, i, j, k , and P'_0 has one of the shapes $\iota_{n,i} + J$, $o_{m,j,k} + J$, or $\iota_{n,i} + o_{m,j,k} + J$, for some interface J . In the first two cases, $C \neq \tau_{n,i,m,j,k}$ because there can be no $u: [n]_i |_j [m] \rightarrow P'$ (one agent is missing in P'), so the previous argument applies. In the third case, letting x and y respectively denote the n - and m -ary initial agents in P'_0 and $a = x \cdot s_i$ and $a' = y \cdot s_j$ the corresponding channels, one easily shows that $g(a) \neq g(a')$, so again there can be no $u: [n]_i |_j [m] \rightarrow P'$. Thus, $C \neq \tau_{n,i,m,j,k}$ and the previous argument again applies. \square

4.6. Restriction of traces. So far, we have shown that seeds and actions admit cartesian liftings in \mathbb{D}_H . We now show that it is also the case for arbitrary traces. We proceed by induction on the length of the considered trace, which requires a few preliminary results.

Lemma 4.19. *In sets, for any commuting diagram*

$$\begin{array}{ccccc} I & \longrightarrow & A & \longrightarrow & C \\ \downarrow & & \downarrow & & \downarrow \\ J & \longrightarrow & B & \longrightarrow & D \end{array}$$

whose exterior rectangle is a pullback, with the marked pushout and monos, the right-hand square is also a pullback.

Proof. We check the universal property of pullback for A , relative to 1 , which is enough in sets. So consider any commuting square

$$\begin{array}{ccc} 1 & \xrightarrow{c} & C \\ b \downarrow & & \downarrow \\ B & \longrightarrow & D. \end{array}$$

First, we observe that there is at most one mediating arrow $1 \rightarrow A$, because $A \rightarrow B$ is mono.

If b has an antecedent in A , say a , then because $C \rightarrow D$ is mono, a makes both required triangles commute and we are done.

Otherwise, by surjectivity of $A + J \rightarrow B$, b admits an antecedent in J , i.e., there exists $j: 1 \rightarrow B$ such that b is $1 \xrightarrow{j} J \rightarrow B$, then we have a cone to $C \rightarrow D \leftarrow J$, so we apply the universal property of pullback to obtain i as in

$$\begin{array}{ccccc} & & & & 1 \\ & & & & \swarrow c \\ & & & & C \\ & & & & \downarrow \\ & & & & D \\ & & & & \uparrow \\ & & & & B \\ & & & & \downarrow b \\ & & & & J \\ & & & & \swarrow j \\ & & & & 1 \\ & & & & \searrow i \\ & & & & I \\ & & & & \longrightarrow A \longrightarrow C \\ & & & & \downarrow \downarrow \\ & & & & B \longrightarrow D \end{array}$$

making everything commute and so $1 \xrightarrow{i} I \rightarrow A$ suits our needs. □

Corollary 4.20. *In any presheaf category, in any commuting cube*

$$\begin{array}{ccccc} I & \longrightarrow & B & & \\ \downarrow & \searrow & \downarrow & \searrow & \\ A & \longrightarrow & C & & \\ \downarrow & & \downarrow & & \downarrow f \\ I' & \longrightarrow & B' & & \\ \downarrow & \searrow & \downarrow & \searrow & \\ A' & \longrightarrow & C' & & \end{array}$$

with the marked pushouts, pullback, and mono, the front square is also a pullback.

Proof. It suffices to show the result in sets, as all involved properties are pointwise in presheaf categories. First, as monos are stable under pullback and pushout, $I \rightarrow B$, $A \rightarrow C$, and $A' \rightarrow C'$ are also monos. Furthermore, pushouts along monos are also pullbacks, so the top and bottom faces are also pullbacks. By the pullback lemma, the rectangle

$$\begin{array}{ccccc} I & \longrightarrow & I' & \longrightarrow & A' \\ \downarrow & \lrcorner & \downarrow & \lrcorner & \downarrow \\ B & \longrightarrow & B' & \longrightarrow & C' \end{array}$$

is a pullback. The previous lemma thus entails the result. \square

Lemma 4.21. *For any commuting diagram as the solid part of*

$$\begin{array}{ccccc} T & \longrightarrow & V' & \longrightarrow & U' \\ t \downarrow & \nearrow k & h \downarrow & \dashrightarrow & \downarrow f \\ C & \longrightarrow & V & \longrightarrow & U \end{array}$$

with $t \in \mathcal{V}_0$, $h \in \mathcal{H}$, and f 1-injective, there is a unique lifting k as shown.

Proof. Because $h \in \mathcal{H}$, there is a unique map from $k': C \rightarrow V'$ making both triangles commute. By composing k' with $V' \rightarrow U'$, we obtain a lifting k for the desired square. Uniqueness follows from 1-injectivity of f . \square

Lemma 4.22. *The codomain functor $\text{cod}: \mathbb{D}_H \rightarrow \mathbb{D}_h$ is a subfibration of $\text{cod}: \mathbb{D}_H^1 \rightarrow \mathbb{D}_h^1$.*

Proof. First of all, it is sufficient to prove that given any trace $Y \xrightarrow{s} U \xleftarrow{t} X$ and $h \in \mathbb{D}_h(X', X)$, any cartesian lifting in \mathbb{D}^1 lies in \mathbb{D} , i.e., the obtained vertical morphism is again a trace and the double cell to U lies in \mathbb{D} (i.e., all its components are 1-injective). Indeed, mediating morphisms computed in \mathbb{D}^1 are automatically in \mathbb{D} by cancellation.

We proceed by induction on U . If U is an equivalence, then the result is obviously true. Otherwise, $U = M \bullet V$ for some action M and trace V . Let us call Z the final position of M . By Lemma 4.18, M admits a lifting P' along h , with a final position Z' , and $Z' \rightarrow Z$ and $f_M: P' \rightarrow M$ are 1-injective. By induction hypothesis, V admits a lifting V' along $Z' \rightarrow Z$ with a double cell to V in \mathbb{D}_H . Therefore, considering the composite $P' \bullet V'$, we have a commuting diagram as in Figure 4.6, where f is obtained by universal property of pushout.

Because pushouts along monos are also pullbacks in presheaf categories, both marked pushouts are also pullbacks. Furthermore, by Lemma 3.20, $Z' = P' \times_M Z$ and $Y' = (V' \times_V Y)$, as shown. Also, by Corollary 4.20, $V' = (P' \bullet V') \times_{M \bullet V} V$, as shown. Furthermore, by Proposition 3.18, f is 1-injective.

By Lemmas 4.7 and 3.20, it suffices to show that f is in \mathcal{H} , i.e., that it is right-orthogonal to any $T \xrightarrow{t} C$ in \mathcal{V}_0 . Consider any commuting square

$$\begin{array}{ccc} T & \xrightarrow{u} & P' \bullet V' \\ t \downarrow & & \downarrow f \\ C & \xrightarrow{v} & M \bullet V. \end{array}$$

Since $M \bullet V$ is the coproduct of M and V in dimensions greater than 1 and C is a representable of dimension greater than 1, we have that v factors either through t_2 or s_2 .

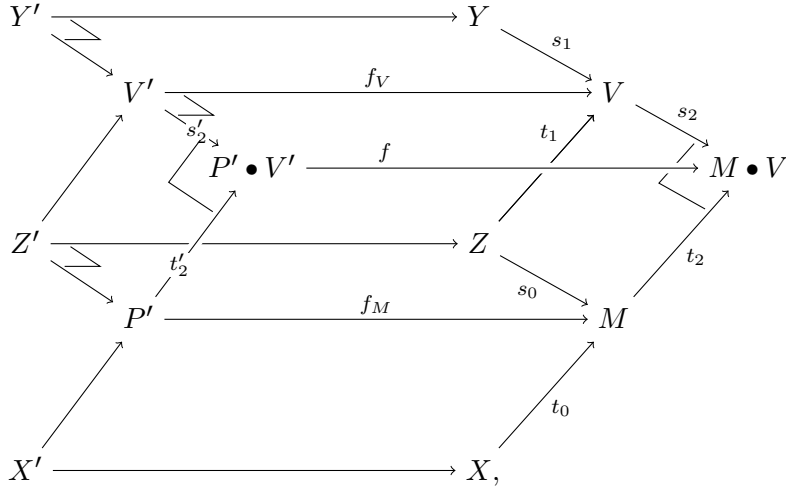


Figure 5: Diagram for induction step in Lemma 4.22

If v factors through s_2 , then by universal property of pullback we find a map $u' : T \rightarrow V'$ making

$$\begin{array}{ccccc}
 & & u & & \\
 & \curvearrowright & & \curvearrowright & \\
 T & \xrightarrow{u'} & V' & \longrightarrow & P' \bullet V' \\
 \downarrow t & & \downarrow & \lrcorner & \downarrow \\
 C & \longrightarrow & V & \xrightarrow{s_2} & M \bullet V \\
 & \curvearrowright & & \curvearrowright & \\
 & & v & &
 \end{array}$$

commute. Then, by Lemma 4.21, we find a unique lifting as desired.

If v factors as $t_2 v'$, then by Lemma 4.21, it is sufficient to show that there is a map $u' : T \rightarrow P'$ making

$$\begin{array}{ccccc}
 & & u & & \\
 & \curvearrowright & & \curvearrowright & \\
 T & \xrightarrow{u'} & P' & \xrightarrow{t_2'} & P' \bullet V' \\
 \downarrow t & & \downarrow & & \downarrow f \\
 C & \xrightarrow{v'} & M & \xrightarrow{t_2} & M \bullet V \\
 & \curvearrowright & & \curvearrowright & \\
 & & v & &
 \end{array}$$

commute. To that end, it is sufficient to show that for every $[n] \xrightarrow{x} T$, there is a map $[n] \xrightarrow{f_x} P'$ such that

$$\begin{array}{ccc}
 [n] & \xrightarrow{x} & T \\
 f_x \downarrow & & \downarrow u \\
 P' & \xrightarrow{t_2'} & P' \bullet V'
 \end{array} \tag{4.4}$$

commutes. Indeed, if that is the case, then the square

$$\begin{array}{ccc}
 \Sigma_{(n,x) \in \text{Agents}(T)} [n] & \xrightarrow{[x]_{(n,x) \in \text{Agents}(T)}} & T \\
 [f_x]_{(n,x) \in \text{Agents}(T)} \downarrow & \swarrow^{u'} & \downarrow u \\
 P' & \xrightarrow{t'_2} & P' \bullet V'
 \end{array}$$

also commutes, and since its bottom map is mono and its top map is epi by Lemma 4.15, there is a unique lifting $u': T \rightarrow P'$ making both triangles commute. The square

$$\begin{array}{ccc}
 T & \xrightarrow{u'} & P' \\
 t \downarrow & & \downarrow \\
 C & \xrightarrow{v'} & M
 \end{array}$$

also commutes because it commutes when composed with t_2 , which is mono.

So we now need to show that for every $[n] \xrightarrow{x} T$, there is a map $[n] \xrightarrow{f_x} P'$ making the square (4.4) commute. Because $P' + V' \xrightarrow{[t'_2, s'_2]} P' \bullet V'$ is epi and $[n]$ is a representable presheaf, $[n] \xrightarrow{ux} P' \bullet V'$ factors through either t'_2 or s'_2 .

If it factors through s'_2 , say as $s'_2 x'$, then we have

$$t_2(v'tx) = vtx = fux = fs'_2 x' = s_2 f_V x',$$

so by universal property of the pullback Z there exists a unique $x'': [n] \rightarrow Z$ such that $s_0 x'' = v'tx$ and $t_1 x'' = f_V x'$. We thus obtain a commuting diagram

$$\begin{array}{ccccc}
 [n] & \xrightarrow{x''} & & \longrightarrow & Z \\
 x \downarrow & & & & \downarrow s_0 \\
 T & \xrightarrow{t} & C & \xrightarrow{v'} & M,
 \end{array}$$

which is impossible because $v'tx$, as one of the agents performing action C in M , cannot remain in the final position of M .

Thus, ux factors through t'_2 and we are done. \square

5. A PLAYGROUND FOR π

In the previous section, we have proved the main playground axiom, asserting that the codomain functor $\mathbb{D}_H \rightarrow \mathbb{D}_h$ of the double category of traces constructed in Section 3 is a fibration. We now prove the remaining axioms. In Section 5.1, we equip \mathbb{D} with playground structure and prove that it satisfies all the needed axioms, except both decomposition axioms ((P6) and (P7)) which require a bit more work. In Section 5.2, we establish a correctness criterion detecting when a given cospan in $\widehat{\mathbb{C}}$ is a trace. We then use this criterion in Section 5.3 to prove both remaining axioms.

5.1. A candidate playground. So we start in this section by defining the needed additional structure on \mathbb{D} .

Definition 5.1. We recall from Lemma 4.9 that \mathbb{I} , the set of *individuals*, consists of representable positions $[n]$. Let \mathbb{B} , the full subcategory of *basic* actions, span all seeds of shape $\tau_n, \pi_n^r, \pi_n^l, \nu_n, \heartsuit_n, \iota_{n,i}$, or $o_{m,j,k}$. *Full* actions (notation \mathbb{F}) are all actions obtained from seeds of shape $\tau_n, \pi_n, \nu_n, \heartsuit_n, \iota_{n,i}, o_{m,j,k}$, or $\tau_{n,i,m,j,k}$. *Closed-world* actions are all actions obtained from seeds of shape $\tau_n, \pi_n, \nu_n, \heartsuit_n$, or $\tau_{n,i,m,j,k}$. Let \mathbb{W} denote the graph with positions as vertices and closed-world actions between them as edges (the initial position being the target). Finally, all decompositions of any trace U into actions have the same length which we denote by $|U|$.

Here is a summary of which actions are basic, full and closed-world:

	τ_n	π_n^l	π_n^r	π_n	$\iota_{n,i}$	$o_{m,j,k}$	$\tau_{n,i,m,j,k}$	ν_n	\heartsuit_n
Basic	✓	✓	✓		✓	✓		✓	✓
Full	✓			✓	✓	✓	✓	✓	✓
Closed-world	✓			✓			✓	✓	✓

Remark 5.2. The definition matches the explanations following Definition 2.30. Basic actions are as small as possible, which here means they start from one agent and only retain one agent in the final position. Full actions are those that retain all possible agents in the final position. So the only kind of actions which are not basic are π_n and $\tau_{n,i,m,j,k}$. They each have one sub-action for each agent in their final positions, namely π_n^r and π_n^l for π_n , and $\iota_{n,i}$ and $o_{m,j,k}$ for $\tau_{n,i,m,j,k}$. All of these sub-actions are basic, but only $\iota_{n,i}$ and $o_{m,j,k}$ are full. Finally, a new class of actions appear here, that of closed-world actions. Intuitively, it consists of those actions that do not involve any interaction with the environment. Or, in other words, those that cannot be extended, even by adding new agents. E.g., $\iota_{n,i}$ may be completed to $\tau_{n,i,m,j,k}$ by adding an agent, hence is not closed-world. But $\tau_{n,i,m,j,k}$ is. Closed-world actions will be at the basis of our semantic notion of testing equivalence (Definition 6.24).

Lemma 5.3.

- (P1) \mathbb{I} , viewed as a subcategory of \mathbb{D}_h , is discrete. Basic actions have no non-trivial automorphisms in \mathbb{D}_H . Vertical identities on individuals have no non-trivial endomorphisms.
- (P2) (Individuality) Basic actions have individuals as both domain and codomain.
- (P3) (Atomicity) For any cell $\alpha: U \rightarrow U'$, if $|U'| = 0$ then also $|U| = 0$. Up to a special isomorphism in \mathbb{D}_H , all traces of length $n > 0$ admit decompositions into n actions. For any $U: X \twoheadrightarrow Y$ of length 0, there is an isomorphism $id_X^\bullet \rightarrow U$ in \mathbb{D}_H as in

$$\begin{array}{ccc}
 X & \xlongequal{\quad} & X \\
 \parallel & \xrightarrow{\alpha^{U'}} & \downarrow U \\
 X & \xrightarrow{\bar{U}} & Y.
 \end{array}$$

- (P4) Restrictions of actions (resp. full actions) to individuals either are actions (resp. full actions), or have length 0.

Proof. (P1) and (P2) are direct by Yoneda. (P4) is also easy in view of Lemma 4.22 and its proof. For (P3), any vertical $X \rightarrow U \leftarrow Y$ of length 0, being a trace, is isomorphic to an identity cospan, say $Z \rightarrow Z \leftarrow Z$. To construct α^U , just take the composite $id_X^\bullet \xrightarrow{\cong} id_Z^\bullet \xrightarrow{\cong} U$. \square

Let us now treat the axiom for views, which is really easy. It actually becomes stronger because of Remark 2.28, though this does not affect the rest of the construction:

Definition 5.4. Let \mathbb{B}_0 be the full subcategory of \mathbb{D}_H having as objects basic actions and vertical identities between individuals.

Lemma 5.5. (P5) For any action $M:Y \dashrightarrow X$ and $y:d \rightarrow Y$ in \mathbb{D}_h with $d \in \mathbb{I}$, there exists a unique cell

$$\begin{array}{ccc} d & \xrightarrow{y} & Y \\ v^{y,M} \downarrow \bullet & \overset{\alpha^{y,M}}{=} & \downarrow M \\ d^{y,M} & \dashrightarrow_{y^M} & X, \end{array}$$

with $v^{y,M} \in \mathbb{B}_0$.

Proof. The result holds for seeds, by case analysis. E.g., $v^{ls,\pi_n} = \pi_n^l$, $v^{rs,\pi_n} = \pi_n^r$, and so on. Now, any action M comes with a cell from its generating seed, say $\beta:M_0 \rightarrow M$. If y is in the image of M_0 , then the required cell $\alpha^{y,M}$ is $\beta \circ \alpha^{y,M_0}$. Otherwise, $v^{y,M} = id_d^\bullet$ admits a cell to M , which suits our needs. Uniqueness follows by (P1). \square

Remark 5.6. The important result that Axiom (P5) entails [39, Proposition 4.24] says that when we replace M with any trace u , we get a double cell $\alpha^{y,u}$ which is only unique up to isomorphism. Below, we still define views up to isomorphism, so our modified Axiom (P5) does not make this any stronger.

We conclude this section with the (straightforward) verification of (P8) and (P9).

Lemma 5.7. (P8) For any X , \mathbb{I}/X is finite.

Proof. All positions are finitely presentable presheaves. \square

Lemma 5.8. (P9) For all $d \in \mathbb{I}$ and actions $M:X \dashrightarrow d$, $M':X' \dashrightarrow d$, and $b:d' \dashrightarrow d$ with M and M' full and b basic, if there exist cells $M \leftarrow b \rightarrow M'$ then $M \cong M'$.

Proof. By case analysis. E.g., if $b = \pi_n^l$, then $M \cong \pi_n \cong M'$. \square

Remark 5.9. While the verification of (P9) is straightforward in our case, this axiom does impose strong constraints on playgrounds. Morally, it demands that the basic subactions of a given full action should be disjoint from those of a different full action. To see why this is restrictive, let, for all $j \in n$, $\iota_{n,i,j}$ denote the quotient of $\iota_{n,i}$ by the equation $s \circ s_{n+1} = s \circ s_j$. The equation says that the received channel was already known as channel number j . Further let $[n]/\{i = j\}$ denote $[n]$ quotiented by $s_i = s_j$. We could be tempted to decree that the cospan

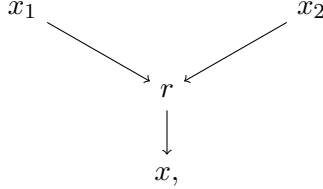
$$[n+1]/\{n+1 = j\} \xrightarrow{s} \iota_{n,i,j} \xleftarrow{t} [n]$$

is an action. An example consequence would be that, e.g., the synchronisation on $[n]_{i,l}|_{j,k}[m]$ where $[m]$ sends k on j , when restricted to the receiver, would give $\iota_{n,i,l}$ instead of $\iota_{n,i}$. But then $\iota_{n,i}$ and $\iota_{n,i,l}$ would be two non-isomorphic full actions sharing a common basic subaction, $\iota_{n,i}$.

We now have proved all playground axioms for \mathbb{D} , except right and left decomposition. These require the development of more machinery, which we undertake in the next section.

5.2. Correctness criterion. In order to prove the remaining playground axioms for \mathbb{D} , we set up a combinatorial characterisation of traces among cospans. Before delving into technicalities, let us briefly map out our correctness criterion. Given a trace $Y \hookrightarrow U \leftarrow X$, we start by forgetting the cospan structure and exploring the properties of U alone.

The main idea is to construct a binary relation over the elements of U , modeling causality. So, e.g., if an agent $x \in U[n]$ forks into x_1 and x_2 , then we will have causal relations



where r denotes the corresponding element in $U(\pi_n)$. In order for U to admit a sequential decomposition into actions, the main criterion is that the causality relation should be acyclic.

In addition to this, a few sanity checks are necessary. First of all, because actions are merely seeds pushed along 1-injective maps from their interfaces, the neighbourhood of each action $x \in U(\mu)$ should not be too degenerate. For instance, the corresponding map $\lceil x \rceil: \mu \rightarrow U$ should be 1-injective. Moreover, for inputs and channel creations, the new channel should really be new. This property, which is a bit tedious to define properly, is called *local 1-injectivity*.

Furthermore, when we add a new action to some trace, it is played by an agent in the final position. This entails that no two actions in U may be performed by the same agent. We call this *target-linearity* below. Symmetrically, no two actions may share their ‘created’ agents, which we call *source-linearity*. *Linearity* is then the conjunction of source- and target-linearity.

These conditions are sufficient, in the sense that if any $U \in \widetilde{\mathcal{C}}_f$ has an acyclic causal relation, and is furthermore locally 1-injective and linear, then it is the middle object of a trace. But in fact, it is then easy to determine the corresponding initial and final positions. We design notions of *initial* and *final* morphisms, so that $Y \xrightarrow{s} U \xleftarrow{t} X$ is a trace iff U satisfies the above conditions, t is initial, and s is final.

Let us first define the causal relation. A first step is to restrict attention to the *cores* of U , in the following sense, which are intuitively the main elements. E.g., for a forking action $x \in U(\pi_n)$, keeping track of x is enough, and handling $x \cdot l$ and $x \cdot r$ tends to get in the way. Technically, an input or output is a core iff it is not part of a synchronisation; and a left or right fork action is a core iff it is not part of a full fork action. Here is a concise definition:

Definition 5.10. A *core* of a presheaf $U \in \widehat{\mathcal{C}}$ is an element of dimension > 1 which is not the image of any element of higher dimension.

Our definition of the causal relation will rely on the preliminary notions of sources and targets of a core, and that of channels created by a core. These notions will fix the direction of the causal relation.

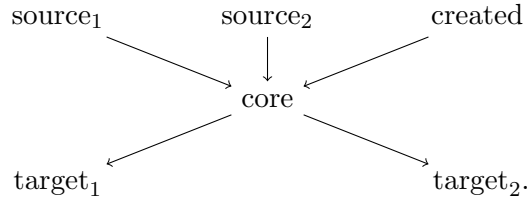
Definition 5.11. For any U and core $\mu \in U(c)$,

- the *sources* of μ are the agents x such that $x = \mu \cdot f \cdot s$ for some f ;
- the *targets* of μ are the agents y such that $y = \mu \cdot f \cdot t$ for some f ;
- a channel $a \in U(\star)$ is *created* by μ iff μ has the shape ν_n or $\iota_{n,i}$, and $a = \mu \cdot s \cdot s_{n+1}$.

Example 5.12. In the representable π_n , there is one target, $l \circ t$ (or equivalently $r \circ t$), and two sources, $s_1 = l \circ s$ and $s_2 = r \circ s$. Another example is $\tau_{n,i,m,j,k}$, which has two targets, $\epsilon \circ t$ and $\rho \circ t$, and two sources. However, $\rho \circ s \circ s_{n+1}$ is not created by the input element ρ , because it is not a core.

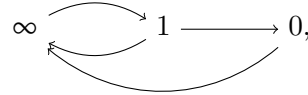
Definition 5.13. For any U , let its *causal graph* G_U have:

- as vertices, all channels, agents, and cores in U ,
- for all $x \in U[n]$ and $i \in n$, an edge $x \rightarrow x \cdot s_i$,
- and, for each core μ , an edge from each of its sources and created channels, and one into each of its targets, as in



Please note that edges $a \rightarrow \mu$ from a channel to an input action exist only if the involved action is not part of a synchronisation; for otherwise the synchronisation is a core, not the input.

The obtained graph is actually a binary relation, since there is at most one edge between any two vertices. It is also a colored graph, in the sense that it comes equipped with a morphism to the graph L :



mapping cores to ∞ , agents to 1, and channels to 0. (In particular, there are no edges from channels to agents or from cores to channels.) For any graph G , equipped with a morphism $l: G \rightarrow L$, we call vertices of G channels, agents, or cores, according to their label.

As expected, we have:

Proposition 5.14. *For any trace U , G_U is acyclic (in the directed sense).*

Proof. By induction on any decomposition of U . □

Let us now consider local 1-injectivity, linearity, initiality and finality. First, let us emphasise that for all seeds $Y \hookrightarrow M \leftarrow X$, M is a representable presheaf, so, e.g., it makes sense to consider $U(M)$.

Definition 5.15. A presheaf U is *locally 1-injective* iff for any seed $Y \hookrightarrow M \leftarrow X$ with interface I and core $\mu \in U(M)$, if two distinct elements of M are identified by the Yoneda morphism $\mu: M \rightarrow U$, then they are in (the image of) $I(\star)$.

This is equivalent to requiring that all morphisms $y_c \rightarrow U$, for all $c \in \mathbb{C}$, are 1-injective, and that for all core inputs and channel creations x of arity n in U , for all $i \in n$, we have

$$x \cdot s \cdot s_{n+1} \neq x \cdot s \cdot s_i.$$

Proposition 5.16. *Any trace U is locally 1-injective.*

Proof. Choose a decomposition of U into actions; μ corresponds to precisely one such action, say M' , obtained, by definition, from some seed M as a pushout (3.1). By construction of pushouts in presheaf categories, M' is obtained from M by identifying some channels according to $I \rightarrow Z$. \square

Let us observe that, because local 1-injectivity is only about cores, an input which is part of a synchronisation may receive an already known channel, even if its $n + 1$ th channel is not part of its interface — because it is not a core.

After local 1-injectivity, let us consider linearity.

Definition 5.17. Any $G \in \mathbf{Gph}/L$ is *source-linear* iff for any cores μ, μ' , and other vertex (necessarily an agent or a channel) x , $\mu \leftarrow x \rightarrow \mu'$ in G , then $\mu = \mu'$; G is *target-linear* iff for any cores μ, μ' and agent x , if $\mu \rightarrow x \leftarrow \mu'$ in G , then $\mu = \mu'$; G is *linear* iff it is both source-linear and target-linear.

Proposition 5.18. For any trace $Y \xrightarrow{s} U \xleftarrow{t} X$, G_U is linear.

Proof. Straightforward, by induction on any decomposition of U into actions, observing that we glue along agents and channels which are initial on one side and final on the other. \square

The last of our necessary sanity checks is about initiality and finality. The idea here is that one may read in U alone what both legs of the cospan $Y \hookrightarrow U \leftarrow X$ should be.

Definition 5.19. An agent is *initial* in U when it is not the source of any action, i.e., for no action $\mu \in U$, $x = \mu \cdot s$. A channel is initial when it is not created by any core.

An agent x in U is *final* iff it is not the target of any action, i.e., for no action $\mu \in U$, $x = \mu \cdot t$. All channels are final.

Lemma 5.20. An agent is initial in U iff it has no edge to any core in G_U .

Lemma 5.21. An agent is final in U iff it has no edge from any core in G_U .

Now, here is the expected characterisation:

Theorem 5.22. A monic cospan $Y \xrightarrow{s} U \xleftarrow{t} X$ of finite presheaves is a trace iff

- (C1) U is locally 1-injective,
- (C2) X contains exactly the initial agents and channels in U ,
- (C3) Y contains exactly the final agents and channels in U ,
- (C4) and G_U is linear and acyclic.

Of course, we have almost proved the ‘only if’ direction, and the rest is easy, so only the ‘if’ direction remains to prove. The rest of this section is devoted to this. So given a cospan satisfying the above conditions, we intend to sequentialise it, i.e., decompose it into actions. We will proceed by induction on the number of cores in U , by picking a core μ which is maximal according to G_U , removing it from U and applying the induction hypothesis to the rest. However, it may not be obvious how we should remove μ from U . E.g., the topos-theoretic difference $U \setminus \mu$ does not yield the expected result, as it removes all sources of μ . Instead, we consider the following operation: for any morphism of presheaves $f: U \rightarrow V$ and set W , let $U - W = \sum_{c \in \mathbb{C}} \text{Im}(U(c)) \setminus W \subseteq \sum_{c \in \mathbb{C}} V(c)$. This is a slight abuse of notation, as f is implicit, but it should be easily inferred from context.

Remark 5.23. We observe that $U - W$ is generally just a set, not a presheaf; i.e., its elements are not necessarily stable under the action of morphisms in \mathbb{C} . Consider for example $U = [1] \parallel [1]$ and let W consist of the first agent and the unique channel. Then $U - W$ does not contain the unique channel of U , so the action of s_1 on the second agent steps outside $U - W$.

But there is one useful case where $U - W$ is indeed a subpresheaf of U , as we show below in Lemma 5.25.

Definition 5.24. For any seed $Y \hookrightarrow M \leftarrow X$, let the *past* $\text{past}(M) = M - Y$ of M be the set of its elements not in the image of Y . For any such M , presheaf U , and core $\mu \in U(M)$, let $\text{past}(\mu) = \text{Im}(\text{past}(M))$ consist of all images of $\text{past}(M)$.

To explain the statement a bit more, by Yoneda, we see μ as a map $M \rightarrow U$, so we have a set-function

$$\text{past}(M) \hookrightarrow \text{ob}(\text{el}(M)) \rightarrow \text{ob}(\text{el}(U))$$

(recalling that el denotes the category of elements). We observe that $\text{past}(\mu)$ is always a set of agents and actions only, since channels present in X always are in Y too.

Given a core $\mu \in U$, the relevant way of removing μ from U will be:

$$U \setminus \mu = \bigcup \{V \hookrightarrow U \mid \text{ob}(\text{el}(V)) \cap \text{past}(\mu) = \emptyset\}.$$

$U \setminus \mu$ is thus the largest subpresheaf of U not containing any element of the past of μ . The good property of this operation is:

Lemma 5.25. *If μ is a maximal core in G_U (i.e., there is no path to any further core) and G_U is target-linear, then $(U \setminus \mu)(c) = U(c) \setminus \text{past}(\mu)$ for all c .*

Proof. The direction $(U \setminus \mu)(c) \subseteq U(c) \setminus \text{past}(\mu)$ is by definition of \setminus . Conversely, it is enough to show that $c \mapsto U(c) \setminus \text{past}(\mu)$ forms a subpresheaf of U , i.e., that for any $f: c \rightarrow c'$ in \mathbb{C} , and $x \in U(c') \setminus \text{past}(\mu)$, $x \cdot f \notin \text{past}(\mu)$. Assume on the contrary that $x' = x \cdot f \in \text{past}(\mu)$. Then, of course f cannot be the identity, and w.l.o.g. we may assume that x' is an agent and x is a core. But then, because $x' \in \text{past}(\mu)$, there is an edge $\mu \rightarrow x'$ in G_U , and because $x' = x \cdot f$, there is an edge $x \rightarrow x'$ or $x' \rightarrow x$ in G_U . The former case is impossible by target-linearity, and the latter case would imply the existence of a path $\mu \rightarrow x$ in G_U , which contradicts the maximality of μ . So $x' \in \text{past}(\mu)$ is impossible altogether. \square

Proof of Theorem 5.22. We proceed by induction on the number of actions in U . If it is zero, then U is a position; by (C2), t is an iso, and by (C3) so is s , hence the cospan is a trace. For the induction step, we first decompose U into

$$Y \xrightarrow{s_2} U' \xleftarrow{t_2} Z \xrightarrow{s_1} M' \xleftarrow{t_1} X,$$

and then show that M' is an action and U' satisfies the conditions of the theorem.

First, by acyclicity, pick a maximal core μ in G_U , i.e., one with no path to any other core. Let

$$\begin{array}{ccccc} & & I_0 & & \\ & \swarrow & \downarrow & \searrow & \\ Y_0 & \longrightarrow & M_0 & \longleftarrow & X_0 \end{array}$$

be the seed with interface corresponding to μ , so we have the Yoneda morphism $\mu: M_0 \rightarrow U$.

Let $U' = (U \setminus \mu)$, and $X_1 = X - \text{Agents}(X_0)$. X_1 is a subpresheaf of X , since it contains all channels. The square

$$\begin{array}{ccc} I_0 & \longrightarrow & X_1 \\ \downarrow & \lrcorner & \downarrow \\ X_0 & \longrightarrow & X \end{array}$$

is a pushout, since it just adds the missing agents to X_1 . Define now Z , M' , s_1 , and t_1 by the pushouts

$$\begin{array}{ccccc} & & Y_0 & \longrightarrow & Z \\ & \nearrow & \downarrow & \lrcorner & \downarrow s_1 \\ & & M_0 & \longrightarrow & M' \\ I_0 & \nearrow & \uparrow & \lrcorner & \downarrow t_1 \\ & & X_0 & \longrightarrow & X \\ & \searrow & & \lrcorner & \downarrow \\ & & & & U \end{array}$$

and the induced arrows. We further obtain arrows to U by universal property of pushout.

Let us show that the arrow $f: M' \rightarrow U$ is mono. First, it is obviously mono in dimensions > 1 . It is also mono in dimension 1, because $M'[n] = X[n] + Y_0[n]$ for all n and $X \rightarrow U$ is mono with image consisting only of initial agents, which are thus disjoint from the image of Y_0 . Finally, for dimension 0, i.e., at \star , the pushout defining M' is isomorphic to

$$\begin{array}{ccc} I_0(\star) = X_0(\star) & \longrightarrow & X_1(\star) = X(\star) \\ \text{inj}_1 \downarrow & & \downarrow \text{inj}_1 \\ M_0(\star) = X_0(\star) + I & \longrightarrow & M'(\star) = X(\star) + I \end{array}$$

where $I = M_0(\star) \setminus X_0(\star)$ is the set of channels created by the action. Consider any $a, b \in M'(\star)$ such that $a \neq b$. Because $X \rightarrow U$ is mono, if $a, b \in X(\star)$ then $f(a) \neq f(b)$. By local 1-injectivity of U , if $a, b \in I$ then $f(a) \neq f(b)$. Finally, if $a \in X(\star)$ and $b \in I$, then we have an edge $f(b) \rightarrow \mu$ in G_U , whereas $f(a)$ is initial by (C2), so $f(a) \neq f(b)$. This shows that $M' \rightarrow U$ is mono, which also entails that $Z \rightarrow U$ is a mono, because s_1 is a pushout of the mono $Y_0 \rightarrow M_0$.

By (C1) and Lemma 5.25, $U = M' \cup U'$, i.e., the square

$$\begin{array}{ccc} Z & \longrightarrow & U' \\ \downarrow & \lrcorner & \downarrow \\ M' & \longrightarrow & U \end{array}$$

is a pushout, so U is indeed a composite as claimed, with $Z \hookrightarrow M' \hookleftarrow X$ an action by construction. So, it remains to prove that $Y \hookrightarrow U' \hookleftarrow Z$ satisfies the conditions. First, as a subpresheaf of U (whose inclusion preserves cores), U' is locally 1-injective and has a linear and acyclic causal graph, so satisfies (C1) and (C4). U' furthermore satisfies (C2) by

construction of Z and source-linearity of G_U , and (C3) because removing $\text{past}(\mu)$ cannot make any non-final agent final. \square

Let us conclude this section with a helpful lemma, whose proof relies on Theorem 5.22:

Lemma 5.26. *There is at most one cell filling any diagram*

$$\begin{array}{ccc} Y' & \xrightarrow{k} & Y \\ u' \downarrow & & \downarrow u \\ X' & \xrightarrow{h} & X \end{array}$$

in \mathbb{D} .

In order to prove this, let us introduce:

Definition 5.27. For any action $x \in U$, let $\text{core}(x)$, the *core associated to x* , be the unique core $\mu \in U$ for which there exists f in \mathbb{C} such that $\mu \cdot f = x$. If x is an agent or a channel, then by definition $\text{core}(x) = x$.

Proof of Lemma 5.26. By definition, we have cospans $Y' \xrightarrow{s'} u' \xleftarrow{t'} X'$ and $Y \xrightarrow{s} u \xleftarrow{t} X$. Suppose we are given $l, l': u' \rightarrow u$ making (k, l, h) and (k, l', h) into cells. By naturality, l and l' are determined by their images on channels, agents, and cores. We show by induction on the ordering induced by $G_{u'}$ that they have to agree on these. For the base case: they have to agree with h on initial agents and channels by definition of cells. For the induction step, we proceed by case analysis on the kind of element to consider. The image of any source of or channel created by a core μ is uniquely determined by naturality, which leaves the case of a core μ , of which we assume that there is an agent x such that $\mu \rightarrow x$ in $G_{u'}$ and $l(x) = l'(x)$. The edge $\mu \rightarrow x$ yields a morphism, say t , in \mathbb{C} such that $\mu \cdot t = x$. But then by naturality we have $l(\mu) \cdot t = l(x) = l'(x) = l'(\mu) \cdot t$. By linearity of G_u we have $\text{core}(l(\mu)) = \text{core}(l'(\mu))$. Now let c_μ denote the object of \mathbb{C} over which μ lies, and let c' be the one over which $\text{core}(l(\mu))$ lies. By inspection of \mathbb{C} , there is exactly one morphism $f: c_\mu \rightarrow c'$, and so we have $l(\mu) = \text{core}(l(\mu)) \cdot f = \text{core}(l'(\mu)) \cdot f = l'(\mu)$, as desired. \square

5.3. A playground. In this section, we finally prove:

Theorem 5.28. \mathbb{D} forms a playground.

Most axioms have been proved in previous sections, and we are left with both decomposition axioms, which are proved in Lemmas 5.29 and 5.30 below, relying on the correctness criterion of the previous section.

Lemma 5.29. (P7) *Any double cell as in the center below, where B is a basic action and M is an action, decomposes in exactly one of the forms on the left and right:*

$$\begin{array}{ccc} \begin{array}{ccc} A & \xrightarrow{\quad} & X \\ \downarrow \nearrow \alpha_1 & & \downarrow \\ C & \xrightarrow{\quad} & Y \\ \downarrow \nearrow \alpha_2 & & \downarrow \\ D & \xrightarrow{\quad} & Z \end{array} & \sim & \begin{array}{ccc} A & \xrightarrow{h} & X \\ U \downarrow \nearrow \alpha & & \downarrow V \\ C & & Y \\ B \downarrow & & \downarrow M \\ D & \xrightarrow{k} & Z \end{array} & \sim & \begin{array}{ccc} A & \xrightarrow{\quad} & X \\ \downarrow \nearrow \alpha_1 & & \downarrow \\ C & & Y \\ \downarrow & \nearrow \alpha_2 & \downarrow \\ D & \xrightarrow{\quad} & Z \end{array} \end{array}$$

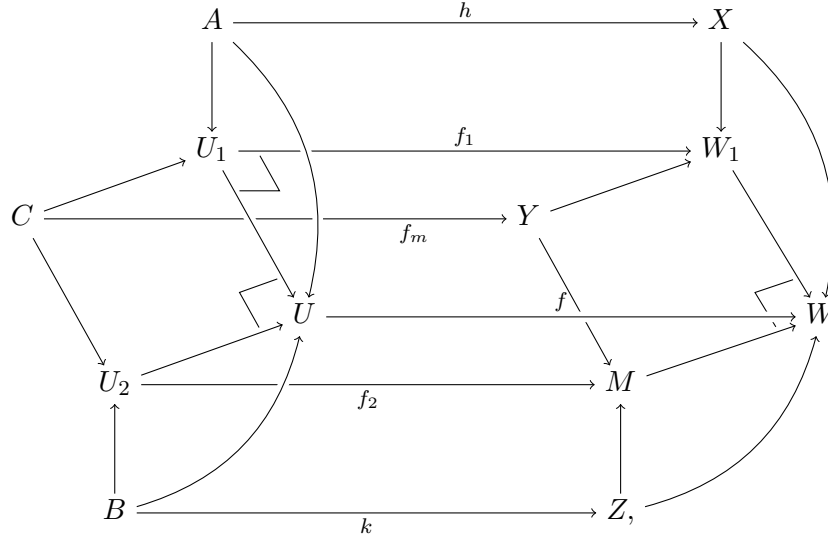


Figure 6: Proof of Lemma 5.30

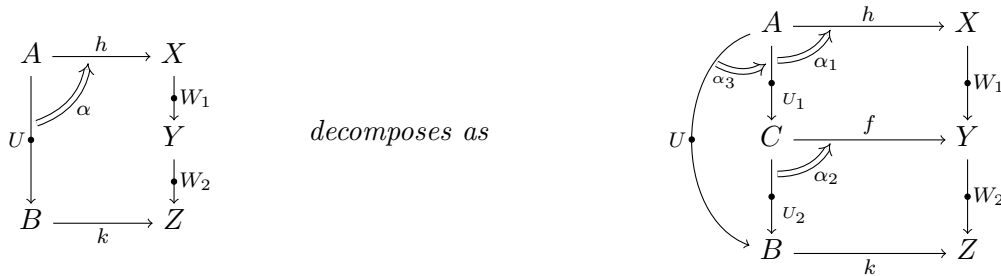
Proof. For any element a over $c \in \mathbb{C}$ of any presheaf $F \in \widehat{\mathbb{C}}$, let its *neighbourhood* consist of all elements in the image of $a: c \rightarrow F$.

Let $b \in B$ and $m \in M$ be the unique cores of B and M , respectively. Let V_m be the neighbourhood of m in $M \bullet V$.

If $\alpha(b) \in V_m$, let us show that the whole of U is mapped to V , and we are in the left-hand case. It is clear for channels. If there exists an element x of U of dimension ≥ 1 mapped to y in $M - V$, i.e., $M - Y$, then we obtain a path $x \rightarrow x'$ to an agent x' of C , in $G_{B \bullet U}$. Via α , this yields a path $M - Y \rightarrow Y$ in $G_{M \bullet V}$ between elements of dimension ≥ 1 , a contradiction.

If now $\alpha(b) \notin V_m$, we show similarly that the whole of $B \bullet U$ is mapped to V , because the contrary would imply the existence of a path $M - Y \rightarrow V - Y$ in $G_{M \bullet V}$, which also is a contradiction. Hence, we are in the right-hand case. \square

Lemma 5.30. (P6) *Any double cell*



with α_3 an isomorphism, in an essentially unique way.

Proof. Let $\alpha = (h, f, k)$. We first treat the case where W_2 is an action M , recalling Definition 5.27. We construct U_1 and U_2 , as depicted in Figure 6. First, let $U_1 = U \times_W W_1$, where $W = W_2 \bullet W_1$, and let $A \rightarrow U_1$ denote the induced arrow. By construction, all of $A \rightarrow U_1 \rightarrow U$ are monos and, by Lemma 3.20 and the pullback lemma, $A = U_1 \times_{W_1} X$.

Let us show that the projection $f_1: U_1 \rightarrow W_1$ preserves initiality of channels and agents. We proceed by contrapositive: consider any channel or agent $x \in U_1$. If $x' = f_1(x)$ is not

initial in W_1 , then we have an edge $x' \rightarrow \mu'$ for some core μ' of W_1 . But, since f is a morphism between traces, it preserves initiality, so x cannot be initial in U , hence we find $x \rightarrow \mu$ in G_U . By source linearity of G_W , $\text{core}(f(\mu)) = \mu'$, so the action $f(\mu) \in W$ has antecedents both in U and W_1 . By universal property of pullback, there exists an action $y \in U_1$, respectively mapped to μ and $f(\mu)$, which by naturality and injectivity of $U_1 \rightarrow U$ entails that $x \rightarrow y$ in G_{U_1} . Therefore, x is not initial in G_{U_1} , as required.

Let now $C \hookrightarrow U_1$ denote the subpresheaf of U_1 consisting of initial channels and agents (a subpresheaf because if x is an initial, n -ary agent, then $x \cdot s_i$ is an initial channel for any $i \in n$, by Theorem 5.22 and Lemma 5.21). Since f_1 preserves initiality, $C \rightarrow U_1 \rightarrow W_1$ factors through $Y \rightarrow W_1$, uniquely since the latter is mono, say as f_m (see Figure 6). By Theorem 5.22, $A \rightarrow U_1 \leftarrow C$ is a trace and (h, f_1, f_m) defines a morphism to $X \rightarrow W_1 \leftarrow Y$.

Let then $U_2 \hookrightarrow U$ denote the subpresheaf of U consisting of elements below C in G_U , i.e.,

$$x \in U_2 \Leftrightarrow \exists c \in C. c \rightarrow^*_{G_U} \text{core}(x).$$

A first observation is that all initial channels and agents of U are in U_2 , so that $B \rightarrow U$ factors through U_2 . Indeed, consider any such initial x . By acyclicity of G_U , each initial element is reachable from some final element, so x is reachable from some final y . But by source-linearity the corresponding path $y \rightarrow^* x$ goes through C , so we find a path $c \rightarrow^* x$ for some $c \in C$, as desired.

Now, because $U_2 \rightarrow U$ and $M \rightarrow W$ are monos, showing that f maps all elements x of U_2 to M will imply that $U_2 \rightarrow U \rightarrow W$ uniquely factors through $M \rightarrow W$. Let us do this by case analysis:

- If x is not a channel, then f preserves paths from agents to x , so we find some path $f(c) \rightarrow^*_{G_W} \text{core}(f(x))$ with $c \in C$ hence $f(c) \in Y$, which implies that $f(x) \in M$ ($f(x) \in W_1 - M$ would contradict initiality of Y in W_1).
- If x is some channel initial in U , then since f preserves initiality x is mapped to Z hence to M .
- If finally x is some non-initial channel, then $x \rightarrow \mu$ for some core $\mu \in U$. Now $\mu \in U_2$, as witnessed by the path $c \rightarrow^* x \rightarrow \mu$. But then $x = \mu \cdot u$ for some morphism u of \mathbb{C} , so since by the above $f(\mu) \in M$, we have that $f(x) = f(\mu) \cdot u$ is in M too, as desired.

We thus get a diagram as in Figure 6, which commutes because $M \rightarrow W$ is mono.

By Theorem 5.22, $C \rightarrow U_2 \leftarrow B$ is a trace, and $U = U_2 \bullet U_1$, which shows existence of the desired decomposition.

For any decomposition as in Figure 6, we have $C = U_2 \times_M Y$ by Lemma 3.20, so by Corollary 4.20, we also have $U_1 = U \times_W W_1$. Thus, U_1 is uniquely determined up to canonical isomorphism. But by Theorem 5.22, $C \rightarrow U_1$ is so too, as the subobject of initial agents and channels. But then U_2 precisely consists of elements below C . Indeed, by finiteness of G_{U_2} and (C3) in Theorem 5.22, all of U_2 clearly lies below C . Conversely, for any $x \in U_1 - U_2$, by finiteness of G_{U_1} and (C3) in Theorem 5.22, we have a path $x \rightarrow^+ c$ to some $c \in C$, so x cannot lie below C by (C4). Our decomposition is thus unique up to canonical isomorphism. \square

6. A SHEAF MODEL

In the previous sections, we have constructed a double category \mathbb{D} and equipped it with playground structure. We now instantiate constructions from [39] on \mathbb{D} , which lead to the definition of our sheaf model for π . We first recall various notions of strategy in Section 6.1:

naive strategies, innocent strategies, and behaviours. Behaviours are further studied in Section 6.2, where we introduce a kind of calculus for them. Using this calculus, we then define our interpretation of π in Section 6.3. Finally, in Section 6.4, we state our semantic definition of fair testing equivalence and our main result.

6.1. Strategies and behaviours. We first recall notions of strategies. As announced in the introduction, we define a category $\mathbb{T}(X)$ combining prefix ordering and isomorphism of traces: $\mathbb{T}(X)$ has traces $u: Y \rightarrow X$ as objects, and as morphisms $u \rightarrow u'$ all pairs (w, α) with $w: Y' \rightarrow Y$ and α an isomorphism $u \bullet w \rightarrow u'$ in the hom-category $\mathbb{D}_v(Y', X)$, as in

$$\begin{array}{ccc} & Y' & \\ w \swarrow & & \searrow \\ Y & \xrightarrow[\alpha]{\cong} & u' \\ u \swarrow & & \searrow \\ & X & \end{array}$$

considered up to the smallest equivalence relation identifying (w, α) and $(w', \alpha \circ (u \bullet \gamma))$, for any $w': Y' \rightarrow Y$ and special $\gamma: w' \rightarrow w$. Thus, u' is an extension of u by w .

Definition 6.1. Let the category of (*naive*) strategies on X be $\widetilde{\mathbb{T}(X)}$.

Strategies do not yield a satisfactory model for π :

Example 6.2. Consider the position X with three agents x, y, z sharing a channel a , and the following traces on it: in $u_{x,y}$, x sends a on a , and y receives it; in $u_{x,z}$, x sends a on a , and z receives it; in i_z , z inputs on a . One may define a strategy S mapping $u_{x,y}$ and i_z to a singleton, and $u_{x,z}$ to \emptyset . Because $u_{x,y}$ is accepted, x accepts to send a on a ; and because i_z is accepted, z accepts to input on a . The problem is that S rejecting $u_{x,z}$ roughly amounts to x refusing to synchronise with z , or conversely.

We want to rule out this kind of strategy from our model, by adapting the idea of innocence. We start by extending $\mathbb{T}(X)$ with objects representing traces on subpositions of X . For this, we consider the following category \mathbb{T}_X . It has as objects pairs (u, h) of a trace $u: Z \rightarrow Y$ and a morphism $h: Y \rightarrow X$ in \mathbb{D}_h . A morphism $(u, h) \rightarrow (u', h')$ consists of a trace $w: T \rightarrow Z$ and a cell as below left with $h'r = h$. Morphisms are considered up to the smallest equivalence relation identifying (w, α) with $(w', \alpha \circ (u \bullet \gamma))$, for any w' and γ as below right.

$$\begin{array}{ccc} T & \xrightarrow{s} & Z' \\ w \downarrow & & \downarrow \\ Z & \xrightarrow{\alpha} & u' \\ u \downarrow & & \downarrow \\ Y & \xrightarrow{r} & Y' \\ h \searrow & & \swarrow h' \\ & X & \end{array} \qquad \begin{array}{ccc} T' & \rightarrow & T & \xrightarrow{s} & Z' \\ w' \searrow & \xrightarrow{\gamma} & \downarrow w & & \downarrow \\ & & Z & \xrightarrow{\alpha} & u' \\ u \downarrow & & \downarrow & & \downarrow \\ Y & \xrightarrow{r} & Y' & & \\ h \searrow & & \swarrow h' & & \\ & X & & & \end{array}$$

Example 6.3. Recalling the right-hand trace of Figure 4 (page 22), say $u: Y \rightarrow X$, y 's first action is an input on its unique channel b . This yields a trace $\iota_{1,1}: [2] \rightarrow [1]$. Here is an example morphism $(\iota_{1,1}, y) \rightarrow (u, id_X)$ in \mathbb{T}_X :

$$\begin{array}{ccc}
 [3] & \xrightarrow{y''} & Y \\
 \iota_{2,2} \downarrow & \Longrightarrow & \downarrow u \\
 [2] & & \\
 \iota_{1,1} \downarrow & \xrightarrow{y} & \downarrow \\
 [1] & \xrightarrow{y} & X. \\
 & \searrow y & \cong \\
 & & X.
 \end{array}$$

We think of it as an occurrence of the trace $\iota_{1,1}$ in u . Thus, morphisms in \mathbb{T}_X account both for prefix inclusion and for ‘spatial’ inclusion, i.e., inclusion of a trace into some other trace on a larger position.

We now define *views* within \mathbb{T}_X :

Definition 6.4. A *view* is a trace isomorphic to some (possibly empty) composite of basic actions (Definition 5.1). Let \mathbb{V}_X denote the full subcategory of \mathbb{T}_X spanning pairs (u, h) where u is a view.

Intuitively, basic actions follow exactly one agent through an action. An object of \mathbb{V}_X consists of a view, say $v: [n'] \dashrightarrow [n]$, plus a morphism $h: [n] \rightarrow X$ in \mathbb{D}_h , which by Yoneda is just an agent of X . So an object of \mathbb{V}_X is just an agent of X and a view from it.

Definition 6.5. The inclusion $j_X: \mathbb{V}_X \hookrightarrow \mathbb{T}_X$ induces a Grothendieck topology, for which a sieve $((u_i, h_i) \xrightarrow{(w_i, \alpha_i)} (u, h))_{i \in I}$ of morphisms to some trace u is *covering* iff it contains all morphisms from views into u . Let the category $\mathbb{S}_X \hookrightarrow \widetilde{\mathbb{T}}_X$ of *innocent* strategies be the category of sheaves of finite sets for this topology. Let the category \mathbb{B}_X of *behaviours* over X be $\widetilde{\mathbb{V}}_X$.

As announced in the introduction, letting $\text{ran}_{j_X}^{op}$ denote *right Kan extension* [53] along the inclusion $j_X^{op}: \mathbb{V}_X^{op} \hookrightarrow \mathbb{T}_X^{op}$, we have:

Proposition 6.6. *The embedding $\text{ran}_{j_X}^{op}$ induces an equivalence of categories $\mathbb{B}_X \simeq \mathbb{S}_X$.*

Proof. See [39, Lemma 4.34]. □

We thus obtain the innocent strategy S_B associated to a behaviour $B \in \mathbb{B}_X$ by taking its right Kan extension as in

$$\begin{array}{ccccc}
 \mathbb{V}_X^{op} & \xrightarrow{j_X^{op}} & \mathbb{T}_X^{op} & \xleftarrow{k_X^{op}} & \mathbb{T}(X)^{op} \\
 & \searrow B & \downarrow S_B & & \swarrow \mathcal{U}(S_B) \\
 & & \text{set.} & &
 \end{array}$$

Explicitly, using standard results, we obtain the *end*

$$S_B(u, h) = \int_{(v,x) \in \mathbb{V}_X} B(v, x)^{\mathbb{T}_X((v,x), (u,h))},$$

which is a kind of generalised product. In the boolean setting (functors to 2), this end reduces to the conjunction $\bigwedge_{\{(v,x) \in \mathbb{V}_X \mid \exists \alpha: (v,x) \rightarrow (u,h)\}} B(v, x)$, demanding precisely that all views of u are accepted by B . In the general case, the intuition is that a way of accepting u for S_B is a compatible family of ways of accepting the views of u for B . The forgetful functor \mathcal{U} to naive strategies is then given by restricting along the opposite of $k_X: \mathbb{T}(X) \hookrightarrow \mathbb{T}_X$ as above. Some local information may be forgotten by \mathcal{U} , which is hence neither injective on objects, nor full, nor faithful. E.g., if two behaviours differ on one agent, but are both empty on the views of another, then both are mapped to the empty naive strategy.

Example 6.7. Recalling X and S from Example 6.2, let us show that for any $B \in \mathbb{B}_X$, the associated strategy $\mathcal{U}(S_B) \in \overline{\mathbb{T}(X)}$ cannot be S . Indeed, if $\mathcal{U}(S_B)$ was S , then because S accepts $u_{x,y}$ and i_z , B has to accept the following views: (1) i_z , (2) o_x , in which x sends a on a (without any matching input), (3) i_y , in which y inputs on a , and (4) all identity views on x , y , and z . But then $\mathcal{U}(S_B)$ has to accept both $u_{x,y}$ and $u_{x,z}$, because B accepts all views mapping into them.

6.2. Decomposing behaviours. In this section, we study behaviours a bit more in depth, which yields the calculus announced at the beginning of Section 6. The starting point is that the assignment $X \mapsto \mathbb{B}_X$ may be equipped with useful structure, describing how a behaviour B on some given position restricts to any subposition, and also what remains of it after a given action has been played. Otherwise said, morphisms of \mathbb{D} act contravariantly on behaviours:

- horizontal morphisms $h: X \rightarrow X'$ induce functors $\mathbb{B}_h: \mathbb{B}_{X'} \rightarrow \mathbb{B}_X$, and
- vertical morphisms $u: Y \twoheadrightarrow X$ induce functors $\mathbb{B}_u: \mathbb{B}_X \rightarrow \mathbb{B}_Y$.

Furthermore, any cell as below left induces a natural isomorphism as below right:

$$\begin{array}{ccc}
 Y & \xrightarrow{k} & Y' \\
 \downarrow u & \nearrow \alpha & \downarrow u' \\
 X & \xrightarrow{h} & X'
 \end{array}
 \qquad
 \begin{array}{ccc}
 \mathbb{B}_Y & \xleftarrow{\mathbb{B}_k} & \mathbb{B}_{Y'} \\
 \mathbb{B}_u \uparrow \cong & & \uparrow \mathbb{B}_{u'} \\
 \mathbb{B}_X & \xleftarrow{\mathbb{B}_h} & \mathbb{B}_{X'}
 \end{array}$$

which notably says that $B \cdot u' \cdot k \cong B \cdot h \cdot u$ for any behaviour $B \in \mathbb{B}_{X'}$. This is worked out in detail and in full generality in [39], and extended to a pseudo double functor $\mathbb{D}^{op} \rightarrow \mathbb{Q}\text{Cat}$, where $\mathbb{Q}\text{Cat}$ denotes Ehresmann's double category of *quintets* over Cat . But let us explain how both actions look like in the present, concrete case.

The first, horizontal action is really easy: any horizontal morphism $k: X' \rightarrow X$ acts on a given behaviour $B \in \mathbb{B}_X$ by returning the behaviour $B \cdot k$ such that for all $(v, h) \in \mathbb{V}_{X'}$, $(B \cdot k)(v, h) = B(v, k \circ h)$.

Proposition 6.8. *The functor $\mathbb{B}_X \rightarrow \prod_{n,x:[n] \rightarrow X} \mathbb{B}_{[n]}$ given at (n, x) by horizontal action of x , i.e., $B \mapsto B \cdot x$, is an isomorphism.*

Proof. We have $\mathbb{V}_X \cong \sum_{n,x:[n] \rightarrow X} \mathbb{V}_{[n]}$. □

Notation 6.9. If $(B_x)_{x:[n] \rightarrow X}$ is a family of behaviours indexed by the agents of X , we accordingly denote its unique antecedent by $[B_x]_{x:[n] \rightarrow X}$.

Vertical action is a bit harder. Let us start by recalling the following result from [39], which generalises Lemma 5.5:

Lemma 6.10. *For any trace $P: Y \twoheadrightarrow X$ and $y: d \rightarrow Y$ in \mathbb{D}_h with $d \in \mathbb{I}$, there exists an essentially unique cell*

$$\begin{array}{ccc}
 d & \xrightarrow{y} & Y \\
 v^{y,P} \downarrow & \cong \square \square \square \triangleright & \downarrow P \\
 d^{y,P} & \dashrightarrow_{y^P} & X,
 \end{array}$$

with $v^{y,P}$ a view.

Proof. By induction on P using Lemma 5.5. See [39] for the precise meaning of essential uniqueness in this case. □

This result allows us to describe what remains of a behaviour after a trace:

Definition 6.11. Given any behaviour $B \in \mathbf{B}_X$ and trace $P: Y \dashrightarrow X$, the *residual* $B \cdot P$ of B along P is the behaviour determined (up to isomorphism) by

$$(B \cdot P \cdot y)(w, id_{[n_y]}) = B(v^{y,P} \bullet w, y^P)$$

for all agents $y: [n_y] \rightarrow Y$.

So we may consider residuals of behaviours along arbitrary traces. Conversely, any behaviour is determined by its initial states and residuals. Let us first consider the following special kind of behaviour:

Definition 6.12. A behaviour B on $[n]$ is *definite* iff $B(id^\bullet, id_{[n]}) \cong 1$. A behaviour B on an arbitrary position X is *definite* iff for all agents x of X , $B \cdot x$ is definite. Let \mathbf{D}_X denote the full subcategory of \mathbf{B}_X spanning definite behaviours.

This means that B has exactly one initial state.

Definition 6.13. For any $n \in \mathbb{N}$ and family $B_b \in \mathbf{B}_{[n_b]}$ indexed by all basic actions $b: [n_b] \dashrightarrow [n]$, let $\langle B_b \rangle_b$ denote the definite behaviour B determined by $B \cdot b = B_b$ for all b .

By construction, we have:

Proposition 6.14. For any definite behaviour $D \in \mathbf{B}_{[n]}$, $D \cong \langle D \cdot b \rangle_b$.

This extends to arbitrary behaviours on individuals using the fact that any behaviour on an individual is a coproduct of definite behaviours.

Notation 6.15. For any behaviour $B = \sum_{k \in \gamma} D_k$ on any $[n]$, let $|B| = \gamma$ and, for any $k \in \gamma$, let the *restriction* $B|_k$ of B to k be D_k .

Remark 6.16. In other words, $k \in \gamma$ is just an element of $B(id^\bullet_{[n]}, id_{[n]})$ and $B|_\sigma$ is determined by

$$B|_\sigma(v, id_{[n]}) = \{\sigma' \in B(v, id_{[n]}) \mid \sigma' \cdot !_v = \sigma\},$$

where $!_v$ denotes the unique morphism $(id^\bullet_{[n]}, id_{[n]}) \rightarrow (v, id_{[n]})$ in $\mathbb{V}_{[n]}$.

We obtain:

Proposition 6.17. For any behaviour $B \in \mathbf{B}_{[n]}$,

$$B \cong \sum_{k \in |B|} \langle (B|_k) \cdot b \rangle_b.$$

Putting this together with spatial decomposition, we obtain:

Corollary 6.18. For any behaviour $B \in \mathbf{B}_X$,

$$B \cong \left[\sum_{k \in |B \cdot x|} \langle ((B \cdot x)|_k) \cdot b \rangle_{b: [n_b] \dashrightarrow [n_x]} \right]_{x: [n_x] \rightarrow X}.$$

The fact that both actions of \mathbb{D} yield a pseudo double functor $\mathbb{D}^{op} \rightarrow \mathbb{Q}\text{Cat}$ essentially boils down to:

Lemma 6.19. For any $B \in \mathbf{B}_X$ and cell

$$\begin{array}{ccc} Y' & \xrightarrow{k} & Y \\ u' \downarrow & \xrightarrow{\alpha} & \downarrow u \\ X' & \xrightarrow{h} & X, \end{array}$$

we have $D \cdot h \cdot u' \cong D \cdot u \cdot k$.

The result is stated (and proved) in full generality as [39, Proposition 4.31].

6.3. Interpretation of π . We now define our interpretation of π -calculus configurations. We start with processes and then cover configurations. Because the notation for behaviours introduced in the previous section only covers behaviours on representable positions, while π -calculus syntax is name-based, we bridge the gap by keeping track, along the recursive definition, of a bijection between the set γ of free channels of the considered process and its cardinal $|\gamma|$. So we define a family of maps in

$$\prod_{\gamma \in \mathcal{P}_f(\mathbb{N})} \mathbb{D}_{[|\gamma|]}^{P_i \gamma \times \text{Bij}(\gamma, |\gamma|)},$$

where $\text{Bij}(A, B)$ denotes the set of all bijections $A \cong B$. For any such γ, P , and h in the domain, we denote the result by $\llbracket P \rrbracket_h$, or $\llbracket P \rrbracket_{\gamma, h}$ when needed. Letting $n = |\gamma|$, it is coinductively defined by

$$\begin{aligned} \llbracket \sum_i P_i \rrbracket_h &= \boxplus_i \llbracket P_i \rrbracket_h \\ \llbracket P \mid Q \rrbracket_h &= \left\langle \begin{array}{l} \pi_n^l \mapsto \llbracket P \rrbracket_h \\ \pi_n^r \mapsto \llbracket Q \rrbracket_h \end{array} \right\rangle \\ \llbracket \nu b.P \rrbracket_h &= \left\langle \nu_n \mapsto \llbracket P \rrbracket_{h'} \right\rangle \\ \llbracket \heartsuit.P \rrbracket_h &= \left\langle \heartsuit_n \mapsto \llbracket P \rrbracket_h \right\rangle \\ \llbracket \tau.P \rrbracket_h &= \left\langle \tau_n \mapsto \llbracket P \rrbracket_h \right\rangle \\ \llbracket a(b).P \rrbracket_h &= \left\langle \iota_{n, h(a)} \mapsto \llbracket P \rrbracket_{h'} \right\rangle \\ \llbracket \bar{a}(b).P \rrbracket_h &= \left\langle o_{n, h(a), h(b)} \mapsto \llbracket P \rrbracket_h \right\rangle \end{aligned}$$

where

- in any list $\langle b_1 \mapsto B_1, \dots, b_m \mapsto B_m \rangle$, all unmentioned basic actions are meant to be mapped to the empty behaviour;
- the *definite sum* $\boxplus_i D_i$ of definite behaviours D_i is the definite behaviour determined by $(\boxplus_i D_i) \cdot b = \sum_i (D_i \cdot b)$, for all basic actions $b: [n'] \rightarrow [n]$;
- and $h': \gamma, b \cong n+1$ maps any $a \in \gamma$ to $h(a)$, and b to $n+1$.

Example 6.20. Let us briefly illustrate the translation. Consider any $h: \gamma \rightarrow |\gamma|$ and processes $\gamma, b \vdash P$, $\gamma, b \vdash Q$, and $\gamma \vdash R$, with $a \in \gamma$. We can form $a(b).P + a(b).Q + \bar{a}(a).R$, which is mapped to

$$\langle \iota_{|\gamma|, h(a)} \mapsto (\llbracket P \rrbracket_{h'} + \llbracket Q \rrbracket_{h'}), o_{\gamma, h(a), h(a)} \mapsto \llbracket R \rrbracket_h \rangle.$$

To emphasise the difference, using coproduct of behaviours instead of definite sum in the translation would yield a behaviour with three distinct initial states, closer to the internal choice $a(b).P \oplus a(b).Q \oplus \bar{a}(a).R$ than to the original process.

Generalising this to configurations should really be intuitive: we map any $\langle \gamma \parallel P_1, \dots, P_n \rangle$ to some behaviour on the position X with $X(\star) = \gamma$ and n agents of arity $|\gamma|$, given for each agent $i \in n$ by $\llbracket P_i \rrbracket$. In order to fully define such a position, we need to specify maps $f_i: |\gamma| \rightarrow \gamma$. We use for all of them the inverse of the canonical bijection h_γ defined by:

Definition 6.21. Let $h_\gamma: \gamma \cong |\gamma|$ map each $a \in \gamma$ to its position in the ordering induced by the one on natural numbers.

We call the obtained position $X(\gamma, n)$.

Definition 6.22. Let $\llbracket - \rrbracket : \text{ob}(\text{Conf}) \rightarrow \sum_X \mathbf{B}_X$ map any configuration $C = \langle \gamma \parallel P_1, \dots, P_n \rangle$ to the pair $(X(\gamma, n), \llbracket C \rrbracket)$, where $\llbracket C \rrbracket$ is defined through Proposition 6.8 by

$$\llbracket C \rrbracket(|\gamma|)(i) = \llbracket P_i \rrbracket_{h_\gamma},$$

for all $i \in n$. We implicitly consider processes P over γ as configurations $\langle \gamma \parallel P \rangle$, and hence allow ourselves to write $\llbracket P \rrbracket$ for $\llbracket \langle \gamma \parallel P \rangle \rrbracket$.

6.4. Semantic fair testing. In order to state our main result, it remains to define our semantic analogue of fair testing equivalence. It rests on two main ingredients: a notion of *closed-world* trace, and an analogue of *parallel composition* in game semantics.

The intuitive purpose of parallel composition is to let behaviours interact. If we partition the agents of a position X into two teams, we obtain two subpositions $X_1 \hookrightarrow X \leftarrow X_2$, each agent of X belonging to X_1 or X_2 according to its team. The crucial fact is that the category \mathbb{V}_X of views on X is isomorphic to the coproduct category $\mathbb{V}_{X_1} + \mathbb{V}_{X_2}$. Parallel composition of any $B_1 \in \widetilde{\mathbb{V}}_{X_1}$ and $B_2 \in \widetilde{\mathbb{V}}_{X_2}$ is then simply given by copairing $[B_1, B_2]$ (following Notation 6.9), as in

$$\begin{array}{ccc} \mathbb{V}_{X_1}^{op} & \hookrightarrow & \mathbb{V}_X^{op} & \longleftarrow & \mathbb{V}_{X_2}^{op} \\ & \searrow & \downarrow & \swarrow & \\ & B_1 & [B_1, B_2] & & B_2 \\ & & \text{set.} & & \end{array}$$

We now describe closed-world traces, which are then used as a criterion for success of tests. Closed-world actions were defined (Definition 5.1) as those not involving any interaction with the environment, i.e., formally, pushouts of a seed of any shape among $\nu_n, \tau_n, \heartsuit_n, \pi_n$, and $\tau_{n,a,m,c,d}$. A trace is *closed-world* when it is a composite of closed-world actions. Let $\mathbb{W}(X) \xrightarrow{\hookrightarrow} \mathbb{T}(X)$ denote the full subcategory of $\mathbb{T}(X)$ consisting of closed-world traces, and let the category of *closed-world strategies* be $\widetilde{\mathbb{W}}(X)$.

Notation 6.23. Recalling the discussion below Proposition 6.6, we denote by $B \mapsto \overline{B}$ the composite functor

$$\widetilde{\mathbb{V}}_X \xrightarrow{\text{ran } j_X^{op}} \widetilde{\mathbb{T}}_X \xrightarrow{\Delta_k^{op}} \overline{\mathbb{T}(X)} \xrightarrow{\Delta_l^{op}} \overline{\mathbb{W}(X)},$$

where Δ_f denotes restriction along f .

A closed-world trace is *successful* when it contains a \heartsuit action, and *unsuccessful* otherwise. A state $\sigma \in S(u)$ of any $S \in \overline{\mathbb{W}(Z)}$ over a closed-world trace $u: Z' \dashrightarrow Z$ is successful iff u is. Define \perp_Z as the set of closed-world strategies $S \in \overline{\mathbb{W}(Z)}$ such that any unsuccessful closed-world state admits a successful extension, i.e. $S \in \perp_Z$ iff for all unsuccessful $u \in \mathbb{W}(Z)$ and $\sigma \in S(u)$, there exists a successful $u' \in \mathbb{W}(Z)$, a morphism $f: u \rightarrow u'$, and a state $\sigma' \in S(u')$ such that $\sigma' \cdot f = \sigma$. Finally, in order to compare behaviours for semantic fair testing equivalence, we specify what a test is for a given behaviour $B \in \mathbf{B}_X$. A *test* consists of a position Y and a behaviour $T \in \mathbf{B}_Y$. Recalling Definition 3.8, we say that the pair (X, B) , with $B \in \mathbf{B}_X$, *should pass* the test (Y, T) iff $I_X = I_Y$ and $\overline{[B, T]} \in \perp_Z$, where Z is the pushout $X +_{I_X} Y$ (X and Y thus form two teams on Z). At last, we define *semantic fair testing equivalence*, for any $B \in \mathbf{B}_X$ and $B' \in \mathbf{B}_{X'}$:

Definition 6.24. Let $(X, B) \sim_f (X', B')$ iff they should pass the same tests.

We may at last state:

Theorem 6.25. *The translation $\llbracket - \rrbracket : \text{ob}(\text{Conf}) \rightarrow \sum_X \mathbf{B}_X$ is intensionally fully abstract for \sim_f , i.e.,*

- *For all configurations C_1 and C_2 , $C_1 \sim_f^{Pi} C_2$ iff $\llbracket C_1 \rrbracket \sim_f \llbracket C_2 \rrbracket$;*
- *Furthermore, for all positions X and behaviours $B \in \mathbf{B}_X$, there exists $C \in Pi_{X(\star)}$ such that $\llbracket C \rrbracket \sim_f B$.*

The proof is the subject of the next section.

7. INTENSIONAL FULL ABSTRACTION

In the previous section, exploiting the playground structure of \mathbb{D} established in Sections 4 and 5, we have defined and studied the notion of behaviour, into which we have translated π -calculus processes and configurations. We have then defined our semantic analogue of fair testing equivalence and stated our main result. We now work towards proving it. In Section 7.1, we define a graph with testing \mathcal{S} whose vertices are pairs (X, B) with B a definite behaviour on the position X (Definition 6.12), such that fair testing equivalence in \mathcal{S} coincides with fair testing equivalence in the model. In order to prove that this is the case, we introduce an intermediate graph with testing \mathcal{C} which is in fact quite intricate. We are then in a position where our main result is reduced to intensional full abstraction of a translation between two graphs with testing, Conf and \mathcal{S} , for which we may hope to apply the results of Section 2.3. In fact, in Section 7.2, we further reduce to a translation to a more syntactic graph with testing, \mathcal{M} . In Section 7.3, we prove intensional full abstraction of the translation to \mathcal{M} , from which we deduce Theorem 6.25. Finally, we generalise to a large class of testing equivalences in Section 7.4.

7.1. A first graph with testing for behaviours.

Definition 7.1. Let \mathcal{S} denote the graph with vertices in $\sum_X \mathbf{D}_X$, where we recall (Definition 6.12) that \mathbf{D}_X denotes the category of definite behaviours on X , and with non-identity edges $(X, D) \leftarrow (Y, D')$ all closed-world actions $M: Y \twoheadrightarrow X$ such that for all agents y in Y , there exists $\sigma_y \in |D \cdot M \cdot y|$ such that

$$D' \cdot y \cong (D \cdot M \cdot y)_{|\sigma_y}.$$

Moreover, let $\mathbf{p}^{\mathcal{S}} : \mathcal{S} \rightarrow \Sigma$ denote the map sending $(X, D) \xleftarrow{M} (X', D')$ to the \heartsuit edge in Σ if M is a tick action and to τ otherwise.

Let us now equip \mathcal{S} with testing structure.

Definition 7.2. We define the relation $|_{\mathcal{S}}$ by $(Z, D) \in ((X_1, D_1) |_{\mathcal{S}} (X_2, D_2))$ iff $X_1(\star) = X_2(\star)$ and there is a pushout square

$$\begin{array}{ccc} I_{X_1} & \longrightarrow & X_2 \\ \downarrow & \lrcorner & \downarrow \text{inj}_r \\ X_1 & \xrightarrow{\text{inj}_l} & Z \end{array}$$

such that $D_1 \cong D \cdot \text{inj}_l$ and $D_2 \cong D \cdot \text{inj}_r$, or otherwise said $D \cong [D_1, D_2]$.

Lemma 2.18 entails:

Proposition 7.3. *The morphism $\text{fc}(\mathfrak{p}^{\mathcal{S}}): \text{fc}(\mathcal{S}) \rightarrow \text{fc}(\Sigma)$, with $\mid_{\mathcal{S}}$ as testing relation, forms a free graph with testing.*

Proof. We exhibit a weak bisimulation relating any two such pushouts (Z, D) and (Z', D') . The relation containing two such pairs as soon as there exists a horizontal isomorphism $h: Z \rightarrow Z'$ such that $D \cong D' \cdot h$ does the job. \square

The crucial result for proving that fair testing equivalence in \mathcal{S} coincides with semantic fair testing equivalence is:

Lemma 7.4. *For any definite behaviour $D \in \mathbb{D}_X$, we have $D \in \perp_X$ iff $(X, D) \in \perp^{\mathcal{S}}$.*

Our strategy for proving this is to introduce and study an intermediate graph with testing \mathcal{C} , which is closer to semantic fair testing in that its transitions may comprise several actions. However, defining \mathcal{C} just as \mathcal{S} with arbitrary traces instead of actions would be wrong:

Example 7.5. Consider the trace $P = (\tau_0 \bullet \pi_0)$ consisting of a nullary agent performing a τ action and then forking. Consider now any definite behaviour D such that $D(\tau_0) = D(\tau_0 \bullet \pi_0^l) = D(\tau_0 \bullet \pi_0^r) = 2$, which maps both inclusions $\tau_0 \bullet \pi_0^l \hookrightarrow \tau_0 \hookrightarrow \tau_0 \bullet \pi_0^r$ to the identity. Then $\overline{D}(P) = 2$: it consists of pairs (σ^l, σ^r) in $D(\tau_0 \bullet \pi_0^l) \times D(\tau_0 \bullet \pi_0^r)$ whose restrictions to $D(\tau_0)$ coincide, which leaves just $(1, 1)$ and $(2, 2)$. Using the decompositions of Section 6.2, another way to say this is that $D = \langle \tau_0 \mapsto D_1 + D_2 \rangle$, with $D_i = \langle \pi_0^l \mapsto D_i^l, \pi_0^r \mapsto D_i^r \rangle$, for $i = 1, 2$. In order for \mathcal{C} to correspond to the model, assuming $D_1^l \not\cong D_2^l$ and $D_1^r \not\cong D_2^r$, there should be two transitions from $([0], D)$, one to $([0] \mid [0], [D_1^l, D_1^r])$ and the other to $([0] \mid [0], [D_2^l, D_2^r])$. But if we naively generalise Definition 7.1 to arbitrary traces, we obtain additional, ‘incoherent’ transitions, to $([0] \mid [0], [D_1^l, D_2^r])$ and $([0] \mid [0], [D_2^l, D_1^r])$.

Instead of relying on $\prod_{y \in \text{Agents}(Y)} |D \cdot P \cdot y|$ as in Definition 7.1, we would like to rely on $\overline{D}(P)$. This may be done by constructing a map

$$\psi_P^D: \overline{D}(P) \rightarrow \prod_{y: [n_y] \rightarrow Y} |D \cdot P \cdot y|, \quad (7.1)$$

whose image will consist precisely of all ‘coherent’ elements. (From now on, we omit the superscript D when clear from context.) Intuitively, this map associates to each global state the corresponding family of local states. Furthermore, ψ_P is always injective, but Example 7.5 shows that it is not surjective in general. In order to construct ψ_P , recalling Definition 6.11, we have by definition: $\prod_{y: [n_y] \rightarrow Y} |D \cdot P \cdot y| = \prod_{y: [n_y] \rightarrow Y} D(v^{y,P}, y^P)$. Furthermore, we also have

$$\overline{D}(P) \cong \int_{(v,x) \in \mathbb{V}_X} D(v,x)^{\mathbb{T}_X((v,x),(P, id_X))},$$

which is a subset of $\prod_{(v,x) \in \mathbb{V}_X} D(v,x)^{\mathbb{T}_X((v,x),(P, id_X))}$. We may thus define:

$$\psi_P(\sigma)(y) = \sigma(v^{y,P}, y^P)(id_{[n_y]}^{\bullet}, \alpha^{y,P}).$$

Here, $\sigma(v^{y,P}, y^P)$ is in $D(v^{y,P}, y^P)^{\mathbb{T}_X((v^{y,P}, y^P), (P, id_X))}$. So by applying it to $\alpha^{y,P}$ viewed as a morphism in $(v^{y,P}, y^P) \rightarrow (P, id_X)$ in \mathbb{T}_X , we obtain an element of $D(v^{y,P}, y^P) = |D \cdot P \cdot y|$, as desired.

We may now define the intermediate graph with testing \mathcal{C} . We first extend the notion of restriction:

Notation 7.6. We extend Notation 6.15: if $B \in \mathbf{B}_X$ and $\sigma \in \prod_{n,x:[n] \rightarrow X} |B \cdot x|$, let $B|_\sigma$ be defined up to isomorphism by

$$B|_\sigma \cdot x = (B \cdot x)|_{\sigma(x)}.$$

Definition 7.7. Let \mathcal{C} denote the graph with $\text{ob}(\mathcal{C}) = \text{ob}(\mathcal{S})$, and where $\mathcal{C}((X', D'), (X, D))$ is the set of closed-world traces $W: X' \dashrightarrow X$ such that there exists a state $\sigma \in \overline{D}(W)$ satisfying $(D \cdot W)|_{\psi_W(\sigma)} \cong D'$.

Thus, \mathcal{C} is a generalisation of \mathcal{S} from closed-world actions to closed-world traces. Let us turn it into a graph with testing.

Definition 7.8. Let $\mathbb{D}_v^{\mathbb{W}}$ denote the smallest locally full subcategory of \mathbb{D}_v containing all closed-world traces. The graph morphism $\mathbb{W} \rightarrow \Sigma$, where we recall that \mathbb{W} denotes the graph of closed-world actions (Definition 5.1), extends to a pseudo functor $\mathbf{p}^{\mathbb{W}}: \mathbb{D}_v^{\mathbb{W}} \rightarrow \text{fc}(\Sigma)$, which essentially counts the number of ticks. Let $\mathbf{p}^{\mathcal{C}}: \mathcal{C} \rightarrow \mathbb{D}_v^{\mathbb{W}}$ denote the obvious projection.

Proposition 7.9. *The composite projection $\mathcal{C} \xrightarrow{\mathbf{p}^{\mathcal{C}}} \mathbb{D}_v^{\mathbb{W}} \xrightarrow{\mathbf{p}^{\mathbb{W}}} \text{fc}(\Sigma)$, with $|\mathcal{S}$ as testing relation, makes \mathcal{C} into a graph with testing.*

Proof. Just as Lemma 7.3. □

As announced, \mathcal{C} is an example of a non-free graph with testing. The rest of this section is devoted proving Lemma 7.4, and reducing Theorem 6.25 to a statement about \mathcal{S} . Lemma 7.4 follows from the fact that both poles are equivalent to $\perp^{\mathcal{C}}$, as we now set out to prove. We start with a lemma saying that \mathcal{C} has essentially the same transitions over two (specially) isomorphic traces.

Notation 7.10. For any morphism $p: G \rightarrow H$ in \mathbf{Gph} , we denote by $p_{A,B}: G(A, B) \rightarrow H(p(A), p(B))$ the component of p at A and B .

Lemma 7.11. *For any $(X, D), (X', D') \in \mathcal{C}$, if there exists any special isomorphism $W_1 \cong W_2$ in $\mathbb{D}_v^{\mathbb{W}}(X', X)$, we have*

$$(\mathbf{p}^{\mathcal{C}})_{(X', D'), (X, D)}^{-1}(W_1) \cong (\mathbf{p}^{\mathcal{C}})_{(X', D'), (X, D)}^{-1}(W_2).$$

Proof. The given special isomorphism induces by pseudo double functoriality of \mathbf{B} an isomorphism $\varphi: D \cdot W_1 \cong D \cdot W_2$, hence an isomorphism

$$\overline{\varphi}_{id_{X'}}: \overline{D}(W_1) \cong \overline{D}(W_2).$$

This isomorphism is such that for any σ ,

$$(D \cdot W_1)|_{\psi_{W_1}(\sigma)} \cong (D \cdot W_2)|_{\psi_{W_2}(\overline{\varphi}_{id_{X'}}(\sigma))}.$$

Thus, $(D \cdot W_1)|_{\psi_{W_1}(\sigma)} \cong D'$ iff $(D \cdot W_2)|_{\psi_{W_2}(\overline{\varphi}_{id_{X'}}(\sigma))} \cong D'$. □

In order to relate \mathcal{C} to \mathcal{S} , let us now show that transitions in \mathcal{C} behave well w.r.t. composition of traces. First, transitions compose, and second, transitions over any composite (closed-world) trace $W_1 \bullet W_2$ always decompose into a transition over W_1 followed by one over W_2 .

Lemma 7.12. *For all edges $(X, D) \xleftarrow{W} (X', D') \xleftarrow{W'} (X'', D'')$ in \mathcal{C} , there is an edge $(X, D) \xleftarrow{W \bullet W'} (X'', D'')$.*

Proof. Consider $\sigma \in \overline{D}(W)$ such that $(D \cdot W)|_{\psi_W(\sigma)} \cong D'$ and $\sigma' \in \overline{D'}(W')$ such that $(D' \cdot W')|_{\psi_{W'}(\sigma')} \cong D''$. We want to construct an edge $W \bullet W': (X'', D'') \rightarrow (X, D)$, i.e., find $\sigma'' \in \overline{D}(W \bullet W')$ such that $(D \cdot (W \bullet W'))|_{\psi_{W \bullet W'}(\sigma'')} \cong D''$. Now, the isomorphism $\varphi: (D \cdot W)|_{\psi_W(\sigma)} \cong D'$ yields a state $\sigma_1 = \overline{\varphi}_{W'}^{-1}(\sigma') \in \overline{(D \cdot W)}|_{\psi_W(\sigma)}(W')$ such that

$$((D \cdot W)|_{\psi_W(\sigma)} \cdot W')|_{\psi_{W'}(\sigma_1)} \cong (D' \cdot W')|_{\psi_{W'}(\sigma')}. \quad (7.2)$$

Now we have

$$\overline{(D \cdot W)}|_{\psi_W(\sigma)}(W') \cong \{\sigma'' \in \overline{D}(W \bullet W') \mid \sigma''|_W = \sigma\},$$

where $\sigma''|_W$ denotes restriction of σ'' along the prefix inclusion $W \hookrightarrow W \bullet W'$. So the left-hand side in (7.2) is just $(D \cdot (W \bullet W'))|_{\psi_{W \bullet W'}(\sigma_1)}$, which yields the desired transition. \square

Lemma 7.13. *The projection $\mathfrak{p}^{\mathcal{C}}: \mathcal{C} \rightarrow \mathbb{D}_v^{\mathbb{W}}$ satisfies the following weak Conduché condition: for all $X'' \xrightarrow{W_2} X' \xrightarrow{W_1} X$, if there is an edge $(X'', D'') \xrightarrow{W_1 \bullet W_2} (X, D)$ in \mathcal{C} , then there exists $D' \in \mathcal{D}_{X'}$ and edges $(X'', D'') \xrightarrow{W_2} (X', D') \xrightarrow{W_1} (X, D)$.*

Proof. Consider any $(X, D) \in \mathcal{C}$ and $\sigma \in \overline{D}(W_1 \bullet W_2)$ witnessing the given edge. Consider also the morphism $u: W_1 \rightarrow (W_1 \bullet W_2)$ given by (W_2, id) , and let $\sigma_1 = \sigma \cdot u \in \overline{D}(W_1)$. Let $D_1 = (D \cdot W_1)|_{\psi_{W_1}(\sigma_1)}$. We have $\sigma \in \{\sigma' \in \overline{D}(W_1 \bullet W_2) \mid \sigma' \cdot u = \sigma_1\}$, hence $\sigma \in \overline{D}_1(W_2)$. Furthermore,

$$(D_1 \cdot W_2)|_{\psi_{W_2}(\sigma)} \cong (D \cdot (W_1 \bullet W_2))|_{\psi_{W_1 \bullet W_2}(\sigma)} \cong D'',$$

so we have two edges

$$(X, D) \xleftarrow{(W_1, \sigma_1)} (X', D_1) \xleftarrow{(W_2, \sigma)} (X'', D'')$$

as desired. \square

The previous result generalises by induction to n -ary composites:

Notation 7.14. By default, composition in $\mathbb{D}_v^{\mathbb{W}}$ associates to the right, i.e., $W \bullet W' \bullet W''$ denotes $W \bullet (W' \bullet W'')$.

Corollary 7.15. *For any path p , say*

$$X = X_0 \xleftarrow{M_1} X_1 \xleftarrow{M_2} \dots X_n = X', \quad (7.3)$$

in \mathbb{W} and edge $(X', D') \xrightarrow{W} (X, D)$ in \mathcal{C} over its right-associated, n -ary composition $W = (M_1 \bullet (\dots \bullet M_n))$, there is a path $e = (e_1, \dots, e_n)$ in $\mathcal{C}^((X', D'), (X, D))$ such that*

$$(\mathfrak{p}^{\mathcal{C}})_{(X', D'), (X, D)}^*(e) = p.$$

Proof. By induction on n . \square

Our next goal is to relate transitions in \mathcal{C} to sequences of transitions in \mathcal{S} . First of all, \mathcal{C} and \mathcal{S} coincide on actions:

Lemma 7.16. *If $W: X' \dashrightarrow X$ is a closed-world action (i.e., has length 1), then for all D and D' both fibres of $\mathcal{C}((X', D'), (X, D))$ and $\mathcal{S}((X', D'), (X, D))$ over W are equal.*

Proof. By [39, Proposition 5.23]. \square

Now, let us show that for any sequence of closed-world actions, sequences of transitions in \mathcal{S} correspond to transitions over the composite in \mathcal{C} .

Corollary 7.17. *For all closed-world paths as in (7.3), and $(X, D), (X', D') \in \mathcal{S}$, we have*

$$((\mathbf{p}^{\mathcal{S}})_{(X', D'), (X, D)}^*)^{-1}(p) \neq \emptyset \quad \text{iff} \quad (\mathbf{p}_{(X', D'), (X, D)}^{\mathcal{C}})^{-1}(P) \neq \emptyset,$$

for any special isomorphism $P \cong (M_1 \bullet (\dots \bullet M_n))$.

Proof. Consider any special isomorphism $\alpha: P \cong (M_1 \bullet (\dots \bullet M_n))$. We have

$$\begin{aligned} & ((\mathbf{p}^{\mathcal{S}})_{(X', D'), (X, D)}^*)^{-1}(p) \neq \emptyset \\ \text{iff } & ((\mathbf{p}^{\mathcal{C}})_{(X', D'), (X, D)}^*)^{-1}(p) \neq \emptyset && \text{(by Lemma 7.16)} \\ \text{iff } & (\mathbf{p}_{(X', D'), (X, D)}^{\mathcal{C}})^{-1}(M_1 \bullet (\dots \bullet M_n)) \neq \emptyset && \text{(by Lemma 7.12 and Corollary 7.15)} \\ \text{iff } & (\mathbf{p}_{(X', D'), (X, D)}^{\mathcal{C}})^{-1}(P) \neq \emptyset && \text{(by Lemma 7.11).} \quad \square \end{aligned}$$

As a corollary, we get that the identity relation on objects is a strong bisimulation between $\text{fc}(\mathcal{S})$ and \mathcal{C} :

Corollary 7.18. *For all $w \in \Sigma^*(\star, \star)$ and $(X, D), (X', D') \in \mathcal{S}$, we have $(X, D) \xleftarrow{w} (X', D')$ in \mathcal{S} iff $(X, D) \xleftarrow{\tilde{w}} (X', D')$ in \mathcal{C} .*

The last statement is slightly subtle, in that $(X, D) \xleftarrow{\tilde{w}} (X', D')$ denotes a single edge in \mathcal{C} , lying over the composite \tilde{w} in $\text{fc}(\Sigma)$.

Proof. If $(X, D) \xleftarrow{w} (X', D')$ in \mathcal{S} , then there exists $p \in \mathbb{W}^*$ such that $\text{fc}(\mathbf{p}^{\mathbb{W}})(p) = \tilde{w}$ and there is a path $e: (X', D') \rightarrow (X, D)$ over p in \mathcal{S} . Let $W: X' \dashrightarrow X$ denote the composition of p . By Corollary 7.17, we get an edge $(X', D') \rightarrow (X, D)$ over W in \mathcal{C} . So since $\mathbf{p}^{\mathbb{W}}(W) = \tilde{w}$, this gives us the expected transition.

Conversely, if $(X, D) \xleftarrow{\tilde{w}} (X', D')$ in \mathcal{C} , then let $W: X' \dashrightarrow X$ denote the corresponding edge in $\mathbb{D}_v^{\mathbb{W}}$. In particular, we have $\mathbf{p}^{\mathbb{W}}(W) = \tilde{w}$. Decomposing W as some path p in \mathbb{W} , we obtain by Corollary 7.17 a transition sequence $(X, D) \xleftarrow{(\mathbf{p}^{\mathbb{W}})^*(p)} (X', D')$ in \mathcal{S} . But $(\mathbf{p}^{\mathbb{W}})^*(p) = \mathbf{p}^{\mathbb{W}}(W) = \tilde{w}$, as desired. \square

As promised, we readily obtain:

Corollary 7.19. *We have $\perp^{\mathcal{S}} = \perp^{\mathcal{C}}$.*

Let us also prove the analogous result with the semantic pole \perp .

Lemma 7.20. *We have $D \in \perp_X$ iff $(X, D) \in \perp^{\mathcal{C}}$.*

Proof. Assume $D \in \perp_X$, and consider any $(X, D) \leftarrow (X', D')$. The latter is witnessed by some unsuccessful, closed-world trace $W: X' \dashrightarrow X$, state $\sigma \in \overline{D}(W)$, and isomorphism $h: (D \cdot W)|_{\psi_W(\sigma)} \cong D'$.

By hypothesis, σ admits an extension $\sigma' \in \overline{D}(W \bullet W')$ for some successful $W': X'' \dashrightarrow X'$. Letting $D'' = (D \cdot (W \bullet W'))|_{\psi_{W \bullet W'}(\sigma')}$, we have

$$D'' \cong (((D \cdot W)|_{\psi_W(\sigma)} \cdot W')|_{\psi_{W'}(\sigma')}) \cong (D' \cdot W')|_{\psi_{W'}(\bar{h}_{W'}(\sigma'))},$$

and hence $(X', D') \xleftarrow{\heartsuit^n} (X'', D'')$ for some $n > 0$. This shows that $(X, D) \in \perp^{\mathcal{C}}$.

Conversely, assume $(X, D) \in \perp^{\mathcal{C}}$ and consider any unsuccessful, closed-world trace $W: X' \dashrightarrow X$ and state $\sigma \in \overline{D}(W)$. Letting $D' = (D \cdot W)|_{\psi_W(\sigma)}$, we have $(X, D) \leftarrow (X', D')$.

By hypothesis, we find some transition $(X', D') \xleftarrow{\heartsuit} (X'', D'')$, witnessed by some successful $W': X'' \rightarrow X'$. Hence, $D'' \cong (D' \cdot W')|_{\psi_{W'}(\sigma')}$ for a certain $\sigma' \in \overline{D'}(W')$. By definition of D' , σ' is a state in $\overline{D}(W \bullet W')$ such that $\sigma' \cdot u = \sigma$, where $u: W \rightarrow (W \bullet W')$ is (W', id) . This gives the desired successful extension of σ , which shows that $D \in \perp_X$. \square

Combining the last two results, we may now prove that the semantic pole coincides with that of \mathcal{S} :

Proof of Lemma 7.4. We have $D \in \perp_X$ iff $(X, D) \in \perp^c$ iff $(X, D) \in \perp^{\mathcal{S}}$ by Corollary 7.19 and Lemma 7.20. \square

As expected, this entails preservation and reflection of semantic fair testing equivalence:

Corollary 7.21. *For all $D \in \mathcal{D}_X$ and $D' \in \mathcal{D}_{X'}$, we have $(X, D) \sim_f (X', D')$ iff $(X, D) \sim_f^{\mathcal{S}} (X', D')$*

Proof. Let us first show that semantic fair testing equivalence may as well be defined only with definite tests. Indeed, if $I_X \neq I_{X'}$ then the result holds trivially. So assuming $I_X = I_{X'}$, consider any test $B \in \mathcal{B}_Y$ with $I_X = I_Y$, and, w.l.o.g., $\overline{[D, B]} \in \perp_Z$ and $\overline{[D', B]} \notin \perp_{Z'}$ with $Z = X +_{I_X} Y$ and $Z' = X' +_{I_{X'}} Y$. Then, letting $B = \sum_{i \in \gamma} D_i$ with each D_i definite, there exists i such that $\overline{[D, D_i]} \in \perp_Z$ and $\overline{[D', D_i]} \notin \perp_{Z'}$, hence D_i also distinguishes D from D' .

Returning to our main goal, for any definite test $T \in \mathcal{D}_Y$ with $I_Y = I_X$, by Lemma 7.4, $\overline{[D, T]} \in \perp_Z$ iff $(Z, \overline{[D, T]}) = ((X, D) | (Y, T)) \in \perp^{\mathcal{S}}$, which easily entails the result. \square

From this we may reduce our main theorem to a result on \mathcal{S} :

Corollary 7.22. *If the translation $\llbracket - \rrbracket: \text{ob}(\text{Conf}) \rightarrow \sum_X \mathcal{D}_X$ is intensionally fully abstract for $\sim_f^{\mathcal{S}}$, then Theorem 6.25 holds, i.e., $\llbracket - \rrbracket$ is also intensionally fully abstract for \sim_f .*

Proof. For all configurations C_1 and C_2 , by Corollary 7.21, $\llbracket C_1 \rrbracket \sim_f \llbracket C_2 \rrbracket$ iff $\llbracket C_1 \rrbracket \sim_f^{\mathcal{S}} \llbracket C_2 \rrbracket$, which holds iff $C_1 \sim_f^{\text{Conf}} C_2$ by hypothesis.

It remains to prove that surjectivity up to \sim_f reduces to surjectivity up to $\sim_f^{\mathcal{S}}$. For this, let us first show that any behaviour is \sim_f -equivalent to some definite one. Indeed, consider any $B \in \mathcal{B}_X$. Letting $B \cdot x = \sum_{i \in n_x} D_i^x$ for all agents x in X , B is fair testing equivalent to the definite behaviour D such that

$$D \cdot x = \langle \tau_{n_x} \mapsto \sum_{i \in n_x} D_i^x \rangle,$$

except if $B \cdot x = \emptyset$ for some x . But in the latter case, B is fair testing equivalent to the definite behaviour on one nullary player with the same interface which merely ticks.

Thus, we may restrict attention to definite behaviours. So consider any definite $D \in \mathcal{D}_X$. By hypothesis, there exists C such that $\llbracket C \rrbracket \sim_f^{\mathcal{S}} (X, D)$, hence $\llbracket C \rrbracket \sim_f (X, D)$ by Corollary 7.21, which concludes the proof. \square

7.2. A further graph with testing for behaviours. In the previous section, we have characterised semantic fair testing equivalence using the graph with testing \mathcal{S} , and reduced intensional full abstraction of $\llbracket - \rrbracket$ w.r.t. \sim_f to intensional full abstraction w.r.t. $\sim_f^{\mathcal{S}}$. We now define a further graph with testing, \mathcal{M} , which will help us bridge the gap between \mathcal{S} and π -calculus configurations. Indeed, we define a surjective morphism $m: \mathcal{S} \rightarrow \mathcal{M}$ over Σ , and we then prove that m is intensionally fully abstract (Proposition 7.30), from which we deduce that our main result follows from intensional full abstractness of the composite translation $T = m \circ \llbracket - \rrbracket$ (Lemma 7.32).

Recall from Definition 2.3 that $-^{\circ}$ denotes the finite multiset monad on sets.

Definition 7.23. Let the set \mathcal{M}_0 of *mixed behaviours* be

$$\sum_{\gamma \in \mathcal{P}_f(\mathbb{N})} \left(\sum_{n \in \mathbb{N}} D_{[n]} \times \gamma^n \right)^{\circ}.$$

The graph \mathcal{M} over Σ is inductively defined by the rules in Figure 7, where $S[\gamma_1 \subseteq \gamma_2]$ denotes pointwise composition of the substitution component with the inclusion $h: \gamma_1 \subseteq \gamma_2$, i.e., each $D[\sigma]$ is replaced by $D[h \circ \sigma]$.

Let $p^{\mathcal{M}}: \mathcal{M} \rightarrow \Sigma$ denote the projection.

$$\begin{array}{c} \frac{i \in |D \cdot \pi_n^l| \quad j \in |D \cdot \pi_n^r|}{\langle \gamma \| D[\sigma] \rangle \xleftarrow{id} \langle \gamma \| (D \cdot \pi_n^l)_i[\sigma], (D \cdot \pi_n^r)_j[\sigma] \rangle} \qquad \frac{i \in |D \cdot \tau_n|}{\langle \gamma \| D[\sigma] \rangle \xleftarrow{id} \langle \gamma \| (D \cdot \tau_n)_i[\sigma] \rangle} \\ \\ \frac{i \in |D \cdot \heartsuit_n|}{\langle \gamma \| D[\sigma] \rangle \xleftarrow{\heartsuit} \langle \gamma \| (D \cdot \heartsuit_n)_i[\sigma] \rangle} \qquad \frac{i \in |D \cdot \nu_n| \quad a \notin \gamma}{\langle \gamma \| D[\sigma] \rangle \xleftarrow{id} \langle \gamma, a \| (D \cdot \nu_n)_i[n+1 \xrightarrow{\sigma + \ulcorner a \urcorner} \gamma, a] \rangle} \\ \\ \frac{i \in |D_1 \cdot \iota_{n_1, a_1}| \quad j \in |D_2 \cdot \circ_{n_2, a_2, b_2}| \quad \sigma_1(a_1) = \sigma_2(a_2)}{\langle \gamma \| D_1[\sigma_1], D_2[\sigma_2] \rangle \xleftarrow{id} \langle \gamma \| (D_1 \cdot \iota_{n_1, a_1})_i[n_1+1 \xrightarrow{[\sigma_1, \ulcorner \sigma_2(b_2) \urcorner]} \gamma], (D_2 \cdot \circ_{n_2, a_2, b_2})_j[\sigma_2] \rangle} \\ \\ \frac{\langle \gamma_1 \| S_1 \rangle \xleftarrow{\alpha} \langle \gamma_2 \| S_2 \rangle}{\langle \gamma_1 \| S \cup S_1 \rangle \xleftarrow{\alpha} \langle \gamma_2 \| S[\gamma_1 \subseteq \gamma_2] \cup S_2 \rangle} \qquad \frac{}{\langle \gamma \| S \rangle \xleftarrow{id} \langle \gamma \| S \rangle} \end{array}$$

Figure 7: Transitions in \mathcal{M}

Notation 7.24. Similarly to the notation for configurations, we denote

$$\langle \gamma, [(n_1, D_1, \sigma_1), \dots, (n_p, D_p, \sigma_p)] \rangle \quad \text{by} \quad \langle \gamma \| D_1[\sigma_1], \dots, D_p[\sigma_p] \rangle.$$

For the testing structure of \mathcal{M} , we mimick Definition 2.7 and put:

Definition 7.25. For any $\langle \gamma \| S \rangle, \langle \gamma' \| S' \rangle \in \mathcal{M}$, let $\langle \gamma \| S \rangle @ \langle \gamma' \| S' \rangle$ denote $\langle \gamma \| S \cup S' \rangle$ if $\gamma = \gamma'$ and be undefined otherwise. Let furthermore $\epsilon_{\gamma} = \langle \gamma \| \rangle$.

By Lemma 2.18, we have:

Proposition 7.26. *The morphism $\text{fc}(p^{\mathcal{M}}): \text{fc}(\mathcal{M}) \rightarrow \text{fc}(\Sigma)$, with the graph of $@$ as testing relation, forms a free graph with testing.*

Let us now reduce Theorem 6.25 to a statement about \mathcal{M} . In order to do this, we will use Lemma 2.22 and hence need to exhibit a fair relation $\text{ob}(\mathcal{S}) \rightarrow \mathcal{M}_0$. We use the graph of the following map:

Definition 7.27. Let $\mathfrak{m}: \text{ob}(\mathcal{S}) \rightarrow \mathcal{M}_0$ map any (X, D) to the mixed behaviour

$$\langle X(\star) \parallel [(D \cdot x)[\sigma_x] \mid (n, x) \in \text{Agents}(X)] \rangle,$$

where σ_x is the map $n \xrightarrow{[x \cdot s_i]_{i \in n}} X(\star)$.

We now need to show that \mathfrak{m} yields a fair relation. Most points are direct, like totality or the fact that $(X, D) \subset (X', D')$ iff $\mathfrak{m}(X, D) \subset \mathfrak{m}(X', D')$. Furthermore, we have:

Proposition 7.28. *The map $\mathfrak{m}: \text{ob}(\mathcal{S}) \rightarrow \mathcal{M}_0$ is surjective. Let \mathfrak{a} denote any section of \mathfrak{m} .*

The point about bisimilarity is trickier:

Proposition 7.29. *We have $(X, D) \sim_{\Sigma} \mathfrak{m}(X, D)$ for all $(X, D) \in \text{ob}(\mathcal{S})$.*

Proof. It is enough to prove that (the graph of) \mathfrak{m} is a strong bisimilarity up to strong bisimilarity. For this, let us record that clearly for any isomorphism $h: X \rightarrow Y$ of positions and $D \in \text{D}_Y$, we have $(Y, D) \sim_{\Sigma} (X, D \cdot h)$ in \mathcal{S} . Let us call \mathcal{J} (for isomorphism) the relation given by all pairs $((Y, D), (X, D \cdot h))$, so that we have $\mathcal{J} \subseteq \sim_{\Sigma}$. By case analysis, we can show that \mathfrak{m} is a strong bisimulation up to \mathcal{J} , i.e., for all $(X, D) \xleftarrow{\alpha} (Y, D')$, there exists $(Y, D') \mathcal{J} (Z, D'')$ such that $\mathfrak{m}(X, D) \xleftarrow{\alpha} \mathfrak{m}(Z, D'')$, as below left. And more tightly, for all $\mathfrak{m}(X, D) \xleftarrow{\alpha} M'$, there exists $(X, D) \xleftarrow{\alpha} (X', D')$ such that $\mathfrak{m}(X', D') = M'$, as below right.

$$\begin{array}{ccc} (X, D) & \xrightarrow{\mathfrak{m}} & \mathfrak{m}(X, D) \\ \alpha \uparrow & & \uparrow \alpha \\ (Y, D') \mathcal{J} (Z, D'') & \xrightarrow{\mathfrak{m}} & \mathfrak{m}(Z, D'') \end{array} \qquad \begin{array}{ccc} (X, D) & \xrightarrow{\mathfrak{m}} & \mathfrak{m}(X, D) \\ \alpha \uparrow & & \uparrow \alpha \\ (Y, D') & \xrightarrow{\mathfrak{m}} & M' \end{array}$$

The right-hand diagram is a tedious yet straightforward case analysis. The left-hand one is also a tedious case analysis, whose main point is that for all transitions $(X, D) \xleftarrow{\alpha} (Y, D')$, some renaming of elements of X may take place, which cannot happen in any transition from $\mathfrak{m}(X, D)$. So in each case we need to find the Z and corresponding D'' which avoids such renaming. In fact, this goes by indentifying the right transition from $\mathfrak{m}(X, D)$ and showing that the obtained M' is of the form $\mathfrak{m}(Z, D'')$ for $(Y, D') \mathcal{J} (Z, D'')$. Let us do one case. If $(X, D) \xleftarrow{\heartsuit} (Y, D')$, then we have an iso $h: X \cong Y$, and there is some agent (n_0, x_0) in X and i such that $D' \cdot h(x_0) = (D \cdot x_0 \cdot \heartsuit_{n_0})_i$ and furthermore for all $(n, x) \neq (n_0, x_0)$ in $\text{Agents}(X)$, we have $D' \cdot h(x) = D \cdot x$. So, letting $y = h(x)$ for all such x , we indeed have

$M = \langle \gamma \parallel (D \cdot x_0)[\sigma_{x_0}] :: [(D \cdot x)[\sigma_x] \mid x \neq x_0] \rangle \xleftarrow{\heartsuit} \langle \gamma \parallel (D' \cdot y_0)[\sigma_{y_0}] :: [(D \cdot x)[\sigma_x] \mid x \neq x_0] \rangle = M'$ with $M = \mathfrak{m}(X, D)$, and furthermore letting D'' be determined by $D'' \cdot x = D' \cdot h(x) = D \cdot x$ for $x \neq x_0$ and $D'' \cdot x_0 = D' \cdot h(x_0)$, we have $D'' = D' \cdot h$ and hence $(Y, D') \mathcal{J} (X, D'')$ with $\mathfrak{m}(X, D'') = M'$, as desired. \square

We are now ready to show:

Proposition 7.30. *The map $m: \text{ob}(\mathcal{S}) \rightarrow \text{ob}(\mathcal{M})$ is intensionally fully abstract for fair testing equivalence.*

Proof. As announced, for preservation and reflection of fair testing equivalence we apply Lemma 2.22: we have established all necessary hypotheses, except the last one, which follows by choosing a pushout with the same set of channels as its summands.

For surjectivity up to $\sim_f^{\mathcal{M}}$, consider any $\langle \gamma \| S \rangle \in \mathcal{M}_0$. Because a is a section of m , we have $m(a(\langle \gamma \| S \rangle)) = \langle \gamma \| S \rangle$, hence $m(a(\langle \gamma \| S \rangle)) \sim_f^{\mathcal{M}} \langle \gamma \| S \rangle$, thus providing the desired antecedent. \square

Let us now reduce Theorem 6.25 to its analogue about \mathcal{M} .

Definition 7.31. Let $\mathsf{T}: \text{ob}(\text{Conf}) \rightarrow \text{ob}(\mathcal{M})$ denote the composite

$$\text{ob}(\text{Conf}) \xrightarrow{[-]} \text{ob}(\mathcal{S}) \xrightarrow{m} \text{ob}(\mathcal{M}).$$

Concretely, we have

$$\mathsf{T}\langle \gamma \| P_1, \dots, P_n \rangle = \langle \gamma \| \llbracket P_1 \rrbracket_{h_\gamma} [h_\gamma^{-1}], \dots, \llbracket P_n \rrbracket_{h_\gamma} [h_\gamma^{-1}] \rangle.$$

Lemma 7.32. *The translation $[-]$ from Definition 6.22 is intensionally fully abstract for $\sim_f^{\mathcal{S}}$ if T is for $\sim_f^{\mathcal{M}}$.*

Proof. Assuming T is intensionally fully abstract, then for all configurations C and C' , we have that $C \sim_f C'$ iff $\mathsf{T}(C) \sim_f^{\mathcal{M}} \mathsf{T}(C')$. But by Proposition 7.30, we have

$$\mathsf{T}(C) = m\llbracket C \rrbracket \sim_f^{\mathcal{M}} m\llbracket C' \rrbracket = \mathsf{T}(C') \quad \text{iff} \quad \llbracket C \rrbracket \sim_f^{\mathcal{S}} \llbracket C' \rrbracket,$$

hence $C \sim_f C'$ iff $\llbracket C \rrbracket \sim_f \llbracket C' \rrbracket$.

Finally, for any $(X, D) \in \mathcal{S}$, by intensional full abstractness of T , we find a configuration C such that $\mathsf{T}(C) \sim_f^{\mathcal{M}} m(X, D)$, hence by Proposition 7.30 again $\llbracket C \rrbracket \sim_f (X, D)$. \square

7.3. Intensional full abstraction. We at last prove our main result, by proving intensional full abstractness of T . Our strategy is to define a relation $\bowtie: \text{Conf} \leftrightarrow \mathcal{M}$ over Σ which

- relates any configuration to its image under T , and
- is surjective, i.e., relates any mixed behaviour to some configuration.

We will then show that this relation is a weak bisimulation over Σ , which will entail the result.

Let us start with the second point, and define a map $Z \in \text{ob}(\mathcal{M}) \rightarrow \text{ob}(\text{Conf})$, which associates a configuration to each mixed behaviour. We first coinductively define ζ for definite behaviours on representable positions by:

$$\zeta(n \vdash D) = \left(\begin{array}{l} \sum_{i \in |D \cdot \pi_n^l|, j \in |D \cdot \pi_n^r|} \tau. (\zeta(n \vdash (D \cdot \pi_n^l)_i) \mid \zeta(n \vdash (D \cdot \pi_n^r)_j)) \\ + \sum_{i \in |D \cdot \tau_n|} \tau. \zeta(n \vdash (D \cdot \tau_n)_i) \\ + \sum_{i \in |D \cdot \heartsuit_n|} \heartsuit. \zeta(n \vdash (D \cdot \heartsuit_n)_i) \\ + \sum_{i \in |D \cdot \nu_n|} \nu(n+1). \zeta(n+1 \vdash (D \cdot \nu_n)_i) \\ + \sum_{a \in n, i \in |D \cdot \iota_{n,a}|} a(n+1). \zeta(n+1 \vdash (D \cdot \iota_{n,a})_i) \\ + \sum_{a, b \in n, i \in |D \cdot o_{n,a,b}|} \bar{a}\langle b \rangle. \zeta(n \vdash (D \cdot o_{n,a,b})_i) \end{array} \right),$$

where $n \vdash D$ means $D \in \mathcal{D}_{[n]}$.

Except perhaps for the first term of the sum, this should be rather natural: each definite behaviour on a representable position corresponds to a guarded sum, with one term for each state over each basic action – the translation is direct. The twist in the first term is due to the fact that forking is not a guard in π , so $P|Q$ cannot appear in any guarded sum. But $\tau.(P|Q)$ can, and it is clearly weakly bisimilar to $P|Q$, so this is precisely what ζ does.

Let us now extend ζ to arbitrary configurations:

Definition 7.33. Let $Z:\text{ob}(\mathcal{M}) \rightarrow \text{ob}(\text{Conf})$ be defined by

$$Z\langle \gamma \| D_1[\sigma_1], \dots, D_n[\sigma_n] \rangle = \langle \gamma \| \zeta(D_1)[\sigma_1], \dots, \zeta(D_n)[\sigma_n] \rangle.$$

Remark 7.34. Let us emphasise that brackets on the right denote proper substitutions, while on the left $D_i[\sigma_i]$ is just syntactic sugar for (n_i, D_i, σ_i) by Notation 7.24.

In order for Z to return an antecedent up to $\sim_f^{\mathcal{M}}$, we immediately observe that for any D and $i \in |D \cdot \pi_n^l|$ and $j \in |D \cdot \pi_n^r|$,

- on the one hand D has a silent transition to $\partial_{i,j}D$, the behaviour on $[n] | [n]$ such that $(\partial_{i,j}D) \cdot x_1 = (D \cdot \pi_n^l)|_i$ and $(\partial_{i,j}D) \cdot x_2 = (D \cdot \pi_n^r)|_j$ (where x_1 and x_2 denote the two agents of $[n] | [n]$);
- on the other hand $\zeta(D)$ has a silent transition to

$$\zeta'_{i,j}(D) = (\zeta((D \cdot \pi_n^l)|_i) | \zeta((D \cdot \pi_n^r)|_j)),$$

which then has a further silent transition to the two-process configuration consisting of $\zeta((D \cdot \pi_n^l)|_i)$ and $\zeta((D \cdot \pi_n^r)|_j)$.

Thus, when we try to relate D and $\zeta(D)$, the transition $\zeta(D) \xleftarrow{id} \zeta'_{i,j}(D)$ has to be matched by the former transition $D \xleftarrow{id} \partial_{i,j}D$. So our relation \bowtie should somehow include pairs $(\zeta'_{i,j}(D), \partial_{i,j}D)$.

Definition 7.35. Let the relation $\bowtie:\text{Conf} \rightarrow \mathcal{M}$ over Σ be defined inductively by the rules in Figure 8.

(In the last rule, ϵ_γ is understood in Conf on the left, and in \mathcal{M} on the right.)

$$\frac{\gamma' \vdash P \quad h: \gamma' \cong n \quad \sigma: n \rightarrow \gamma}{P[\sigma \circ h] \bowtie [[P]]_h[\sigma]} \quad \frac{n \vdash D \quad \sigma: n \rightarrow \gamma}{\zeta(D)[\sigma] \bowtie D[\sigma]}$$

$$\frac{n \vdash D^1, D^2 \quad \sigma: n \rightarrow \gamma}{(\zeta(D^1) | \zeta(D^2))[\sigma] \bowtie D^1[\sigma], D^2[\sigma]} \quad \frac{C \bowtie M \quad D \bowtie N}{C @ D \bowtie M @ N} \quad \frac{}{\epsilon_\gamma \bowtie \epsilon_\gamma}.$$

Figure 8: The relation \bowtie

Lemma 7.36. *We have $C \bowtie \top(C)$ for all C and $Z(M) \bowtie M$ for all M , and so \bowtie is total and surjective.*

Proof. By construction. □

Lemma 7.37. *The relation \bowtie is a weak bisimulation over Σ .*

Proof. First, we observe that \bowtie may equivalently be defined by first letting \bowtie_0 be generated by all rules except the last two, and then adding the rule

$$\frac{C_1 \bowtie_0 M_1 \quad \dots \quad C_n \bowtie_0 M_n}{C_1 @ \dots @ C_n \bowtie M_1 @ \dots @ M_n} \quad (n \in \mathbb{N}).$$

The advantage of this presentation is that proofs of $C \bowtie M$ all have depth at most 1.

The rest is then a tedious case analysis, which we defer to Appendix A. In summary, we easily observe that the ‘forwards’ clause of weak bisimulation is satisfied, the only subtlety being that heating $(\zeta(D^1) \mid \zeta(D^2))[\sigma]$ should be matched by the identity edge on the corresponding behaviour. Furthermore, the ‘backwards’ clause is also easily satisfied, the only subtlety being that if the considered behaviour performs a transition involving one or several $[D^1[\sigma], D^2[\sigma]]$ ’s, related on the left to $(\zeta(D^1) \mid \zeta(D^2))[\sigma]$ ’s, then all of the latter first have to heat to $[\zeta(D^1)[\sigma], \zeta(D^2)[\sigma]]$, and only then perform the matching transition. \square

This easily entails:

Lemma 7.38. *For all C_1, C_2, M_1, M_2 , if $C_1 \bowtie M_1$ and $C_2 \bowtie M_2$, then $C_1 \sim_f C_2$ iff $M_1 \sim_f M_2$.*

Proof. The relation \bowtie is weakly fair, the only difficult points being proved by Lemmas 7.36 and 7.37 above. We thus conclude by Corollary 2.25. \square

Theorem 7.39. *The map $\mathsf{T}: \text{ob}(\text{Conf}) \rightarrow \text{ob}(\mathcal{M})$ is intensionally fully abstract.*

Proof. Lemma 7.38 directly entails preservation and reflection of fair testing equivalence. Regarding surjectivity up to $\sim_f^{\mathcal{M}}$, for any $M \in \mathcal{M}_0$, we have $M \sim_f^{\mathcal{M}} \mathsf{T}(Z((M)))$. Indeed, for any M' , we have

$$\begin{aligned} M @ M' \in \perp & \\ \text{iff } Z(M) @ Z(M') \in \perp & \\ & \text{(because } Z(M) @ Z(M') \bowtie M @ M' \text{ and by Lemma 7.37)} \\ \text{iff } \mathsf{T}(Z(M)) @ M' \in \perp & \\ & \text{(because } Z(M) @ Z(M') \bowtie \mathsf{T}(Z(M)) @ M' \text{ and by Lemma 7.37),} \end{aligned}$$

as desired. \square

We are now able to prove our main result:

Proof of Theorem 6.25. By Theorem 7.39, T is intensionally fully abstract for $\sim_f^{\mathcal{M}}$, so by Lemma 7.32 $\llbracket - \rrbracket$ is intensionally fully abstract for $\sim_f^{\mathcal{S}}$. We thus conclude by Corollary 7.22. \square

7.4. Generalisation. We now show that our main results generalise beyond fair testing equivalence. Indeed, let us put:

Definition 7.40. A *pole* is a property of states over $\text{fc}(\Sigma)$ which is stable under strong bisimilarity.

There is a slight size issue in this definition, as it quantifies over elements of all graphs over $\text{fc}(\Sigma)$. The reader may understand this using whatever fix they prefer, e.g., using a universe or some modal logic.

Example 7.41. Consider any $x \in G$ over $\text{fc}(\Sigma)$. We have

- x is in the pole for fair testing equivalence iff for all $x \leftarrow x'$ there exists $x' \overset{\heartsuit}{\leftarrow} x''$;
- x is in the pole for may testing equivalence iff there exists $x \overset{\heartsuit}{\leftarrow} x'$.

Must testing equivalence is less easy to capture, for reasons explained in [41]. Here is an exotic, yet perhaps relevant pole: x is in it iff for all finite, not-necessarily silent transition sequences $x \leftarrow^* x'$, there exists $x' \overset{\heartsuit}{\leftarrow} x''$. In other words, x never loses the ability to tick. The induced equivalence is clearly at least as fine as fair testing equivalence, but we leave open the question of whether or not it is strictly finer.

Definition 7.42. For any such pole \perp , let \sim_{\perp} denote the testing equivalence induced by replacing \perp^G by \perp in the definition of fair testing equivalence (Definition 2.14).

Semantic testing equivalence may then be taken to be testing equivalence on \mathcal{C} (Definition 7.7), and we get the exact analogue of Theorem 6.25 (without changing the model in any way).

8. CONCLUSION AND FUTURE WORK

We have described our playground for π and the induced sheaf model, which we have proved intensionally fully abstract for a wide range of testing equivalences.

Regarding future work: our proof that traces form a playground uses a new technique based on factorisation systems. Since submission of this paper, we have designed [22, 23] a general setting where this technique applies, and used it to bridge the gap between our notion of plays based on string diagrams and the standard one based on justified sequences [71]. We also consider applying our notion of trace to error diagnostics [32] or efficient machine representation of reversible π -calculus processes [19]. Longer-term directions include applying the approach to more complex calculi, e.g., calculi with passivation [50] or functional calculi, and eventually consider some full-fledged functional language with concurrency primitives. Finally, deriving the complex notion of trace evoked in Section 1.5 from the one exposed here is akin to deriving LTSS from reduction rules [48, 65]. Since the issue still seems easier on traces than on a full operational semantics specification, this might be worthwhile to investigate further. In the same vein, the emphasis we put on traces suggests that we might be able to deduce properties of type systems (soundness, progress, etc) or compilers (correctness) from corresponding properties on traces.

REFERENCES

- [1] LICS 1996. *Proc. 11th Symposium on Logic in Computer Science*, 1996. IEEE.
- [2] LICS 2015. *Proc. 30th Symposium on Logic in Computer Science*, 2015. IEEE.
- [3] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2):409–470, 2000. doi:10.1006/inco.2000.2930.
- [4] Jiří Adámek and Jiří Rosický. *Locally Presentable and Accessible Categories*. Cambridge University Press, 1994. doi:10.1017/CBO9780511600579.
- [5] Gérard Berry and Gérard Boudol. The chemical abstract machine. In *Proc. 17th International Symposium on Principles of Programming Languages*, pages 81–94, 1990. doi:10.1145/96709.96717.

- [6] Filippo Bonchi, Marcello M. Bonsangue, Jan J. M. M. Rutten, and Alexandra Silva. Deriving syntax and axioms for quantitative regular behaviours. In *Proc. 21st International Conference on Concurrency Theory*, volume 5710 of *LNCS*, pages 146–162. Springer, 2009. doi:10.1007/978-3-642-04081-8_11.
- [7] Michele Boreale and Davide Sangiorgi. A fully abstract semantics for causality in the π -calculus. *Acta Informatica*, 35(5):353–400, 1998.
- [8] A. K. Bousfield. Constructions of factorization systems in categories. *Journal of Pure and Applied Algebra*, 9(2-3):287–329, 1977.
- [9] Ed Brinksma, Arend Rensink, and Walter Vogler. Fair testing. In *CONCUR 1995* [17], pages 313–327. ISBN 3-540-60218-6. doi:10.1007/3-540-60218-6_23.
- [10] Roberto Bruni, Hernán C. Melgratti, and Ugo Montanari. Event structure semantics for nominal calculi. In *Proc. 18th International Conference on Concurrency Theory*, volume 4137 of *LNCS*, pages 295–309. Springer, 2006. ISBN 3-540-37376-4. doi:10.1007/11817949_20.
- [11] Nadia Busi and Roberto Gorrieri. A Petri net semantics for π -calculus. In *CONCUR 1995* [17], pages 145–159. ISBN 3-540-60218-6. doi:10.1007/3-540-60218-6_11.
- [12] Nadia Busi and Roberto Gorrieri. Distributed semantics for the π -calculus based on Petri nets with inhibitor arcs. *Journal of Logic and Algebraic Programming*, 78(3):138–162, 2009. doi:10.1016/j.jlap.2008.08.002.
- [13] Diletta Cacciagrano, Flavio Corradini, and Catuscia Palamidessi. Explicit fairness in testing semantics. *Logical Methods in Computer Science*, 5(2), 2009. doi:10.2168/LMCS-5(2:15)2009.
- [14] Simon Castellan, Pierre Clairambault, and Glynn Winskel. The parallel intensionally fully abstract games model of pcf. In *LICS 2015* [2].
- [15] Gian Luca Cattani and Peter Sewell. Models for name-passing processes: Interleaving and causal. In *Proc. 15th Symposium on Logic in Computer Science*, pages 322–333. IEEE, 2000. ISBN 0-7695-0725-5. doi:10.1109/LICS.2000.855781.
- [16] Gian Luca Cattani, Ian Stark, and Glynn Winskel. Presheaf models for the π -calculus. In *Category Theory and Computer Science*, volume 1290 of *LNCS*, pages 106–126. Springer, 1997. ISBN 3-540-63455-X. doi:10.1007/BFb0026984.
- [17] *CONCUR 1995. Proc. 6th International Conference on Concurrency Theory*, volume 962 of *LNCS*, 1995. Springer. ISBN 3-540-60218-6.
- [18] Silvia Crafa, Daniele Varacca, and Nobuko Yoshida. Event structure semantics of parallel extrusion in the pi-calculus. In *Proc. 15th Foundations of Software Science and Computational Structures*, volume 7213 of *LNCS*, pages 225–239. Springer, 2012. ISBN 978-3-642-28728-2. doi:10.1007/978-3-642-28729-9_15.
- [19] Ioana Cristescu, Jean Krivine, and Daniele Varacca. A compositional semantics for the reversible π -calculus. In *Proc. 28th Symposium on Logic in Computer Science*, pages 388–397. IEEE, 2013. ISBN 978-1-4799-0413-6. doi:10.1109/LICS.2013.45.
- [20] Rocco De Nicola and Matthew Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1984. doi:10.1016/0304-3975(84)90113-0.
- [21] Pierpaolo Degano and Corrado Priami. Non-interleaving semantics for mobile processes. *Theoretical Computer Science*, 216(1-2):237–270, 1999. doi:10.1016/S0304-3975(99)80003-6.
- [22] Clovis Eberhart and Tom Hirschowitz. Justified sequences in string diagrams: a comparison between two approaches to concurrent game semantics. Preprint, 2016.

- [23] Clovis Eberhart and Tom Hirschowitz. Justified sequences in string diagrams: a comparison between two approaches to concurrent game semantics. In *Proc. 7th International Conference on Algebra and Coalgebra in Computer Science*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. To appear.
- [24] Charles Ehresmann. Catégories structurées. *Annales scientifiques de l'Ecole Normale Supérieure*, 80(4):349–426, 1963.
- [25] Charles Ehresmann. *Catégories et structures*. Dunod, 1965.
- [26] Joost Engelfriet. A multiset semantics for the pi-calculus with replication. *Theoretical Computer Science*, 153(1&2):65–94, 1996. doi:10.1016/0304-3975(95)00118-2.
- [27] Marcelo P. Fiore and Daniele Turi. Semantics of name and value passing. In *Proc. 16th Symposium on Logic in Computer Science*, pages 93–104. IEEE, 2001. ISBN 0-7695-1281-X. doi:10.1109/LICS.2001.932486.
- [28] Marcelo P. Fiore, Eugenio Moggi, and Davide Sangiorgi. A fully-abstract model for the pi-calculus (extended abstract). In *LICS 1996 [1]*, pages 43–54. doi:10.1109/LICS.1996.561302.
- [29] P. J. Freyd and G. M. Kelly. Categories of continuous functors, I. *Journal of Pure and Applied Algebra*, 2:169–191, 1972. doi:10.1016/0022-4049(72)90001-1.
- [30] Richard H. G. Garner. *Polycategories*. PhD thesis, University of Cambridge, 2006.
- [31] Jean-Yves Girard. Locus solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3):301–506, 2001. doi:10.1007/3-540-44802-0.3.
- [32] Gregor Gösler, Daniel Le Métayer, and Jean-Baptiste Raclet. Causality analysis in contract violation. In *Proc. of Runtime Verification*, volume 6418 of *LNCS*, pages 270–284. Springer, 2010. ISBN 978-3-642-16611-2. doi:10.1007/978-3-642-16612-9_21.
- [33] Marco Grandis and Robert Paré. Limits in double categories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 40(3):162–220, 1999.
- [34] Marco Grandis and Robert Paré. Adjoints for double categories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 45(3):193–240, 2004.
- [35] Russell Harmer, Martin Hyland, and Paul-André Mellès. Categorical combinatorics for innocent strategies. In *Proc. 22nd Symposium on Logic in Computer Science*, pages 379–388. IEEE, 2007. doi:10.1109/LICS.2007.14.
- [36] Matthew Hennessy. A fully abstract denotational semantics for the π -calculus. *Theoretical Computer Science*, 278(1-2):53–89, 2002. doi:10.1016/S0304-3975(00)00331-5.
- [37] Thomas T. Hildebrandt. Towards categorical models for fairness: fully abstract presheaf semantics of SCCS with finite delay. *Theoretical Computer Science*, 294(1/2):151–181, 2003. doi:10.1016/S0304-3975(01)00247-X.
- [38] Tom Hirschowitz. Full abstraction for fair testing in CCS. In *Proc. 5th International Conference on Algebra and Coalgebra in Computer Science*, volume 8089 of *LNCS*, pages 175–190. Springer, 2013. doi:10.1007/978-3-642-40206-7_14.
- [39] Tom Hirschowitz. Full abstraction for fair testing in CCS (expanded version). *Logical Methods in Computer Science*, 10(4), 2014. doi:10.2168/LMCS-10(4:2)2014.
- [40] Tom Hirschowitz and Damien Pous. Innocent strategies as presheaves and interactive equivalences for CCS. In *Proc. of Interaction and Concurrency Experience*, pages 2–24. Electronic Proceedings in Theoretical Computer Science, 2011. doi:10.4204/EPTCS.59.2.
- [41] Tom Hirschowitz and Damien Pous. Innocent strategies as presheaves and interactive equivalences for CCS. *Scientific Annals of Computer Science*, 22(1):147–199, 2012. doi:10.7561/SACS.2012.1.147. Selected papers from ICE '11.

- [42] J. M. E. Hyland and C.-H. L. Ong. On full abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285–408, 2000. doi:10.1006/inco.2000.2917.
- [43] Bart Jacobs. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.
- [44] Lalita Jategaonkar Jagadeesan and Radha Jagadeesan. Causality and true concurrency: A data-flow analysis of the pi-calculus (extended abstract). In *Proc. of Algebraic Methodology and Software Technology*, volume 936 of *LNCS*, pages 277–291. Springer, 1995. doi:10.1007/3-540-60043-4_59.
- [45] André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation and open maps. In *Proc. 8th Symposium on Logic in Computer Science*, pages 418–427. IEEE, 1993. doi:10.1109/LICS.1993.287566.
- [46] Stephen Lack and Paweł Sobociński. Adhesive categories. In *Proc. 7th Foundations of Software Science and Computational Structures*, volume 2987 of *LNCS*, pages 273–288. Springer, 2004. doi:10.1007/978-3-540-24727-2_20.
- [47] James Laird. Game semantics for higher-order concurrency. In *Proc. 26th Foundations of Software Technology and Theoretical Computer Science*, volume 4337 of *LNCS*, pages 417–428. Springer, 2006. doi:10.1007/11944836_38.
- [48] James J. Leifer and Robin Milner. Deriving bisimulation congruences for reactive systems. In *Proc. 11th International Conference on Concurrency Theory*, volume 1877 of *LNCS*, pages 243–258. Springer, 2000. ISBN 3-540-67897-2. doi:10.1007/3-540-44618-4_19.
- [49] Tom Leinster. *Higher Operads, Higher Categories*, volume 298 of *London Mathematical Society Lecture Notes*. Cambridge University Press, Cambridge, 2004.
- [50] Sergueï Lenglet, Alan Schmitt, and Jean-Bernard Stefani. Normal bisimulations in calculi with passivation. In *Proc. 12th Foundations of Software Science and Computational Structures*, volume 5504 of *LNCS*, pages 257–271. Springer, 2009. doi:10.1007/978-3-642-00596-1_19.
- [51] Paul Blain Levy. Morphisms between plays. GaLoP, 2013.
- [52] Saunders Mac Lane. Duality for groups. *Bulletin of the American Mathematical Society*, 56, 1950.
- [53] Saunders Mac Lane. *Categories for the Working Mathematician*. Number 5 in Graduate Texts in Mathematics. Springer, 2nd edition, 1998.
- [54] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Universitext. Springer, 1992.
- [55] Guy McCusker, John Power, and Cai Wingfield. A graphical foundation for schedules. *Electronic Notes in Theoretical Computer Science*, 286:273–289, 2012. doi:10.1016/j.entcs.2012.08.018.
- [56] Paul-André Mellies. Asynchronous games 2: the true concurrency of innocence. In *Proc. 15th International Conference on Concurrency Theory*, volume 3170 of *LNCS*, pages 448–465. Springer, 2004. doi:10.1007/978-3-540-28644-8_29.
- [57] Paul-André Mellies. Asynchronous games 4: A fully complete model of propositional linear logic. In *Proc. 20th Symposium on Logic in Computer Science*, pages 386–395. IEEE, 2005. ISBN 0-7695-2266-1. doi:10.1109/LICS.2005.6.
- [58] Paul-André Mellies. Game semantics in string diagrams. In *Proc. 27th Symposium on Logic in Computer Science*, pages 481–490. IEEE, 2012. doi:10.1109/LICS.2012.58.
- [59] Ugo Montanari and Marco Pistore. Concurrent semantics for the π -calculus. *Electronic Notes in Theoretical Computer Science*, 1:411–429, 1995. doi:10.1016/S1571-0661(04)00024-6.

- [60] Ugo Montanari and Marco Pistore. Structured coalgebras and minimal HD-automata for the π -calculus. *Theoretical Computer Science*, 340(3):539–576, 2005. doi:10.1016/j.tcs.2005.03.014.
- [61] V. Natarajan and Rance Cleaveland. Divergence and fair testing. In *Proc. 22nd International Colloquium on Automata, Languages and Programming*, volume 944 of *LNCS*, pages 648–659. Springer, 1995. ISBN 3-540-60084-1. doi:10.1007/3-540-60084-1_112.
- [62] Hanno Nickau. Hereditarily sequential functionals. In *Proc. Logical Foundations of Computer Science*, volume 813 of *LNCS*, pages 253–264. Springer, 1994. doi:10.1007/3-540-58140-5_25.
- [63] Mogens Nielsen, Gordon Plotkin, and Glynn Winskel. Event structures and domains, part 1. *Theoretical Computer Science*, 13:65–108, 1981.
- [64] Robert Paré. Yoneda theory for double categories. *Theory and Applications of Categories*, 25(17):436–489, 2011.
- [65] Julian Rathke and Paweł Sobociński. Deconstructing behavioural theories of mobility. In *IFIP TCS*, volume 273 of *IFIP*, pages 507–520. Springer, 2008. doi:10.1007/978-0-387-09680-3_34.
- [66] Arend Rensink and Walter Vogler. Fair testing. *Information and Computation*, 205(2):125–198, 2007. doi:10.1016/j.ic.2006.06.002.
- [67] Silvain Rideau and Glynn Winskel. Concurrent strategies. In *Proc. 26th Symposium on Logic in Computer Science*, pages 409–418. IEEE, 2011. doi:10.1109/LICS.2011.13.
- [68] Davide Sangiorgi and Jan Rutten, editors. *Advanced Topics in Bisimulation and Coinduction*. Number 52 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2011.
- [69] Davide Sangiorgi and David Walker. *The π -calculus - a theory of mobile processes*. Cambridge University Press, 2001.
- [70] Ian Stark. A fully abstract domain model for the π -calculus. In *LICS 1996* [1], pages 36–42. doi:10.1109/LICS.1996.561301.
- [71] Takeshi Tsukada and C.-H. Luke Ong. Nondeterminism in game semantics via sheaves. In *LICS 2015* [2].
- [72] Glynn Winskel. Event structure semantics for CCS and related languages. In *Proc. 9th International Colloquium on Automata, Languages and Programming*, volume 140 of *LNCS*, pages 561–576. Springer, 1982. ISBN 3-540-11576-5. doi:10.1007/BFb0012800.
- [73] Glynn Winskel. Strategies as profunctors. In *Proc. 16th Foundations of Software Science and Computational Structures*, volume 7794 of *LNCS*, pages 418–433. Springer, 2013. doi:10.1007/978-3-642-37075-5_27.

APPENDIX A. PROOF OF LEMMA 7.37

One wants to check two properties:

- (LH) for all transitions $C' \xrightarrow{a}_A C$ with $C \bowtie M$, there exists $M' \xrightarrow{a}_A M$ with $C' \bowtie M'$;
 (RH) for all transitions $M \xleftarrow{a}_A M'$ with $C \bowtie M$, there exists $C \xleftarrow{a}_A C'$ with $C' \bowtie M'$.

The attentive reader will have noticed that (LH) imposes y to answer with a single transition. This means we actually prove that \bowtie is an *expansion* [68, Chapter 6]. Any expansion being in particular a weak bisimulation, this suffices.

Notation A.1. We sometimes cast processes P (resp. pairs $D[\sigma]$) over γ into configurations $\langle \gamma \parallel P \rangle$ (resp. mixed behaviours $\langle \gamma \parallel D[\sigma] \rangle$). We proceed similarly for multisets of processes.

We start by proving (LH) for all cases, before proving that (RH) holds as well.

Synchro, (LH). We begin by the case of a synchronisation, i.e., when one has a transition

$$C = \langle \gamma \parallel a(b).P +_{k_1} R_1, \bar{a}\langle c \rangle.Q +_{k_2} R_2 \rangle @ C_0 \xleftarrow{id} \langle \gamma \parallel P[b \mapsto c], Q \rangle @ C_0 = C'.$$

We want to show that there exists a transition $M \xleftarrow{id} M'$ with $C' \bowtie M'$.

We write $P_1 = a(b).P +_{k_1} R_1$ and $P_2 = \bar{a}\langle c \rangle.Q +_{k_2} R_2$. Neither of them are of the form $(\cdot | \cdot)$ so they can only be related to mixed behaviours using the first two rules. Therefore, four sub-cases should be considered, as detailed in Figure 9. If we are in case i_1 for P_1 and i_2 for P_2 , then we have two mixed behaviours $D_1[\sigma_1]$ and $D_2[\sigma_2]$ such that $n_i \vdash_D D_i$, $\sigma_i : n_i \rightarrow \gamma$, and $P_i \bowtie D_i[\sigma_i]$ for $i = 1, 2$, plus $M = \langle \gamma \parallel D_1[\sigma_1], D_2[\sigma_2] \rangle @ M_0$ with $C_0 \bowtie M_0$.

- **Case 1 for both P_1 and P_2 .** We have $M = \llbracket P'_1 \rrbracket_{h_1}[\sigma_1] @ \llbracket P'_2 \rrbracket_{h_2}[\sigma_2] @ M_0$, and there is a transition

$$M \xleftarrow{id} \llbracket P' \rrbracket_{h'_1}[\sigma'_1] @ \llbracket Q' \rrbracket_{h_2}[\sigma_2] @ M_0,$$

where h'_1 is $\gamma'_1, b' \xrightarrow{h_1+!} n_1 + 1$ and σ'_1 is $n_1 + 1 \xrightarrow{[\sigma_1, c]} \gamma$.

Since $\sigma'_1 \circ h'_1$ equals $\gamma'_1 \xrightarrow{h_1+!} n_1 + 1 \xrightarrow{\sigma_1 + [b]} (\gamma, b) \xrightarrow{b \mapsto c} \gamma$, we have that $P'[\sigma'_1 \circ h'_1] = P'[(\sigma_1 + [b]) \circ (h_1+!)] [b \mapsto c] = P[b \mapsto c]$, and therefore $P[b \mapsto c] \bowtie \llbracket P' \rrbracket_{h'_1}[\sigma'_1]$. Moreover, it is clear that $Q = Q'[\sigma_2 \circ h_2] \bowtie \llbracket Q' \rrbracket_{h_2}[\sigma_2]$, and finally $\llbracket P[b \mapsto c], Q \rrbracket @ C_0 \bowtie M'$.

- **Case 1 for P_1 , Case 2 for P_2 .** We have a transition

$$M \xleftarrow{id} \llbracket P' \rrbracket_{h'_1}[\sigma'_1] @ (D_2 \cdot o_{n_2, a_2, c_2})_{|j}[\sigma_2] @ M_0 = M',$$

where h'_1 is $\gamma'_1, b' \xrightarrow{h_1+!} n_1 + 1$ and σ'_1 is $n_1 + 1 \xrightarrow{[\sigma_1, c]} \gamma$.

As is the previous case, one can check that $P[b \mapsto c] \bowtie \llbracket P' \rrbracket_{h'_1}[\sigma'_1]$. Moreover, $Q \bowtie (D_2 \cdot o_{n_2, a_2, c_2})_{|j}[\sigma_2]$ and therefore $\llbracket P[b \mapsto c], Q \rrbracket @ C_0 \bowtie M'$.

- **Case 2 for P_1 , Case 1 for P_2 .** We have a transition

$$M \xleftarrow{id} (D_1 \cdot \iota_{n_1, a_1})_{|i} [n_1 + 1 \xrightarrow{[\sigma_1, c]} \gamma] @ \llbracket Q' \rrbracket_{h'_2}[\sigma'_2] @ M_0 = M'.$$

As in the first case, we have $Q = Q'[\sigma_2 \circ h_2] \bowtie \llbracket Q' \rrbracket_{h_2}[\sigma_2]$. Furthermore, since

$$\begin{aligned} & \zeta((D_1 \cdot \iota_{n_1, a_1})_{|i} [n_1 + 1 \xrightarrow{[\sigma_1, c]} \gamma]) \\ &= \zeta((D_1 \cdot \iota_{n_1, a_1})_{|i} [n_1 + 1 \xrightarrow{\sigma_1 + [b]} \gamma, b] [b \mapsto c]) \\ &= P[b \mapsto c], \end{aligned}$$

	Case 1	Case 2
P_1	There exist $\gamma'_1 \vdash P'_1 = a'_1(b').P' +_{k_1} R'_1$ and $h_1: \gamma'_1 \cong n_1$ such that $P_1 = P'_1[\sigma_1 \circ h_1] \quad \sigma_1(h_1(a'_1)) = a$ $P = P'[\gamma'_1, b' \xrightarrow{h_1+!} n_1 + 1 \xrightarrow{\sigma_1+!b'} \gamma, b]$ $D_1 = \llbracket P'_1 \rrbracket_{h_1} \quad R_1 = R'_1[\sigma_1 \circ h_1].$	There exist $n_1 \vdash D_1$, $a_1 \in n_1$, and $i \in D_1 \cdot \iota_{n_1, a_1} $ such that $\sigma_1(a_1) = a \quad P_1 = \zeta(D_1)[\sigma_1]$ $P = \zeta((D_1 \cdot \iota_{n_1, a_1}) _i)[n_1 + 1 \xrightarrow{\sigma_1+!b'} \gamma, b].$
P_2	There exist $\gamma'_2 \vdash P'_2 = \bar{a}'_2(c').Q' +_{k_2} R'_2$ and $h_2: \gamma'_2 \cong n_2$ such that $P_2 = P'_2[\sigma_2 \circ h_2]$ $Q = Q'[\gamma'_2, b \xrightarrow{h_2+1} n_2 + 1 \xrightarrow{\sigma_2+!b'} \gamma, b]$ $\sigma_2(h_2(a'_2)) = a \quad \sigma_2(h_2(c'_2)) = c$ $D_2 = \llbracket P'_2 \rrbracket_{h_2} \quad R_2 = R'_2[\sigma_2 \circ h_2].$	There exist $n_2 \vdash D_2$, $a_2, c_2 \in n_2$, and $j \in D_2 \cdot o_{n_2, a_2, c_2} $ such that $\sigma_2(a_2) = a \quad \sigma_2(c_2) = c$ $P_2 = \zeta(D_2)[\sigma_2]$ $Q = \zeta((D_2 \cdot o_{n_2, a_2, c_2}) _j)[\sigma_2].$

Figure 9: Synchro, (LH) cases

we have that $[P[b \mapsto c], Q]@C_0 \varkappa M'$.

- **Case 2 for both P_1 and P_2 .** We have a transition

$$M \xleftarrow{id} (D_1 \cdot \iota_{n_1, a_1})|_i[n_1 + 1 \xrightarrow{[\sigma_1, !c]} \gamma]@(D_2 \cdot o_{n_2, a_2, c_2})|_j[\sigma_2]@M_0 = M'.$$

As in the previous cases, one can show that $[P[b \mapsto c], Q]@C_0 \varkappa M'$.

Heating, (LH). We now consider the case of heating, i.e., when one has a transition

$$C = (P | Q)@C_0 \xleftarrow{\tau} [P, Q]@C_0 = C'.$$

We want to show that there exists a transition $M \xleftarrow{\tau} M'$ with $[P, Q]@C_0 \varkappa M'$.

We now have to consider a few cases, depending on which rule is applied for $P | Q$ in the proof of $(P | Q)@C_0 \varkappa M$. Notice that $P | Q$ cannot be of the form $\zeta(D)[\sigma]$. We are therefore left with two cases, depending on whether the first or third rule is applied.

- If the first rule is applied, we find M_0 , $\gamma' \vdash P' | Q'$, $h: \gamma' \cong n$, and $\sigma: n \rightarrow \gamma$ such that $P | Q = (P' | Q')[\sigma \circ h]$ and $M = \llbracket P' | Q' \rrbracket_h[\sigma]@M_0$, with $C_0 \varkappa M_0$. Letting $D = \llbracket P' | Q' \rrbracket_h$, we notice that

$$D = \langle \pi_n^l \mapsto \llbracket P' \rrbracket_h, \pi_n^r \mapsto \llbracket Q' \rrbracket_h \rangle.$$

Thus, there is a transition

$$M \xleftarrow{\tau} M' = \llbracket \llbracket P' \rrbracket_h[\sigma], \llbracket Q' \rrbracket_h[\sigma] \rrbracket @M_0$$

with $[P, Q]@C_0 \varkappa M'$, as desired.

- If the third rule is applied, we find M_0 , $n \vdash D^1, D^2$ and $\sigma: n \rightarrow \gamma$ such that

$$P | Q = (\zeta(D^1) | \zeta(D^2))[\sigma]$$

and $M = [D^1[\sigma], D^2[\sigma]]@M_0$. We notice that $[P, Q] = [\zeta(D^1)[\sigma], \zeta(D^2)[\sigma]]$, so $[P, Q] \varkappa [D^1[\sigma], D^2[\sigma]]$, hence $[P, Q]@C_0 \varkappa M$. The identity transition $M \xleftarrow{\tau} M$ thus fits our needs.

Nu, (LH). We now consider the case of a ν rule, i.e., when one has a transition

$$C = \langle \gamma \| \nu a.P +_k R \rangle @C_0 \xleftarrow{\tau} \langle \gamma, a \| P \rangle @C_0[\gamma \subset \gamma, a] = C'.$$

We want to show that there exists a transition $M \xleftarrow{\tau} M'$ with $\langle \gamma, a \| P \rangle @C_0[\gamma \subset \gamma, a] \varkappa M'$. We notice that $\nu a.P +_k R$ cannot be obtained from the third rule. We thus consider two cases corresponding to the first and second rules.

- If the first rule is applied, there exist M_0 , $\gamma' \vdash \nu a.P' +_k R'$, $h: \gamma' \cong n$, and $\sigma: n \rightarrow \gamma$ such that $M = [\nu a.P' +_k R']_h[\sigma]@M_0$ and

$$\nu a.P +_k R = (\nu a.P' +_k R')[\sigma \circ h].$$

We write $D = [\nu a.P' +_k R']_h$ as $\langle \nu_n \mapsto [P']_{h'} \rangle \boxplus [R']_h$, where h' is $\gamma', a \xrightarrow{h+1} n+1$. Thus, there is a transition

$$M \xleftarrow{\tau} M' = \langle \gamma, a \| [P']_{h'}[n+1 \xrightarrow{\sigma+a'} \gamma, a] \rangle @M_0[\gamma \subset \gamma, a]$$

and, modulo the fact that $C \varkappa M$ implies $C[\sigma] \varkappa M[\sigma]$, we have $P@C_0[\gamma \subset \gamma, a] \varkappa [P']_{h'}[\sigma + \ulcorner a \urcorner]@M_0[\gamma \subset \gamma, a]$ since $P = P'[\gamma', a \xrightarrow{h'} n+1 \xrightarrow{\sigma+1} \gamma, a]$, as desired.

- If the second rule is applied, there exist M_0 , $n \vdash D$, and $\sigma: n \rightarrow \gamma$ such that $M = \zeta(D)[\sigma]@M_0$ and

$$\nu a.P +_k R = \zeta(D)[\sigma].$$

Thus, there exists $i \in |D \cdot \nu_n|$ such that $\zeta((D \cdot \nu_n)_i)[n+1 \xrightarrow{\sigma+[a]} \gamma, a] = P$. There is thus a transition

$$M \xleftarrow{\tau} M' = \langle \gamma, a \| (D \cdot \nu_n)_i[\sigma + \ulcorner a \urcorner] \rangle @M_0[\gamma \subset \gamma, a]$$

with $P@C_0[\gamma \subset \gamma, a] \varkappa M'$, as desired.

Tick and Tau, (LH). We now consider the cases \heartsuit and τ , i.e., when one has a transition

$$C = (\xi.P +_k R)@C_0 \xleftarrow{\xi} P@C_0,$$

where $\xi \in \{\heartsuit, \tau\}$. We want to show that there exists a transition $M \xleftarrow{\xi} M'$ with $P@C_0 \varkappa M'$.

Once again, the third rule could not have been applied, and we are left with two cases corresponding to the first and second rules.

- If the first rule is applied, then we find M_0 , $\gamma' \vdash \xi.P' +_k R'$, $h: \gamma' \cong n$, and $\sigma: n \rightarrow \gamma$ such that $M = [\xi.P' +_k R']_h[\sigma]@M_0$, $C_0 \varkappa M_0$, and

$$\xi.P +_k R = (\xi.P' +_k R')[\sigma \circ h].$$

We write $D = [\xi.P' +_k R']_h$ as $\langle \xi_n \mapsto [P']_h \rangle \boxplus [R']_h$. Thus, there is a transition

$$M = D[\sigma]@M_0 \xleftarrow{\xi} [P']_h[\sigma]@M_0 = M',$$

with $P = P'[\sigma \circ h]$ and thus $P@C_0 \varkappa M'$, as desired.

- If the second rule is applied, then we find M_0 , $n \vdash D$, and $\sigma: n \rightarrow \gamma$ such that $C_0 \bowtie M_0$, $M = D[\sigma]@M_0$, and

$$\xi.P +_k R = \zeta(D)[\sigma].$$

Then, there exists $i \in |D \cdot \xi_n|$ such that $\zeta((D \cdot \xi_n)|_i)[\sigma] = P$. Hence, there is a transition

$$M \xleftarrow{\xi} (D \cdot \xi_n)|_i[\sigma]@M_0 = M'$$

with $P@C_0 \bowtie M'$, as desired.

We have thus proved that (LH) holds. We now proceed with the case analysis for (RH).

Synchro, (RH). We start, once again, with the case of the synchronisation, i.e., when M has the shape $\langle \gamma \| D_1[\sigma_1], D_2[\sigma_2] \rangle @M_0$ and we consider a silent transition to

$$M' = \langle \gamma \| (D_1 \cdot \iota_{n_1, a_1})|_i[n_1 + 1 \xrightarrow{[\sigma_1, \sigma_2(b_2)]} \gamma], (D_2 \cdot o_{n_2, a_2, b_2})|_j[\sigma_2] \rangle @M_0,$$

where $n_i \vdash D_i$ and $\sigma_i: n_i \rightarrow \gamma$ for $i = 1, 2$, $\sigma_1(a_1) = a = \sigma_2(a_2)$, and $\sigma_2(c_2) = c$. We want to show that there exists a transition $C \xleftarrow{id} C'$ with $C' \bowtie M'$. There are exactly 10 cases here. Firstly, we have the case where the third rule is applied to $D_1[\sigma_1], D_2[\sigma_2]$. Otherwise, one could have used each of the three rules for \bowtie_0 for each of D_1 and D_2 , yielding nine cases. We start with the first case, and then treat the nine others.

- If the third rule is applied on $D_1[\sigma_1], D_2[\sigma_2]$ (hence $n_1 = n_2 = n$ and $\sigma_1 = \sigma_2 = \sigma$), then we find C_0 such that $C_0 \bowtie M_0$ and $C = (\zeta(D_1) | \zeta(D_2))[\sigma]@C_0$. Then, we have a transition

$$(\zeta(D_1) | \zeta(D_2))[\sigma]@C_0 \xleftarrow{id} [\zeta(D_1)[\sigma_1], \zeta(D_2)[\sigma_2]]@C_0.$$

Since the latter configuration is again related to M , this reduces to the case where the second rule is applied for both $D_1[\sigma_1]$ and $D_2[\sigma_2]$.

- If the third rule is applied for $D_1[\sigma_1]$ and any of the three rules is applied for $D_2[\sigma_2]$, then we find P, M_1, M_2, C_0 , and $n_1 \vdash D_3$ such that M_1 has length 1 if the third rule is also used for D_2 and 0 otherwise, $M_0 = D_3[\sigma_3]@M_1@M_2$, $P \bowtie D_2[\sigma_2]@M_1$, $C_0 \bowtie M_2$, and $C = (\zeta(D_1) | \zeta(D_3))[\sigma_1]@P@C_0$.

Thus, there is a transition

$$(\zeta(D_1) | \zeta(D_3))[\sigma_1]@P@C_0 \xleftarrow{id} [\zeta(D_1)[\sigma_1], \zeta(D_3)[\sigma_1]]@P@C_0$$

which reduces this case to the one where the second rule is applied to D_1 .

- If the third rule is applied for $D_2[\sigma_2]$ and any of the first two rules is applied for D_1 , then we find M_1, P, C_0 , and $n_2 \vdash D_3$ such that $M_0 = D_3[\sigma_3]@M_1$, $P \bowtie D_1[\sigma_1]$, $C_0 \bowtie M_1$, and $C = (\zeta(D_2) | \zeta(D_3))[\sigma_2]@P@C_0$. Again, we are reduced to the case where the second rule is applied for D_2 , using the transition

$$(\zeta(D_2) | \zeta(D_3))[\sigma_2]@P@C_0 \xleftarrow{id} [\zeta(D_2)[\sigma_2], \zeta(D_3)[\sigma_2]]@P@C_0.$$

In the remaining cases, we have $C = P_1^0@P_2^0@C_0$ with $P_i^0 \bowtie D_i[\sigma_i]$ for $i = 1, 2$, and $C_0 \bowtie M_0$. We thus have four cases, described in Figure 10 just as we did in Figure 9.

- If the second rule is applied for both D_1 and D_2 , then there is a transition

$$C \xleftarrow{id} C' = \zeta((D_1 \cdot \iota_{n_1, a_1})|_i)[[\sigma_1, [c]]]@ \zeta((D_2 \cdot o_{n_2, a_2, c_2})|_j)[\sigma_2]@M_0$$

with $C' \bowtie M'$ as desired.

	Case 1	Case 2
D_1	<p>There exist $\gamma'_1 \vdash P_1 = a'_1(b).P'_1 +_{k_1} R_1$ and $h_1: \gamma'_1 \cong n_1$ such that</p> $P_1^0 = P_1[\sigma_1 \circ h_1] \quad h_1(a'_1) = a_1$ $D_1 = \llbracket P_1 \rrbracket_{h_1} \quad i \in D_1 \cdot \iota_{n_1, a_1} $ $(D_1 \cdot \iota_{n_1, a_1}) _i = \llbracket P'_1 \rrbracket_{h'_1}$ <p>where h'_1 is $\gamma'_1, b \xrightarrow{h_1+1} n_1 + 1$.</p>	$P_1^0 = \zeta(D_1)[\sigma_1]$.
P_2	<p>There exist $\gamma'_2 \vdash P_2 = a'_2(c').P'_2 +_{k_2} R_2$ and $h_2: \gamma'_2 \cong n_2$ such that</p> $P_2^0 = P_2[\sigma_2 \circ h_2] \quad h_2(a'_2) = a_2$ $h_2(c'_2) = c_2 \quad D_2 = \llbracket P_2 \rrbracket_{h_2}$ $j \in D_2 \cdot o_{n_2, a_2, c_2} $ $(D_2 \cdot o_{n_2, a_2, c_2}) _j = \llbracket P'_2 \rrbracket_{h_2}$	$P_2^0 = \zeta(D_2)[\sigma_2]$.

Figure 10: Synchro, (RH) cases

- If we apply the second rule for D_1 and the first rule for D_2 , then there is a transition

$$C \xleftarrow{id} C' = \zeta((D_1 \cdot \iota_{n_1, a_1})|_i)[[\sigma_1, [c]]] @ P'_2[\sigma_2 \circ h_2] @ C_0$$

with $C' \varkappa M'$ as desired.

- If the first rule is applied for both D_1 and D_2 , then we have a synchronisation between $P_1[\sigma_1 \circ h_1]$ and $P_2[\sigma_2 \circ h_2]$. In order to determine the result of this synchronisation we need to choose a representative for $P_1[\sigma_1 \circ h_1]$, i.e., pick a channel for what b becomes after substitution. A reasonable choice here is $\gamma + 1$, we choose

$$P_1[\sigma_1 \circ h_1] = a(\gamma + 1).(P'_1[(\sigma_1 + 1) \circ h'_1] +_{k_1} R_1[\sigma_1 \circ h_1]).$$

We thus have a transition to

$$C' = P'_1[(\sigma_1 + 1) \circ h'_1][\gamma + 1 \mapsto c] @ P'_2[\sigma_2 \circ h_2] @ C_0.$$

But the diagram

$$\begin{array}{ccc} n_1 + 1 & \xrightarrow{\sigma_1 + 1} & \gamma + 1 \\ & \searrow [\sigma_1, \ulcorner c \urcorner] & \downarrow [\gamma + 1 \mapsto c] \\ & & \gamma \end{array}$$

commutes, so $C' = P'_1[[\sigma_1, \ulcorner c \urcorner] \circ h'_1] @ P'_2[\sigma_2 \circ h_2] @ C_0$. Finally, we also know:

$$\begin{aligned} M' &= ((D_1 \cdot \iota_{n_1, a_1})|_i)[[\sigma_1, \ulcorner c \urcorner]] @ (D_2 \cdot o_{n_2, a_2, c_2})|_j[\sigma_2] @ M_0 \\ &= \llbracket P'_1 \rrbracket_{h'_1}[[\sigma_1, \ulcorner c \urcorner]] @ \llbracket P'_2 \rrbracket_{h_2}[\sigma_2] @ M_0, \end{aligned}$$

which entails $C' \varkappa M'$ as desired.

- If we apply the first rule for D_1 and the second rule for D_2 , then, choosing the same representative as before for $P_1[\sigma_1 \circ h_1]$, there is a transition

$$C \xleftarrow{id} P'_1[(\sigma_1 + 1) \circ h'_1][\gamma + 1 \mapsto c] @ \zeta((D_2 \cdot \sigma_{n_2, a_2, c_2})|_j)[\sigma_2] @ C_0 = C'$$

satisfying $C' \varkappa M'$ (for the same reason as in the last case).

Fork, (RH). We consider the case of a forking action, i.e., $M = \langle \gamma \| D[\sigma] \rangle @ M_0$ with $n \vdash_D D$ and $\sigma: n \rightarrow \gamma$, and we have a transition

$$\langle \gamma \| D[\sigma] \rangle @ M_0 \xleftarrow{id} \langle \gamma \| (D \cdot \pi_n^l)|_i[\sigma], (D \cdot \pi_n^r)|_j[\sigma] \rangle @ M_0$$

for some i and j . We want to show that there exists a transition $C \xleftarrow{id} C'$ with $C' \varkappa M'$. We proceed by case analysis on the rule applied for $D[\sigma]$ in the proof of $C \varkappa M$.

- If the first rule is applied, then we find $C_0 \varkappa M_0$, $\gamma' \vdash P = P_1 | P_2$, and $h: \gamma' \cong n$ such that $D = \llbracket P \rrbracket_h$, $C = P[\sigma \circ h] @ C_0$, $(D \cdot \pi_n^l)|_i = \llbracket P_1 \rrbracket_h$, and $(D \cdot \pi_n^r)|_j = \llbracket P_2 \rrbracket_h$.

Thus, there is a transition

$$P[\sigma \circ h] @ C_0 \xleftarrow{id} P_1[\sigma \circ h] @ P_2[\sigma \circ h] @ C_0 = C'$$

with $C' \varkappa M'$ as desired.

- If the second rule is applied, then we find $C_0 \varkappa M_0$ such that $C = \zeta(D)[\sigma] @ C_0$. Thus, $\zeta(D)[\sigma]$ has the shape

$$\tau.(\zeta((D \cdot \pi_n^l)|_i | \zeta((D \cdot \pi_n^r)|_j)))[\sigma] +_k R \tag{A.1}$$

so we have

$$\zeta(D)[\sigma] @ C_0 \xleftarrow{id} (\zeta((D \cdot \pi_n^l)|_i | \zeta((D \cdot \pi_n^r)|_j)))[\sigma] @ C_0 = C'$$

with $C' \varkappa M'$ (using the third rule) as desired.

- If the third rule is applied, then we find $n \vdash_D D'$, M_1 , and C_0 such that $C_0 \varkappa M_1$, $C = (\zeta(D) | \zeta(D'))[\sigma] @ C_0$, and $M_0 = D'[\sigma] @ M_1$. But then, as in the previous case, $\zeta(D)[\sigma]$ has the shape (A.1) and we have transitions:

$$\begin{aligned} (\zeta(D) | \zeta(D'))[\sigma] @ C_0 &\xleftarrow{id} \zeta(D)[\sigma] @ \zeta(D')[\sigma] @ C_0 \\ &\xleftarrow{id^2} \zeta((D \cdot \pi_n^l)|_i)[\sigma] @ \zeta((D \cdot \pi_n^r)|_j)[\sigma] @ \zeta(D')[\sigma] @ C_0 \\ &= C' \end{aligned}$$

with $C' \varkappa M'$ as desired.

Nu, (RH). We consider the case of a ν rule, i.e., one has a transition

$$C = \langle \gamma \| D[\sigma] \rangle @ M_0 \xleftarrow{id} \langle \gamma \| (D \cdot \nu_n)|_i[n + 1 \xrightarrow{\sigma+[a]} \gamma, a] \rangle @ M_0[\gamma \subset \gamma, a]$$

with $n \vdash_D D$ and $\sigma: n \rightarrow \gamma$. We want to show that there exists a transition $C \xleftarrow{id} C'$ with $C' \varkappa M'$. We again proceed by case analysis on the rule applied for $D[\sigma]$.

- If the first rule is applied, then we find $C_0 \varkappa M_0$, $\gamma' \vdash P = \nu a.P' +_k R$, and $h: \gamma' \cong n$ such that $C = P @ C_0$, $D = \llbracket P \rrbracket_h$ and $(D \cdot \nu_n)|_i = \llbracket P' \rrbracket_{h+1}$. There are thus transitions

$$C = P[\sigma \circ h] @ C_0 \xleftarrow{id} P'[\gamma', a \xrightarrow{h+1} n + 1 \xrightarrow{\sigma+[a]} \gamma, a] @ C_0[\gamma \subset \gamma, a] = C'$$

with $C' \varkappa M'$ as desired.

- If the second rule is applied, then we find $C_0 \varkappa M_0$ such that $C = \zeta(D)[\sigma]@C_0$, and there is a transition

$$\zeta(D)[\sigma]@C_0 \xleftarrow{id} \zeta((D \cdot \nu_n)|_i)[n+1] \xrightarrow{\sigma+[a]} \gamma, a @C_0[\gamma \subset \gamma, a] = C'$$

with $C' \varkappa M'$ as desired.

- If the third rule is applied, then we find M_1, C_0 , and $n \vdash D'$ such that $M_0 = D'[\sigma]@M_1$, $C_0 \varkappa M_1$, and $C = (\zeta(D) | \zeta(D'))[\sigma]@C_0$. But then we have

$$\begin{aligned} & (\zeta(D) | \zeta(D'))[\sigma]@C_0 \\ & \xleftarrow{id} \zeta(D)[\sigma]@C_0 \zeta(D')[\sigma]@C_0 \\ & \xleftarrow{id} \zeta((D \cdot \nu_n)|_i)[n+1] \xrightarrow{\sigma+[a]} \gamma, a @C_0[\zeta(D')[\sigma][\gamma \subset \gamma, a]@C_0[\gamma \subset \gamma, a] = C' \end{aligned}$$

and $C' \varkappa M'$ as desired.

Tick and Tau, (RH). We now consider the cases \heartsuit and τ , i.e., when one has a transition

$$\langle \gamma || D[\sigma] \rangle @M_0 \xleftarrow{\xi} \langle \gamma || (D \cdot \xi_n)|_i[\sigma] \rangle @M_0$$

where $\xi \in \{\heartsuit, \tau\}$, with $n \vdash D$ and $\sigma: n \rightarrow \gamma$. We want to show that there exists a transition $C \xleftarrow{\xi} C'$ with $C' \varkappa M'$. Again, we proceed by case analysis on the rule applied for $D[\sigma]$.

- If the first rule is applied, we find $C_0 \varkappa M_0$, $\gamma' \vdash P = \xi.P' +_k R$, and $h: \gamma' \cong n$ such that $C = P[\sigma \circ h]@C_0$, $D = \llbracket P \rrbracket_h[\sigma]$, and $(D \cdot \xi_n)|_i = \llbracket P' \rrbracket_h$. But then we have

$$P[\sigma \circ h]@C_0 \xleftarrow{\xi} P'[\sigma \circ h]@C_0 \varkappa \llbracket P' \rrbracket_h[\sigma]@M_0,$$

as expected.

- If the second rule is applied, then $C = \zeta(D)[\sigma]@C_0$ for some $C_0 \varkappa M_0$ and we have

$$\zeta(D)[\sigma]@C_0 \xleftarrow{\xi} \zeta((D \cdot \xi_n)|_i)[\sigma]@C_0 \varkappa (D \cdot \xi_n)|_i[\sigma]@M_0,$$

as desired.

- Finally, if the third rule is applied, then we find C_0, M_1 , and $n \vdash D'$ such that $C = (\zeta(D) | \zeta(D'))[\sigma]@C_0$, $M_0 = D'[\sigma]@M_1$, and $C_0 \varkappa M_1$. But then, we have

$$(\zeta(D) | \zeta(D'))[\sigma]@C_0 \xleftarrow{id} \zeta(D)[\sigma]@C_0 \zeta(D')[\sigma]@C_0 \xleftarrow{\xi} \zeta((D \cdot \xi_n)|_i)[\sigma]@C_0 \zeta(D')[\sigma]@C_0 = C'$$

with $C' \varkappa M'$ as desired.