Full abstraction for fair testing in CCS

Tom Hirschowitz

Full abstraction for fair testing in CCS

Tom Hirschowitz[⋆]

CNRS and Université de Savoie

Abstract. In previous work with Pous, we defined a semantics for CCS which may both be viewed as an innocent presheaf semantics and as a concurrent game semantics. It is here proved that a behavioural equivalence induced by this semantics on CCS processes is fully abstract for fair testing equivalence.

The proof relies on a new algebraic notion called *playground*, which represents the 'rule of the game'. From any playground, two languages, equipped with labelled transition systems, are derived, as well as a strong, functional bisimulation between them.

Keywords: Programming languages; categorical semantics; presheaf semantics; game semantics; concurrency; process algebra.

1 Introduction

Motivation and previous work Innocent game semantics, invented by Hyland and Ong [20], led to fully abstract models for a variety of functional languages, where programs are interpreted as strategies in a game. Presheaf models [22, 6] were introduced by Joyal et al. as a semantics for process algebras, in particular Milner's CCS [28]. Previous work with Pous [19] (HP) proposes a semantics for CCS, which reconciles these apparently very different approaches. Briefly, (1) on the one hand, we generalise innocent game semantics to both take seriously the possibility of games with more than two players and consider strategies which may accept plays in more than one way; (2) on the other hand, we refine presheaf models to take parallel composition more seriously. This leads to a model of CCS which may both be seen as a concurrent game semantics, and as an innocent presheaf model, as we now briefly recall.

To see that presheaf models are a concurrent, non-innocent variant of game semantics, recall that the base category, say $\mathbb{C}$, for such a presheaf model typically has as objects sequences of labels, or configurations in event structures, morphisms being given by prefix inclusion. Such objects may be understood as plays in some game. Now, in standard game semantics, a strategy is a prefix-closed (non-empty) set of plays. Unfolding the definition, this is the same as a functor $\mathbb{C}^{op} \to 2$, where 2 is the poset category $0 \leqslant 1$: the functor maps a play to 1 when it is accepted by the strategy, and to 0 otherwise. It is known since

[⋆] Partially funded by the French ANR projet blanc "Formal Verification of Distributed Components" PiCoq ANR 2010 BLAN 0305 01 and CNRS PEPS CoGIP.

Harmer and McCusker [15] that this notion of strategy does not easily adapt to non-determinism or concurrency. Presheaf semantics only slightly generalises it by allowing strategies to accept a play in several ways. A strategy S now maps each play p to a *set* $S(p)$. The play is accepted when $S(p)$ is non-empty, and, because there are then no functions $S(p) \to \varnothing$, being accepted remains a prefix-closed property of plays. The passage from 2 to more general sets allows to express branching-time semantics.

This links presheaf models with game models, but would be of little interest without the issue of *innocence*. Game models, indeed, do not always accept *any* prefix-closed set of plays S as a strategy: they demand that any choice of move in S depends only on its *view*. E.g., consider the CCS process $P = (a|(b \oplus c))$, where $\oplus$ denotes internal choice, and a candidate strategy accepting the plays $\epsilon, (a), (b), (c), (ab)$, but not (ac). This strategy refuses to choose c after a has been played. Informally, there are two players here, one playing a and the other playing $b \oplus c$; the latter should have no means to know whether a has been played or not. We want to rule out this strategy on the grounds that it is not innocent.

Our technical solution for doing so is to refine the notion of play, making the number of involved players more explicit. Plays still form a category, but they admit a subcategory of *views*, which represent a single player's possible perceptions of the game. This leads us to two equivalent categories of strategies. In the first, strategies are presheaves on views. In the second category, strategies are certain presheaves on arbitrary plays, satisfying an innocence condition. Parallel composition, in the game semantical sense, is best understood in the former category: it merely amounts to copairing. Parallel composition, in the CCS sense, which in standard presheaf models is a complex operation based on some labelling of transitions or events, is here just a move in the game. The full category of plays is necessary for understanding the global behaviour of strategies. It is in particular needed to define our semantic variant of fair testing equivalence, described below. One may think of presheaves on views as a syntax, and of innocent presheaves on plays as a semantics. The combinatorics of passing from local (views) to global (arbitrary plays) are dealt with by right Kan extension.

Discussion of main results In this paper, we further study the semantics of HP, to demonstrate how close it is to operational semantics. For this, we provide two results. The most important, in the author's view, is full abstraction w.r.t. *fair testing semantics*. But the second result might be considered more convincing by many: it establishes that our semantics is fully abstract w.r.t. weak bisimilarity. The reason why it is here considered less important is that it relies on something external to the model itself, namely an LTS for strategies, constructed in an *ad hoc* way. Considering that a process calculus is defined by its reduction semantics, rather than by its possibly numerous LTSs, testing equivalences, which rely on the former, are more intrinsic than various forms of bisimilarity.

Now, why consider fair testing among the many testing equivalences? First of all, let us mention that we could probably generalise our result to any reasonable testing equivalence. Any testing equivalence relies on a ‘testing predicate’ $\bot$. E.g., for fair testing, it is the set of processes from which any unsuccessful, finite reduction sequence extends to a successful one. We conjecture that for any other predicate $\bot'$, if $\bot'$ is stable under weak bisimilarity, i.e, $P \simeq Q \in \bot'$ implies $P \in \bot'$, then we may interpret the resulting equivalence in terms of strategies, and get a fully abstract semantics. However, this paper is already quite complicated, and pushes generalisation rather far in other respects (see below). We thus chose to remain concrete about the considered equivalence. It was then natural to consider fair testing, as it is both one of the most prominent testing equivalences, and one of the finest. It was introduced independently by Natarajan and Cleaveland [30], and by Brinksma et al. [3, 33] (under the name of *should* testing in the latter paper), with the aim of reconciling the good properties of observation congruence [29] w.r.t. divergence, and the good properties of previous testing equivalences [7] w.r.t. choice. Typically, $a.b + a.c$ and $a.(b \oplus c)$ (where $+$ denotes guarded choice and $\oplus$ denotes internal choice) are not observation congruent, which is perceived as excessive discriminating power of observation congruence. Conversely, $(!\tau) \mid a$ and a are not must testing equivalent, which is perceived as excessive discriminating power of must testing equivalence. Fair testing rectifies both defects, and has been the subject of further investigation, as summarised, e.g., in Cacciagrano et al. [5].

Overview We now give a bit more detail on the contents, warning the reader that this paper is only an extended abstract, and that more technical details may be found in a (submitted) long version [18]. After recalling the game from HP in Section 2, as well as strategies and our semantic fair testing equivalence $\sim_f$ in Section 3, we prove that the translation $(\!|-|\!)$ of HP from CCS to strategies is such that $P \sim_{f,s} Q$ iff $(\!|P|\!) \sim_f (\!|Q|\!)$, where $\sim_{f,s}$ is standard fair testing equivalence (Theorem 4.6).

Our first attempts at proving this where obscured by easy, yet lengthy case analyses over moves. This prompted the search for a way of factoring out what holds ‘for all moves’. The result is the notion of *playground*, surveyed in Section 4.1. It is probably not yet in a mature state, and hopefully the axioms will simplify in the future. We show how the game recalled above organises into such a playground $\mathbb{D}^{CCS}$. We then develop the theory in Section 4.2, defining, for any playground $\mathbb{D}$, two LTSs, $\mathcal{T}_{\mathbb{D}}$ and $\mathcal{S}_{\mathbb{D}}$, of *process terms* and *strategies*, respectively, over an alphabet $\mathbb{F}_{\mathbb{D}}$. We then define a map $[\![-]\!] \colon \mathcal{T}_{\mathbb{D}} \to \mathcal{S}_{\mathbb{D}}$ between them, which we prove is a strong bisimulation.

Returning to the case of CCS in Section 4.3, we obtain that $\mathcal{S}_{\mathbb{D}^{CCS}}$ indeed has strategies as states, and that $\sim_f$ may be characterised in terms of this LTS. Furthermore, unfolding the definition of $\mathcal{T}_{\mathbb{D}^{CCS}}$, we find that its states are terms in a language containing CCS. So, we have maps $\mathrm{ob}(CCS) \overset{\theta}{\hookrightarrow} \mathrm{ob}(\mathcal{T}_{\mathbb{D}^{CCS}}) \xrightarrow{[\![-]\!]} \mathrm{ob}(\mathcal{S}_{\mathbb{D}^{CCS}})$, where ob takes the set of vertices, and with $[\![-]\!] \circ \theta = (\!|-|\!)$. Now, a problem is that CCS and the other two are LTSs on different alphabets, respectively $\mathbb{A}$ and $\mathbb{F}_{\mathbb{D}^{CCS}}$. We thus define morphisms $\mathbb{A} \xleftarrow{\xi} \mathcal{L} \xrightarrow{\chi} \mathbb{F}_{\mathbb{D}^{CCS}}$ and obtain by successive change of base (pullback when rewinding an arrow, postcomposition when following one) a strong bisimulation $[\![-]\!] \colon \mathcal{T}^{\mathbb{A}}_{\mathbb{D}_{CCS}} \to \mathcal{S}^{\mathbb{A}}_{\mathbb{D}_{CCS}}$ over $\mathbb{A}$. We then prove that θ, viewed as a map $\mathrm{ob}(CCS) \hookrightarrow \mathrm{ob}(\mathcal{T}^{\mathbb{A}}_{\mathbb{D}_{CCS}})$, is included in weak bisimilarity, which yields for all P, $P \simeq_{\mathbb{A}} (\!| P |\!)$ (Corollary 4.5). Finally, drawing inspiration from Rensink et al. [33], we prove that CCS and $\mathcal{S}^{\mathbb{A}}_{\mathbb{D}_{CCS}}$ both have enough $\mathbb{A}$-*trees*, in a suitable sense, and that this, together with Corollary 4.5, entails the main result.

Related work Trying to reconcile two mainstream approaches to denotational semantics, we have designed a (first version of a) general framework aiming at an effective theory of programming languages. Other such frameworks exist [31, 32, 36, 10, 4, 2, 17, 1], but most of them, with the notable exception of Kleene coalgebra, attempt to organise the traditional techniques of syntax with variable binding and reduction rules into some algebraic structure. Here, as in Kleene coalgebra, syntax and its associated LTS are derived notions. Our approach may thus be seen as an extension of Kleene coalgebra to an innocent/multi-player setting, yet ignoring quantitative aspects.

In another sense of the word 'framework', recent work of Winskel and colleagues [34] investigates a general notion of concurrent game, based on earlier work by Melliès [26]. In our approach, the idea is that each programming language is interpreted as a playground, and that morphisms of playgrounds denote translations between languages. Winskel et al., instead, construct a (large) bicategory, into which each programming language should embed. Beyond this crucial difference, both approaches use presheaves and factorisation systems, and contain a notion of innocent, concurrent strategy. The precise links between the original notion of innocence, theirs, and ours remain to be better investigated.

Melliès's work [27], although in a deterministic and linear setting, incorporates some 'concurrency' into plays by presenting them as string diagrams. Our innocentisation procedure further bears some similarity with Harmer et al.'s [14] presentation of innocence based on a distributive law. Hildebrandt's approach to fair testing equivalence [16] uses closely related techniques, e.g., presheaves and sheaves — indeed, our innocence condition may be viewed as a sheaf condition. However, (1) his model falls in the aforementioned category of presheaf models for which parallel composition is a complex operation; and (2) he uses sheaves to correctly incorporate infinite behaviour in the model, which is different from our notion of innocence. Finally, direct inspiration is drawn from Girard [12], one of whose aims is to bridge the gap between syntax and semantics.

Perspectives We plan to adapt our semantics to more complicated calculi like π, the Join and Ambients calculi, functional calculi, possibly with extra features (e.g., references, data abstraction, encryption), with a view to eventually generalising it. Preliminary investigations already led to a playground for π, whose adequacy remains to be established. More speculative directions include (1) defining a notion of morphisms for playgrounds, which should induce translations between strategies, and find sufficient conditions for such morphisms to preserve, resp. reflect testing equivalences; (2) generalising playgrounds to apply them beyond programming language semantics; in particular, preliminary work shows that playgrounds easily account for cellular automata; this raises the question of how morphisms of playgrounds would compare with existing notions of simulations between cellular automata [8]; (3) trying and recast the issue of deriving transition systems (LTSs) from reductions [35] in terms of playgrounds.

Notation Set is the category of sets; set is a skeleton of the category of finite sets, namely the category of finite ordinals and arbitrary maps between them; ford is the category of finite ordinals and monotone maps between them. For any category $\mathbb{C}$, $\widehat{\mathbb{C}} = [\mathbb{C}^{op}, \mathsf{Set}]$ denotes the category of presheaves on $\mathbb{C}$, while $\widehat{\mathbb{C}}^f = [\mathbb{C}^{op}, \mathsf{set}]$ and $\hat{\mathbb{C}} = [\mathbb{C}^{op}, \mathsf{ford}]$ respectively denote the categories of presheaves of finite sets and of finite ordinals. One should distinguish, e.g., 'presheaf of finite sets' $\mathbb{C}^{op} \to \mathsf{set}$ from 'finite presheaf of sets' $F \colon \mathbb{C}^{op} \to \mathsf{Set}$. The latter means that the disjoint union $\sum_{c \in \mathrm{ob}(\mathbb{C})} F(c)$ is finite. Throughout the paper, any finite ordinal n is seen as $\{1, \ldots, n\}$ (rather than $\{0, \ldots, n-1\}$).

The notion of LTS that we'll use here is a little more general than the usual one, but this does not change much. We thus refer to the long version for details. Let us just mention that we work in the category Gph of reflexive graphs, and that the category of LTSs over A is for us the slice category Gph/A. LTSs admit a standard change of base functor given by pullback, and its left adjoint given by postcomposition. Given any LTS $p \colon G \to A$, an edge in G is *silent* when it is mapped by p to an identity edge. This straightforwardly yields a notion of weak bisimilarity over A, which is denoted by $\simeq_A$.

Our (infinite) CCS terms are coinductively generated by the typed grammar

$$\frac{\Gamma \vdash P \qquad \Gamma \vdash Q}{\Gamma \vdash P|Q} \qquad \frac{\Gamma, a \vdash P}{\Gamma \vdash \nu a.P} \qquad \frac{\ldots \qquad \Gamma \vdash P_i \qquad \ldots}{\Gamma \vdash \sum_{i \in n} \alpha_i.P_i} \; (n \in \mathbb{N}),$$

where α_i is either a, $\overline{a}$, for $a \in \Gamma$, or $\heartsuit$. The latter is a 'tick' move used in the definition of fair testing equivalence. As a syntactic facility, we here understand Γ as ranging over $\mathbb{N}$, i.e., the free names of a process always are $1 \ldots n$ for some n. E.g., Γ, a denotes just $n+1$, and $a \in \Gamma$ means $a \in \{1, \ldots, \Gamma\}$.

Definition 1.1. *Let $\mathbb{A}$ be the reflexive graph with vertices given by finite ordinals, edges $\Gamma \to \Gamma'$ given by $\emptyset$ if $\Gamma \neq \Gamma'$, and by $\Gamma + \Gamma + \{id, \heartsuit\}$ otherwise, $id \colon \Gamma \to \Gamma$ being the identity edge on Γ. Elements of the first summand are denoted by $a \in \Gamma$, while elements of the second summand are denoted by $\overline{a}$.*

We view terms as a graph CCS over $\mathbb{A}$ with the usual transition rules. The graph $\mathbb{A}$ only has 'endo'-edges; some LTSs below do use more general graphs.

2 Recalling the game

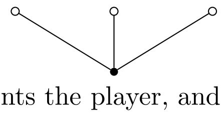
2.1 Positions, Moves, and Plays

In this section, we define plays in our game. For lack of space, we cannot be completely formal. A formal definition, with a gentle introduction to the required techniques, may be found in HP (Section 3). Here is a condensed account. We start by defining a category $\mathbb{C}$. Then, *positions* in our game are defined to be particular finite presheaves in $\widehat{\mathbb{C}}^f$. *Moves* in our game are defined as certain *cospans* $X \xrightarrow{s} M \xleftarrow{t} Y$ in $\widehat{\mathbb{C}}^f$, where t indicates that Y is the *initial* position of the move, while s indicates that X is the *final* position. *Plays* are then defined as finite composites of moves in the bicategory $\mathsf{Cospan}(\widehat{\mathbb{C}}^f)$ of cospans in $\widehat{\mathbb{C}}^f$. By construction, positions and plays form a subbicategory, called $\mathbb{D}_v^{ccs}$.

In order to motivate the definition of our base category $\mathbb{C}$, recall that (directed, multi) graphs may be seen as presheaves over the category freely generated by the graph with two objects $\star$ and $[1]$, and two edges $s, t \colon \star \to [1]$. Any 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 $G(t)(e)$. A way to visualise how such presheaves represent graphs is to compute their *categories of elements* [25]. Recall that the category of elements $\int G$ for a presheaf G over $\mathbb{C}$ has as objects pairs (c, x) with $c \in \mathbb{C}$ and $x \in F(c)$, and as morphisms $(c, x) \to (d, y)$ all morphisms $f \colon c \to d$ in $\mathbb{C}$ such that $F(f)(y) = x$. This category admits a canonical functor π_F to $\mathbb{C}$, and F is the colimit of the composite $\int F \xrightarrow{\pi_F} \mathbb{C} \xrightarrow{\mathsf{y}} \widehat{\mathbb{C}}$ with the Yoneda embedding. Hence, e.g., the category of elements for the representable presheaf over $[1]$ is the poset $(\star, s) \to ([1], id_{[1]}) \leftarrow (\star, t)$, which could be pictured as •——▶——•, thus recovering some graphical intuition.

We now define our base category $\mathbb{C}$. Let us first give the raw definition, and then explain. $\mathbb{C}$ is freely generated from the graph $\mathbb{G}$, defined as follows, plus some equations. As objects, $\mathbb{G}$ has (1) an object $\star$, (2) an object $[n]$ for all $n \in \mathbb{N}$, (3) objects $o_{n,i}$ (output), $\iota_{n,i}$ (input), ν_n (channel creation), π^l_n (left fork), π^r_n (right fork), π_n (fork), $\heartsuit_n$ (tick), $\tau_{n,i,m,j}$ (synchronisation), for all $i \in n, j \in m, n, m \in \mathbb{N}$. $\mathbb{G}$ has edges, for all n, (1) $s^n_1, \ldots, s^n_n \colon \star \to [n]$, (2) $s^c, t^c \colon [n] \to c$, for all $c \in \{\pi^l_n, \pi^r_n, \heartsuit_n\} \cup (\cup_{i \in n}\{o_{n,i}, \iota_{n,i}\})$, (3) $[n+1] \xrightarrow{s^{\nu_n}} \nu_n \xleftarrow{t^{\nu_n}} [n]$, (4) $\pi^l_n \xrightarrow{l^n} \pi_n \xleftarrow{r^n} \pi^r_n$, $o_{n,i} \xrightarrow{\epsilon^{n,i,m,j}} \tau_{n,i,m,j} \xleftarrow{\rho^{n,i,m,j}} \iota_{m,j}$, for all $i \in n, j \in m$. In the following, we omit superscripts when clear from context. As equations, we require, for all n, m, $i \in n$, and $j \in m$, (1) $s^c \circ s^n_i = t^c \circ s^n_i$, (2) $s^{\nu_n} \circ s^{n+1}_i = t^{\nu_n} \circ s^n_i$, (3) $l \circ t = r \circ t$, (4) $\epsilon \circ t \circ s_i = \rho \circ t \circ s_j$.

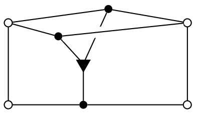
$$\begin{tikzcd}
(\star, s_1) \arrow[dr] & (\star, s_2) \arrow[d] & (\star, s_3) \arrow[dl] \\
 & ([3], id_{[3]}) &
\end{tikzcd}$$



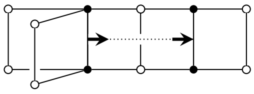
In order to explain this seemingly arbitrary definition, let us compute a few categories of elements for representable presheaves. Let us start with an easy one, that of $[3]$ (we implicitly identify any $c \in \mathbb{C}$ with $\mathsf{y}c$). An easy computation shows that it is the poset pictured above. We will think of it as a position with one player $([3], id_{[3]})$ connected to three channels, and draw it as above, where the bullet represents the player, and circles represent channels. (The graphical representation is slightly ambiguous, but nevermind.) In particular, elements over $[3]$ represent ternary players, while elements over $\star$ represent channels. *Positions* are finite presheaves empty except perhaps on $\star$ and $[n]$'s. Let $\mathbb{D}_h^{ccs}$ be the subcategory of $\widehat{\mathbb{C}}^f$ consisting of positions and monic arrows between them.

A more difficult category of elements is that of π_2. It is the poset generated by the graph on the left:

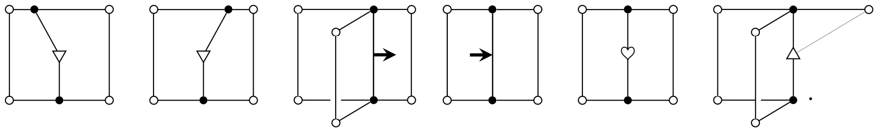
$$\begin{tikzcd}
lss_1 = rss_1 \arrow[r] \arrow[rrr, bend left] \arrow[dd, equal] & ls \arrow[d] & & rs \arrow[d] & lss_2 = rss_2 \arrow[l] \arrow[lll, bend right] \arrow[dd, equal] \\
& l \arrow[r] & id_{\pi_2} & r \arrow[l] & \\
lts_1 = rts_1 \arrow[rr] & & lt = rt \arrow[ul] \arrow[ur] & & lts_2 = rts_2 \arrow[ll]
\end{tikzcd}$$


.

We think of it as a binary player (lt) forking into two players (ls and rs), and draw it as on the right. The vertical edges on the outside are actually identities: the reason we draw separate vertices is to identify the top and bottom parts of the picture as the respective images of both legs of the following cospan. First, consider the inclusion $[2] \mid [2] \hookrightarrow \pi_2$: its domain is any pushout of $[s_1, s_2] \colon (\star + \star) \to [2]$ with itself, i.e., the position consisting of two binary players sharing their channels; and the inclusion maps it to the top part of the picture. Similarly, we have a map $[2] \hookrightarrow \pi_2$ given by the player lt and its channels (the bottom part). The cospan $[2] \mid [2] \to \pi_2 \leftarrow [2]$ is called the *local fork move* of arity 2.



For lack of space, we cannot spell out all such categories of elements and cospans. We give pictorial descriptions for $(m, j, n, i) = (3, 3, 2, 1)$ of $\tau_{m,j,n,i}$ on the right and of π^l_n, π^r_n, $o_{m,j}$, $\iota_{n,i}$, $\heartsuit_n$, and ν_n below:

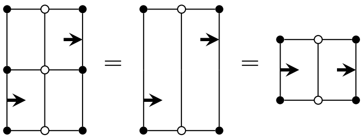


In each case, the representable is the middle object of a cospan determined by the top and bottom parts of the picture. E.g., for synchronisation we have $[m]\,{}_j|_i\,[n] \xrightarrow{s} \tau_{m,j,n,i} \xleftarrow{t} [m]\,{}_j|_i\,[n]$, where $[m]\,{}_j|_i\,[n]$ denotes the position X with one m-ary player x, one n-ary player y, such that $X(s_j)(x) = X(s_i)(y)$. Note that there is a crucial design choice in defining the legs of these cospans, which amounts to choosing initial and final positions for our moves.

$$\begin{tikzcd}
& I \arrow[dl] \arrow[d] \arrow[dr] & \\
X \arrow[r] & M & Y \arrow[l]
\end{tikzcd} \tag{1}$$

These cospans altogether form the set of *local moves*, and are the ‘seeds’ for (global) moves, in the following sense. Calling an *interface* any presheaf consisting only of channels, local moves may be equipped with a canonical interface, consisting of the channels of their initial position. If $X \xrightarrow{s} M \xleftarrow{t} Y$ is a local move (with final position X), and I is its canonical interface, we obtain a commuting diagram (1) in $\widehat{\mathbb{C}}^f$ (with all arrows monic). For any morphism $I \to Z$ to some position Z, pushing $I \to X$, $I \to M$, and $I \to Y$ along $I \to Z$ yields, by universal property of pushout, a new cospan, say $X' \to M' \leftarrow Y'$. Letting *(global) moves* be all cospans obtained in this way, and plays be all composites of moves in $\mathsf{Cospan}(\widehat{\mathbb{C}}^f)$, we obtain, as promised a subbicategory $\mathbb{D}_v^{CCS}$.

Passing from local to global moves allows moves to occur in larger positions. Furthermore, we observe that plays feature some concurrency. For instance, composing two global moves as on the right, we obtain a play in which the order of appearance of moves is no longer visible. In passing, this play embeds into a synchronisation, but is not one, since the input and output moves are not related. This play may be understood as each player communicating with the outside world. We conclude with a useful classification of moves.



(2)

Definition 2.1. *A move is* full *iff it is neither a left nor a right fork. We call* $\mathbb{F}$ *the graph of global, full moves.*

Intuitively, a move is full when its final position contains all possible avatars of involved players.

3 Behaviours, strategies, and fair testing

3.1 Behaviours

Recall from HP the category $\mathbb{E}$ whose objects are maps $U \leftarrow X$ in $\widehat{\mathbb{C}}$, such that there exists a play $Y \to U \leftarrow X$, i.e., objects are plays, where we forget the final position. Its morphisms $(U \leftarrow X) \to (U' \leftarrow X')$ are commuting diagrams as on the right with all arrows monic. Morphisms $U \to U'$ in $\mathbb{E}$ represent extensions of U, both spatially (i.e., embedding into a larger position) and dynamically (i.e., adding more moves).

$$\begin{tikzcd}
U \arrow[r] & U' \\
X \arrow[u] \arrow[r] & X' \arrow[u]
\end{tikzcd}$$

We may relativise this category $\mathbb{E}$ to a particular position X, yielding a category $\mathbb{E}(X)$ of plays on X: take the fibre over X of the functor $\mathrm{cod}\colon \mathbb{E} \to \mathbb{D}_h^{ccs}$ mapping any play $U \leftarrow X$ to its initial position X. The objects of $\mathbb{E}(X)$ are just plays $(U \leftarrow X)$ on X, and morphisms are morphisms of plays whose lower border is id_X. This leads to a category of 'naive' strategies, called behaviours.

Definition 3.1. *The category* B_X *of* behaviours *on* X *is the category* $\widehat{\mathbb{E}(X)}^f$ *of presheaves of finite sets on* $\mathbb{E}(X)$*.*

Behaviours suffer from the deficiency of allowing unwanted cooperation between players. HP (Example 12) exhibits a behaviour where players choose with whom they synchronise, which clearly is not allowed in CCS.

3.2 Strategies

To rectify this, we consider the full subcategory $\mathbb{V}$ of $\mathbb{E}$ consisting of *views*, i.e., compositions of basic local moves. We relativise views to a position X, as follows. Let, for any $n \in \mathbb{N}$, $[n]$ denote the single n-ary player, i.e., a single player connected to n distinct channels. Players of X are in 1-1 correspondence with pairs (n, x), with $x \colon [n] \to X$ in $\mathbb{D}_h^{ccs}$. Relativisation of $\mathbb{V}$ to X is given by the category $\mathbb{V}_X$ with as objects all pairs (V, x), where $x \colon [n] \to X$, and V is a view with initial position $[n]$. Morphisms are induced by those of $\mathbb{E}$.

Definition 3.2. *The category S_X of* strategies *on X is the category $\widehat{\mathbb{V}_X}$ of presheaves of finite ordinals on $\mathbb{V}_X$.*

$$\begin{tikzcd}
\mathbb{V}_X^{op} \arrow[r, hook] \arrow[d, "S"'] & \mathbb{E}_X^{op} \arrow[d, "S'"'] & \mathbb{E}(X)^{op} \arrow[l, hook', "j"] \arrow[dl, "\overline{S}"] \\
\mathsf{ford} \arrow[r, hook, "i"] & \mathsf{set},
\end{tikzcd}$$

This rules out undesired behaviours. Recall from HP how to map strategies to behaviours: let first $\mathbb{E}_X$ be the category obtained as $\mathbb{V}_X$ from all plays instead of just views. Then, starting from a strategy S, let S' be obtained by right Kan extension of $i \circ S$ (by $\mathbb{V}_X^{op} \hookrightarrow \mathbb{E}_X^{op}$ being full and faithful), and let $\overline{S} = S' \circ j$. The assignment $S \mapsto \overline{S}$ extends to a full and faithful functor $\overline{(-)} \colon \mathsf{S}_X \to \mathsf{B}_X$. Furthermore, $\overline{(-)}$ admits a left adjoint, which we call *innocentisation*, maping naive strategies (behaviours) to innocent ones. By standard results [24], we have for any S: $\overline{S}(U) = \int_{v \in \mathbb{V}_X} S(v)^{\mathbb{E}_X(v,U)}$. Equivalently, $\overline{S}(U)$ is a limit of $(\mathbb{V}_X/U)^{op} \xrightarrow{\text{dom}} \mathbb{V}_X^{op} \xrightarrow{S} \mathsf{ford} \hookrightarrow \mathsf{set}$.

3.3 Decomposition: a syntax for strategies

Our definition of strategies is rather semantic in flavour. Indeed, presheaves are akin to domain theory. However, they also lend themselves well to a syntactic description. First, it is shown in HP that strategies on an arbitrary position X are in 1-1 correspondence with families of strategies indexed by the players of X. Recall that $[n]$ is the position consisting of one n-ary player, and that players of X may be defined as elements of $\mathrm{Pl}(X) = \sum_{n \in \mathbb{N}} \mathbb{D}_h^{ccs}([n], X)$.

Proposition 3.3. *We have $\mathsf{S}_X \cong \prod_{(n,x) \in \mathrm{Pl}(X)} \mathsf{S}_{[n]}$. For any $S \in \mathsf{S}_X$, we denote by $S_{(n,x)}$ the component corresponding to $(n, x) \in \mathrm{Pl}(X)$ under this isomorphism.*

This result yields a construction letting two strategies interact along an *interface*, i.e., a position consisting only of channels. This will be the basis of our semantic definition of fair testing equivalence. Consider any pushout Z of $X \leftarrow I \rightarrow Y$ where I is an interface. We have

Corollary 3.4. $\mathsf{S}_Z \cong \mathsf{S}_X \times \mathsf{S}_Y$*.*

Proof. We have $\mathbb{V}_Z \cong \mathbb{V}_X + \mathbb{V}_Y$, and conclude by universal property of coproduct.

We denote by $[S, T]$ the image of $(S, T) \in \mathsf{S}_X \times \mathsf{S}_Y$ under this isomorphism.

So, strategies over arbitrary positions may be decomposed into strategies over ‘typical’ players $[n]$. Let us now explain that strategies over such players may be further decomposed. For any strategy S on $[n]$ and basic move $b \colon [n'] \to [n]$, let the *residual* $S \cdot b$ of S after b be the strategy playing like S after b, i.e., for all $v \in \mathbb{V}_{[n']}$, $(S \cdot b)(v) = S(b \bullet v)$, where $\bullet$ denotes composition in $\mathbb{D}_v^{ccs}$. S is almost determined by its residuals. The only information missing from the $S \cdot b$'s to reconstruct S is the set of initial states and how they relate to the initial states of each $(S \cdot b)$. Thus, for any position X, let id^v_X denote the identity play on X (i.e., nothing happens). For any initial state $\sigma \in S(id_{[n]})$, let $S_{|\sigma}$ be the restriction of S to states derived from σ, i.e., for all v, those $\sigma' \in S(v)$ which are mapped to σ under the restriction $S(!)\colon S(v) \to S(id_{[n]})$. S is determined by its set $S(id_{[n]})$ of initial states, plus the function $(\sigma, b) \mapsto (S_{|\sigma} \cdot b)$. Since $S(id_{[n]})$ is a finite ordinal m, we have for all n:

Theorem 3.5. $\mathsf{S}_{[n]} \cong \sum_{m \in \mathbb{N}} (\prod_{b\colon [n'] \to [n]} \mathsf{S}_{[n']})^m \cong (\prod_{b\colon [n'] \to [n]} \mathsf{S}_{[n']})^\star$.

This result may be understood as saying that strategies form a fixpoint of a certain (polynomial [23]) endofunctor of $\mathsf{Set}/\mathbb{I}$, where $\mathbb{I}$ is the set of 'typical' players $[n]$. This may be strengthened to show that they form a terminal coalgebra, i.e, that they are in bijection with infinite terms in the following typed grammar, with judgements $n \vdash_{\mathsf{D}} D$ and $n \vdash S$, where D is called a *definite prestrategy* and S is a *strategy*:

$$\frac{\ldots\ n_b \vdash S_b\ \ldots\ (\forall b\colon [n_b] \to [n] \in [\mathbb{B}]_n)}{n \vdash_{\mathsf{D}} \langle (S_b)_{b \in [\mathbb{B}]_n} \rangle} \qquad \frac{\ldots\ n \vdash_{\mathsf{D}} D_i\ \ldots\ (\forall i \in m)}{n \vdash \oplus_{i \in m} D_i}\ (m \in \mathbb{N}),$$

where $[\mathbb{B}]_n$ denotes the set of all isomorphism classes of basic moves from $[n]$. We need to use isomorphism classes here, because strategies may not distinguish between different, yet isomorphic basic moves. This achieves the promised syntactic description of strategies. We may readily define the translation of CCS processes, coinductively, as follows. For processes with channels in Γ, we define

$$\begin{array}{rclcrcl}
\llparenthesis \sum_{i \in n} \alpha_i.P_i \rrparenthesis &=& \langle b \mapsto \oplus_{\{i \in n \mid b = \llparenthesis \alpha_i \rrparenthesis\}} \llparenthesis P_i \rrparenthesis \rangle & \qquad & \llparenthesis a \rrparenthesis &=& \iota_{\Gamma,a} \\
\llparenthesis \nu a.P \rrparenthesis &=& \langle \nu_\Gamma \mapsto \llparenthesis P \rrparenthesis, \_ \mapsto \varnothing \rangle & & \llparenthesis \overline{a} \rrparenthesis &=& o_{\Gamma,a} \\
\llparenthesis P \mid Q \rrparenthesis &=& \langle \pi^l_\Gamma \mapsto \llparenthesis P \rrparenthesis, \pi^r_\Gamma \mapsto \llparenthesis Q \rrparenthesis, \_ \mapsto \varnothing \rangle & & \llparenthesis \heartsuit \rrparenthesis &=& \heartsuit_\Gamma .
\end{array}$$

E.g., $a.P + a.Q + \bar{b}.R$ is mapped to $\langle \iota_{\Gamma,a} \mapsto (\llparenthesis P \rrparenthesis \oplus \llparenthesis Q \rrparenthesis), o_{\Gamma,b} \mapsto \llparenthesis R \rrparenthesis, \_ \mapsto \varnothing \rangle$.

3.4 Semantic fair testing

We may now recall our semantic analogue of fair testing equivalence.

Definition 3.6. Closed-world *moves are (the global variants of)* ν,$\heartsuit$,π_n, *and* $\tau_{n,i,m,j}$. *A play is* closed-world *when it is a composite of closed-world moves.*

Let a closed-world play be *successful* when it contains a $\heartsuit$ move. Let then $⫫_Z$ denote the set of behaviours B such that for any unsuccessful, closed-world play $U \leftarrow Z$ and $\sigma \in B(U)$, there exists $f\colon U \to U'$, with U' closed-world and successful, and $\sigma' \in B(U')$ such that $B(f)(\sigma') = \sigma$. Finally, let us say that a triple (I, h, S), for any $h\colon I \to X$ and strategy $S \in \mathsf{S}_X$, *passes* the test consisting of a morphism $k\colon I \to Y$ of positions and a strategy $T \in \mathsf{S}_Y$ iff $\overline{[S, T]} \in ⫫_Z$, where Z is the pushout of h and k. Let $S^{⫫}$ denote the set of all such (k, T).

Definition 3.7. *For any* $h\colon I \to X$, $h'\colon I \to X'$, $S \in \mathsf{S}_X$, *and* $S' \in \mathsf{S}_{X'}$, $(I, h, S) \sim_f (I, h', S')$ *iff* $(I, h, S)^{⫫} = (I, h', S')^{⫫}$.

This yields an equivalence relation, analogous to standard fair testing equivalence, which we hence also call fair testing equivalence.

We have defined a translation $(\!|-|\!)$ of CCS processes to strategies, which raises the question of whether it preserves or reflects fair testing equivalence. The rest of the paper is devoted to proving that it does both.

4 Playgrounds and main result

4.1 Playgrounds: a theory of individuality and atomicity

$$\begin{tikzcd}
X \arrow[r, "h"] \arrow[d, "u"'] & X' \arrow[r, "k"] \arrow[d, "u'"'] & X'' \arrow[d, "u''"] \\
Y \arrow[r, "h'"] \arrow[d, "v"'] \arrow[ur, Rightarrow, shorten=3mm, "\alpha"'] & Y' \arrow[r, "k'"] \arrow[d, "v'"'] \arrow[ur, Rightarrow, shorten=3mm, "\alpha'"'] & Y'' \arrow[d, "v''"] \\
Z \arrow[r, "h''"'] \arrow[ur, Rightarrow, shorten=3mm, "\beta"'] & Z' \arrow[r, "k''"'] \arrow[ur, Rightarrow, shorten=3mm, "\beta'"'] & Z'',
\end{tikzcd}$$

We start by trying to give an idea of the notion of playground. To start with, we organise the game into a *(pseudo) double category* [13, 11]. This is a weakening of Ehresmann's double categories [9], where one direction has non strictly associative composition. Although we consider proper pseudo double categories, we often may treat them safely as double categories. A pseudo double category $\mathbb{D}$ consists of a set $\mathrm{ob}(\mathbb{D})$ of *objects*, shared by two categories $\mathbb{D}_h$ and $\mathbb{D}_v$. $\mathbb{D}_h$ is called the *horizontal* category of $\mathbb{D}$, and $\mathbb{D}_v$ is the *vertical* category. Composition in $\mathbb{D}_h$ is denoted by $\circ$, while we use $\bullet$ for $\mathbb{D}_v$. $\mathbb{D}$ is furthermore equipped with a set of *double cells* α, which have vertical, resp. horizontal, domain and codomain, denoted by $\mathrm{dom}_v(\alpha)$, $\mathrm{cod}_v(\alpha)$, $\mathrm{dom}_h(\alpha)$, and $\mathrm{cod}_h(\alpha)$. We picture this as, e.g., α above, where $u = \mathrm{dom}_h(\alpha)$, $u' = \mathrm{cod}_h(\alpha)$, $h = \mathrm{dom}_v(\alpha)$, and $h' = \mathrm{cod}_v(\alpha)$. $\mathbb{D}$ 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 [11], and we here only mention the *interchange law*, which says that the two ways of parsing the above diagram coincide: $(\beta' \circ \beta) \bullet (\alpha' \circ \alpha) = (\beta' \bullet \alpha') \circ (\beta \bullet \alpha)$.

$$\begin{tikzcd}
X \arrow[r, "h"'] \arrow[d, "s"'] & X' \arrow[d, "s'"] \\
U \arrow[r, "k"] & V \\
Y \arrow[r, "l"] \arrow[u, "t"] & Y' \arrow[u, "t'"']
\end{tikzcd}$$

Example 4.1. Returning to the game, we have seen that positions are the objects of the category $\mathbb{D}_h^{CCS}$, whose morphisms are embeddings of positions. But positions are also the objects of the bicategory $\mathbb{D}_v^{CCS}$, whose morphisms are plays. It should seem natural to define a pseudo double category structure with double cells given by commuting diagrams as on the right in $\widehat{\mathbb{C}}$. Here, Y is the initial position and X is the final one; all arrows are mono. This indeed forms a pseudo double category $\mathbb{D}^{CCS}$. Furthermore, for any double category $\mathbb{D}$, let $\mathbb{D}_H$ be the category with objects all morphisms of $\mathbb{D}_v$, and with morphisms $u \to u'$ all double cells α such that $\mathrm{dom}_h(\alpha) = u$ and $\mathrm{cod}_h(\alpha) = u'$. A crucial feature of $\mathbb{D}^{CCS}$ is that the canonical functor $\mathrm{cod}_v \colon \mathbb{D}_H \to \mathbb{D}_h$ mapping any such α to $\mathrm{cod}_v(\alpha)$ is a Grothendieck fibration [21]. This means that one may canonically 'restrict' a play, say $u' \colon X' \to Y'$, along a horizontal morphism $h' \colon Y \to Y'$, and obtain a universal cell as α above, in a suitable sense.

$$\begin{tikzcd}
d \arrow[r, "y"] \arrow[d, dashed, "v^{y,M}"'] \arrow[dr, phantom, "\alpha^{y,M}"] & Y \arrow[d, "M"] \\
d^{y,M} \arrow[r, dashed, "y^M"] & X
\end{tikzcd}$$

Playgrounds are pseudo double categories with extra data and axioms, the first of which is that cod_v should be a fibration. To give a brief idea of further axioms, a playground $\mathbb{D}$ is equipped with a set of objects $\mathbb{I}$, called *individuals*, which correspond to our 'typical' players above. Let $\mathrm{Pl}(X) = \sum_{d \in \mathbb{I}} \mathbb{D}_h(d, X)$ denote the set of players of X. It also comes with classes $\mathbb{F}$ and $\mathbb{B}$ of *full*, resp. *basic* moves; and every play (i.e., vertical morphism) is assumed to admit a decomposition into moves in $\mathbb{F} \cup \mathbb{B}$ (hence *atomicity*). Basic moves are assumed to have individuals as both domain and codomain, and *views* are defined to be composites of basic moves. The crucial axiom for innocence to behave well assumes that, for any position Y and player $y \colon d \to Y$, there exists a cell $\alpha^{y,M}$ as above, with $v^{y,M}$ a view, which is unique up to canonical isomorphism of such. Intuitively: any player in the final position of a play has an essentially unique view of the play. A last, sample axiom shows how some sequentiality is enforced, which is useful to tame the concurrency observed in (2). It says that any double cell as in the center below, where b is a basic move and M is any move, decomposes in exactly one of the forms on the left and right:

$$\begin{tikzcd}
A \arrow[r] \arrow[d] \arrow[dr, phantom, "\alpha_1"] & X \arrow[d] & & A \arrow[r, "h"] \arrow[d, "w"'] \arrow[dr, phantom, "\alpha"] & X \arrow[d, "u"] & & A \arrow[r] \arrow[d] \arrow[dr, phantom, "\alpha_1"] & X \arrow[d] \\
B \arrow[r] \arrow[d] \arrow[dr, phantom, "\alpha_2"] & Y \arrow[d] & \leftsquigarrow & B \arrow[d, "b"'] & Y \arrow[d, "M"] & \rightsquigarrow & B \arrow[d] & Y \arrow[d] \arrow[dl, phantom, "\alpha_2"] \\
C \arrow[r] & Z & & C \arrow[r, "k"'] & Z & & C \arrow[r] \arrow[ur] & Z.
\end{tikzcd}$$

The idea is that, C being an individual, if M has a non-trivial restriction to C, then b must be one of its views. Again, for the formal definition, see [18].

Proposition 4.2. $\mathbb{D}^{CCS}$ *forms a playground (basic moves being the* local *ones).*

4.2 Syntaxes and labelled transition systems

Notions of residuals and restrictions defined above for CCS are easily generalised to arbitrary playgrounds. They lead to the exact same syntax as in the concrete case (below Theorem 3.5). They further yield a first, naive LTS over full moves for strategies. The intuition is that there is a transition $S \xrightarrow{M} S'$, for any full move M, when $S \cdot M = S'$. (Residuals $S \cdot M$ are here defined analogously to the case of basic moves $S \cdot b$ above.) An issue with this LTS is that $S \cdot M$ may have several possible initial states, and we have seen that it makes more sense to restrict to a single state before taking residuals. We thus define our LTS $\mathbb{S}_{\mathbb{D}}$ to have as vertices pairs (X, S) of a position X and a *definite* strategy S, i.e., a strategy with exactly one initial state (formally, $S_{(d,x)}(id_d) = 1$ for all $(d, x) \in \mathrm{Pl}(X)$ — recalling that id_d is an (initial) object in $\mathbb{V}_d$). We then say that there is a transition $(X, S) \xrightarrow{M} (X', S')$ for any full move $M \colon X' \to X$, when $S' = (S \cdot M)_{|\sigma'}$, for some initial state σ' of $S \cdot M$.

Example 4.3. Consider a strategy of the shape $S = \langle \pi_n^r \mapsto S_1, \pi_n^l \mapsto S_2, \_ \mapsto \varnothing \rangle$ on $[n]$, with definite S_1 and S_2. There is a π_n transition to the position with two n-ary players x_1 and x_2, equipped with S_1 and S_2, respectively. If now S_1 and S_2 are not definite, any π_n transition has to pick initial states $\sigma_1 \in S_1(id_{[n]})$ and $\sigma_2 \in S_2(id_{[n]})$, i.e., $S \xrightarrow{\pi_n} [(S_1)_{|\sigma_1}] \mid [(S_2)_{|\sigma_2}]$. Here, we use a shorthand notation for pairs (X, S), defined as follows. First, for any strategy S over $[n]$ and position X with exactly one n-ary player x and names in Γ, we denote by $\Gamma \vdash [x : S](a_1, \ldots, a_n)$ the pair (X, S), where $a_i = X(s_i)(x)$, for all $i \in n$. If now X has several players, say $x_1, \ldots, x_p$, of respective arities $n_1, \ldots, n_p$, and $S_1, \ldots, S_p$ are strategies of such arities, we denote by $\Gamma \vdash [x_1 : S_1](a_1^1, \ldots, a_{n_1}^1) \mid \ldots \mid [x_p : S_p](a_1^p, \ldots, a_{n_p}^p)$ the pair $(X, [S_1, \ldots, S_p])$. When they are irrelevant, we often omit Γ, the x_j's, and the a_i^j's, as in our example.

$$\frac{\ldots\ d_x \vdash T_x\ \ldots}{d \vdash M\langle (T_x)_{x \in \mathrm{Pl}(M)} \rangle} \qquad \frac{\ldots\ d_i \vdash T_i\ \ldots\ (\forall i \in n)}{d \vdash \sum_{i \in n} M_i.T_i}$$

Beyond the one for strategies, there is another syntax one can derive from any playground. Instead of relying on basic moves as before, one now relies on full moves. Thinking of full moves as inference rules (e.g., in natural deduction), the premises of the rule for any full $M \colon X \to Y$ should be those players (d_x, x) of X whose view through M is non-trivial, i.e., is a basic move. We call this set of players $\mathrm{Pl}(M)$. The natural syntax rule is thus the first one above (glossing over some details), which defines *process terms* T. We add a further rule for guarded sum allowing to choose between several moves. One has to be a little careful here, and only allow moves $M \colon X \to Y$ such that $\mathrm{Pl}(M)$ is a singleton. This yields the second rule above, where $n \in \mathbb{N}$, and $\forall i \in n$, M_i is such a move and d_i is the arity of the unique element of $\mathrm{Pl}(M_i)$. Calling T_d the set of infinite terms for this syntax, there is a natural translation map $[\![-]\!] \colon \mathsf{T}_d \to \mathsf{S}_d$ to strategies, for all $d \in \mathbb{I}$, which looks a lot like $(\!|-|\!)$, and an LTS $\mathcal{T}_{\mathbb{D}}$, whose vertices are pairs (X, T) of a position X, with $T \in \prod_{d, x \in \mathrm{Pl}(X)} \mathsf{T}_d$. The main result on playgrounds is

Theorem 4.4. *The map $[\![-]\!] \colon \mathcal{T}_{\mathbb{D}} \to \mathcal{S}_{\mathbb{D}}$ is a functional, strong bisimulation.*

4.3 Change of base and main result

The LTS $\mathcal{S}_{\mathbb{D}^{ccs}}$ obtained for $\mathbb{D}^{ccs}$ is much too fine to be relevant for bisimilarity to make behavioural sense. E.g., the translations of $a|b$ and $b|a$ are not bisimilar. Indeed, labels, i.e., full moves in $\mathbb{F}_{\mathbb{D}^{ccs}}$, bear the information of which player is involved in the transition. So both strategies have a π_Γ translation to a position with two Γ-ary players, say x_1 and x_2. But then, $a \mid b$ has a transition where x_1 plays an input on a, which $b \mid a$ cannot match. Refining the above notation, and omitting $(\!|-|\!)$, we may write the former transitions as $[a \mid b] \xrightarrow{\pi_\Gamma} [a] \mid [b] \xrightarrow{x_1, \iota_{\Gamma,a}} [0] \mid [b]$. There is another problem with this LTS, namely that there are undue transitions. E.g., we have $[\nu a.a] \xrightarrow{\nu_0} [a] \xrightarrow{\iota_{(a),a}} 0$. The transition system does not yet take privacy of channels into account.

Let us first rectify the latter deficiency. To this end, we pull back our LTS $\mathcal{S}_{\mathbb{D}^{ccs}} \to \mathbb{F}_{\mathbb{D}^{ccs}}$ along a morphism of graphs $\mathcal{L} \to \mathbb{F}_{\mathbb{D}^{ccs}}$ defined as follows. Let $\mathcal{L}$

have *interfaced positions* as vertices, i.e., morphisms $h\colon I \to X$ from an interface to a position. I specifies the public channels, and hence we let edges $h \to h'$ be commuting diagrams of the shape (1), where M may be any full move (X being the final position), except inputs and outputs on a channel outside the image of I. We then straightforwardly define $\chi\colon \mathcal{L} \to \mathbb{F}_{\mathbb{D}^{CCS}}$ to map h to X and any diagram above to M. The pullback $\mathcal{S}^{\mathcal{L}}_{\mathbb{D}^{CCS}} \to \mathcal{L}$ of $\mathcal{S}_{\mathbb{D}^{CCS}}$ along χ is rid of undue communications on private channels.

To rectify the other deficiency mentioned above, recalling from Definition 1.1 that $\mathbb{A}$ is the alphabet for CCS, we define a morphism $\xi\colon \mathcal{L} \to \mathbb{A}$ by mapping $(I \to X)$ to its set $I(\star)$ of channels, and any M to (1) $\heartsuit$ if M is a tick move, (2) *id* if M is a synchronisation, a fork, or a channel creation, (3) a if M is an input on $a \in I(\star)$, (4) $\overline{a}$ if M is an output on $a \in I(\star)$. (Positions are formally defined as presheaves to set, hence channels directly form a finite ordinal number.) It is here crucial to have restricted attention to $\mathcal{L}$ beforehand, otherwise we would not know what to do with communications on private channels. Let $\mathcal{S}^{\mathbb{A}}_{\mathbb{D}^{CCS}} = \xi_!(\mathcal{S}^{\mathcal{L}}_{\mathbb{D}^{CCS}})$ be the post-composition of $\mathcal{S}^{\mathcal{L}}_{\mathbb{D}^{CCS}} \to \mathcal{L}$ with ξ.

The obtained LTS $\mathcal{S}^{\mathbb{A}}_{\mathbb{D}^{CCS}} \to \mathbb{A}$ is now ready for our purposes. Proceeding similarly for the LTS $\mathcal{T}_{\mathbb{D}^{CCS}}$ of process terms, we obtain a strong, functional bisimulation $[\![-]\!]\colon \mathrm{ob}(\mathcal{T}^{\mathbb{A}}_{\mathbb{D}^{CCS}}) \to \mathrm{ob}(\mathcal{S}^{\mathbb{A}}_{\mathbb{D}^{CCS}})$ over $\mathbb{A}$. We then prove that $\theta\colon \mathrm{ob}(CCS) \hookrightarrow \mathrm{ob}(\mathcal{T}^{\mathbb{A}}_{\mathbb{D}^{CCS}})$ is included in weak bisimilarity over $\mathbb{A}$, and, easily, that $(\!|-|\!) = [\![-]\!] \circ \theta$.

Corollary 4.5. *For all P, $P \simeq_{\mathbb{A}} (\!|P|\!)$.*

Furthermore, we prove that $\sim_f$ coincides with the standard, LTS-based definition of fair testing, i.e., $P \sim_{f,s} Q$ iff for all sensible T, $(P \mid T \in \bot_s) \Leftrightarrow (Q \mid T \in \bot_s)$, where $P \in \bot_s$ iff any $\heartsuit$-free reduction sequence $P \Rightarrow P'$ extends to one with $\heartsuit$. To obtain our main result, we finally generalise an observation of Rensink and Vogler [33], which essentially says that for fair testing equivalence in CCS, it is sufficient to consider a certain class of tree-like tests, called *failures*. We first slightly generalise the abstract setting of De Nicola and Hennessy [7] for testing equivalences, e.g., to accomodate the fact that strategies are indexed over interfaces. This yields a notion of *effective graph*. We then show that, for any effective graph G over an alphabet A, the result on failures goes through, provided *G has enough A-trees*, in the sense that, up to mild conditions, for any tree t over A, there exists $x \in G$ such that $x \simeq_A t$. Consequently, for any relation $R\colon G \mathrel{-\!\!\!\mapsto} G'$ between two such effective graphs with enough A-trees, if R is included in weak bisimilarity over A, then R preserves and reflects fair testing equivalence. We thus obtain our main result:

Theorem 4.6. *For any $\Gamma \in \mathbb{N}$, let I_Γ be the interface consisting of Γ channels, and $h_\Gamma\colon I_\Gamma \to [\Gamma]$ be the canonical inclusion. For any CCS processes P and Q over Γ, we have $P \sim_{f,s} Q$ iff $(I_\Gamma, h_\Gamma, (\!|P|\!)) \sim_f (I_\Gamma, h_\Gamma, (\!|Q|\!))$.*

Remark 4.7. Until now, we have considered arbitrary, infinite CCS processes. Let us now restrict ourselves to recursive processes (e.g., in the sense of HP). We obviously still have that $(\!|P|\!) \sim_f (\!|Q|\!)$ implies $P \sim_{f,s} Q$. The converse is less obvious and may be stated in very simple terms: suppose you have two recursive CCS processes P and Q and a test process T, possibly non-recursive, distinguishing P from Q; is there any recursive T' also distinguishing P from Q?

References

[1] B. Ahrens. Initiality for typed syntax and semantics. In C.-H. L. Ong, R. J. G. B. de Queiroz, eds., *WoLLIC*, vol. 7456 of *Lecture Notes in Computer Science*. Springer, 2012.

[2] M. M. Bonsangue, J. J. M. M. Rutten, A. Silva. A Kleene theorem for polynomial coalgebras. In L. de Alfaro, ed., *FOSSACS*, vol. 5504 of *Lecture Notes in Computer Science*. Springer, 2009.

[3] E. Brinksma, A. Rensink, W. Vogler. Fair testing. In I. Lee, S. A. Smolka, eds., *CONCUR*, vol. 962 of *Lecture Notes in Computer Science*. Springer, 1995.

[4] R. Bruni, U. Montanari. Cartesian closed double categories, their lambda-notation, and the pi-calculus. In *LICS '99*. IEEE Computer Society, 1999.

[5] D. Cacciagrano, F. Corradini, C. Palamidessi. Explicit fairness in testing semantics. *Logical Methods in Computer Science*, 5(2), 2009.

[6] G. L. Cattani, G. Winskel. Presheaf models for concurrency. In D. van Dalen, M. Bezem, eds., *CSL*, vol. 1258 of *Lecture Notes in Computer Science*. Springer, 1996.

[7] R. De Nicola, M. Hennessy. Testing equivalences for processes. *Theor. Comput. Sci.*, 34, 1984.

[8] M. Delorme, J. Mazoyer, N. Ollinger, G. Theyssier. Bulking I: An abstract theory of bulking. *Theoretical Computer Science*, 412(30), 2011.

[9] C. Ehresmann. *Catégories et structures*. Dunod, 1965.

[10] F. Gadducci, U. Montanari. The tile model. In G. D. Plotkin, C. Stirling, M. Tofte, eds., *Proof, Language, and Interaction*. The MIT Press, 2000.

[11] R. Garner. *Polycategories*. PhD thesis, University of Cambridge, 2006.

[12] J.-Y. Girard. Locus solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3), 2001.

[13] M. Grandis, R. Pare. Limits in double categories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 40(3), 1999.

[14] R. Harmer, M. Hyland, P.-A. Melliès. Categorical combinatorics for innocent strategies. In *LICS*. IEEE Computer Society, 2007.

[15] R. Harmer, G. McCusker. A fully abstract game semantics for finite nondeterminism. In *LICS '99*, 1999.

[16] T. T. Hildebrandt. Towards categorical models for fairness: fully abstract presheaf semantics of SCCS with finite delay. *Theoretical Computer Science*, 294(1/2), 2003.

[17] T. Hirschowitz. Cartesian closed 2-categories and permutation equivalence in higher-order rewriting. Preprint, 2010.

[18] T. Hirschowitz. Full abstraction for fair testing in CCS. Draft available from http://lama.univ-savoie.fr/~hirschowitz, 2012.

[19] T. Hirschowitz, D. Pous. Innocent strategies as presheaves and interactive equivalences for CCS. *Scientific Annals of Computer Science*, 22(1), 2012. Selected papers from ICE '11.

[20] J. M. E. Hyland, C.-H. L. Ong. On full abstraction for PCF: I, II, and III. *Inf. Comput.*, 163(2), 2000.

[21] B. Jacobs. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.

[22] A. Joyal, M. Nielsen, G. Winskel. Bisimulation and open maps. In *LICS '93*. IEEE Computer Society, 1993.

[23] J. Kock. Polynomial functors and trees. *International Mathematics Research Notices*, 2011(3), 2011.

[24] S. Mac Lane. *Categories for the Working Mathematician*. Number 5 in Graduate Texts in Mathematics. Springer, 2nd edition, 1998.

[25] S. MacLane, I. Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Universitext. Springer, 1992.

[26] P.-A. Melliès. Asynchronous games 2: the true concurrency of innocence. In *Proc. CONCUR '04*, vol. 3170 of *LNCS*. Springer Verlag, 2004.

[27] P.-A. Melliès. Game semantics in string diagrams. In *LICS*. IEEE, 2012.

[28] R. Milner. *A Calculus of Communicating Systems*, vol. 92 of *LNCS*. Springer, 1980.

[29] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[30] V. Natarajan, R. Cleaveland. Divergence and fair testing. In Z. Fülöp, F. Gécseg, eds., *ICALP*, vol. 944 of *Lecture Notes in Computer Science*. Springer, 1995.

[31] T. Nipkow. Higher-order critical pairs. In *LICS '91*. IEEE Computer Society, 1991.

[32] G. D. Plotkin. A structural approach to operational semantics. DAIMI Report FN-19, Computer Science Department, Aarhus University, 1981.

[33] A. Rensink, W. Vogler. Fair testing. *Inf. Comput.*, 205(2), 2007.

[34] S. Rideau, G. Winskel. Concurrent strategies. In *LICS '11*. IEEE Computer Society, 2011.

[35] P. Sewell. From rewrite rules to bisimulation congruences. In D. Sangiorgi, R. de Simone, eds., *CONCUR*, vol. 1466 of *Lecture Notes in Computer Science*. Springer, 1998.

[36] D. Turi, G. D. Plotkin. Towards a mathematical operational semantics. In *LICS '97*, 1997.