



HAL
open science

Strategies as sheaves and interactive equivalences for CCS

Tom Hirschowitz, Damien Pous

► **To cite this version:**

Tom Hirschowitz, Damien Pous. Strategies as sheaves and interactive equivalences for CCS. 2011.
hal-00555144v1

HAL Id: hal-00555144

<https://hal.science/hal-00555144v1>

Preprint submitted on 12 Jan 2011 (v1), last revised 12 Dec 2012 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Strategies as sheaves and interactive equivalences for CCS

Tom Hirschowitz
CNRS and Université de Savoie, France

Damien Pous
CNRS, France

I. OVERVIEW

We are concerned with reconciling, in the particular case of Milner's CCS [1],

- Joyal, Nielsen, and Winskel's [2], [3] approach (JNW) to concurrency theory, with
- the interactive approach to behavioural equivalences, as advocated, e.g., in De Nicola and Hennessy's *testing* equivalences [4], later revisited by Beffara [5], in Krivine realisability [6], in game semantics [7], or in Girard's *ludics* [8].

JNW propose presheaf categories as a generalisation of transition systems, and hence as a candidate semantics for process algebra. They accordingly equip them with a general notion of *bisimilarity*, which is the queen equivalence of concurrency theory, mainly because it is the finest. JNW argue that their general definition indeed gives bisimilarity on various important examples.

However, presheaf categories do not directly feature any notion of *internal* interaction. Bisimilarity may be considered interactive in a certain sense, but interaction takes place with an external observer [9]. In game semantics, realisability, or ludics, instead, interaction is part of the formalism, and observability is defined relatively to the model at hand. This is also what happens in *testing* equivalences in concurrency, in particular *may*, *must*, and *fair* testing [4], [5].

We are here interested in refining JNW with an internal notion of interaction, and deriving a notion of interactive equivalence. As a first step in this direction, we study the particular case of Milner's CCS [1].

The idea of JNW consists in considering a category \mathbb{P} of (non-empty) *observations* and taking presheaves on it. JNW then argue that functional bisimulations may be characterised by a lifting property w.r.t. representable morphisms. For interleaving semantics, the relevant category has:

- objects: non-empty words over an alphabet \mathcal{A} ;
- morphisms: prefix extensions, e.g., $abc \rightarrow abcd$.

Processes are then presheaves on \mathbb{P} , i.e., functors $\mathbb{P}^{op} \rightarrow \text{Set}$. Presheaves are like trees with edges labelled in \mathcal{A} , while natural transformations are like functional simulations. In traditional game semantics [7], a *strategy* is a prefix-closed set of plays with conditions. If the letters of \mathcal{A} are thought of as moves, a prefix-closed set of plays is the same as a

functor $\mathbb{P}^{op} \rightarrow 2$, where 2 is the category $0 \rightarrow 1$. This notion of strategy is too coarse to detect *branching*, namely, two functors $\mathbb{P}^{op} \rightarrow 2$ are isomorphic when the underlying processes are *trace equivalent*. E.g., this equates $a.(b+c)$ and $ab+bc$. Replacing functors $\mathbb{P}^{op} \rightarrow 2$ with functors $\mathbb{P}^{op} \rightarrow \text{Set}$ yields a new notion of strategy, which is fine enough to detect branching.

The starting point of this paper is to define a category of observations \mathbb{E} derived from CCS syntax, in a way which we think may be made systematic (although we leave this for further work). Observing that this category has a topological nature, we define strategies to be not only presheaves, but *sheaves* [10] over \mathbb{E} , relative to a fixed base position X . Sheaves are here to presheaves what *innocent* strategies are to plain strategies¹.

Seen as indexed over X , the categories of strategies S_X form a *stack* [12], which yields a notion of interaction by amalgamation. We prove a spatial decomposition result, which says that giving a strategy on a position X is the same as giving a strategy for each of its *sequents*, a kind of atomic position. We furthermore prove a temporal decomposition result saying that a strategy is determined by its set of initial states, plus what remains of each of them (as a strategy) after each possible first move.

Finally, we investigate interactive equivalences in this model. We first define a category of *global behaviours* G_X on a position X , which comes with a functor $\text{Gl}: S_X \rightarrow G_X$ from strategies. We then let an *observable criterion* on X be a subcategory $\perp_X \hookrightarrow G_X$, and derive from it an assignment $S \mapsto S^\perp$ of *orthogonal contexts* to each strategy S , and a notion of \perp -equivalence $S \sim_\perp S'$, holding exactly when $S^\perp = S'^\perp$.

We then instantiate this with the *must* criterion \perp^m . Traditionally, *must* testing has to do with *maximal* traces of a process satisfying some condition, a trace being maximal when it is infinite. We define *must* testing in our setting in a similar way. But here, thanks to the topological aspect of our approach, an infinite observation may not be maximal. We then define the *fair* criterion \perp^f , mimicking its standard definition. In the traditional setting, *may* and *must* testing differ; we show that in our setting they coincide.

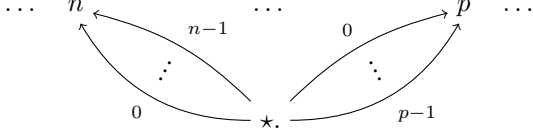
This work has been partially funded by French ANR projects Choco ANR-07-BLAN-0324 and PiCoq ANR-2010-BLAN-0305.

¹In passing, sheaves yield a notion of non-deterministic, innocent strategy, which might lead to an alternative to Harmer's approach [11].

II. OBSERVATIONS AS PRESHEAVES

A. The base category

To define our category of observations, we start with a category of *positions* \mathbb{B} which is the category of presheaves on the category \mathbb{C}_1 freely generated by the graph

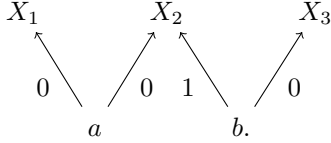


Formally, \mathbb{C}_1 has as objects the natural numbers plus one special object \star , with as non-trivial homsets the $\mathbb{C}_1(\star, n)$ which are n , seen as an ordinal, i.e., $0 \dots n-1$.

Positions are like hypergraphs. For example, the presheaf F with:

- $F(\star) = \{a, b\}$,
- $F(1) = \{X_1, X_3\}$,
- $F(2) = \{X_2\}$,
- $F(\star \xrightarrow{0} 1)(X_1) = a$, notation: $X_1 \cdot 0 = a$,
- $X_2 \cdot 0 = a$, $X_2 \cdot 1 = b$,
- $X_3 \cdot 0 = b$.

has a category of elements looking like:



We think of $F(\star)$ as the set of *channels*, or *names*, on which the *players* (the other elements) will communicate.

Communication takes place via *moves*, which we will define as certain cospans of presheaves on an extension \mathbb{C} of \mathbb{C}_1 , which we now gradually construct.

Recall the syntax of finitary CCS²

$$P, Q, \dots ::= 0 \mid (P \mid Q) \mid \bar{a}.P \mid a.P \mid \nu a.P,$$

where only the ν operation is binding a (which is hence subject to α renaming). The way we locally augment \mathbb{C}_1 for each construction is depicted in Fig. 1 (left), together with the category of elements of an example representable (right) on one of the introduced objects.

Let us start with input. The formation rule is

$$\frac{a_0, \dots, a_{n-1} \vdash P}{a_0, \dots, a_{n-1} \vdash a_i.P}.$$

Here, a_1, \dots, a_n is a list of names, and the rule says that if we have formed a process P using these names, then we may form the process $a_i.P$ which tries to input on a_i and then continues with P . To model this, we add an object $\iota_{n,i}^-$ to \mathbb{C}_1

²We rely on finitary CCS to define \mathbb{C} , but our model actually leaves room for infinitary processes. We have defined a translation from CCS processes with recursive definitions to our model, but not yet proved anything on it.

for each pair $i \in n$, with morphisms $s, t: n \rightarrow \iota_{n,i}^-$, and we quotient out by the equations $s \circ j = t \circ j$ for all $j \in n$. Pictorially, we obtain a graph locally looking like the left of the first row in Fig. 1. For example, the category of elements of the representable $\iota_{3,2}^-$ is the partially ordered set generated by the right of the first row in Fig. 1, where we write composition as juxtaposition, e.g., $t1 = t \circ 1$. For the output formation rule $\bar{a}.P$, we do exactly the same, naming the new objects $\iota_{n,i}^+$.

For parallel composition $P \mid Q$, we add a vertex π_n to \mathbb{C}_1 for each n , plus morphisms $s, t_1, t_2: n \rightarrow \pi_n$, and quotient out by $t_1 \circ i = s \circ i = t_2 \circ i$ for all $i \in n$. The extension is depicted in Fig. 1 in the second row, following the pattern of the first. The depicted representable is π_3 .

For name creation $\nu a.P$, depicted in row 3, we add a vertex ν_n to \mathbb{C}_1 for each n , plus morphisms $n \xrightarrow{s} \nu_n \xleftarrow{t} n+1$, and quotient out by $s \circ i = t \circ i$ for $i \in n$. The depicted representable is ν_2 .

Finally, we introduce a *tick* move, which will be important for observation. We add a vertex \heartsuit_n for each n to \mathbb{C}_1 with morphisms $s, t: n \rightarrow \heartsuit_n$, and quotient out by $s \circ i = t \circ i$ for all $i \in n$. The depicted representable is \heartsuit_3 .

Up to now, we have extended \mathbb{C}_1 to model process formation rules. Altogether, this yields a category \mathbb{C}_2 . We now model the *synchronisation* relation

$$a.P \mid \bar{a}.Q \rightarrow P \mid Q.$$

We do so by adding a new object $\tau_{n,i,m,j}$ to \mathbb{C}_2 , for all $i \in n, j \in m$, with morphisms $\iota_{n,i}^+ \xrightarrow{\epsilon} \tau_{n,i,m,j} \xleftarrow{\rho} \iota_{m,j}^-$, and quotienting out by $\epsilon \circ s \circ i = \rho \circ s \circ j$ (for only the given i and j this time). The depicted representable is $\tau_{3,2,2,1}$.

Calling this new category \mathbb{C}_3 , or $\widehat{\mathbb{C}}$ for short, this makes us a sequence of categories $\mathbb{C}_0 = \{\star\} = \text{Set}$, \mathbb{C}_1 , \mathbb{C}_2 , \mathbb{C}_3 , which yields a sequence of truncation functors

$$\widehat{\mathbb{C}}_3 \rightarrow \widehat{\mathbb{C}}_2 \rightarrow \widehat{\mathbb{C}}_1 \rightarrow \widehat{\mathbb{C}}_0.$$

We further say that a presheaf has dimension i when it is empty above \mathbb{C}_i . We call *interfaces* the presheaves of dimension 0, and *positions* the presheaves of dimension 1. The elements of dimension 1, i.e., above some n , are the *players*, those of dimension 2 are the *moves*, while those of dimension 3 are the *synchronisations*.

Presheaves on $\widehat{\mathbb{C}}$ model executions, or traces, of CCS processes, up to certain permutations. Consider for example the process $b \mid \bar{a}$. We may naively model its two maximal traces as the colimit in $\widehat{\mathbb{C}}$ of the diagram

$$\iota_{2,1}^- \xleftarrow{s} 2 \xrightarrow{t_1} \pi_2 \xleftarrow{t_2} 2 \xrightarrow{s} \iota_{2,0}^+.$$

If we want to model traces of $a \mid \bar{a}$, then we may replace 2 by 1 above, and there is one more maximal trace (the τ transition $a \mid \bar{a} \rightarrow 0 \mid 0$). Presheaves also model infinite executions, e.g., of the process $\mu X.(X \mid X)$ which indefinitely forks.

However, there are also some presheaves which do not correspond to any sensible trace.

Example 1. *The quotient of the representable $\iota_{1,0}$ under $s = t$ could be meaningful in other contexts, but here we want to rule*

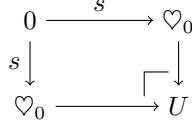
Local augmentation	Example representable

Fig. 1. Augmentations of \mathbb{C} and example representables

it out (actually following JNW, which systematically unfold cyclic processes).

Example 2. The quotient of π_0 under $t_1 = t_2$, in which the two players obtained after forking are actually equal.

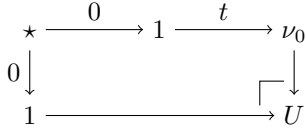
Example 3. The pushout



in which one player, here s , plays two moves at the same time.

Example 4. There is a similar problem one level up: two synchronisations may have one input or one output in common, which should also be considered as wrong.

Example 5. The pushout

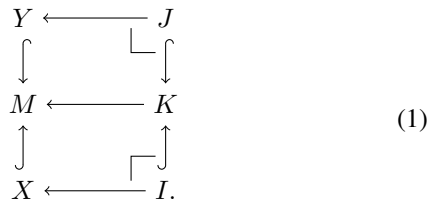


in which the name created by ν_0 is already known by the lower-left player, or alternatively where the latter player spontaneously appears during execution.

To control this, we now define particular classes of cospans of presheaves, moves and observations.

B. Moves and observations

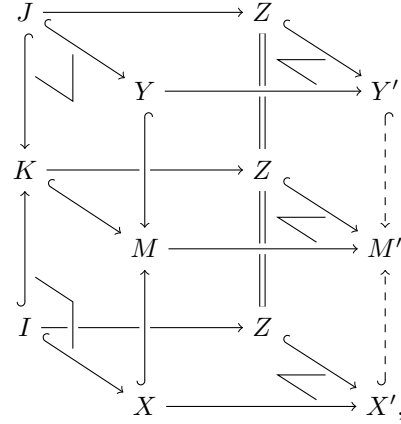
Basic moves will be certain diagrams $\mathbb{S} \rightarrow \widehat{\mathbb{C}}$, where \mathbb{S} is the category generated by:



In basic moves, both squares will be pullbacks, the indicated arrows will be monos, the images of I, J, K will have dimension 0, and X and Y will be positions.

We will then use the functor category $[\mathbb{S}, \widehat{\mathbb{C}}]$ to define general moves as basic moves embedded in larger positions. The intuition is close to some graph transformation formalisms: the left border is the typical move in isolation, i.e., without any passive context; the right border is the interface to the rest of the world; moves will be obtained by glueing the move in isolation to a passive context along the interface. A passive context is a cospan of the form $Z = Z = Z$.

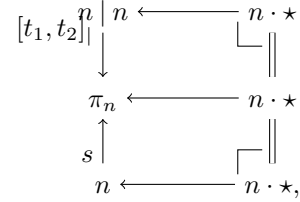
We will define basic moves below, but we may already define: a *move* is a cospan $X' \hookrightarrow M' \hookrightarrow Y'$ obtained from a basic move $X \hookrightarrow M \hookrightarrow Y$ as in



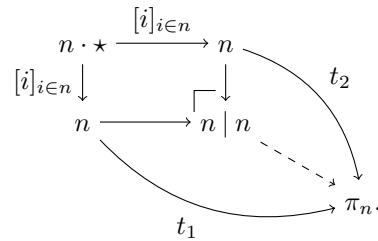
where the dashed arrows are obtained by universal property of pushout. The dashed arrows are indeed mono (using the fact that monos in Set are coproduct injections).

There will be six classes of basic moves, indexed by the arity of the involved players: forking, name creation, tick, and three classes of basic moves for synchronisation. For synchronisation, the full synchronisation τ is not enough: we also want to consider players synchronising with the external world, i.e., with players not part of the position. The three communication moves will thus be synchronisation, emission, and reception. Let us define the six classes of basic moves precisely.

Forking is the n -indexed family of diagrams



where $n \cdot \star$ is the n -fold coproduct of (the representable) \star , which has actually n elements over \star hence the notation, and where $n \mid n$ is the pushout and $[t_1, t_2] \downarrow$ is the induced dashed arrow in



All arrows from $n \cdot \star$ are the obvious ones.

Name creation is the n -indexed family of diagrams

Definition 2. Let a sieve S on $X \hookrightarrow U$ in \mathbb{E} be view-covering when it is jointly surjective in dimension 1.

Let a sieve S on X in \mathbb{B} be covering when it is jointly surjective in dimension 1.

Proposition 1. The functor $\pi: \mathbb{E} \rightarrow \mathbb{B}$ is a morphism of sites.

We now define strategies as sheaves, for which we need to relativise to a base position X .

Definition 3. Let $(\mathbb{E})_X$ have as objects $U \hookrightarrow Y \rightarrow X$, and as morphisms transformations

$$\begin{array}{ccc} U & \longrightarrow & U' \\ \uparrow & & \uparrow \\ Y & \longrightarrow & Y' \\ & \searrow & \swarrow \\ & & X. \end{array}$$

There is an obvious projection functor $p: (\mathbb{E})_X \rightarrow \mathbb{E}$ sending $U \hookrightarrow Y \rightarrow X$ to $Y \hookrightarrow U$, through which the categories $(\mathbb{E})_X$ inherit a Grothendieck topology from \mathbb{E} : a sieve S covering $U \hookrightarrow Y \rightarrow X$ iff the projection $p(S)$ covers $Y \hookrightarrow U$.

Definition 4. Let the category S_X of strategies on X be $\text{Sh}((\mathbb{E})_X)$.

For such a strategy S and observation $X \hookrightarrow U$, we can think of each element of $S(U)$ as a set of *states* for S at each stage of U .

We end the section by describing a canonical covering for observations, which will also yield a canonical covering for positions.

A *sequent* is a connected position with exactly one player, or equivalently a presheaf x obtained by pushout

$$\begin{array}{ccc} n \cdot \star & \xrightarrow{e} & I \\ \downarrow & \lrcorner & \downarrow \\ n & \longrightarrow & x \end{array}$$

with I of dimension 0, and e epi. A sequent is (isomorphic to) a representable exactly when e is an isomorphism.

A *elementary quasi-view* V from x to y is a cospan $x \hookrightarrow V \hookrightarrow y$ isomorphic to a composite of

- a move from a sequent x ,
- followed by a restriction to a sequent y ,

i.e., a colimit

$$\begin{array}{ccccc} & & & X & \\ & & & \parallel & \\ x & \hookrightarrow & M & \hookrightarrow & X \\ & \searrow & \lrcorner & \swarrow & \\ & & V & & y \end{array} \quad (2)$$

where x and y are sequents, and M is a move. When x is representable, then V is an *elementary view*.

Lemma 1. Any such elementary view has y representable.

Proof: By case inspection. \square

Lemma 2. Any elementary quasi-view is the codomain of an epimorphism from an elementary view.

Proof: By definition of moves from basic moves, and of sequents. \square

Definition 5. A quasi-view is an observation isomorphic to a possibly denumerable “composition” of elementary quasi-views.

A view is the same with elementary views.

A consequence of Lemma 2 is:

Lemma 3. Any quasi-view is the codomain of an epimorphism from a view.

Finite views form a canonical covering, in the sense that the next two propositions hold.

Proposition 2. For any observation $X \hookrightarrow U$, the sieve generated by morphisms from finite views into U is covering.

Proposition 3. Any covering sieve contains all morphisms from finite views.

For proving them, we introduce the following tool. Let the *causal relation* on an observation $X \hookrightarrow U$ be the set of pairs $x \prec y$ of players of U generated by:

- $y = x \cdot t$ or $y = x \cdot t_1$ or $y = x \cdot t_2$, or
- $x = y \cdot s$.

Let also $\downarrow x$ be the set of all elements $y \prec x$.

Lemma 4. The causal relation on an observation $X \hookrightarrow U$ is a forest, whose roots are exactly the players in X .

Lemma 5. For any observation $X \hookrightarrow U$ and element $x \in U$ of dimension 1, there is a view $n \hookrightarrow V$ with a morphism

$$\begin{array}{ccc} V & \longrightarrow & U \\ \uparrow & & \uparrow \\ n & \longrightarrow & X, \end{array} \quad (3)$$

such that x is in the image of V .

If x is furthermore maximal for \prec_U , then V is unique up to isomorphism, and for fixed V the morphism (3) is unique.

Proof: For any such element x of U , consider the smallest subpresheaf $W \hookrightarrow U$ containing the downwards cone $\downarrow x$. This W is a finite quasi-view with a morphism to U

$$\begin{array}{ccc} W & \hookrightarrow & U \\ \uparrow & & \uparrow \\ y & \hookrightarrow & X, \end{array}$$

where y is the bottom element of $\downarrow x$. By Lemma 3, there is an epimorphism from a view V to W , hence the composite morphism $V \rightarrow U$ has x in its image.

This V is furthermore a minimal such view, up to isomorphism, since any other such view would have to have the whole $\downarrow x$ in its image. Thus, if x was further maximal for \prec_U in its dimension, V is also maximal view of U , hence is unique up to isomorphism. \square

Proof of Proposition 2: By Lemma 5. \square

Proof of Proposition 3: Consider any sieve S covering U , and a morphism $h: V \rightarrow U$ from a finite view V .

If V is the empty composition and h amounts to $x: n \rightarrow X$, then S has a morphism $h: U' \rightarrow U$ with $x \in \text{Im}(h)$. Let $x' \in U'$ be an antecedent of x . The morphism $n \xrightarrow{x'} U' \xrightarrow{h} U$ is in S and is equal to x .

If V has moves, then these are ordered linearly (via \prec_V), and hence there is a maximum such move $m \in V$. Because S is covering, there is a morphism $h': U' \rightarrow U$ in S with $m \in \text{Im}(h')$. Because U' is an observation, it has to contain $\downarrow m$, hence there is a morphism $V \rightarrow U'$, and thus the composite $V \rightarrow U' \rightarrow U$ is in S . But this has to be h be the uniqueness part of Lemma 5. \square

B. Strategies as a stack

Stepping back a little, we have defined a category S_X of strategies for each position X , but we have not yet understood the global structure of this assignment, e.g., how strategies on X relate to strategies on Y through a morphism of positions $Y \rightarrow X$.

For any morphism $h: Y \rightarrow X$ in \mathbb{B} , composition with h induces a functor $\Sigma_h: (\mathbb{E})_Y \rightarrow (\mathbb{E})_X$ sending $(U \leftarrow Z \rightarrow Y)$ to $(U \leftarrow Z \rightarrow Y \rightarrow X)$, which preserves covering sieves. Restriction along this functor (i.e., composition with its opposite) induce a functor $h^*: S_X \rightarrow S_Y$ between the categories of sheaves.

Recall the Grothendieck topology on \mathbb{B} in Definition 2, and the definition of a *stack* [12].

Proposition 4. *The assignments $X \mapsto S_X$ and $h \mapsto h^*$ form a functor $S: \mathbb{B}^{op} \rightarrow \text{CAT}$, which is a stack.*

Proof: Essentially the same arguments as in Vistoli [12, Examples 3.20 and 4.11], with the difference that the slice is here taken through the functor $\pi: \mathbb{E} \rightarrow \mathbb{B}$. \square

In a sheaf F on any category \mathbb{C} , seeing a covering sieve on an object U as a subpresheaf $S \hookrightarrow U$ of the representable U , the map induced by the inclusion

$$\widehat{\mathbb{C}}(U, F) \rightarrow \widehat{\mathbb{C}}(S, F)$$

is a bijection. Stacks are like sheaves of categories, and the inclusion induces not only a map but a functor

$$\widehat{\mathbb{C}}(U, F) \rightarrow \widehat{\mathbb{C}}(S, F)$$

which is required to be not only an isomorphism but an equivalence of categories. Since strategies S_X form for each X a proper category, they only yield a stack, not a sheaf.

Remark 1. *Strictly speaking, each S_X is only locally small. Hence, S is a functor to the large category of locally small categories.*

Observe now that any position X is covered by the sieve S generated by morphisms $n \rightarrow X$. For any two distinct such morphisms, their pullback I contains only channels, and hence $S_I \simeq 1$, i.e., there is essentially a unique strategy on I . So, giving a compatible family of strategies for the sieve S amounts to giving a strategy on n for each $n \rightarrow X$, since the restrictions on intersections, having to be the unique strategy there, are automatically compatible. Letting $\text{Sq}(X) = \prod_n X(n)$, we have proved:

Proposition 5. *For any position X , $S_X \simeq \prod_{(n,x) \in \text{Sq}(X)} S_n$.*

C. Strategies and sketches

We have just shown that in the static direction, i.e., relatively to positions, S is a stack. We now show that in the dynamic direction S is close to being a model for a sketch, in the following sense. For any representable n , we may choose exactly one representative of each isomorphism class (considered in \mathbb{E}_n) of moves $n \hookrightarrow M \leftarrow X_M$, this yielding a (finite) set \mathcal{M}_n of moves. Let us denote by $(M: X_M \rightarrow n)_{M \in \mathcal{M}_n}$ the corresponding family.

Definition 6. *Let \mathbb{E}_v be the (bi)category with positions as objects, as morphisms $Y \rightarrow X$ the diagrams $X \hookrightarrow U \leftarrow Y$ where $X \hookrightarrow U$ is an observation, and Y is a position containing players not in the image of s , and as 2-cells all morphisms of cospans.*

We call the 1-cells of \mathbb{E}_v *stories*, to emphasise the distinction with observations. Observations $X \hookrightarrow U$ may be seen as stories $U: 0 \rightarrow X$.

The families $(M: X_M \rightarrow n)_{M \in \mathcal{M}_n}$, seen as cones from n in \mathbb{E}_v^{op} , equip the latter with a (product) sketch structure. Recalling that a *model* for a sketch \mathbb{C} in a category \mathbb{D} is a functor $\mathbb{C} \rightarrow \mathbb{D}$ sending the distinguished cones in \mathbb{C} to limit cones, S being a model for the above sketch in CAT would amount to isomorphisms of categories

$$S_n \cong \prod_{M \in \mathcal{M}_n} S_{X_M}, \quad (4)$$

for all representables n . These do not hold in general because of non-determinism. Instead, for a category \mathbb{C} , recall from Jacobs [13, Definition 1.2.1] that $\text{Fam}(\mathbb{C})$ denote the category with

- objects set-indexed families of objects in \mathbb{C} , i.e., pairs of a set X and a mapping $c: X \rightarrow \mathbb{C}_0$, and
- as morphisms $(X, c) \rightarrow (Y, c')$ the pairs of a function $f: X \rightarrow Y$ and $u: X \rightarrow \mathbb{C}_1$ such that $s \circ u = c$, and $t \circ u = c' \circ f$.

We will now show that S is a model for the sketch \mathbb{E}_v^{op} in the relaxed sense that for any representable n , there is an equivalence of categories

$$S_n \simeq \text{Fam} \left(\prod_{M \in \mathcal{M}_n} S_{X_M} \right).$$

Comparing with (4), this is a way of taking into account both non-determinism and pseudo-ness, not so surprising when taking models in a 2-category instead of a category. Otherwise said, a strategy is determined by

- its initial states, and
- what remains of them after each possible move.

But before showing that \mathbb{S} decomposes this way, we must exhibit it as a pseudo-functor $\mathbb{E}_v^{op} \rightarrow \text{CAT}$. A first observation is that views induce change-of-base operations on strategies. For any representable n' , a strategy on n' is determined by its value on finite views from n' , so we may view it as a functor $(\mathbb{V}_{n'})^{op} \rightarrow \text{Set}$, where $\mathbb{V}_{n'}$ is the full subcategory of $\mathbb{E}_{n'}$ consisting of finite views. So, given a strategy $S \in \mathbb{S}_n$ and a view $V: n' \rightarrow n$, the following assignment determines a strategy $S(V)(S)$ on n' :

$$\begin{aligned} (\mathbb{V}_{n'})^{op} &\rightarrow \text{Set} \\ U &\mapsto S(V \otimes U) \\ h &\mapsto S(V \otimes h) \quad \text{for } h: U \rightarrow U' \text{ in } \mathbb{V}_{n'} \end{aligned}$$

Here, $V \otimes U$ denotes the composition

$$0 \xrightarrow{U} n' \xrightarrow{V} n$$

in \mathbb{E}_v , and $V \otimes h$ denotes the morphism of observations of base n induced between the colimits of the columns in

$$\begin{array}{ccc} U & \xrightarrow{h} & U' \\ \updownarrow & & \updownarrow \\ n' & \xlongequal{\quad} & n' \\ \updownarrow & & \updownarrow \\ V & \xlongequal{\quad} & V \\ \updownarrow & & \updownarrow \\ n & \xlongequal{\quad} & n \end{array}$$

by universal property of colimits.

Now, for a morphism of sheaves $f: S \rightarrow S'$ in \mathbb{S}_n , the assignment $U \mapsto f_{V \otimes U}$ defines a morphism

$$S(V)(f): S(V)(S) \rightarrow S(V)(S'),$$

which induces a pseudo-functor

$$S(V): \mathbb{S}_n \rightarrow \mathbb{S}_{n'}.$$

We may extend the construction to arbitrary positions and stories. For an arbitrary story $U: Y \rightarrow X$, by Proposition 5, we have

$$\mathbb{S}_Y \simeq \prod_{(n', y) \in \text{Sq}(Y)} \mathbb{S}_{n'}.$$

We distinguish notationally by indicating (n', y) as a subscript, e.g., $S(U \leftarrow Y \rightarrow X)$ means the value of S at $U \leftarrow Y \rightarrow X$, while $S_{(n,x)}$ denotes the strategy on n obtained by restricting S along x .

Recall now Lemma 5. When we view $U: Y \rightarrow X$ as a morphism in \mathbb{E}_v , any $x: n \rightarrow Y$ corresponds to an element of

U , maximal for \prec_U . Let $e_x: V_x \rightarrow U$ be the corresponding morphism with V_x a view, and x in its image, and let $[x]: n_x \rightarrow X$ be the corresponding morphism between bases. This induces a diagram:

$$\begin{array}{ccc} n & \xrightarrow{x} & Y \\ \downarrow & & \downarrow \\ V_x & \xrightarrow{e_x} & U \\ \updownarrow & [x] & \updownarrow \\ n_x & \xrightarrow{\quad} & X. \end{array}$$

Define then:

$$S(U)(S)_{(n,x)} = S(V_x)(S_{(n_x, [x])}). \quad (5)$$

By definition, on a view $V \leftarrow n \xrightarrow{x} Y$, we have

$$S(U)(S)(x, V) = S(V_x \otimes V \leftarrow n_x \xrightarrow{[x]} X), \quad (6)$$

which is enough to define $S(U)(S)$ as a strategy on Y .

Proposition 6. *This assignment yields a functor $S(U): \mathbb{S}_X \rightarrow \mathbb{S}_Y$, and thus beyond being a stack $\mathbb{B}^{op} \rightarrow \text{CAT}$, \mathbb{S} is also a (bi)functor $\mathbb{E}_v^{op} \rightarrow \text{CAT}$.*

Let us now return to our decomposition result. Given a strategy $S \in \mathbb{S}_n$, let $S_0 = S(id_n)$. Each $s \in S_0$ determines a substrategy $S|_s \subseteq S$ by taking pullbacks for each view V on n :

$$\begin{array}{ccc} S|_s(V) & \longrightarrow & 1 \\ \downarrow \lrcorner & & \downarrow s \\ S(V) & \xrightarrow{\quad} & S(!), \end{array}$$

where $!$ denotes the inclusion $id_n \hookrightarrow V$. $S|_s$ is indeed a sheaf because the property of having s as restriction to id_n is stable under amalgamation in any sheaf S . Actually, restriction to s extends to a functor $S/\mathbb{S}_n \rightarrow S|_s/\mathbb{S}_n^!$, where $\mathbb{S}_n^!$ denotes the full subcategory of \mathbb{S}_n spanning the sheaves S' having a singleton as $S'(id_n)$. This functor $-|_s$ sends any $\alpha: S \rightarrow S'$ to the natural transformation defined on each view V by universal property of pullback as in

$$\begin{array}{ccccc} & & S'_{|s'}(V) & \xrightarrow{\quad} & 1 \\ & \swarrow (\alpha|_s)_V & \downarrow & \lrcorner & \downarrow s' \\ S|_s(V) & \xrightarrow{\quad} & 1 & \xrightarrow{\quad} & S'(id_n) \\ \downarrow & \downarrow & \downarrow & \downarrow s & \downarrow \\ S(V) & \xrightarrow{\quad} & S'(V) & \xrightarrow{\quad} & S'(id_n) \\ \downarrow & \downarrow \alpha_V & \downarrow & \downarrow S'(!) & \downarrow \alpha_{id_n} \\ S(V) & \xrightarrow{\quad} & S(!) & \xrightarrow{\quad} & S(id_n), \end{array} \quad (7)$$

where $s' = \alpha_{id_n}(s)$.

Let now $E(S): S_0 \rightarrow \prod_{M \in \mathcal{M}_n} S_{X_M}$ be defined by

$$E(S)(s)(M) = S(M)(S|_s).$$

For any morphism $\alpha: S \rightarrow S'$ in S_n , let $E(\alpha)$ be the pair of the mapping $\alpha_{id_n}: S_0 \rightarrow S'_0$, and the mapping sending $s \in S_0$ to the product over $M \in \mathcal{M}_n$ of all

$$S(M)(\alpha|_s): S(M)(S|_s) \rightarrow S(M)(S'|_s),$$

with $\alpha|_s$ as in (7). This defines a functor

$$E: S_n \rightarrow \text{Fam} \left(\prod_{M \in \mathcal{M}_n} S_{X_M} \right),$$

and we have

Theorem 1. *E is an equivalence of categories.*

Proof: The proof consists of Lemmas 8, 9, and 10 in Appendix A, respectively showing that E is faithful, full, and essentially surjective. \square

Corollary 1. *For any position X, we have*

$$S_X \simeq \prod_{(n,x) \in \text{Sq}(X)} \text{Fam} \left(\prod_{M \in \mathcal{M}_n} \prod_{(n',x') \in \text{Sq}(X_M)} S_{n'} \right).$$

IV. INTERACTIVE EQUIVALENCES

A. *Fair testing vs. must testing: the standard case*

An important part of concurrency theory consists in studying *behavioural equivalences*. Each behavioural equivalence being supposed to define when two processes behave the same, it might seem paradoxical to consider several of them. Van Glabbeek [9] argues that each behavioural equivalence corresponds to a physical scenario for observing processes.

A distinction we wish to make here is between *fair* scenarios, and *potentially unfair* ones. An example of a fair scenario is when parallel composition of processes is thought of as modelling different physical agents, e.g., in a game with several players. Otherwise said, players are really independent. On the other hand, an example of a potentially unfair scenario is when parallelism is implemented via a scheduler.

Mainstream notions of processes, e.g., transition systems or automata, are actually unfair, as the following example shows. Consider a looping process Ω , which has a silent transition τ to itself. The process $P = (\Omega|\bar{a})$, which in parallel plays Ω and tries to synchronise on a , has an infinite trace

$$P \xrightarrow{\tau} P \xrightarrow{\tau} \dots$$

This has consequences on so-called *testing* equivalences [4]. Let \heartsuit be a fixed action.

Definition 7. *A process P is must orthogonal to a context C, notation $P \perp^m C$, when all maximal traces of $C[P]$ play \heartsuit at some point.*

Here, maximal means either infinite or finite without extensions. Let P^{\perp^m} be the set of all contexts must orthogonal to P.

Definition 8. *P and Q are must equivalent, notation $P \sim_m Q$, when $P^{\perp^m} = Q^{\perp^m}$.*

In transition systems, or automata, recalling P above and letting $Q = \Omega$, we have

$$P \sim_m Q.$$

This might be surprising, because the context $C = a.\heartsuit \mid \square$ intuitively should distinguish P from Q, by being orthogonal to P but not to Q. However, it is not orthogonal to P, because $C[P]$ has an infinite looping trace giving priority to Ω .

This looping trace is unfair, because the synchronisation on a is never performed. Thus, one may view the equivalence $P \sim_m Q$ as taking into account potential unfairness of a hypothetical scheduler. Usually, concurrency theorists consider this too coarse, and resort to *fair* equivalence.

Definition 9. *A process P is fair orthogonal to a context C, notation $P \perp^f C$, when all finite traces of $C[P]$ extend to traces that play \heartsuit at some point.*

Again, P^{\perp^f} denotes the set of all contexts fair orthogonal to P.

Definition 10. *P and Q are fair equivalent, notation $P \sim_f Q$, when $P^{\perp^f} = Q^{\perp^f}$.*

This solves the issue, i.e., $P \sim_f Q$.

Thus, the mainstream setting for behavioural equivalences is unfair. Our setting is more flexible, in the sense that it leaves room for a less linear notion of a maximal trace. In the terms of the previous section, it allows viewing the looping trace

$$P \xrightarrow{\tau} P \xrightarrow{\tau} \dots$$

as non-maximal. In the next sections, we define fair and must testing in our context, and show that they coincide. We start by defining an abstract notion of interactive equivalence (still in the particular case of CCS), and then instantiate this to define fair and must testing.

B. *Interactive equivalence*

First, testing equivalences rest on a notion of a global computation, i.e., one that may not interact with the outside world. Hence, we define an observation $X \hookrightarrow U$ to be *closed-world* when all inputs and outputs are part of a synchronisation, i.e., when both maps

$$\prod_{n,i} U(t_{n,i}^+) \xleftarrow{\epsilon} \prod_{n,i,m,j} U(\tau_{n,i,m,j}) \xrightarrow{\rho} \prod_{n,i} U(t_{n,i}^-) \quad (8)$$

are surjective.

Definition 11. *Let $\mathbb{W} \hookrightarrow \mathbb{E}$ be the full subcategory of closed-world observations.*

Let $\mathbb{W}(X)$ be the fibre over X for the projection functor $\mathbb{W} \rightarrow \mathbb{B}$.

Definition 12. *Let the category of global behaviours on X be simply $\mathbb{G}_X = \overline{\mathbb{W}(X)}$.*

The canonical inclusion $\mathbb{W}(X) \hookrightarrow (\mathbb{E})_X$ induces a functor $\text{Gl}: \mathbb{S}_X \rightarrow \mathbb{G}_X$.

Definition 13. An observable criterion consists for all positions X , of a subcategory $\perp_X \hookrightarrow \mathbb{G}_X$.

Definition 14. For any strategy S on X and any pushout P

$$\begin{array}{ccc} I & \longrightarrow & Y \\ \downarrow & & \downarrow \\ X & \longrightarrow & Z \end{array} \quad (9)$$

of positions with I of dimension 0, let S^{\perp_P} be the class of all strategies T on Y such that $\text{Gl}(S \parallel T) \in \perp_Z$.

Here \parallel denotes amalgamation in the stack \mathbb{S} .

Definition 15. Two strategies $S, S' \in \mathbb{S}_X$ are \perp -equivalent, notation $S \sim_{\perp} S'$, iff for all pushouts P , $S^{\perp_P} = S'^{\perp_P}$.

C. Fair vs. must

We start by defining fair and must testing. Let a closed-world observation be *successful* when it contains a \heartsuit_n . Furthermore, given a global behaviour $G \in \mathbb{G}_X$, an *extension* of a state $\sigma \in G(U)$ to U' is a $\sigma' \in G(U')$ with $i: U \rightarrow U'$ and $\sigma' \cdot i = \sigma$. The extension σ' is *successful* when U' is.

Definition 16. The fair criterion \perp^f contains all global behaviours G such that any state $\sigma \in G(U)$ for finite U admits a successful extension.

Now call an extension of $\sigma \in G(U)$ *strict* when $U \rightarrow U'$ is not surjective. For any global behaviour $G \in \mathbb{G}_X$, a state $\sigma \in G(U)$ is *G-maximal* when it has no strict extension.

Definition 17. Let the must criterion \perp^m consist of all global behaviours G such that for all closed-world U , and *G-maximal* $\sigma \in G(U)$, U is *successful*.

We now show that fair and must equivalence coincide. The key result for this is:

Theorem 2. For any strategy S on X , any state $\sigma \in \text{Gl}(S)(X \hookrightarrow U)$ with finite U admits a $\text{Gl}(S)$ -maximal extension.

The proof is in Section B. Thanks to the theorem, we have:

Lemma 6. For all $S \in \mathbb{S}_X$, $\text{Gl}(S) \in \perp_X^m$ iff $\text{Gl}(S) \in \perp_X^f$.

Proof: Let $G = \text{Gl}(S)$.

(\Rightarrow) By the theorem, any state $\sigma \in G(U)$ has a *G-maximal* extension $\sigma' \in G(U')$, which is successful by hypothesis, hence σ has a successful extension.

(\Leftarrow) Any *G-maximal* $\sigma \in G(U)$ admits by hypothesis a successful extension which may only be on U by *G-maximality*, and hence U is successful. \square

Now comes the expected result:

Theorem 3. For all $S, S' \in \mathbb{S}_X$, $S \sim_{\perp^m} S'$ iff $S \sim_{\perp^f} S'$.

Proof: (\Rightarrow) Consider two strategies S and S' on X , and a strategy T on Y (as in the pushout P). We have:

$$\begin{aligned} \text{Gl}(S \parallel T) \in \perp^f & \quad \text{iff } \text{Gl}(S \parallel T) \in \perp^m \\ & \quad \text{iff } \text{Gl}(S' \parallel T) \in \perp^m \\ & \quad \text{iff } \text{Gl}(S' \parallel T) \in \perp^f. \end{aligned}$$

(\Leftarrow) Symmetric. \square

V. CONCLUSION

Our next task is clearly to link with CCS. Namely, we should provide a translation of CCS terms into strategies, and explore which equivalence on CCS is induced by interactive equivalence, for a given \perp .

Longer-term perspectives include the treatment of more complicated calculi like π or λ . In particular, contraction will challenge our approach seriously. An even longer-term hope is to be able to abstract over our approach. Is it possible to systematise the process starting from a calculus as studied in programming language theory, and generating its stack of behaviours? If this is ever understood, the next question is: when does a translation between two such calculi preserve a given behavioural equivalence? Finding general criteria for this might have useful implications in programming languages, especially compilation.

REFERENCES

- [1] R. Milner, *A Calculus of Communicating Systems*, ser. LNCS. Springer, 1980, vol. 92.
- [2] A. Joyal, M. Nielsen, and G. Winskel, “Bisimulation and open maps,” in *LICS*. IEEE Computer Society, 1993, pp. 418–427.
- [3] S. Kasangian and A. Labella, “Observational trees as models for concurrency,” *Mathematical Structures in Computer Science*, vol. 9, no. 6, pp. 687–718, 1999.
- [4] R. D. Nicola and M. Hennessy, “Testing equivalences for processes,” *Theor. Comput. Sci.*, vol. 34, pp. 83–133, 1984.
- [5] E. Beffara, “Logique, réalisabilité et concurrence,” Ph.D. dissertation, Université Paris 7, Dec. 2005.
- [6] J.-L. Krivine, “Dependent choice, ‘quote’ and the clock,” *Theor. Comput. Sci.*, vol. 308, no. 1-3, pp. 259–276, 2003.
- [7] M. Hyland, *Semantics and Logics of Computation*. Cambridge University Press, 1997, ch. Game Semantics.
- [8] J.-Y. Girard, “Locus solum: From the rules of logic to the logic of rules,” *Mathematical Structures in Computer Science*, vol. 11, no. 3, pp. 301–506, 2001.
- [9] R. J. van Glabbeek, “The linear time-branching time spectrum (extended abstract),” in *CONCUR*, ser. Lecture Notes in Computer Science, J. C. M. Baeten and J. W. Klop, Eds., vol. 458. Springer, 1990, pp. 278–297.
- [10] S. MacLane and I. Moerdijk, *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*, ser. Universitext. Springer, 1992. [Online]. Available: <http://www.worldcat.org/isbn/0387977104>
- [11] R. Harmer and G. McCusker, “A fully abstract game semantics for finite nondeterminism,” in *LICS*, 1999, pp. 422–430.
- [12] A. Vistoli, “Notes on Grothendieck topologies, fibered categories and descent theory,” *ArXiv Mathematics e-prints*, 2007.
- [13] B. Jacobs, *Categorical Logic and Type Theory*, ser. Studies in Logic and the Foundations of Mathematics. Amsterdam: North Holland, 1999, no. 141.

APPENDIX A
DYNAMIC DECOMPOSITION

To start with, we have:

Lemma 7. *Any view $n \hookrightarrow V$ either is an isomorphism or is the colimit of a diagram*

$$\begin{array}{ccccc}
 n & & & n_V & \\
 \searrow & & \swarrow & \searrow & \\
 & E_V & & & V' \\
 & \swarrow & & \searrow & \\
 & & V & &
 \end{array} \quad (10)$$

where $n_V \hookrightarrow E_V \hookrightarrow n$ is an elementary view, and $n_V \hookrightarrow V'$ is a view or $0 \hookrightarrow 0$. The corresponding diagram is furthermore unique up to canonical natural transformation.

As a notational convenience, we confuse $n \hookrightarrow V$ with the corresponding morphism $0 \rightarrow n$ in the bicategory \mathbb{E}_v of Definition 6, and say that V factors as

$$0 \xrightarrow{V'} n_V \xrightarrow{h_V} X_V \xrightarrow{M_V} n,$$

with V' a view, h_V an restriction, and hence $M_V \circ h_V$ an elementary view.

Lemma 8. *E is faithful.*

Proof: Consider two natural transformations $\alpha, \beta: S \rightarrow S'$ in \mathcal{S}_n , such that on a given view $h: n \hookrightarrow V$, there is an $s \in S(V)$ such that $\alpha_V(s) \neq \beta_V(s)$. If V is isomorphic to n , then w.l.o.g. we may assume it to be just n , and then $E(\alpha)$ and $E(\beta)$ differ by the first mapping $S(n \hookrightarrow n) \rightarrow S'(n \hookrightarrow n)$, at s .

Otherwise, by Lemma 7, V canonically factors as

$$0 \xrightarrow{V'} n_V \xrightarrow{h_V} X_V \xrightarrow{M_V} n.$$

Letting $s_0 = S(!)(s) \in S(n \hookrightarrow n)$, and $s'_0 = \alpha_{n \hookrightarrow n}(s_0)$, we then have

$$S(M_V)(S_{|s_0})(h_V, V') = S_{|s_0}(V)$$

by (6), and $s \in S_{|s_0}(V)$ by definition of s_0 , so

$$\begin{aligned}
 & E(\alpha)(s_0)(M_V)(h_V, V'): \\
 & S(M_V)(S_{|s_0})(h_V, V') \rightarrow S(M_V)(S'_{|s'_0})(h_V, V')
 \end{aligned}$$

sends s to $\alpha_V(s)$, thus differing from $E(\beta)(s_0)(M_V)(h_V, V')$, which sends s to $\beta_V(s)$. \square

Lemma 9. *E is full.*

Proof: Consider any morphism $E(S) \rightarrow E(S')$ of families. It amounts to a mapping $h: S(id_n) \rightarrow S'(id_n)$, and for each $s \in S(id_n)$ and $M \in \mathcal{M}_n$, to a morphism

$$k(s)(M): S(M)(S_{|s}) \rightarrow S(M)(S'_{|h(s)})$$

of strategies.

We define a natural transformation $\alpha: S \rightarrow S'$ such that $E(\alpha) = (h, k)$. It is enough to define it on finite views. On views V which are isomorphisms let α_V be defined as the composite:

$$S(V) \cong S(id_n) \xrightarrow{h} S'(id_n) \cong S'(V).$$

Now, for any non-isomorphism, finite view $V: n' \rightarrow n$, viewing $S(V) \xrightarrow{S!} S(id_n)$ as a set indexed over $S(id_n)$, we have

$$S(V) \cong \prod_{s \in S(id_n)} S_{|s}(V).$$

But by Lemma 7, such a V factors canonically as

$$X \xrightarrow{V'} n_V \xrightarrow{h_V} X_V \xrightarrow{M_V} n,$$

and by (6) we have $S(M_V)(S_{|s})(h_V, V') \cong S_{|s}(V)$. We may thus define α at V to be the composite

$$\begin{array}{ccc}
 S(V) & \overset{\alpha_V}{\dashrightarrow} & S'(V) \\
 \cong \downarrow & & \uparrow \cong \\
 \prod_{s \in S(id_n)} S_{|s}(V) & & \prod_{s' \in S'(id_n)} S'_{|s'}(V) \\
 \cong \downarrow & & \uparrow g \\
 \prod_s S(M_V)(S_{|s})(h_V, V') & \xrightarrow{f} & \prod_s S(M_V)(S'_{|h(s)})(h_V, V')
 \end{array}$$

where f is $\prod_s k(s)(M_V)(h_V, V')$, and g is

$$\begin{array}{ccc}
 \prod_s S(M_V)(S'_{|h(s)})(h_V, V') & & \\
 \downarrow [(h(s), x)]_s & & \\
 \prod_{s' \in S'(id_n)} S(M_V)(S'_{|s'})(h_V, V') & & \\
 \downarrow \cong & & \\
 \prod_{s' \in S'(id_n)} S'_{|s'}(V). & &
 \end{array}$$

That α is natural in V and sent by E to (h, k) is an easy verification, using naturality of the $k(s)(M)$'s. \square

Lemma 10. *E is essentially surjective.*

Proof: Consider any family $S: S_0 \rightarrow (\prod_{M \in \mathcal{M}_n} S_{X_M})_0$. Recall that for any M , an object of S_{X_M} is determined up to canonical isomorphism by its value on pairs of a morphism $n' \rightarrow X_M$ and a finite view V on n' . Define the sheaf $l(S)$ on \mathbb{E}_n to send any finite view V on n to:

- S_0 if V is an isomorphism;
- $\prod_{s \in S_0} S(s)(M_V)(h_V, V')$ if, recalling Lemma 7, V factors as

$$0 \xrightarrow{V'} n_V \xrightarrow{h_V} X_{M_V} \xrightarrow{M_V} n.$$

On morphisms $h: V \hookrightarrow W$ of views, which have to be inclusions, if V is an isomorphism and W is not, we define

the action $\mathbb{I}(S)(h)$ to be the projection

$$\coprod_{s \in S_0} S(s)(M_W)(h_W, W') \xrightarrow{\pi} S_0.$$

If otherwise W is also an isomorphism, then the action of h is the identity. Finally, if V is not an isomorphism, then neither is W , and V and W factor through the same M and h , and the inclusion factors as

$$\begin{array}{ccc} V' & \xleftarrow{h'} & W' \\ \uparrow & & \uparrow \\ n_V & \xlongequal{\quad} & n_W \\ h_V \downarrow & & \downarrow h_W \\ X_M & \xlongequal{\quad} & X_M \\ \downarrow & & \downarrow \\ M & \xlongequal{\quad} & M \\ \uparrow & & \uparrow \\ n & \xlongequal{\quad} & n. \end{array}$$

We then define the action $\mathbb{I}(S)(h)$ to be

$$\begin{array}{c} \coprod_{s \in S_0} S(s)(M)(h_W, W') \\ \downarrow \coprod_{s \in S_0} S(s)(M)(h') \\ \coprod_{s \in S_0} S(s)(M)(h_V, V'). \end{array}$$

Now, we show that $S \cong \mathbb{E}(\mathbb{I}(S))$. First, we have $(\mathbb{E}(\mathbb{I}(S)))(id_n) = S_0$, and, for $s \in S_0$,

- $(\mathbb{I}(S))|_s(V) = \{s\}$ if V is an isomorphism, and
- $(\mathbb{I}(S))|_s(V) = S(s)(M_V)(h_V, V')$ if V factors as

$$0 \xrightarrow{V'} n_V \xrightarrow{h_V} X_V \xrightarrow{M_V} n.$$

Thus, for any morphism $h: n' \hookrightarrow X_V$ (which has to be mono), and view $n' \hookrightarrow V'$, we have

$$\begin{aligned} & (\mathbb{E}(\mathbb{I}(S)))(s)(M_V)(h, V') \\ & \cong S(M_V)((\mathbb{I}(S))|_s)(h, V') \quad (\text{by definition}) \\ & \cong (\mathbb{I}(S))|_s(M_V \circ h \circ V') \quad (\text{by (6)}) \\ & \cong S(s)(M_V)(h, V') \quad (\text{by the last remark}). \quad \square \end{aligned}$$

APPENDIX B

MAXIMAL EXTENSIONS

This section is a proof of Theorem 2. We proceed by implementing a simple scheduler. Concretely, we construct a diagram $u: \omega \rightarrow \sigma/\int S$, and then take the colimit U' of the composite $\omega \xrightarrow{u} \int S \xrightarrow{p} (\mathbb{E})_X$, which is indeed an observation. The family $(p(u_i))_{i \in \omega}$ covers U' and the family $(u_i)_{i \in \omega}$ is compatible, so, S being a sheaf, it has a unique amalgamation $\sigma' \in S(U')$, which we show is S -maximal.

First, since U is finite, we may view it as a cospan $Y_0 \hookrightarrow U \hookrightarrow X$, where Y_0 is the smallest position containing all the elements of $\coprod_n U(n)$ not in the image of s , i.e., which are ready to play.

In order to define u , we actually define a bit more than the announced sequence: we define a diagram as in Fig. 3, where the $Y_i \hookrightarrow M_i \hookrightarrow Y_{i+1}$ are moves, together with

- a sequence of states $\sigma_i \in S(X \hookrightarrow U_i)$ such that $(\sigma_i)|_U = \sigma$, and
- a sequence of functions $f_i: \coprod_n Y_i(n) \rightarrow \mathbb{N}$ called the *queue*, constant on the intersection

$$\begin{array}{ccc} P_i & \xrightarrow{\quad} & \coprod_n Y_{i+1}(n) \\ \downarrow \lrcorner & & \downarrow \\ \coprod_n Y_i(n) & \xrightarrow{\quad} & \coprod_n M_i(n). \end{array} \quad (11)$$

The last condition will ensure that players waiting to play at step i which end up not playing (by a choice of the scheduler) keep their rank in the queue.

We consider the diagram as advancing from one

$$\begin{array}{ccc} V_i & \xrightarrow{\quad} & Y_{i+1} \\ \downarrow & & \\ U_i, & & \end{array} \quad (12)$$

to another (where the V_i 's are moves except if $i = -1$).

We fix the case $i = -1$ to be

$$\begin{array}{ccc} U & \xrightarrow{\quad} & Y_0 \\ \downarrow & & \\ U, & & \end{array}$$

with f_0 the constant 0 function.

Assuming the sequence to be defined up to step (12), consider the set \mathcal{M}_i^0 of all closed-world moves $Z \hookrightarrow M \hookrightarrow Y_{i+1}$ from Y_{i+1} . Let then \mathcal{M}_i^1 be the set of pairs (M, σ') with $M \in \mathcal{M}_i^0$ and $\sigma' \in S(M +_{Y_{i+1}} U_i)$ such that $(\sigma')|_{U_i} = \sigma_i$, where $M +_{Y_{i+1}} U_i$ is the pushout

$$\begin{array}{ccccc} M_i & \xrightarrow{\quad} & Y_{i+1} & \xrightarrow{\quad} & M \\ \downarrow & & & & \downarrow \\ U_i & \xrightarrow{\quad} & & \rightarrow & M +_{Y_{i+1}} U_i. \end{array}$$

If \mathcal{M}_i^1 is empty then σ_i is S -maximal and we are done.

Otherwise, among these pairs (M, σ') , choose one for which at least one of the active players in M is minimal for f_i , and call it (M_{i+1}, σ_{i+1}) . Formally, let $\text{active}(M)$ be the image of $M(s)$; let f'_i be the restriction of f_i to $\bigcup_{(M, \sigma') \in \mathcal{M}_i^1} \text{active}(M)$. The function f'_i admits a minimum, say m . Then choose (M_{i+1}, σ_{i+1}) in the pullback

$$\begin{array}{ccc} \mathcal{M}_i^2 & \xrightarrow{\quad} & \mathcal{M}_i^1 \\ \downarrow \lrcorner & & \downarrow \text{active} \\ \{H \mid H \cap ((f'_i)^{-1}\{m\}) \neq \emptyset\} & \xrightarrow{\quad} & \mathcal{P}(\coprod_n Y_{i+1}(n)). \end{array}$$

Putting $U_{i+1} = (M +_{Y_{i+1}} U_i)$ and $Y_{i+2} = Z$, we obtain a diagram

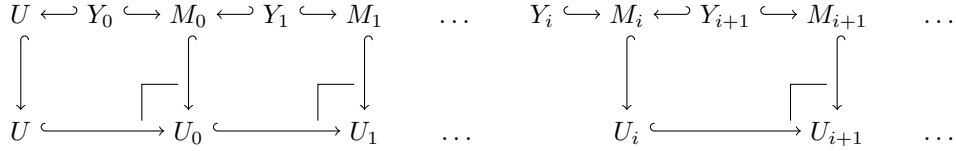
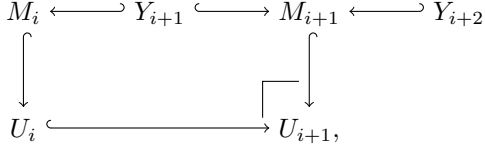


Fig. 3. Sequence of observations



and it only remains to define f_{i+1} . On the intersection P_i (11), we let $f_{i+1} = f_i$ as forced by the requirements on the sequence. Now, f_i has a finite domain, so it has a maximum m' . On $x \notin P_i$, we let $f_{i+1}(x) = m' + 1$.

This completes the description of our sequence. Let U' be the (directed) colimit of the U_i 's, and let $i: U \hookrightarrow U'$ be the colimiting morphism. Because $X \hookrightarrow U$ is a finite observation, U' is an observation, which is covered by the U_i 's. Hence, the σ_i 's form a compatible family and thus have an amalgamation in $S(X \hookrightarrow U')$, which we call σ' . It obviously satisfies $\sigma' \cdot i = \sigma$. We now prove by contradiction that σ' is S -maximal, in the case where U' is infinite (we have seen that it is otherwise). Let Y' be the smallest position containing all the elements of

$\coprod_n U'(n)$ not in the image of s , i.e., which are ready to play.

Assume a non-surjective $j: U' \rightarrow U''$ with some $\sigma'' \in S(U'')$ such that $\sigma'' \cdot j = \sigma'$. Let x be a minimal move in $U'' \setminus U'$, which has its active players in Y' . Because U' is the colimit of the U_i 's, there is an i such that Y_i already contains the set A of active players of x . Let $n_A = \min(f_i(A))$. Let, for any $k \geq i$, $\mu(k)$ be the cardinal of the pullback

$$\begin{array}{ccc}
\mathcal{M}_k^3 & \hookrightarrow & \mathcal{M}_k^1 \\
\downarrow & \lrcorner & \downarrow \text{active} \\
\{H \mid H \cap ((f_k')^{-1}\{0, \dots, n_A\}) \neq \emptyset\} & \hookrightarrow & \mathcal{P}(\coprod_n Y_{i+1}(n)),
\end{array}$$

which is finite. Because the elements of A could play, but never do in U' , by construction of the sequence, for all $k \geq i$, $\min(f_k(\text{active}(M_k))) \leq n_A$, so that $\mu(k) > \mu(k+1)$, contradicting the fact that the sequence is infinite.