

# Making concurrency functional

Glynn Winskel

Edinburgh Research Centre, Central Software Institute, Huawei;  
University of Strathclyde, Glasgow, UK

**Abstract**—The article bridges between two major paradigms in computation, the *functional*, at basis computation from input to output, and the *interactive*, where computation reacts to its environment while underway. Central to any compositional theory of interaction is the dichotomy between a system and its environment. Concurrent games and strategies address the dichotomy in fine detail, very locally, in a distributed fashion, through distinctions between Player moves (events of the system) and Opponent moves (those of the environment). A functional approach has to handle the dichotomy more ingeniously, via its blunter distinction between input and output. This has led to a variety of functional approaches, specialised to particular interactive demands. Through concurrent games we can see what separates and connects the differing paradigms, and show how:

- to lift functions to strategies; how to turn functional dependency to causal dependency and so exploit functional techniques.
- several approaches of functional programming and logic arise naturally as full subcategories of concurrent games, including stable domain theory; nondeterministic dataflow; geometry of interaction; the dialectica interpretation; lenses and optics, and their extensions to containers in dependent lenses and optics.
- the enrichments of strategies (e.g. to probabilistic, quantum or real-number computation) specialise to the functional cases.

## I. INTRODUCTION

The view of computation as functions is at the very foundation of computer science: the Church-Turing thesis expresses the coincidence of different notions of computable function; programming with higher-order functions is commonplace.

In contrast the view of computation as interaction is more recent and less settled, and often obscured by adherence to one syntax or another, perhaps each with its own mechanism of interaction. Instead our approach is maths-driven. Its tools are those of distributed/concurrent games and strategies [1], a causal model which allows for highly distributed interaction. Concurrent games and strategies are built on the mathematical foundations of categories of models for interaction [2], chiefly on the central model of event structures [3].<sup>1</sup>

Whereas the basic mechanism of interaction of functions is clear—ultimately by function composition—a functional approach can struggle with finding quite the right way to approach computation which isn't simply from input to output. The literature includes approaches via lenses, optics, combs, containers, dependent lenses, open games and learners [8]–[14]. The difficulties are compounded by enrichments to, say, probabilistic, quantum or real-number computation.

<sup>1</sup>A core language for concurrent strategies derives from the mathematical structure, although we shall only glimpse it here in Section IV-G: it is *higher-order* and an interesting hybrid of *dataflow*, c.f. TensorFlow [4], *concurrent process calculi*, c.f. CSP, CCS and *Session Types* [5]–[7].

By adopting a model which addresses interaction from the outset, we can better understand and explore the space of possible modes of interaction, functional or otherwise. Concurrent games and strategies provide a way to describe and orchestrate temporal patterns of interaction between functions, their fine-grained dependencies and dynamic linkage—Sections VI and VII. They support enrichments to strategies for probabilistic, quantum and real-number computation.

This article bridges between the two paradigms of computation, the functional and the interactive. Broadly, it shows:

- How to convert a general class of functions to concurrent strategies—Section V; this helps in describing and programming strategies by functional techniques.
- How in many cases we can describe concurrent strategies as interacting patterns of functions; it reveals many approaches in functional programming arise as full subcategories associated with special cases of concurrent games—Section VI.
- How concurrent strategies enrich in a general symmetric monoidal category  $\mathcal{M}$  and determine interacting patterns of “functions” (maps in  $\mathcal{M}$ ), composed through the composition of strategies; in this sense a subcategory of strategies determines a potential functional paradigm, helping systematise the way we explore interaction via functions—Section VII.

Amplifying the second point above, it was a surprise to the author how many functional approaches and paradigms arise simply by specialising to full subcategories of concurrent games. By restricting to deterministic strategies between concurrent games where all moves are Player moves we recover *stable functions* and Berry's stable domain theory, of which Girard's qualitative domains and coherence spaces are special cases. For such restricted games, general, nondeterministic, strategies correspond to *stable spans*, a model underpinning compositional accounts of nondeterministic dataflow. Only marginally more complicated are games comprising two parallel components, one a purely Player game and the other with purely Opponent moves. Strategies between such games yield models for *Geometry of Interaction* built on stable functions and stable spans [15]–[18]. Adjoining winning conditions and imperfect information to these games, so Opponent can see the moves of Player but not the converse, we recover a *dialectica category* [19], so Gödel's dialectica interpretation [20], from deterministic strategies. Dialectica categories mark an early occurrence of *lenses*, used in functional programming to make composable local changes on data-structures [8], [9]. The newer paradigm of *optics* appears in characterising arbitrary, nondeterministic, strategies between dialectica games, and when we move to more general container games, associated

with container types [10], [12], [21]. The definition of *dependent optic* here is derived as a characterisation of general strategies between container games; it seems to be new [22].

After the basics on event structures, the tools of stable families, and concurrent strategies, the new contribution comes in three parts, roughly: how to describe strategies by functions; how to describe functions and functional paradigms by strategies; and, how enriched strategies describe interacting patterns of functions. The first Section V, introduces a powerful method for lifting a very broad class of functions to strategies, turning functional into causal dependency. The second part, Section VI, concerns how causal dependency determines functional dependency, and shows how many discoveries in making functions interactive arise as subcategories of concurrent games. The third part, Section VII, outlines how strategies enrich in a symmetric monoidal category—this transfers to functional approaches. Enriched strategies orchestrate a dynamic pattern of interaction between “functions” understood as maps in the enriching, symmetric monoidal category. Further details and background, in particular on the dependent-type constructions on event structures can be found in the extended version [23].

## II. EVENT STRUCTURES

An *event structure* [3] comprises  $(E, \leq, \text{Con})$ , consisting of a set  $E$  of *events* which are partially ordered by  $\leq$ , the *causal dependency relation*, and a nonempty *consistency relation*  $\text{Con}$  consisting of finite subsets of  $E$ . The relation  $e' \leq e$  expresses that event  $e$  causally depends on the previous occurrence of event  $e'$ ; the consistency relation, those events which may occur together. We insist that the partial order is *finitary*, i.e.

- $[e] := \{e' \mid e' \leq e\}$  is finite for all  $e \in E$ ,

and that consistency satisfies

- $\{e\} \in \text{Con}$  for all  $e \in E$ ,
- $Y \subseteq X \in \text{Con}$  implies  $Y \in \text{Con}$ , and
- $X \in \text{Con}$  &  $e \leq e' \in X$  implies  $X \cup \{e'\} \in \text{Con}$ .

There is an accompanying notion of state or history. A *configuration* is a, possibly infinite, subset  $x \subseteq E$  which is:

- *consistent*,  $X \subseteq x$  &  $X$  is finite implies  $X \in \text{Con}$ ; and
- *down-closed*,  $e' \leq e \in x$  implies  $e' \in x$ .

Two events  $e, e'$  are called *concurrent* if the set  $\{e, e'\}$  is in  $\text{Con}$  and neither event is causally dependent on the other; then we write  $e \text{ co } e'$ . In games the relation of *immediate dependency*  $e \rightarrow e'$ , meaning  $e$  and  $e'$  are distinct with  $e \leq e'$  and no event in between, plays a very important role. We write  $[X]$  for the down-closure of a subset of events  $X$ . Write  $\mathcal{C}(E)$  for the configurations of  $E$  and  $\mathcal{C}(E)^\circ$  for its *finite configurations*.

Let  $E$  and  $E'$  be event structures. A *map* of event structures  $f : E \rightarrow E'$  is a partial function on events  $f : E \rightarrow E'$  such that for all  $x \in \mathcal{C}(E)$  its direct image  $fx \in \mathcal{C}(E')$  and

if  $e, e' \in x$  and  $f(e) = f(e')$  (with both defined), then  $e = e'$ .

Maps of event structures compose as partial functions. Notice that for a *total* map  $f$ , i.e. when the function  $f$  is total, the condition on maps says it is *locally injective*, in the sense that w.r.t. any configuration  $x$  of the domain the restriction of

$f$  to a function from  $x$  is injective; the restriction of total  $f$  to a function from  $x$  to  $fx$  is thus bijective.

Although a map  $f : E \rightarrow E'$  of event structures does not generally preserve causal dependency, it does reflect causal dependency locally: whenever  $e, e' \in x$ , a configuration of  $E$ , and  $f(e)$  and  $f(e')$  are both defined with  $f(e') \leq f(e)$ , then  $e' \leq e$ . Consequently,  $f$  preserves the concurrency relation: if  $e \text{ co } e'$  in  $E$  then  $f(e) \text{ co } f(e')$ , when defined.

A total map of event structures is *rigid* when it preserves causal dependency. Rigid maps induce discrete fibrations:

**Proposition 1.** *A total map  $f : E \rightarrow E'$  of event structures is rigid iff for all  $x \in \mathcal{C}(E)$  and  $y \in \mathcal{C}(E')$ ,*

$$y \subseteq fx \implies \exists z \in \mathcal{C}(E). z \subseteq x \text{ and } fz = y.$$

*The configuration  $z$  is necessarily unique by local injectivity.*

## III. STABLE FAMILIES

In an event structure, defined above, an event  $e$  has a unique causal history, the prime configuration  $[e]$ . Constructions directly on such event structures can be unwieldy, as often an event is more immediately associated with several mutually inconsistent causal histories. In this case the broader model of stable families is apt, especially so, as any stable family yields an event structure [3], [24].

A subset  $X$  of a family of sets  $\mathcal{F}$  is *compatible* if there is an element of  $\mathcal{F}$  which includes all elements of  $X$ ; we say  $X$  is *finitely compatible* if every finite subset of  $X$  is compatible. A *stable family* is a non-empty family of sets  $\mathcal{F}$  which is

- *Complete*: if  $Z \subseteq \mathcal{F}$  is finitely compatible,  $\bigcup Z \in \mathcal{F}$ ;
- *Stable*: If  $Z \subseteq \mathcal{F}$  is compatible and nonempty,  $\bigcap Z \in \mathcal{F}$ ;
- *Finitary*: If  $e \in x$  &  $x \in \mathcal{F}$ , there is  $x_0 \in \mathcal{F}$  with  $x_0$  finite,  $e \in x_0$  &  $x_0 \subseteq x$ ;
- *Coincidence-free*: For all  $x \in \mathcal{F}$ ,  $e, e' \in x$  with  $e \neq e'$ ,

$$\exists x_0 \in \mathcal{F}. x_0 \subseteq x \text{ & } (e \in x_0 \iff e' \notin x_0).$$

We call elements of  $\mathcal{F}$  its *configurations*,  $\bigcup \mathcal{F}$  its *events* and write  $\mathcal{F}^\circ$  for its *finite configurations*.

A *map*  $f : \mathcal{F} \rightarrow \mathcal{G}$  between stable families  $\mathcal{F}$  and  $\mathcal{G}$  is a partial function  $f$  from the events of  $\mathcal{F}$  to those of  $\mathcal{G}$  such that for all  $x \in \mathcal{F}$  its direct image  $fx \in \mathcal{G}$  and if  $e, e' \in x$  and  $f(e) = f(e')$  then  $e = e'$ . The choice of map ensures a full inclusion functor from the category of event structures to that of stable families. The inclusion functor has a right adjoint  $\text{Pr}$  giving a coreflection (an adjunction with unit an isomorphism). The construction  $\text{Pr}(\mathcal{F})$  essentially replaces the original events of a stable family  $\mathcal{F}$  by the minimal, prime configurations at which they occur. Let  $x$  be a configuration of a stable family  $\mathcal{F}$ . Define the *prime* configuration of  $e$  in  $x$  by

$$[e]_x := \bigcap \{y \in \mathcal{F} \mid e \in y \text{ & } y \subseteq x\}.$$

By coincidence-freeness, the function  $\text{top} : \mathcal{C}(\text{Pr}(\mathcal{F})) \rightarrow \mathcal{F}$  which takes a prime configuration  $[e]_x$  to  $e$  is well-defined; it is the counit of the adjunction [3], [24].

**Theorem 1.** Let  $\mathcal{F}$  be a stable family. Then,  $\text{Pr}(\mathcal{F}) := (P, \text{Con}, \leq)$  is an event structure where

$$\begin{aligned} P &= \{[e]_x \mid e \in x \ \& \ x \in \mathcal{F}\}, \\ Z \in \text{Con} &\text{ iff } Z \subseteq P \ \& \ \bigcup Z \in \mathcal{F}, \text{ and} \\ p \leq p' &\text{ iff } p, p' \in P \ \& \ p \subseteq p'. \end{aligned}$$

There is an order isomorphism  $\theta : (\mathcal{C}(\text{Pr}(\mathcal{F})), \subseteq) \cong (\mathcal{F}, \subseteq)$  where  $\theta(y) := \text{top } y = \bigcup y$  for  $y \in \mathcal{C}(\text{Pr}(\mathcal{F}))$ ; its mutual inverse is  $\varphi$  where  $\varphi(x) = \{[e]_x \mid e \in x\}$  for  $x \in \mathcal{F}$ .

The partial orders represented by configurations under inclusion are the same whether for event structures or stable families. They are Gérard Berry’s *dl-domains* [3], [24], [25].

#### A. Hiding—the defined part of a map

Let  $(E, \leq, \text{Con})$  be an event structure. Let  $V \subseteq E$  be a subset of ‘visible’ events. Define the *projection* on  $V$ , by  $E \downarrow V := (V, \leq_V, \text{Con}_V)$ , where  $v \leq_V v'$  iff  $v \leq v' \ \& \ v, v' \in V$  and  $X \in \text{Con}_V$  iff  $X \in \text{Con} \ \& \ X \subseteq V$ . The operation *hides* all events outside  $V$ . It is associated with a *partial-total factorization system*. Consider a partial map of event structures  $f : E \rightarrow E'$ . Let  $V := \{e \in E \mid f(e) \text{ is defined}\}$ . Then  $f$  clearly factors into the composition

$$E \xrightarrow{f_0} E \downarrow V \xrightarrow{f_1} E'$$

of  $f_0$ , a partial map of event structures taking  $e \in E$  to itself if  $e \in V$  and undefined otherwise, and  $f_1$ , a total map of event structures acting like  $f$  on  $V$ . Note that any  $x \in \mathcal{C}(E \downarrow V)$  is the image under  $f_0$  of a *minimum* configuration, viz.  $[x]_E \in \mathcal{C}(E)$ . We call  $f_0$  a *projection* and  $f_1$  the *defined part* of the map  $f$ .

#### B. Pullbacks

The coreflection from event structures to stable families is a considerable aid in constructing limits in the former from limits in the latter. The *pullback* of total maps of event structures is essential in composing strategies. We can define it via the pullback of stable families, obtained as a stable family of *secured bijections*. Let  $\sigma : S \rightarrow B$  and  $\tau : T \rightarrow B$  be total maps of event structures. There is a composite bijection

$$\psi : x \cong \sigma x = \tau y \cong y,$$

between  $x \in \mathcal{C}(S)$  and  $y \in \mathcal{C}(T)$  such that  $\sigma x = \tau y$ ; because  $\sigma$  and  $\tau$  are total they induce bijections between configurations and their image. The bijection is *secured* when the transitive relation generated on  $\psi$  by  $(s, t) \leq (s', t')$  if  $s \leq_S s'$  or  $t \leq_T t'$  is a finitary partial order.

**Theorem 2.** Let  $\sigma : S \rightarrow B$ ,  $\tau : T \rightarrow B$  be total maps of event structures. The family  $\mathcal{R}$  of secured bijections between  $x \in \mathcal{C}(S)$  and  $y \in \mathcal{C}(T)$  such that  $\sigma x = \tau y$  is a stable family. The functions  $\pi_1 : \text{Pr}(\mathcal{R}) \rightarrow S$ ,  $\pi_2 : \text{Pr}(\mathcal{R}) \rightarrow T$ , taking a secured bijection with *top to*, respectively, the left and right components of its *top*, are maps of event structures.  $\text{Pr}(\mathcal{R})$  with  $\pi_1, \pi_2$  is the pullback of  $\sigma, \tau$  in the category of event structures.

**Notation III-C.** W.r.t.  $\sigma : S \rightarrow B$  and  $\tau : T \rightarrow B$ , define  $x \wedge y$  to be the configuration of their pullback which corresponds via  $\theta : \mathcal{C}(\text{Pr}(\mathcal{R})) \cong \mathcal{R}$  to a secured bijection between  $x \in \mathcal{C}(S)$  and  $y \in \mathcal{C}(T)$ , necessarily with  $\sigma x = \tau y$ ; any configuration of the pullback takes the form  $x \wedge y$  for unique  $x$  and  $y$ .

### IV. CONCURRENT GAMES AND STRATEGIES

The driving idea is to replace the traditional role of game trees by that of event structures. Both games and strategies will be represented by *event structures with polarity*, which comprise  $(A, \text{pol}_A)$  where  $A$  is an event structure and a polarity function  $\text{pol}_A : A \rightarrow \{+, -, 0\}$  ascribing a polarity + (Player) or – (Opponent) or 0 (neutral) to its events. The events correspond to (occurrences of) moves. It will be technically useful to allow events of neutral polarity; they arise, for example, in a play between a strategy and a counterstrategy. Maps are those of event structures which preserve polarity. A *game* is represented by an event structure with polarities restricted to + or –, with no neutral events.

**Definition 3.** In an event structure with polarity, with configurations  $x$  and  $y$ , write  $x \subseteq^- y$  to mean inclusion in which all the intervening events  $y \setminus x$  are Opponent moves. Write  $x \subseteq^+ y$  for inclusion in which the intervening events are neutral or Player moves. For a subset of events  $X$  we write  $X^+$  and  $X^-$  for its restriction to Player and Opponent moves, respectively. The *Scott order* [26] will play a central role: between  $x, y \in \mathcal{C}(A)$ , where  $A$  is a game, it is defined by

$$y \subseteq_A x \iff y \supseteq^- x \cap y \subseteq^+ x$$

and is also characterised by

$$y \subseteq_A x \iff y^- \supseteq x^- \ \& \ y^+ \subseteq x^+.$$

The Scott order reduces to Scott’s pointwise order on functions in special cases and is central in relating games to Scott domains and “generalised domain theory” [26], [27].

There are two fundamentally important operations on games. One is that of forming the *dual game*. On a game  $A$  this amounts to reversing the polarities of events to produce the dual  $A^\perp$ . The other operation, a *simple parallel composition*  $A \parallel B$ , is achieved on games  $A$  and  $B$  by simply juxtaposing them, ensuring a finite subset of events is consistent if its overlaps with the two games are individually consistent; any configuration  $x$  of  $A \parallel B$  decomposes into  $x_A \parallel x_B$  where  $x_A$  and  $x_B$  are configurations of  $A$  and  $B$  respectively.

A *strategy* in a game  $A$  is a total map  $\sigma : S \rightarrow A$  of event structures with polarity such that

- (i) if  $\sigma x \subseteq^- y$ , for  $x \in \mathcal{C}(S), y \in \mathcal{C}(A)$ , there is a unique  $x' \in \mathcal{C}(S)$  with  $x \subseteq x'$  and  $\sigma x' = y$ ;
- (ii) if  $s \rightarrow_S s'$  and  $(\text{pol}(s) = + \text{ or } \text{pol}(s') = -)$ , then  $\sigma(s) \rightarrow_A \sigma(s')$ .

The conditions prevent Player from constraining Opponent’s behaviour beyond the constraints of the game. Condition (i) is *receptivity*, ensuring that the strategy is open to all moves of Opponent permitted by the game. Condition (ii), called *innocence* in [28], ensures that the only additional immediate causal dependencies a strategy can enforce beyond those of the game are those in which a Player move awaits moves of

Opponent. A map  $f : \sigma \Rightarrow \sigma'$  of strategies  $\sigma : S \rightarrow A$  and  $\sigma' : S' \rightarrow A$  is a map  $f : S \rightarrow S'$  such that  $\sigma = \sigma'f$ ; this determines when strategies are isomorphic.

Following [29], [30], a *strategy from a game  $A$  to a game  $B$*  is a strategy in the game  $A^\perp \parallel B$ . Given a strategy from  $B$  to a game  $C$ , so in  $B^\perp \parallel C$ , we compose the two strategies essentially by playing them against each other in the common game  $B$ , where if one strategy makes a Player move the other sees it as a move of Opponent. The conditions of receptivity and innocence precisely ensure that the copycat strategy behaves as identity w.r.t. composition, detailed below [1].

#### A. Copycat

Let  $A$  be a game. The copycat strategy  $\alpha_A : \mathbb{C}_A \rightarrow A^\perp \parallel A$  is an instance of a strategy from  $A$  to  $A$ . The event structure  $\mathbb{C}_A$  is based on the idea that Player moves in one component of the game  $A^\perp \parallel A$  always copy corresponding moves of Opponent in the other component. For  $c \in A^\perp \parallel A$  we use  $\bar{c}$  to mean the corresponding copy of  $c$ , of opposite polarity, in the alternative component. The event structure  $\mathbb{C}_A$  comprises  $A^\perp \parallel A$  with extra causal dependencies  $\bar{c} \leq c$  for all events  $c$  with  $pol_{A^\perp \parallel A}(c) = +$ ; with the original causal dependency they generate a partial order  $\leq$ ; a finite subset is consistent in  $\mathbb{C}_A$  iff its down-closure w.r.t.  $\leq$  is consistent in  $A^\perp \parallel A$ . The map  $\alpha_A$  acts as the identity function. In characterising the configurations of  $\mathbb{C}_A$  we recall the Scott order of Defn 3.

**Lemma 4.** *Let  $A$  be a game. Let  $x \in \mathcal{C}(A^\perp)$  and  $y \in \mathcal{C}(A)$ . Then,  $x \parallel y \in \mathcal{C}(\mathbb{C}_A)$  iff  $y \sqsubseteq_A x$ .*

#### B. Composition

Two strategies  $\sigma : S \rightarrow A^\perp \parallel B$  and  $\tau : T \rightarrow B^\perp \parallel C$  compose via pullback and hiding, summarised below.

$$\begin{array}{ccccc}
 & T \otimes S & \dashrightarrow & T \odot S & \\
 \pi_1 \swarrow & \downarrow & \searrow \pi_2 & \downarrow & \\
 S \parallel C & \tau \otimes \sigma & A \parallel T & \tau \odot \sigma & \\
 \sigma \parallel C \swarrow & \downarrow & \swarrow A \parallel \tau & \downarrow & \\
 & A \parallel B \parallel C & \longrightarrow & A \parallel C & 
 \end{array}$$

Ignoring polarities, by forming the pullback of  $\sigma \parallel C$  and  $A \parallel \tau$  we obtain the synchronisation of complementary moves of  $S$  and  $T$  over the common game  $B$ ; subject to the causal constraints of  $S$  and  $T$ , the effect is to instantiate the Opponent moves of  $T$  in  $B^\perp$  by the corresponding Player moves of  $S$  in  $B$ , and *vice versa*. Reinstating polarities we obtain the *interaction* of  $\sigma$  and  $\tau$

$$\tau \otimes \sigma : T \otimes S \rightarrow A^\perp \parallel B^0 \parallel C,$$

where we assign neutral polarities to all moves in or over  $B$ . Neutral moves over the common part  $B^0$  remain unhidden. The map  $A^\perp \parallel B^0 \parallel C \rightarrow A^\perp \parallel C$  is undefined on  $B^0$  and otherwise mimics the identity. Pre-composing this map with  $\tau \otimes \sigma$  we obtain a partial map  $T \otimes S \rightarrow A^\perp \parallel C$ ; it is undefined on precisely the neutral events of  $T \otimes S$ . The defined part of its partial-total factorization yields

$$\tau \odot \sigma : T \odot S \rightarrow A^\perp \parallel C$$

—this is the *composition* of  $\sigma$  and  $\tau$ .

**Notation IV-C.** For  $x \in \mathcal{C}(S)$  and  $y \in \mathcal{C}(T)$ , let  $\sigma x = x_A \parallel x_B$  and  $\tau y = y_B \parallel y_C$  where  $x_A \in \mathcal{C}(A)$ ,  $x_B, y_B \in \mathcal{C}(B)$ ,  $y_C \in \mathcal{C}(C)$ . Define  $y \otimes x = (x \parallel y_C) \wedge (x_A \parallel y)$ . This is a partial operation only defined if the  $\wedge$ -expression is. It is defined and glues configurations  $x$  and  $y$  together at their common overlap over  $B$  provided  $x_B = y_B$  and a finitary partial order of causal dependency results. Any configuration of  $T \otimes S$  has the form  $y \otimes x$ , for unique  $x \in \mathcal{C}(S), y \in \mathcal{C}(T)$ .

#### D. A bicategory of strategies

We obtain a bicategory **Strat** for which the objects are games, the arrows  $\sigma : A \dashrightarrow B$  are strategies  $\sigma : S \rightarrow A^\perp \parallel B$ ; with 2-cells  $f : \sigma \Rightarrow \sigma'$  maps of strategies. The vertical composition of 2-cells is the usual composition of maps. Horizontal composition is the composition of strategies  $\odot$  (which extends to a functor via the universality of pullback and partial-total factorisation). We can restrict the 2-cells to be rigid maps and still obtain a bicategory. The bicategory of strategies is compact-closed, though with the addition of winning conditions—Section IV-E—this weakens to  $*$ -autonomous.

A strategy  $\sigma : S \rightarrow A$  is *deterministic* if  $S$  is deterministic, *viz.*

$$\forall X \subseteq_{\text{fin}} S. [X]^- \in \text{Con}_S \implies X \in \text{Con}_S,$$

where  $[X]^- := \{s' \in S \mid \exists s \in X. pol_S(s') = - \ \& \ s' \leq s\}$ . So, a strategy is deterministic if consistent behaviour of Opponent is answered by consistent behaviour of Player. Copycat  $\alpha_A$  is deterministic iff the game  $A$  is *race-free*, *i.e.* if  $x \sqsubseteq^- y$  and  $x \sqsubseteq^+ z$  in  $\mathcal{C}(A)$  then  $y \cup z \in \mathcal{C}(A)$ . The bicategory of strategies restricts to a bicategory of deterministic strategies between race-free games [31], [32].

There are several ways to reformulate strategies. Deterministic strategies coincide with the *receptive* ingenuous strategies of Mellies and Mimram based on asynchronous transition systems [31], [33]. Via the Scott order, we can see strategies as a refinement of profunctors: a strategy in a game  $A$  induces a discrete fibration, so presheaf, on  $(\mathcal{C}(A)^\circ, \sqsubseteq_A)$ , a construction which extends to strategies as profunctors between games [26].

#### E. Winning conditions

*Winning conditions* of a game  $A$  specify a subset  $W$  of its configurations, an outcome in which is a win for Player. Informally, a strategy (for Player) is *winning* if it always prescribes moves for Player to end up in a winning configuration, no matter what the activity or inactivity of Opponent.

Formally, a game with winning conditions  $(A, W_A)$  comprises a concurrent game  $A$  with winning conditions  $W_A \subseteq \mathcal{C}(A)$ . A strategy  $\sigma : S \rightarrow A$  is *winning* if  $\sigma x$  is in  $W_A$  for all  $+$ -maximal configurations  $x$  of  $S$ ; in general, a configuration is  $+$ -maximal if no additional Player, or neutral, moves can occur from it. That  $\sigma$  is winning can be shown equivalent to: all plays of  $\sigma$  against any counterstrategy of Opponent result in a win for Player [32], [34].

As the dual of a game with winning conditions  $(A, W_A)$  we again reverse the roles of Player and Opponent, and take its winning conditions to be the set-complement of  $W_A$ , i.e.  $(A, W_A)^\perp = (A^\perp, \mathcal{C}(A) \setminus W_A)$ .

In a parallel composition of games with winning conditions, we deem a configuration  $x$  of  $A \parallel B$  winning if its component  $x_A$  is winning in  $A$  or its component  $x_B$  is winning in  $B$ :  $(A, W_A) \parallel (B, W_B) := (A \parallel B, W)$  where  $W = \{x \in \mathcal{C}(A \parallel B) \mid x_A \in W_A \text{ or } x_B \in W_B\}$ .

With these extensions, we take a winning strategy from a game  $(A, W_A)$  to a game  $(B, W_B)$ , to be a winning strategy in the game  $A^\perp \parallel B$ —its winning conditions form the set

$$\{x \in \mathcal{C}(A^\perp \parallel B) \mid x_A \in W_A \Rightarrow x_B \in W_B\}.$$

When games are race-free, copycat will be a winning strategy. The composition of winning strategies is winning [32], [34].

### F. Imperfect information

In a game of *imperfect information* some moves are inaccessible and strategies with dependencies on inaccessible moves are ruled out. Games extend with imperfect information in a way that respects the operations of concurrent games and strategies [35]. Moves of a game are assigned a level in an order of access levels; moves of the game or its strategies can only causally depend on moves at equal or lower levels.

In more detail, presupposing a fixed preorder of *access levels*  $(\Lambda, \preceq)$ , a  $\Lambda$ -game comprises a game  $A$  with a *level function*  $l : A \rightarrow \Lambda$  such that if  $a \leq_A a'$  then  $l(a) \preceq l(a')$ . A  $\Lambda$ -strategy in the  $\Lambda$ -game is a strategy  $\sigma : S \rightarrow A$  for which if  $s \leq_S s'$  then  $l\sigma(s) \preceq l\sigma(s')$  for all  $s, s'$  in  $S$ . The access levels of moves in a game are left undisturbed in forming the dual and parallel composition of  $\Lambda$ -games. As before, a  $\Lambda$ -strategy from a  $\Lambda$ -game  $A$  to a  $\Lambda$ -game  $B$  is a  $\Lambda$ -strategy in the game  $A^\perp \parallel B$ . It can be shown that  $\Lambda$ -strategies compose [35].

### G. A language for strategies

We recall briefly the language for strategies introduced in [36]. Games  $A, B, C, \dots$  play the role of types. Operations on games include dual  $A^\perp$ , simple parallel composition  $A \parallel B$ , a sum  $\sum_{i \in I} A_i$ , event-prefixing and recursive games.

Terms, denoting strategies, have typing judgements

$$x_1 : A_1, \dots, x_m : A_m \vdash t \dashv y_1 : B_1, \dots, y_n : B_n,$$

where all the variables are distinct, interpreted as a strategy from  $\vec{A} = A_1 \parallel \dots \parallel A_m$  to  $\vec{B} = B_1 \parallel \dots \parallel B_n$ . We can think of the term  $t$  as a box with input and output wires  $\vec{x}$  and  $\vec{y}$ .

The term  $t$  denotes a strategy  $\sigma : S \rightarrow \vec{A}^\perp \parallel \vec{B}$ . It does so by describing witnesses, configurations of  $S$ , to a relation between configurations  $\vec{x}$  of  $\vec{A}$  and  $\vec{y}$  of  $\vec{B}$ . For example, the term

$$x : A \vdash y \sqsubseteq_A x \dashv y : A$$

denotes the copycat strategy on a game  $A$ ; it describes configurations of copycat,  $\mathbb{C}_A$ , as witnesses, viz. those configurations  $x \parallel y$  of  $\mathbb{C}_A$  for which  $y \sqsubseteq_A x$  in the Scott order. There are

other operations, such as sum  $\sqcup$  and pullback  $\wedge$  on strategies of the same type. Duality is caught by the rules

$$\frac{\Gamma, x : A \vdash t \dashv \Delta}{\Gamma \vdash t \dashv x : A^\perp, \Delta} \quad \frac{\Gamma \vdash t \dashv x : A, \Delta}{\Gamma, x : A^\perp \vdash t \dashv \Delta}$$

and composition of strategies by

$$\frac{\Gamma \vdash t \dashv \Delta \quad \Delta \vdash u \dashv \text{H}}{\Gamma \vdash \exists \Delta. [t \parallel u] \dashv \text{H}}$$

which, in the picture of strategies as boxes, joins the output wires of one strategy to input wires of the other. Simple parallel composition of strategies arises when  $\Delta$  is empty.

## V. FROM FUNCTIONS TO STRATEGIES

The language for strategies in [36] included a judgement

$$x : A \vdash g(y) \sqsubseteq_C f(x) \dashv y : B$$

for building strategies out of expressions  $f(x)$  and  $g(y)$  denoting “affine functions.” It breaks down into a composition

$$x : A \vdash \exists z : C. [g(y) \sqsubseteq_C z \parallel z \sqsubseteq_C f(x)] \dashv y : B.$$

Here we present a considerably broader class of affine-stable functions  $f$  and “co-affine-stable” functions  $g$  with which to define strategies in this manner. The work hinges on the Scott order to convert functional dependency to causal dependency, in the sense captured by Theorem 6 below.

### A. Affine-stable maps and their strategies

**Definition 5.** An *affine-stable map* between games from  $A$  to  $B$  is a function  $f : \mathcal{C}(A) \rightarrow \mathcal{C}(B)$  which is

- *polarity-respecting*: for  $x, y \in \mathcal{C}(A)$ ,

$$x \sqsubseteq^- y \Rightarrow f(x) \sqsubseteq^- f(y) \quad \text{and} \quad x \sqsubseteq^+ y \Rightarrow f(x) \sqsubseteq^+ f(y);$$

- *+continuous*: if  $x \in \mathcal{C}(A)$ ,  $b \in f(x)$  &  $\text{pol}_B(b) = +$ ,

$$\exists x_0 \in \mathcal{C}(A)^\circ. x_0 \sqsubseteq x \ \& \ b \in f(x_0);$$

- *--image finite*: for all finite configurations  $x \in \mathcal{C}(A)^\circ$  the set  $f(x)^\circ$  is finite;

- *affine*: for all compatible families  $\{x_i \mid i \in I\}$  in  $\mathcal{C}(A)$ ,

$$\bigcup_{i \in I} f(x_i) \sqsubseteq^+ f(\bigcup_{i \in I} x_i)$$

—when  $I$  is empty this amounts to  $\emptyset \sqsubseteq^+ f(\emptyset)$ ; and

- *stable*: for all compatible families  $\{x_i \mid i \in I\} \neq \emptyset$  in  $\mathcal{C}(A)$ ,

$$f(\bigcap_{i \in I} x_i) \sqsubseteq^- \bigcap_{i \in I} f(x_i).$$

When all the moves of games  $A$  and  $B$  are those of Player, the definition reduces to that of stable function. If all moves are those of Opponent, it becomes that of demand maps—see Section VI-B [37]. Affine-stable maps include maps of event structures with polarity, including partial maps between games, and the affine maps of [36]. They are the most general maps out of which we can construct a strategy, now described.

**Theorem 6.** Let  $f : \mathcal{C}(A) \rightarrow \mathcal{C}(B)$  be an affine-stable map between games  $A$  and  $B$ . Then

$$\mathcal{F} := \{x \parallel y \in \mathcal{C}(A^\perp \parallel B) \mid y \sqsubseteq_B f(x)\}$$

is a stable family. The map  $\text{top} : \text{Pr}(\mathcal{F}) \rightarrow A^\perp \parallel B$  is a strategy  $f_! : A \dashrightarrow B$ . The strategy  $f_!$  is deterministic if  $A$  and  $B$  are race-free and  $f$  reflects  $-$ -compatibility, i.e.  $x \subseteq^- x_1$  and  $x \subseteq^- x_2$  in  $\mathcal{C}(A)$  and  $f x_1 \cup f x_2 \in \mathcal{C}(B)$  implies  $x_1 \cup x_2 \in \mathcal{C}(A)$ .

The theorem explains how to convert functional dependency, expressed as  $y \sqsubseteq_B f(x)$ , to causal dependency between moves  $\text{Pr}(\mathcal{F})$ , obtained as primes of the stable family  $\mathcal{F}$ .

For  $f$  an affine-stable map from  $A$  to  $B$  we can write  $f_!$  as

$$x : A \vdash y \sqsubseteq_B f(x) \dashv y : B.$$

Through suitable choices of  $f$  we can create strategies from structural maps, make injections and projections into strategies, create strategies for conditional and case statements and, in general, express much of the causal wiring that is often explained informally in a diagrammatic way. Generally, if  $\sigma$  is a strategy in  $A$  then  $f_! \circ \sigma$  is its “pushforward” to a strategy in  $B$ . Some other basic examples:

**Example 7. (Projectors)** Let  $f : A \parallel B \rightarrow B$  be the map undefined on game  $A$  but acting as identity on game  $B$ ; on configurations  $f x \parallel y = y$ . Let  $\sigma$  be a strategy in the game  $A \parallel B$ . Strategy  $f_! \circ \sigma$  is its projection to a strategy in  $B$ .  $\square$

**Example 8. (Duplicators)** Let  $A$  be a game. Consider the function  $d_A : x \mapsto x \parallel x$  from  $\mathcal{C}(A)$  to  $\mathcal{C}(A \parallel A)$ . It is easily checked to be affine-stable. Hence there is a *duplicator* strategy  $\delta_A = d_{A!} : A \dashrightarrow A \parallel A$ . (The strategy  $\delta_A$  is not natural in  $A$  as  $\parallel$  is not a product, except in subcategories.)  $\square$

**Example 9. (Detectors)** Let  $A$  be a game. Let  $X \in \text{Con}_A$  with  $X \subseteq A^+$ . Let  $\boxplus$  be a single “detector” event, of +ve polarity. Let  $d_X : \mathcal{C}(A) \rightarrow \mathcal{C}(\boxplus)$  be the function such that  $d_X(x) = \{\boxplus\}$  if  $X \subseteq x, \emptyset$  otherwise. The function  $d_X$  is affine-stable. There is a *detector* strategy  $d_{X!} : A \dashrightarrow \boxplus$ . The strategy simply adjoins extra causal dependencies  $a \rightarrow \boxplus$  from  $a \in X$ . It detects the presence of  $X$ . Similarly, one can extend detectors to detect the occurrence of one of a family  $\langle X_i \rangle_{i \in I}$  of  $X_i \in \text{Con}_A$  provided  $X_i \cup X_j \in \text{Con}_A$  implies  $i = j$ .  $\square$

**Example 10. (Blockers)** Let  $A$  be a game and  $Y \subseteq A^-$ . Let  $h_Y : \mathcal{C}(A) \rightarrow \mathcal{C}(\boxminus)$  be the function which acts so  $h_Y(x) = \{\boxminus\}$  if  $x \cap Y \neq \emptyset, \emptyset$  otherwise. Then  $h_Y$  is affine-stable. The *blocker* strategy  $h_{Y!}$  adjoins causal dependencies  $\boxminus \rightarrow a$  from  $\boxminus$  to each  $a \in Y$ . Absence of move  $\boxminus$  blocks all moves  $Y$ .  $\square$

**Theorem 11.** The operation  $(\_)!$  is a (pseudo)functor from the category of affine-stable maps to concurrent strategies **Strat**.

### B. co-Affine-stability

We examine the dual, or co-notion, to affine-stability. An affine-stable map  $f$  from  $A^\perp$  to  $B^\perp$  yields a strategy  $f_! : A^\perp \dashrightarrow B^\perp$ , so by duality a strategy  $f^* : B \dashrightarrow A$ . We obtain a dual to Theorems 6 and 11 as a corollary:

**Corollary 12.** Let  $g : \mathcal{C}(A) \rightarrow \mathcal{C}(B)$  be such that  $g : \mathcal{C}(A^\perp) \rightarrow \mathcal{C}(B^\perp)$  is affine-stable, then

$$\mathcal{G} := \{y \parallel x \in \mathcal{C}(B^\perp \parallel A) \mid g(x) \sqsubseteq_B y\}$$

is a stable family. The map  $\text{top} : \text{Pr}(\mathcal{G}) \rightarrow B^\perp \parallel A$  is a strategy  $g^* : B \dashrightarrow A$ . The strategy  $g^*$  is deterministic if  $A$  is race-free and  $g$  reflects  $+$ -compatibility. The operation  $(\_)^*$  is a contravariant (pseudo)functor from the category of affine-stable maps to **Strat**.

For  $g$  an affine-stable map  $A^\perp$  to  $B^\perp$  we can write  $g^*$  as

$$y : B \vdash g(y) \sqsubseteq_B x \dashv x : A.$$

For a strategy  $\sigma$  in game  $B$  the operation  $g^* \circ \sigma$  yields the strategy in  $A$  got as the pullback of  $\sigma$  along  $g$ . In particular, if  $A$  prefixed game  $B$  by some initial move,  $g^* \circ \sigma$  would be a prefix operation on strategies.

### C. An adjunction

An affine-stable map  $f$  from  $A$  to  $B$  is not generally an affine-stable map from  $A^\perp$  to  $B^\perp$ . The next definition, of an *additive-stable* map  $f$  from  $A$  to  $B$ , bluntens affine-stability to ensure  $f$  is also a additive-stable map from  $A^\perp$  to  $B^\perp$ ; hence is associated with both a strategy  $f_! : A \dashrightarrow B$  and a converse strategy  $f^* : B \dashrightarrow A$ . Together they form an adjunction.

**Definition 13.** A *additive-stable* map between event structures with polarity,  $A$  to  $B$ , is a function  $f : \mathcal{C}(A) \rightarrow \mathcal{C}(B)$  which is

- *polarity-respecting*: for  $x, y \in \mathcal{C}(A)$ ,

$$x \subseteq^- y \Rightarrow f(x) \subseteq^- f(y) \quad \text{and} \quad x \subseteq^+ y \Rightarrow f(x) \subseteq^+ f(y);$$

- *image finite*: if  $x \in \mathcal{C}(A)^\circ$  then  $f(x) \in \mathcal{C}(B)^\circ$ ;
- *additive*: for all compatible families  $\{x_i \mid i \in I\}$  in  $\mathcal{C}(A)$ ,

$$\bigcup_{i \in I} f(x_i) = f(\bigcup_{i \in I} x_i);$$

- *stable*: for all compatible families  $\{x_i \mid i \in I\} \neq \emptyset$  in  $\mathcal{C}(A)$ ,

$$f(\bigcap_{i \in I} x_i) = \bigcap_{i \in I} f(x_i).$$

The usual maps of games are additive-stable, including those which are partial, as are Girard’s linear maps. Additive-stability is indifferent to a switch of polarities:

**Proposition 2.** An additive-stable map  $f$  from  $A$  to  $B$  is an additive-stable map  $f$  from  $A^\perp$  to  $B^\perp$  and vice versa.

Given an additive-stable map  $f$  from  $A$  to  $B$  we obtain a strategy  $f_! : A \dashrightarrow B$  and, via  $f$  from  $A^\perp$  to  $B^\perp$ ,  $f^* : B \dashrightarrow A$ .

**Theorem 14.** Let  $f$  be an additive-stable map from  $A$  to  $B$  between event structures with polarity. The strategies  $f_!$  and  $f^*$  form an adjunction  $f_! \dashv f^*$  in the bicategory **Strat**.

This says **Strat** forms a *pseudo double category* [38], [39]. In Section VI-C1 we exemplify a use of Theorem 14 to relate different semantics, there by connecting deterministic strategies in general to those of Geometry of Interaction. The adjunction in **Strat** yields a traditional adjunction:

**Corollary 15.** Let  $f$  be an additive-stable map from game  $A$  to game  $B$ . Let  $\text{Strat}_A$  be the category of strategies in the game  $A$ , and  $\text{Strat}_B$  that of strategies in  $B$ . Then there are functors  $f_! \circ (\_) : \text{Strat}_A \rightarrow \text{Strat}_B$  and  $f^* \circ (\_) : \text{Strat}_B \rightarrow \text{Strat}_A$  with  $f_! \circ (\_) left adjoint to  $f^* \circ (\_)$ .$

## VI. FROM STRATEGIES TO FUNCTIONS

We recover familiar notions of games from those based on event structures. A game is *tree-like* when any two events are either inconsistent or causally dependent. When such a game is race-free, at any finite configuration, the next possible moves, if there are any, belong purely to Player, or purely to Opponent. Then, at each position where Player may move, a deterministic strategy either chooses a unique move or to stay put. In contrast to many presentations of games, in a concurrent strategy Player isn't forced to make a move, though that can be encouraged through suitable winning conditions. A counterstrategy, as a strategy in the dual game, picks moves for Opponent at their configurations. The interaction  $\tau \circledast \sigma$  of a deterministic strategy  $\sigma$  with a deterministic counterstrategy  $\tau$  determines a finite or infinite branch in the tree of configurations, which in the presence of winning conditions will be a win for one of the players.

On tree-like games we recover familiar notions. More surprising is that by exploiting the richer structure of concurrent games we can recover other familiar approaches, not traditionally tied to games, or if so only somewhat informally. We start by rediscovering Berry's stable domain theory, of which Jean-Yves Girard's qualitative domains and coherence spaces are special cases. The other examples, from dataflow, logic and functional programming, concern ways of handling interaction within a functional approach. We shall restrict to race-free games, so guaranteeing that deterministic strategies have an identity w.r.t. composition, given by copycat.

## A. Stable functions

Consider games in which all moves are Player moves. Consider a strategy  $\sigma$  from one such purely Player game  $A$  to another  $B$ . This is a map  $\sigma : S \rightarrow A^\perp \parallel B$  which is receptive and innocent. Notice that in  $A^\perp \parallel B$  all the Opponent moves are in  $A^\perp$  and all the Player moves are in  $B$ . By receptivity any configuration of  $A$  can be input. The only new immediate causal connections, beyond those in  $A^\perp$  and  $B$ , that can be introduced in a strategy are those from Opponent moves of  $A^\perp$  to Player moves of  $B$ . Beyond the causal dependencies of the games, a strategy  $\sigma$  can only make a Player move of  $B$  causally depend on a finite subset of moves of  $A^\perp$ .

When  $\sigma$  is deterministic, all inconsistencies are inherited from those between Opponent moves. Then the strategy  $\sigma$  gives rise to a stable function from the configurations of  $A$  to the configurations of  $B$ . Conversely, such a stable function  $f$  yields a deterministic strategy  $f_! : A \multimap B$ , by Theorem 6.

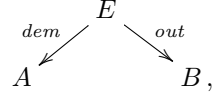
**Theorem 16.** *The category  $\mathbf{dI}$  of dI-domains and stable functions, enriched by the stable order, is equivalent to the bicategory of deterministic strategies between purely Player games with rigid 2-cells. (With all 2-cells it is equivalent to the category of dI-domains enriched by the pointwise order.)*

The category of dI-domains and stable functions is well-known to be cartesian-closed; its function space and product are realised by constructions  $[A \rightarrow B]$  and  $A \parallel B$  on event structures. When the games are further restricted to have trivial

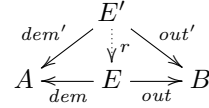
causal dependency we recover Girard's *qualitative domains*. Girard's models for polymorphism there generalise to dI-domains, with dependent types  $\prod_{x:A} B(x)$  and  $\Sigma_{x:A} B(x)$  on event structures [3], [40], [41]—see the extended version [23].

## B. Stable spans

When between games in which all the moves are Player moves, a general, nondeterministic, strategy corresponds to a *stable span*, a form of many-valued stable function which has been discovered, and rediscovered, in giving semantics to higher-order processes and especially nondeterministic dataflow [37], [42], [43]; the trace of strategies, derived from their compact closure, specialises to the feedback operation of dataflow. Recall a *stable span* comprises



with event structure  $E$  relating input given by an event structure  $A$  and output by an event structure  $B$ . The map  $\text{out} : E \rightarrow B$  is a *rigid* map. The map  $\text{dem} : E \rightarrow A$ , associated to input, is of a different character. It is a *demand* map, i.e., a function from  $\mathcal{C}(E)$  to  $\mathcal{C}(A)$  which preserves unions and finite configurations;  $\text{dem}(x)$  is the minimum input for  $x$  to occur and is the union of the demands of its events. The occurrence of an event  $e$  in  $E$  demands minimum input  $\text{dem}([e])$  and is observed as the output event  $\text{out}(e)$ . Spans from  $A$  to  $B$  are related by the usual 2-cells, here (necessarily) rigid maps  $r$  making the diagram below commute:



Stable spans compose via the usual pullback construction of spans, as both demand and output maps extend to functions between configurations. A stable span  $E$  corresponds to a (special) profunctor

$$\tilde{E}(x, y) = \{w \in \mathcal{C}(E)^\circ \mid \text{dem}(w) \subseteq x \ \& \ \text{out } w = y\},$$

between the partial-order categories  $\mathcal{C}(A)^\circ$  and  $\mathcal{C}(B)^\circ$  — a correspondence that respects composition. Recalling the view of profunctors as Kleisli maps w.r.t. the presheaf construction [44], we borrow from Moggi [45] and describe the composition of stable spans  $F : A \multimap B$ ,  $G : B \multimap C$  as

$$G \circ F(x) = \text{let } y \Leftarrow F(x) \text{ in } G(y)$$

—which, via the correspondence with profunctors, stands for the coend  $\int^{y \in \mathcal{C}(B)^\circ} \tilde{F}(x, y) \times \tilde{G}(y, \_)$ . In using let-notation we can take account of the shape of the configuration  $y$  in the definition of  $G$ , in effect an informal pattern matching.

Stable spans are monoidal closed [42], [46]: w.r.t. an event structure  $A$ , the functor  $(\_ \parallel A)$  has a right adjoint, the function space  $[A \multimap \_]$ . The construction  $[A \multimap B]$  extends to a dependent product  $\prod_{x:A}^s B(x)$  for stable spans: the type of stable spans which on input  $x : A$  yield output  $y : B(x)$

nondeterministically [23]. Stable spans are trace monoidal closed; their trace is described concretely in [43].

Let  $A$  and  $B$  be purely Player games. A strategy  $\sigma : S \rightarrow A^\perp \parallel B$  gives rise to a stable span

$$A \xleftarrow{dem} E \xrightarrow{out} B,$$

where  $E = S^+$ , and *out* gives the image of its events in  $B$  and *dem* those events in  $A$  on which they casually depend. Conversely given a stable span, as above, we obtain a strategy as the composition  $out_! \odot dem^*$ , by the results of Section V; as, regarding  $E$  and  $A$  as purely Player games, both *out* and *dem* :  $\mathcal{C}(E^\perp) \rightarrow \mathcal{C}(A^\perp)$  are affine-stable. In the strategy  $out_! \odot dem^* : S \rightarrow A^\perp \parallel B$  so obtained,  $S$  comprises the disjoint union of  $A$  and  $E$  with the additional causal dependencies of  $e \in E$  on  $a \in A^\perp$  prescribed by *dem*.

**Theorem 17.** *The bicategory **Stab** of stable spans is equivalent to the bicategory of strategies between purely Player games with rigid 2-cells.*

We show that in a similar way, we obtain geometry of interaction, dialectica categories, containers, lenses, open games and learners, optics and dependent optics by moving to slightly more complicated subcategories of games, sometimes with winning conditions and imperfect information.

### C. Geometry of Interaction

Let's now consider slightly more complex games. A *GoI game* comprises a parallel composition  $A := A_1 \parallel A_2$  of a purely Player game  $A_1$  with a purely Opponent game  $A_2$ . Consider a strategy  $\sigma$  from a GoI game  $A := A_1 \parallel A_2$  to a GoI game  $B := B_1 \parallel B_2$ . Rearranging the parallel compositions,

$$A^\perp \parallel B = A_1^\perp \parallel A_2^\perp \parallel B_1 \parallel B_2 \cong (A_1 \parallel B_2^\perp)^\perp \parallel (A_2^\perp \parallel B_1).$$

So  $\sigma$ , as a strategy in  $A^\perp \parallel B$ , corresponds to a strategy from the purely Player game  $A_1 \parallel B_2^\perp$  to the purely Player game  $A_2^\perp \parallel B_1$ . We are back to the simple situation considered in the previous section, of strategies between purely Player games.

Strategies between GoI games, from  $A$  to  $B$ , correspond to stable spans from  $A_1 \parallel B_2^\perp$  to  $A_2^\perp \parallel B_1$ . The maps are familiar from models of geometry of interaction built as free compact-closed categories from traced monoidal categories [17], [18], though here lifted to the *bicategory **Stab*** of stable spans.

**Theorem 18.** *The bicategory of strategies on GoI games with rigid 2-cells is equivalent to the free compact-closed bicategory built on the trace monoidal bicategory **Stab**.*

When deterministic, strategies from GoI game  $A$  to GoI game  $B$  correspond to a stable function from  $\mathcal{C}(A_1 \parallel B_2^\perp)$  to  $\mathcal{C}(A_2^\perp \parallel B_1)$ . Note that a configuration of a parallel composition of games splits into a pair of configurations:

$$\mathcal{C}(A_1 \parallel B_2^\perp) \cong \mathcal{C}(A_1) \times \mathcal{C}(B_2), \quad \mathcal{C}(A_2^\perp \parallel B_1) \cong \mathcal{C}(A_2) \times \mathcal{C}(B_1)$$

Thus deterministic strategies from  $A$  to  $B$  correspond to stable functions

$$S = \langle g, f \rangle : \mathcal{C}(A_1) \times \mathcal{C}(B_2) \rightarrow \mathcal{C}(A_2) \times \mathcal{C}(B_1),$$

associated with a pair of stable functions  $g : \mathcal{C}(A_1) \times \mathcal{C}(B_2) \rightarrow \mathcal{C}(A_2)$  and  $f : \mathcal{C}(A_1) \times \mathcal{C}(B_2) \rightarrow \mathcal{C}(B_1)$ , summarised diagrammatically by:

$$\begin{array}{ccc} A_1 & f & B_1 \\ & \curvearrowright S \curvearrowleft & \\ & & \\ & g & B_2 \end{array}$$

Such maps are obtained by Abramsky and Jagadeesan's GoI construction, here starting from *stable domain theory* [16].

The composition of deterministic strategies between GoI games,  $\sigma$  from  $A$  to  $B$  and  $\tau$  from  $B$  to  $C$  coincides with the composition of GoI given by "tracing out"  $B_1$  and  $B_2$ . Precisely, supposing  $\sigma$  corresponds to the stable function

$$S : \mathcal{C}(A_1) \times \mathcal{C}(B_2) \rightarrow \mathcal{C}(A_2) \times \mathcal{C}(B_1)$$

and  $\tau$  to the stable function

$$T : \mathcal{C}(B_1) \times \mathcal{C}(C_2) \rightarrow \mathcal{C}(B_2) \times \mathcal{C}(C_1),$$

we see a loop in the functional dependency at  $B$ :

$$\begin{array}{ccccc} A_1 & & B_1 & & C_1 \\ & \curvearrowright S \curvearrowleft & & \curvearrowright T \curvearrowleft & \\ & & & & \\ & & B_2 & & C_2 \end{array}$$

Accordingly, the composition  $\tau \odot \sigma$  corresponds to the stable function taking  $(x_1, z_2) \in \mathcal{C}(A_1) \times \mathcal{C}(C_2)$  to  $(x_2, z_1) \in \mathcal{C}(A_2) \times \mathcal{C}(C_1)$  in the least solution to the equations

$$(x_2, y_1) = S(x_1, y_2) \quad \text{and} \quad (y_2, z_1) = T(y_1, z_2).$$

**Theorem 19.** *The bicategory of deterministic strategies on GoI games with rigid 2-cells is equivalent to the free compact-closed category  $\text{Int}(\mathbf{dI})$  of [17] and the Geometry of Interaction category  $\mathcal{G}(\mathbf{dI})$  of [18] built on the category  $\mathbf{dI}$  of *dI*-domains and stable functions.*

Geometry of Interaction started as an investigation of the nature of proofs of linear logic, understood as networks [15]. It has subsequently been tied to optimal reduction in the  $\lambda$ -calculus [47], and inspired implementations via token machines on networks [48], [49]; when the two components of a GoI game match events of exit and entry of a token at a link.

It is straightforward to extend GoI games with winning conditions. A winning condition on a GoI game  $A = A_1 \parallel A_2$  picks out a subset of the configurations  $\mathcal{C}(A)$ , so amounts to specifying a property  $W_A(x_1, x_2)$  of pairs  $(x_1, x_2)$  in  $\mathcal{C}(A_1) \times \mathcal{C}(A_2)$ . That a deterministic strategy from GoI game  $A$  to GoI game  $B = B_1 \parallel B_2$  is winning means

$$W_A(x, g(x, y)) \implies W_B(f(x, y), y),$$

for all  $x \in \mathcal{C}(A_1), y \in \mathcal{C}(B_2)$ , when expressed in terms of the pair of stable functions the strategy determines. In particular, a deterministic winning strategy in the individual GoI game  $B$ , with winning conditions  $W_B$ , corresponds to a stable function  $f : \mathcal{C}(B_2) \rightarrow \mathcal{C}(B_1)$  such that  $\forall y \in \mathcal{C}(B_2). W_B(f(y), y)$ .

With stable spans, unlike with *dI*-domains with stable functions, the operation of parallel composition  $\parallel$  is no longer a



product. Hence, while general strategies  $\sigma : A \dashv\vdash B$  between GoI games are expressible as stable spans  $A_1 \parallel B_2^\perp \dashv\vdash A_2^\perp \parallel B_1$ , their expression doesn't project to an equivalent pair of separate components, as in the deterministic case.

1) *The GoI adjunctions*: For any game  $A$  there is a map of event structures with polarity

$$f_A : A \rightarrow A^+ \parallel A^-,$$

where  $A^+$  is the projection of  $A$  to its +ve events and  $A^-$  is the projection to its -ve events: the map  $f_A$  acts as the identity function on events; it sends a configurations  $x \in \mathcal{C}(A)$  to  $f_A x = x^+ \parallel x^-$ . It determines an adjunction  $f_{A!} \dashv f_{A^*}$  from  $A$  to  $A^+ \parallel A^-$ . Because the game  $A$  is race-free, both  $f_{A!}$  and  $f_{A^*}$  are deterministic strategies. This provides a lax functor from deterministic strategies in general, to those between GoI games. Let  $\sigma : A \dashv\vdash B$  be a deterministic strategy between games  $A$  and  $B$ . Defining  $goi(\sigma) = f_{B!} \circ \sigma \circ f_{A^*}$  we obtain a deterministic strategy

$$goi(\sigma) : A^+ \parallel A^- \dashv\vdash B^+ \parallel B^-.$$

The strategy  $goi(\sigma)$  corresponds to a stable function from  $A^+ \parallel B^-$  to  $A^- \parallel B^+$ , so to a GoI map. The operation  $goi$  only forms a lax functor however: for  $\sigma : A \dashv\vdash B$  and  $\tau : B \dashv\vdash C$ , there is, in general, a nontrivial 2-cell  $goi(\tau \circ \sigma) \Rightarrow goi(\tau) \circ goi(\sigma)$ . This puts pay to  $goi$  being right adjoint to the inclusion functor in a pseudo adjunction from the category of GoI games to deterministic strategies. But, there is a lax pseudo adjunction, of potential use in abstract interpretation.

#### D. Dialectica games

Dialectica categories were devised in the late 1980's by Valeria de Paiva in her Cambridge PhD work with Martin Hyland [19]. The motivation then was to provide a model of linear logic underlying Kurt Gödel's *dialectica interpretation* of first-order logic [20]. They have come to prominence again recently because of a renewed interest in their maps in a variety of contexts, in formalisations of reverse differentiation and back propagation, open games and learners, and as an early occurrence of maps as lenses. The dialectica interpretation underpins most proof-mining techniques [50], [51].

We obtain a particular dialectica category, based on Berry's stable functions, as a full subcategory of deterministic strategies on dialectica games. Dialectica games are obtained as GoI games of imperfect information, intuitively by not allowing Player to see the moves of Opponent.

A *dialectica game* is a GoI game  $A = A_1 \parallel A_2$  with winning conditions, and with imperfect information given as follows. The imperfect information is determined by particularly simple order of access levels:  $1 \prec 2$ . All Player moves, those in  $A_1$ , are assigned to 1 and all Opponent moves, those in  $A_2$ , are assigned to 2. It is helpful to think of the access levels 1 and 2 as representing two rooms separated by a one-way mirror allowing anyone in room 2 to see through to room 1. In a dialectica game, Player is in room 1 and Opponent in room 2. Whereas Opponent can see the moves of Player, and in a counterstrategy make their moves dependent on those of

Player, the moves of Player are made blindly, in that they cannot depend on Opponent's moves.

Although we are mainly interested in strategies *between* dialectica games it is worth pausing to think about strategies *in* a single dialectica game  $A = A_1 \parallel A_2$  with winning conditions  $W_A$ . Because Player moves cannot causally depend on Opponent moves, a deterministic strategy in  $A$  corresponds to a configuration  $x \in \mathcal{C}(A_1)$ ; that it is winning means  $\forall y \in \mathcal{C}(A_2). W_A(x, y)$ . So to have a winning strategy for the dialectica game means

$$\exists x \in \mathcal{C}(A_1) \forall y \in \mathcal{C}(A_2). W_A(x, y).$$

Consider now a deterministic winning strategy  $\sigma$  from a dialectica game  $A = A_1 \parallel A_2$  with winning conditions  $W_A$  to another  $B = B_1 \parallel B_2$  with winning conditions  $W_B$ . Ignoring access levels,  $\sigma$  is also a deterministic strategy between GoI games, so corresponds to a pair of stable functions

$$f : \mathcal{C}(A_1) \times \mathcal{C}(B_2) \rightarrow \mathcal{C}(B_1) \text{ and } g : \mathcal{C}(A_1) \times \mathcal{C}(B_2) \rightarrow \mathcal{C}(A_2).$$

But moves in  $B_2$  have access level 2, moves of  $B_1$  access level 1; a causal dependency in the strategy  $\sigma$  of a move in  $B_1$  on a move in  $B_2$  would violate the access order  $1 \prec 2$ . That no move in  $B_1$  can causally depend on a move in  $B_2$  is reflected in the functional independence of  $f$  on its second argument. As a deterministic strategy between dialectica categories,  $\sigma$  corresponds to a pair of stable functions

$$f : \mathcal{C}(A_1) \rightarrow \mathcal{C}(B_1) \text{ and } g : \mathcal{C}(A_1) \times \mathcal{C}(B_2) \rightarrow \mathcal{C}(A_2),$$

which we can picture as:

$$\begin{array}{ccc} A_1 & \xrightarrow{f} & B_1 \\ & \searrow & \swarrow \\ & & A_2 \\ & \swarrow & \searrow \\ & & B_2 \end{array}$$

That  $\sigma$  is winning means, for all  $x \in \mathcal{C}(A_1), y \in \mathcal{C}(B_2)$ ,

$$W_A(x, g(x, y)) \implies W_B(f(x), y).$$

Pairs of functions  $f, g$  satisfying this winning condition are precisely the maps of de Paiva's construction of a dialectica category from Berry's stable functions.

Such pairs of functions are the *lenses* of functional programming [8], [9]. We recover their at first puzzling composition from the composition of strategies. Let  $\sigma$  be a deterministic strategy from dialectica game  $A$  to dialectica game  $B$ ; and  $\tau$  a deterministic strategy from  $B$  to another dialectica game  $C$ . Assume  $\sigma$  corresponds to a pair of stable functions  $f$  and  $g$ , as above, and analogously that  $\tau$  corresponds to stable functions  $f'$  and  $g'$ . Then, the composition of strategies  $\tau \circ \sigma$  corresponds to the composition of lenses: with first component  $f' \circ f$  and second component taking  $x \in \mathcal{C}(A_1)$  and  $y \in \mathcal{C}(C_2)$  to  $g(x, g'(f(x), y))$ .

**Theorem 20.** *The bicategory of deterministic strategies on dialectica games with rigid 2-cells is equivalent to the dialectica category of [19] built on dI-domains and stable functions.*

*Girard's variant:* In the first half of de Paiva's thesis she concentrates on the construction of dialectica categories. In the second half, she follows up on a suggestion of Girard to explore a variant. This too is easily understood in the context of concurrent games: imitate the work of this section, with GoI games extended with imperfect information, but now with access levels modified to the *discrete* order on 1, 2. Then the causal dependencies of strategies are further reduced and deterministic strategies from  $A = A_1 \parallel A_2$  to  $B = B_1 \parallel B_2$  correspond to pairs of stable functions

$$f : \mathcal{C}(A_1) \rightarrow \mathcal{C}(B_1) \text{ and } g : \mathcal{C}(B_2) \rightarrow \mathcal{C}(A_2).$$

*Combs:* Discussions of causality in science, and quantum information in particular, are often concerned with what causal dependencies are feasible; then structures similar to orders of access levels are used to capture *one-way signalling*, as in dialectica games, and *non-signalling*, as in Girard's variant. In this vein, through another variation of games with imperfect information, we obtain the generalisation of lenses to *combs*, used in quantum architecture and information [11], [52]. Combs provide a commonly used method for imposing higher-order structure on quantum circuits or string diagrams.

Combs arise as strategies between *comb games* which, at least formally, are an obvious generalisation of dialectica games; their name comes from their graphical representation as structures that look like (hair) combs, with teeth representing successive transformations from input to output. An *n-comb game*, for a natural number  $n$ , is an  $n$ -fold parallel composition  $A_1 \parallel A_2 \parallel \dots \parallel A_n$  of purely Player or purely Opponent games  $A_i$  of alternating polarity; it is a game of imperfect information associated with access levels  $1 \prec 2 \prec \dots \prec n$  with moves of component  $A_i$  having access level  $i$ . Dialectica games are 2-comb games with winning conditions.

*Open games and learners:* *Open games and learners* [13], [14] have recently been presented as *parameterised* lenses or optics—in the case of open games with a concept of equilibrium or winning condition [53]. As an example, we obtain a form of open game between dialectica games  $A$  and  $B$  as a strategy  $A \parallel P \dashv\vdash B$ , where  $P$  is a dialectica game of which the configurations specify *strategy profiles*. A variation based on optimal strategies between dialectica games with payoff, following [54], introduces Nash equilibria and takes us into game-theory territory, and to a testing ground for open games and the notions being developed there.

### E. Optics

We show that general, possibly nondeterministic, strategies between dialectica games are precisely *optics* [10], [21] based on stable spans [37], [42], [43]. Recall that a dialectica game comprises  $A_1 \parallel A_2$  where  $A_1$  is a purely Player game, all events of which have access level 1 and  $A_2$  is a purely Opponent game with all events of access level 2, w.r.t. access order  $\Lambda$  specifying  $1 \prec 2$ . We ignore winning conditions.

Let  $A$  and  $B$  be dialectica games. Let  $Q$  be a purely Player  $\Lambda$ -game. Recall that nondeterministic strategies be-

tween purely Player games correspond to stable spans. Consider strategies

$$F : A_1 \dashv\vdash B_1 \parallel Q \text{ and } G : Q \parallel B_2^\perp \dashv\vdash A_2^\perp.$$

Then the strategies  $F$  and  $G$  are between purely Player games, so correspond to stable spans.

As any causal dependencies of  $F$  or  $G$  respect  $\Lambda$ , they are  $\Lambda$ -strategies. Hence the composition

$$A_1 \parallel B_2^\perp \xrightarrow{F \parallel B_2^\perp} B_1 \parallel Q \parallel B_2^\perp \xrightarrow{B_1 \parallel G} B_1 \parallel A_2^\perp$$

is also a  $\Lambda$ -strategy and, being between purely Player games, corresponds to a stable span. The composition, rearranges to a strategy

$$\sigma : A_1 \parallel A_2 \dashv\vdash B_1 \parallel B_2,$$

which is a  $\Lambda$ -strategy, so to a strategy between the original dialectica games  $A$  and  $B$ . We call this strategy  $\text{optic}(F, G)$  and call  $(F, G)$  its *presentation* from  $A$  to  $B$  with *residual*  $Q$ . The terminology is apt, as we'll show strategies obtained in this way coincide with *optics* as usually defined. Presentations can be represented diagrammatically:

$$\begin{array}{ccc} A_1 & \xrightarrow{F} & B_1 \\ & \searrow & \downarrow Q \\ & & A_2 \xleftarrow{G} B_2 \end{array}$$

illustrating how  $F$  and  $G$  are “coupled” via the residual  $Q$ .

As usually defined, an optic is an equivalence class of presentations. Let  $(F, G)$  and  $(F', G')$  be presentations from  $A$  to  $B$  with residuals  $Q$  and  $Q'$  respectively. The equivalence relation  $\sim$  on presentations is that generated by taking  $(F, G) \sim (F', G')$  if, for some  $f : Q \dashv\vdash Q'$ , the following triangles commute

$$\begin{array}{ccc} A_1 \xrightarrow{F'} B_1 \parallel Q' & & Q' \parallel B_2 \xrightarrow{G'} B_1 \\ \searrow F & \uparrow B_1 \parallel f & f \parallel B_2 \uparrow \\ & B_1 \parallel Q & Q \parallel B_2 \end{array} \quad (\sim \text{ def})$$

Presentations of optics compose. Let  $A$ ,  $B$  and  $C$  be dialectica games. Given a presentation  $(F, G)$  from  $A$  to  $B$  with residual  $Q$  and another  $(F', G')$  from  $B$  to  $C$  with residual  $P$  we obtain a presentation from  $A$  to  $C$  with residual  $P \parallel Q$  guided by the diagram

$$\begin{array}{ccccc} A_1 & \xrightarrow{F} & B_1 & \xrightarrow{F'} & C_1 \\ & \searrow & \downarrow Q & \searrow & \downarrow P \\ & & & & A_2 \xleftarrow{G} B_2 \xleftarrow{G'} C_2 \end{array}$$

precisely, as  $((F' \parallel Q) \odot F, G \odot (Q \parallel G') \odot (s_{PQ} \parallel C_2))$ , where  $s_{PQ}$  expresses the symmetry  $P \parallel Q \cong Q \parallel P$ .

Composition preserves  $\sim$  and has the evident identity presentation, with residual the empty game. It follows that optic is functorial and that if  $(F, G) \sim (F', G')$  then

$$\text{optic}(F, G) \cong \text{optic}(F', G').$$

To show any strategy between container games is an optic, we can exploit the monoidal-closure of stable spans. A presentation  $(F, G)$  is  $\sim$ -equivalent to a *canonical presentation*  $(F', G')$  with residual  $Q' = [B_2^\perp \multimap A_2^\perp]$  and  $G'$  as *application* apply: in ( $\sim$ -def), take  $f = \text{curry } G$  and  $F' = (B_1 \parallel f) \odot F$ .

Now, strategies  $\sigma : A \multimap B$ , between dialectica games  $A$  and  $B$ , correspond to canonical presentations. To see this, ignoring the access levels for the moment, a general strategy

$$\sigma : A_1 \parallel A_2 \multimap B_1 \parallel B_2$$

corresponds to a strategy between purely Player games

$$\sigma_1 : A_1 \parallel B_2^\perp \multimap B_1 \parallel A_2^\perp,$$

so to a stable span. From the monoidal-closure of stable spans we can curry  $\sigma_1$ , to obtain a corresponding strategy

$$\sigma_2 : A_1 \multimap [B_2^\perp \multimap (B_1 \parallel A_2^\perp)]$$

with the property  $\sigma_1 \cong \text{apply}_{B_1 \parallel A_2^\perp} \odot (\sigma_2 \parallel B_2^\perp)$ . Recalling the access levels, no event of  $B_1$  can causally depend on an event of  $B_2$ , ensuring that  $\sigma_2$  corresponds to

$$\sigma^+ : A_1 \multimap B_1 \parallel [B_2^\perp \multimap A_2^\perp] \text{ where}$$

$$\sigma_1 \cong \text{apply}_{B_1 \parallel A_2^\perp} \odot (\sigma_2 \parallel B_2^\perp) \cong (B_1 \parallel \text{apply}_{A_2^\perp}) \odot (\sigma^+ \parallel B_2^\perp).$$

It follows that  $(\sigma^+, \text{apply}_{A_2^\perp})$  is a canonical presentation for which

$$\sigma \cong \text{optic}(\sigma^+, \text{apply}_{A_2^\perp}),$$

giving a correspondence between strategies  $\sigma : A \multimap B$  between dialectica games and canonical presentations  $(\sigma^+, \text{apply}_{A_2^\perp})$ .

Via canonical presentations we obtain a bicategory of optics. Its objects are dialectica games. Its maps are stable spans  $A_1 \multimap B_1 \parallel [B_2^\perp \multimap A_2^\perp]$ , with the associated 2-cells, from dialectica game  $A$  to dialectica game  $B$ .

**Theorem 21.** *The bicategories of strategies on dialectica games with rigid 2-cells and that of optics built on stable spans are equivalent.*

### F. Containers

A *container game* is a game of imperfect information  $A$  w.r.t. access levels  $1 \prec 2$ ; each Player move of  $A$  is sent to 1 and each Opponent move to 2. So in  $A$  the only causal dependencies between moves of different polarity are  $\boxplus \leq \boxminus$ .

The configurations of a container game  $A$  have a dependent-type structure. Opponent moves can causally depend on Player moves, but not conversely. Let  $A_1$  denote the subgame comprising the initial substructure of purely Player moves of  $A$ . A configuration  $x \in \mathcal{C}(A_1)$  determines a subgame  $A_2(x)$  comprising the substructure of  $A$  based on all those Opponent moves for which all the Player moves on which they depend appear in  $x$ . A configuration of  $A$  breaks down uniquely into a union  $x \cup y$ , so a pair  $(x, y)$ , where  $x \in \mathcal{C}(A_1)$  and  $y \in \mathcal{C}(A_2(x))$ . We can see the configurations of a container game  $A$  as forming a dependent sum  $\Sigma_{x:A_1} A_2(x)$ . In this way

a container game represents a container type, familiar from functional programming [12]; configurations  $x$  of  $A_1$  are its “shapes,” indexing “positions” configurations  $y$  of  $A_2(x)$ .

We can of course extend a container game  $A$  with winning conditions which we identify with a property  $W_A$  of the dependent sum  $\Sigma_{x:A_1} A_2(x)$ . A deterministic winning strategy in the container game corresponds to a configuration  $x \in \mathcal{C}(A_1)$  such that  $\forall y \in \mathcal{C}(A_2(x)). W_A(x, y)$ .

Strategies between container games respect  $\preceq$  on access levels. A deterministic strategy  $\sigma$  from a container game  $A$  to a container game  $B$  corresponds to a map of container types, also called a *dependent lens*, having type

$$\Sigma_{f:[A_1 \rightarrow B_1]} \Pi_{x:A_1} [B_2(f(x)) \rightarrow A_2(x)]; \quad (*)$$

so  $\sigma$  corresponds to a pair of stable functions

$$f : [A_1 \rightarrow B_1] \text{ and } g : \Pi_{x:A_1} [B_2(f(x)) \rightarrow A_2(x)],$$

where we are using the function space, dependent sum and product of stable functions [41]. Container types built on dI-domains and stable functions arise as a full subcategory of deterministic concurrent games.

**Theorem 22.** *The bicategory of deterministic strategies on container games with rigid 2-cells is equivalent to a full subcategory of containers of dI-domains [12].*

### G. Dependent optics

What about general, nondeterministic, strategies between container games? A way to motivate their characterisation is to observe the isomorphism of the type of a dependent lens  $(*)$  above with

$$\Pi_{x:A_1} \Sigma_{y:B_1} [B_2(y) \rightarrow A_2(x)].$$

This way to present the type of lenses generalises to the monoidal-closed bicategory of stable spans, when we move to the dependent product  $\Pi^s$  and function space of stable spans.

Ignoring winning conditions, a general strategy between container games corresponds to a new form of optic. A *dependent optic* between container games, from  $A$  to  $B$ , is a stable span of type

$$\mathbf{dOp}[A, B] = \Pi_{x:A_1}^s \Sigma_{y:B_1} [B_2(y) \multimap A_2(x)],$$

so a rigid map into  $\mathbf{dOp}[A, B]$ . A 2-cell  $f : F \Rightarrow F'$  between dependent optics  $F, F' : \mathbf{dOp}[A, B]$  is a 2-cell of stable spans. Composition of dependent optics is the stable span

$$\circ : \mathbf{dOp}[B, C] \parallel \mathbf{dOp}[A, B] \multimap \mathbf{dOp}[A, C]$$

described by

$$\begin{aligned} G \circ F &:= \lambda x : A_1. \text{ let } (y, F') \Leftarrow F(x) \text{ in} \\ &\quad \text{let } (z, G') \Leftarrow G(y) \text{ in } (z, F' \circ G'), \end{aligned}$$

where  $F' \circ G' : [C_2(z) \multimap A_2(x)]$  is the composition of stable spans  $G' : [C_2(z) \multimap B_2(y)]$  and  $F' : [B_2(y) \multimap A_2(x)]$ . The identity optic of container game  $A$  acts on  $x : A_1$  to return the identity at the  $x$ -component of  $\Sigma_{x:A_1} [A_2(x) \multimap A_2(x)]$ .

The equivalence of strategies between container games with dependent optics, hinges on recasting  $\mathbf{dOp}[A, B]$  as a strategy  $do[A, B] : A \multimap B$  between container games  $A$  and  $B$ . Any strategy between container games is of course a strategy where we forget the access levels. We can express that a strategy  $\sigma : A \multimap B$  respects the access levels, so is truly a strategy between container games, precisely through the presence of a rigid 2-cell

$$\begin{array}{ccc} & \sigma & \\ A & \xrightarrow{\quad} & B \\ & \downarrow r & \\ & do[A, B] & \end{array}$$

The 2-cell  $r$  is unique, making the strategy  $do[A, B]$  terminal amongst strategies  $\sigma$  between container games, from  $A$  to  $B$ . By restricting  $r$  to Player moves we obtain the dependent optic  $\sigma^+ : \mathbf{dOp}[A, B]$  which corresponds to  $\sigma$ .

**Theorem 23.** *The bicategory of strategies between container games, with rigid 2-cells, is equivalent to the bicategory of dependent optics.*

## VII. ENRICHMENT

Concurrent games and strategies support enrichments, to: probabilistic strategies, also with continuous distributions [55], [56]; quantum strategies [57]; and strategies on the reals [58]. The enrichments now transfer automatically to approaches in functional programming, domain theory and GoI.

The enrichments named above were developed individually and are not always the final story. For instance, the assignment of quantum operators to configurations of strategies in [57] is not functorial w.r.t. inclusion on configurations, a defect when it comes to understanding how the operator of a configuration is built up. The authors' remedy also achieves all the enrichments just named, now uniformly by the same construction.

The construction is w.r.t. a symmetric monoidal category  $(\mathcal{M}, \otimes, \mathbf{I})$ . For example,  $\mathcal{M}$  can be the monoid  $([0, 1], \cdot, 1)$  comprising the unit interval under multiplication (for probabilistic strategies); measurable spaces with Markov kernels (for probabilistic strategies with continuous distributions); CPM, finite-dimensional Hilbert spaces with completely positive maps (for quantum strategies); or Euclidean spaces with smooth maps, to support (reverse) differentiation.

We first extend  $\mathcal{M}$  to allow interaction beyond that from argument to result. The *parameterised category*  $\text{Para}(\mathcal{M})$  has the same objects, now with maps  $(P, f, Q) : X \rightarrow Y$  consisting of  $f : X \otimes P \rightarrow Q \otimes Y$  in  $\mathcal{M}$ ; the parameters  $P$  and  $Q$  allow input and output with the environment. Composition accumulates parameters:  $(R, g, S) \circ (P, f, Q) := (P \otimes R, (Q \otimes g) \circ (f \otimes R), Q \otimes S)$ . Then,

(1) moves  $a$  of a game  $A$  are assigned objects  $\mathcal{H}(a)$  in  $\mathcal{M}$ , extended to  $X \in \text{Con}_A$  by  $\mathcal{H}(X) := \bigotimes_{a \in X} \mathcal{H}(a)$ . (Neutral moves, appearing in interaction, are assigned the  $\otimes$ -unit  $\mathbf{I}$ .)

(2) an  $\mathcal{M}$ -enriched strategy  $\sigma : S \rightarrow A$  has a functor  $\mathcal{Q} : (\mathcal{C}(S)^\circ, \subseteq) \rightarrow \text{Para}(\mathcal{M})$ . To an interval  $x \subseteq x'$  in  $\mathcal{C}(S)^\circ$  it assigns a parameterised map  $\mathcal{Q}(x \subseteq x')$  with input parameters  $\mathcal{H}(\sigma(x' \setminus x)^-)$  and output parameters  $\mathcal{H}(\sigma(x' \setminus x)^+)$ .

The assignment in (2) describes how the internal state is transformed in moving from  $x$  to  $x'$  under interaction with the environment through events  $x' \setminus x$ . It is assumed *oblivious*, i.e.  $\mathcal{Q}(x \subseteq x')$  is always an isomorphism in  $\mathcal{M}$ , expressing that all the input from  $x' \setminus x$  is adjoined to the internal state; this ensures enriched copycat is identity w.r.t. composition.

In the quantum and probabilistic cases, observation is contextual, reflected in the presence of an extra *drop condition*, a form of inclusion-exclusion principle [55], [57]; it requires  $\mathcal{M}$  be enriched over, at least, cancellative commutative monoids.

Moves, their positions, dependencies and polarities, orchestrate the functional dependency and dynamic linkage in composing enriched strategies. Enrichment restricts directly to sub(bi)categories, and the functional approaches we have considered. For example, stable spans when enriched by probability, via the monoid  $([0, 1], \cdot, 1)$ , become Markov kernels, and this enrichment extends to the various forms of optics we have uncovered. Enrichment w.r.t. CPM specialises to an enrichment of Geometry of Interaction with quantum effects.

## VIII. CONCLUSION

Functional paradigms help tame the wild world of interactive computation. On the other hand, discovering the simplifying paradigms has often required considerable ingenuity, for example, in Gödel's Dialectica Interpretation, or in Girard's Geometry of Interaction. The challenges to a functional approach are even more acute with enrichments, say to probabilistic, quantum or real-number computation. The traditional categories of mathematics do not often support all the features required by computation. They often don't have function spaces or support recursion; their extension to computational features has often to be dealt with separately.

As a model of interaction, concurrent games and strategies are more technically challenging and require a new, local way of thinking. But, as has been shown here, they can provide a general context for interaction which specialises to functional paradigms, also in providing enrichments to probabilistic, quantum and real-number computation, without requiring extensions to the traditional categories of mathematics.

Concurrent games and strategies can provide a rationale for new definitions. The form of dependent optic described here appears to be new. It is *derived* as a characterisation of nondeterministic strategies between container games—cf. [22].

One project within reach is that of connecting concurrent games and strategies with the theory of effects [45], [59], specifically with understanding effect handlers [60] as concurrent strategies. Though superficially rather different, effect handlers and concurrent strategies have similar roles: both are concerned with orchestrating the future of a computation contingent on its past and its environment. The results of Section V suggest ways to extend the language of strategies (IV-G) to effects and effect handlers. Such an investigation should also lead to ways to enhance effects and effect handlers to support richer forms of concurrent computation, taking the form of an event structure. As a beginning, the “detectors” of Example 9 generalise to give a form of “event handler.”

## ACKNOWLEDGMENT

I'm grateful for discussions with Bob Atkey, Matteo Capucci, Fredrik Nordvall Forsberg, Bruno Gavranovic, Neil Ghani, Dan Ghica, Samuel Ben Hamou, Jules Hedges, Martin Hyland, Clemens Kupke, J r my Ledent, Sam Lindley, Simon Mirwasser, Valeria de Paiva, Hugo Paquet, Gordon Plotkin and James Wood; I learnt of optics and combs from Jules and J r my. Samuel Ben Hamou, ENS Paris-Saclay, verified the early part of the dialectica-games section for his student internship. Section VII is directly inspired by joint work with Pierre Clairambault and Marc de Visme.

## REFERENCES

- [1] S. Rideau and G. Winskel, "Concurrent strategies," in *LICS 2011*, 2011.
- [2] G. Winskel and M. Nielsen, *Handbook of Logic in Computer Science 4*. OUP, 1995, ch. Models for Concurrency, pp. 1–148.
- [3] G. Winskel, "Event structures," in *Advances in Petri Nets*, ser. LNCS, vol. 255. Springer, 1986, pp. 325–392.
- [4] M. Abadi, M. Isard, and D. G. Murray, "A computational model for tensorflow: an introduction," in *Proceedings of the 1st ACM SIGPLAN International Workshop on Machine Learning and Programming Languages, MAPL@PLDI 2017*. ACM, 2017, pp. 1–7. [Online]. Available: <https://doi.org/10.1145/3088525.3088527>
- [5] S. Brookes, C. A. R. Hoare, and A. W. Roscoe, "A theory of communicating sequential processes," *J. ACM*, vol. 31, pp. 560–599, 1984.
- [6] R. Milner, *A Calculus of Communicating Systems*, ser. Lecture Notes in Computer Science. Springer, 1980, vol. 92. [Online]. Available: <https://doi.org/10.1007/3-540-10235-3>
- [7] S. Castell n and N. Yoshida, "Two sides of the same coin: session types and game semantics: a synchronous side and an asynchronous side," *Proc. ACM Program. Lang.*, vol. 3, no. POPL, pp. 27:1–27:29, 2019. [Online]. Available: <https://doi.org/10.1145/3290340>
- [8] F. J. Oles, *A category theoretic approach to the semantics of programming languages*. PhD Thesis, University of Syracuse, 1982.
- [9] J. N. Foster, M. B. Greenwald, J. T. Moore, B. C. Pierce, and A. Schmitt, "Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem," *ACM Trans. Program. Lang. Syst.*, vol. 29, no. 3, p. 17, 2007. [Online]. Available: <https://doi.org/10.1145/1232420.1232424>
- [10] M. Pickering, J. Gibbons, and N. Wu, "Profunctor optics: Modular data accessors," *Art Sci. Eng. Program.*, vol. 1, no. 2, p. 7, 2017. [Online]. Available: <https://doi.org/10.22152/programming-journal.org/2017/1/7>
- [11] G. Chiribella, G. M. D'Ariano, and P. Perinotti, "Quantum circuit architecture," *Physical Review Letters*, vol. 101, no. 6, Aug 2008.
- [12] M. G. Abbott, T. Altenkirch, and N. Ghani, "Containers: Constructing strictly positive types," *Theor. Comput. Sci.*, vol. 342, no. 1, pp. 3–27, 2005. [Online]. Available: <https://doi.org/10.1016/j.tcs.2005.06.002>
- [13] N. Ghani, J. Hedges, V. Winschel, and P. Zahn, "Compositional game theory," in *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, A. Dawar and E. Gr del, Eds. ACM, 2018, pp. 472–481. [Online]. Available: <https://doi.org/10.1145/3209108.3209165>
- [14] B. Fong, D. I. Spivak, and R. Tuy ras, "Backprop as functor: A compositional perspective on supervised learning," in *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*. IEEE, 2019, pp. 1–13. [Online]. Available: <https://doi.org/10.1109/LICS.2019.8785665>
- [15] J.-Y. Girard, "Geometry of interaction I: Interpretation of System F," in *Logic Colloquium '88*, C. Bonotto, S. R. Ferro, Valentini, and A. Zanardo, Eds. North-Holland, 1989, p. 221–260.
- [16] S. Abramsky and R. Jagadeesan, "New foundations for the Geometry of Interaction," *Inf. Comput.*, vol. 111, no. 1, pp. 53–119, 1994. [Online]. Available: <https://doi.org/10.1006/inco.1994.1041>
- [17] R. S. Andr  Joyal and D. Verity, "Traced monoidal categories," *Math. Proc. Camb. Phil. Soc. (1996)*, 119, pp. 447–468, 1996.
- [18] S. Abramsky, E. Haghverdi, and P. J. Scott, "Geometry of interaction and linear combinatory algebras," *Math. Struct. Comput. Sci.*, vol. 12, no. 5, pp. 625–665, 2002. [Online]. Available: <https://doi.org/10.1017/S0960129502003730>
- [19] V. de Paiva, *The Dialectica categories*. PhD Thesis, University of Cambridge, 1988.
- [20] J. Avigad and S. Feferman, "G del's functional ("dialectica") interpretation," p. 337–405, 1999.
- [21] M. Riley, "Categories of optics," 2018. [Online]. Available: <http://arxiv.org/abs/1809.00738>
- [22] J. Hedges, "Towards dependent optics (blog post)," 2020. [Online]. Available: <https://julesh.com/2020/06/10/towards-dependent-optics/>
- [23] G. Winskel, "Making concurrency functional (extended version)," 2023. [Online]. Available: <http://arxiv.org/abs/2202.13910>
- [24] —, "Event structure semantics for CCS and related languages," in *ICALP'82*, ser. LNCS, vol. 140. Springer, A full version is available from Winskel's Cambridge homepage, 1982.
- [25] G. Berry, "Stable models of typed lambda-calculi," in *ICALP*, ser. Lecture Notes in Computer Science, vol. 62. Springer, 1978, pp. 72–89.
- [26] G. Winskel, "Strategies as profunctors," in *FOSSACS 2013*, ser. Lecture Notes in Computer Science. Springer, 2013.
- [27] M. Hyland, "Some reasons for generalising domain theory," *Mathematical Structures in Computer Science*, vol. 20, no. 2, pp. 239–265, 2010.
- [28] C. Faggian and M. Piccolo, "Partial orders, event structures and linear strategies," in *TLCA '09*, ser. LNCS, vol. 5608. Springer, 2009.
- [29] J. Conway, *On Numbers and Games*. Wellesley, MA: A K Peters, 2000.
- [30] A. Joyal, "Remarques sur la th orie des jeux   deux personnes," *Gazette des sciences math matiques du Qu bec*, 1(4), 1997.
- [31] G. Winskel, "Deterministic concurrent strategies," *Formal Asp. Comput.*, vol. 24, no. 4-6, pp. 647–660, 2012.
- [32] —, *ECSYM Notes: Event Structures, Stable Families and Concurrent Games*, 2017. [Online]. Available: <http://www.cl.cam.ac.uk/~gw104/ecsym-notes.pdf>
- [33] P.-A. Mellis and S. Mimram, "Asynchronous games : innocence without alternation," in *CONCUR '07*, ser. LNCS, vol. 4703. Springer, 2007.
- [34] P. Clairambault, J. Gutierrez, and G. Winskel, "The winning ways of concurrent games," in *LICS 2012: 235-244*, 2012.
- [35] G. Winskel, "Winning, losing and drawing in concurrent games with perfect or imperfect information," in *Festschrift for Dexter Kozen*, ser. LNCS, vol. 7230. Springer, 2012.
- [36] S. Castell n, J. Hayman, M. Lasson, and G. Winskel, "Strategies as concurrent processes," *Electr. Notes Theor. Comput. Sci.*, vol. 308, pp. 87–107, 2014.
- [37] G. Winskel, "Events, causality and symmetry," *Comput. J.*, vol. 54, no. 1, pp. 42–57, 2011.
- [38] N. Johnson and D. Yau, "2-dimensional categories," 2020. [Online]. Available: <http://arxiv.org/abs/2002.06055>
- [39] H. Paquet, "Probabilistic concurrent game semantics," Ph.D. dissertation, Computer Laboratory, University of Cambridge, UK, 2020.
- [40] T. Coquand, C. A. Gunter, and G. Winskel, "dI-domains as a model of polymorphism," in *Mathematical Foundations of Programming Language Semantics, 3rd Workshop, Tulane University, New Orleans, Louisiana, USA, April 8-10, 1987, Proceedings*, ser. Lecture Notes in Computer Science, M. G. Main, A. Melton, M. W. Mislove, and D. A. Schmidt, Eds., vol. 298. Springer, 1987, pp. 344–363. [Online]. Available: [https://doi.org/10.1007/3-540-19020-1\\_18](https://doi.org/10.1007/3-540-19020-1_18)
- [41] —, "Domain theoretic models of polymorphism," *Inf. Comput.*, vol. 81, no. 2, pp. 123–167, 1989. [Online]. Available: [https://doi.org/10.1016/0890-5401\(89\)90068-0](https://doi.org/10.1016/0890-5401(89)90068-0)
- [42] M. Nygaard, *Domain theory for concurrency*. PhD Thesis, Aarhus University, 2003. [Online]. Available: <https://www.brics.dk/DS/03/13/BRICS-DS-03-13.pdf>
- [43] L. Saunders-Evans and G. Winskel, "Event structure spans for nondeterministic dataflow," *Electr. Notes Theor. Comput. Sci.* 175(3): 109-129, 2007.
- [44] M. Fiore, N. Gambino, M. Hyland, and G. Winskel, "Relative pseudomonads, kelisli bicategories and substitution monoidal structures," *Selecta Mathematica - New Series*, vol. 24, no. 3, pp. 2791–2830, 2018.
- [45] E. Moggi, "Computational lambda-calculus and monads," in *Proceedings of the Fourth Annual Symposium on Logic in Computer Science (LICS '89), Pacific Grove, California, USA, June 5-8, 1989*. IEEE Computer Society, 1989, pp. 14–23. [Online]. Available: <https://doi.org/10.1109/LICS.1989.39155>
- [46] M. Nygaard and G. Winskel, "Linearity in process languages," in *LICS'02*. IEEE Computer Society, 2002.
- [47] G. Gonthier, M. Abadi, and J. L vy, "The geometry of optimal lambda reduction," in *Conference Record of the Nineteenth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming*

- Languages, Albuquerque, New Mexico, USA, January 19-22, 1992*, R. Sethi, Ed. ACM Press, 1992, pp. 15–26. [Online]. Available: <https://doi.org/10.1145/143165.143172>
- [48] I. Mackie, “The geometry of interaction machine,” in *Conference Record of POPL’95: 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Francisco, California, USA, January 23-25, 1995*, R. K. Cytron and P. Lee, Eds. ACM Press, 1995, pp. 198–208. [Online]. Available: <https://doi.org/10.1145/199448.199483>
- [49] K. Muroya and D. R. Ghica, “The dynamic geometry of interaction machine: A token-guided graph rewriter,” *Log. Methods Comput. Sci.*, vol. 15, no. 4, 2019. [Online]. Available: [https://doi.org/10.23638/LMCS-15\(4:7\)2019](https://doi.org/10.23638/LMCS-15(4:7)2019)
- [50] S. Feferman, “Kreisel’s ‘unwinding program’, Kreiselianna: about and around Georg Kreisel,” p. 247–273, 1996.
- [51] U. Kohlenbach, *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Springer Monographs in Mathematics, 2008.
- [52] A. Kissinger and S. Uijlen, “A categorical semantics for causal structure,” in *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*. IEEE Computer Society, 2017, pp. 1–12. [Online]. Available: <https://doi.org/10.1109/LICS.2017.8005095>
- [53] M. Capucci, B. Gavranović, J. Hedges, and E. F. Rischel, “Towards foundations of categorical cybernetics,” 2021. [Online]. Available: <http://arxiv.org/abs/2105.06332>
- [54] P. Clairambault and G. Winskel, “On concurrent games with payoff,” *Electr. Notes Theor. Comput. Sci.* 298: 71-92, 2013.
- [55] G. Winskel, “Distributed probabilistic and quantum strategies,” *Electr. Notes Theor. Comput. Sci.* 298: 403-425, 2013.
- [56] H. Paquet and G. Winskel, “Continuous probability distributions in concurrent games,” in *Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2018, Dalhousie University, Halifax, Canada, June 6-9, 2018*, ser. Electronic Notes in Theoretical Computer Science, S. Staton, Ed., vol. 341. Elsevier, 2018, pp. 321–344. [Online]. Available: <https://doi.org/10.1016/j.entcs.2018.11.016>
- [57] P. Clairambault, M. de Visme, and G. Winskel, “Game semantics for quantum programming,” *Proc. ACM Program. Lang.*, vol. 3, no. POPL, pp. 32:1–32:29, 2019. [Online]. Available: <https://doi.org/10.1145/3290345>
- [58] A. Alcolei, “Enriched concurrent games : witnesses for proofs and resource analysis. (jeux concurrents enrichis : témoins pour les preuves et les ressources),” Ph.D. dissertation, University of Lyon, France, 2019. [Online]. Available: <https://tel.archives-ouvertes.fr/tel-02448974>
- [59] G. D. Plotkin and A. J. Power, “Computational effects and operations: An overview,” *Electron. Notes Theor. Comput. Sci.*, vol. 73, pp. 149–163, 2004. [Online]. Available: <https://doi.org/10.1016/j.entcs.2004.08.008>
- [60] G. D. Plotkin and M. Pretnar, “Handlers of algebraic effects,” in *Programming Languages and Systems, 18th European Symposium on Programming, ESOP 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*, ser. Lecture Notes in Computer Science, G. Castagna, Ed., vol. 5502. Springer, 2009, pp. 80–94. [Online]. Available: [https://doi.org/10.1007/978-3-642-00590-9\\_7](https://doi.org/10.1007/978-3-642-00590-9_7)