BCS-FACS Landin Seminar, 17 December 2021
Making Concurrency Functional

Glynn Winskel, Huawei Edinburgh

Two principal paradigms of computation

Functional: historically important paradigm, computation as functions; with interaction by function composition.
Many refinements: lenses, optics, combs, containers, dependent lenses, dependent optics, open games and learners, ...

Interactive: a more recent paradigm, computation as interacting processes. Many approaches, less settled, often syntax-driven.
Here a maths-driven foundation based on distributed/concurrent games based on event structures, with interaction by composition of strategies.
Idea: types/constraints are games and programs/processes are strategies.

This talk:

A bridge: how specialising games yields the functional paradigms.

Peter Landin (1930 - 2009)

My meeting with Peter Landin:

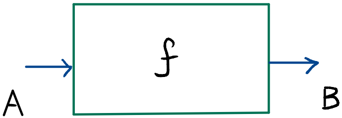
Interview for a PhD place at
Queen Mary College, London, 1977.
Friendly informality: student cafe,
coffee and Gauloises

"researcher in the style of Strachey"

Concurrency: "extent of variables,"
their temporal overlap of activation and
dependence

On a corner of computer printout
"Glynn, Are you still interested? Peter"
(I went to Edinburgh)

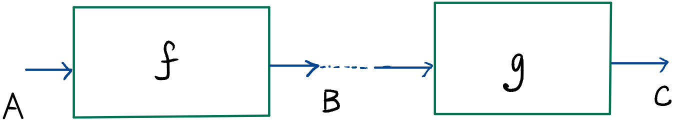
Interaction via functions



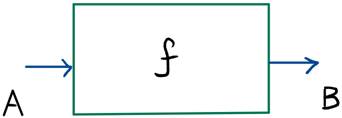
a function $f: A \to B$

Interaction via functions

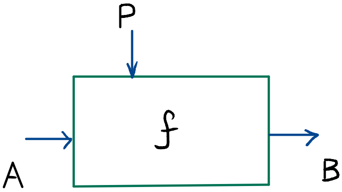
$g \circ f$



Interaction via functions



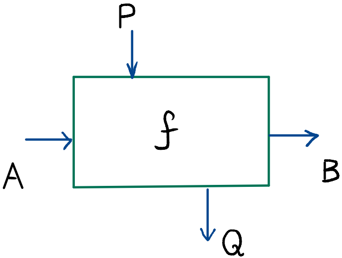
Interaction via functions



a parameterised function

$$f : A \otimes P \longrightarrow B$$

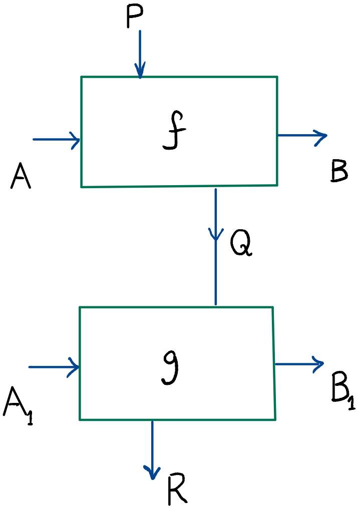
Interaction via functions



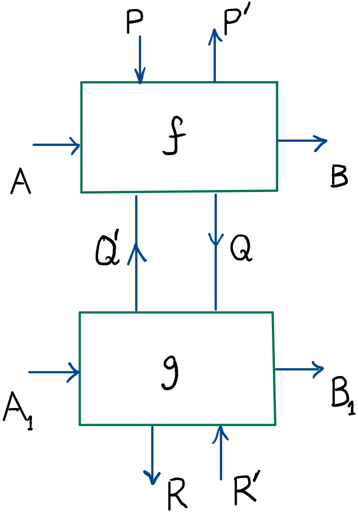
a parameterised function

$$f : A \otimes P \longrightarrow Q \otimes B$$

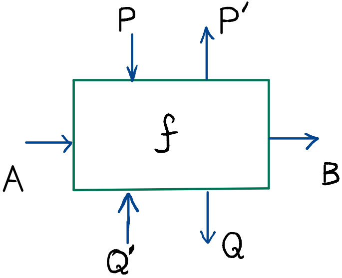
Interaction via functions



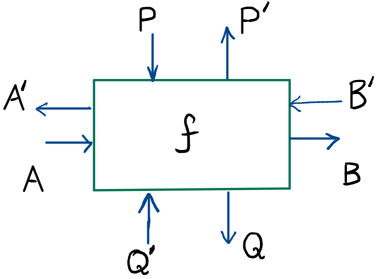
Interaction via functions



Interaction via functions



Interaction via functions



+ back propagation

- Nature of functions? Generally need enrichments to functions which are partial, continuous, nondeterministic, probabilistic, quantum, smooth, ...
- Functions and their usual IO types can only give a static, partial picture of the dynamics of interaction. Dependent types can sometimes help, but ...
- Need a way to describe and orchestrate temporal pattern of interaction, its fine-grained dependencies and dynamic linkage.

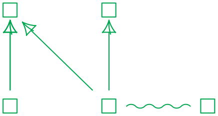
Event structures - of the simplest kind

Definition

An event structure comprises $(E, \leqslant, \#)$, consisting of a set of events E
- partially ordered by $\leqslant$, the causal dependency relation, and
- a binary irreflexive symmetric relation, the conflict relation,

which satisfy $\{e' \mid e' \leqslant e\}$ is finite and $e_1' \geqslant e_1 \# e_2 \leqslant e_2' \implies e_1' \# e_2'$.

Two events are concurrent when neither in conflict nor causally related.



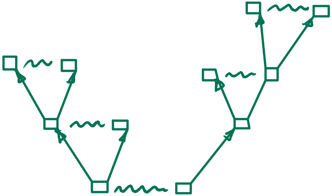
Definition

The configurations, $\mathcal{C}(E)$, of an event structure E consist of those subsets $x \subseteq E$ which are

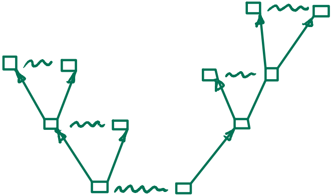
Consistent: don't have $e \# e'$ for any events $e, e' \in x$, and

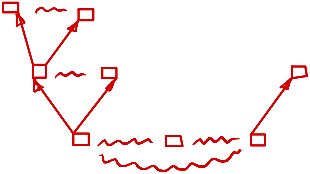
Down-closed: $e' \leqslant e \in x \implies e' \in x$.

Trees as event structures

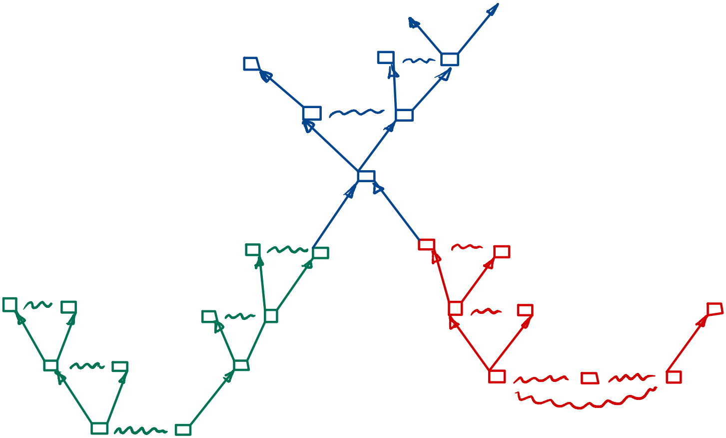


Trees as event structures





Trees as event structures



Games for interaction: the paradigm of Conway, Joyal

In 2-party games read Player vs. Opponent as Process vs. Environment.

Assume operations on (2-party) games:

Dual game $G^\perp$ *- interchange the role of Player and Opponent;*
Counter-strategy = strategy for Opponent = strategy for Player in dual game.

Parallel composition of games $G \| H$.

A strategy (for Player) from a game G *to a game* H *= strategy in* $G^\perp \| H$.
A strategy (for Player) from a game H *to a game* K *= strategy in* $H^\perp \| K$.

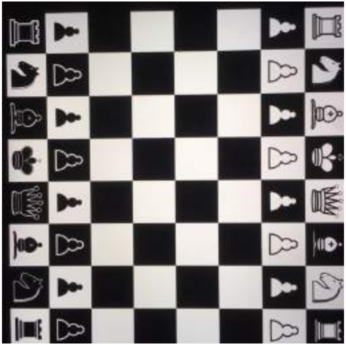
Compose by letting them play against each other in the common game H.

$\rightsquigarrow$ *has identity the Copycat strategy in* $G^\perp \| G$, *so from* G *to* G *...*

Copycat strategy illustrated

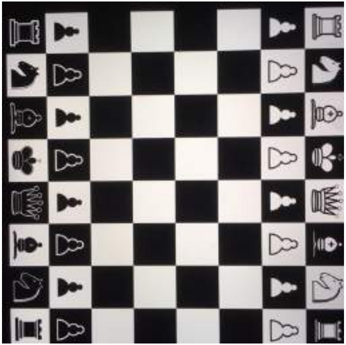
Chess, the game in which Player plays Black.

Chess$^{\perp}$



GM1 Player

Chess



Player GM2

Distributed games = Conway-Joyal on event structures

Games are represented by event structures with polarity, an event structure $(E, \leqslant, \#)$ where events E carry a polarity, plus $+$ or $-$ for Player/Opponent. Assume race-free: no immediate *conflict between Player and Opponent events.*

Dual, $B^{\perp}$, of an event structure with polarity B is a copy of the event structure B with a reversal of polarities; this switches the roles of Player and Opponent.

(Simple) Parallel composition: $A \| B$, by non-conflicting juxtaposition.

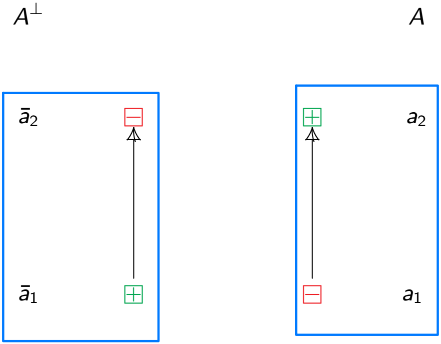
A strategy from a game A to a game B is a strategy in $A^{\perp} \| B$, written

$$\sigma : A \longrightarrow\!\!\!\!\!+\!\!\!\!> B$$

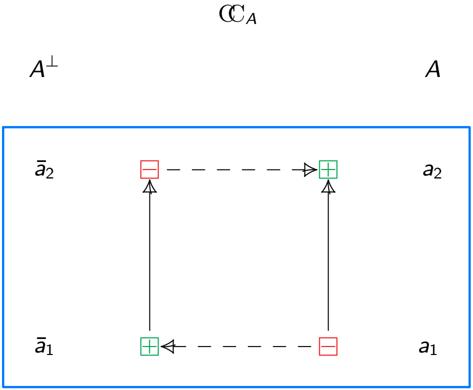
But what's a strategy in game?

Roughly, a strategy (for Player) should be a choice of moves for Player together with their causal dependencies on Opponent moves.

Example: Copycat strategy from A to A



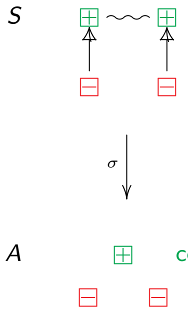
Example: Copycat strategy from A to A



In general a strategy in a game A comprises

*an event structure S and a function on events $\sigma : S \to A$,
so for all configurations x of S, its image $\sigma\, x$ is a configuration of A and
if $s_1, s_2 \in x$ and $\sigma(s_1) = \sigma(s_2)$ then $s_1 = s_2$.*

e.g.

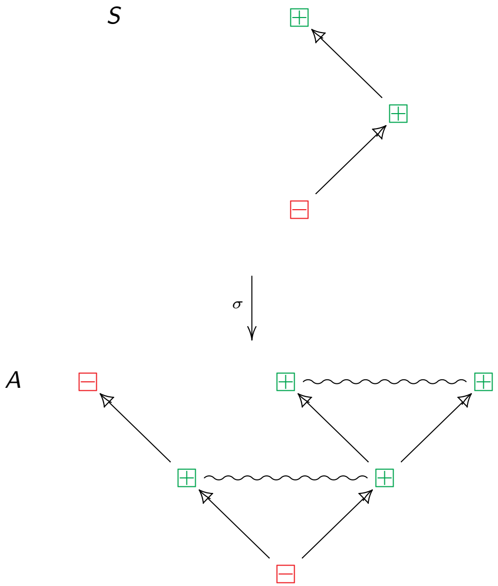


configurations of S = "states of play"

configurations of A = "positions of the game,"

*which is (1) receptive and (2) innocent:
(1) any Opponent move at a position in A is allowed at the state of play in S;
(2) in S the only additional causal dependencies are $\boxminus \rightarrow \boxplus$.*

When games are trees:



The strategy: Player takes the initiative.

Composition of strategies $\sigma : A \twoheadrightarrow B$ and $\tau : B \twoheadrightarrow C$

To compose

$$\begin{tikzcd} S \arrow[d, "\sigma"'] & T \arrow[d, "\tau"] \\ A^{\perp} \| B & B^{\perp} \| C \end{tikzcd}$$

synchronise complementary moves over common game B (via pullback); then hide synchronisations (via partial-total factorisation):

$$\begin{tikzcd} \textit{before hiding} \quad T \circledast S \arrow[r, dotted] \arrow[d, "\tau \circledast \sigma"'] & T \odot S \quad \textit{after hiding} \arrow[d, "\tau \odot \sigma"] \\ A^{\perp} \| B \| C \arrow[r, dotted] & A^{\perp} \| C \end{tikzcd}$$

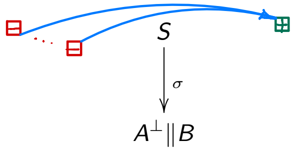
Theorem (Rideau, W)

Conditions of receptivity and innocence on a strategy are precisely those needed to make copycat identity w.r.t. composition.

Special case: Gérard Berry's dI-domains and stable functions

A concurrent strategy is deterministic when conflicting behaviour of Player implies conflicting behaviour of Opponent.

Let A and B be purely Player games.
A strategy from A to B is a strategy in $A^{\perp} \| B$:



Deterministic strategies $\sigma : A \dashrightarrow B$ correspond to stable functions $f : (\mathcal{C}(A), \subseteq) \to (\mathcal{C}(B), \subseteq)$ between dI-domains. They have a function space $[A \to B]$ w.r.t. product $\|$. A stable function preserves least upper bounds of directed sets and meets of compatible elements.

Theorem

There is an equivalence between deterministic strategies between purely Player games and Gérard Berry's dI-domains and stable functions.
The equivalence restricts to one is with Jean-Yves Girard's coherence spaces when causal dependencies are the trivial identity relation.

Special case: stable spans

Let A and B be purely Player games. Strategies, possibly nondeterministic, $\sigma : A \longrightarrow\!\!\!\!\!+\!\!\!\!> B$ correspond to stable spans

$$\begin{tikzcd}
 & S^{+} \arrow[dl, "\text{demand}"'] \arrow[dr, "\text{output}"] & \\
A & & B\,,
\end{tikzcd}$$

roughly, nondeterministic stable functions; they are (special) profunctors.
Stable spans have been central in providing a compositional model for nondeterministic dataflow; the feedback of nondeterministic dataflow is given by the trace of strategies. Stable spans have a function space $[A \multimap B]$ w.r.t. $\|$.

Theorem

There is an equivalence between strategies between purely Player games and stable spans between event structures.

Girard's Geometry of Interaction (the nature of proofs as networks)

> *A GoI game A comprises a parallel composition $A_1 \| A_2$ where A_1 is a purely Player game and A_2 is a purely Opponent game.*

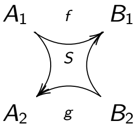
A strategy σ from a GoI game $A := A_1 \| A_2$ to a GoI game $B := B_1 \| B_2$ is a strategy in $A^\perp \| B$, *i.e.*

$$\begin{array}{cccc} \boxminus & A_1^\perp & B_1 & \boxplus \\ \\ \\ \boxplus & A_2^\perp & B_2 & \boxminus \end{array}$$

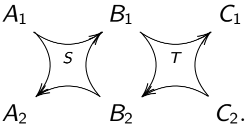
so a strategy $A_1 \| B_2^\perp \twoheadrightarrow A_2^\perp \| B_1$ between purely Player games.
A deterministic strategy from A to B corresponds to a pair of stable functions

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

summarised by



with composition



We recover Abramsky and Jagadeesan's GoI construction, but now starting from *stable* domain theory. Applications: optimal reduction, token machines

Extensions - winning conditions and imperfect information

A *winning condition* on a game A specifies those of its configurations which are a win for Player.
A strategy in A is *winning* (for Player) if in any maximal play for Player results in a winning configuration of A.
A strategy from A to B, *i.e.* in $A^\perp \| B$, is *winning* if, in any maximal play for Player, a win in A implies a win in B.
Winning strategies compose.

Imperfect information via an access order $(\Lambda, \preceq)$ on moves of games.
Idea: moves have an access level and can only depend on moves $\preceq$-lower.
Causal dependency of the game and strategy must respect $\preceq$:

$$\begin{tikzcd} S \arrow[d, "\sigma"'] \\ A \arrow[r, "\lambda"] & (\Lambda, \preceq) \end{tikzcd} \qquad s' \leqslant_S s \quad \text{implies} \quad \lambda\sigma(s') \preceq \lambda\sigma(s)$$

Such Λ-strategies compose.

Gödel's dialectica interpretation

A *dialectica game* *is a GoI game* $A = A_1 \| A_2$ *with winning conditions and access levels*

$$1 \prec 2$$

with Player moves A_1 *assigned* 1 *and Opponent moves* A_2 *assigned* 2.

A deterministic strategy between dialectica games, from A to B, is a lens:

$$\begin{tikzcd} A_1 \arrow[r, "f"] \arrow[d, bend left=60, "g"] & B_1 \\ A_2 & B_2 \arrow[l] \end{tikzcd}$$

It's winning means

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

for all configurations x of A_1 and y of B_2.

Deterministic strategies between dialectica games coincide with Gödel's dialectica interpretation of proofs in arithmetic as higher-order functions [Gödel, de Paiva, Hyland]. Applications: proof mining [Kreisel, Kohlenbach].

Girard's variant and Combs

Girard's variant [de Paiva]: Just changing Λ to the discrete order

$$1 \bullet \quad \bullet 2$$

enforces *non-signalling*, rather than the *one-way signalling* of dialectica games, between moves of the two different access levels.

A deterministic strategy $A \longrightarrow\!\!\!\!\!\!\!+\;\; B$ now corresponds to a pair of stable functions

$$A_1 \xrightarrow{f} B_1$$
$$A_2 \xleftarrow[g]{} B_2.$$

Combs of quantum architecture arise as strategies between "comb games," comprising n-fold alternating-polarity, parallel compositions

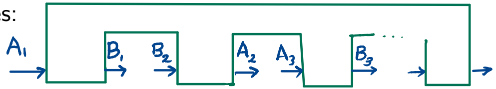
$$A_1 \| A_2 \| \cdots \| A_n$$

of purely Player and purely Opponent games over access levels

$$1 \prec 2 \prec \cdots \prec n$$

A strategy between comb games:

$\sigma : A \longrightarrow\!\!\!\!\!\!\!+\;\; B$



Optics

General, possibly nondeterministic, strategies between dialectica games $\sigma : A \mathrel{\rightarrow\!\!\!\!\!\!+} B$ correspond to optics built from stable spans

$$F : A_1 \mathrel{\rightarrow\!\!\!\!\!\!+} B_1 \| Q \quad \text{and} \quad G : Q \| B_2^{\perp} \mathrel{\rightarrow\!\!\!\!\!\!+} A_2^{\perp}$$

as the composite

$$\begin{tikzcd} A_1 \arrow[rr, "F"] \arrow[dr, bend left] & & B_1 \\ & Q \arrow[dl, bend left] & \\ A_2 & & B_2 \arrow[ll, "G"] \end{tikzcd} .$$

Composition of strategies coincides with composition of optics

$$\begin{tikzcd} A_1 \arrow[rr, "F"] \arrow[dr, bend left] & & B_1 \arrow[rr, "F'"] \arrow[dr, bend left] & & C_1 \\ & Q \arrow[dl, bend left] & & P \arrow[dl, bend left] & \\ A_2 & & B_2 \arrow[ll, "G"] & & C_2 \arrow[ll, "G'"] \end{tikzcd} .$$

Strategies $\sigma : A \mathrel{\rightarrow\!\!\!\!\!\!+} B$ between dialectica games, so optics on stable spans, correspond to stable spans of type $A_1 \multimap B_1 \| [B_2 \multimap A_2]$.

Containers (data structures where "shapes" index "positions")

A container game is a game of imperfect information A w.r.t. access levels $1 \prec 2$; each Player move of A assigned 1 and each Opponent move 2.

The only causal dependencies in A relating moves of different polarities: $\boxplus < \boxminus$
The game A comprises an initial Player part A_1 followed by a dependent Opponent part A_2; its configurations have form $x \cup y$ where x comprises solely Player moves and y solely Opponent moves, dependent on x.
Hence the container game A corresponds to a dependent type $\Sigma_{x:A_1} A_2(x)$.

A deterministic strategy between container games $\sigma : A \twoheadrightarrow B$ corresponds to a dependent lens, a pair of stable functions

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

I.e. to an element of type

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

so, less standardly, to an element of the isomorphic type

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

Dependent optics (new?)

General, possibly nondeterministic, strategies between container games $\sigma : A \twoheadrightarrow B$ correspond to "dependent optics" of type

$$\mathrm{dOp}[A, B] = \Pi^{\circ}_{x:A_1} \Sigma_{y:B_1} \ [B_2(y) \multimap A_2(x)] \,,$$

where Π° is a dependent product of stable spans.
Dependent optics compose by

$$\circ : \mathrm{dOp}[B, C] \parallel \mathrm{dOp}[A, B] \twoheadrightarrow \mathrm{dOp}[A, C]$$

described, a little informally, as

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

where $F' \odot G'$ is the composition of stable spans

$$G' : [C_2(z) \multimap B_2(y)] \ \text{ and } \ F' : [B_2(y) \multimap A_2(x)] \,.$$

Functional paradigms that arise as special strategies inherit the enrichments of strategies, probabilistic, quantum, real-number, ...

Enrichments via parameterised maps

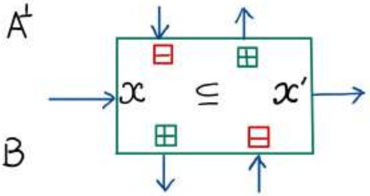
Games and strategies support enrichments to: probabilistic strategies, also with continuous distributions; quantum strategies; smooth functions to support differentiation.

Recent realisation: all enrichments can be achieved uniformly by the same construction, using parameterised "functions" [Clairambault, de Visme, W].

W.r.t. a symmetric monoidal category $(\mathcal{M}, \otimes, I)$, *e.g.* $([0,1], \cdot, 1)$ or CPM,

1. moves of a game are assigned objects in $\mathcal{M}$;

2. intervals $x \subseteq x'$ of finite configurations of S in a strategy are assigned parameterised maps over $\mathcal{M}$, the polarity of events deciding which way the parameter maps point, as input or output:



The events, their dependencies and polarities, orchestrate the functional dependency and dynamic linkage in composing *enriched* games and strategies.

Conclusion

Functional approach can help tame wild world of concurrent interaction through providing simpler models.
But adapting functions to interaction often requires considerable ingenuity, especially when requiring enrichments, *e.g.* probabilistic, quantum, real-number.

Distributed games and strategies provide a broad general context for interaction which can be specialised to functional paradigms;
also in providing enrichments to probabilistic, quantum and real number computation *etc.*

THANK YOU!