The Next-Generation Semantics?

Glynn Winskel, University of Cambridge Computer Laboratory

The next-generation domain theory? An intensional theory to capture the *ways* of computing, to near operational concerns and reasoning? An event-based theory?

To give an idea of

• **partial-order/causal models**, a form of model becoming important in a range of areas from security, systems, model checking, systems biology, to proof theory

• why I believe such models will become central in **semantics** of computation and can combine the two approaches, **operational** and **denotational** semantics

• a new result characterizing **concurrent strategies**

What is a computational process?

Pre 1930's: **An algorithm** *(informal)*

Post 1930's: **An effective partial function** $\mathbf{f} : \mathbb{N} \to \mathbb{N}$ *(mathematical)*

Mid 1960's : Christopher Strachey founded denotational semantics to understand *stored programs*, *loops*, *recursive programs* on *advanced datatypes*, often with *infinite objects* (at least conceptually): infinite lists, infinite sets, functions, functions on functions on functions, ...
A program denotes a term within the λ-calculus, a calculus of functions (but is it?): $\mathbf{t} ::= \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}')$

Late 1960's: Dana Scott: Computable functions acting on infinite objects can only do so via approximations (topology!). **A computational process is an (effective) continuous function $\mathbf{f} : \mathbf{D} \to \mathbf{E}$ between special topological spaces, 'domains.'** Recursive definitions as least fixed points.

... ? ...

Representations of traditional domains

What is the information order? What are the 'units' of information?
Two answers:

('Topological') [Scott]: *Propositions* about finite properties;
more information corresponds to more propositions being true.
Functions are ordered pointwise. Can represent domains via logical theories ('Information systems', 'Logic of domains').

('Temporal') [Berry]: *Events* (atomic actions);
more information corresponds to more events having occurred.
Intensional 'stable order' on 'stable' functions. ('Stable domain theory')
Can represent Berry's dI domains as event structures.

Event structures

An *event structure* comprises $(E, \mathrm{Con}, \leq)$, consisting of a set of *events* E

- partially ordered by $\leq$, the *causal dependency relation*, and

- a nonempty family Con of finite subsets of E, the *consistency relation*,

which satisfy

$$\{e' \mid e' \leq e\} \text{ is finite for all } e \in E,$$

$$\{e\} \in \mathrm{Con} \text{ for all } e \in E,$$

$$Y \subseteq X \in \mathrm{Con} \Rightarrow Y \in \mathrm{Con}, \quad \text{and}$$

$$X \in \mathrm{Con} \ \& \ e \leq e' \in X \Rightarrow X \cup \{e\} \in \mathrm{Con}.$$

Configurations of an event structure

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

Consistent: $\forall X \subseteq_{\mathrm{fin}} x.\ X \in \mathrm{Con}$ and

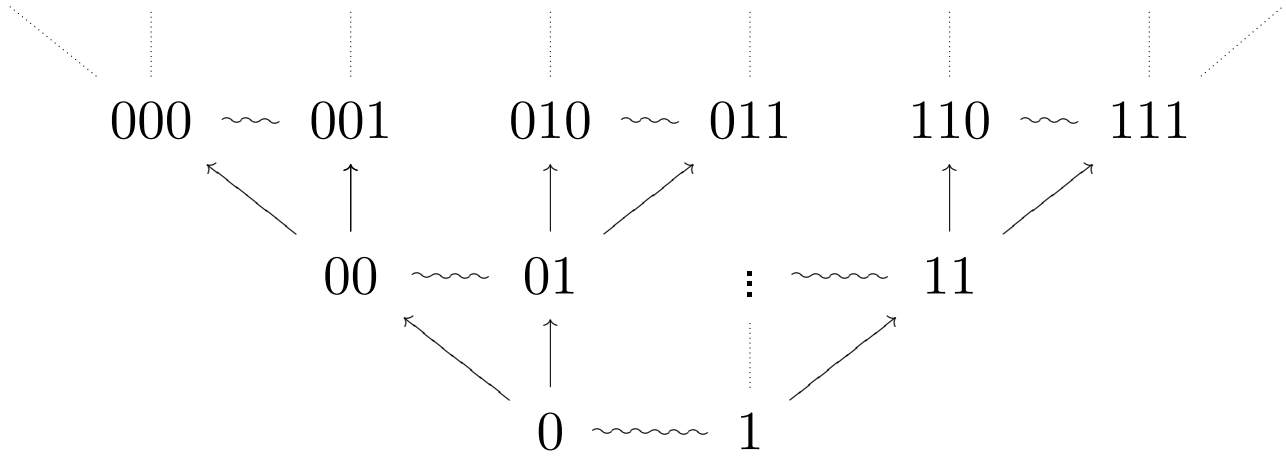
Down-closed: $\forall e, e'.\ e' \leq e \in x \Rightarrow e' \in x$.

For an event e the set $[e] =_{\mathrm{def}} \{e' \in E \mid e' \leq e\}$ is a configuration describing the whole causal history of the event e.

$x \subseteq x'$, *i.e.* x is a sub-configuration of x', means that x is a sub-history of x'.

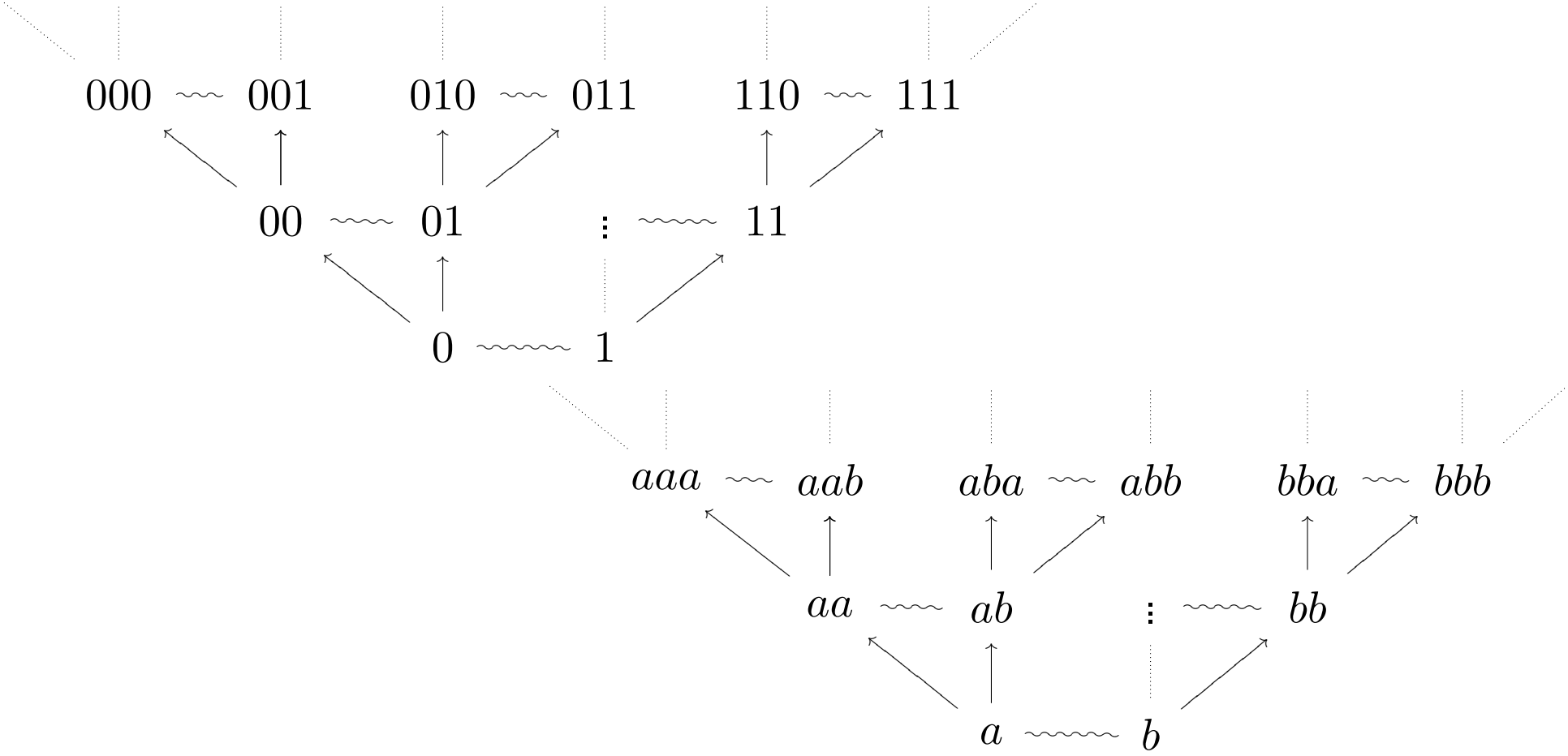
If E is countable, $(\mathcal{C}(E), \subseteq)$ is a dI domain (and all such are so obtained).

Event structures as types, e.g., Streams as event structures



$\rightsquigarrow$ conflict (inconsistency) $\longrightarrow$ causal dependency $\leq$

Simple parallel composition



Event structures as processes

- Semantics of synchronising processes [Hoare, Milner] can be expressed in terms of universal constructions on event structures, and other models.
- Relations between models via adjunctions.
- Strong bisimulation via open maps, defined diagrammatically.

In this context, a *simulation map* of event structures $f : E \to E'$ is a partial function on events $f : E \rightharpoonup E'$ such that for all $x \in \mathcal{C}(E)$

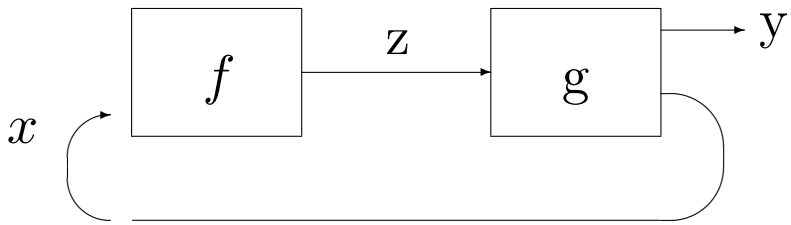
$$fx \in \mathcal{C}(E') \text{ and}$$

$$\text{if } e_1, e_2 \in x \text{ and } f(e_1) = f(e_2), \text{ then } e_1 = e_2. \qquad \textit{('event linearity')}$$

Idea: the occurrence of an event e in E induces the coincident occurrence of the event $f(e)$ in E' whenever it is defined.

[Event structures as types and processes?]

Deterministic dataflow—Kahn networks



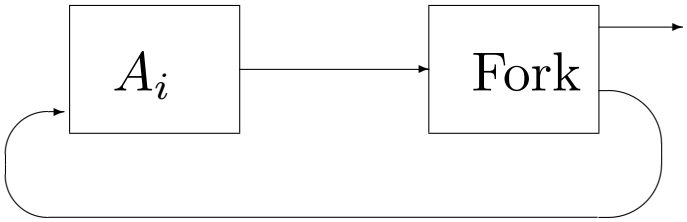
A process built from basic processes connected by channels at which they input and output.

Simple semantics: Associate channels with streams x, y, z.
Provided f and g are continuous functions on streams there is a least fixed point

$$(x, y, z) = (g(z)_2, g(z)_1, f(x)) \ .$$

Nondeterministic dataflow—the Brock-Ackerman anomaly

$C[A_i] =$



Both nondeterministic processes

$$A_1 = O + OIO \qquad \text{and} \qquad A_2 = O + IOO$$

have the same I/O relation, comprising

$$(\varepsilon, O),\ (I, O),\ (I, OO)\ .$$

But

$$C[A_1] = O + OO \qquad \text{and} \qquad C[A_2] = O\ .$$

‘Stable’ spans for nd dataflow

A process with input A and output B:

$$\begin{tikzcd} & E \arrow[dl, "\mathrm{dem}"'] \arrow[dr, "out"] & \\ A & & B \end{tikzcd}$$

where A, B and E are event structures, $out : E \to B$ is a *rigid* map, *i.e.* a function from events E to events B s.t.

$$\begin{tikzcd} x' \arrow[r, phantom, "\subseteq"] \arrow[d, dotted, mapsto] & x \arrow[d, mapsto] \\ y' \arrow[r, phantom, "\subseteq"] & out\, x \end{tikzcd}$$

$$e' \leq e \Rightarrow out(e') \leq out(e), \text{ and}$$

$$\forall x \in \mathcal{C}(E).\ out\, x \in \mathcal{C}(B)\ \&\ \forall e_1, e_2 \in x.\ out(e_1) = out(e_2) \Rightarrow e_1 = e_2\ ,$$

and $\mathrm{dem} : E \to A$ is a *demand* map, *i.e.*, $\mathrm{dem} : \mathcal{C}(E) \to \mathcal{C}(A)$ preserving finite configurations and unions; $\mathrm{dem}[e]$ is minimum input for e to occur.

Deterministic **stable spans correspond to Berry’s stable functions.**

Sequential composition

Spans

$$\begin{tikzcd} & F \arrow[dl, "f"'] \arrow[dr, "out"] & \\ A & & B \end{tikzcd} \qquad \begin{tikzcd} & E \arrow[dl, "\mathrm{dem}"'] \arrow[dr, "g"] & \\ B & & C \end{tikzcd}$$

compose sequentially

$$\begin{tikzcd} & & G \arrow[dl, dashed] \arrow[dr, dashed] & & \\ & F \arrow[dl, "f"'] \arrow[dr, "out"] & & E \arrow[dl, "\mathrm{dem}"'] \arrow[dr, "g"] & \\ A & & B & & C \end{tikzcd}$$

G has events (x, e) s.t. $out\, x = \mathrm{dem}(e)$

A parallel composition

Spans

$$\begin{tikzcd} & E_1 \arrow[dl, "\mathrm{dem}_1"'] \arrow[dr, "out_1"] & \\ A_1 & & B_1 \end{tikzcd} \qquad \begin{tikzcd} & E_2 \arrow[dl, "\mathrm{dem}_2"'] \arrow[dr, "out_2"] & \\ A_2 & & B_2 \end{tikzcd}$$

compose in parallel by juxtaposition:

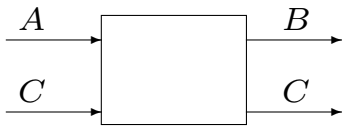
$$\begin{tikzcd} & E_1 \parallel E_2 \arrow[dl, "\mathrm{dem}_1 \| \mathrm{dem}_2"'] \arrow[dr, "out_1 \| out_2"] & \\ A_1 \parallel A_2 & & B_1 \parallel B_2 \end{tikzcd}$$

Feedback

Given

$$\begin{tikzcd} & E \arrow[dl, "\mathrm{dem}"'] \arrow[dr, "out"] & \\ A \parallel C & & B \parallel C \end{tikzcd}$$

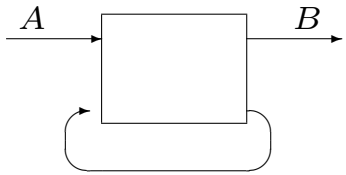
representing



there is

$$\begin{tikzcd} & \mathrm{trace}(E) \arrow[dl, "\mathrm{dem}'"'] \arrow[dr, "out'"] & \\ A & & B \end{tikzcd}$$

representing



Stable spans and profunctors

• A stable span $A \xleftarrow{\mathrm{dem}} E \xrightarrow{out} B$ determines a *profunctor*

$$\widetilde{E} : \mathcal{C}_{\mathit{fin}}(A) \times \mathcal{C}_{\mathit{fin}}(B)^{\mathrm{op}} \to \mathbf{Set} \qquad \text{where}$$

$$\widetilde{E}(p,\, q) = \{x \in \mathcal{C}_{\mathit{fin}}(E) \;\mid\; \mathrm{dem}\, x \subseteq p \;\&\; out\, x = q\} \;.$$

the *set of ways* an I/O pair of finite configurations (p, q) is realized.

• $\rightsquigarrow$ semantics of affine Higher Order Process LAnguage (affine-HOPLA) in *strong correspondence* with an operational semantics.
Event-structure semantics explains its tensor as $\|$ and 'entanglement':

$$(a \parallel a') + (b \parallel b')$$

But affine-HOPLA does not support the usual event-structure semantics of communicating processes.

$\rightsquigarrow$ Working hypothesis: processes are *generalized relations*

spans

$$\begin{tikzcd} & E \arrow[dl, "in"'] \arrow[dr, "out"] & \\ S(A) & & T(B) \end{tikzcd}$$

for monads S, T **up to symmetry** on event structures with basic (rigid) maps, subject to . . .

Leading to

• old things in new ways: nondeterminism in stable domain thy, presheaves represented by event structures with symmetry, game semantics

• new concepts, languages and techniques: event types, event linearity, event induction, concurrent games, quantitative semantics.

Goal: New intensional semantics in which ways of computation correspond to derivations in an operational semantics. Developed in tandem with applications. [ERC Project]

Concurrent games [+ Silvain Rideau]

Basics

Games and strategies are represented by *event structures with polarity*.

The two polarities $+$ and $-$ express the dichotomy:
player/opponent; process/environment; ally/enemy.

An *event structures with polarity* is one in which all events carry a polarity $+/-$, respected by maps.

The *dual*, $E^{\perp}$, of an event structure with polarity E comprises the same underlying event structure but with a reversal of polarities.

A (non-deterministic) concurrent *pre-strategy* in game A is a map $\sigma : S \to A$ of event structures with polarity.

Concurrent pre-strategies

A concurrent *pre-strategy* $\sigma : A \nrightarrow B$ is a total map of event structures with polarity

$$\sigma : S \to A^{\perp} \parallel B\,.$$

It determines a span of event structures with polarity

$$\begin{tikzcd} & S \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\ A^{\perp} & & B \end{tikzcd}$$

where σ_1, σ_2 are *partial* maps of event structures with polarity; one and only one of σ_1, σ_2 is defined on each event of S.

Concurrent copy-cat

Identities on games A are given by copy-cat strategies $\gamma_A : \mathbb{CC}_A \to A^\perp \parallel A$ —strategies for player based on copying the latest moves made by opponent.

$\mathbb{CC}_A$ has the same events, consistency and polarity as $A^\perp \parallel A$ but with causal dependency $\leq_{\mathbb{CC}_A}$ given as the transitive closure of the relation

$$\leq_{A^\perp \parallel A} \cup \{(\overline{c}, c) \ \mid \ c \in A^\perp \parallel A \ \ \& \ \ pol_{A^\perp \parallel A}(c) = +\}$$

where $\overline{c} \leftrightarrow c$ is the natural correspondence between $A^\perp$ and A. The map γ is the identity on the common underlying set of events. Then,

$$x \in \mathcal{C}(\mathbb{CC}_A) \ \text{ iff } \ x \in \mathcal{C}(A^\perp \parallel A) \ \& \ \forall c \in x.\ pol_{A^\perp \parallel A}(c) = + \ \Rightarrow \ \overline{c} \in x\,.$$

Composing pre-strategies

Two pre-strategies $\sigma : A \nrightarrow B$ and $\tau : B \nrightarrow C$ as spans:

$$\begin{tikzcd} & S \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\ A^\perp & & B \end{tikzcd} \qquad \begin{tikzcd} & T \arrow[dl, "\tau_1"'] \arrow[dr, "\tau_2"] & \\ B^\perp & & C \end{tikzcd} .$$

Their composition

$$\begin{tikzcd} & T \circ S \arrow[dl, "(\tau\circ\sigma)_1"'] \arrow[dr, "(\tau\circ\sigma)_2"] & \\ A^\perp & & C \end{tikzcd}$$

where $T \circ S =_{\mathrm{def}} ((S \times T) \restriction \textbf{Syn}) \downarrow \textbf{Vis}$ where ...

Theorem characterizing concurrent strategies

Receptivity $\sigma : S \to A^{\perp} \parallel B$ is *receptive* when $\sigma(x) \subseteq^{-} y$ implies there is a *unique* $x' \in \mathcal{C}(S)$ such that $x \subseteq x'$ & $\sigma(x') = y$.

$$\begin{tikzcd} x \arrow[r, dotted, no head, "\subseteq"] \arrow[d, mapsto] & x' \arrow[d, mapsto, dotted] \\ \sigma(x) \arrow[r, phantom, "\subseteq^{-}"] & y \end{tikzcd}$$

Innocence $\sigma : S \to A^{\perp} \parallel B$ is *innocent* when it is

$+$-*Innocence:* If $s \rightarrowtriangle s'$ & $pol(s) = +$ then $\sigma(s) \rightarrowtriangle \sigma(s')$ and

$-$-*Innocence:* If $s \rightarrowtriangle s'$ & $pol(s') = -$ then $\sigma(s) \rightarrowtriangle \sigma(s')$.

$\rightarrowtriangle$ stands for *immediate* causal dependency.

Theorem Receptivity and innocence are necessary and sufficient for copy-cat to act as identity w.r.t. composition: $\sigma \circ \gamma_A \cong \sigma$ and $\gamma_B \circ \sigma \cong \sigma$ for all $\sigma : A \nrightarrow B$.

Special cases

$\rightsquigarrow$ In general: A *concurrent strategy* is a receptive, innocent pre-strategy.

• Concurrent strategies $\sigma : A \nrightarrow B$ where A and B have purely positive polarities correspond to stable spans.

• Linear sequential algorithms, where prime configurations of the strategy S correspond to the 'schedules' of game semantics [Harmer-Hyland-Melliès LICS'07].

Current problem: In these special cases, strategies can be transformed s.t. composition of strategies can be expressed a the usual composition of spans. I believe this is so in general(?)

Conclusion

The next-generation semantics involves causal models, also becoming important in a range of areas from security, systems, model checking, systems biology, to proof theory.

ERC Project: Events, Causality and Symmetry—the next-generation semantics:

Objective 1 (40%) Intensional semantics: games; strong correspondence with operational semantics; metalanguage(s); higher-dimensional algebra; names

Objective 2 (30%) Event-based reasoning: event types; event induction; causal reasoning; program logics ("Reynolds' conjecture" for conc. sepn. logic); names

Objective 3 (10%) Quantitative reasoning: probabilistic; stochastic; quantum(?)

Objective 4 (20%) Application methods: security; rule-based systems biology; distributed algorithms; extending SOS to causal models