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**

Christmas Lecture MGS 2010, Leicester 15 December 2010
What is a computational process?

Pre 1930’s: An algorithm (informal)

Post 1930’s: An effective partial function  \( f : \mathbb{N} \rightarrow \mathbb{N} \) (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 \( \lambda \)-calculus, a calculus of functions (but is it?):

\[
t ::= x \mid \lambda x. t \mid (t t')
\]

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 \( f : D \rightarrow E \) 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 dl domains as event structures.
Event structures

An event structure comprises \((E, \text{Con}, \leq)\), consisting of a set of events \(E\) - partially ordered by \(\leq\), the causal dependency relation, and - a nonempty family \(\text{Con}\) of finite subsets of \(E\), the consistency relation, which satisfy

\[
\{e' \mid e' \leq e\} \text{ is finite for all } e \in E,
\]
\[
\{e\} \in \text{Con} \text{ for all } e \in E,
\]
\[
Y \subseteq X \in \text{Con} \Rightarrow Y \in \text{Con}, \text{ and}
\]
\[
X \in \text{Con} \& e \leq e' \in X \Rightarrow X \cup \{e\} \in \text{Con}.
\]
Configurations of an event structure

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

**Consistent:** $\forall X \subseteq \text{fin } x. \ X \in \text{Con}$ and

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

For an event $e$ the set $[e] = \text{def } \{ e' \in E \mid e' \leq e \}$ 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, $(C(E), \subseteq)$ is a dl domain (and all such are so obtained).
Event structures as types, e.g., Streams as event structures

conflict (inconsistency) \rightarrow \text{causal dependency} \leq
Simple parallel composition

000 \sim 001 \sim 010 \sim 011 \sim 110 \sim 111

00 \sim 01 \sim \vdots \sim 11

0 \sim 1

aaa \sim aab \sim aba \sim abb \sim bba \sim bbb

aa \sim ab \sim \vdots \sim bb

a \sim b
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 \( f : E \rightarrow E' \) is a partial function on events \( f : E \rightarrow E' \) such that for all \( x \in C(E) \)

\[
fx \in C(E') \quad \text{and} \\
\text{if } e_1, e_2 \in x \text{ and } f(e_1) = f(e_2), \text{ then } e_1 = e_2. \quad ('event linearity')
\]

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

Both nondeterministic processes

\[ A_1 = O + OIO \quad \text{and} \quad A_2 = O + IOO \]

have the same I/O relation, comprising

\[ (\varepsilon, O), (I, O), (I, OO) \] .

But

\[ C[A_1] = O + OO \quad \text{and} \quad C[A_2] = O . \]
'Stable' spans for nd dataflow

A process with input $A$ and output $B$:

$$
\begin{array}{c}
\text{dem} \\
E \\
\text{out} \\
A \quad \quad \quad B
\end{array}
$$

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

$$
e' \leq e \Rightarrow \text{out}(e') \leq \text{out}(e), \text{ and}
\forall x \in C(E). \text{out } x \in C(B) \& \forall e_1, e_2 \in x. \text{out}(e_1) = \text{out}(e_2) \Rightarrow e_1 = e_2 ,
$$

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

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

Spans

compose sequentially

$G$ has events $(x, e)$ s.t. $\text{out } x = \text{dem}(e)$
A parallel composition

Spans

\[
\begin{array}{c}
\text{dem}_1 \quad E_1 \quad \text{out}_1 \\
A_1 \quad B_1
\end{array}
\quad \quad
\begin{array}{c}
\text{dem}_2 \quad E_2 \quad \text{out}_2 \\
A_2 \quad B_2
\end{array}
\]

compose in parallel by juxtaposition:

\[
\begin{array}{c}
\text{dem}_1 \| \text{dem}_2 \quad E_1 \| E_2 \\
A_1 \| A_2 \quad \text{out}_1 \| \text{out}_2 \\
B_1 \| B_2
\end{array}
\]
Feedback

Given

\[
\begin{align*}
\text{dem} & \quad \text{E} & \quad \text{out} \\
A \parallel C & \quad & B \parallel C
\end{align*}
\]

there is

\[
\begin{align*}
\text{dem}' & \quad \text{trace}(E) & \quad \text{out}' \\
A & \quad & B
\end{align*}
\]

representing

\[
\begin{align*}
A & \quad \quad C & \quad \quad B \\
C & \quad & C
\end{align*}
\]

representing

\[
\begin{align*}
A & \quad \quad B \\
A & \quad \quad B
\end{align*}
\]
Stable spans and profunctors

- A stable span \( A \xrightarrow{\text{dem}} E \xrightarrow{\text{out}} B \) determines a profunctor
  \[
  \tilde{E} : \mathcal{C}_{\text{fin}}(A) \times \mathcal{C}_{\text{fin}}(B)^{\text{op}} \to \text{Set}
  \]
  where
  \[
  \tilde{E}(p, q) = \{ x \in \mathcal{C}_{\text{fin}}(E) \mid \text{dem } x \subseteq p \text{ & out } x = q \}.
  \]
  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 \(\parallel\) and ‘entanglement’:
  \[
  (a \parallel a') + (b \parallel b')
  \]
  But affine-HOPLA does not support the usual event-structure semantics of communicating processes.
Working hypothesis: processes are *generalized relations* spans 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* $\sigma : A \rightarrow B$ is a total map of event structures with polarity

$$ \sigma : S \rightarrow A \perp \parallel B. $$

It determines a span of event structures with polarity

$$ \sigma_1 \quad S \quad \sigma_2 $$

$$ A \perp \quad B $$

where $\sigma_1, \sigma_2$ are *partial* maps of event structures with polarity; one and only one of $\sigma_1, \sigma_2$ is defined on each event of $S$. 
Concurrent copy-cat

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

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

$$\leq_{A^\bot \parallel A} \cup \{(\overline{c}, c) \mid c \in A^\bot \parallel A \land \text{pol}_{A^\bot \parallel A}(c) = +\}$$

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

$$x \in C(\mathbb{CC}_A) \text{ iff } x \in C(A^\bot \parallel A) \land \forall c \in x. \text{pol}_{A^\bot \parallel A}(c) = + \Rightarrow \overline{c} \in x.$$
Composing pre-strategies

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

$$
\begin{array}{cccc}
\sigma_1 & S & \sigma_2 & T \\
A^\perp & & B & & B^\perp & & C \\
\end{array}
$$

Their composition

$$
\begin{array}{cccc}
(\tau \circ \sigma)_1 & T \circ S & (\tau \circ \sigma)_2 \\
A^\perp & & C \\
\end{array}
$$

where $T \circ S =_{\text{def}} ((S \times T) \upharpoonright \text{Syn}) \downarrow \text{Vis}$ where ...
Theorem characterizing concurrent strategies

**Receptivity** $\sigma : S \rightarrow A^\perp \parallel B$ is receptive when $\sigma(x) \subseteq^\rightarrow y$ implies there is a unique $x' \in C(S)$ such that $x \subseteq x' \land \sigma(x') = y$.

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

+ -Innocence: If $s \rightarrow s'$ & $\text{pol}(s) = +$ then $\sigma(s) \rightarrow \sigma(s')$ and

- -Innocence: If $s \rightarrow s'$ & $\text{pol}(s') = -$ then $\sigma(s) \rightarrow \sigma(s')$.

$\rightarrow$ 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 $\sigma : A \rightarrow B$. 
In general: A concurrent strategy is a receptive, innocent pre-strategy.

- Concurrent strategies $\sigma : A \leftrightarrow B$ 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 as 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