Section 1d: Extended Synopsis

State-of-the-art and objectives

Headed by Principal Investigator Glynn Winskel, the project ECSYM assembles a world-leading team of theoretical computer scientists and mathematicians. Its goal: to build the next generation of semantics of computation, a new mathematical foundation with which to understand and analyze computation of the complexity we begin to see today.

Why? At present there are three main methodologies used in representing and analyzing the behaviour of computer programs and processes, all of them with their advantages, but all with their inadequacies and problems:

• *Denotational semantics and domain theory.* This has provided a global mathematical setting for sequential computation, and thereby placed programming languages in connection with each other. It connects with the mathematical worlds of algebra, topology and logic and has inspired new programming languages, type disciplines and methods of reasoning. But today it has become clear that many aspects of computation do not fit within the traditional framework of denotational semantics and domain theory—see Sec 1.2. It has abstracted away from operational concerns too early.

• *Structural operational semantics.* This method has become very popular and is based on syntax-directed rule-based inductive definitions to specify the evaluation and execution of programs. It is easy to use and flexible, but presently underdeveloped mathematically, lacking in algebraic techniques, and gives very little guidance about the space of possibilities and variations around the specific language dealt with. It generally represents parallelism through a global nondeterministic interleaving of actions, which obscures the local causal dependencies of the underlying events. Such local dependencies often play a key role in the design and analysis of computing systems.

• *Causal models.* These include Petri nets and event structures which represent systems in terms of the events associated with their behaviour together with relations expressing the local dependencies and conflicts between events. They have become much more prevalent in recent years, often being rediscovered as *the* natural model in a variety of contexts from security protocols to systems biology and the structure of proof—see Sec 1.1. But they lack a comprehensive theory or even a methodology with which systematically to provide semantics to a broad range of programming languages and systems. Their breadth of application is not yet widely appreciated. It is hampered by anomalies within causal models—see Sec 1.1. These are partly due to their overly-concrete nature, which blocks their algebraic development, and to the conceptual puzzle that in *ad hoc* uses of causal models they are used *both* to represent data types as well as the process of computation—a mismatch with traditional denotational semantics.

There are crying needs for a new semantics which
• extends the methodology of denotational semantics and domain theory to the challenges in analyzing computation today;
• maintains clean algebraic structure and abstraction alongside the operational nature of computation;
• provides a comprehensive theory of causal models.
As we will see, a solution to any one issue requires solutions to the other two. The solutions bear with them methods to incorporate quantitative reasoning and a range of potential applications, touching on other sciences, which will motivate and help guide the mathematics—see Sec 3.

How? If semantics is to encompass operational concerns it has to be an *intensional* theory, capturing the *ways* in which computation proceeds and not merely input and output. The evidence for this comes from: sequential programming, where *e.g.* game semantics informs operational semantics; interactive/distributed computation, where *e.g.* analysis of security protocols often relies on clever encryption to ensure desired event dependencies; from anomalies, as in nondeterministic dataflow, where intensionality is forced in order to achieve a compositional semantics. This intensionality immediately sets the next generation of semantics apart from traditional domain theory and denotational semantics. Once achieved it would make the current distinctions between operational and denotational semantics disappear; a denotation would carry its operational semantics.

The form the new semantics should take is based on recent discoveries, and is explained over the next few pages. They point the way to an event-based theory to describe ways of computation in terms of patterns of events. A core insight is the increased expressivity a formal treatment of behavioural symmetry brings to causal models, to the types, processes, operations and applications they can support. Through the

key elements of events, causality and symmetry there is a clear, though challenging, way forward to a next-generation semantics, which combines: the mathematical richness of domain theory; a comprehensive theory of causal models; structured operational semantics.

What? The objectives of ECSYM are:

Objective 1. A comprehensive semantic theory—one which includes that of causal models—together with rich metalanguage(s) and structured operational semantics, extracted from the denotational semantics;

Objective 2. New techniques for event-based, causal reasoning, the beginnings of which are suggested by the mathematics;

Objective 3. To incorporate quantitative reasoning through enrichment with probability and time—the mathematics and applications suggest a way;

Objective 4. To develop application methods, especially where causal models (have the potential to) play a central role, including distributed and parallel computation, and systems biology.

We shall expand on the nature of the mathematics, the objectives' feasibility and timeliness.

Why this PI and this team? The PI, Winskel, is a recognized world leader in semantics of computation (*e.g.* his book is the major semantics text in europe, US and Asia). He pioneered the theory of event structures, the categorical view of processes (which became an influential OUP handbook chapter) and recent extensions on which this proposal rests. He would be backed up by an ECSYM team consisting of world-class senior researchers Marcelo Fiore (leader in the application of sheaves in CS); Martin Hyland (coinventor of game semantics and international expert on category theory and logic); Andrew Pitts (pioneer in techniques for operational semantics); and outstanding junior researchers: Richard Garner (higher-dimensional algebra, type theories); Jonathan Hayman (Petri nets, separation logic, tools); Chung-Kil Hur (equational reasoning, verification,tools, theorem proving); Sam Staton (name-generation, operational semantics, security protocols). Together they share a formidable expertise and beyond this an enthusiasm to push into new applications. A project of this ambition requires funding at the ERC level, above that possible from UK sources. Currently none of the junior researchers have future funding, and in the current UK economic climate there is the very real risk of losing the concentration of expertise this project requires.

1 Background

1.1 Causal models

One reason why ECSYM is especially timely is the current rebirth of interest in causal models over a diversity of applications: *security protocols*; *systems biology*; *asynchronous hardware; types and proof; nondeterministic dataflow; network diagnostics; concurrent separation logic; partial order model checking; distributed and parallel computation.*

Work of the PI plays a significant role in all the above areas. For instance, several of the areas make use of the unfolding of a 'safe' Petri net into its occurrence net, and from there to an event structure, a construction (due to Nielsen, Plotkin and the PI) as fundamental as the better-known unfolding of a transition system to a tree. The simplest form of event structure arises by abstracting away the conditions in an occurrence net and capturing their effect in relations of causal dependency and conflict between event occurrences. Event structures are used, for instance, in the analysis of multicore memory, their extension with name generation in security protocols, and with probability in network diagnostics and Bayesian models of trust in distributed systems.

The relations between the different forms of causal models, and classical models such as transition systems, are well understood. The PI's introduction of maps to the models led to relations between models being expressed by adjunctions, with the algebraic benefits these entail. In particular, there is an algebraic characterization of the net unfolding as a universal construction. Its algebraic properties play a key role, for example in network diagnostics, in correctly combining local diagnoses at components.

Problems with traditional causal models

Unfoldings of general Petri nets: While occurrence net unfoldings can be defined for all Petri nets, where conditions can hold with multiplicities, there can be no universal characterisation like that for the unfolding of safe nets; symmetry intrinsic to nets with multiplicities spoils an essential uniqueness property.

Weak bisimulation, an important process equivalence: Just as for traditional models weak bisimulation between causal models (abstracting from invisible actions) can be explained as strong bisimulation between the results of 'hiding' the invisible actions. Whereas the 'hiding' operation on a transition system is again

a transition system, the hiding operation *e.g.* on event structures does not always yield an event structure.

Name generation: There are methods to represent the generation of new names in causal models, but the methods ignore the implicit symmetry on names. Presently causal models lack a key construction, a form of new-name abstraction.

Varying maps: Recently a need for several different forms of maps on causal models has become clear. Changing the maps generally changes important categorical constructions. One would like to settle on some basic maps and then have a systematic way to vary the nature of maps.

Higher-order processes: causal models, as used traditionally, do not represent general higher-order processes (higher-order in the sense that they could treat processes themselves as input and output values).

It is surprising, but *all* these anomalies stem from ignoring the symmetry intrinsic to the constructions needed.

1.2 Domain theory

In the earliest days of computer science it became accepted that a computation was essentially an (effective) partial function between the natural numbers. As computer science matured it demanded increasingly sophisticated representations of processes. The pioneering work of Strachey and Scott in the denotational semantics of programs assumed a view of a process still as a function but now acting in a continuous fashion between datatypes represented as special information orders, 'domains'; reflecting the fact that computers can act on conceptually-infinite objects, but only by virtue of their finite approximations.

What is the information order in domains? There are essentially two answers: the '*topological*,' the most well-known, from Scott's work; and the '*temporal*,' from the work of Berry on 'stable domain theory,' where more information reflects the occurrence of more events—this reading, based on the realization that Berry's domains were precisely those domains represented by event structures is due to the PI.

Problems with domain theory

Nondeterminism: For traditional ('topological') domain theory the problem of adjoining nondeterminism was solved by Plotkin through the introduction of powerdomains. But for stable domain theory the information orders of all but the simplest powerdomain construction fail to be temporal.

Concurrency/interaction: The intricacy of models for distributed computation means that they don't fit within information orders. Their intricacy suggests that they belong to an extended notion of domain as a category. Only rarely do the wanted equivalences on processes arise from traditional domain theory.

Probability and *nondeterminism:* Combining probability and nondeterminism is problematic because the two forms of powerdomain together do not satisfy a distributive law (their combination forces extra laws). However, with the intensional *indexed probabilistic powerdomain* where probability is carried by the *ways* values are computed, one recovers a distributive law.

Nondeterministic dataflow: While deterministic dataflow is a shining use of simple domain theory, nondeterministic dataflow is beyond its scope. A compositional account needs generalized relations which specify the *ways* input-output pairs are realized.

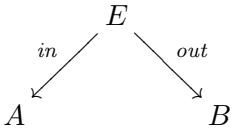
2 Methodology

Anomalies in traditional domain theory lead us to seek a more *intensional* theory of computation, concerned not just with the value computed but also with the *ways* the value is computed.

This makes certain generalized relations called profunctors unavoidable. Profunctors are generalizations to categories of the set-theoretic idea of relations. The key idea is that the usual truth values, true and false, are replaced by sets—an individual set, now a form of generalized truth value, specifying the *set of ways* an output is produced from input.

The rich mathematical structure enjoyed by profunctors led to new powerful higher-order process languages. In analyzing the operational content of the profunctor semantics, came a surprising discovery, that the sets of ways intrinsic to the semantics could often be understood as the states of a causal model—an event structure. Process constructions which, although natural mathematically, were puzzling in the profunctor semantics suddenly obtained an operational reading.

Simultaneously, the curious dual role of causal models, both to represent data and computation, became explained. A computation between datatypes could now be seen as a *stable span* of event structures,

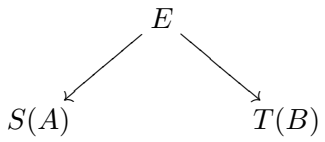


comprising an event structure E (the computational process) and a pair of maps in and out, specifying at what input (in event structure A, the input datatype) and in which manner the output (in event structure B, the output datatype) is produced; technically, the maps

in is 'demand' map and out a 'rigid' map of event structures. Stable spans were shown to be generalizations of *stable functions* (central to stable domain theory) and to have been used, rather implicitly, in the semantics of nondeterministic dataflow. This extension of stable functions with nondeterminism supports a compositional semantics of nondeterministic dataflow.

This is the start of a key idea, processes as spans. But stable spans are insufficient in various ways; for example, because output maps of stable spans are rigid, stable spans are too restrictive to support a broad range of parallel compositions without resorting to interleaving. What is required is a systematic method to vary the nature of the maps in and out. A systematic way to modify maps is through monads; one can obtain other kinds maps from an object E to an object B out of rigid maps from E to $T(B)$, where T is an appropriate monad. This suggests that stable spans be generalized to *general spans* of event structures

General span



with input the event structure A, output B and process E, w.r.t. suitable monads S and T to moderate the regimes of input and output. (More should hold for the spans to compose.)

But here we quickly run up against the overly-concrete nature of traditional causal models. It soon becomes clear that many wished-for monads do not exist—the monad laws do not hold on the nose. But the laws do hold up to an intrinsic symmetry. This observation is also the key to other anomalies in causal models—see Sec 1.1. Their common solution: a formal treatment of symmetry in process behaviour.

2.1 Symmetry and its consequences

The treatment of symmetry on models makes use of a general method of open maps in defining bisimulation in a variety of models. Briefly, a symmetry in an object (be it an event structure, Petri net or some other model) is expressed as a bisimulation equivalence (given as a span of open maps) that says when computation paths are similar according to the symmetry. It is now sensible to consider whether maps, which must preserve symmetry, are equal *up to symmetry*. It is a surprising fact that this feature considerably enhances the mathematical theory and potential application areas of causal models.

The introduction of symmetry settles the anomalies of causal models: there is now a universal characterisation of unfoldings of *general* Petri nets, now with uniqueness up to symmetry; operations for weak bisimulation and name generation can now be shown expressible on event structures with symmetry.

But of more general importance, many wished-for monads are indeed monads *up to symmetry*, and many more, useful monads are undoubtedly waiting to be discovered. For example, the monad for demand maps creates events in the form of 'input histories,' describing the way that input is explored. This opens the way to a robust notion of process as a general span of event structures with symmetry. General spans provide semantics to potentially rich process languages and event types, and support case analysis on events, a form of 'event induction,' in definition and reasoning. One such higher-order language can induce the usual event-structure semantics for Milner's CCS—here CCS parallel composition appears as a higher-order process taking pairs of event structures with symmetry to their parallel composition. In particular, the stable spans used in the semantics of nondeterministic dataflow can be realized as particular general spans, with only one monad to modify the input map.

3 Research plan

The situation is challenging. It is analogous to that in the late sixties, when there was a clear idea of how to model programs—as continuous functions—but where the development of denotational semantics and domain theory lay ahead. But not quite. We have that history and the considerable developments in CS to inform us. That suggests we seek a general metalanguage, and accompanying operational semantics and logic, based on the mathematical semantics. The way forward is to develop the mathematics in tandem with applications and examples—a source of guidance and inspiration. Here we are fortunate in causal models being the focal point of so many different applications.

Mathematical objectives

(1) Metalanguage(s) and strong correspondence: The goal is to read the operational semantics directly from the intensional semantics. A key vehicle will be high-level syntax for general spans and associated types (generalizing Moggi's monadic metalanguage, which has been very successful in a more limited scenario of traditional domain theory and functional programming). We expect to lift to general spans earlier results of Nygaard and the PI on the 'strong correspondence' between derivations in an operational semantics and elements of denotations as generalized relations. We can exploit the PI's recent discovery that basic (so probably all) game semantics fits within spans of event structures. The combinatorially-defined

schedules of Harmer, Hyland and Melliès, important for abstract machines, now appear automatically as prime configurations of an event structure denoting the strategy. There are important sub-projects on: game semantics via spans; mechanisms for name generation; the algebra of operational methods; and the underlying mathematics.

(2) Event-based reasoning: Event types express not just the type of events a process can perform but also constraints of causality, linearity and atomicity. They suggest specification logics by analogy with existing logic for domains and how the verification of such properties might be reduced to type-checking. Equational theories of equivalences, some of them only now supported within causal models through the addition of symmetry, will play an essential role in reasoning, in counterbalancing the inbuilt intensionality of the models. Through a causal semantics of concurrent separation logic (that by Hayman and the PI, or a development), we are in a strong position to settle an old conjecture of John Reynolds on robustness of the logic under command-refinement.

(3) Quantitative semantics: The use of spans provides, relatively unexplored, methods to represent types and computation with probabilistic, stochastic and even quantum behaviour. For example, spans can be enriched with probability, essentially by taking the vertex in a span of event structures to be a probabilistic event structure. The probabilistic event structure expresses both the ways, and with which probability, output is obtained. The development will be guided by uses of probabilistic event structures in security, Bayesian models of trust and network diagnostics, and by the appearance of stochastic event structures in systems biology. There are subprojects on: combining probability and nondeterminism; stochastic event structures; and quantum event structures, where (concurrent) events label (commuting) projectors on Hilbert space.

(4) Application methods The new semantics opens up a new landscape of models and leads to the prospect of new methods, descriptive and analytic, in a range of applications. At the same time, its development will rely on being tested on applications.

Petri-net SOS: It is planned to design and promote an accessible structural operational semantics based on Petri nets (with symmetry), to update the traditional and highly-influential techniques of structural operational semantics (SOS) based on transition systems. The idea is to replace the rule-based inductive definition of configurations and transitions in SOS by rule-based definitions of conditions and events. A planned, short manual illustrating this on a range of applications is likely to have a broad, lasting impact.

Distributed and parallel computation: The introduction of symmetry to event structures opens up a new landscape of models in which event structures both stand for types and the process of computation between them. Event types can express local causal constraints and symmetries, and provide semantics to name generation. This calls out for experiments in the semantics and analysis of distributed and parallel algorithms, where traditional formal analysis has not been in a position to exploit causal reasoning. One rich area for ECSYM is that of security protocols, where the PI has experience in reasoning through causal relations. Another ripe and important area—where there is considerable local expertise—is the analysis of multicore memory, recent verification of which uses a form of event structure. A positive solution to Reynolds' conjecture would, for instance, have an immediate impact here as conditions of race-freedom play a key role in the verification of multicore memory. The treatment of symmetry extends unfolding techniques and tools to general Petri nets, often used to describe distributed algorithms.

Systems biology: Systems biology provides new challenges to semantics, puts into action ideas from causal models, symmetry, as well as stochastic and probabilistic event structures. Biochemical reactions are by nature determined in a local fashion and so fit concepts from causal models. By invoking ideas of independence and conflict from causal models stochastic simulations can to an important degree rely on an accumulation of local updates to control the state explosion, as is done in the kappa-system, developed in Harvard, Edinburgh and Paris. Symmetry plays several important roles: in determining the stochastic rates of rules; in reductions via abstract interpretation; and in the analysis of biochemical pathways. Once account is taken of independence, rule-based simulations generate (stochastic) event structures, with symmetry induced by the similarity of molecules of the same species. One might expect that the event structures described the biochemical pathways recognised by biologists. But the requirements of biologists drove kappa to a much more compressed account of the pathways. On a recent visit to Harvard the PI provided a mathematical rationale for the 'compression algorithms' of kappa. The solution introduces maps to express local state, evolution and symmetry of biochemical systems. We need to push this mathematical analysis into stochastic simulation and further state-reduction methods exploiting symmetry.