Extensional Semantics of Program Behaviour: Case for Support

Introduction

Game semantics is a way of characterizing programming languages and logical calculi *intensionally* by interpreting proofs or programs as strategies for interacting with the environment; this process of interaction can be thought of as playing a two-person game. Games can capture precisely the behaviour of higher-order programs in sequential languages, in a fashion which is very difficult or impossible for traditional forms of denotational semantics. This precision is known formally as *full abstraction* or *full completeness*.

A remarkable feature of game semantics is its flexibility: games have been used to give fully abstract or fully complete models not only of complete logics such as linear logic [4] and functional languages such as PCF [5, 17, 29], but also languages with powerful imperative features such as references [2, 3], continuations [18] and exceptions [19]. As the subject matures, games are being applied successfully to discover new techniques for reasoning about programming languages, such as *algorithmic game semantics*, initiated by Ghica and McCusker [15].

Barriers to the adoption of games as a general theory of program semantics still exist, such as the difficulty of reasoning mathematically about games and strategies (which leads to difficulty in using game semantics to reason about programs) and the rather technical and specific nature of some of the constructions used. Moreover, the notion of information flow implicit in most games models is rather specific; interaction is represented as a sequence of actions, rendering it difficult to represent non-deterministic languages in the absence of state, for example.

We may contrast this situation with that of *domain theory*, in which programs are modelled *extensionally* using sets and functions, and which has formed the basis for much research in denotational semantics by virtue of its conceptual simplicity. Domains now form a wide-ranging mathematical theory which is currently being used in the study of new paradigms such as exact (real number) computation [14]. The search for order-theoretic characterizations of higher-order sequentiality has led to the discovery of several candidate notions, such as stability [7] and strong stability [9].

However, a significant obstacle to the use of domains to reason about programming languages has been a lack of accurate models of imperative and concurrent features of the sort provided by game semantics. For example, typical domain-based models of mutable variables are based on a *global state*, a function recording what values are stored in each memory location. This corresponds closely to intuition about how variables are implemented, but such models fail to be fully abstract; on the other hand, the fully abstract games models do not employ any notion of state explicitly, and this is vital to their precision.

date to

Thus it seems likely that a more profound understanding of the relationship between domain theory and game semantics would be mutually beneficial. On the one hand, the insights gained by the exciting discoveries in game semantics in the area of computational effects could be used to construct and study domain theoretic models in a way which retained the advantages of domain theory, while enjoying the precision afforded by games. On the other, if games can be represented as "sets with structure" and strategies as functions, then this opens the possibility of discovering new kinds of game semantics, perhaps embodying different notions of sequentiality, and new ways of reasoning about games and strategies (for instance, equational reasoning about strategies using simple algebraic manipulations). This is the main objective of this research project; to capture *intensional* aspects of program behaviour using *extensional* models.

Objectives

The aim of this project is to develop a rich semantic theory based on games and domains and the connections between them. Specifically:

- by importing the insights of game semantics into domain theory, to discover fully abstract domain-theoretic models of programming languages incorporating state, local control, nondeterminism and concurrency;
- by constructing games models corresponding to different kinds of domain, to discover fully abstract and effectively presentable game-based models of the lazy -calculus and functional languages with nondeterminism;
- by analyzing the common structure of the domain and game-theoretic approaches, to extract a range of techniques for constructing and reasoning about such models.

Programme and Methodology

A possible basis for connecting the intensional world of games with the extensional world of domains is the remarkable observation of Cartwright and Felleisen [10] that intensional properties can become extensional in the presence of of errors; in particular adding errors to PCF allows the order of evaluation of functions to be observed. Using this insight, they have constructed a fully abstract and effectively presentable model of SPCF (PCF with catch and errors) using *observably sequential functions*. With Curien [11], they have also shown that this is essentially equivalent to a model of SPCF in the category of *sequential data structures and sequential algorithms* (and Curien has shown that the latter category can be presented in a style which makes apparent its closeness to game semantics [12]).

However, one can argue that Cartwright and Felleisen's model is not extensional in character because functions are supplied together with intensional information about how they are to be computed.

A route to a truly extensional characterization of observable sequentiality is has been proposed by by Laird [23, 22]. He has observed that Berry's notion of *bidomain* [7] can be adapted to give universal models of a simple hierarchy of -calculi with constants and , and some combination of the operators (sequential composition) and (parallel composition) (see Fig. 1). The lower three vertices may be thought of as each representing different notions of sequentiality.

- ( ) (which is "unary PCF") is sequential with respect to but not ; it has a universal model in the original category of bidomains [23].
- ( ) is sequential with respect to but not ; its universal bidomain model is obtained by replacing stability with co-stability, its order-theoretic dual.
- is sequential with respect to and . Its universal bidomain model is based on a new notion, the *bistable* order [22], which is essentially the intersection of stability and co-stability.

provides the connection to observably sequential functions, sequential algorithms and games; it is a version of SPCF in which the ground type has no values and a single error , and so it has a universal sequential algorithms model which is isomorphic to the bistable semantics.

The research proposed here has its basis in an exploration of this diamond from both domain theoretic and game semantic directions. We have bistable and (games-style) sequential algorithms models corresponding to the bottom vertex, so the questions here are: how can we relate the two, and how can we use that relationship to transfer techniques and results from one to the other? For the other vertices we have no directly corresponding games models, so the question here is: can we construct such models and carry out a similar programme?

As simple languages which express different but connected notions of higher-order computation, the four -calculi form a useful conceptual basis for our investigation. However, our main objective is to study more expressive languages, and in particular, features such as control, state and concurrency. Some techniques for achieving this are already known. For example, we can construct models of call-by-name and call-by-value languages with

(,)

( ) ( )

Figure 1: A hierarchy of simple functional calculi

datatypes (SPCF is a simple example) using *continuation-passing-style* (CPS) interpretations into . In the case of SPCF, this corresponds to a direct game semantics interpretation [18]. Another example is the use of lifting and recursive types to construct a fullly abstract bidomain model of the lazy -calculus, based on the model of unary PCF [23].

We shall develop and use a range of further methods based on domain theory, category theory, linear logic and realizability for constructing and refining models based on game semantics. Indeed, we anticipate that these methods would constitute a significant element of the contribution that the project could make to the understanding of programming languages, potentially forming the basis of reasoning techniques such as metalanguages and equational theories, new type-systems and model-checking techniques.

This research is divided into three phases: a foundational stage in which we will establish a basic correspondence between games and domains; a stage in which we will use this correspondence to construct and study domain-theoretic models of imperative features; and a final stage in which we will explore new kinds of games models corresponding to existing domains.

Phase One: Foundations

The first stage of the research will be to lay the foundations for the remainder of the programme by a detailed study of the underlying notions — stability, bistability and bidomains — from a domain theoretic and game semantic perspective.

Starting from the universal and fully abstract model of , we will develop the domain theory of bistable bidomains. Using established methods, we will explore constructions such as recursive types. We shall also seek to clarify the relationship between bistable functions and sequential algorithms. It is clear from the relevant full abstraction results that sequential algorithms over sequential data structures with a single error behave as bistable functions (and this has been confirmed by communication with Curien) and thus the Cartesian closed category of sequential algorithms embeds in the category of bistable bidomains. This leaves open the question of what the image of this embedding is; it appears that it may be characterized by some algebraicity conditions.

This leads naturally to a further task, constructing bistable models of linear logic, by recovering the linear logic structure of sequential algorithms [12] in terms of bistability. Thus we will seek a decomposition of a Cartesian closed category of bistable bidomains with a suitable algebraic structure into a model of intuitionistic linear logic. Examination of sequential algorithms suggests that this can be achieved by considering functions which are linear in the stable sense. Games can also be used to construct models of polarized MALL [25]; emulating this with bidomains could capture the Player/Opponent duality which is fundamental to game semantics. We will also examine the possibility of constructing *fully complete* models of linear logic, either by further structure-preservation properties, or by realizability constructions.

Phase Two: From games to domains

Having reconstructed basic categories of games as categories of bidomains, we will then attempt to emulate some of the extensions and refinements to the games which give rise to fully abstract models. For each language we

may identify two main tasks; firstly, to discover a way to construct sound models, and then to identify structure corresponding to definability and/or observational equivalence in these models. For example, many of the full abstraction results for games models have been achieved by identifying constraints on strategies which correspond precisely to definability, and we shall investigate the possibilities of characterizing these constraints as preservation properties for functions.

Imperative Features One of the most significant achievements of game semantics is the description of fully abstract models of functional languages with imperative features such as the locally bound integer references of Idealized Algol [2]. One of our main objectives will be to give domain theoretic analogues of this semantics.

Idealized Algol is not an extensional language — we cannot simply model programs of type $A \quad B$ as partial functions from values in A to values in B (the number of uses of each argument must also be recorded, for example). However, two recent developments in our research suggest that we can interpret programs at arrow type as functions which take a *stream* of values of argument type to a value of result type.

Laird has devised a new kind of categorical model for functional languages with state-like features (such as first-order and higher order references) based on the notion of a *sequoidal category*. This is a category with a non-commutative and non-associative bifunctor (a sequoid), which may still possess interesting coherence and closure properties. The asymmetric nature of the sequoid is used to represent temporal priority, which would seem to be a prerequisite for a precise model of state. Categorical notions of sound and fully abstract models of Idealized Algol, and an idealization of Scheme have been developed using these ideas [20]. The category of sequential algorithms possesses a sequoidal structure, which we will seek to identify in its extensional presentations. Thus far, the studied examples of sequoidal categories which give rise to models of store have been games models, so the true value of these concepts will become apparent when they are applied to domain theoretical constructions.

McCusker has shown that a stable model of *Basic SCI* (a variant of Idealized Algol with control over interference), due to Reddy [30], is fully abstract [28], although it is not universal. It appears that this model may be characterized using similar constructions, and thus we may investigate its relation to a bistable bidomain model of Idealized Algol.

Control As noted above, models of languages with first-class continuations can be constructed from models of in a way which preserves full abstraction. A more difficult task is to identify the elements of these models which exhibit more restricted forms of control flow. This is an important problem not simply because it is part of the search for fully abstract models, but also because of its relevance to the study of secure information flow. (Games semantics is already being applied in this area [27, 26].) We have two ways to approach this problem; in game semantics, local control flow is represented using the "bracketing condition" — a kind of stack discipline for moves — and we will endeavour to formulate an analogous condition as a preservation property in bistable models of imperative languages. A more general technique for constructing models of control flow using linear types may be derived from the *linear* CPS interpretation [6] (which has also been applied to secure information flow [31]). Thus we may use our work on the linear decomposition of bidomains to obtain a semantics via this construction. Laird has described a game semantics of linear CPS [24] and how it relates closely to well-bracketing; domains may give another perspective.

Non-Determinism and Concurrency Games offer a natural account of non-deterministic behaviour in imperative programs by abandoning the determinacy condition on strategies; Harmer and McCusker [16] have given a game semantics of Idealized Algol with erratic choice which is fully abstract with respect to may and must convergence. The duality between and which is central to bistability would appear to offer a good account of the duality between may and must (for example, there is an obvious operational interpretation of the and operators in ( ) and ( ) as non-deterministic choice, the bidomain semantics is then fully abstract with respect to must convergence and its dual with respect to may convergence). We will aim for similar results for the domain-theoretic models of imperative features, and to connect them to the various powerdomain semantics of non-determinism.

The main objective of our study of non-determinism will be to use it, along with interpretations of state to study bidomain models of concurrent languages such as Idealized Parallel Algol [8]. This will be based

on an interleaving semantics, which has a clear game-semantic basis, but we will need to develop domain-theoretic or category theoretic methods to describe this. Our goal will be to obtain a full abstraction result with respect to may and must testing.

Nondeterminism is also the scene of a notable failure of game semantics to date: while it is relatively straightforward to give accurate models of languages with both state and nondeterminism, no good games model for a nondeterministic functional language is known. This is one of the problems which will be addressed in the final phase of the project.

Phase Three: New Directions

In the final phase, we will investigate how the intensional properties of functions in some pre-existing versions of domain theory can be determined and specified in game-semantic style. This phase can be viewed initially as an exploration of the game semantics possibilities of the upper three vertices on the diamond in Fig. 1. We anticipate that the insights generated in the foundational phase will help in the association of sequential functions to strategies, whilst the methods devised in the phase two for characterizing models of non-functional features may also be applied to both the game and domain models studied in this theme.

Our starting point will be an attempt to construct games models of unary PCF which are fully abstract without any observational quotient. To do so, it would seem likely that we will need a new representation of the interactions which make up a game, other than the usual form of sequences of moves (as it is not possible, for example, to observe the order in which the operator is evaluated). From our semantics of unary PCF we can expect to derive fully abstract models of languages such as the lazy -calculus and call-by-value -calculus, as the arguments which allow this to be done in the case of bidomains are quite general. We will study the relationship between these models (and their bidomain equivalents) and (the unary fragment of) the original games models of PCF [5, 17] and the lazy -calculus [1]. This will presumably take the form of an extensional collapse, but it would be interesting to characterize it in a more concrete way, as was done for the games and sequential algorithms models of in [21]. We will also seek a semantics of the dual calculus to ( ) — that is, ( ) — along similar lines, as by using both we can obtain (by CPS interpretation) models of languages such as SPCF augmented with a non-deterministic choice operator which are fully abstract with respect to may and must testing.

A "linear decomposition" of bidomains already exists in the form of bistructures [13]. We will explore the possibility of constructing a games model which corresponds to at least a sequential fragment of this, and investigate whether there is another linear decomposition of bidomains.

Beneficiaries

This is fundamental research, in an area in which the UK plays a leading role. Its long-term objective is to improve understanding of some very subtle features of higher-order programming languages, leading to more accurate ways of reasoning about properties such as correctness, security and optimization. Its immediate impact will be within the international research community, where we anticipate that a diverse range of researchers will be able to make use of it to develop techniques for reasoning about these properties.

As we have observed, game semantics is already being used as the basis for model checking techniques for imperative languages [15], as well as in program analysis of control-flow and secure information flow [27, 26]. We anticipate that our research will give further impetus to efforts such as these, by abstracting some of the key properties of games and thus making them less dependent on the technical details of individual models.

Our research will make accurate models of functional-imperative computation accessible without requiring expertise in game semantics, as we aim to preserve the conceptual simplicity of domain theory. The latter is currently being used in a range of research into new computational paradigms. Our results will broaden the applicability of some of this research to a more realistic range of language features.

References

[1] S. Abramsky and G. McCusker. Games and full abstraction for the lazy -calculus. In *Proceedings of the Tenth Annual IEEE Symposium on Logic in Computer Science*, pages 234–243. IEEE Computer Society Press, 1995.

[2] S. Abramsky and G. McCusker. Linearity, Sharing and State: a fully abstract game semantics for Idealized Algol with active expressions. In P. O'Hearn and R. Tennent, editors, *Algol-like languages*. Birkhauser, 1997.

[3] S. Abramsky, K. Honda, G. McCusker. A fully abstract games semantics for general references. In *Proceedings of the 13th Annual Symposium on Logic In Computer Science, LICS '98*, 1998.

[4] S. Abramsky, R. Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59:543–574, 1994.

[5] S. Abramsky, R. Jagadeesan and P. Malacaria. Full abstraction for PCF. *Information and Computation*, 163:409–470, 2000.

[6] J. Berdine, P. O'Hearn, U. Reddy, and H. Thielecke. Linear continuation-passing. *Higher Order Symbolic Computation*, 15(2/3):181–208, Sept. 2002.

[7] G. Berry. Stable models of typed -calculi. In *Proceedings of the 5th International Colloquium on Automata, Languages and Programming*, number 62 in LNCS, pages 72–89. Springer, 1978.

[8] S. Brookes. The essence of Parallel Algol. In *Proceedings of LICS '96*, 1996.

[9] A. Bucciarelli and T. Ehrhard. A theory of sequentiality. *Theoretical Computer Science*, 113:273–292, 1993.

[10] R. Cartwright and M. Felleisen. Observable sequentiality and full abstraction. In *Proceedings of POPL '92*, 1992.

[11] R. Cartwright, P.-L. Curien and M. Felleisen. Fully abstract semantics for observably sequential languages. *Information and Computation*, 1994.

[12] P.-L. Curien. On the symmetry of sequentiality. In *Mathematical Foundations of Computer Science*, number 802 in LNCS. Springer, 1993.

[13] P.-L. Curien, G. Winskell, and G. Plotkin. Bistructures, bidomains and linear logic. In *Milner Festschrift*. MIT Press, 1997.

[14] A. Edalat and P. Sünderhauf. A domain-theoretic approach to real-number computation. *Theoretical Computer Science*, 210:73 – 98, 1998.

[15] D. Ghica and G. McCusker. Reasoning about Idealised Algol using regular languages. In *Proceedings of the Twenty-Seventh International Colloquium on Automata, Languages and Programming (ICALP 2000)*, 2000.

[16] R. Harmer and G. McCusker. A fully abstract games semantics for finite non-determinism. In *Proceedings of the Fourteenth Annual Symposium on Logic in Computer Science, LICS '99*. IEEE Computer Society Press, 1998.

[17] J. M. E. Hyland and C.-H. L. Ong. On full abstraction for PCF: I, II and III. *Information and Computation*, 163:285–408, 2000.

[18] J. Laird. *A Semantic Analysis of Control*. PhD thesis, Department of Computer Science, University of Edinburgh, 1998.

[19] J. Laird. A fully abstract game semantics of local exceptions. In *Proceedings of the Sixteenth International Symposium on Logic In Computer Science, LICS '01*. IEEE Computer Society Press, 2001.

[20] J. Laird. A categorical semantics of higher-order store. In *Proceedings of CTCS '02*, ENTCS, 2002.

[21] J. Laird. Games and sequential algorithms. *Theoretical Computer Science*, 2002. To appear.

[22] J. Laird. Bistability: an extensional characterization of sequentiality. In *Proceedings of CSL '03 (to appear)*, 2003.

[23] J. Laird. A fully abstract and effectively presentable model of unary FPC. To appear in the proceedings of TLCA '03, 2003.

[24] J. Laird. A game semantics of linearly-used continuations. In *Proceedings of FoSSaCS '03*, 2003.

[25] O. Laurent. Polarized games. In *Proceedings of the Seventeenth International Symposium on Logic In Computer Science, LICS '02*, 2002.

[26] P. Malacaria and C. Hankin. Non-deterministic games and program analysis: An application to security. In *Proceedings of LICS '99*, 1999.

[27] P. Malacaria and C. Hankin. Generalised flowcharts and games. In *Proceedings of the 25th International Colloquium on Automata, Langugages and Programming*, 1998.

[28] G. McCusker. A fully abstract relational model of Syntactic Control of Interference. In *Proceedings of Computer Science Logic '02*, number 2471 in LNCS. Springer, 2002.

[29] H. Nickau. Hereditarily sequential functionals. In *Proceedings of the Symposium on Logical Foundations of Computer Science:Logic at St. Petersburg*, LNCS. Springer-Verlag, 1994.

[30] U. S. Reddy. Global state considered unnecesary: Object-based semantics for interference-free imperative programs. *Lisp and Symbolic Computation*, 9(1), 1996.

[31] S. Zdancewic and A. C. Myers. Secure information flow via linear continuations. *Higher-Order and Symbolic Computation*, 15, 2002.