Annals of Pure and Applied Logic 164 (2013) 1038–1078

Contents lists available at SciVerse ScienceDirect

Annals of Pure and Applied Logic

www.elsevier.com/locate/apal

Imperative programs as proofs via game semantics

Martin Churchill [b,*], Jim Laird [a], Guy McCusker [a]

[a] University of Bath, United Kingdom
[b] Swansea University, United Kingdom

ARTICLE INFO

Article history:
Received 6 November 2011
Accepted 13 March 2013
Available online 18 June 2013

MSC:
68Q55
03B70
03F52
18C50

Keywords:
Game semantics
Full completeness
History-sensitive strategies
Sequentiality

ABSTRACT

Game semantics extends the Curry–Howard isomorphism to a three-way correspondence: proofs, programs, strategies. But the universe of strategies goes beyond intuitionistic logics and lambda calculus, to capture stateful programs. In this paper we describe a logical counterpart to this extension, in which proofs denote such strategies. The system is expressive: it contains all of the connectives of Intuitionistic Linear Logic, and first-order quantification. Use of Laird's *sequoid* operator allows proofs with imperative behaviour to be expressed. Thus, we can embed first-order Intuitionistic Linear Logic into this system, Polarized Linear Logic, and an imperative total programming language.

The proof system has a tight connection with a simple game model, where games are forests of plays. Formulas are modelled as games, and proofs as history-sensitive winning strategies. We provide a strong *full completeness* result with respect to this model: each finitary strategy is the denotation of a unique analytic (cut-free) proof. Infinite strategies correspond to analytic proofs that are infinitely deep. Thus, we can normalise proofs, via the semantics.

1. Introduction

The Curry–Howard isomorphism between proofs in intuitionistic logics and functional programs is a powerful theoretical and practical principle for specifying and reasoning about programs. Game semantics provides a third axis to this correspondence: each proof/program at a given type denotes a strategy for the associated game, and typically a *full completeness* result establishes that this correspondence is also an isomorphism [3]. However, in languages with side-effects such as mutable state it is evident that there are many programs which do not correspond to intuitionistic proofs. Game semantics has achieved notable success in providing models of such programs [5,2,20], in which they typically denote "history-sensitive" strategies — strategies which may break the constraints of innocence [14] or history-freeness [3] imposed in fully complete models of intuitionistic or linear logic. The full completeness of these models means there is a precise correspondence between programs and history-sensitive strategies, which raises the question: is there a logic to flesh out the proofs/imperative programs/history-sensitive strategies correspondence?

In this paper we present a first-order logic, WS1, and a games model for it in which proofs denote history-sensitive strategies. Thus total imperative programs correspond, via the game semantics, to proofs in WS1. Moreover, because WS1 is more expressive than the typing system for a typical programming language, it can express finer behavioural properties of strategies. In particular, we can embed first-order intuitionistic logic with equality, Polarized Linear Logic, and a finitary imperative language with ground store, coroutines and some infinite data structures. We also take first steps towards

* Corresponding author.
E-mail address: m.d.churchill@swansea.ac.uk (M. Churchill).

0168-0072/$ – see front matter
http://dx.doi.org/10.1016/j.apal.2013.05.005

answering some of the questions posed by the logic and its semantics: Are there any formulas which only have 'imperative proofs', but no proofs in a traditional 'functional' proof system? Can we use the expressivity of WS1 to specify imperative programs?

1.1. Related work

The games interpretation of linear logic upon which WS1 is based was introduced by Blass in a seminal paper [7]. Blass also gives instances of history-sensitive strategies which are not denotations of linear logic proofs; these do, however, correspond to proofs in WS1. The particular symmetric monoidal closed category of games underlying our semantics has been studied extensively from both logical and programming perspectives [11,26,15]. Longley's project to develop a programming language based on it [30] may be seen as complementary to our aim of understanding it from a logical perspective.

Several logical systems have taken games or interaction as a semantic basis yielding a richer notion of meaning than classical or intuitionistic truth, including Ludics [12] and Computability Logic [18]. The latter also provides an analysis of Blass's examples, suggesting further connections with our logic, although there is a difference of emphasis: the research described here is focused on investigating the structural properties of the games model on which it is based.

Perhaps closest in spirit to our work is tensorial logic, introduced in [34]. Like WS1, tensorial logic is directly inspired by the structure of strategies in game semantics, and in [33], Melliès demonstrates a tight correspondence between the logic and categories of innocent strategies on dialogue games. Our focus in this paper is somewhat different, because we are primarily concerned with the history-sensitive behaviour characteristic of (game semantics of) imperative programs, rather than the purely functional programs that denote innocent strategies.

In [9] a proof theory for Conway games is presented, where formulas are the game trees themselves. In [13], the $\lambda\bar{\lambda}$-calculus is presented, where individual moves of game semantics are represented by variables and binders. Both settings deal with history-sensitive strategies, and have dynamics corresponding to composition of strategies.

A quite different formalisation of game semantics for first-order logic is given in [29], also with a full completeness result.

1.2. Contribution

The main contribution of this paper is to present an expressive logical system and its semantics, in which proofs correspond to history-sensitive strategies. Illustrating the expressive power of this system, we show how proofs of intuitionistic first-order logic, Polarized Linear Logic and imperative programming constructs may be embedded in it. We also demonstrate how formulas in the logic can be used to represent some properties of imperative programs: for example, we describe a formula for which any proof corresponds to a well-behaved (single write) Boolean storage cell.

The interpretation of WS1 includes some interesting developments of game semantics. In particular, the exponentials are treated in a novel way: we use the fact that the semantic exponential introduced in [15] is a final coalgebra, and reflect this explicitly in the logic in the style of [8]. This formulation allows us to express the usual exponential introduction rules (promotion and dereliction) but also proofs that correspond to strategies on $!A$ that act differently on each interrogation, such as the reusable Boolean reference cell. Another development is the interpretation of first-order logic with equality. A proof corresponds to a family of winning strategies — one for each possible interpretation of the atoms determined by a standard notion of $\mathcal{L}$-structure — which must be *uniform* across $\mathcal{L}$-structures. This notion of uniformity is precisely captured by the requirement that strategies are *lax natural transformations* between the relevant functors.

The main technical results of this paper concern the sharp correspondence between proofs and strategies: *full completeness* results. We show that any bounded uniform winning strategy is the denotation of a unique (cut-free) *analytic proof*. In the exponential-free fragment, where all strategies are bounded, it follows that many rules such as cut are admissible; and it allows us to normalise proofs to analytic proofs via the semantics. For the full logic, since the exponentials correspond to final coalgebras, proofs can be unfolded to infinitary form. Extending semantics-based normalisation to the full WS1, the resulting normal forms are *infinitary* analytic proofs.

2. Games and strategies

Our notion of game is essentially that introduced by [7], and similar to that of [3,25], augmented with winning conditions introduced as in [15]. We make use of the categorical structure on games and strategies first introduced in [19].

Informally, a game is a tree where Player and Opponent own alternate nodes, together with a polarity specifying which protagonist owns the starting node. A play proceeds down a particular branch, with Opponent/Player choosing the subtree for nodes they control. A strategy for Player specifies which choice Player should make in response to Opponent's moves so far. The winner of a finite play is the last protagonist to play a move. The winner of an infinite play is specified by a winning condition for each game.

If A is a set, let A^* denote the free monoid (set of sequences) over A, A^ω the set of infinite sequences over A, and ϵ the empty sequence. We write $s \sqsubseteq t$ if s is a prefix of s, and $s \sqsubset t$ if s is a strict (finite) prefix of (possibly infinite) t. If $X \subseteq A^*$, write $\overline{X} = \{s \in A^\omega\colon\ \forall t \sqsubset s, t \in X\}$.