



Citation for published version:

Churchill, M, Laird, J & McCusker, G 2011, Imperative programs as proofs via game semantics. in 26th Annual IEEE Symposium on Logic in Computer Science (LICS) 2011. Annual Symposium on Logic in Computer Science, IEEE, pp. 65-74, 26th Annual IEEE Symposium on Logic in Computer Science, Toronto, ON, Canada, 21/06/11. <https://doi.org/10.1109/LICS.2011.19>

DOI:

[10.1109/LICS.2011.19](https://doi.org/10.1109/LICS.2011.19)

Publication date:

2011

[Link to publication](#)

© 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

University of Bath

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Imperative Programs as Proofs via Game Semantics

Martin Churchill
Department of Computer Science
University of Bath
m.d.churchill@bath.ac.uk

James Laird
Department of Computer Science
University of Bath
j.d.laird@bath.ac.uk

Guy McCusker
Department of Computer Science
University of Bath
g.a.mccusker@bath.ac.uk

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. We can embed intuitionistic first-order linear logic into this system, as well as an imperative total programming language. The logic makes explicit use of the fact that in the game semantics the exponential can be expressed as a final coalgebra. We establish a full completeness theorem for our logic, showing that every bounded strategy is the denotation of a proof.

I. INTRODUCTION

A. Motivation

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 [1]. 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 [2], [3], in which they typically denote “history-sensitive” strategies — strategies which may break the constraints of innocence [4] or history-freeness [1] imposed in fully complete models of intuitionistic or linear logic.

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*. However, because *WS1* is more expressive than the typing system for a typical programming language, it can express more behavioural properties of strategies. In particular, we can embed both first-order intuitionistic logic with equality, and the total fragment of Idealized Algol [5] over bounded datatypes in *WS1*. We also take first steps towards 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 and reason about imperative programs?

B. Related Work

The games interpretation of linear logic upon which *WS1* is essentially based was introduced by Blass in a seminal paper [6]. This also contains 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 [7]–[9]. In particular, Longley’s project to develop a programming language based on it [10] may be seen as complementary to our aim of understanding it from a logical perspective.

WS1 and its semantics are a direct development of the logic *WS* introduced in [11] together with a notion of categorical model based on the sequoidal structure identified in [12] and a fully complete game semantics. To be precise, *WS1* augments *WS* with quantifiers, predicates, propositional atoms and exponentials; and thus significantly increases its expressive power.

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 [13] and Computability Logic [14]. 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 more focussed on investigating the structural properties of the games model on which it is based. In [15] a proof theory for Conway games is presented, where formulas are the game trees themselves. This system thus contains low-level proof rules representing individual moves of strategies, and the cut rewriting procedure corresponds to explicit composition of strategies. Our work contains the analogy of this for tree-games with local alternation.

A quite different formalisation of game semantics for first order logic is given in [16], also with a full completeness result.

C. 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, and programs of total Idealized Algol 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 [9] is a final coalgebra, and reflect this explicitly in the logic in the style of [17]. 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.

We extend the full completeness result of [11]. We show that any bounded uniform winning strategy is the denotation of a unique (cut-free) *core 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 core 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 core proofs*.

II. THE LOGIC WS1

The formulas of **WS1** are based on first-order linear logic, with some additional connectives, and subject to a notion of *polarity* (essentially, distinguishing conjunctive from disjunctive formulas, *pace* polarized linear logic). Let \mathcal{L} be a collection of complementary pairs of predicate symbols ϕ (negative) and $\bar{\phi}$ (positive), each with an arity in \mathbb{N} such that $\text{ar}(\phi) = \text{ar}(\bar{\phi})$. The negative predicates must include equality, and we write \neq for its complement. The negative and positive formulas of **WS1** over \mathcal{L} are defined by the following grammar (variables range over some global set \mathcal{V}):

$M, N :=$	1		\perp		$\phi(\vec{x})$	
	$M \otimes N$		$M \otimes N$		$N \triangleleft P$	
	$\forall x.P$		$M \& N$		$!N$	
$P, Q :=$	0		\top		$\bar{\phi}(\vec{x})$	
	$P \wp Q$		$P \triangleleft Q$		$P \otimes N$	
	$\exists x.P$		$P \oplus Q$		$?P$	

The involutive negation operation $(_)^\perp$ sends negative formulas to positive ones and vice-versa by exchanging each atom, unit or connective for its dual — i.e. **1** for **0**, \perp for \top , $\phi(\vec{x})$ for $\bar{\phi}(\vec{x})$, \otimes for \wp , \otimes for \triangleleft , \forall for \exists , $\&$ for \oplus and $!$ for $?$.

A. Informal Semantics

By an \mathcal{L} -structure L we mean the standard notion: a set $|L|$ together with an interpretation function I_L sending each predicate symbol (with arity n) to a function $|L|^n \rightarrow \{\text{tt}, \text{ff}\}$ such

that $I_L(\phi)(\vec{a}) \neq I_L(\bar{\phi})(\vec{a})$ for all \vec{a} and $I_L(=)(a, b) = \text{tt}$ iff $a = b$. If $X \subseteq \mathcal{V}$ an \mathcal{L} -model over X is a pair (L, v) where L is an \mathcal{L} -structure and $v : X \rightarrow |L|$ a valuation function, yielding an assignment of truth values to all atomic formulas with variables in X .

We may use this assignment to derive a denotation with respect to (L, v) for each formula as an alternating dialogue game between two protagonists, Player and Opponent, such that in dialogues corresponding to positive (respectively, negative) formulas Player (respectively, Opponent) must make the first move. We sketch the interpretation of formulas as games (which, for the connectives of linear logic, is based on that of [6]) here, and give further details of the model in Section 3.

- **0** and **1** denote the game with no moves.
- \perp and \top denote the games with a single move (belonging to Opponent or Player, respectively).
- Positive atoms which are satisfied in (L, v) are interpreted as the game with a single (Player) move; positive atoms which are not satisfied are interpreted as the game with no moves. Conversely, negative atoms which are satisfied in (L, v) are interpreted as the empty game, whilst negative atoms which are not satisfied are interpreted as the game with a single Opponent move.
- In $M \& N$ and $P \oplus Q$ the starting protagonist chooses an opening move in either of the two components, and play remains in that component.
- In $M \otimes N$ and $P \wp Q$, dialogues are interleavings of plays from the two components, with the starting protagonist able to switch between them. (Note that **1** and **0** are units for both the additive and multiplicative connectives, and that our interpretation is therefore *affine*.)
- If A is a formula of either polarity, the dialogues in $A \otimes M$ and $A \triangleleft P$ are left-merges: interleavings where the starting protagonist must begin in the left-hand component, but it is the protagonist who starts in the right-hand component who may switch between them.
- In $!M$ (resp. $?P$) dialogues are interleavings of arbitrarily many plays in M (resp. P); the starting protagonist may choose at any point to open a new copy, and may switch between copies.
- In $\forall x.N(x)$ (resp. $\exists x.P(x)$) dialogues are played in $N(a)$ (resp. $P(a)$) for some value $a \in |L|$ chosen by the starting protagonist.

Note that the negation operation corresponds precisely to exchanging the rôles of Player and Opponent.

A game also includes a specification of its winning positions, which will be described in Section 3. A proof of a formula will denote a uniform family of “winning strategies” for Player on the games denoted by that formula in each \mathcal{L} -model.

B. Proofs

With this interpretation in mind, we can define proof rules for **WS1**. A *sequent* of **WS1** is of the form $X; \Theta \vdash \Gamma$ where $X \subseteq \mathcal{V}$, Θ is a set of positive atomic formulas and Γ is a nonempty list of formulas such that $FV(\Theta, \Gamma) \subseteq X$.

The explicit free variable set X is required for the tight correspondence between the syntax and semantics.

We shall interpret such a sequent as a (family of) dialogue games by interpreting the comma operator in Γ as left-associative left-merge (either \otimes or \triangleleft depending on the polarity of the right-hand operand), so that the first move must occur in the first element (or head formula) of Γ . This family is indexed over the collection of Θ -satisfying \mathcal{L} -models over X .

Proof rules for **WS1** are given in Figure 1. Δ^+ ranges over lists of positive formulas, Γ^* over non-empty lists of formulas, and Φ over $X; \Theta$ contexts. We obtain the *affine fragment* of **WS1** by removing the exponentials and all rules that mention them.

Note that the core introduction rules have a particular shape: they operate on the head formula of the sequent, and the only connectives corresponding to a choice of introduction rule are \wp , \oplus and \exists . Core elimination rules work on the second formula in a sequent, shortening the tail. For the affine fragment, this provides the basis of a cut-free core subsystem, in which proof search is particularly simple, and which is sufficient to express all uniform winning strategies, and so all of the ‘other’ rules are admissible in this fragment.

All of the core rules are additive. This is particularly striking in the case of the \otimes introduction rule: unlike e.g. linear logic, we explicitly use the fact that \otimes represents an interleaving of its subgames, rather than just an arbitrary monoidal structure. In particular, the rule decomposes it into the possibilities of starting on the left or starting on the right.

Finally, the exponential rules are based on the fact that $!N$ is the final coalgebra of the functor $X \mapsto N \otimes X$ — this is represented explicitly in the rule P_{ana} . Together with P_{con} and P_{der} , we may derive promotion.

C. Embedding of Linear Logic

For any negative formulas M, N , define $M \multimap N$ to be $N \triangleleft M^\perp$. Thus any formula of first-order intuitionistic linear logic is a negative formula of **WS1**. To illustrate its expressive power, we sketch an embedding into **WS1** of proofs of ILL (over the connectives $\otimes, \multimap, \forall, \&, \mathbf{1}, \perp, !$ and (negative) atoms, formulated with left- and right- introduction rules as in [18]).

Proposition 1: For any proof p of $M_1, \dots, M_n \vdash N$ in ILL there is a proof $\kappa(p)$ in **WS1** of $\vdash N, M_1^\perp, \dots, M_n^\perp$.

Proof: We show that for each rule of ILL there is a derivation in **WS1** of the conclusion from the premises. In most cases, this consists of a single rule, or pair of rules, of **WS1**. The important exception is the right-introduction rule for $!$ (promotion):

$$\frac{\Gamma \vdash N}{\Gamma \vdash !N}$$

If Γ is empty, we can use P_1^\top and P_0^\top together with P_{ana} . In the case that Γ contains a single formula L , the translation is as follows:

$$P_{\text{mix}} \frac{\frac{P_{\text{id}} \frac{\vdash N, ?L^\perp}{\vdash !L, ?L^\perp}}{P_{\wp}^\top \frac{\vdash N, !L, ?L^\perp, ?L^\perp}{\vdash N, !L, ?L^\perp \wp ?L^\perp}}}{P_{\text{cut}} \frac{\vdash N, !L, ?L^\perp}{\vdash !L \otimes !L, ?L^\perp}} \quad \text{con} \vdash !L \otimes !L, ?L^\perp$$

$$P_{\text{ana}} \frac{\vdash N, ?L^\perp}{\vdash !N, ?L^\perp}$$

where con is as follows:

$$P_{\text{con}} \frac{P_{\text{id}} \frac{\vdash !L, ?L^\perp}{\vdash !L, !L, ?L^\perp}}{P_{\otimes} \frac{\vdash !L, !L, ?L^\perp}{\vdash !L \otimes !L, ?L^\perp}} \quad P_{\text{con}} \frac{P_{\text{id}} \frac{\vdash !L, ?L^\perp}{\vdash !L, !L, ?L^\perp}}{P_{\otimes} \frac{\vdash !L, !L, ?L^\perp}{\vdash !L \otimes !L, ?L^\perp}}$$

To deal with cases where Γ contains more than one formula, one must use the fact that we can derive the equivalence of $!M \otimes !N$ and $!(M \& N)$ in **WS1**. ■

D. New Provable Formulas

We next sketch some examples of formulas that are not provable in ILL but are provable in **WS1** — i.e. they denote games on which there are uniform winning history-sensitive strategies which are expressible in **WS1**.

1) *Memoization:* The formula $\phi_{\text{ex}} = (\phi \& (\phi \multimap \perp)) \multimap \perp$ corresponds to an ‘‘additive excluded middle’’ in (negative) ILL, and is not provable. This formula is not provable in **WS1** either. However, consider the formula $\phi_{\text{ex}} \multimap \phi_{\text{ex}} \otimes \phi_{\text{ex}}$. This is not provable in ILL but it is provable in **WS1** (as $\phi_{\text{ex}} \otimes \phi_{\text{ex}} \triangleleft \phi_{\text{ex}}^\perp$). While Player can only access the input ϕ_{ex} once, in the corresponding game, he can ‘remember’ whether ϕ was true or false, to give a winning history-sensitive uniform strategy. It is straightforward to derive a proof of $\phi_{\text{ex}} \multimap \phi_{\text{ex}} \otimes \phi_{\text{ex}}$ from this strategy, although there is no space to give it here. This example can be extended to the exponentials: while $\phi_{\text{ex}} \multimap !\phi_{\text{ex}}$ is not provable in ILL, it is provable in **WS1**.

2) *Medial Rule:* The formulas

$$((A \otimes B \multimap \perp) \otimes (C \otimes D \multimap \perp) \multimap \perp) \multimap ((A \multimap \perp) \otimes (C \multimap \perp) \multimap \perp) \otimes ((B \multimap \perp) \otimes (D \multimap \perp) \multimap \perp)$$

are not provable, in general, in intuitionistic linear logic (in particular, when A, B, C, D are instantiated as negative atoms) — they are a counterpart in ILL of the *medial* rule, $[(A \otimes B) \wp (C \otimes D)] \multimap [(A \wp C) \otimes (B \wp D)]$. As observed by Blass [6], however, there are (uniform) history-sensitive winning strategies for medial. For example, if Opponent first chooses the left hand component in the output and the right hand component in the input, Player can choose to play copycat between the copies of C , and so on. By our full completeness theorem (Section 4), therefore, there are proofs of all (exponential-free) instantiations of this formula in **WS1**.

Similarly, the following formula (from [6]) is not provable in ILL, but is provable in **WS1**.

$$[A \otimes (C \& D)] \& [B \otimes (C \& D)] \& [(A \& B) \otimes C] \& [(A \& B) \otimes D] \multimap (A \& B) \otimes (C \& D)$$

Fig. 1: Proof rules for WS1

Core rules:			
$P_1 \frac{}{\Phi \vdash \mathbf{1}, \Gamma}$	$P_{\top} \frac{}{\Phi \vdash \top}$	$P_{\text{at-}} \frac{X; \Theta, \bar{\phi}(\vec{x}) \vdash \perp, \Gamma}{X; \Theta \vdash \phi(\vec{x}), \Gamma}$	$P_{\text{at+}} \frac{X; \Theta, \bar{\phi}(\vec{x}) \vdash \top, \Gamma}{X; \Theta, \bar{\phi}(\vec{x}) \vdash \bar{\phi}(\vec{x}), \Gamma}$
$P_{\otimes} \frac{\Phi \vdash A, N, \Gamma}{\Phi \vdash A \otimes N, \Gamma}$	$P_{\triangleleft} \frac{\Phi \vdash A, P, \Gamma}{\Phi \vdash A \triangleleft P, \Gamma}$	$P_{\otimes} \frac{\Phi \vdash M, N, \Gamma \quad \Phi \vdash N, M, \Gamma}{\Phi \vdash M \otimes N, \Gamma}$	$P_{\&} \frac{\Phi \vdash M, \Gamma \quad \Phi \vdash N, \Gamma}{\Phi \vdash M \& N, \Gamma}$
$P_{\wp 1} \frac{\Phi \vdash P, Q, \Gamma}{\Phi \vdash P \wp Q, \Gamma}$	$P_{\wp 2} \frac{\Phi \vdash Q, P, \Gamma}{\Phi \vdash P \wp Q, \Gamma}$	$P_{\oplus 1} \frac{\Phi \vdash P, \Gamma}{\Phi \vdash P \oplus Q, \Gamma}$	$P_{\oplus 2} \frac{\Phi \vdash Q, \Gamma}{\Phi \vdash P \oplus Q, \Gamma}$
$P_{\top}^+ \frac{\Phi \vdash \top, \Gamma}{\Phi \vdash \top, P, \Gamma}$	$P_{\top}^- \frac{\Phi \vdash N}{\Phi \vdash \top, N}$	$P_{\triangleleft}^+ \frac{\Phi \vdash \top, N \triangleleft P, \Gamma}{\Phi \vdash \top, N, P, \Gamma}$	$P_{\otimes}^+ \frac{\Phi \vdash \top, M \otimes N, \Gamma}{\Phi \vdash \top, M, N, \Gamma}$
$P_{\perp}^- \frac{\Phi \vdash \perp, \Gamma}{\Phi \vdash \perp, N, \Gamma}$	$P_{\perp}^+ \frac{\Phi \vdash P}{\Phi \vdash \perp, P}$	$P_{\perp}^{\otimes} \frac{\Phi \vdash \perp, P \otimes N, \Gamma}{\Phi \vdash \perp, P, N, \Gamma}$	$P_{\perp}^{\wp} \frac{\Phi \vdash \perp, P \wp Q, \Gamma}{\Phi \vdash \perp, P, Q, \Gamma}$
$P_{\text{ma}} \frac{(X; \Theta \vdash \Gamma) \left[\frac{z}{x}, \frac{z}{y} \right] \quad X; \Theta, x \neq y \vdash \Gamma}{X; \Theta \vdash \Gamma}$		$P_{\neq} \frac{}{X; \Theta, x \neq x \vdash \Gamma}$	$P_{\exists}^y \frac{X; \Theta \vdash P[y/x], \Gamma}{X; \Theta \vdash \exists x. P, \Gamma} \quad y \in X$
$P_{\vee} \frac{X \uplus \{x\}; \Theta \vdash N, \Gamma}{X; \Theta \vdash \forall x. N, \Gamma} \quad x \notin FV(\Theta, \Gamma)$		$P_{!} \frac{\Phi \vdash M, !M, \Gamma}{\Phi \vdash !M, \Gamma}$	$P_{?} \frac{\Phi \vdash P, ?P, \Gamma}{\Phi \vdash ?P, \Gamma}$
Other rules:			
$P_{\otimes}^{\top} \frac{\Phi \vdash \Gamma^*, M, N, \Delta}{\Phi \vdash \Gamma^*, M \otimes N, \Delta}$	$P_{\mathbf{1}}^{\top} \frac{\Phi \vdash \Gamma^*, \Delta}{\Phi \vdash \Gamma^*, \mathbf{1}, \Delta}$	$P_{\text{wk}} \frac{\Phi \vdash \Gamma^*, M, \Delta}{\Phi \vdash \Gamma^*, \Delta}$	$P_{\vee}^e \frac{X; \Theta \vdash \Gamma, \forall x. N, \Delta}{X; \Theta \vdash \Gamma, N[y/x], \Delta} \quad y \in X$
$P_{\wp}^{\top} \frac{\Phi \vdash \Gamma^*, P, Q, \Delta}{\Phi \vdash \Gamma^*, P \wp Q, \Delta}$	$P_{\mathbf{0}}^{\top} \frac{\Phi \vdash \Gamma^*, \Delta}{\Phi \vdash \Gamma^*, \mathbf{0}, \Delta}$	$P_{\text{str}} \frac{\Phi \vdash \Gamma^*, \Delta}{\Phi \vdash \Gamma^*, P, \Delta}$	$P_{\exists}^{\top} \frac{X; \Theta \vdash \Gamma, P[y/x], \Delta}{X; \Theta \vdash \Gamma, \exists x. P, \Delta} \quad y \in X$
$P_{\top}^{\top} \frac{}{\Phi \vdash N, \Delta^+, \top, \Delta_1^+}$	$P_{\oplus 1}^{\top} \frac{\Phi \vdash \Gamma, P, \Delta}{\Phi \vdash \Gamma, P \oplus Q, \Delta}$	$P_{\oplus 2}^{\top} \frac{\Phi \vdash \Gamma, Q, \Delta}{\Phi \vdash \Gamma, P \oplus Q, \Delta}$	$P_{\text{cut}} \frac{\Phi \vdash \Gamma^*, N^{\perp}, \Gamma_1 \quad \Phi \vdash N, \Delta^+}{\Phi \vdash \Gamma^*, \Delta^+, \Gamma_1}$
$P_{\text{der}} \frac{\Phi \vdash \Gamma, !M, \Delta}{\Phi \vdash \Gamma, M, \Delta}$	$P_{\& 1}^e \frac{\Phi \vdash \Gamma, M \& N, \Delta}{\Phi \vdash \Gamma, M, \Delta}$	$P_{\& 2}^e \frac{\Phi \vdash \Gamma, M \& N, \Delta}{\Phi \vdash \Gamma, N, \Delta}$	$P_{\text{mix}} \frac{\Phi \vdash M, \Gamma, \Delta^+ \quad \Phi \vdash N, \Delta_1^+}{\Phi \vdash M, \Gamma, N, \Delta^+, \Delta_1^+}$
$P_{\text{con}} \frac{\Phi \vdash \Gamma, !M, \Delta}{\Phi \vdash \Gamma, !M, !M, \Delta}$	$P_{\text{sym}}^+ \frac{\Phi \vdash \Gamma^*, P, Q, \Delta}{\Phi \vdash \Gamma^*, Q, P, \Delta}$	$P_{\text{sym}}^- \frac{\Phi \vdash \Gamma^*, M, N, \Delta}{\Phi \vdash \Gamma^*, N, M, \Delta}$	$P_{\rightarrow} \frac{\Phi \vdash M, \Gamma, P \quad \Phi \vdash N, \Delta^+}{\Phi \vdash M, \Gamma, P \otimes N, \Delta^+}$
$P_{\text{ana}} \frac{\Phi \vdash M, P^{\perp}, P}{\Phi \vdash !M, P}$	$P_{\text{id}} \frac{}{\Phi \vdash N, N^{\perp}}$	$P_{\text{cut}}^0 \frac{\Phi \vdash N^{\perp} \quad \Phi \vdash N, Q}{\Phi \vdash Q}$	$P_{\text{id}\otimes} \frac{\Phi \vdash N, Q}{\Phi \vdash M, N, M^{\perp} \triangleleft Q}$

3) *Exponentials*: We make a comment on the power of explicit anamorphisms in our logic for the exponential. In ILL there is a polynomial bound on resources used by proofs: if $\sigma : !A \multimap !B$ is the denotation of an ILL proof then there exists a polynomial p such that if $s \in \sigma$ and $s|_{!B}$ enters n copies of B then $s|_{!A}$ enters $p(n)$ copies of A (contraction corresponds to addition and promotion to multiplication). However, we can define a proof in WS1 representing the (history-sensitive) strategy on $!(\perp \triangleleft \top) \multimap !(\perp \triangleleft \top)$ which, on the n th interrogation of the output, interrogates the input 2^n times.

E. Imperative Programs

We next describe how imperative programs and their properties can be explicitly modelled in WS1. First, we define a (negative) formula of WS1 corresponding to the type of Booleans: $\mathbf{B} = \perp \triangleleft \top \oplus \top$ (i.e. $\perp \& \perp \multimap \perp$) — this has one initial Opponent-move \mathfrak{q} and two possible Player responses,

representing True or False. There is a conditional $\mathbf{B} \multimap N \& N \multimap N$ for each N .

We can also define $\mathbf{Bi} = (\perp \& \perp) \triangleleft \top$, which has two initial Opponent-moves `inputTrue` and `inputFalse` and one possible response to this, `ok`. The formula $!\mathbf{Bi} \otimes !\mathbf{B}$ represents the type of a Boolean variable — it has a `write` method which takes a Boolean input and has a single output move, and a `read` method which on interrogation outputs a Boolean value, and these methods can be used arbitrarily many times. This formula is equivalent to $\text{var } =!(\mathbf{Bi} \& \mathbf{B})$.

In WS1 we can use the P_{ana} rule to give a proof of $\mathbf{B} \multimap \text{var}$ representing a Boolean cell with a given starting value. In particular, we take the anamorphism of a map $\mathbf{B} \multimap (\mathbf{B} \& \mathbf{Bi}) \otimes \mathbf{B}$. This proof is given in Figure 2. In this proof, if a rule is not labelled it is the unique core rule, and some steps are omitted for brevity. Its semantics is the history-sensitive Boolean cell strategy given in [2]. The proof p_{read} corresponds to the map $\mathbf{B} \multimap \mathbf{B} \otimes \mathbf{B}$ which reads its argument and propagates it to

the next call, and p_{write} corresponds to the map $\mathbf{B} \multimap \mathbf{Bi} \otimes \mathbf{B}$ which ignores its argument and propagates the written value to the next call.

Similarly, we may define a proof of $\vdash \text{var}$ representing a stack with unbounded memory: the \mathbf{Bi} component represents a `push`, and the \mathbf{B} component a `pop`.

1) *Language Embedding*: We have now seen how terms of the simply-typed lambda calculus (in the guise of proofs of intuitionistic linear logic), simple finitary data types such as the Booleans, and Boolean reference types and cells may be expressed as formulas and proofs of **WS1**. Thus we may translate the recursion-free fragment of *Idealized Algol* [5] over finite datatypes into **WS1**, for example. We also have proofs corresponding to non-local control operators such as `catch` [19] and coroutines [3]. Interpretation via **WS1** thus gives a semantics of an imperative total programming language with all of these features as games and history sensitive strategies, consistent with existing games models of such languages [2], [3].

The formulas of **WS1** are more expressive than the types of languages such as *Idealized Algol*, and hence they enable the behaviour of history sensitive strategies to be specified more precisely. For example, formulas can specify the order in which arguments are interrogated, how many times they are interrogated, and relationships between inputs and outputs of ground type (using the first-order structure).

F. Good Variables

One respect in which the game semantics of *Idealized Algol* (and other imperative languages) fails to reflect its syntax fully is in the existence in the model of *bad variables* which do not return the last value assigned to them [2]. In **WS1** we may define formulas for which all proofs denote good variables.

The formula $\mathbf{worm} = \mathbf{Bi} \otimes \mathbf{B}$ represents a Boolean variable which can be written once, then read many times. One proof/strategy of this formula will indeed be a valid Boolean cell: if Opponent plays `inputX` then Player responds with `ok`, if Opponent then tries to read the cell `q`, then Player responds with `X`. But there are also bad variables: for example, the read method may always return `True` regardless of what was written.

To exclude such behaviour, we can replace the input/output moves with atoms. Define $\mathbf{B}' = \perp \triangleleft (\bar{\phi} \oplus \bar{\psi})$ and $\mathbf{Bi}' = (\phi \& \psi) \triangleleft \top$, with $\mathbf{worm}' = \mathbf{Bi}' \otimes \mathbf{B}'$. If ϕ and ψ are assigned `tt`, then this denotes the same dialogue as \mathbf{worm} . However, the denotation of any proof of \mathbf{worm}' at such a model must be the good variable strategy. The rule for ϕ (and semantically, uniformity of strategies) ensures that ϕ must be played before $\bar{\phi}$, and ψ before $\bar{\psi}$. Consequently, Player can only respond with a particular Boolean in the `read` component if it has previously been given as an input in the `write` component. A proof of this formula is next shown, using only the core rules and the promotion operator defined in Section 2C. Note, however, that this example does not scale to variables that allow multiple `write` operations.

$$\begin{array}{c}
\frac{}{\bar{\phi} \vdash \top} \\
\frac{}{\bar{\psi} \vdash \top} \\
\text{P}_{\oplus 1} \frac{}{\bar{\phi} \vdash \bar{\phi}} \quad \text{P}_{\oplus 2} \frac{}{\bar{\psi} \vdash \bar{\psi}} \\
\frac{}{\bar{\phi} \vdash \bar{\phi} \oplus \bar{\psi}} \quad \frac{}{\bar{\psi} \vdash \bar{\phi} \oplus \bar{\psi}} \\
\frac{}{\bar{\phi} \vdash \perp, \bar{\phi} \oplus \bar{\psi}} \quad \frac{}{\bar{\psi} \vdash \perp, \bar{\phi} \oplus \bar{\psi}} \\
\text{prom} \frac{}{\bar{\phi} \vdash \perp \triangleleft \bar{\phi} \oplus \bar{\psi}} \quad \text{prom} \frac{}{\bar{\psi} \vdash \perp \triangleleft \bar{\phi} \oplus \bar{\psi}} \\
\frac{}{\bar{\phi} \vdash !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \quad \frac{}{\bar{\psi} \vdash !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \\
\frac{}{\bar{\phi} \vdash \top, !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \quad \frac{}{\bar{\psi} \vdash \top, !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \\
\frac{}{\bar{\phi} \vdash \top \otimes !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \quad \frac{}{\bar{\psi} \vdash \top \otimes !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \\
\frac{}{\bar{\phi} \vdash \perp, \top \otimes !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \quad \frac{}{\bar{\psi} \vdash \perp, \top \otimes !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \\
\frac{}{\bar{\phi} \vdash \perp, \top, !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \quad \frac{}{\bar{\psi} \vdash \perp, \top, !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \\
\frac{}{\vdash \phi, \top, !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \quad \frac{}{\vdash \psi, \top, !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \\
\frac{}{\vdash (\phi \& \psi), \top, !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \\
\frac{}{\vdash (\phi \& \psi) \triangleleft \top, !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})} \\
\frac{}{\vdash ((\phi \& \psi) \triangleleft \top) \otimes !(\perp \triangleleft \bar{\phi} \oplus \bar{\psi})}
\end{array}$$

III. SEMANTICS OF WS1

We now give a formal semantics of **WS1** — proofs are given semantics as uniform families of winning history-sensitive strategies, indexed by \mathcal{L} -models, where *uniform* means that the family can be expressed as a lax natural transformation. To formulate this we need to interpret formulas as functors from a category of \mathcal{L} -models to a category of games and strategies.

A. A Category of Games and Strategies

We first recall the symmetric monoidal category $(\mathcal{G}_w, \otimes, I)$ of (negative) games and winning history-sensitive strategies introduced in [9]. A *game* is a tree together with a winning condition: a triple (M, P, W) where M is a set (of ‘‘moves’’), P is a prefix-closed subset of sequences (plays) from M , and W is a function from infinite plays to $\{O, P\}$ identifying the winner of each such play. Assuming that Opponent starts, we may then derive the ‘owner’ of each subsequent move in a play.

The games $M \otimes N$ and $M \multimap N$ are defined as in Section 2A (with $M \multimap N = N \triangleleft M^\perp$) and are formally played over the disjoint sum of the moves of M and N . A play is P-winning on $M \otimes N$ if it is P-winning (by restriction) on M and N , and it is P-winning on $M \multimap N$ if it is P-winning on N or O-winning on M . The unit I for \otimes is the empty game (with no moves, and one empty play).

A strategy on a negative game (M, P, W) is a non-empty subset σ of P containing plays of even-length, that is even-prefix-closed and deterministic (if $sa, sb \in \sigma$ then $a = b$). We may form a symmetric monoidal category \mathcal{G} in which objects are games, and arrows from M to N are strategies on $M \multimap N$, in which composition of strategies $\sigma : M \multimap N$ and $\tau : N \multimap L$ is by ‘parallel composition plus hiding in N ’ to yield a strategy $\tau \circ \sigma : M \multimap L$. \mathcal{G} is poset-enriched by inclusion of strategies.

A strategy $\sigma : A$ is *total* if $s \in \sigma \wedge so \in P_A \Rightarrow \exists p. sop \in \sigma$. Total strategies are maximal with respect to inclusion. A

Fig. 2: Proof Denoting a Boolean Cell

$$\frac{p_{\text{write}} \text{ :! } (\perp \& \perp) \triangleleft \top, \perp \triangleleft (\top \oplus \top), \top \otimes (\perp \& \perp) \quad p_{\text{read}} \text{ :! } \perp \triangleleft (\top \oplus \top), \perp \triangleleft (\top \oplus \top), \top \otimes (\perp \& \perp)}{P_{\text{ana}} \frac{\vdash ((\perp \& \perp) \triangleleft \top) \& (\perp \triangleleft (\top \oplus \top)), \perp \triangleleft (\top \oplus \top), \top \otimes (\perp \& \perp)}{\vdash !(((\perp \& \perp) \triangleleft \top) \& (\perp \triangleleft (\top \oplus \top))), \top \otimes (\perp \& \perp)}}$$

where p_{write} is

$$\frac{\begin{array}{c} \overline{\vdash \top} \\ \overline{\vdash \top, (\top \otimes (\perp \& \perp))} \\ P_{\oplus 1} \frac{\vdash \top, (\top \otimes (\perp \& \perp))}{\vdash \top \oplus \top, (\top \otimes (\perp \& \perp))} \\ P_{\wp 1} \frac{\vdash \top \oplus \top, (\top \otimes (\perp \& \perp))}{\vdash (\top \oplus \top) \wp (\top \otimes (\perp \& \perp))} \\ \vdash \perp, (\top \oplus \top) \wp (\top \otimes (\perp \& \perp)) \\ \vdash (\perp \triangleleft (\top \oplus \top)) \triangleleft (\top \otimes (\perp \& \perp)) \\ \vdash \top, (\perp \triangleleft (\top \oplus \top)) \triangleleft (\top \otimes (\perp \& \perp)) \\ \vdash (\top \otimes (\perp \triangleleft (\top \oplus \top))), \top \otimes (\perp \& \perp) \\ P_{\wp 1} \frac{\vdash (\top \otimes (\perp \triangleleft (\top \oplus \top))), \top \otimes (\perp \& \perp)}{\vdash (\top \otimes (\perp \triangleleft (\top \oplus \top))) \wp (\top \otimes (\perp \& \perp))} \\ \vdash \perp, (\top \otimes (\perp \triangleleft (\top \oplus \top))) \wp (\top \otimes (\perp \& \perp)) \\ \vdash \perp, \top, \perp \triangleleft (\top \oplus \top), \top \otimes (\perp \& \perp) \end{array}}{\begin{array}{c} \overline{\vdash \top} \\ \overline{\vdash \top, (\top \otimes (\perp \& \perp))} \\ P_{\oplus 2} \frac{\vdash \top, (\top \otimes (\perp \& \perp))}{\vdash \top \oplus \top, (\top \otimes (\perp \& \perp))} \\ P_{\wp 1} \frac{\vdash \top \oplus \top, (\top \otimes (\perp \& \perp))}{\vdash (\top \oplus \top) \wp (\top \otimes (\perp \& \perp))} \\ \vdash \perp, (\top \oplus \top) \wp (\top \otimes (\perp \& \perp)) \\ \vdash (\perp \triangleleft (\top \oplus \top)) \triangleleft (\top \otimes (\perp \& \perp)) \\ \vdash \top, (\perp \triangleleft (\top \oplus \top)) \triangleleft (\top \otimes (\perp \& \perp)) \\ \vdash (\top \otimes (\perp \triangleleft (\top \oplus \top))), \top \otimes (\perp \& \perp) \\ P_{\wp 1} \frac{\vdash (\top \otimes (\perp \triangleleft (\top \oplus \top))), \top \otimes (\perp \& \perp)}{\vdash (\top \otimes (\perp \triangleleft (\top \oplus \top))) \wp (\top \otimes (\perp \& \perp))} \\ \vdash \perp, (\top \otimes (\perp \triangleleft (\top \oplus \top))) \wp (\top \otimes (\perp \& \perp)) \\ \vdash \perp, \top, \perp \triangleleft (\top \oplus \top), \top \otimes (\perp \& \perp) \end{array}}{\frac{\vdash \perp \& \perp, \top, \perp \triangleleft (\top \oplus \top), \top \otimes (\perp \& \perp)}{\vdash (\perp \& \perp) \triangleleft \top, \perp \triangleleft (\top \oplus \top), \top \otimes (\perp \& \perp)}}$$

and p_{read} is

$$\frac{\begin{array}{c} \overline{\vdash \top} \\ \overline{\vdash \top \oplus \top} \\ P_{\oplus 1} \frac{\vdash \perp, (\top \oplus \top)}{\vdash \top, \perp \triangleleft (\top \oplus \top)} \\ P_{\oplus 1} \frac{\vdash \top \oplus \top, \perp \triangleleft (\top \oplus \top)}{\vdash \perp, (\top \oplus \top) \otimes (\perp \triangleleft (\top \oplus \top))} \end{array}}{\frac{\vdash \perp \& \perp, (\top \oplus \top) \otimes (\perp \triangleleft (\top \oplus \top))}{\vdash \top, (\perp \& \perp) \triangleleft ((\top \oplus \top) \otimes (\perp \triangleleft (\top \oplus \top)))}} \frac{\begin{array}{c} \overline{\vdash \top} \\ \overline{\vdash \top \oplus \top} \\ P_{\oplus 2} \frac{\vdash \perp, (\top \oplus \top)}{\vdash \top, \perp \triangleleft (\top \oplus \top)} \\ P_{\oplus 2} \frac{\vdash \top \oplus \top, \perp \triangleleft (\top \oplus \top)}{\vdash \perp, (\top \oplus \top) \otimes (\perp \triangleleft (\top \oplus \top))} \end{array}}{\frac{\vdash \top \otimes (\perp \& \perp), (\top \oplus \top) \otimes (\perp \triangleleft (\top \oplus \top))}{\vdash ((\top \oplus \top) \otimes (\perp \triangleleft (\top \oplus \top))) \wp (\top \otimes (\perp \& \perp))}} \frac{\begin{array}{c} P_{\wp 2} \frac{\vdash ((\top \oplus \top) \otimes (\perp \triangleleft (\top \oplus \top))) \wp (\top \otimes (\perp \& \perp))}{\vdash \perp, ((\top \oplus \top) \otimes (\perp \triangleleft (\top \oplus \top))) \wp (\top \otimes (\perp \& \perp))} \\ \vdash \perp \triangleleft (\top \oplus \top), \perp \triangleleft (\top \oplus \top), \top \otimes (\perp \& \perp) \end{array}}{\vdash \perp \triangleleft (\top \oplus \top), \perp \triangleleft (\top \oplus \top), \top \otimes (\perp \& \perp)}$$

strategy is *winning* if it is total and if, given infinite s such that all even prefixes of s are in σ , $W(s) = P$. Winning strategies form a full-on-objects symmetric monoidal closed subcategory \mathcal{G}_w of \mathcal{G} (but total strategies do not compose in general). Both categories also have products given by $\&$.

B. Sequoidal Structure

Let \mathcal{G}_s be the full-on-objects subcategory of \mathcal{G} consisting of strict arrows: an arrow $\sigma : M \rightarrow L$ is *strict* if it responds to the first move in L with a move in M , if it responds at all. We may define an *action* $_ \otimes _$ of the SMC $(\mathcal{G}, \otimes, I)$ on \mathcal{G}_s based on the game $M \otimes N$ in which plays (resp. winning plays) are those (resp. winning) plays in $M \otimes N$ in which the first move

is in M . In fact we have a *sequoidal closed category* [12] in which $_ \otimes B : \mathcal{G}_s \rightarrow \mathcal{G}_s$ is left adjoint to $B \multimap _ : \mathcal{G}_s \rightarrow \mathcal{G}_s$.

The sequoidal closed structure restricts to \mathcal{G}_w and $\mathcal{G}_{s,w}$ (the category of strict, winning strategies). In the terminology of [11], we have a WS-category, which is a semantics for WS (the affine, unit-only fragment of WS1) in which proofs are given semantics as winning strategies. A proof of $\vdash M, \Gamma$ is interpreted as an arrow $I \rightarrow \llbracket M, \Gamma \rrbracket$. Positive formulas are handled by setting $\llbracket P \rrbracket = \llbracket P^\perp \rrbracket$ and a proof of $\vdash P, \Gamma$ is interpreted as an arrow $\llbracket P, \Gamma \rrbracket \rightarrow o$, where o is the game with a single maximal play of one move.

The categories \mathcal{G} and \mathcal{G}_w are equipped with an exponential

comonad [9]. The move set of $!N$ is $M_N \times \omega$; a play is an interleaving of plays in multiple copies of N tagged with natural numbers, such that the tags are introduced in successive order. An infinite play is P-winning just if the play restricted to each copy is P-winning. As well as $!$ enjoying the structure of a linear exponential comonad [18], we can also express $!N$ as a terminal coalgebra.

Proposition 2: In the category \mathcal{G}_w , $!N$ is the terminal coalgebra of the functor $X \mapsto N \circ X$.

Proof: There is an evident morphism $\alpha : !N \rightarrow N \circ !N$. Let $\sigma : M \rightarrow N \circ M$ be a winning strategy. We must construct $\llbracket \sigma \rrbracket : M \rightarrow !N$. Define $\llbracket \sigma \rrbracket_n : M \rightarrow (N \circ _)^n(M)$ by $\llbracket \sigma \rrbracket_1 = \sigma$ and $\llbracket \sigma \rrbracket_{n+1} = (\text{id} \circ \llbracket \sigma \rrbracket_n) \circ \sigma$.

Any play s in $!N$ must take place in a finite number of copies of N since it is finite: let $\text{tg}(s)$ be the largest tag occurring in s . Then s is also a play in $(N \circ _)^{\text{tg}(s)}(M)$. In particular, if s is a play in $M \rightarrow !N$ then we can consider s as a play in $M \rightarrow (N \circ _)^{\text{tg}(s|_{!N})}(M)$. Thus we let $\llbracket \sigma \rrbracket = \{s : s \in \llbracket \sigma \rrbracket_{\text{tg}(s|_{!N})}\}$.

It is routine to show that $\llbracket \sigma \rrbracket$ is the unique winning strategy such that $\alpha \circ \llbracket \sigma \rrbracket = (\text{id} \circ \llbracket \sigma \rrbracket) \circ \sigma$. ■

Constructing $!N$ as a limit can also be performed in the setting of Conway games [20].

C. Semantics of Sequents as Functors

Let M and N be games. An *embedding-projection pair* (or just an embedding) $M \rightarrow N$ is a pair of strategies ($\text{in} : M \rightarrow N, \text{out} : N \rightarrow M$) with $\text{out} \circ \text{in} = \text{id}$ and $\text{in} \circ \text{out} \sqsubseteq \text{id}$. We can construct a category \mathcal{G}_e of games and embeddings, with functors $i : \mathcal{G}_e \rightarrow \mathcal{G}_s$ and $p : \mathcal{G}_e \rightarrow \mathcal{G}_s^{\text{op}}$ selecting the embedding and projection components respectively.

A morphism of \mathcal{L} -models over X from (L, v) to (L', v') is a function $f : |L| \rightarrow |L'|$ such that $f(v(x)) = v'(x)$ for all $x \in X$, and for all positive atomic predicates $\vec{\phi}$ in \mathcal{L} , $I_L(\vec{\phi})(\vec{a}) = \text{tt}$ implies $I_{L'}(\vec{\phi})(f(\vec{a})) = \text{tt}$. Note that this implies that f is injective as inequality is a positive atom.

Let \mathcal{M}_X denote the category of \mathcal{L} -models over X and such morphisms. If Θ is a set of positive atoms, we can consider the full subcategory of \mathcal{M}_X consisting of only the models satisfying Θ — we denote this subcategory by \mathcal{M}_X^Θ .

A sequent $X; \Theta \vdash \Gamma$ is interpreted as a functor $\mathcal{M}_X^\Theta \rightarrow \mathcal{G}_e$. We can describe this map just for the negative formulas, defining $\llbracket P \rrbracket = \llbracket P^\perp \rrbracket$. The constructs $!, \multimap, \otimes, \&$ all extend to covariant (bi)functors on \mathcal{G}_e . This leaves only the interpretation of the atoms and quantifiers.

We set $\llbracket \phi(\vec{x}) \rrbracket(L, v) = o$ (the single move game) if $(L, v) \models \vec{\phi}(\vec{x})$ and I (the empty game) otherwise. To extend this to a functor, consider $f : (L, v) \rightarrow (L', w)$. If the truth value of $\phi(\vec{x})$ is the same in (L, v) and (L', w) , we use the identity embedding (id, id) . If the truth value of $\phi(\vec{x})$ is different, we must have $(L, v) \models \phi(\vec{x})$ and $(L', w) \models \vec{\phi}(\vec{x})$ since morphisms in \mathcal{M}_X^Θ preserve truth of positive atoms. Thus we need an embedding $I \rightarrow o$ — we can take (ϵ, ϵ) where ϵ

is the strategy containing just the empty sequence. This action is functorial.

We let $\llbracket \forall x.N \rrbracket(L, v) = \prod_{l \in |L|} \llbracket N \rrbracket(L, v[x \mapsto l])$. Suppose $f : (L, v) \rightarrow (M, w)$. We need to give an embedding $\llbracket \forall x.N \rrbracket(f) : \prod_{l \in |L|} \llbracket N \rrbracket(L, v[x \mapsto l]) \rightarrow \prod_{m \in |M|} \llbracket N \rrbracket(M, w[x \mapsto m])$. The embedding part (left to right) is given by $\langle g_m \rangle_m$ where $g_m = \epsilon$ if m is not in the image of f , and $g_m = i[\llbracket N \rrbracket(f) \circ \pi_l]_l$ if $m = f(l)$ (note in this case l is unique by injectivity of f). The projection part is given by $\langle p[\llbracket N \rrbracket(f) \circ \pi_{f(l)}]_l \rangle_l$. It is routine to verify that this forms a valid embedding/projection pair.

We give the semantics of positive formulas via duality: $\llbracket P \rrbracket = \llbracket P^\perp \rrbracket$. Proofs of positive formulas are interpreted as strategies on $\llbracket P \rrbracket \multimap o$ (Opponent plays an initial ‘dummy’ move first).

D. Semantics of Proofs as Uniform Strategies

Let $F, G : \mathcal{C} \rightarrow \mathcal{G}$. A *lax natural transformation* $F \Rightarrow G$ is a family of maps $\eta_A : F(A) \rightarrow G(A)$ with $\eta_B \circ F(f) \sqsubseteq G(f) \circ \eta_A$.

Definition 3: A *uniform winning strategy* from F to G is a lax natural transformation $F \Rightarrow G$ such that for each object A in \mathcal{C} , σ_A is winning.

We give semantics to a proof of $X; \Theta \vdash N, \Gamma$ as a uniform winning strategy $I \Rightarrow i \circ \llbracket N, \Gamma \rrbracket$ and a proof of $X; \Theta \vdash P, \Gamma$ as a uniform winning strategy $i \circ \llbracket P, \Gamma \rrbracket \Rightarrow o$. In the negative case, this is a winning strategy $\sigma_{(L, v)}$ on the game $\llbracket N, \Gamma \rrbracket(L, v)$ for each Θ -satisfying \mathcal{L} -model (L, v) over X such that if $f : (L, v) \rightarrow (M, w)$ then $\sigma_{(M, w)} \sqsupseteq i[\llbracket N, \Gamma \rrbracket(f) \circ \sigma_{(L, v)}]$.

The proof rules for the units and connectives $\otimes, \otimes, \&$ (and their duals) are interpreted by adapting the semantics from [11]: the operations on strategies lift to operations on uniform strategies. To do this we need to use vertical composition and horizontal composition (in general lax natural transformations do not compose horizontally, but they do if one of the transformations is the identity, which is enough). We also need to check that we can curry uniform strategies, which is routine.

We thus focus on the new rules. For $\text{P}_{\text{at}+}$, we start with a lax natural transformation $\llbracket p \rrbracket : \llbracket \top, \Gamma \rrbracket \Rightarrow o$ with functors mapping $\mathcal{M}_X^{\Theta, \vec{\phi}(\vec{x})} \rightarrow \mathcal{G}_e$. But for any (L, v) in $\mathcal{M}_X^{\Theta, \vec{\phi}(\vec{x})}$ we have $\llbracket \vec{\phi}(\vec{x}), \Gamma \rrbracket(L, v) = \llbracket \top, \Gamma \rrbracket(L, v)$. Hence $\llbracket p \rrbracket : \llbracket \vec{\phi}(\vec{x}), \Gamma \rrbracket \Rightarrow o$, and we take $\llbracket \text{P}_{\text{at}+}(p) \rrbracket = \llbracket p \rrbracket$.

For the rule $\text{P}_{\text{at}-}$, suppose $\llbracket p \rrbracket : I \Rightarrow \llbracket \perp, \Gamma \rrbracket$ with functors $\mathcal{M}_X^{\Theta, \vec{\phi}(\vec{x})} \rightarrow \mathcal{G}_e$. Then set $\llbracket \text{P}_{\text{at}-}(p) \rrbracket(L, v) = \epsilon$ if $(L, v) \models \phi(\vec{x})$ and $\llbracket p \rrbracket(L, v)$ if $(L, v) \models \vec{\phi}(\vec{x})$. For lax naturality, we need to check that the appropriate diagram lax commutes:

$$\begin{array}{ccc}
 I & \xrightarrow{\llbracket \text{P}_{\text{at}-}(p) \rrbracket(M, w)} & \llbracket \vec{\phi}(\vec{x}), \Gamma \rrbracket(M, w) \\
 \text{id} \downarrow & \sqsupseteq & \downarrow i[\llbracket \vec{\phi}(\vec{x}), \Gamma \rrbracket(f)] \\
 I & \xrightarrow{\llbracket \text{P}_{\text{at}-}(p) \rrbracket(L, v)} & \llbracket \vec{\phi}(\vec{x}), \Gamma \rrbracket(L, v)
 \end{array}$$

Note that if (L, v) and (M, w) agree on $\phi(\vec{x})$ then the diagram lax commutes by lax naturality of ϵ or $\llbracket p \rrbracket$. If they disagree, then we must have $(L, v) \models \phi(\vec{x})$ and $(M, w) \not\models \phi(\vec{x})$. We need to show that $\llbracket P_{\text{at-}}(p) \rrbracket(L, v) \sqsupseteq i[\llbracket \phi(\vec{x}), \Gamma \rrbracket(f) \circ \llbracket P_{\text{at-}}(p) \rrbracket(M, w)]$. To see this, note that $p[\llbracket \phi(\vec{x}), \Gamma \rrbracket(f) \circ \llbracket P_{\text{at-}}(p) \rrbracket(L, v)] = \llbracket P_{\text{at-}}(p) \rrbracket(M, w)$ as both sides map into the terminal object, so $\llbracket P_{\text{at-}}(p) \rrbracket(L, v) \sqsupseteq i[\llbracket \phi(\vec{x}), \Gamma \rrbracket(f) \circ p[\llbracket \phi(\vec{x}), \Gamma \rrbracket(f) \circ \llbracket P_{\text{at-}}(p) \rrbracket(L, v)]] = i[\llbracket \phi(\vec{x}), \Gamma \rrbracket(f) \circ \llbracket P_{\text{at-}}(p) \rrbracket(M, w)]$.

For the quantifiers, we have $\llbracket P_{\forall}(p) \rrbracket(L, v) = \text{dist} \circ \langle \llbracket p \rrbracket(L, v[x \mapsto l]) \rangle_l$ where $\text{dist} : \prod_{l \in L} \llbracket N, \Gamma \rrbracket(L, v[x \mapsto l]) \cong \llbracket \forall x. N, \Gamma \rrbracket(L, v)$. Set $\llbracket P_{\exists}(p) \rrbracket(L, v) = \llbracket p \rrbracket(L, v) \circ \pi_{v(y)} \circ \text{dist}^{-1}$. We can check that these define uniform winning strategies.

Semantics of P_{\neq} is trivial, as \mathcal{M}_X^{Θ} has no objects. For P_{ma} , we note that if $x, y \in X$ then \mathcal{M}_X^{Θ} is the disjoint sum of the categories $\mathcal{M}_X^{\Theta, x=y}$ and $\mathcal{M}_X^{\Theta, x \neq y}$ (there are no maps between these subcategories, as morphisms are injective). Thus to give a lax natural transformation $F \Rightarrow G$ with $F, G : \mathcal{M}_X^{\Theta} \rightarrow \mathcal{G}_e$ is precisely to give a pair of lax natural transformations $F|_{\mathcal{M}_X^{\Theta, x=y}} \Rightarrow G|_{\mathcal{M}_X^{\Theta, x=y}}$ and $F|_{\mathcal{M}_X^{\Theta, x \neq y}} \Rightarrow G|_{\mathcal{M}_X^{\Theta, x \neq y}}$. The two premises of P_{ma} provide these transformations. In the latter case this is direct; in the former case one must note that $\llbracket \Theta, x = y \vdash \Gamma \rrbracket = \llbracket (\Theta \vdash \Gamma)[\frac{z}{x}, \frac{z}{y}] \rrbracket \circ H$ where $H : \mathcal{M}_X^{\Theta, x=y} \rightarrow \mathcal{M}_{X/\{x, y\} \uplus \{z\}}^{\Theta[\frac{z}{x}, \frac{z}{y}]}$ and use horizontal composition.

Semantics of non-core rules involving these constructs can be given similarly.

For the exponentials, semantics of $P_{\text{ana}}, P_?, P_!$ and P_{der} use the fact that $!N$ is the final coalgebra of $X \mapsto N \otimes X$ (we can check that the anamorphism of a lax natural transformation is lax natural). For semantics of P_{con} , we use the fact that $!N$ is the carrier of a comonoid.

E. Consistency

It follows that **WS1** is consistent: if we had proofs $\vdash M$ and $\vdash M^\perp$, their denotations would be uniform winning strategies $I \Rightarrow \llbracket M \rrbracket$ and $\llbracket M \rrbracket \Rightarrow o$, which would compose to give a winning strategy on $I \Rightarrow o$ — but there are no such strategies.

IV. FULL COMPLETENESS

We now show that the core rules suffice to represent any uniform winning strategy σ on a type object provided σ is *bounded* — i.e. there is a bound on the size of plays occurring in σ . In particular, such a strategy is the semantics of a unique *core proof*. Given a sequent $X; \Theta \vdash \Gamma$, we say Θ is *lean* if it contains $x \neq y$ for all distinct x and y in X and does not contain $x \neq x$. A proof in **WS1** is *core* if it uses only core rules and has the following additional restrictions: all rules that are not P_{\neq} or P_{ma} can only be applied to conclude sequents with a lean Θ ; if P_{ma} is used to conclude $X; \Theta \vdash \Gamma$ then x and y are the least two variables in X not to be declared distinct by Θ and z is the least fresh variable.

Theorem 4: Let $X; \Theta \vdash \Gamma$ be a sequent of **WS1** and σ a bounded uniform winning strategy on $\llbracket X; \Theta \vdash \Gamma \rrbracket$. Then there is a unique core proof p of $X; \Theta \vdash \Gamma$ with $\llbracket p \rrbracket = \sigma$.

All strategies on the denotations of affine sequents are bounded. Consequently, in the affine fragment we can perform reduction-free normalisation from proofs to (cut-free) core proofs, by reification of their semantics. We thus see that all of the non-core rules are admissible (when restricted to this fragment). In particular, we can extract a syntactic cut elimination procedure from the full completeness result, as defined explicitly in [11].

The rest of this section sketches the proof of this full completeness result, and describes an extension to reify unbounded strategies as *infinitary* core proofs.

A. Uniform Choice

First, we show that in any uniform winning strategy, each component makes the same choice when the outermost connective is a \oplus or \exists (and that in the later case the value chosen by Player must be the value of some variable).

The first key observation is that if Θ is lean and $(L, v), (M, w) \in \mathcal{M}_X^{\Theta}$ then there exists an \mathcal{L} -model $(L, v) \sqcup (M, w)$ and maps $f_{(L, v), (M, w)} : (L, v) \rightarrow (L, v) \sqcup (M, w)$ and $g_{(L, v), (M, w)} : (M, w) \rightarrow (L, v) \sqcup (M, w)$. If (L, v) is an \mathcal{L} -model, define $U_{(L, v)}$ to be the elements of $|L|$ not in the image of v . Then the carrier of $(L, v) \sqcup (M, w)$ is defined to be $X \uplus U_{(L, v)} \uplus U_{(M, w)}$. The \mathcal{L} -structure validates all positive atoms, and the valuation is just inj_1 . Then the map $f_{(L, v), (M, w)}$ sends $v(x)$ to $\text{inj}_1(x)$ and $u \in U_{(L, v)}$ to $\text{inj}_2(u)$. This is an injection because Θ is lean. $g_{(L, v), (M, w)}$ is defined similarly.

The second key observation is that if $f : (L, v) \rightarrow (M, w)$ then $\sigma_{(L, v)}$ is determined entirely by f and $\sigma_{(M, w)}$. In particular, uniformity for positive strategies $\sigma : N \Rightarrow o$ requires that $\sigma_{(L, v)} \sqsubseteq \sigma_{(M, w)} \circ N(f)$ but since $\sigma_{(L, v)}$ is total, it is maximal in the ordering and so we must have $\sigma_{(L, v)} = \sigma_{(M, w)} \circ N(f)$.

Proposition 5: Suppose Θ is lean, and let $\sigma : M_1 \times M_2 \Rightarrow o$ be a uniform winning strategy. Then $\sigma = \tau \circ \pi_1$ for some uniform winning strategy $\tau : M_1 \Rightarrow o$, or $\sigma = \tau \circ \pi_2$ for some uniform winning strategy $\tau : M_2 \Rightarrow o$.

Proof: We know that each $\sigma_{(L, v)}$ is of the form $\tau_{(L, v)} \circ \pi_i$ for some $i \in \{1, 2\}$ since in the game $M_1(L, v) \times M_2(L, v) \multimap o$ we must respond to the initial Opponent-move either with a move in M_1 or a move in M_2 . But we need to check that i is uniform across components. Suppose that i is not uniform — then we have (L, v) and (T, w) with $\sigma_{(L, v)} = \tau_{(L, v)} \circ \pi_1$ and $\sigma_{(T, w)} = \tau_{(T, w)} \circ \pi_2$. Now consider $(L, v) \sqcup (T, w)$ and let k be such that $\sigma_{(L, v) \sqcup (T, w)} = \tau_{(L, v) \sqcup (T, w)} \circ \pi_k$. By our second key observation, $\sigma_{(L, v)} = \sigma_{(L, v) \sqcup (T, w)} \circ (M_1 \times M_2)(f_{(L, v), (T, w)}) = \tau_{(L, v) \sqcup (T, w)} \circ \pi_k \circ (M_1 \times M_2)(f_{(L, v), (T, w)}) = \tau_{(L, v) \sqcup (T, w)} \circ M_k(f_{(L, v), (T, w)}) \circ \pi_k$. But since $\sigma_{(L, v)}$ is of the form $\tau_{(L, v)} \circ \pi_1$, we must have $k = 1$. But we can reason similarly using $\sigma_{(T, w)}$ and $g_{(L, v), (T, w)}$ and discover that $k = 2$. This is a contradiction.

Thus there is some i such that each $\sigma_{(L, v)}$ can be decomposed into $\tau_{(L, v)} \circ \pi_i$. We know that τ is lax natural as it is equal to $\sigma \circ \langle \text{id}, \epsilon \rangle$ if $i = 1$ or $\sigma \circ \langle \epsilon, \text{id} \rangle$ if $i = 2$. ■

Proposition 6: Suppose Θ is lean, and let $\sigma : \forall x.M \Rightarrow o$ be a uniform winning strategy. Then there exists a unique variable $y \in X$ and uniform winning strategy $\tau : M \Rightarrow o$ such that $\sigma_{(L,v)} = \tau_{(L,v)} \circ \pi_{v(y)}$.

Proof: First, we show that given any \mathcal{L} -model (L, v) there is some y with $\sigma_{(L,v)} = \tau_{(L,v)} \circ \pi_{v(y)}$. Suppose for contradiction that $\sigma_{(L,v)} = \tau_{(L,v)} \circ \pi_u$ for some $u \in U_{(L,v)}$. Build the \mathcal{L} -model $L' = X \uplus \{a, b\} \uplus U_{(L,v)}$ with the obvious valuation and validating all positive atoms. Then there are two distinct maps $m_1, m_2 : (L, v) \rightarrow L'$ both behaving in the obvious way, except m_1 sends u to a , and m_2 sends u to b . By using similar reasoning to the above, we can use m_1 to show that $\sigma_{L'} = \tau_{L'} \circ \pi_a$ and m_2 to show that $\sigma_{L'} = \tau_{L'} \circ \pi_b$, which is a contradiction.

Similarly, we can show that y is uniform across all (L, v) . Finally we need to check that $\tau : M \Rightarrow o$ is lax natural: we can show that it is $\sigma \circ \rho$ where $\rho : M \Rightarrow \forall x.M$ is defined by $\langle g_m \rangle_m$ where $g_m = \epsilon$ if $m \neq v(y)$ and $g_{v(y)} = \text{id}$. ■

B. Reification Procedure

We next define the function *reify* from uniform winning strategies on a formula-object to proofs. Informally, *reify* is a semantics-guided proof-search procedure. It is defined by case analysis on the head of Γ , by induction on a compound measure involving the size of the strategy, the number of pairs of free variables that are not declared distinct by Θ , and a further measure that depends on the nature of the head formula. Informally, if Θ is not lean:

- If Θ contains $x \neq x$ we use P_{\neq} and halt.
- Otherwise, we consider the least two variables $x, y \in X$ that are not declared distinct by Θ and split the family into those models that identify x and y , and those that do not. In the former case, we can substitute fresh z for both x and y . We then apply the inductive hypothesis to both halves and apply P_{ma} .

If Θ is lean, then:

- If the head formula is $\mathbf{1}$, then σ is the unique (empty) strategy on this game. The head formula cannot be $\mathbf{0}$, since there are no uniform winning strategies on this game.
- If the head formula is \perp , then we may use the P_{\perp}^- , P_{\perp}^{\otimes} , P_{\perp}° rules to shorten the tail of the sequent until it is a single positive formula P or empty. The latter case is impossible, as there are no winning strategies on this game. In the former case, we can obtain a uniform strategy on P by removing the first move in σ . We can then proceed inductively using P_{\perp}^+ . If the head formula is \top we may proceed similarly (but in this case an empty tail is possible, in which case we use P_{\top}).
- If the head formula is a positive atom $\overline{\phi(\vec{x})}$ then we must have $\overline{\phi(\vec{x})}$ in Θ , as otherwise there can be no uniform winning strategies on $\llbracket \Gamma \rrbracket$ (since some games in that family have no winning strategies). Thus we can proceed inductively and apply $P_{\text{at}+}$.
- If the head formula is a negative atom $\phi(\vec{x})$ then we can split the family σ into those models that satisfy $\phi(\vec{x})$ and

those that do not. All strategies in the latter group must be empty, as there are no moves to play. All strategies in the former group form a uniform strategy on $\llbracket \Theta, \phi(\vec{x}) \vdash \perp, \Gamma \rrbracket$ and we can proceed inductively using $P_{\text{at}-}$.

- If the outermost connective of the head formula is $\&$, \otimes , \circ , \triangleleft , $!$, $?$ or \forall , then we may reverse the associated rule to decompose the head formula.
- If the outermost connective of the head formula is \oplus — i.e. $\Gamma = P_1 \oplus P_2, \Gamma'$ — then σ must play its first move in either $\llbracket P_1 \rrbracket$ or $\llbracket P_2 \rrbracket$ for each model. By Proposition 5, it must choose the same component i for each valuation. This yields a uniform strategy on $\llbracket \vdash P_i, \Gamma' \rrbracket$ and we can proceed inductively using the rule $P_{\oplus i}$. If the outermost connective is \wp , then similar reasoning applies.
- If $\Gamma = \exists x.P, \Gamma'$ then each component of σ must choose an x in the appropriate model, and give a corresponding substrategy. By Proposition 6, the value of x must be $v(y)$ for some unique $y \in X$, and further y must be constant throughout all components. This can be used to construct a uniform strategy on $\llbracket \Theta \vdash P[y/x], \Gamma' \rrbracket$, we can apply the inductive hypothesis and use the P_{\exists}^y rule.

C. Termination

We next argue for termination of our procedure. Intuitively, the full completeness procedure first breaks down the head formula until it is \perp or \top . It then uses the core elimination rules to compose the tail into (at most) a single formula. These steps do not increase the size of the strategy. Finally, the head is removed using P_{\perp}^+ or P_{\top}^- , strictly reducing the size of the strategy. If Θ is not lean, the number of distinct variable pairs that are not declared distinct in Θ is reduced by using P_{ma} .

Formally, we can see this as a lexicographical ordering of four measures on $\sigma, X, \Theta, \Gamma$:

- The most dominant measure is the length of the longest play in σ .
- The second measure is the length of Γ as a list if the head of Γ is \perp or \top , and ∞ otherwise.
- The third measure is the size of the head formula of Γ .
- The fourth measure is

$$|\{(x, y) \in X \times X : x \neq y \wedge x \neq y \notin \Theta\}|$$

If Θ is lean:

- If $\Gamma = \perp, P$ or \top, N then the first measure decreases in the call to the inductive hypothesis.
- Otherwise, if $\Gamma = A, \Gamma'$ with $A \in \perp, \top$ the first measure does not increase and the second measure decreases.
- If $\Gamma = A, \Gamma'$ with $A \notin \{\perp, \top\}$, the first measure does not increase and either the second or third measure decreases.

If Θ is not lean and the P_{ma} rule is applied, in the call to the inductive hypotheses the first three measures stay the same and the fourth measure decreases.

Thus, the inductive hypothesis is used with a smaller value in the compound measure on $\mathbb{N} \times \mathbb{N} \cup \{\infty\} \times \mathbb{N} \times \mathbb{N}$ ordered lexicographically.

D. Proof Normalisation

Our full completeness theorem ensures that in the affine fragment, we can normalise a proof to a core proof. We can in fact do something similar in full WS1, except in this case the resulting normal form will be an *infinitary* core proof — i.e. a proof using the core rules that may be infinite. The collection of such proofs is the final coalgebra of the core proof rules. Using this coalgebraic formulation, we can give semantics of infinitary core proofs as uniform total strategies (but these strategies need not be winning).

Theorem 7: Let $X; \Theta \vdash \Gamma$ be a sequent of WS1 and σ a uniform total strategy on $\llbracket X; \Theta \vdash \Gamma \rrbracket$. Then σ is the denotation of a unique infinitary core proof.

To show this, we use our reification procedure as described above. The termination of this procedure depends on boundedness of σ ; for unbounded σ we can use a coalgebraic formulation to construct an infinitary core proof.

Thus the infinitary core proofs correspond precisely to the uniform total strategies. It is possible to introduce a constraint to say when an infinitary core proof is winning by emulating the definition of play restrictions in strategies, but we will not pursue this here.

We can thus perform proof normalisation for arbitrary proofs in WS1, with the caveat that the resulting normal form is an infinitary core proof: we take the semantics of a proof as a uniform winning strategy, and then generate the corresponding infinitary core proof. This normal form will have the same semantics as the proof we started with. Indeed, two proofs have the same semantics if and only if they have the same infinitary normal form. Similarly, for our language embedding, two programs are observationally equivalent if and only if their proof translations have the same (infinitary) normal form.

V. FURTHER DIRECTIONS

In this paper, we have given some simple examples of “stateful proofs”. We aim to investigate further examples in more expressive logics, and to specify additional properties of programs in more powerful programming languages (such as the games-based language in e.g. [10]). Further extensions to our work which may be required in order to do so include:

- WS1 has been presented as a general first-order logic. By adding axioms, we may specify and study programs in particular domains. For example, can we derive a version of Peano Arithmetic in which proofs have constructive, stateful content (cf [21])?
- WS1 and its semantics may be extended with function-symbols. Establishing full completeness then becomes a non-trivial unification problem.
- Extension with *propositional variables* (and potentially, second-order quantification) would allow generic “copy-cat strategies” to be captured. On the programming side, this would allow us to model languages with polymorphism.

- We have interpreted the exponentials as greatest fixpoints. Adding general inductive and coinductive types, as in μLJ [17] would extend WS1 to a rich collection of datatypes (including finite and infinite lists, for example).

ACKNOWLEDGMENTS

The authors would like to thank Pierre Clairambault for useful discussion and anonymous reviewers for valuable comments. This work was supported by the (UK) EPSRC grant EP/HO23097.

REFERENCES

- [1] S. Abramsky and R. Jagadeesan, “Games and full completeness for multiplicative linear logic,” *J. Symb. Logic*, vol. 59, no. 2, pp. 543–574, 1994.
- [2] S. Abramsky and G. McCusker, “Linearity, Sharing and State: a fully abstract game semantics for Idealized Algol with active expressions: Extended Abstract,” *Electronic Notes in Theoretical Computer Science*, vol. 3, pp. 2 – 14, 1996, linear Logic 96 Tokyo Meeting.
- [3] J. Laird, “A calculus of coroutines,” *Theoretical Computer Science*, vol. 350, no. 2-3, pp. 275 – 291, 2006, automata, Languages and Programming: Logic and Semantics (ICALP-B 2004).
- [4] J. M. E. Hyland and C.-H. L. Ong, “On full abstraction for PCF: I, II, and III,” *Inf. Comput.*, vol. 163, no. 2, pp. 285–408, 2000.
- [5] J. C. Reynolds, “The essence of Algol,” in *Proceedings of the 1981 International Symposium on Algorithmic Languages*. North-Holland, 1981, pp. 345–372.
- [6] A. Blass, “A game semantics for linear logic,” *Annals of Pure and Applied Logic*, vol. 56, no. 1-3, pp. 183 – 220, 1992.
- [7] P.-L. Curien, “On the symmetry of sequentiality,” in *Mathematical Foundations of Computer Science*, ser. LNCS. Springer, 1993, no. 802.
- [8] F. Lamarche, “Games semantics for full propositional linear logic,” in *Proceedings, Tenth Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1995, pp. 464–473.
- [9] M. Hyland, “Game semantics,” in *Semantics and Logics of Computation*, ser. Publications of the Newton Institute, P. Dybjer and A. M. Pitts, Eds. Cambridge University Press, 1997, pp. 131–184.
- [10] J. Longley, “Some programming languages suggested by game models,” in *Proceedings of MFPS XXV*. ENTCS, 2009.
- [11] M. Churchill and J. Laird, “A logic of sequentiality,” in *Computer Science Logic*, ser. Lecture Notes in Computer Science, A. Dawar and H. Veith, Eds. Springer Berlin / Heidelberg, 2010, vol. 6247, pp. 215–229, 10.1007/978-3-642-15205-4-19.
- [12] J. Laird, “A categorical semantics of higher order store,” in *Proceedings, 9th Conference on Category Theory and Computer Science, CTCS 2002, Electronic Notes in Theoretical Computer Science*. Elsevier, 2002.
- [13] J.-Y. Girard, “Locus solum,” *Mathematical Structures in Computer Science*, vol. 11, no. 3, pp. 301–506, 2001.
- [14] G. Japaridze, “Introduction to Computability Logic,” *Annals of Pure and Applied Logic*, vol. 123, pp. 1 – 99, 2003.
- [15] J. R. B. Cockett, G. S. H. Cruttwell, and K. Saff, “Combinatorial game categories,” submitted to *Mathematical Structures in Computer Science*, March 2010.
- [16] O. Laurent, “Game semantics for first-order logic,” *Logical Methods in Computer Science*, vol. 6, no. 4, p. 3, Oct. 2010.
- [17] P. Clairambault, “Least and greatest fixpoints in game semantics,” in *Foundations of Software Science and Computational Structures*, ser. Lecture Notes in Computer Science, L. de Alfaro, Ed. Springer Berlin / Heidelberg, 2009, vol. 5504, pp. 16–31, 10.1007/978-3-642-00596-1-3.
- [18] A. Schalk, “What is a categorical model of linear logic,” Tech. Rep., 2004.
- [19] R. Cartwright, P.-L. Curien, and M. Felleisen, “Fully abstract semantics for observably sequential languages,” *Information and Computation*, vol. 111, no. 2, pp. 297–401, 1994.
- [20] P.-A. Mellès, N. Tabareau, and C. Tasson, “An explicit formula for the free exponential modality of linear logic,” in *ICALP ’09: Proceedings of the 36th International Colloquium on Automata, Languages and Programming*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 247–260.
- [21] T. Coquand, “A semantics of evidence for classical arithmetic,” *Journal of Symbolic Logic*, vol. 60, pp. 325–337, 1995.