Please scroll down to view the document.
Imperative Programs as Proofs via Game Semantics

Martin Churchill\textsuperscript{b}, Jim Laird\textsuperscript{a}, Guy McCusker\textsuperscript{a}

\textsuperscript{a}University of Bath
\textsuperscript{b}Swansea University

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.

Keywords: game semantics, full completeness, history-sensitive strategies, sequentiality

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. 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, in which they typically denote “history-sensitive” strategies — strategies which may break the constraints of innocence or history-freeness 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, WS\textsubscript{1}, 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 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 λι-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 L-structure — which must be uniform across 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, $A^\omega$ the set of infinite sequences over A, and $\epsilon$ the empty sequence. We write $s \sqsubseteq t$ if s is a prefix of t, and $s \sqsubseteq t$ if s is a strict (finite) prefix of (possibly infinite) t. If $X \subseteq A^*$, write $X = \{s \in A^\omega : \forall t \sqsubseteq s, t \in X\}$.

**Definition** A game is a tuple $(M_A, \lambda_A, b_A, P_A, W_A)$ where

- $M_A$ is a set of moves
- $\lambda_A : M_A \rightarrow \{O, P\}$
  - We call m an O-move if $\lambda_A(m) = O$ and a P-move if $\lambda_A(m) = P$.
- $b_A \in \{O, P\}$ specifies a starting player
  - We call $s \in M^*_A$ alternating if s starts with a $b_A$-move and alternates between O-moves and P-moves. Write $M^*_A$ for the set of such sequences.
Whenever \( s \in A \) is a strategy on a game \( \sigma \), these come in dual pairs, determined by involutive negation.

2.1. Connectives

For finite plays, the last player to play a move wins: let \( W_A^* = W_A \cup E_A \) where \( E_A \) is the set of plays that end in a P-move. We will call a game \( A \) negative if \( b_A = O \) and positive if \( b_A = P \). We write \( A, B, C, \ldots \) for arbitrary games; \( L, M, N, \ldots \) for arbitrary negative games and \( P, Q, R, \ldots \) for arbitrary positive games.

**Definition** If \( A \) is a game, we define its negation by changing its polarity, and swapping its Player/Opponent labelling. Define \( \neg : \{O, P\} \to \{O, P\} \) by \( \neg(O) = P \) and \( \neg(P) = O \).

\[ A^\perp = (M_A, \neg \lambda_A, \neg b_A, P_A, \overline{P_A} - W_A). \]

Negation is evidently an involutive bijection between negative and positive games.

**Definition** A strategy \( \sigma \) for a game \( (M_A, \lambda_A, b_A, P_A, W_A) \) is a subset of \( P_A \) (a set of traces) satisfying:

- If \( sa \in \sigma \), then \( \lambda_A(a) = P \)
- If \( sab \in \sigma \), then \( s \in \sigma \)
- If \( sa, sb \in \sigma \), then \( a = b \)
- If \( \sigma = \emptyset \) then \( b_A = P \), and if \( \epsilon \in \sigma \) then \( b_A = O \).

We say a strategy \( \sigma \) is bounded if \( \exists k \in \mathbb{N} \forall s \in \sigma \|s\| \leq k \); in which case we write \( \text{depth}(\sigma) \) for the smallest such \( k \) (the length of the longest play in \( \sigma \)).

**Definition** A strategy on a game \( A \) is total if it is nonempty and whenever \( s \in \sigma \) and \( sa \in P_A \), there is some \( b \in M_A \) such that \( sab \in \sigma \). A total strategy \( \sigma \) is winning if whenever \( s \in \overline{P_A} \) and all prefixes of \( s \) ending in a P-move are in \( \sigma \), then \( s \in W_A \).

2.1. Connectives

We next describe operations on games, which will correspond to connectives in our logic. These come in dual pairs, determined by involutive negation.

First, some notation. If \( X \) and \( Y \) are sets, let \( X + Y = \{in_1(x) : x \in X\} \cup \{in_2(y) : y \in Y\} \). We use standard notation \([f, g]\) for copairing. If \( s \in (X + Y)^* \) or \( s \in (X + Y)^\omega \) then \( s|_1 \) is the subsequence of \( s \) consisting of elements of the form \( in_1(z) \). If \( X_1 \subseteq X^* \) and \( Y_1 \subseteq Y^* \) let \( X_1\|Y_1 = \{s \in (X + Y)^* : s|_1 \in X_1 \land s|_2 \in Y_1\} \). If \( X_1 \subseteq X^\omega \) and \( Y_1 \subseteq Y^\omega \) let \( X_1\|Y_1 = \{s \in (X + Y)^\omega : s|_1 \in X_1 \land s|_2 \in Y_1\} \).

**Empty Game.** We define a negative game with no moves:

\[ 1 = (\emptyset, \emptyset, O, \{\epsilon\}, \emptyset). \]

There is one strategy on \( 1 \) given by \( \{\epsilon\} \), and this strategy is total (and winning, as \( \overline{P_1} \) is empty).

There is one strategy, \( \emptyset \), on the empty positive game \( 0 = 1^\perp \). This strategy is not total (intuitively, it is Player’s turn to play first but he has no moves to play).
**One-move Game.** We write $\perp$ for the negative game with a single move $q$ and maximal play consisting of $q$:
$$\perp = (\{q\}, \{q \mapsto O\}, O, \{\epsilon, q\}, \emptyset).$$

There is a single strategy $\{\epsilon\}$ on $\perp$: this is not total.

We write $\top$ for the positive game with a single move, $\perp\!\!\perp$. There are two strategies on $\top$: $\emptyset$ (which is evidently not total) and $\{q\}$ which is total (and thus, trivially winning).

**Disjoint Union.** The negative game $L \& N$ is played over the disjoint union of the moves of $L$ and $N$: a play in this game is either a (tagged) play in $L$ or a (tagged) play in $N$. A play is $P$-winning if it is a $P$-winning play from $L$ or a $P$-winning play from $N$. Thus, on Opponent’s first move he chooses to play either in $L$ or $N$, and thereafter play remains in that component. Formally, define
$$L \& N = (M_L + M_N, [\lambda_L, \lambda_N], O, P_L +^* P_N, \{in^*_W(s) : s \in W_L\} \cup \{in^*_W(s) : s \in W_N\})$$

where $X_1 +^* Y_1 = \{s \in X_1||Y_1 : s_{|1} = \epsilon \lor s_{|2} = \epsilon\}$ if $X_1 \subseteq X^*$ and $Y_1 \subseteq Y^*$, and if $s \in X^*_i$ (resp. $X^*_i$) we write $in^*_i(s)$ (resp. $in^*_i$) for the corresponding sequence in $(X_1 + X_2)^*$ (resp. $(X_1 + X_2)^*$). A (winning) strategy on $L \& N$ corresponds to a pairing of a (winning) strategy on $L$ with a (winning) strategy on $N$ — hence the identification of this connective with the “with” of linear logic.

Similarly, the positive game $Q \oplus R = (Q^\perp \& R^\perp)^\perp$ corresponds to a disjoint union of plays from $Q$ and $R$ where Player’s first move constitutes a choice to play either in $Q$ or $R$. An infinite play in $Q \oplus R$ is $P$-winning if it is $P$-winning in the relevant component. Thus a winning strategy on $Q \oplus R$ corresponds to either a winning strategy on $Q$ or a winning strategy on $R$.

We may form any set-indexed conjunctions and disjunctions in this way. Let $X$ be a set and $\{N_x : x \in X\}$ a family of negative games indexed by $X$. We define the game $\prod_{x \in X} N_x$ by
$$(\sum_{x \in X} M_{N_x}, in_x(m) \mapsto \lambda_{N_x}(m), O, \{in^*_W(s) : x \in X, s \in P_{N_x}\}, \{in^*_W(s) : x \in X, s \in W_{N_x}\}).$$

If $\{Q_x : x \in X\}$ is a family of positive games then $\bigoplus_{x \in X} Q_x = (\prod_{x \in X} N_x^\perp)^\perp$.

**Symmetric Merge.** If $L$ and $N$ are negative games, a play in the negative game $L \otimes N$ is an interleaving of a play in $L$ with a play in $N$. Define
$$L \otimes N = (M_L + M_N, [\lambda_L, \lambda_N], O, (P_L||P_N) \cap M_{L \otimes N}^\otimes, \{s \in P_{L \otimes N} : s_{|1} \in W_L^* \land s_{|2} \in W_N^*\}).$$

The fact that the play restricted to each component must be alternating, that the play overall must be alternating, ensures that only Opponent may switch between components. This operation may be used to interpret the “times” of linear logic \cite{7}. An infinite play in $L \otimes N$ is $P$-winning if its restriction to $L$ is $P$-winning and its restriction to $N$ is $P$-winning.

Similarly, if $Q$ and $R$ are positive games, plays in the positive game $Q \bowtie R = (Q^\perp \otimes R^\perp)^\perp$ consist of interleavings of plays in $Q$ and $R$ in which Player may switch between the two components. An infinite play in $Q \bowtie R$ is $P$-winning if its restriction to $Q$ is $P$-winning or its restriction to $R$ is $P$-winning.
**Left Merge.** Let $A$ be a game of polarity $a$ (positive or negative), and $N$ a negative game. The game $A \odot N$ has polarity $a$: a play in this game is an interleaving of a play in $A$ with a play in $N$ such that the first move, if any, is in $A$. An infinite play in $A \odot N$ is $P$-winning if both of its restrictions are $P$-winning. Formally, define

$$A \odot N = (M_A + M_N, [\lambda_A, \lambda_N], b_A, (P_A \lVert L_P N) \cap M^\#_{A \odot N}, \{s \in P^\#_{A \odot N} : s|_1 \in W_A^1 \land s|_2 \in W_N^1\}).$$

where $X_1 \lVert Y_1 = \{s \in X_1 \lVert Y_1 : s|_1 = \epsilon \Rightarrow s|_2 = \epsilon\}$. Observe that it is Opponent who switches between components: if $A$ is negative then $A \odot N$ consists of the plays in $A \odot N$ which start in $A$ (or are empty). This connective on games, the *sequoid*, was introduced in [21] and its properties can be used to model stateful effects [21, 24].

If $Q$ is a positive game, the game $A \triangleright Q = (A^\perp \odot Q^\perp)^\perp$ has the same polarity as $A$, and consists of interleavings of a play in $A$ and a play in $Q$, starting in $A$ and with Player switching between components and winning an infinite play if he wins in either $A$ or $Q$.

**Exponentials.** Let $N$ be a negative game. The negative game $!N$ consists of countably many copies of $N$, tagged with natural numbers. A play over $!N$ is an interleaving of plays in each copy, such that any move in $N_{i+1}$ is preceded by a move in $N_i$. An infinite play is winning just if it is winning in each component. Define

$$!N = (M_N \times N, \lambda_N \circ \pi_1, \{s : \forall i. s|_i \in P_N \land s|_i = \epsilon \Rightarrow s|_{i+1} = \epsilon\}, \{s : \forall i. s|_i \in W_N^1\}).$$

As with the tensor, there is an implicit switching condition: only Opponent can open new copies and switch between copies. This operation may be used to interpret the “of course” of linear logic [15].

Dually, if $Q$ is a positive game, $?Q = (!Q^\perp)^\perp$ is the game consisting of an infinite number of copies of $Q$, where Player can spawn new copies and switch between them. An infinite play in $?Q$ is winning if it is winning in at least one component.

2.1.1. *Derived Connectives*

We shall also make use of the following derived operations:

**Lifts.** We can use left merge to add a single move at the beginning of a game. If $N$ is a negative game, a play in the positive game

$$\downarrow N = \top \odot N$$

consists of a play in $N$ prefixed by an extra $P$-move. A strategy on $\downarrow N$ is either $\emptyset$ or corresponds to a strategy on $N$. A winning strategy on $\downarrow N$ corresponds to a winning strategy on $N$. If $P$ is a positive game, a play in the negative game

$$\uparrow P = \bot \triangleright P$$

consists of a play in $P$ prefixed by an extra $O$-move. A (winning) strategy on $\uparrow P$ corresponds to a (winning) strategy on $P$. 

6
Affine Implication. If $M$ and $N$ are negative games, we may define

$$M \rightarrow N = N \triangleleft M^\perp.$$ 

A play in $M \rightarrow N$ consists of a play in $N$ interleaved with a play in $M^\perp$ (an ‘input version’ of $M$), starting in $N$. It is winning if its restriction to $N$ is $P$-winning or its restriction to $M^\perp$ is $P$-winning (i.e. its restriction to $M$ is $O$-winning), agreeing with [15].

2.1.2. Isomorphisms of Games

Given two games $A$ and $B$, we say that $A$ and $B$ are forest isomorphic if $b_A = b_B$ and there is a bijection from $P_A$ to $P_B$ which is monotone with respect to the prefix order, and restricts to a bijection on the $P$-winning plays. Some forest isomorphisms between games are given in Figure 1. Each isomorphism $M \cong N$ gives rise to winning strategies $M \rightarrow N$ and $N \rightarrow M$, which are mutually inverse. Thus, winning strategies on $M$ are in bijective correspondence with winning strategies on $N$.

![Figure 1: Some Characteristic Isomorphisms of Games](image)

2.2. Imperative Objects as Strategies

We may model higher-order programming languages with imperative features by interpreting types as games and programs as strategies. (Such a semantics of a full object-oriented language, using essentially the notion of game described here, is described in [35].) Here, we illustrate the capacity of our games and strategies to represent imperative objects by describing a strategy with the behaviour of a Boolean reference cell, on a game corresponding to the type of imperative Boolean variables — essentially the cell strategy first described, for a different notion of game, in [5]. (We will later see how this strategy can be represented as a proof in our logic.)
Let $B = \perp \otimes \top$ be the (negative) game of “Boolean output” — this has one initial Opponent-move $q$ and two possible Player responses, representing True or False. Let $Bi = (\perp \& \perp) \otimes \top$ be the (negative) game of “Boolean input” which has two starting Opponent-moves $in(tt)$ and $in(ff)$ and one possible response to this, $ok$. The game $!(Bi \& B)$ represents the type of a Boolean variable — it is a product of a write method which accepts a Boolean input and a read method which on interrogation produces a Boolean output, under an exponential which allows these methods to be used arbitrarily many times.

The strategy cell on this game represents a reference cell which accepts Boolean input on the left, and returns the last value written to it as output on the right (we assume it is initialised with $ff$). For readability, we will omit the tags on the product and the exponential (since they can be inferred).

$$
!(Bi \& B)
\begin{array}{c|c|c}
q & O & P \\
ff & O & P \\
in(tt) & O & P \\
ok & q & o \\
tt & P & P \\
\end{array}
$$

In contrast with the history-free strategies which denote proofs of linear logic in the model of [3], this strategy is history-sensitive — the move prescribed by the strategy depends on the entire play so far. It is this property which allows the state of the object to be described implicitly, as in [5].

3. The Logic WS1

3.1. Formulas of WS1

The formulas of WS1 are based on first-order linear logic, with some additional connectives, and subject to a notion of polarity. A first-order language consists of:

- A collection of complementary pairs of predicate symbols $\phi$ (negative) and $\overline{\phi}$ (positive), each with an arity in $\mathbb{N}$ such that $ar(\phi) = ar(\overline{\phi})$. This must include the binary symbol $=$ (negative), and we write $\neq$ for its complement.
- A collection of function symbols, each with an arity.

The negative and positive formulas of WS1 over $\mathcal{L}$ are defined by the following grammar. $M, N$ range over negative formulas and $P, Q$ over positive formulas; variables range over some global set $\mathcal{V}$.

$$
\begin{array}{c|c|c|c|c}
M, N := & 1 & \bot & \phi(\overline{\phi}) & N \triangleleft P \\
M \otimes N & M \otimes N & N \otimes P & !N \\
\forall x. N & M \& N & !N \\
P, Q := & 0 & \top & \overline{\phi}(\overline{\overline{\phi}}) & P \triangleleft Q \\
P \& Q & P \otimes Q & P \otimes Q & ?P \\
\exists x. P & P \oplus Q & P \oplus Q & ?P \\
\end{array}
$$
Here, $s$ ranges over $L$-terms, $x$ over variables, and $\phi(\overrightarrow{s})$ over $n$-ary predicates $\phi$ applied to a tuple of terms $\overrightarrow{s} = (s_1, \ldots, s_n)$.

The involutive negation operation $(\cdot)^+$ sends negative formulas to positive ones and vice versa by exchanging each atom, unit or connective for its dual — i.e., $\top$ for $0$, $\bot$ for $\top$, $\forall$ for $\exists$, $\exists$ for $\forall$, $\land$ for $\lor$ and $\lor$ for $\land$.

3.1.1. Interpreting Formulas as Games

We may interpret each positive formula as a positive game, and each negative formula as a negative game, by fixing a truth assignment for the atomic formulas via a standard notion of first-order structure.

**Definition** An $L$-structure $L$ is a set $|L|$ together with an interpretation function $I_L$ sending:

- each predicate symbol (with arity $n$) to a function $|L|^n \to \{\text{tt, ff}\}$ such that $I_L(\phi)(\overrightarrow{a}) \neq I_L(\overrightarrow{\phi})$ for all $\overrightarrow{a}$ and $I_L(\neg)(a, b) = \text{tt}$ if $a = b$;
- each function symbol $f$ (with arity $n$) to a function $I_L(f) : |L|^n \to |L|$.

For any $X \subseteq \mathcal{V}$, an $L$-model over $X$ is a pair $(L, v)$ where $L$ is an $L$-structure and $v : X \to |L|$ a valuation function, yielding an assignment of truth values to all atomic formulas with variables in $X$.

Given a $L$-model $(L, v)$ over $X$, we may interpret each formula $A$ with free variables in $X$ as a game $[A][L, v]$ in as follows:

- Each of the units and connectives $\land, \lor, \lnot, \lnot, 0, 1, \top, \bot, !, ?, \oplus$ is interpreted as the corresponding operation on games from Section 2.1 lifted to an action on families of games.
- Positive atoms which are assigned true in $(L, v)$ are interpreted as the game with a single (Player) move ($\top$); positive atoms which are assigned false are interpreted as the game with no moves ($0$). Conversely, negative atoms which are assigned true in $(L, v)$ are interpreted as the empty game ($1$), whilst negative atoms which are assigned false are interpreted as the game with a single Opponent move ($\bot$).
- Quantifiers are interpreted as additive conjunctions and disjunctions over the domain of $L$ — i.e., $[\forall x.N](L, v) = \prod_{x \in |L|}[N](L, v[x \mapsto l])$ and $[\exists x.P](L, v) = \bigoplus_{x \in |L|}[P](L, v[x \mapsto l])$. In the case of $\forall x.N$, this is equivalent to Opponent choosing an $x \in |L|$ and play proceeding in $N(x)$. In the case of $\exists x.P$, this is equivalent to Player choosing an $x \in |L|$ and play proceeding in $P(x)$.

Note that $[A^+] = [A]^+$.

3.2. Proofs

A proof of a formula $\vdash A$ will be interpreted as a uniform family of winning strategies on $[A][L, v]$ for each $(L, v)$. We will formalise this interpretation (and, importantly, the meaning of “uniformity”) in Section 6, but with this in mind, we can define proof rules for WS1. A sequent of WS1 is of the form $X; \Theta \vdash \Gamma$ where $X \subseteq \mathcal{V}$, $\Theta$ is a set of positive atomic formulas and $\Gamma$ 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. For brevity, let \( \Phi \) range over \( X; \Theta \) contexts.

We shall interpret such a sequent as a (family of) dialogue games by interpreting the comma operator in \( \Gamma \) as left-associative left-merge (i.e. either \( \odot \) or \( \lnot \) depending on the polarity of the right-hand operand), so that the first move must occur in the first element (or head formula) of \( \Gamma \). For example, if \( M, N \) are negative formulas and \( P, Q \) positive formulas, the sequent

\[ \vdash M, P, Q, N \]

is semantically equivalent to

\[ \vdash (\lnot P \lnot Q) \odot N. \]

Thus, in the game interpretation of a sequent \( \Gamma \) the first move must occur in the first (or head) formula of \( \Gamma \).

The derivation rules for proofs are partitioned into core rules and other rules. Here \( M, N \) range over negative formulas, \( P, Q \) over positive formulas, \( \Gamma, \Delta \) over lists of formulas, \( \Gamma^* \) over non-empty lists of formulas and \( \Gamma^+, \Delta^+ \) over lists of positive formulas.

### 3.2.1. Core Rules

Each \( n \)-ary connective \( \odot \) of WS1 is associated with core introduction rules which introduce that connective in the head position of a sequent: they conclude \( \Phi \vdash (A_1, \ldots, A_n), \Gamma \) from some premises. These rules are given in Figure 2. These core introduction rules are all additive (by contrast to linear logic: note in particular the difference with respect to the \( \odot \) introduction rule).

![Figure 2: Core Introduction Rules for WS1](image)

We may interpret each of the core introduction rules with respect to \((L, v)\) as follows:

- The interpretation of \( P_1 \) is the unique total strategy on the game \( 1, \Gamma \) (where it is Opponent’s turn to start, but there are no moves for him to play since the first move must take place in the empty game \( 1 \)).
The interpretation of $P_\top$ is the unique total strategy on the game $\top$, where Player plays a move and the game is over.

The interpretation of the unary rule $P_\exists$ is the identity function, as the game denoted by the conclusion is the same game as that denoted by the premise. The interpretation of $P_\forall$ is similar.

For $P_{\otimes 1}$ we note that given strategies $\sigma : M, \Gamma$ and $\tau : N, \Gamma$ we can construct a strategy on $M \otimes N, \Gamma$ which plays as $\sigma$ if Opponent’s first move is in $M$, and as $\tau$ if Opponent’s first move is in $N$.

Similarly, for $P_{\otimes 2}$ we note that given strategies $\sigma : M, N, \Gamma$ and $\tau : N, M, \Gamma$ we can construct a strategy on $M \otimes N$, $\Gamma$ which plays as $\sigma$ if Opponent’s first move is in $M$, and as $\tau$ if Opponent’s first move is in $N$. Here we are making use of the isomorphism $M \otimes N \cong (M \otimes N) \otimes (N \otimes M)$ — each play in $M \otimes N$ must either start in $M$ (and thus be a play in $M \otimes N$) or in $N$ (and thus be a play in $N \otimes M$). Thus, WS1 commits to a particular interpretation of $\otimes$, rather than an arbitrary monoidal structure.

For $P_{\oplus 1}$ we note that given a strategy $\sigma : P, \Gamma$ we can construct a strategy on $P \oplus Q, \Gamma$ with Player choosing to play his first move in $P$ and thereafter playing as $\sigma$. For $P_{\oplus 2}$ Player can play his first move in $Q$ and then play as the given strategy.

Similarly, for the $P_{O}$ rules, we note that in a strategy on $P \odot Q, \Gamma$ Player may choose to either play his first move in $P$ (requiring a strategy on $P, Q, \Gamma$) or in $Q$ (requiring a strategy on $Q, P, \Gamma$).

The interpretation of $P_+^\top$ uses the observation that total strategies on $\bot, P = \top P$ are in correspondence with total strategies on $P$. Similarly, the interpretation of $P_\top^\bot$ uses the observation that total strategies on $\top, N = \bot N$ are in correspondence with total strategies on $N$.

For $P_{\exists-}$, we know that $\phi(\vec{s})$, $\Gamma$ is interpreted by $1, \Gamma$ if $(L, v) \models \phi(\vec{s})$ and by $\bot, \Gamma$ if $(L, v) \not\models \phi(\vec{s})$. In the former case, there are no moves to respond to, so we only need to consider the case when $(L, v) \models \phi(\vec{s})$.

For $P_{\exists+}$, we can only provide a family of strategies on a game whose first move is in $\phi(\vec{s})$ if we know that $(L, v) \models \phi(\vec{s})$ since otherwise our family has to contain a winning strategy on the empty positive game $0$, of which there are none.

For $P_{\forall}$, to give a family of strategies on $\forall x, N, \Gamma$ we must give a strategy on $N, \Gamma$ for each choice of $x$ — that is, a family of strategies on the set of $\Theta$-satisfying $\mathcal{L}$-models over $X \cup \{x\}$.

For $P_{\exists}$, to give a family of strategies on $\exists x P, \Gamma$ we must choose a value $s$ for $x$ and give a family of strategies on $P[s/x], \Gamma$.

As well as the core introduction rules, there is a small set of core elimination rules, found in Figure 3. These permit decomposition of the second and third formula in a sequent, if the first formula is $\bot$ or $\top$. They correspond to isomorphisms between the
Figure 3: Core Elimination Rules for WS1

\[
\begin{align*}
\text{P} & : \Gamma, \Phi \vdash \bot \\
\text{P} & : \Gamma, \Phi \vdash \bot, N, \Gamma \\
\text{P} & : \Gamma, \Phi \vdash \bot, P, N, \Gamma \\
\text{P} & : \Gamma, \Phi \vdash \bot, P, \Gamma \\
\text{P} & : \Gamma, \Phi \vdash \bot, P, Q, \Gamma \\
\text{P} & : \Gamma, \Phi \vdash \bot, P, O, \Gamma \\
\text{P} & : \Gamma, \Phi \vdash \bot, P, N, \Gamma \\
\end{align*}
\]

Figure 4: Core Equality Rules for WS1

\[
\begin{align*}
\text{P} & : x \neq y, \Theta \vdash \Gamma \\
\text{P} & : x \neq y, \Theta \vdash \Gamma \\
\text{P} & : x \neq y, \Theta \vdash \Gamma \\
\text{P} & : x \neq y, \Theta \vdash \Gamma \\
\end{align*}
\]

premise and conclusion in the semantics, which induces a bijection between the winning strategies on each. For example, \( \text{P} \vdash \bot \) uses the isomorphism \( \bot \sim \bot \), and \( \text{P} \vdash \bot \) the isomorphism \( \bot \sim \bot \). Finally, there are core equality rules which deal with equality, given in Figure 4. We can interpret the core equality rules at a model \((L,v)\) as follows:

- To interpret \( \text{P} \vdash x \neq y \) (reflexivity of identity), we take the empty family of strategies, since there are no \( \Theta \)-satisfying \( L \)-models if \( \Theta \) contains \( x \neq x \).

- To interpret the matching rule \( \text{P} \vdash x \neq y \), we note that the collection of \( \Theta \)-satisfying \( L \)-models can be decomposed into those where \( x \) and \( y \) are identified (the left-hand premise) and those where they are distinct (the right-hand premise).

Once a discipline regarding where the matching rule is applied has been introduced, proof search in this core subsystem is particularly simple, as the form of the sequent to be proved determines the choice of final rule. We will later show that the core rules are sufficient to denote any finitary family of uniform winning strategies.

We make a brief note on polarities and reversibility, and a comparison with focused proof systems. In such systems, polarisation is used to differentiate between connectives whose corresponding rules are reversible or irreversible [6]. Irreversible rules act on positive formulas. An irreversible rule is one where (reading upwards) in applying the rule one must make some definite choice, a choice which could determine whether the proof search succeeds or not. Thus, additive disjunction introduction is always an irreversible rule, and in linear logic so is the tensor introduction rule, since a choice must be made regarding how the context is split.

In WS1, the core introduction rule for tensor (as for all such rules) is additive, not multiplicative. Thus, this rule is reversible, and \( \otimes \) is resultantly a negative connective. In contrast, \( \otimes \) is a positive connective as there are two different core introduction rules, which are not reversible. Thus, as well as the semantic motivation, we can view our distinction between positive and negative formulas in the same light as the polarities of focused systems.
However, there is an important distinction. In focused systems, the proof search alternates between negative phases, in which reversible rules are applied, and positive phases, in which irreversible rules are applied. Analytic proof search in WS follows a different two-phase discipline, in which we first decompose the first formula of a sequent into a unit using the core introduction rules, and then collate the tail formulas together using the core elimination rules. We will give an embedding of LLP inside WS in Section 3.3.

### 3.2.2. Other Rules

![Diagram of rules](image)

The non-core rules of WS1 are given in Figure 5 with $\Delta^+$ ranging over lists of positive formulas, $\Gamma^+$ over non-empty lists of formulas. These rules reflect some of the categorical structure enjoyed by our games model, and allow straightforward interpretation of other logics and programming languages inside WS1. They include a cut rule, a multiplicative $\otimes$ rule, a restricted form of the exchange rule, weakening, and so on. We will later see that these rules are admissible with respect to the rules in Figures 2, 3, and 4, when formulas, $\Gamma^+$.

- In the cases of $P^+_\otimes$, $P^+_\oplus$, $P_{\otimes}^+$, $P_{\oplus}^+$, and $P_{\otimes}^+$, the premise and conclusion are the same game, up to retagging, and can be interpreted using game isomorphisms.

- In the cases of $P^+_\bot$, $P_{\otimes}^*_\bot$, $P_{\otimes}^*$, a strategy on the conclusion can be obtained by using only part of the strategy on the premise. For example, for $P_{\otimes}^*$, we remove all moves in $M$.

13
In the cases of $P_{\oplus 1}^T$, $P_{\oplus 2}^T$, $P_{\oplus}^{\top}$, $P_{\bot}^{\top}$, a strategy on the conclusion can be obtained by using the strategy on the premise and ignoring the extra moves available to Player.

The $P_{id}$ rule requires a strategy on $N \rightarrow N$: we can use a \textit{copycat} strategy in which Player always switches component, playing the move that Opponent previously played. The $P_{id}$ rule can be interpreted by playing copycat in the $M$ component.

The $P_{cut}$ and $P_{cut}^0$ rules can be interpreted by playing the two strategies given by the premises against each other in the $N$ component: “parallel composition plus hiding”.

The $P_{mul}$ rule can be interpreted by combining the strategies given by the premises in a multiplicative manner: Opponent’s moves in $M, \Gamma$ are responded to in accordance with the first premise, and moves in $N$ in accordance with the second. The $P_{\rightarrow}$ rule can be interpreted similarly.

To interpret $P_{\text{con}}$, we can construct a strategy on the conclusion by identifying the two copies of $?P$ in the premise. To interpret $P_{\text{con}}^!$, we can construct a strategy on the conclusion by identifying the two copies of $!M$ in the conclusion.

We can interpret $P_{\text{ana}}$ using the following construction: given a map $N \rightarrow M \otimes N$, we may “unwrap” it an infinite number of times to yield a strategy on $N \rightarrow !M$.

The $N$ component represents a parameter that can be used to pass information between the separate threads, to admit history-sensitive behaviour.

### 3.2.3. Embedding of Intuitionistic Linear Logic

For any negative formulas $M, N$, define $M \rightarrow N$ to be $N \otimes M^\bot$. Thus any formula of first-order Intuitionistic Linear Logic is a negative formula of WS1. We sketch an embedding into WS1 of proofs of ILL (over the connectives $\otimes, \rightarrow, \forall, \&$, $\bot$, $!$ and (negative) atoms, formulated with left- and right- introduction rules as in [37]).

**Proposition 3.1.** For any proof $p$ of $M_1, \ldots, M_n \vdash N$ in ILL with free variables in $X$, there is a proof $\kappa(p)$ in WS1 of $X; \emptyset \vdash N, M_1^\bot, \ldots, M_n^\bot$.

**Proof** We show that for each rule of ILL there is a derivation in WS1 of the conclusion from the premises.

The left $\otimes$ rule corresponds to $P_{\otimes}^T$. For the right $\otimes$ rule, with $\Gamma = G_1, \ldots, G_n$ and $\Delta = D_1, \ldots, D_m$, we duplicate the proof and use $P_{mul}$ as follows:

\[
\begin{align*}
\text{P}_{\otimes} & \vdash M, G_1, \ldots, G_n \quad \text{P}_{\otimes} & \vdash N, D_1, \ldots, D_m \\
\text{P}_{\otimes} & \vdash M, N, G_1, \ldots, G_n, D_1, \ldots, D_m \\
\text{P}_{\otimes} & \vdash M \otimes N, G_1, \ldots, G_n, D_1, \ldots, D_m \\
\text{P}_{\otimes} & \vdash M, G_1, \ldots, G_n, D_1, \ldots, D_m \\
\text{P}_{\otimes} & \vdash M, N, D_1, \ldots, D_m \\
\text{P}_{\otimes} & \vdash M, G_1, \ldots, G_n, D_1, \ldots, D_m \\
\text{P}_{\otimes} & \vdash M, N, D_1, \ldots, D_m \\
\text{P}_{\otimes} & \vdash M, G_1, \ldots, G_n, D_1, \ldots, D_m \\
\end{align*}
\]

The left $1$ rule corresponds to $P_{\otimes}^0$. The right $1$ rule corresponds to $P_1$. The left $\rightarrow$ rule can be derived as follows:

\[
\begin{align*}
\text{P}_{\rightarrow} & \vdash N, D_1, \ldots, D_m \\
\text{P}_{\rightarrow} & \vdash M, G_1, \ldots, G_n \\
\text{P}_{\rightarrow} & \vdash M, N, G_1, \ldots, G_n, D_1, \ldots, D_m \\
\text{P}_{\rightarrow} & \vdash M \otimes N, G_1, \ldots, G_n, D_1, \ldots, D_m \\
\end{align*}
\]
We will later refer to this derived rule as \( P_{\text{der}} \). The left \& rules correspond to the \( P_{\text{sym}}^+ \) rules. The right \& rule corresponds to \( P_{\text{con}} \). The right-\forall rule corresponds to \( P_{\forall} \) and the left-\forall rule corresponds to \( P_{\forall}^+ \).

The dereliction, contraction and weakening rules for the exponential correspond to \( P_{\text{der}}, P_{\text{con}} \) and \( P_{\text{wk}}^+ \), respectively. We next give the translation of the right ! rule (promotion). We first assume \( \Gamma \) consists of a single formula \( L \).

We will later refer to this derived rule as \( P_{\text{prom}} \). If \( \Gamma \) contains more than one formula, we use the equivalence of \( !M \& N \) and \( !(M \& N) \) in \( \text{WS1} \).

The first direction \( p_1 \vdash !M \& N \rightarrow !!(M \& N) \) is defined as follows:

The second direction \( p_2 \vdash !(M \& N) \rightarrow !M \& N \) is given as follows:

We can then generalise \( P_{\text{prom}} \) to
and interpret the right ! rule of ILL.

A detailed proof-theoretic analysis of the properties of this translation is beyond the scope of this paper. However, we note that the translation is semantically natural, in the following sense. We shall see in Section 5 that the categorical models of WS1 have (among other properties) the structure of a standard categorical model of ILL: they are Lafont categories [32]. The semantics of the quantifier-free fragment of ILL induced by translation into WS1 followed by interpretation in a categorical model coincides with the expected semantics of ILL in a Lafont category.

3.2.4. New Theorems

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.

The formulas

\[
(A \otimes B \to \bot) \otimes (C \otimes D \to \bot) \to \bot
\]

\[
((A \to \bot) \otimes (C \to \bot) \to \bot) \otimes ((B \to \bot) \otimes (D \to \bot) \to \bot)
\]

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 \([(A \otimes B) \otimes (C \otimes D)] \to [(A \otimes C) \otimes (B \otimes D)]\), using an interpretation of depolarised formulas in a polarised setting following [34].

As observed by Blass [7], however, there are (uniform) history-sensitive winning strategies for medial. Informally, suppose:

- Opponent first chooses the left hand component in the output (choice 1)
- Opponent then chooses the right hand component in the input (choice 2)

Player can then play copycat in \(C\). If Opponent then switches to the second output component \((B \to \bot) \otimes (D \to \bot) \to \bot\), Player must enter copycat in \(D\). But this decision relies on knowledge of Opponent’s choice 2, which is not possible in an innocent setting and requires history-sensitive knowledge.

An outline WS1 proof of this formula is given in Figure 6. The use of the \(P\otimes\) demonstrates where the proof branches; there are four branches corresponding to the two uses
of $P \otimes$. In each of these four branches different $P_{\otimes i}$ proof rules are chosen at the points labelled $P_{\otimes 1}$ here.

Similarly, the following theorems of $WS1$ are not provable in $ILL$ but are provable in $WS1$:

- $[A \otimes (C \& D)] \& [B \otimes (C \& D)] \& [(A \& B) \otimes C] \& [(A \& B) \otimes D] \rightarrow (A \& B) : (C \& D)$, also discussed in [7]
- $\phi_{ex} \rightarrow \phi_{ex} \otimes \phi_{ex}$ where $\phi_{ex} = (\phi \& (\phi \rightarrow \bot)) \rightarrow \bot$.

### 3.3. Embedding Polarized Linear Logic in $WS1$

Polarized Linear Logic (LLP) [27] is a proof system for a polarisation of linear logic into negative and positive formulas. As we have noted, this is entirely different from the polarisation of $WS1$ formulas employed here: each makes sense within the proof system within which it is defined. Here, we show how proofs of LLP may be represented inside $WS1$ by translation, with two objectives:

- To clarify the relationship between the two logical systems, and their notions of polarisation.
- To capture both call-by-name and call-by-value $\lambda$-calculi via known (and elegant) translations into LLP, which may be composed with our embedding of LLP into $WS1$. In the call-by-name case, this corresponds with interpretation via intuitionistic linear logic, whereas for call-by-value it is new.

The formulas of LLP (over the units) are as follows:

$$
P ::= 1 | 0 | P \otimes Q | P \oplus Q | \downarrow N | \uparrow ! N
\$$

$$
N ::= \bot | \top | M \& N | M \& N | \uparrow P | \uparrow ? P
\$$

There is an operation $(-)^\uparrow$ exchanging polarity, swapping $1$ for $\bot$, $0$ for $\top$, $\otimes$ for $\&$, and so on. The presentation of LLP gives in [27] omits the linear lifts $\uparrow$ and $\downarrow$ of $MALLP$. We will include them in our presentation of LLP and its embedding.

A sequent of LLP is a list of LLP formulas. The proof rules for Polarized Linear Logic are given in Figure [8]. $\Gamma^-$ ranges over lists of negative formulas, and $\Gamma'$ over lists where at most one formula is positive. We say a negative LLP formula $N$ is reusable (and write $reuse(N)$) if every occurrence of $\uparrow$ occurs under a $?$. If we exclude the linear lifts $\uparrow$ and $\downarrow$, all negative formulas are reusable. $reuse(\Gamma^-)$ holds if all formulas in $\Gamma^-$ are reusable.

Each provable sequent has at most one positive formula, so we can restrict our attention to sequents of this form. It is possible to give semantics to LLP proofs as innocent strategies [27], which do not have access to the entire history of play.

We next describe an embedding of LLP inside $WS1$. Apart from some renaming of units, connectives in LLP will be interpreted by the same connective in $WS1$. Broadly speaking, positive formulas of LLP will be mapped to negative formulas of $WS1$, and negative formulas of LLP to positive formulas of $WS1$. However, under this scheme there is a mismatch for the additives: we will therefore need to map formulas of LLP to families of $WS1$ formulas. The formulas that have a lift as their outermost connective will be mapped to singleton families.

Let $WS1^-$ denote the set of negative $WS1$ formulas, and $WS1^+$ the set of positive $WS1$ formulas.
Figure 6: Outline Proof of Medial
Definition A finite family of negative (resp. positive) WS1 formulas is a pair \((I, f)\) where \(I\) is a finite set and \(f : I \to \text{WS1}^-\) (resp. \(I \to \text{WS1}^+\)).

For brevity, given such a family \(F = (I, f)\) we will write \(|F|\) for \(I\) and \(F_x\) for \(f(x)\). We will interpret a negative formula of LLP as a finite family of positive WS1 formulas, and a positive formula of LLP as a finite family of negative WS1 formulas. We describe this mapping in Figure 8. Like [34], we decompose the polarity-reversing exponentials of LLP into polarity-preserving exponentials of and polarity-switching linear lifts.

Note that \(|i(A^+)\)| = \(|i(A)|\) and \(i(A^+)y = i(A)^+_y\). We translate proofs of LLP to families of proofs of WS1 in the following manner:

- Given an LLP proof \(p\) of \(\vdash N_1, \ldots, N_n\) and \(x_i \in |i(N_i)|\) for each \(i\), we construct a proof \(i(p, \vec{x})\) of \(\vdash \perp, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}\).
- Given an LLP proof \(p\) of \(\vdash N_1, \ldots, N_i, Q, N_{i+1}, \ldots, N_n\) and \(x_i \in |i(N_i)|\) for each \(i\), we construct a pair \((y, q)\) where \(y \in |i(Q)|\) and \(q\) is a proof of \(\vdash i(Q)_y, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}\).

Proposition 3.2. Suppose \(N\) is reusable. Then for any \(x\) in \(|i(N)|\), there is a formula \(Q\) and proofs \(p \vdash \! Q\) and \(p' \vdash i(N)^+_x\) such that \(|p|\) and \(|p'|\) are inverses.

Proof Simple induction, making use of isomorphisms \(!(M \& N) \cong!M \& !N\).

Proposition 3.3. For each LLP formula \(P\), \(y \in |i(P)|\) and sequence of negative WS1 formulas \(\Delta^-\) there is a WS1 proof \(P_T \vdash i(P)_y, \Delta^-, \top\).
Simple induction on $P$.

We next show how each of the LLP proof rules is translated. The translation is simple; we demonstrate some representative cases.

- The cut rule, with $p = \text{cut}(q,r)$: Suppose $\Gamma = N_1, \ldots, N_i, P, N_{i+1}, \ldots, N_n$ and $\Delta = M_1, \ldots, M_m$. Let $x_i \in |i(N_i)|$ and $y_i \in |i(M_i)|$. Then $i(r, y_i) = (y, t)$ with $y \in |i(N_i)|$ and $t \vdash i(N_{i+1})$, $i(M_1), y_i, \ldots, i(M_m), y_m$. Then $i(q, \overline{x_i}, y) = (q', y')$ where $y' \in |i(P)|$ and

$$q' \vdash i(P)_{y'}, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}, i(N_{i+1})_{y}, \ldots, i(N_m)_{y_m}$$

Applying $P_{\text{cut}}$ to this proof and $t$ results in a proof $g$ of

$$\vdash i(P)_{y'}, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}, i(M_1)_{y_1}, \ldots, i(M_m)_{y_m}$$

and we set $i(p, \overline{x_i}, y_i) = (y', g)$.

The case where $\Gamma = N_1, \ldots, N_n$ and $\Delta = M_1, \ldots, M_m$ is similar.

- The $\uparrow$ rule, with $p = \uparrow(q)$: Let $\Gamma = N_1, \ldots, N_n$ and $x_i \in |i(N_i)|$. Then $i(q, \overline{x_i}) = (y, q)$ where $q \vdash i(P)_{y}, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}$. We set $i(p, \overline{x_i})$ to be the following proof:

$$q \vdash i(P)_{y}, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}$$

$$\vdash \top \circ i(P)_{y}, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}$$

$$\vdash \top \circ \bigoplus_{j \in |i(P)|} \top \circ i(P)_j, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}$$

$$\vdash \top, i(N_1)_{x_1} \otimes \ldots \otimes i(N_n)_{x_n} \otimes \bigoplus_{j \in |i(P)|} \top \circ i(P)_j$$

$$\vdash \top, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}, \bigoplus_{j \in |i(P)|} \top \circ i(P)_j$$

### Table 8: LLP formulas as families of WS1 formulas

<table>
<thead>
<tr>
<th>$A \in LLP$</th>
<th>$i(A) \in \text{Fam WS1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>${\ast}, \ast \mapsto \top$</td>
</tr>
<tr>
<td>0</td>
<td>${\ast}, \ast \mapsto \bot$</td>
</tr>
<tr>
<td>$P \otimes Q$</td>
<td>${i(P) \times</td>
</tr>
<tr>
<td>$P \oplus Q$</td>
<td>${i(P) \uplus</td>
</tr>
<tr>
<td>$\downarrow N$</td>
<td>${\ast}, \ast \mapsto \uplus_{i \in</td>
</tr>
<tr>
<td>$\downarrow \bot$</td>
<td>${\ast}, \ast \mapsto \top$</td>
</tr>
<tr>
<td>$M \otimes N$</td>
<td>${i(M) \times</td>
</tr>
<tr>
<td>$M \ominus N$</td>
<td>${i(M) \uplus</td>
</tr>
<tr>
<td>$\uparrow P$</td>
<td>${\ast}, \ast \mapsto \bigoplus_{j \in</td>
</tr>
<tr>
<td>$?P$</td>
<td>${\ast}, \ast \mapsto \bot \oplus \bigoplus_{j \in</td>
</tr>
</tbody>
</table>
Note that in the semantics of this rule two moves are played: the opening lift overall (O-move) and the opening lift in the derelicted component (P-move), which corresponds to “focusing” on that component.

• The ?c rule, with \( p = ?c(q) \):

If \( \Gamma = N_1, \ldots, N_n \) and \( x_i \in |i(N_i)| \) and \( x \in |i(N)| \) then \( i(q, \overline{x_i}, x, x) \) is a proof of \( \vdash \bot, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}, i(N)_x \). We can apply Proposition 3.2 and use \( ? \)-contraction in WS1 to yield a proof \( q' \) of

\[
\vdash \bot, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}, i(N)_x
\]

and we set \( i(p, \overline{x_i}, x) = q' \).

If \( \Gamma = N_1, \ldots, N_i, P, N_{i+1}, \ldots, N_n \) and \( x_i \in |i(N_i)| \) and \( x \in |i(N)| \) then \( i(q, \overline{x_i}, x, x) = (y, q') \) where \( q' \vdash i(P)_y, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}, i(N)_x \). We can apply Proposition 3.2 and use \( ? \)-contraction in WS1 to yield a proof \( q'' \) of

\[
\vdash i(P)_y, i(N_1)_{x_1}, \ldots, i(N_n)_{x_n}, i(N)_x
\]

and we set \( i(p, \overline{x_i}, x) = (y, q'') \).

We can hence interpret proofs in LLP as (families of) proofs in WS1.

4. Representing Imperative Programs and their Properties

4.1. Imperative Cell

As an example of a proof of WS1 capturing imperative behaviour (and which does not correspond to a proof of intuitionistic or polarized linear logic), we give a proof which denotes the Boolean reference cell strategy described in Section 2.2, the cell strategy of [5].

Recall that this is a strategy for the game \( !(<B & B_i>) \), where \( B = \bot \triangleleft \top \oplus \top \) and \( B_i = (\bot \triangleleft \bot) \triangleleft \top \). We can parametrise the cell by a starting value, yielding a strategy on \( B \twoheadrightarrow !(<B & B_i>) \). We may obtain this strategy using a finite strategy \( p : B \twoheadrightarrow (B & B_i) \triangleleft B \).

The strategy \( p \) is defined as follows, using the naming conventions from Section 2.2:

\[
\begin{array}{c}
\text{B} \twoheadrightarrow (\text{B} \ & \text{Bi}) \ \triangleleft \text{B} \\
q & b \\
\end{array}
\]

\[
\begin{array}{c}
q \\
b \\
\end{array}
\]

\[
\begin{array}{c}
in(b) \\
\text{ok} \\
q \\
b \\
\end{array}
\]

To obtain the cell strategy, we consider an infinite unwrapping \( \langle p \rangle : B \twoheadrightarrow !(<B & B_i>) \), as performed by the semantics of the \( P_{ana} \) rule.
We can represent this strategy in our system using the anamorphism rule $P_{\text{ana}}$: we may prove $!(B \& B_i)$, $B^\perp$ by applying this rule to a proof of $(B \& B_i)$, $B, B^\perp$. To obtain this, we apply the product rule to a pair of proofs:

- $P_{\text{read}}$, of $B, B, B^\perp$, corresponding to a function which reads its argument, returns it and propagates it to the next call, and
- $P_{\text{write}}$, of $B_i, B, B^\perp$, corresponding to a function which ignores its argument, accepts a Boolean input value and propagates it to the next call.

In this proof, if a rule is not labelled it is the unique applicable core rule, and some steps are omitted for brevity.

\[
\begin{align*}
P_{\text{write}} & \vdash (\bot \& \bot) \triangleleft T, \bot \triangleleft (T \oplus T), T \otimes (\bot \& \bot) & P_{\text{read}} & \vdash \bot \triangleleft (T \oplus T), \bot \triangleleft (T \oplus T), T \otimes (\bot \& \bot) \\
& \vdash (\bot \& \bot) \triangleleft T & & \vdash ((\bot \& \bot) \triangleleft T) \& ((\bot \triangleleft (T \oplus T)), \bot \triangleleft (T \oplus T), T \otimes (\bot \& \bot)) \\
& \vdash T, ((\bot \& \bot) \triangleleft T) & & \vdash T, ((\bot \& \bot) \triangleleft T) \\
& \vdash \bot, (T \triangleleft (T \oplus T)) \& (T \otimes (\bot \& \bot)) & & \vdash \bot, (T \triangleleft (T \oplus T)) \\
& \vdash (T \triangleleft (T \oplus T)) \& (T \otimes (\bot \& \bot)) & & \vdash (\bot \triangleleft (T \oplus T)) \& (T \otimes (\bot \& \bot)) \\
& \vdash \bot, (T \triangleleft (T \oplus T)) \& (T \otimes (\bot \& \bot)) & & \vdash \bot, (T \triangleleft (T \oplus T)) \& (T \otimes (\bot \& \bot)) \\
& \vdash \bot, (T \otimes (\bot \& \bot)) & & \vdash \bot, (T \otimes (\bot \& \bot)) \\
& \vdash \bot \& \bot, T \triangleleft (T \oplus T), T \otimes (\bot \& \bot) & & \vdash \bot \& \bot, T \triangleleft (T \oplus T), T \otimes (\bot \& \bot) \\
& \vdash (\bot \& \bot) \triangleleft T, \bot \triangleleft (T \oplus T), T \otimes (\bot \& \bot) & & \vdash (\bot \& \bot) \triangleleft T, \bot \triangleleft (T \oplus T), T \otimes (\bot \& \bot)
\end{align*}
\]

and $P_{\text{read}}$ is

\[
\begin{align*}
P_{\text{read}} & \vdash \bot, (T \oplus T), T \otimes (\bot \& \bot) \\
& \vdash (\bot \& \bot) \triangleleft T, \bot \triangleleft (T \oplus T), T \otimes (\bot \& \bot) \\
& \vdash (\bot \& \bot) \triangleleft T, \bot \triangleleft (T \oplus T), T \otimes (\bot \& \bot)
\end{align*}
\]
We will later give categorical semantics to WS₁, and so the above proof provides a
categorical account of this Boolean reference cell, using a final coalgebraic property of
the exponential.

We may use this proof to interpret declaration of a Boolean reference in either call-
by-name or call-by-value settings, by composition (cut) with (the translation of) a term-
in-context of the form \( \Gamma, x : \text{var} \vdash M : T \). Thus we may translate the recursion-free
fragments of Idealized Algol [36] and Reduced ML over finite datatypes into WS₁, for
example.

4.2. State Encapsulation

WS₁ is more expressive than total, finitary Idealized Algol: for instance, we may use
the anamorphism rule to capture structures such as stacks, capable of storing an arbi-
trarily large amount of data. A generalised programming construct which corresponds
to this capability is the encapsulation operation which appears as the thread operator
in [38], and as the encaps strategy in [30] where it is used for constructing imperative
objects in a model based on the same underlying notion of game as used here. The
operator has type

\[ (s \to (o \times s)) \to s \to (1 \to o). \]

Here \( s \) is the type of the object’s internal state. The first argument represents an object
which takes an explicit state of type \( s \), and returns a value of type \( o \), together with an
updated state. The second argument represents an initial state. Encapsulation returns
an object of type \( 1 \to o \) (a “thunk” of type \( o \)) in which the state \( s \) is encapsulated — i.e.
hidden from the environment, but shared between separate invocations of the object. On
first invocation (unthunking) the initial state is used as the input state, and thereafter,
each fresh call receives the output state from the previous invocation as its input.

We can represent this operation in WS₁ using the \( P_{\text{ana}} \) rule. To do this, we con-
sider a call-by-value interpretation of types. We may translate call-by-value types as
positive formulas of LLP: \( \phi^+(1) = 1 \), \( \phi^+(A \times B) = \phi^+(A) \otimes \phi^+(B) \) and \( \phi^+(A \to B) = !\left(\phi^+(A) \to o \uparrow \phi^+(B)\right) \). Thus by composition with the embedding of LLP in

\footnote{This is slightly different to the original embedding presented in [27], which uses ? rather than ↑ in the translation of \( \to \), allowing first-class continuations to be interpreted (the \( \lambda \mu \)-calculus). The translation
adopted here is a form of linear CPS interpretation.}
WS1, we may translate the types $o$ and $s$ as the families of WS-formulas $i \circ \phi^+(s)$ and $i \circ \phi^+(o)$. Let us assume for simplicity, that these are singleton families $\{S\}$ and $\{O\}$ respectively (i.e. $s$ represent products of function types). Then $\text{encaps}$ may be translated as a proof of $\vdash \bot$, $T \otimes i \circ \phi^+(1 \rightarrow o), S^\bot, i \circ \phi^+(s \rightarrow (o \times s))^\bot$ — i.e. $\vdash \bot, T \otimes !\uparrow (\mathbf{0} \otimes 1, S^\bot, ? \downarrow (S \otimes (O \times S^\bot)))$ — as follows:

- $P_{\text{id}} \vdash O \rightarrow O^\bot, O, O^\bot, S^\bot$
- $P_{\text{mul}} \vdash O, S, O^\bot, S^\bot$
- $P_{\text{mul \otimes}} \vdash S, S^\bot$
- $P_{\text{sym}} \vdash O \otimes O, S^\bot, O^\bot, S^\bot$
- $P_{\text{sym}}^+ \vdash \bot \rightarrow O^\bot \otimes S, S^\bot$
- $P_{\text{def}} \vdash \bot \rightarrow \bot, S, \otimes (S \otimes (O \times S^\bot)), S^\bot$

### 4.3. Coroutines

We may also give a proof denoting a coroutining operation, permitting a form of deterministic multithreading, defined as a strategy in [23][24]. In a call-by-name setting, this corresponds to an operation taking two terms $s, t$ of type $\text{com} \rightarrow \text{com}$, and returning a command which runs $s$: when (and if) $s$ calls its argument, control passes to $t$. When $t$ calls its argument, control is passed back to $s$, and so on, until either $s$ or $t$ terminates.

We can define a coroutining operator $\text{cocomp} \vdash \Sigma, ?(\Sigma \otimes \Sigma), ?(\Sigma \otimes \Sigma)$, where $\Sigma = \top \otimes \bot$. We first give a proof $o$ of $(\Sigma \otimes \bot) \rightarrow o \Sigma$. 

<table>
<thead>
<tr>
<th>Type</th>
<th>Rule</th>
<th>Proof</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{\text{id}}$</td>
<td>$\vdash O, O^\bot$</td>
<td>$\vdash S, S^\bot$</td>
</tr>
<tr>
<td>$P_{\text{mul}}$</td>
<td>$\vdash O, S, O^\bot, S^\bot$</td>
<td>$\vdash O, S, O^\bot, S^\bot$</td>
</tr>
<tr>
<td>$P_{\text{sym}}$</td>
<td>$\vdash O \otimes O, S^\bot, O^\bot, S^\bot$</td>
<td>$\vdash O^\bot \otimes S, S^\bot$</td>
</tr>
<tr>
<td>$P_{\text{mul \otimes}}$</td>
<td>$\vdash O \otimes O, S^\bot, O^\bot, S^\bot$</td>
<td>$\vdash S, S^\bot$</td>
</tr>
<tr>
<td>$P_{\text{def}}$</td>
<td>$\vdash \bot \rightarrow \bot, S, \otimes (S \otimes (O \times S^\bot)), S^\bot$</td>
<td>$\vdash \bot \rightarrow \bot, S, \otimes (S \otimes (O \times S^\bot)), S^\bot$</td>
</tr>
</tbody>
</table>
We next define a proof $o \vdash (!\Sigma \Rightarrow \bot) \Rightarrow \bot \Rightarrow !\Sigma$, which connects the output move of the first argument to the Player-move in the second argument.

We can then define cocomp.

4.4. Specifying Properties of Programs

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 both more abstractly and 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).
4.4.1. Data-Independent Programming

We can use quantifiers to represent data-independent structures such as cells and stacks, where the underlying ground type at a given \( L \)-structure \( L \) is \( |L| \). As a formula/game, this ground type is represented by \( V = \bot \otimes \exists x. \top \) — a dialogue in this game consists of Opponent playing a question move \( q \) and Player responding with an element of \( |L| \). We can represent a stream of such values using the formula \( \neg V \).

Let \( V_i = \forall x. \bot \otimes \top \) represent an ‘input version’ of \( V \), where Opponent plays an \( |L| \) value and Player then accepts it, analogous to \( \text{Bi} \) above. The type of a stack object can then be given by the formula \( \neg (V \& V_i) \), with a “pop” and a “push” method. We give a proof denoting the behaviour of such a stack, parametrised by a starting stack, of type \( V \rightarrow \neg (V \& V_i) \).

\[
\begin{align*}
\text{P}_{\text{id}} & \quad \vdash (\bot \otimes \exists x. \top), (? \otimes \forall x. \bot) \\
\text{P}_{\text{in}} & \quad \vdash (\bot \otimes \exists x. \top), (? \otimes \forall x. \bot) \\
\text{P}_{\text{der}} & \quad \vdash \bot \otimes \exists x. \top, (? \otimes \forall x. \bot) \\
\text{P}_{\text{ana}} & \quad \vdash \bot \otimes \exists x. \top, (? \otimes \forall x. \bot)
\end{align*}
\]

Once again, we use \( \text{P}_{\text{ana}} \) to obtain the infinite behaviour, applied to a proof \( q \) of \( V \rightarrow (V \& V_i) \). The strategy denoted by \( q \) performs as ‘copycat’ in the \( V \rightarrow V \) component, and in the \( V \rightarrow V_i \) component behaves as follows:

\[
V \rightarrow \neg V_i \quad \text{Vi} \quad \vdash \neg V \\
\text{in}(v) \quad \text{ok} \quad q \\
v
\]

and then enters copycat.

4.4.2. 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 \([5]\). In \( WS1 \) we may define formulas for which the only proof denotes a good variable.

The formula \( \text{worm} = \text{Bi} \otimes \neg B \) 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 \( \text{input} x \) then Player responds with \( \text{ok} \), 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 \( \text{True} \) regardless of what was written.
To exclude such behaviour, we can replace the input/output moves with atoms. Define \( B^{\phi,\psi} = \bot \triangleleft (\overline{\phi} \oplus \overline{\psi}) \) and \( \text{Bi}^{\phi,\psi} = (\phi \& \psi) \triangleleft \top \), with \( \text{worm}^{\phi,\psi} = B^{\phi,\psi} \circ \text{Bi}^{\phi,\psi} \). If \( \phi \) and \( \psi \) are assigned \( \text{tt} \), then this denotes the same dialogue as \( \text{worm}^{\phi,\psi} \). However, the denotation of any proof of \( \text{worm}^{\phi,\psi} \) at such a model must be the good variable strategy. The rule for \( \phi \) (and semantically, uniformity of strategies) ensures that \( \phi \) must be played before \( \overline{\phi} \), and \( \psi \) before \( \overline{\psi} \). Consequently, Player can only respond with a particular Boolean value in the \textit{read} component if that same value has previously been given as an input in the \textit{write} component, so good-variable behaviour is assured. The following proof of this formula uses only the core rules and the promotion rule.

\[
\begin{array}{c}
\phi \vdash \top \\
\phi \vdash \phi \\
\phi \vdash \bot \triangleleft \phi \oplus \psi \\
\phi \vdash !((\bot \triangleleft \phi \oplus \psi) \triangleleft \top) \\
\phi \vdash \text{Prom}((\bot \triangleleft \phi \oplus \psi) \triangleleft \top) \\
\phi \vdash !, T,(\bot \triangleleft \phi \oplus \psi) \\
\phi \vdash T,!(\bot \triangleleft \phi \oplus \psi) \\
\phi \vdash T \bullet T!(\bot \triangleleft \phi \oplus \psi) \\
\phi \vdash \phi, T,(\bot \triangleleft \phi \oplus \psi) \\
\phi \vdash (\phi \& \psi), T,(\bot \triangleleft \phi \oplus \psi) \\
\phi \vdash (\phi \& \psi) \triangleleft T,(\bot \triangleleft \phi \oplus \psi) \\
\phi \vdash ((\phi \& \psi) \triangleleft T) \circ T!(\bot \triangleleft \phi \oplus \psi)
\end{array}
\]

We cannot use \! to obtain a formula which admits only an arbitrarily reusable ‘good variable’, but we can obtain finite approximations. For example, the formula

\[
\text{Bi}^{\alpha,\beta} \circ (\text{worm}^{\phi,\psi} \& (\text{Bi}^{\alpha,\beta} \circ \text{worm}^{\phi,\psi}) \& (\text{Bi}^{\alpha,\beta} \circ (\text{Bi}^{\alpha,\beta} \circ \text{worm}^{\phi,\psi})))
\]

models a good variable that can be written to twice, and can be read at most twice before the second write. Strategies on such formulas then approximate our reusable cell strategy above on \!((\text{Bi} \& \text{Bi})).

5. Categorical Semantics for WS1

To give a formal semantics for our logic, we first introduce a notion of categorical model which captures everything except the first-order structure (quantifiers and atoms). We shall use notation \( \eta : F \Rightarrow G : C \Rightarrow D \) to mean \( \eta \) is a natural transformation from \( F \) to \( G \) with \( F,G : C \Rightarrow D \).

First, we define some categories of games that will form the intended instance of our categorical model. Objects in these categories will be negative games, and an arrow \( A \to B \) will be a strategy on \( A \to B \). We can compose strategies using “parallel composition plus hiding”. Suppose \( \sigma : A \to B \) and \( \tau : B \to C \), define

\[
\sigma \| \tau = \{ s \in (M_A + M_B + M_C)^*: s|_1 \in P_A \land s|_2 \in P_B \land s|_3 \in P_C \}
\]
and set

\[ \tau \circ \sigma = \{ s|_{1,3} : s \in \sigma \| \tau \}. \]

It is well-known that \( \tau \circ \sigma \) is a well-formed strategy on \( A \to C \) (see e.g. [3]).

**Proposition 5.1.** Composition is associative, and there is an identity \( A \to A \) given by the copycat strategy: \( \{ s \in P_{A \to A} : \gamma(s) \} \) where \( \gamma(s) \) holds if and only if \( t|_1 = t|_2 \) for all even-length prefixes \( t \) of \( s \).

**Definition** The category \( \mathcal{G} \) has negative games as objects, and a map \( \sigma : A \to B \) is a strategy on \( A \to B \) with composition and identity as above.

This category has been studied extensively in e.g. [25, 11, 30], and has equivalent presentations using graph games [16] and locally Boolean domains [22]. If \( A, B \) and \( C \) are bounded, \( \sigma : A \to B \) and \( \tau : B \to C \) are total then \( \tau \circ \sigma \) is also total. Total strategies do not compose for unbounded games, however. Winning strategies on unbounded games do compose [15], and the identity strategy is winning.

**Definition** The category \( \mathcal{W} \) has negative games as objects and winning strategies as maps.

A map \( \sigma : A \to B \) is *strict* if it responds to Opponent’s first move with a move in \( A \), if it responds at all. Strict strategies are closed under composition and the identity is strict.

**Definition** The category \( \mathcal{G}_s \) has negative games as objects and strict strategies as maps. The category \( \mathcal{W}_s \) has negative games as objects and strict winning strategies as maps.

Isomorphisms in \( \mathcal{W} \) correspond to forest isomorphisms and all isomorphisms are total and strict [28]. Each of the above categories can be endowed with symmetric monoidal structure, given by \( (I, \otimes) \) where \( I \) is the empty game \( 1 \) and the action of \( \otimes \) on objects is as defined in Section 2.1.

### 5.1. Sequoidal Closed Structure

The notions of *sequoidal category* and *sequoidal closed category* were first introduced in [21].

**Definition** A *sequoidal category* consists of:

- A symmetric monoidal category \( (\mathcal{C}, I, \otimes) \) (we will call the relevant isomorphisms \( \text{assoc} : (A \otimes B) \otimes C \cong A \otimes (B \otimes C), \text{lunit}_\otimes : I \otimes A \cong A, \text{runit}_\otimes : A \otimes I \cong I \) and \( \text{sym} : A \otimes B \cong B \otimes A \))

- A category \( \mathcal{C}_s \)

- A right-action \( \otimes \) of \( \mathcal{C} \) on \( \mathcal{C}_s \). That is, a functor \( \cdot \otimes \cdot : \mathcal{C}_s \times \mathcal{C} \to \mathcal{C}_s \) with natural isomorphisms \( \text{unit}_\otimes : A \otimes I \cong A \) and \( \text{pasc} : A \otimes (B \otimes C) \cong (A \otimes B) \otimes C \) satisfying
An inclusive sequoidal category is a sequoidal category in which \( \mathcal{C}_s \) is a full-on-objects subcategory of \( \mathcal{C} \) containing \( \text{wk} \) and the monoidal isomorphisms; \( J \) is the inclusion functor; and \( J \) reflects isomorphisms.

We can identify this structure in our categories of games: we can extend the left-merge operator \( \odot \) to an action \( \mathcal{G}_s \times \mathcal{G} \to \mathcal{G}_s \). If \( \sigma : A \to B \) and \( \tau : C \to D \) then \( \sigma \odot \tau : A \odot C \to B \odot D \) plays as \( \sigma \) between \( A \) and \( B \) and as \( \tau \) between \( C \) and \( D \). The strictness of \( \sigma \) guarantees that this yields a valid strategy on \( (A \odot C) \odot (B \odot D) \). The isomorphisms \( \text{pasc} \) and \( \text{unit}_\odot \) exist, and there is a natural copycat strategy \( \text{wk} : M \odot N \to M \odot N \) in \( \mathcal{G}_s \), all satisfying the required axioms [24]. The functor \( J \) reflects isomorphisms as the inverse of strict isomorphisms are strict. Thus \( (\mathcal{G}, \mathcal{G}_s) \) forms an inclusive sequoidal category; as does \( (W, W_s) \).

**Definition** An inclusive sequoidal category is **Cartesian** if \( \mathcal{C}_s \) has finite products preserved by \( J \) (we will write \( t_A \) for the unique map \( A \to 1 \)). It is **decomposable** if the natural transformations \( \text{dec} = (\text{wk}, \text{wk} \circ \text{sym}) : A \odot B \Rightarrow (A \odot B) \odot (B \odot A) : \mathcal{C}_s \times \mathcal{C}_s \to \mathcal{C}_s \) and \( \text{dec}^0 = t_I : 1 \Rightarrow 1 : \mathcal{C}_s \) are isomorphisms (so, in particular, \( (\mathcal{C}, \odot, 1) \) is an affine SMC).

A Cartesian sequoidal category is **distributive** if the natural transformations \( \text{dist} = (\pi_1 \circ \text{id}_C, \pi_2 \circ \text{id}_C) : (A \times B) \odot C \Rightarrow (A \odot C) \times (B \odot C) : \mathcal{C}_s \times \mathcal{C}_s \times \mathcal{C} \to \mathcal{C}_s \) and \( \text{dist}^0 = t_{1 \odot C} : 1 \odot C \Rightarrow 1 : \mathcal{C} \to \mathcal{C}_s \) are isomorphisms.
We write $\text{dist}^0 : I \otimes C \cong I$ for the isomorphism $(\text{dec}^0)^{-1} \circ \text{dist}_0 \circ (\text{dec}^0 \otimes \text{id})$.

In the game categories defined above, $M \& N$ is a product of $M$ and $N$, and the empty game $I$ is a terminal object as well as the monoidal unit. The decomposability and distributivity isomorphisms above exist as natural copycat morphisms [24]. In fact, $\mathcal{W}$ and $\mathcal{G}$ have all small products, following the construction in Section 2.3) with the corresponding distributivity isomorphism with respect to $\otimes$.

**Definition** A sequoidal closed category is an inclusive sequoidal category where $\mathcal{C}$ is symmetric monoidal closed and the map $f \mapsto \Lambda(f \circ \text{wk})$ defines a natural isomorphism $\Lambda_s : \mathcal{C}_s(B \otimes A, C) \Rightarrow \mathcal{C}_s(B, A \rightarrow C)$.

We can show that $\mathcal{G}$ and $\mathcal{W}$ are sequoidal closed, with the internal hom given by $\rightarrow [24]$. In any sequoidal closed category, define $\text{app}_s : (A \rightarrow B) \otimes A \rightarrow B$ as $\Lambda_s^{-1}(\text{id})$, and $\text{app} : (A \rightarrow B) \otimes A \rightarrow B = \Lambda^{-1}(\text{id})$, noting that $\text{app} = \text{app}_s \circ \text{wk}$. If $f : A \rightarrow B$ let $\Lambda(f) : I \rightarrow A \rightarrow B$ denote the name of $f$, i.e. $\Lambda(f \circ \text{runit}_\otimes)$. Write $\Lambda_r^{-1}$ for the inverse operation.

**Proposition 5.2.** In any sequoidal closed category, $\rightarrow$ restricts to a functor $\mathcal{C}^\text{op} \times \mathcal{C}_s \rightarrow \mathcal{C}_s$ with natural isomorphisms $\text{unit}_\rightarrow : I \rightarrow A \cong A$ and $\text{pasc}_\rightarrow : A \otimes B \rightarrow C \cong A \rightarrow (B \rightarrow C)$ in $\mathcal{C}_s$.

**Proof** We need to show that if $g$ is in $\mathcal{C}_s$ then $f \rightarrow g$ is in $\mathcal{C}_s$. But $f \rightarrow g = \Lambda(g \circ \text{app} \circ (\text{id} \otimes f)) = \Lambda(g \circ \text{app}_s \circ \text{wk} \circ (\text{id} \otimes f)) = \Lambda(g \circ \text{app}_s \circ (\text{id} \otimes f) \circ \text{wk}) = \Lambda_s(g \circ \text{app}_s \circ (\text{id} \otimes f))$ which is in $\mathcal{C}_s$.

In any symmetric monoidal category the isomorphisms $\text{unit}_\rightarrow$ and $\text{pasc}_\rightarrow$ exist, but we must show that they are strict.

- $\text{unit}_\rightarrow : I \rightarrow A \rightarrow A$ is given by $\text{app} \circ \text{runit}_\rightarrow^{-1}$. This $\text{app}_s \circ \text{wk} \circ \text{runit}_\rightarrow^{-1} = \text{app}_s \circ \text{unit}_\rightarrow^{-1}$ which is a map in $\mathcal{C}_s$.
- $\text{pasc}_\rightarrow : A \otimes B \rightarrow C \cong A \rightarrow (B \rightarrow C)$ is given by $\Lambda(\Lambda(\text{app} \circ \text{assoc})) = \Lambda(\Lambda(\text{app}_s \circ \text{wk} \circ \text{assoc})) = \Lambda(\Lambda(\text{app}_s \circ \text{pasc}^{-1} \circ \text{wk} \circ (\text{wk} \circ \text{id}))) = \Lambda(\Lambda(\text{app}_s \circ \text{pasc}^{-1} \circ \text{wk}) \circ \text{wk}) = \Lambda_s(\Lambda(s(\text{app}_s \circ \text{pasc}^{-1})))$ which is in $\mathcal{C}_s$.

The inverses of the above maps are strict as $J$ reflects isomorphisms. $\square$

In distributive, decomposable sequoidal closed categories we can also define the following natural transformations:

- The isomorphism $\text{psym} : (A \otimes B) \otimes C \cong (A \otimes C) \otimes B$ given by $\text{pasc}_\rightarrow \circ (\text{sym} \circ \text{id}) \circ \text{pasc}_\rightarrow^{-1}$.
- The isomorphism $\text{psym}_\rightarrow : C \rightarrow (B \rightarrow A) \cong B \rightarrow (C \rightarrow A)$ given by $\text{pasc}_\rightarrow \circ (\text{sym} \rightarrow \text{id}) \circ \text{pasc}_\rightarrow^{-1}$.
- The isomorphism $\text{dist}_\rightarrow : A \rightarrow (B \times C) \rightarrow (A \rightarrow B) \times (A \rightarrow C)$ given by $\langle \text{id} \rightarrow \pi_1, \text{id} \rightarrow \pi_2 \rangle$, whose inverse is $\Lambda(\text{app} \circ (\pi_1 \circ \text{id}), \text{app} \circ (\pi_2 \circ \text{id}))$. This isomorphism exists in any monoidal closed category with products.
- The map $\text{af} : A \Rightarrow I$ given by $(\text{dec}^0)^{-1} \circ \text{t}_A$. 30
• The isomorphism \( \text{dist}_f : A \rightarrow I \rightarrow I \) given by \( af \) whose inverse is \( \Lambda(\text{runit}_\otimes \circ (\text{id} \otimes af)) \). We must check that these are inverses: \( af \circ \Lambda(\text{runit}_\otimes \circ (\text{id} \otimes af)) = \text{id} \) as both are maps into the terminal object, and \( \Lambda(\text{runit}_\otimes \circ (\text{id} \otimes af)) \circ af = \Lambda(\text{runit}_\otimes \circ (af \otimes \text{id}) \circ (af \otimes \text{id})) = \Lambda(\text{app}) = \text{id} \) as required. We know that \( \text{runit}_\otimes \circ (af \otimes \text{id}) \circ (af \otimes \text{id}) = \text{app} \) as both are maps into the terminal object.

We can use the structure described above to model the negative connectives of WS1. We will represent positive connectives indirectly, inspired by the fact that strategies on the negative game \( P \rightarrow o \) is the one-move game \( o \). The object \( o \) satisfies a special property: an internalised version of linear functional extensionality \([1]\).

**Definition** An object \( o \) in a sequoidal closed category satisfies linear functional extensionality if the natural transformation \( \text{ife} : (B \rightarrow o) \otimes A \Rightarrow (A \rightarrow B) \rightarrow o : C \times C^{op} \rightarrow C \) given by \( \Lambda_s \circ \text{id} \circ \text{app} \circ (\text{id} \otimes \text{sym}) \circ \text{opasc}^{-1} \) is an isomorphism.

The linear functional extensionality property is characteristic of our history sensitive, locally alternating games model \([21]\): it does not hold in other sequoidal closed categories (e.g. Conway games \([21]\)).

Using linear functional extensionality we can give a natural isomorphism \( \text{abs} : o \otimes A \cong o \) by noticing that \( o \otimes A \cong (I \rightarrow o) \otimes A \cong (A \rightarrow I) \rightarrow o \cong I \rightarrow o \cong o \), and thus setting \( \text{abs} = \text{unit}_\rightarrow \circ ((\text{dist}_o)^{-1} \rightarrow \text{id}) \circ \text{ife} \circ (\text{unit}_1^{-1} \otimes \text{id}) \).

### 5.2. Coalgebraic Exponential Comonoid

We next consider the categorical status of the exponential operator \( ! \). We interpret the core introduction rules for the exponentials, and the key anamorphism rule, by requiring that it is the carrier for a final coalgebra of the functor \( X \rightarrow N \otimes X \).

Recall that a coalgebra for a functor \( F : C \rightarrow C \) is an object \( A \) and a map \( A \rightarrow F(A) \). A final coalgebra is a terminal object in the category of coalgebras, that is a coalgebra \( \alpha : Z \rightarrow F(Z) \) such that for any \( f : A \rightarrow F(A) \) there is a unique \( \xi f : A \rightarrow Z \) such that \( \alpha \circ \xi f = F(\xi f \triangleright) \circ f \).

\[
\begin{array}{ccc}
A & \xrightarrow{f} & F(A) \\
Z & \xrightarrow{\alpha} & F(Z) \\
\end{array}
\]

We call \( \xi f \triangleright > \) the anamorphism of \( f \). Note in particular that if \( (Z, \alpha) \) is a final coalgebra for \( F \), then \( \alpha \) is an isomorphism, with inverse \( \alpha^{-1} = \xi (F(\alpha) \triangleright) \).

In \( W \) we define a coalgebra \((!N, \alpha)\) by taking \( \alpha : !N \rightarrow N \otimes !N \) to be the evident copycat strategy which relabels \( \text{in}_1(a) \) on the right to \( (a,1) \) on the left and \( \text{in}_2(a,n) \) on the right to \( (a,n+1) \) on the left.

**Proposition 5.3.** \((!N, \alpha)\) is the final coalgebra of the functor \( N \otimes \cdot \) in the category \( \mathcal{G} \).

31
Proof Let \( \sigma : M \to N \otimes M \). Define \( \zeta \sigma \! \vdash_n : M \to (N \otimes \_)^n(M) \) by \( \zeta \sigma \! \vdash_0 = \text{id} \) and \( \zeta \sigma \! \vdash_{n+1} = (\text{id} \otimes \_)^n(\sigma) \circ \zeta \sigma \! \vdash_n \).

\[
M \xrightarrow{\zeta \sigma} (N \otimes \_)^n(M) \xrightarrow{(\text{id} \otimes \_)^n(\sigma)} (N \otimes \_)^n(N \otimes M) = (N \otimes \_)^{n+1}(M)
\]

The strategy \( \zeta \sigma \! \vdash_n \) is a partial axiom to \( \zeta \sigma \! \vdash : M \to ! N \). We can show by induction on \( n \) that \( \zeta \sigma \! \vdash_n = (\text{id} \otimes \zeta \sigma \! \vdash_n) \circ \sigma \). Similarly, we can define \( \alpha_k : ! N \cong (N \otimes \_)^k(! N) : \alpha_k \ominus 1 \) by performing the above construction on \( \alpha \). Consider the sequence of maps \( M \to ! N \) defined by \( s_k = \alpha_k \ominus 1 \circ (\text{id} \otimes \_)^k(\epsilon) \circ \zeta \sigma \! \vdash_k \) for \( k \in \omega \). We can show that \( s_{k+1} \supseteq s_k \) by induction on \( k \), and so \( (s_k) \) is a chain. Set \( \zeta \sigma \! \vdash = \bigsqcup \alpha_k \ominus 1 \circ (\text{id} \otimes \_)^k(\epsilon) \circ \zeta \sigma \! \vdash_k \), where \( \epsilon \) is the empty strategy. It is well-known that \( \zeta \) is e-pco-enriched with bottom element \( \epsilon \) [23].

We wish to show that \( \zeta \sigma \! \vdash \) is the unique strategy such that \( \alpha \circ \zeta \sigma \! \vdash = (\text{id} \otimes \zeta \sigma \! \vdash) \circ \sigma \). To show that the equation holds, note that \( \alpha \circ \zeta \sigma \! \vdash = \alpha \circ \bigsqcup \alpha_k \ominus 1 \circ (\text{id} \otimes \_)^k(\epsilon) \circ \zeta \sigma \! \vdash_k = \alpha \circ \bigsqcup \alpha_k \ominus 1 \circ (\text{id} \otimes \_)^{k+1}(\epsilon) \circ \zeta \sigma \! \vdash_{k+1} = \bigsqcup (\text{id} \circ \zeta \sigma \! \vdash_k) \circ \sigma = (\text{id} \otimes \bigsqcup (\alpha_k \ominus 1 \circ (\text{id} \otimes \_)^k(\epsilon) \circ \zeta \sigma \! \vdash_k) \circ \sigma = (\text{id} \otimes \zeta \sigma \! \vdash) \circ \sigma \).

For uniqueness, suppose that \( \gamma : M \to ! N \) is such that \( \alpha \circ \gamma = (\text{id} \otimes \gamma) \circ \sigma \). We wish to show that \( \gamma = \zeta \sigma \! \vdash = \bigsqcup \alpha_k \ominus 1 \circ (\text{id} \otimes \_)^k(\epsilon) \circ \zeta \sigma \! \vdash_k \). To see that \( \gamma \supseteq \zeta \sigma \! \vdash \), it suffices to show that \( \gamma \) is an upper bound of the chain, i.e. \( \gamma \supseteq \alpha_k \ominus 1 \circ (\text{id} \otimes \_)^k(\epsilon) \circ \zeta \sigma \! \vdash_k \) for each \( k \). This can be shown using a simple induction on \( k \). To see that \( \gamma \subseteq \zeta \sigma \! \vdash \), we show that each play in \( \gamma \) is also in \( \zeta \sigma \! \vdash \). Consider a play \( s \in \gamma : M \to ! N \). Since \( s \) is finite, it must visit only a finite number of copies of \( N \) — say, \( k \) copies. Then \( s \) is also a play in \( \alpha_k \ominus 1 \circ (\text{id} \otimes \_)^k(\epsilon) \circ \alpha_k \circ \gamma \).

It is thus sufficient to show that \( (\text{id} \otimes \_)^k(\epsilon) \circ \alpha_k \circ \gamma \subseteq (\text{id} \otimes \_)^k(\epsilon) \circ \zeta \sigma \! \vdash_k \). This is achieved by a simple induction on \( k \). \( \Box \)

**Proposition 5.4.** \( (! N, \alpha) \) is the final coalgebra of \( N \otimes \_ \) in the category \( \mathcal{W} \).

**Proof** It suffices to show that if \( \sigma : M \to N \otimes M \) is a winning strategy, then \( \zeta \sigma \! \vdash \) is winning.

To see that \( \zeta \sigma \! \vdash \) is total, let \( s \in \zeta \sigma \! \vdash \) and so \( s \in \mathcal{P}N \). Then \( s \) visits only finite \( k \) many copies of \( N \), and so up to retagging it is a play in \( M \to (N \otimes \_)^k(M) \), and \( s \) plays a play in \( \zeta \sigma \! \vdash_k \). By totality of \( \zeta \sigma \! \vdash_k \), there is a move \( w \) with \( \zeta \sigma \! \vdash_k \supseteq \zeta \sigma \! \vdash_k \). Then, up to retagging, \( \text{sup} \) is also a play in \( \zeta \sigma \! \vdash_k \).

We next need to check that each infinite play with all even prefixes in \( \zeta \sigma \! \vdash \) is winning. Let \( s \) be such an infinite play, with \( s \mid_M \) winning. We must show that \( s \mid_N \) is winning, i.e. \( s \mid_{(N,i)} \) is winning for each \( i \). The infinite play \( s \) corresponds to an infinite interaction sequence:

\[
M \xrightarrow{\sigma} N \otimes M \xrightarrow{\text{id} \otimes \sigma} N \otimes (N \otimes M) \xrightarrow{\text{id} \otimes (\text{id} \otimes \sigma)} \ldots
\]

Then \( s \mid_{(N,i)} \) can also be found in the \( i \)th column of the above interaction sequence. By hiding all columns other than the first and the \( i \), we see a play in \( M \to (N \otimes \_)^i(M) \) in \( \zeta \sigma \! \vdash_i \). The first column is \( s \mid_M \) (which is winning), and the \( i \)th component of the
Recall that the monoidal unit of a distributive sequoidal category is a terminal object. Thus we may define operations corresponding to dereliction and promotion:

- \( \text{der}_N : N \rightarrow N = \text{unit}_\otimes \circ (\text{id}_\otimes t) \circ \alpha \).
- Given any symmetric comonoid \((B, \eta, \delta)\), and morphism \( f : B \rightarrow N \), let \( f^\dagger : B \rightarrow !N \) be the (comonoid morphism) \( \xi \text{wk} \circ (f \otimes \text{id}) \circ \delta \).

To interpret the contraction rule, we require a further coalgebraic property.

**Definition** A decomposable, distributive sequoidal category \( C \) has coalgebraic monoidal exponentials if:

- For any object \( A \), the endofunctor \( A \otimes - \) has a specified final coalgebra \((!A, \alpha_A)\).
- For any objects \( A, B \), \((!A \otimes !B, \alpha_{A,B})\) is a final coalgebra for the endofunctor \((A \times B) \otimes - \) where \( \alpha_{A,B} : !A \otimes !B \rightarrow (A \times B) \otimes (!A \otimes !B) \) is the isomorphism:

\[
\begin{align*}
!A \otimes !B &\cong (!A \otimes !B) \times (!B \otimes !A) \\
&\cong ((A \otimes !A) \otimes !B) \times ((B \otimes !B) \otimes !A) \\
&\cong (A \otimes (!A \otimes !B)) \times (B \otimes (!B \otimes !B)) \\
&\cong (A \times B) \otimes (!A \otimes !B)
\end{align*}
\]

The second requirement is equivalent to requiring that the morphism \( \langle \text{der}_A \otimes t, t \otimes \text{der}_B \rangle^\dagger \) from \(!A \otimes !B\) to \(!A \times B\) is an isomorphism. Thus we may define a comonoid \((!A, \delta : !A \rightarrow !A \otimes !A, t : A \rightarrow I)\), where \( \delta \) is the anamorphism of the map \( \text{dist}_{A,A,!A} \circ (\alpha_A, \alpha_A) \).

**Proposition 5.5.** If \( C \) has coalgebraic monoidal exponentials then \((!A, \delta, t)\) is the cofree commutative comonoid on \( A \).

**Proof** In other words, the forgetful functor from the category of comonoids on \( C \) into the category \( C \) has a left adjoint which sends \( A \) to \((!A, \delta, t)\). The unit of this adjunction is the dereliction \( \text{der}_A : !A \rightarrow A \): for any \( f : B \rightarrow A \), \( f^\dagger : B \rightarrow !A \) is the unique comonoid morphism such that \( \text{der} \circ f^\dagger \). (Uniqueness follows from finality of \(!A\)).

This cofree commutative comonoid can also be constructed using the technique described in [35]. This approach builds the exponential as a limit of finitary symmetric tensor powers, that is, finite tensor products subject to a quotient so that the order that the components are played in is irrelevant. Our use of the asymmetric \( \otimes \) enforces a strict left-to-right order, providing a concrete (albeit less generally applicable) alternative to such quotienting.

**Proposition 5.6.** The sequoidal closed categories \( W \) and \( G \) are both equipped with coalgebraic monoidal exponentials.

**Proof** Follows from Propositions 5.3, 5.4 and the fact \(!\) is the cofree commutative comonoid in \( G \) and \( W \).[24]

**Definition** A \( \text{WS}!\)-category is a distributive, decomposable sequoidal closed category with an object \( o \) satisfying linear functional extensionality and coalgebraic monoidal exponentials.

**Proposition 5.7.** The categories \((G, G_s)\) and \((W, W_s)\) enjoy the structure of an \( \text{WS}!\)-category.
5.3. Semantics of Rules

We may now describe the interpretation of the rules of our logic (other than those for atoms, quantifiers and equality) in a WS!-category $C$. Suppose that, for a given context of variables and atoms $\Phi$, we have an interpretation of formulas and sequents over $\Phi$ as objects of $C$, satisfying the following:

\[
\begin{align*}
[\Phi \vdash 1] &= I \\
[\Phi \vdash \bot] &= \varnothing \\
[\Phi \vdash M \otimes N] &= [\Phi \vdash M] \otimes [\Phi \vdash N] \\
[\Phi \vdash M \& N] &= [\Phi \vdash M] \times [\Phi \vdash N] \\
[\Phi \vdash M \otimes N] &= [\Phi \vdash M] \otimes [\Phi \vdash N] \\
[\Phi \vdash M < Q] &= [\Phi \vdash Q] \to [\Phi \vdash M] \\
[\Phi \vdash \lnot N] &= [\Phi \vdash N] \\
[\Phi \vdash M, \Gamma, N] &= [\Phi \vdash M, \Gamma] \circ [\Phi \vdash N] \\
[\Phi \vdash M, \Gamma, P] &= [\Phi \vdash P] \rightharpoonup [\Phi \vdash M, \Gamma] \\
[\Phi \vdash P, \Gamma, N] &= [\Phi \vdash N] \rightharpoonup [\Phi \vdash P, \Gamma] \\
[\Phi \vdash P, \Gamma, Q] &= [\Phi \vdash P, \Gamma] \circ [\Phi \vdash Q] \\
[\Gamma, M]^+ &= [\Gamma]^+ \otimes [M]^+ \\
[\Gamma, P]^+ &= [\Gamma]^+ \circ [P]^+ \\
[\Gamma, M]^\ominus &= [\Gamma]^\ominus \otimes [M]^\ominus \\
[\Gamma, P]^\ominus &= [\Gamma]^\ominus \circ [P]^\ominus \\
\end{align*}
\]

(For atom and quantifier-free formulas, these equations define an interpretation of formulas and sequents in $C$.) Then we may give an interpretation of each proof rule except those for atoms, quantifiers, and equality as an operation on morphisms in $C$. These typically involve an operation on the head formula of the sequence “under” a context consisting of its tail, and so we define distributivity maps to allow this:

We define endofunctors $[\Gamma]^b$ on $C$, for each context (possibly empty list of formulas) $\Gamma$ and $b \in \{+, -, \}$, below.

\[
\begin{align*}
[c]^+ &= \text{id} & [c]^\ominus &= \text{id} \\
[\Gamma, M]^+ &= [\Gamma]^+ \otimes [M]^+ \\
[\Gamma, P]^+ &= [\Gamma]^+ \circ [P]^+ \\
[\Gamma, M]^\ominus &= [\Gamma]^\ominus \otimes [M]^\ominus \\
[\Gamma, P]^\ominus &= [\Gamma]^\ominus \circ [P]^\ominus \\
\end{align*}
\]

**Proposition 5.8.** For any sequent $A, \Gamma$ we have $[A, \Gamma] = [\Gamma]^b([A])$ where $b$ is the polarity of $A$.

**Proof** A simple induction on $\Gamma$. \qed

**Proposition 5.9.** For any context $\Gamma$, $[\Gamma]^b$ preserves products.

**Proof** Using the distributivity of $\otimes$ over $\circ$ and $\rightharpoonup$, we can construct isomorphisms $\text{dist}_{b,\Gamma} : [\Gamma]^b(A \times B) \cong [\Gamma]^b(A) \times [\Gamma]^b(B)$ and $\text{dist}^{\ominus}_{b,\Gamma} : [\Gamma]^b(\bot) \cong \mathcal{I}$ by induction on $\Gamma$. 

\[34 \]


### 5.4. Semantics of Proof Rules

Define $\sigma : \vdash \Gamma$ if:

- $\Gamma = N, \Gamma'$, and $\sigma : I \to [\Gamma]$ in $C$.
- $\Gamma = P, \Gamma'$, and $\sigma : [\Gamma'] \to o \in C$.

Semantics of the core rules as operations on morphisms are given in Figure 9 and the other rules in Figures 10 and 11. The rules involving the exponential are treated separately in Figure 12. Note that in each case, the interpretation in the WS!-category $W$ agrees with the informal exposition in Section 3.2.

In the semantics of $P_{\text{out}}$ we use an additional construction. If $\tau : I \to [N, \Delta]$ define (strict) $\tau_{M,\Gamma}^\Delta : [M, \Gamma, N^1] \to [M, \Gamma, \Delta]$ to be $\text{unit}_\to \circ (\tau \to \text{id}_{[M, \Gamma, I]})$ if $|\Delta| = 0$ and $\text{pasc}_o \circ (\Lambda^{-n} \Lambda_1^{-1} \tau \to \text{id}_{[M, \Gamma, I]})$ if $|\Delta| = n + 1$. Define (strict) $\tau_{P,\Gamma}^\Delta : [P, \Gamma, \Delta] \to [P, \Gamma, N^1]$ to be $(\text{id}_{[P, \Gamma]} \circ \tau) \circ \text{unit}_\to^1$ if $|\Delta| = 0$ and $(\text{id} \circ \Lambda^{-n} \Lambda_1^{-1} \tau) \circ ((\text{id}_{[P, \Gamma]} \circ \text{sym}) \circ \text{pasc}_1^{-1})^n$ if $|\Delta| = n + 1$. In some of the rules in Figure 11 we omit some $\text{pasc}$ isomorphisms for clarity.

### 6. Semantics of atoms, quantifiers and equality

We shall now complete the semantics of $WS1$ by interpreting atoms and quantifiers based on our categories of games and strategies. (The requisite structure could be axiomatised for any WS!-category, but we shall not do so here.) We have seen that a sequent
Figure 10: Categorical Semantics for WS1 (other rules, part 1)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta \vdash (\text{psym}) \circ \sigma : [- M', \Gamma, M, N, \Delta] )</td>
<td>( \Delta \vdash (\text{psym}) \circ \sigma : [- M', \Gamma, M, N, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- M', \Gamma, M, N, \Delta] )</td>
<td>( \sigma : [- M', \Gamma, M, N, \Delta] )</td>
</tr>
<tr>
<td>( \Delta \vdash (\text{psym}) \circ \sigma : [- M', \Gamma, M, N, \Delta] )</td>
<td>( \sigma : [- M', \Gamma, M, N, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- M', \Gamma, P, Q, \Delta] )</td>
<td>( \sigma : [- P', \Gamma, P, Q, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- M', \Gamma, P, Q, \Delta] )</td>
<td>( \sigma : [- P', \Gamma, P, Q, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- M, \Gamma, M, \Delta] )</td>
<td>( \sigma : [- M, \Gamma, M, \Delta] )</td>
</tr>
<tr>
<td>( \Delta \vdash (\text{id} \circ \text{id}) \circ \sigma : [- M', \Gamma, M, N, \Delta] )</td>
<td>( \sigma : [- M', \Gamma, M, N, \Delta] )</td>
</tr>
<tr>
<td>( \Delta \vdash (\text{id} \circ \text{id}) \circ \sigma : [- M', \Gamma, M, N, \Delta] )</td>
<td>( \sigma : [- M', \Gamma, M, N, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- M, \Gamma, M, N, \Delta] )</td>
<td>( \sigma : [- M, \Gamma, M, N, \Delta] )</td>
</tr>
<tr>
<td>( \Delta \vdash (\text{pasc}) \circ \sigma : [- M', \Gamma, M \odot N, \Delta] )</td>
<td>( \sigma : [- M', \Gamma, M \odot N, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- M', \Gamma, M \odot N, \Delta] )</td>
<td>( \sigma : [- M', \Gamma, M \odot N, \Delta] )</td>
</tr>
<tr>
<td>( \Delta \vdash (\text{pasc}) \circ \sigma : [- M', \Gamma, M \odot N, \Delta] )</td>
<td>( \sigma : [- M', \Gamma, M \odot N, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- M, \Gamma, P, Q, \Delta] )</td>
<td>( \sigma : [- M, \Gamma, P, Q, \Delta] )</td>
</tr>
<tr>
<td>( \Delta \vdash (\text{pasc}) \circ \sigma : [- M', \Gamma, P \odot \text{q}, \Delta] )</td>
<td>( \sigma : [- M', \Gamma, P \odot \text{q}, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- M, \Gamma, P, Q, \Delta] )</td>
<td>( \sigma : [- M, \Gamma, P, Q, \Delta] )</td>
</tr>
<tr>
<td>( \Delta \vdash (\text{pasc}) \circ \sigma : [- M', \Gamma, M \odot N, \Delta] )</td>
<td>( \sigma : [- M', \Gamma, M \odot N, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- M, \Gamma, P, Q, \Delta] )</td>
<td>( \sigma : [- M, \Gamma, P, Q, \Delta] )</td>
</tr>
<tr>
<td>( \Delta \vdash (\text{id} \circ \text{id}) \circ \sigma : [- M', \Gamma, P_1 \odot P_2, \Delta] )</td>
<td>( \sigma : [- M', \Gamma, P_1 \odot P_2, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- M, \Gamma, M_1 \odot M_2, \Delta] )</td>
<td>( \sigma : [- M, \Gamma, M_1 \odot M_2, \Delta] )</td>
</tr>
<tr>
<td>( \Delta \vdash (\text{id} \circ \text{id}) \circ \sigma : [- N, \Gamma, M_1 \odot M_2, \Delta] )</td>
<td>( \sigma : [- N, \Gamma, M_1 \odot M_2, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- N, \Gamma, M_1 \odot M_2, \Delta] )</td>
<td>( \sigma : [- N, \Gamma, M_1 \odot M_2, \Delta] )</td>
</tr>
<tr>
<td>( \Delta \vdash (\text{id} \circ \text{id}) \circ \sigma : [- N, \Gamma, M_1 \odot M_2, \Delta] )</td>
<td>( \sigma : [- N, \Gamma, M_1 \odot M_2, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- Q, \Gamma, P_1 \odot P_2, \Delta] )</td>
<td>( \sigma : [- Q, \Gamma, P_1 \odot P_2, \Delta] )</td>
</tr>
<tr>
<td>( \sigma : [- Q, \Gamma, M_1 \odot M_2, \Delta] )</td>
<td>( \sigma : [- Q, \Gamma, M_1 \odot M_2, \Delta] )</td>
</tr>
</tbody>
</table>
Figure 11: Categorical Semantics for WS1 (other rules, part 2)

\[
\begin{align*}
P_{\text{mul}} & \quad \sigma : \vdash M, \Gamma, \Delta^+, \pi : \vdash N, \Delta^+ \\
\Lambda_I(\Lambda_I^{-1}(\sigma) \otimes \Lambda_I^{-1}(\tau)) : \vdash M, \Gamma, N, \Delta^+, \Delta^+ \\
P_{\text{cut}} & \quad \sigma : \vdash M, \Gamma, N^+, \Gamma_1 \\
\tau : \vdash N, \Delta^+ \\
\vdash \langle \tau_{\text{M,F}} \rangle \circ \sigma : \vdash M, \Gamma, \Delta^+, \Gamma_1 \\
P_{\text{cut}} & \quad \sigma : \vdash P, \Gamma, N^+, \Gamma_1 \\
\tau : \vdash N, \Delta^+ \\
\sigma \circ \langle \Gamma_1 \rangle^+ (\tau_{P,F}) : \vdash P, \Gamma, \Delta^+, \Gamma_1 \\
P_{\text{id}} & \quad \Lambda_I(\text{id}) : \vdash N, N^+ \\
P_{\text{cut}}^0 & \quad \sigma : \vdash N^+ \\
\tau : \vdash N, Q \\
\sigma \circ \Lambda_I^{-1}(\tau) : \vdash Q \\
P_{\text{psym}} & \quad \Lambda_I(\Lambda_I^{-1}(\tau) \rightarrow \Lambda_I^{-1}(\sigma)) : \vdash M, \Gamma, P, \Delta \triangleleft N, \Delta^+ \\
P_{\text{id} \otimes} & \quad \Lambda_I((\text{id} \otimes \Lambda_I^{-1}(\sigma) \circ \text{sym}) \circ \text{pasc} \circ \Lambda_I^{-1}(\sigma) \circ \text{wk} \circ \text{sym}) : \vdash M, N, M^+ \triangleleft Q, \Delta^+ \\
\end{align*}
\]

\(X : \Theta \vdash \Gamma\) of WS1 can be interpreted as a family of games, indexed over \(\Theta\)-satisfying \(\mathcal{L}\)-models over \(X\). We shall interpret a proof of \(X : \Theta \vdash \Gamma\) as a uniform family of strategies for each such game.

For example, the family denoted by \(\top \otimes (\phi \triangleleft \top)\) has games of the following form:

```
\begin{center}
\begin{tikzpicture}
\node (root) at (0,2) {q};
\node (p) at (0,0) {p};
\node (a) at (0,-2) {a};
\draw (root) -- (p);
\draw (p) -- (a);
\end{tikzpicture}
\end{center}
```

Here we represent the forest of plays \(P_A\) directly. The moves in dotted circles are only available if \((L, v) \models \phi\). There is a unique total strategy on the (positive) game above in both cases, and this family is uniform in the sense that the strategy on models which satisfy \(\phi\) is a substrategy of the strategy on models satisfying \(\neg\phi\) — if \((L, v) \models \phi\) and \((L', v') \models \neg\phi\) then \(\sigma_{[\top \otimes (\phi \triangleleft \top)][(L, v)]} \subseteq \sigma_{[\top \otimes (\phi \triangleleft \top)][(L', v')]\).

In contrast, consider the formula \(\bot \triangleleft (\neg\phi \oplus (\top \otimes \phi))\). The game forest is given as follows, using the same notation as above:
We wish to formalise categorically the notion of a game \( A \) being a subgame of \( B \): we can then state that a family of strategies is uniform if whenever \( A \) is a subgame of \( B \), the
Definition Let $\mathcal{C}$ be a poset-enriched category. The category $\mathcal{C}_e$ has the same objects as $\mathcal{C}$ and a map $A \to B$ in $\mathcal{C}_e$ consists of a pair $(i_f, p_f)$ where $i_f : A \to B$ and $p_f : B \to A$ in $\mathcal{C}$, such that $p_f \circ i_f = \text{id}$ and $i_f \circ p_f \subseteq \text{id}$.

- The identity is given by $(\text{id}, \text{id})$.
- For composition, set $(i_f, p_f) \circ (i_g, p_g) = (i_f \circ i_g, p_g \circ p_f)$. We need to check this is a valid pairing: $p_{f \circ g} \circ i_{f \circ g} = p_g \circ p_f \circ i_f \circ i_g = p_g \circ \text{id} \circ i_g = \text{id}$ and $i_{f \circ g} \circ p_{f \circ g} = i_f \circ i_g \circ p_g \circ p_f \subseteq i_f \circ \text{id} \circ p_f = i_f \circ p_f \subseteq \text{id}$.
- It is clear that composition is associative and that $f = f \circ \text{id} = \text{id} \circ f$.

Let $\mathcal{G}$ denote the poset-enriched category of games and (not-necessarily winning) strategies, and $\mathcal{G}_s$ its subcategory of strict strategies, with $\subseteq$ given by strategy inclusion. A tree embedding of $A$ into $B$ corresponds to a map $A \to B$ in $\mathcal{G}_e$.

**Proposition 6.1.** If $f : A \to B$ in $\mathcal{G}_e$ then $i_f$ and $p_f$ are strict.

**Proof** If $i_f$ responds to an opening move in $B$ with a move in $B$ then so does $i_f \circ p_f$ and so $i_f \circ p_f \subseteq \text{id}$ fails. Similarly, if $p_f$ responds to an opening move in $A$ with a move in $A$ then so does $p_f \circ i_f$ and so $p_f \circ i_f = \text{id}$ fails.

We can thus define identity-on-objects functors $i : \mathcal{G}_e \to \mathcal{G}_s$ and $p : \mathcal{G}_e \to \mathcal{G}_s^{op}$. We can show that our operations on games lift to functors on $\mathcal{G}_e$.

**Proposition 6.2.** All of the operations $\to, \Theta, \otimes, \&,$! extend to covariant (bi)functors on $\mathcal{G}_e$.

**Proof** Each case exploits functoriality and monotonicity of the relevant operation. We just give an example: set $(i, p) \to (i', p') = (p \to i', i \to p')$. Then $(i \to i') \circ (p \to p') = (p \circ i) \to (p' \circ i') = \text{id} \to \text{id} = \text{id}$ and $(p \to i') \circ (i \to p') = (i \circ p) \to (i' \circ p') \subseteq \text{id} \to \text{id} = \text{id}$.

### 6.1.2. Lax natural Transformations

Given an embedding $e : A \to B$ and strategies $\sigma_A : A, \sigma_B : B, \sigma_B$ restricts to $\sigma_A$ if $\sigma_A = p_e \circ \sigma_B$. We generalise this idea using the notion of lax natural transformations.

**Definition** Let $\mathcal{C}$ be a category, $\mathcal{D}$ a poset-enriched category and $F, G : \mathcal{C} \to \mathcal{D}$. A lax natural transformation $F \Rightarrow G$ is a family of arrows $\mu_A : F(A) \to G(A)$ such that $\eta_B \circ F(f) \equiv G(f) \circ \eta_A$.
We can compose lax natural transformations using vertical composition. There is also a form of horizontal composition, provided that one of the two functors is the identity: Let $H, G : C \to D$ and $\mu : G \Rightarrow H$ a lax natural transformation. Then

(a) if $F : B \to C$ then there is a lax natural transformation $\mu F : G F \Rightarrow H F$ given by $(\mu F)_A = \mu_{F(A)}$ and

(b) if $J : D \to E$ is monotonic then there is a lax natural transformation $J \mu : J G \Rightarrow J H$ given by $(J \mu)_A = J(\mu_A)$.

### 6.1.3. Uniform Winning Strategies

**Definition** Let $F, G : C \to G_c$. A uniform strategy from $F$ to $G$ is a lax natural transformation $\sigma : i \circ F \Rightarrow i \circ G$. A uniform total strategy is a uniform strategy $\sigma$ where each $\sigma_A$ is total. A uniform winning strategy is a uniform strategy where each $\sigma_A$ is winning.

If $f : A \to B$, the lax naturality condition is that $i_{G(f)} \circ \sigma_A \subseteq \sigma_B \circ i_{F(f)}$. Thus $\sigma_A = p_{G(f)} \circ i_{G(f)} \circ \sigma_A \subseteq p_{G(f)} \circ \sigma_B \circ i_{F(f)}$. But since $\sigma_A$ is total, it is maximal in the ordering $\subseteq$ and we must have $\sigma_A = p_{G(f)} \circ \sigma_B \circ i_{F(f)}$. Similarly, we see that $\sigma_A = p_{G(f)} \circ \sigma_B \circ i_{F(f)}$ implies the lax naturality condition as $i_{G(f)} \circ \sigma_A = i_{G(f)} \circ p_{G(f)} \circ \sigma_B \circ i_{F(f)} \subseteq \sigma_B \circ i_{F(f)}$. Thus, lax naturality captures the fact that $\sigma_A$ is determined by $\sigma_B$ via restriction. If $F$ is the constant functor $k_I$, this reduces to $\sigma_A = p_{G(o)} \circ \sigma_B$.

We can construct a WS-category of uniform strategies over a base category $C$. Let $G^c$ be the category where:

- **Objects** are functors $C \to G_c$
- **An arrow** $F \to G$ is a uniform strategy $F \Rightarrow G$
- **Composition** is given by vertical composition of lax natural transformations
- **The identity** on a functor $F$ is given by the lax natural transformation $\eta : F \Rightarrow F$ where $\eta_A = \text{id}_{F(A)}$. It is clear that this is lax natural.

Similarly, we can construct a category $W^c$ of functors and uniform winning strategies.

**Proposition 6.3.** $G^c$ is a WS!-category.

**Proof** We first exhibit the symmetric monoidal structure. $F \otimes G$ is defined to be $\otimes \circ (F \times G) \circ \Delta$ where $\Delta : C \to C \times C$ is the diagonal. So, $(F \otimes G)(A) = F(A) \otimes G(A)$.

On arrows, we set $(\eta \otimes \rho)_A = \eta_A \otimes \rho_A$. We need to show that if $f : L \to K$ then $(i_{A(f)} \otimes i_{C(f)}) \circ (\eta_K \otimes \rho_K) \subseteq (i_{B(f)} \otimes i_{D(f)}) \circ (\eta_L \otimes \rho_L)$. That is, we need to show that $(i_{A(f)} \circ \eta_K) \circ (i_{C(f)} \circ \rho_K) \subseteq (i_{B(f)} \circ \eta_L) \circ (i_{D(f)} \circ \rho_L)$. But this is clear by lax naturality of $\eta$ and $\rho$ and monotonicity of $\otimes$.

The tensor unit $I$ is the constant functor, sending all objects to the game $I$ and arrows to $\text{id}_I$.

The morphisms $\text{assoc}$, $\text{runit}_\otimes$, $\text{lunit}_\otimes$ and $\text{sym}$ are defined pointwise: for example, $(\text{assoc}_{F,G,H})_X = \text{assoc}_{F(X),G(X),H(X)}$. To check for lax naturality, we must use horizontal composition. For example, consider the map $\text{assoc} : (F \otimes G) \otimes H \to F \otimes (G \otimes H)$ defined pointwise as described. The domain is $(F \otimes G) \otimes H = ((\otimes \otimes) \otimes) \circ (i \circ F \times i \circ G \times i \circ H) \circ \Delta_3$ where $\Delta_3$ is the diagonal functor $C \to C \times C$. Similarly, the codomain is $(\otimes (\otimes \otimes) \circ (i \circ F \times i \circ G \times i \circ H) \circ \Delta_3$. We can thus see that $\text{assoc}$ is equal to the
We know that \( f \) is a natural transformation \( \Delta_3 \) where \( J = (i \circ F \times i \circ G \times i \circ H) \circ \Delta_3 \) and \( \text{assoc} \) is the natural transformation \( \circ \circ (\circ \circ .) \Rightarrow (\circ \circ .) \circ \circ \) in \( \mathcal{G}_s \).

\[
\begin{array}{c}
\Delta_3 \\
\Delta_3
\end{array}
\begin{array}{ccc}
\mathcal{C} & \xrightarrow{\Delta_3} & \mathcal{C} \times \mathcal{C} \\
\id & & \id \\
\downarrow & & \downarrow \\
\mathcal{C} \times \mathcal{C} \times \mathcal{C} & \xrightarrow{i \circ F \times i \circ G \times i \circ H} & \mathcal{G}_s \times \mathcal{G}_s \times \mathcal{G}_s \\
\assoc & & \assoc \\
\downarrow & & \downarrow \\
\mathcal{C} \times \mathcal{C} & \xrightarrow{\Delta_3} & \mathcal{C} \times \mathcal{C} \times \mathcal{C} \\
\id & & \id \\
\downarrow & & \downarrow \\
\mathcal{C} \times \mathcal{C} \times \mathcal{C} \times \mathcal{C} & \xrightarrow{i \circ F \times i \circ G \times i \circ H} & \mathcal{G}_s \times \mathcal{G}_s \times \mathcal{G}_s \times \mathcal{G}_s \\
\assoc & & \assoc \\
\end{array}
\]

One can similarly express the other monoidal isomorphisms in this way to see lax naturality. The coherence equations of symmetric monoidal categories inherit pointwise from \( \mathcal{G} \).

Symmetric monoidal closure, products, sequoidal closure and linear functional extensionality lift pointwise from \( \mathcal{G} \) using horizontal composition. We can also show that the coalgebraic monoidal exponential structure lifts from \( \mathcal{G} \).

**Proposition 6.4.** \( \mathcal{W}^C \) is a WS!-category.

**Proof** We proceed precisely as in Proposition 6.3, lifting the structure of a WS!-category in \( \mathcal{W} \) to that in \( \mathcal{W}^C \). In particular, pointwise-winningness of the relevant morphisms in \( \mathcal{W}^C \) inherits from the winningness in \( \mathcal{W} \).

### 6.2. Quantifiers

#### 6.2.1. Category of \( \mathcal{L} \)-structures

**Definition** Given a set of variables \( X \) and set of atomic formulas \( \Theta \), we let \( \mathcal{M}^X_\Theta \) denote the category of \( \Theta \)-satisfying \( \mathcal{L} \)-models over \( X \). Objects are \( \mathcal{L} \)-models over \( X \) that satisfy each formula in \( \Theta \). A morphism \((L, v) \to (L', v')\) is a map \( f : |L| \to |L'| \) such that:

- For each \( x \in X \), \( v'(x) = f(v(x)) \)
- If \((L, v) \models \overrightarrow{\phi} \) for \( \overrightarrow{a} \in |L|^{\text{ar}(\phi)} \) then \((L', v') \models \overrightarrow{f(\phi)} \)
- For each function symbol \( g \in \mathcal{L} \), \( f(I_L(g)(\overrightarrow{a})) = I_{L'}(g)(\overrightarrow{f(a)}) \).

Note that since the positive atoms include inequality, such morphisms must be injective. Also note that if \( f : (L, v) \to (L', v') \) and \((L, v) \models \overrightarrow{\phi} \) then \((L', v') \models \overrightarrow{f(\phi)} \).

If \( v \) is a valuation on \( X \), define \( v[x \mapsto l] \) on \( X \cup \{x\} \) to be the valuation sending \( y \) to \( v(y) \) if \( y \neq x \), and \( x \) to \( l \). Given \( f : (L, v) \to (M, w) \) in \( \mathcal{M}^X_\Theta \) and \( s \) a term with \( \text{FV}(s) \subseteq X \), \( f \) is also a map \( (L, v[x \mapsto v(s)]) \to (M, w[x \mapsto w(s)]) \) in \( \mathcal{M}_{\Theta}^{X_{\cup \{x\}}} \).

We know that \( f \) preserves all of the valuations other than \( x \), and for \( x \) we see that \( f(v[x \mapsto v(s)](x)) = f(v(s)) = w(s) = w[x \mapsto w(s)](x) \).

We will give semantics of sequents \( X; \Theta \vdash \Gamma \) as functors \( \mathcal{M}_\Theta^X \to \mathcal{G}_c \), and proofs as uniform winning strategies.
6.2.2. Quantifiers as Adjoints

In this section, we will describe an adjunction that will allow us to interpret the quantifiers.

- If \( FV(s) \subseteq X \) we can define a functor \( \text{set}^x_s : \mathcal{M}^\Theta_X \to \mathcal{M}^\Theta_{X \cup \{x\}} \) by \( \text{set}^x_s(L, v) = (L, v[x \mapsto v(s)]) \) and if \( f : (L, v) \to (M, w) \) we set \( \text{set}^x_s(f) = f \). We need to check that \( \text{set}^x_s(f) \) is a valid morphism. We know that \( \text{set}^x_s(f) \) preserves all variables in \( X \), and \( \text{set}^x_s(f)(v[x \mapsto v(s)](x)) = f(v(s)) = w(s) = w[x \mapsto w(s)](x) \) as required. It is clear that \( \text{set}^x_s \) is functorial.

From this we can extract a functor \( \text{set}^x_s : \mathcal{W}^{\mathcal{M}^\Theta_{X \cup \{x\}}} \to \mathcal{W}^{\mathcal{M}^\Theta_X} \), mapping \( F \) to \( F \circ \text{set}^x_s \), with an action on arrows defined by horizontal composition.

- Provided \( x \) does not occur in \( \Theta \), there is an evident forgetful functor \( U_x : \mathcal{M}^\Theta_{X \cup \{x\}} \to \mathcal{M}^\Theta_X \) mapping \( (L, v) \) to \( (L, v \setminus x) \). From this we can extract a functor \( U'_x : \mathcal{W}^{\mathcal{M}^\Theta_X} \to \mathcal{W}^{\mathcal{M}^\Theta_{X \cup \{x\}}} \) mapping \( F \) to \( F \circ U_x \), with an action on arrows defined by horizontal composition. Note that \( U_x \circ \text{set}^x_s = \text{id} \) and so \( \text{set}^x_s \circ U'_x = \text{id} \).

We will show that \( U'_x \) has a right adjoint \( \forall x \_ \). Assuming empty \( \Gamma \), this allows us to interpret the rules \( P \_ \) and \( P \_ \).

**Definition** Let \( C \) be a category. We define the category \( \text{FamInj}(C) \). An object is a set \( I \) and a family of \( C \)-objects \( \{ A_i : i \in I \} \). An arrow \( \{ A_i : i \in I \} \to \{ B_j : j \in J \} \) is a pair \( (f, \{ f_i : i \in I \}) \) where \( f \) is an injective function \( I \to J \) and each \( f_i : A_i \to B_{f(i)} \). We will often write such a map as \( (f, \{ f_i \}) \) when we wish to leave the indexing set implicit.

- Composition is defined by \( (f, \{ f_i \}) \circ (g, \{ g_i \}) = (f \circ g, \{ f(g(i)) \}) \).

- The identity \( \{ A_i : i \in I \} \to \{ A_i : i \in I \} \) is given by \( (\text{id}, \{ \text{id}_{A_i} \}) \).

- Satisfaction of the categorical axioms is inherited from \( C \).

**Definition** Let \( F : C \to D \). We define \( \text{FamInj}(F) : \text{FamInj}(C) \to \text{FamInj}(D) \). On objects, \( \text{FamInj}(F)(\{ A_i : i \in I \}) = \{ F(A_i) : i \in I \} \). On arrows, we set \( \text{FamInj}(F)(f, \{ f_i \}) = (f, \{ F(f_i) \}) \).

We define a distributivity functor \( \text{dst} : \text{FamInj}(C) \times D \to \text{FamInj}(C \times D) \) by \( \text{dst}(\{ A_i : i \in I \}, B) = \{ (A_i, B) : i \in I \} \) and \( \text{dst}(f, \{ f_i \}, g) = (f, \{ f_i \circ g \}) \).

Suppose \( F \) is an object in \( \mathcal{W}^{\mathcal{M}^\Theta_{X \cup \{x\}}} \) (a functor \( \mathcal{M}^\Theta_{X \cup \{x\}} \to \mathcal{G}_e \)). We define \( \forall x . F \) as an object in \( \mathcal{W}^{\mathcal{M}^\Theta_X} \) (a functor \( \mathcal{M}^\Theta_X \to \mathcal{G}_e \)). We first define a product functor \( \text{FamInj}(\mathcal{G}_e) \to \mathcal{G}_e \). We will define a product functor \( \text{FamInj}(\mathcal{G}_e) \to \mathcal{G}_e \). On objects, \( \text{prod} \) sends \( \{ G_i : i \in I \} \) to \( \prod_{i \in I} G_i \). On arrows, let \( f : \{ G_j : j \in J \} \to \{ H_h : h \in H \} \). The embedding part of \( \text{prod}(f) \) is given by \( \langle g_{h(i)} \rangle \) where \( g_h = i_{f(j)} \circ \pi_j \) if \( h = f(j) \) and \( \epsilon \) otherwise. The projection part is given by \( \langle p_{f(j)} \circ \pi_{h(j)} \rangle \).

We can check that \( \text{prod} \) defines a functor into \( \mathcal{G}_e \). Finally, given \( F : \mathcal{M}^\Theta_{X \cup \{x\}} \to \mathcal{G}_e \) we define \( \forall x . F : \mathcal{M}^\Theta_X \to \mathcal{G}_e \) to be \( \text{prod} \circ \text{FamInj}(F) \circ \text{add}_x \).

**Proposition 6.5.** The functor \( U'_x : \mathcal{W}^{\mathcal{M}^\Theta_X} \to \mathcal{W}^{\mathcal{M}^\Theta_{X \cup \{x\}}} \) has a right adjoint given by \( \forall x . \_ = \text{prod} \circ \text{FamInj}(\_ \_ \circ \text{add}_x : \mathcal{W}^{\mathcal{M}^\Theta_{X \cup \{x\}}} \to \mathcal{W}^{\mathcal{M}^\Theta_X} \).
Proof We must first give the unit of this adjunction. For each \( F \), we must give a uniform winning strategy \( \eta : U'_2(\forall x.F) \Rightarrow F \). Such an \( \eta \) is a winning uniform strategy \( \text{prod} \circ \text{Famlnj}(F) \circ \text{add}_x \circ U_x \Rightarrow F \). Note that \((\text{prod} \circ \text{Famlnj}(F) \circ \text{add}_x \circ U_x)(L,v) = \text{prod}((\{ (L,v - x[x \mapsto l]) : l \in L \}) = \prod_{l \in L} F(L,v[x \mapsto l]) \). Thus \( \eta(L,v) \) must be a winning strategy \( \prod_{l \in L} F(L,v[x \mapsto l]) \rightarrow F(L,v) \) and we take \( \eta(L,v) = \pi_{v(x)} \). One can check that this transformation is lax natural.

Given \( f : U'_2(F) \rightarrow G \) we must show that there is a unique \( \hat{f} : F \rightarrow \forall x.G \) such that \( f = \eta_G \circ U'_2(\hat{f}) \). Let \( \hat{f} \) be such a uniform winning strategy. Then we must give winning strategies \( f(L,v) : F(L,v) \rightarrow \prod_{l \in L} G(L,v[x \mapsto l]) \). Set \( \hat{f}(L,v) = (\eta_l)_l \), where \( \eta_l : F(L,v) \rightarrow G(L,v[x \mapsto l]) \) is defined by \( f(L,v) \). We can check that \( \hat{f} \) satisfies lax naturality.

We next need to show that \( \hat{f} \) satisfies the universal property. First, we must show that \( f = \eta_G \circ U'_2(\hat{f}) \). It suffices to show that for each \( (L,v) \), \( f(L,v) = (\eta_G \circ U'_2(\hat{f}))_{L,v} \). Composition in is given by vertical composition. Thus, the RHS is given by \( \pi_v(x) \circ (\langle f_{(L,v[x \mapsto l])} \rangle) = f_{(L,v[x \mapsto l])} \) as required.

We need to show that \( \hat{f} : F \rightarrow \forall x.G \) is the unique uniform strategy satisfying \( f = \eta_G \circ U'_2(\hat{f}) \). Suppose \( h : F \rightarrow \forall x.G \) in \( \text{W}^{\mathcal{M}_X^N} \) satisfies this property. Then given \( (L,v) \) in \( \mathcal{M}^N_{X \subseteq \{x\}} \), we know that \( f(L,v) = \eta_G(L,v) \circ h(L,v-x) = \pi_v(x) \circ h(L,v-x) \). Let \( (L,v) \in \mathcal{M}^N_{X \subseteq \{x\}} \).

We must show that \( h(L,v) = f(L,v) \). But consider the model \( (L,v[x \mapsto l]) \). This is \( f(L,v[x \mapsto l]) = \pi_v(x) \circ h(L,v(x-[x-l])) = \pi_v(x) \circ h(L,v), \) as required.\( \square \)

If \( N : \mathcal{M}^N_{X \subseteq \{x\}} \rightarrow \mathcal{G}_c \) then on objects \( \forall x.N(L,v) = \prod_{l \in L} \{N(L,v[x \mapsto l]) \rightarrow G \} \). For the action of \( \forall x.N \) on arrows, suppose \( f : \langle L,v \rangle \rightarrow \langle L',w \rangle \). Then \( \forall x.N(f) : \prod_{l \in L} \{N(L,v[x \mapsto l]) \rightarrow G \} \). The projection part (left to right) is given by \( (g_m)_m \) where \( g_m = \epsilon \) if \( m \) is not in the image of \( f \), and \( g_m = i[N](f) \circ \pi_l \) if \( m = f(l) \) (note in this case \( l \) is unique by injectivity of \( f \)). The projection part is given by \( (p[N](f) \circ \pi_l)_l \).

Consider the map \( \text{set}^{\mathcal{G}_c}(\eta) : \forall x.F = \text{set}^2(U'_2(\forall x.F)) \rightarrow \text{set}^2(F) \) in the category \( \mathcal{M}^N_X \). Pointwise, \( \text{set}^2(\eta)(L,v) : \prod_{l \in L} F(L,v[x \mapsto l]) \rightarrow F(L,v[x \mapsto v(s)]) \) is given by \( \pi_v(x) \), and so we will write \( \pi_v \) for this map.

6.3. Semantics of Sequents

We define the semantics of sequents \( X; \Theta \vdash \Gamma \) as functors \( \mathcal{M}^N_{X \subseteq \{x\}} \rightarrow \mathcal{G}_c \) inductively, via the equations given in the previous section, extended with the following interpretations of atoms and quantifiers:

\[
\begin{align*}
[\Phi \vdash \phi(\vec{s})](L,v) & = I \text{ if } (L,v) \models \phi(\vec{s})\\
[\Phi \vdash \phi(\vec{s})](L,v) & = o \text{ if } (L,v) \models \phi(\vec{s})\\
[X; \Theta \vdash \forall x.N] & = \forall x. [X \cup \{x\}; \Theta \vdash N]\\
[X; \Theta \vdash \exists x.P] & = \forall x. [X \cup \{x\}; \Theta \vdash P]
\end{align*}
\]

In the case of atoms, the functors are specified pointwise on objects, and we must also define the (functorial) action on arrows. Let \( f : (L,v) \rightarrow (L',v') \). If the truth value of \( \phi(\vec{s}) \) is the same in \( (L,v) \) and \( (L',v) \), we use the identity embedding \((\text{id}, \text{id})\). If the
truth value of $\phi(\overline{s})$ is different, we must have $(L, v) \models \phi(\overline{s})$ and $(L', v) \models \overline{s}(\overline{s})$ since morphisms in $M_X$ preserve truth of positive atoms. Thus we need an embedding $I \rightarrow o$. We can take $(\epsilon_{I-o}, \epsilon_{o-I})$ where $\epsilon$ is the strategy containing just the empty sequence. Note that $\epsilon_{I-o} \circ \epsilon_{o-I} = \epsilon_I = id_I$ and $\epsilon_{I-o} \circ \epsilon_{o-I} = \epsilon \subseteq id_o$ ($\epsilon$ is the bottom element with respect to $\sqsubseteq$).

We must check functoriality. We have already noted that if the truth value of $\phi(\overline{s})$ is the same in $(L, v)$ and $(L', v')$ then $\phi(\overline{s})((f) = id$, so in particular $\phi(\overline{s})[id] = id$. For composition, suppose $f : (L, v) \rightarrow (L', v')$ and $g : (L', v') \rightarrow (L'', v''')$. We can consider the truth value of $\phi(\overline{s})$ in each of these models (only some cases are possible, as morphisms preserve truth of positive atoms).

<table>
<thead>
<tr>
<th>$(L, v) \models \phi(\overline{s})$</th>
<th>$(L', v') \models \phi(\overline{s})$</th>
<th>$(L'', v''') \models \phi(\overline{s})$</th>
<th>$(g \circ f) \models \phi(\overline{s})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi(\overline{s})$</td>
<td>$\phi(\overline{s})$</td>
<td>$\phi(\overline{s})$</td>
<td>$(id, id) \circ (id, id) = (id, id)$</td>
</tr>
<tr>
<td>$\phi(\overline{s})$</td>
<td>$\phi(\overline{s})$</td>
<td>$\phi(\overline{s})$</td>
<td>$(\epsilon, \epsilon) \circ (id, id) = (\epsilon, \epsilon)$</td>
</tr>
<tr>
<td>$\phi(\overline{s})$</td>
<td>$\phi(\overline{s})$</td>
<td>$\phi(\overline{s})$</td>
<td>$(id, id) \circ (id, \epsilon) = (id, \epsilon)$</td>
</tr>
<tr>
<td>$\phi(\overline{s})$</td>
<td>$\phi(\overline{s})$</td>
<td>$\phi(\overline{s})$</td>
<td>$(id, id) \circ (id, id) = (id, id)$</td>
</tr>
</tbody>
</table>

### 6.4. Semantics of Proofs

We now extend the semantics of proof rules given in the previous section with interpretations for the rules for quantifiers, atoms and equality, completing the semantics of WS1.

We first show that if $x \not\in FV(\Gamma)$ there is an isomorphism $\text{dist}_\Gamma : \forall \pi x. A, \Gamma \cong \forall \pi x. A, \Gamma$ in $W^{\mathcal{M}^O_X}$. Observe that there is a natural isomorphism

$$\text{dist}_\emptyset : \_ \circ \_ \circ (\text{prod} \times \text{id}) \Rightarrow \text{prod} \circ \text{Famlnj}(\_ \circ \_ \circ \_ \circ 0) \circ \text{dst} : \text{Famlnj}(\phi_s) \times G^s \rightarrow G^s$$

which is concretely a family of winning strategies

$$\text{prod}((\pi_i : i \in I)) \circ M \rightarrow \text{prod}((\pi_i : M : i \in I))$$

given by $\text{dist}_\emptyset = (\pi_i \circ \text{id})$. Each $\text{dist}_\emptyset$ is a natural isomorphism in $W_s$.

Similarly, we can define a natural isomorphism

$$\text{dist}_\rightarrow : \text{prod}((\pi_i : i \in I)) \Rightarrow M \rightarrow \text{prod}((\pi_i : i \in I))$$

which is concretely a family of winning strategies

$$\rightarrow \circ \_ \circ (\text{prod} \times \text{id}) \Rightarrow \text{prod} \circ \text{Famlnj}(\rightarrow \circ \_ \circ \_ \circ 0) \circ \text{dst} : \text{Famlnj}(\phi_s) \times G^s \rightarrow G^s$$

For each $\Gamma$, we can then construct a map

$$\text{dist}_{\rightarrow, \Gamma} : \Gamma^{\rightarrow}_\pi \circ (\text{prod} \times \text{id}) \cong \text{prod} \circ \text{Famlnj}(\Gamma^{\rightarrow}_\pi) \circ \text{dst} : \text{Famlnj}(\phi_s) \times \mathcal{M}^O_X \rightarrow G^s$$

proceeding by induction on $\Gamma$.

Finally, given a sequent $A, \Gamma$ we define $\text{dist}_r$ as the following horizontal composition, where $b$ is the polarity of $A$. It is easy to see by checking pointwise that the functor $\forall x. [A, \Gamma]$ is equal to the given decomposition.

$\begin{tikzpicture}
  \node (A0) at (0,0) {$\forall x. [A, \Gamma] \times \mathcal{M}^O_X$};
  \node (A1) at (0,1) {$\mathcal{M}^O_X \times \mathcal{M}^O_X$};
  \node (A2) at (0,2) {$\mathcal{M}^O_X \times \mathcal{M}^O_X$};

  \node (B0) at (3,0) {$\mathcal{M}^O_X \times \mathcal{M}^O_X$};
  \node (B1) at (3,1) {$\mathcal{M}^O_X \times \mathcal{M}^O_X$};
  \node (B2) at (3,2) {$\mathcal{M}^O_X \times \mathcal{M}^O_X$};

  \node (C0) at (6,0) {$\mathcal{M}^O_X \times \mathcal{M}^O_X$};
  \node (C1) at (6,1) {$\mathcal{M}^O_X \times \mathcal{M}^O_X$};
  \node (C2) at (6,2) {$\mathcal{M}^O_X \times \mathcal{M}^O_X$};

  \draw[->] (A0) -- (A1) node[midway, above] {$\text{add} \circ \text{id}$};
  \draw[->] (A1) -- (A2) node[midway, above] {$\text{Famlnj}(\phi_s) \times \text{id}$};
  \draw[->] (A2) -- (B0) node[midway, above] {$\text{prod} \circ \text{Famlnj}(\phi_s) \times \text{id}$};
  \draw[->] (B0) -- (B1) node[midway, above] {$\text{Famlnj}(\phi_s) \times \text{id}$};
  \draw[->] (B1) -- (B2) node[midway, above] {$\text{dist}_{\rightarrow, \Gamma} \circ (\text{prod} \times \text{id})$};
  \draw[->] (B2) -- (C0) node[midway, above] {$\phi_s$};

  \draw[->] (C0) -- (C1) node[midway, above] {$\text{add} \circ \text{id}$};
  \draw[->] (C1) -- (C2) node[midway, above] {$\text{Famlnj}(\phi_s) \times \text{id}$};
  \draw[->] (C2) -- (A0) node[midway, above] {$\phi_s$};
\end{tikzpicture}$
Since distr is a natural isomorphism, and pointwise winning, it is an isomorphism in \( \mathcal{W}_1^{\mathcal{M}_X^\alpha} \).

**Proposition 6.6.** \( \pi_{v(x)} \circ \text{distr}(L,v) = [\Gamma]^b(\pi_{v(x)}) \)

**Proof** We can check this by induction on \( \Gamma \), as in Proposition 5.9. \( \square \)

We next give semantics to the rules involving atoms and quantifiers. We first introduce some notation. Suppose \( C \) is the coproduct of two categories \( D \) and \( E \) (the disjoint union of the two categories, where there are no maps between them). If \( F : C \to G \) we write \( F|_D \) and \( F|_E \) for the restriction of \( F \) to \( D \) and \( E \) respectively. If \( \eta : F \Rightarrow G \) then we can restrict \( \eta \) to a natural transformation \( F|_D \Rightarrow G|_D \), and we write \( \eta|_D \) for this restriction. If \( \eta : F|_D \Rightarrow G|_D \) and \( \sigma : F|_E \Rightarrow G|_E \) then we write \( [\eta, \sigma]_{D,E} \) for the lax natural transformation defined by \( [\eta, \sigma]_A = \eta_A \) if \( A \in D \) and \( [\eta, \sigma]_A = \sigma_A \) if \( A \in E \). Lax naturality of \( [\eta, \sigma] \) inherits from lax naturality of \( \eta \) and \( \sigma \), since there are no maps between \( D \) and \( E \) when viewed as subcategories of \( C \). If \( C = \mathcal{M}_X^\alpha \) then we will write \( [\eta, \sigma]_{\alpha, \beta} \) for \( [\eta, \sigma]_{\mathcal{M}_X^{\alpha, \alpha}, \mathcal{M}_X^{\beta, \beta}} \).

We construct an isomorphism

\[
H_{x,y,z} : \mathcal{M}_X^{\theta, x=y} \cong \mathcal{M}_X^{\theta, [z \mapsto \frac{x}{y}]} : H_{x,y,z}^{-1}
\]

with \( H_{x,y,z}(M, v) = (M, v[z \mapsto v(x)] - x - y) \) and \( H_{x,y,z}^{-1}(M, v) = (M, v[x \mapsto v(z), y \mapsto v(z)] - z) \). We can show that \( [(X; \Theta \vdash \Gamma)[\frac{z}{x}, \frac{y}{z}]] = [X; \Theta, x = y \vdash \Gamma]H_{x,y,z}^{-1}\) by induction on \( \Gamma \).

Semantics of the rules involving atoms and quantifiers are given in Figure 13. We must justify lax naturality of \( P_{\exists t} \): the following diagram must lax commute:

\[
\begin{array}{ccc}
I & [P_{\forall t}(p)](M, w) & [\phi(\overrightarrow{x}), \Gamma](M, w) \\
\text{id} & \sqsupseteq & i[\phi(\overrightarrow{x}), \Gamma](f) \\
\text{I} & [P_{\exists t}(p)](L, v) & [\phi(\overrightarrow{x}), \Gamma](L, v)
\end{array}
\]

To see this, note that if \( (L, v) \) and \( (M, w) \) agree on \( \phi(\overrightarrow{x}) \) then the diagram lax commutes by lax naturality of \( \epsilon \) or \( [p] \). If they disagree, then we must have \( (L, v) \models \overrightarrow{\phi(\overrightarrow{x})} \) and \( (M, w) \models \overrightarrow{\phi(\overrightarrow{x})} \). We need to show that \( [P_{\forall t}(p)](L, v) \models i[\phi(\overrightarrow{x}), \Gamma](f) \circ [P_{\exists t}(p)](M, w) \). But \( [P_{\forall t}(p)](M, w) = p[\phi(\overrightarrow{x}), \Gamma](f) \circ [P_{\exists t}(p)](L, v) \) as both sides map into the terminal object, so \( [P_{\forall t}(p)](L, v) \models i[\phi(\overrightarrow{x}), \Gamma](f) \circ p[\phi(\overrightarrow{x}), \Gamma](f) \circ [P_{\exists t}(p)](M, w) \).

7. Full Completeness

We next show a full completeness result for the function-free fragment of WS1: in this section we assume that \( \mathcal{L} \) contains no function symbols. Thus, the only uses of the \( P_3 \) rule are of the form \( P_3^y \) where \( y \) is some variable in scope.
We show that the core rules suffice to represent any uniform winning strategy \( \sigma \) on a type object provided \( \sigma \) is bounded — i.e. there is a bound on the size of plays occurring in \( \sigma \). In particular, such a strategy is the semantics of a unique analytic proof — a proof using only the core rules, with some further restrictions on the use of the matching rule.

Given a sequent \( X; \Theta \vdash \Gamma \), we say \( \Theta \) is lean if it contains \( x \neq y \) for all distinct \( x \) and \( y \) in \( X \) and does not contain \( x \neq x \). We assume an arbitrary ordering on variables.

**Definition** A proof in WS1 is analytic if it uses only core rules and has the following additional restrictions:

- Rules other than \( P_\times \) and \( P_{m2}^{x,y,z} \) can only conclude sequents with a lean \( \Theta \).

- If \( P_{m2}^{x,y,z} \) is used to conclude \( X; \Theta \vdash \Gamma \) then \( \Theta \) does not contain \( w \neq w \) for any \( w \);

\( (x, y) \) is the least pair with \( x, y \in X \), \( x \neq y \) and \( x \neq y \notin \Theta \); and \( z \) is the least variable in \( \text{Fr}(X; \Theta \vdash \Gamma) \) (the least fresh variable).

**Theorem 7.1.** Let \( X; \Theta \vdash \Gamma \) be a sequent of WS1 and \( \sigma \) a bounded uniform winning strategy.
strategy on \([X;\Theta \vdash \Gamma]\). Then there is a unique analytic proof \(p\) of \(X;\Theta \vdash \Gamma\) with \([p] = \sigma\).

All strategies on the denotations of exponential-free 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).

The rest of this section sketches the proof of this full completeness result, and describes an extension to reify unbounded strategies as infinitary analytic proofs. We perform a semantics-guided proof search procedure, following \([14, 4, 27, 31]\).

### 7.1. Uniform Choice

When constructing a proof of a given sequent out of core rules there is a choice of which rule to use when the outermost head connective is \(\oplus\) (either \(P_1^\dagger\) or \(P_2^\dagger\)) or \(\exists\) (which \(s\) to use in \(P_s^\dagger\)). Our choice of rule will depend on the given strategy, depending on which component Player plays in first. However, the input to our procedure is a family of strategies, and we need to ensure that the same component choice is made in each strategy. We will next show that our uniformity condition ensures this.

**Proposition 7.2.** If \(\Theta\) is lean and \((L, v), (M, w) \in \mathcal{M}_X^\Theta\) there exists an \(\mathcal{L}\)-model \((L, v) \sqcup (M, w)\) with maps \(f_{(L,v,M,w)} : (L, v) \to (L, v) \sqcup (M, w)\) and \(g_{(L,v,M,w)} : (M, w) \to (L, v) \sqcup (M, w)\).

**Proof** 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 \(\text{inj}_1\). 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 \(\Theta\) is lean. \(g_{(L,v,M,w)}\) is defined similarly. \(\square\)

We also recall that if \(f : (L, v) \to (M, w)\) 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 7.3.** Let \(X;\Theta \vdash \Gamma\) be a sequent and suppose \(\Theta\) is lean. Then there exists an object in \(\mathcal{M}_X^\Theta\).

**Proof** Note that \(\Theta\) just contains positive atoms. We can take \((X,\text{id})\), with \((X,\text{id}) \models \Theta\). Then each formula in \(\Theta\) is satisfied: each such formula is either \(\bar{o}(\bar{x})\), or \(x \neq y\) for distinct \(x, y\). \(\square\)

We now use the above lemmas to show that in any uniform winning strategy on a sequent whose head formula is \(P \uplus Q\), either all strategies play their first move in \(P\), or all strategies play their first move in \(Q\).

**Proposition 7.4.** Let \(M_1, M_2 : \mathcal{M}_X^\Theta \to G_e\). Suppose \(\Theta\) is lean, and let \(\sigma : M_1 \times M_2 \Rightarrow o\) be a uniform total (resp. winning) strategy. Then \(\sigma = \tau \circ \pi_1\) for some uniform total (resp. winning) strategy \(\tau : M_1 \Rightarrow o\), or \(\sigma = \tau \circ \pi_2\) for some uniform total (resp. 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 $M_1(L,v) \times M_2(L,v) \models o$ we must respond to the initial Opponent-move either with a move in $M_1$ or a move in $M_2$ (the $\pi$-atomicity condition). 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 $(L,v) \cup (T,w)$ and let $k$ be such that $\tau_{(L,v) \cup (T,w)} = \tau_{(L,v) \cup (T,w)} \circ \pi_k$. By uniformity and totality, $\sigma_{(L,v)} = \tau_{(L,v) \cup (T,w)} \circ (M_1 \times M_2)(f_{L,v,T,w}) = \tau_{(L,v) \cup (T,w)} \circ \pi_k \circ (M_1 \times M_2)(f_{L,v,T,w}) = \tau_{(L,v) \cup (T,w)} \circ \pi_k \circ (M_1 \times M_2)(f_{L,v,T,w}) = \tau_{(L,v) \cup (T,w)} \circ \pi_k$. 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 $\tau_{(L,v)} \circ \pi_i$. In particular, we can take $i$ such that $\sigma_{(X, id)} = \tau_{(X, id)} \circ \pi_i$ where $(X, id)$ is as defined in Proposition 7.3 We only need to show that $\tau$ is lax natural. We can construct a natural transformations $\iota_1 : (id, e) : M_1 \to M_1 \times M_2$ and $\iota_2 : (\epsilon, id) : M_2 \to M_1 \times M_2$. Then $\tau = \sigma \circ \iota_i$, and so is lax natural.

We next show that in any uniform family of winning strategies on a sequent with head $\exists x. P$, Player chooses the same $x$ in each strategy component. Moreover, the chosen $x$ is the value of some variable in scope.

Proposition 7.5. Let $M : \mathcal{M}^p_{X \forall y(x)} \to \mathcal{G}_c$. Suppose $\Theta$ is lean, and let $\sigma : \forall x. M \Rightarrow o$ be a uniform total (resp. winning) strategy. Then there exists a unique variable $y \in X$ and uniform total (resp. winning) strategy $\tau : \mathcal{M}^p_{X \forall y} \Rightarrow o$ such that $\sigma = \tau \circ \pi_y$.

Proof We firstly show that given any $\mathcal{L}$-model $(L,v)$ there is some $x$ with $\sigma_{(L,v)} = \tau_{(L,v)} \circ \pi_x$. 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 valuation $\text{inj}_1$ and validating all positive atoms. Let $\sigma_{(L', \text{inj}_1)} = \tau_{(L', \text{inj}_1)} \circ \pi_{r'}$. Define $m_1 : (L,v) \to (L', \text{inj}_1)$ sending $v(x)$ to $\text{inj}_1(x)$, $u$ to $\text{inj}_2(a)$ and $v \in U_{(L,v)} = \{u\}$ to $\text{inj}_2(v)$. Then $\sigma_{(L,v)} = \sigma_{(L', \text{inj}_1)} \circ \pi_r \circ \forall x. M(m_1)$.

- If $r = \text{inj}_2(b)$ then this is $\sigma_{(L', \text{inj}_1)} \circ \epsilon$ which is $\epsilon$ as $\sigma_{(L', \text{inj}_1)}$ must be strict (as its total and a map into $a$). This is impossible.

- If $r = \text{inj}_2(x)$ then this is $\sigma_{(L', \text{inj}_1)} \circ M(m_1) \circ \pi_{v(x)}$, which is impossible by assumption.

- Hence we must have $r = \text{inj}_2(a)$.

Define $m_2 : (L,v) \to (L', \text{inj}_1)$ sending $v(x)$ to $\text{inj}_2(x)$, $u$ to $\text{inj}_2(b)$ and $v \in U_{(L,v)} = \{u\}$ to $\text{inj}_2(v)$. We can use similar reasoning to show that $r = \text{inj}_2(b)$. This is a contradiction.

Hence, given any $(L,v)$ there is some variable $x$ such that $\sigma_{(L,v)} = \tau_{(L,v)} \circ \pi_{v(x)}$.

Let $y \in X$ be the unique variable such that $\sigma_{(X, id)} = \tau_{(X, id)} \circ \pi_y$ where $(X, id)$ is constructed as in Proposition 7.3 We now show the stronger fact that $\sigma_{(L,v)} = \tau_{(L,v)} \circ \pi_{v(y)}$. Suppose that $\sigma_{(L,v)} = \tau_{(L,v)} \circ \pi_{v(x)}$ and $\sigma_{(L,v) \cup (X, id)} = \tau_{(L,v) \cup (X, id)} \circ \pi_{\text{inj}_1(z)}$. By lax naturality, $\tau_{(L,v)} \circ \pi_{v(x)} = \sigma_{(L,v)} = \sigma_{(L,v) \cup (X, id)} \circ \forall x. M(f_{L,v,X,id}) = \tau_{(L,v) \cup (X, id)} \circ \pi_{v(z)} \circ \forall x. M(f_{L,v,X,id})$. Since $\text{inj}_1(z) = f_{L,v,X,id}(v(z))$, we have $\sigma_{(L,v)} = \tau_{(L,v) \cup (X, id)} \circ M(f_{L,v,X,id}) \circ \pi_{v(z)}$ and so we must have $x = z$. By similar reasoning using $\forall x. M(f_{L,v,X,id})$, we see that $y = z$, so $x = y$.

Hence there is a variable $y$ such that for all $(L,v)$, $\sigma_{(L,v)} = \tau_{(L,v)} \circ \pi_{v(y)}$ for some $\tau_{(L,v)} : M(L,v[x \mapsto v(y)]) \Rightarrow o$. Since $\Theta$ is lean, $y$ is the unique variable such that
\[ \sigma_{(L,v)} = \tau_{(L,v)} \circ \pi_v(y). \]

Note that \( M(L,v[x \mapsto v(y)]) = M(\text{set}^{x}_v(L,v)). \) We can easily check that the resulting transformation \( \tau : M\text{set}^{x}_v \Rightarrow o \) is lax natural.

7.2. Reification of Strategies

We define a procedure \texttt{reify} which transforms a bounded uniform winning strategy on a formula object into a proof of that formula. It may be seen as a semantics-guided proof search procedure: given such a strategy \( \sigma \) on the interpretation of \( \Gamma \), \texttt{reify} finds a proof which denotes it. Reading upwards, the procedure first decomposes the head formula into a unit (nullary connective) using the head introduction rules. If this unit is \( 1 \), we are done. It cannot be \( 0 \), as there are no (total) strategies on this game. If the unit is \( \top \) or \( \bot \), the procedure then consolidates the tail of \( \Gamma \) into a single formula, using the core elimination rules. Once this is done, the head unit is removed using \( P_{+\bot} \) or \( P_{-\top} \), strictly decreasing the size of the sequent. These steps are then repeated until termination.

We further have to deal with equality: whenever a free variable is introduced, we must consider if it is equal to each of the other free variables using the \( P_{\text{ma}} \) rule.

Informally, if \( \Theta \) is not lean:

- If \( \Theta \) 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 \( \Theta \) 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 \( P_{x,y,z} \) using \( H_{x,y,z}^{-1} \).

If \( \Theta \) is lean, then:

- The case \( \Gamma = 0, \Gamma' \) is impossible: there are no total strategies on this game.
- If \( \Gamma = 1, \Gamma' \) then \( \sigma \) must be the empty strategy, since it is the unique total strategy on this game. This is the interpretation of the proof \( P_1 \).
- If \( \Gamma = \top \) then \( \sigma \) must similarly be the unique total strategy on this game, i.e. the interpretation of \( P_\top \).
- If \( \Gamma = \top, P, \Gamma' \) then \( \sigma \) can never play in \( P \) since if it did the play restricted to \( \top, P \) would not be alternating. Thus \( \sigma \) is a strategy on \( \top, \Gamma' \). We can call \texttt{reify} inductively yielding a proof of \( \vdash \top, \Gamma' \), and apply \( P_{+\top} \) to yield a proof of \( \top, P, \Gamma \).
- If \( \Gamma = \top, N, P, \Gamma' \) then \( \sigma \) is a total strategy on \( \top, N \triangleleft P, \Gamma \) up to retagging and we can proceed inductively using \( P_{\land} \). If \( \Gamma = \top, N, M, \Gamma' \) we can proceed similarly, using \( P_{\lor} \).
- If \( \Gamma = \top, N \) then \( \sigma \) is a total strategy on \( \downarrow N \); we can strip off the first move yielding a total strategy on \( N \), apply \texttt{reify} inductively yielding a proof of \( \vdash N \), and finally apply \( P_{-\top} \) yielding a proof of \( \vdash \top, N \).
- If \( \Gamma = \bot \) is impossible: there are no total strategies on this game. Other cases where \( \bot \) is the head formula proceed as with \( \top \): if the tail is a single positive formula, we remove the first move and apply \( P_{+\bot} \), otherwise we shorten the tail using \( P_{-\bot}, P_{\lor} \) or \( P_{\lor} \).
• If $\Gamma = A \otimes N, \Gamma'$ then $\sigma$ is also a strategy on $A, N, \Gamma$. We can call $\text{reify}$ inductively yielding a proof of $\vdash A, N, \Gamma$ that denotes $\sigma$, and apply $P_{\otimes}$. We can proceed similarly in the following case $\Gamma = A \lhd P, \Gamma'$.

• If $\Gamma = M \& N, \Gamma'$ then we can split $\sigma$ into those plays that start with $M$ and those that start with $N$. This yields total strategies on $M, \Gamma'$ and $N, \Gamma$ respectively, which we can $\text{reify}$ inductively and apply $P_{\&}$.

• If $\Gamma = M \otimes N, \Gamma'$ then we can split $\sigma$ into those plays that start with $M$ and those that start with $N$. This yields total strategies on $M, N, \Gamma$ and $N, M, \Gamma$ respectively, which we can $\text{reify}$ inductively and apply $P_{\otimes}$.

• If $\Gamma = P \otimes Q, \Gamma'$ then $\sigma$ specifies a first move that must either be in $P$ or in $Q$. In the former case, we have a strategy on $P, \Gamma$ and can $\text{reify}$ inductively, finally applying $P_{\otimes 1}$. In the latter case, we have a strategy on $Q, \Gamma'$ and can $\text{reify}$ inductively and apply $P_{\otimes 2}$. The case of $\Gamma = P \& Q, \Gamma'$ is similar.

• If the head formula is a positive atom $\varphi(x)$ then we must have $\overline{\varphi}(x)$ in $\Theta$, as otherwise there can be no uniform winning strategies on $[\Gamma]$ (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 $\varphi(x)$ then we can split the family $\sigma$ into those models that satisfy $\varphi(x)$ 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 winning strategy on $[\Theta, \varphi(x) \vdash \bot, \Gamma]$ and we can proceed inductively using $P_{\text{at}-}$.

• If $\sigma : [X; \Theta \vdash \Gamma = \forall x, N, \Gamma']$ then $\text{dist}_x \circ \sigma : I \Rightarrow \forall x, [N, \Gamma']$. Using our adjunction, this corresponds to a map $\eta \circ U_{\otimes}(\text{dist}_x \circ \sigma) : I \Rightarrow [N, \Gamma']$ in $\mathcal{W}^{\mathcal{A}_{\mathcal{M}_{\mathcal{X}_{\otimes}}(\cdot)}}$. We can then $\text{reify}$ this inductively to yield a proof of $X \cup \{x\}; \Theta \vdash N, \Gamma'$ and apply $P_{\otimes}$.

• If $\Gamma = \exists x.P, \Gamma'$ then $\sigma \circ \text{dist}_x, \Gamma' : \forall x. [P, \Gamma'] \Rightarrow o$. By Proposition 7.5 there is a unique $y$ and natural transformation $\tau : [P, \Gamma'] \text{set}^y_x \Rightarrow o$ such that $\sigma \circ \text{dist}_x, \Gamma' = \tau \circ \pi_y$. Since $x$ does not occur in $\Gamma$, we have $[P, \Gamma'] \text{set}^y_x = [P[y/x], \Gamma']$. This yields a lax natural transformation $[P[y/x], \Gamma'] \Rightarrow o$. We can then apply the inductive hypothesis use the $P_{\otimes}$ rule.

We will later show that $\text{reify}$ is well founded by giving a measure on sequents that decreases on each call to the inductive hypothesis.

### 7.3. Definition of $\text{reify}$

$\text{reify}$ is defined inductively in Figure 14. Following the above remarks, the following properties hold:

1a The unique map $i : \Theta \Rightarrow C(I, o)$ is a bijection.

1b The map $d = [\lambda x.f \circ \pi_1, \lambda y.f \circ \pi_2] : C(M, o) + C(N, o) \Rightarrow C(M \times N, o)$ is a bijection. (\textit{π-atomicity} \[1\]).

2 The map $\lambda \vdash o : C(I, M) \Rightarrow C(M \vdash o, I \vdash o)$ is a bijection.

50
For non-lean $\Theta$:
\[
\begin{align*}
\text{reify}_{X,x \neq x',w,\Theta}(&\sigma) = P_x \\
\text{reify}_{X,x'\neq x,w,\Theta}(&\sigma) = P_{m_x(x',x,w,\Theta)}(\text{reify}(\sigma |_{M_X^{x\mapsto x} \circ H_{x',y}^{-1}}), \text{reify}(\sigma |_{M_X^{x\mapsto x'}}))
\end{align*}
\]
if $(x, y) \in X \times X$ is least such that $x \neq y$ and $(x \neq y) \notin \Theta$ and $z$ is the least element in $Fr(X; \Theta \vdash \Gamma)$.

For lean $\Theta$:
\[
\begin{align*}
\text{reify}_{X,x \neq x',\Theta}(&\sigma) = P_{x\Theta}(\text{reify}(\sigma |_{M_X^{x\mapsto x} \circ \Theta})) \\
\text{reify}_{X,x'\neq x,\Theta}(&\sigma) = P_{x\Theta}(\text{reify}(\sigma)) \\
\text{reify}_{X,x \neq x',\Theta}(&\sigma) = P_{m_x(x',x,\Theta)}(\text{reify}(\eta \circ U_{\Theta}^x(\text{dist} \circ \sigma))) \\
\text{reify}_{X,x' \neq x,\Theta}(&\sigma) = P_{x\Theta}(\text{reify}(\sigma))where &\sigma \circ \text{dist}^{-1} = \tau \circ \pi_y \\
\text{reify}_{x,\Theta}(&\sigma) = P_1 \\
\text{reify}_{x,N,\Theta}(&\sigma) = P_1(\text{reify}_x(\text{abs} \circ \sigma)) \\
\text{reify}_{x,p,\Theta}(&\sigma) = P_1(\text{reify}_x(\text{abs} \circ \sigma)) \\
\text{reify}_{x,p,q,\Theta}(&\sigma) = P_1(\text{reify}_x(\text{abs} \circ \sigma)) where &\sigma = \text{dist} \circ [\Gamma]^- \circ (\text{dec} \circ \sigma) \\
\text{reify}_{M,K,\Theta}(&\sigma) = P_1(\text{reify}_x(\text{abs} \circ \sigma)) \\
\text{reify}_{M,K,\Theta}(&\sigma) = P_1(\text{reify}_x(\text{abs} \circ \sigma)) where &\sigma = \text{dist} \circ [\Gamma]^- \circ (\text{dec} \circ \sigma)
\end{align*}
\]
7.4. Termination of Reify

We next argue for termination of our procedure. Intuitively, the full completeness procedure first breaks down the head formula until it is $\bot$ 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 $P^+_{\bot}$ or $P^+_{\top}$, strictly reducing the size of the strategy. If $\Theta$ is not lean, the number of distinct variable pairs that are not declared distinct in $\Theta$ is reduced by using $P_{\text{ma}}$.

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 $\sigma$.
- The second measure is the length of $\Gamma$ as a list if the head of $\Gamma$ is $\bot$ or $\top$, and $\infty$ otherwise.
- The third measure is the size of the head formula of $\Gamma$.
- The fourth measure is $|\{(x, y) \in X \times X : x \not\equiv y \land x \not= y \notin \Theta\}|$

If $\Theta$ is lean:

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

If $\Theta$ is not lean and the $P_{\text{ma}}$ 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.

7.5. Soundness and Uniqueness

Lemma 7.6. For all $\sigma : \Gamma \vdash \Gamma$ we have $[\text{reify}_\Gamma(\sigma)] = \sigma$.

Proof We proceed by induction on our reification measure $([\Gamma], \text{tl}(\Gamma), \text{hd}(\Gamma))$ using equations that hold in the categorical model. We perform case analysis on $\Gamma$. The calculation is routine, we demonstrate only a few cases.

- If $\Theta$ is not lean with $(x, y) \in X \times X$ least such that $x \not= y$ and $(x \not= y) \notin \Theta$ and $z$ is the least element in Fr$(X; \Theta \vdash \Gamma)$, then $[\text{reify}(\sigma)]$
  $= [P^x_{\text{ma}} (\text{reify}(\sigma)|_{M^0_{X, x=y} \circ H^{-1}_{x,y,z}} \circ [\text{reify}(\sigma)|_{M^0_{X, x=y}}])]
  = [[\text{reify}(\sigma)|_{M^0_{X, x=y} \circ H^{-1}_{x,y,z}}]H_{x,y,z}, [\text{reify}(\sigma)|_{M^0_{X, x=y}}]]\mid_{x=y, x \not= y}
  = [\sigma|_{M^0_{X, x=y} \circ H^{-1}_{x,y,z} \circ H_{x,y,z}}\mid_{x=y, x \not= y}]
  = [\sigma|_{M^0_{X, x=y, z=x=y}}\mid_{x=y, x \not= y} = \sigma].$
• If $\Theta$ is lean and $\Gamma = \phi(\vec{x})$, $\Gamma'$ then
  \[
  \text{[reify}(\sigma)\text{]} = \|P_{\text{st-}(\sigma |_{M^\Theta_X, \pi(\vec{x})})}\| = \|\sigma |_{M^\Theta_X, \pi(\vec{x})}, \epsilon |_{\phi(\vec{x}), \phi(\vec{x})}\| = \sigma
  \]
as we must have $\sigma |_{M^\Theta_X, \sigma(\vec{x})} = \epsilon$ since $[\phi(\vec{x}), \Gamma']_A$ is the terminal object for each $A$ in $M^\Theta_X, \phi(\vec{x})$.

• If $\Gamma = \forall x. N, \Gamma'$ then $\text{[reify}(\sigma)\text{]} = \|P_T(\text{reify}(\eta \circ U_\sigma^{(dist_T \circ \sigma)}))\|$
  \[
  \text{dist}_{\pi_1} \circ \text{[reify}(\eta \circ U_\sigma^{(dist_T \circ \sigma)}))\| = \text{dist}_{\pi_1} \circ \text{[reify}(\eta \circ U_\sigma^{(dist_T \circ \sigma)})\| = \text{dist}_{\pi_1} \circ \text{dist}_{\pi_2} \circ \sigma = \sigma
  \]\nas required.

• If $\Gamma = P_1 \ll P_2, \Delta$ then $\text{[reify}\ll P_1, P_2, \Delta(\sigma)]\| = \|P_1 \circ \text{reify}_1, P_2, \Delta, P_2 \circ \text{reify}_2, P_1, \Delta\| \circ \text{d}^{-1}(\sigma \circ [\Delta]^+ \circ \text{dec}^{-1} \circ \text{dist}_{\pi_1}^{-1} \circ \Delta)\|$. Suppose $\text{d}^{-1}(\sigma \circ [\Delta]^+ \circ \text{dec}^{-1} \circ \text{dist}_{\pi_1}^{-1} \circ \Delta) = \text{in}(\pi)$, $\pi_1 \circ \sigma = \sigma \circ [\Delta]^+ \circ \text{dec}^{-1} \circ \text{dist}_{\pi_1}^{-1} \circ \Delta$.

If $i = 1$ then $\|P_1 \circ \text{reify}_1, P_1, \Delta, P_2 \circ \text{reify}_2, P_1, \Delta \circ \text{d}^{-1}(\sigma \circ [\Delta]^+ \circ \text{dec}^{-1} \circ \text{dist}_{\pi_1}^{-1} \circ \Delta)\| = \|P_1 \circ \text{reify}_1, P_1, \Delta, (\tau)\| \circ [\Delta]^+ \circ \text{dec}^{-1} \circ \text{dist}_{\pi_1}^{-1} \circ \Delta = \|P_1 \circ \text{reify}_1, P_1, \Delta, (\pi_1 \circ \text{dec})\| = \tau \circ \pi_1 \circ \text{dist}_{\pi_1}^{-1} \circ \Delta \circ [\Delta]^+ \circ \text{dec}^{-1} \circ \text{dist}_{\pi_1}^{-1} \circ \Delta = \sigma \circ \Delta \circ \text{dist}_{\pi_1}^{-1} \circ \Delta \circ \text{dist}_{\pi_1}^{-1} \circ \Delta = \sigma$.

The case for $i = 2$ is similar. \square

**Lemma 7.7.** For any analytic proof $p$ of $\Gamma$ we have $\text{reify}_T([p]) = p$.

**Proof** We proceed by induction on $p$. The calculation is routine, we demonstrate only a few cases.

• If $p = P_1^\perp(p')$ with $\Gamma = \perp, \Gamma$ then $\text{reify}_T([p]) = \text{reify}_T(\perp(p') = P_1^\perp(\text{reify}_T(\perp(p')) = P_1^\perp(p') = p$.

• If $p = P_\perp(p_1, p_2)$ with $\Gamma = M \& N, \Delta$ then $\text{reify}_T([p])$
  \[
  = P_{\perp}(\text{reify}_M, \Delta, \pi_1 \circ \text{dist}_{-\Delta} \circ [p_1], \text{reify}_N, \Delta, \pi_2 \circ \text{dist}_{-\Delta} \circ [p_2]) = P_{\perp}(\text{reify}_M, \Delta, \pi_1 \circ \text{dist}_{-\Delta} \circ [p_1], [p_2]) = P_{\perp}(\text{reify}_M, \Delta, [p_1], [p_2]) = P_{\perp}(p_1, p_2)
  \]

• If $p = P_\perp(p')$ with $\Gamma = P_1 \ll P_2, \Delta$ then $\text{reify}_T([p]) = \text{reify}_T([p'] \circ [\Delta]^+ \circ \text{dec}^{-1} \circ \text{dist}_{\pi_1}^{-1} \circ \Delta) = [P_\perp \circ \text{reify}_1, P_1, \Delta, P_2 \circ \text{reify}_2, P_1, \Delta \circ d^{-1}(p') \circ [\Delta]^+ \circ \text{dec}^{-1} \circ \text{dist}_{\pi_1}^{-1} \circ \Delta = [P_\perp \circ \text{reify}_1, P_1, \Delta, P_2 \circ \text{reify}_2, P_1, \Delta \circ d^{-1}(p') \circ [\Delta]^+ \circ \text{dec}^{-1} \circ \text{dist}_{\pi_1}^{-1} \circ \Delta = P_\perp(p') = p$ as required. \square

This completes our proof of Theorem 7.1.
7.6. Infinitary Analytic Proofs

We have seen that any bounded winning strategy is the denotation of a unique analytic proof of WS1. We cannot use this to normalise proofs to their analytic form because proofs do not necessarily denote bounded strategies. We will next show that our reification procedure can be extended to winning strategies that may be unbounded, provided the resulting analytic proofs are allowed to be infinitary — that is, proofs using the core rules that may be infinitely deep. More precisely, we will show that total strategies on a type object correspond precisely to the infinitary analytic proofs. Thus we can normalise any proof of WS1 to an infinitary normal form, by taking its semantics and then constructing the corresponding infinitary analytic proof. Two proofs of WS1 are semantically equivalent if and only if they have the same normal form as an infinitary analytic proof.

7.6.1. Infinitary Proofs as a Final Coalgebra

Let \( L \) be a set. Let \( T_L \) denote the final coalgebra of the functor \( X \mapsto L \times X^* \) in Set. The inhabitants of \( T_L \) are \( L \)-labelled trees of potentially infinite depth. We let \( \alpha : T_L \to L \times T_L^* \) describe the arrow part of this final coalgebra: this maps a tree to its label and sequence of subtrees. Given a natural number \( n \), we define a function \( N_n : T_L \to \mathcal{P}(L \times T_L^*) \), by induction:

\[
N_0(T) = \emptyset \quad \text{and} \quad N_{n+1}(T) = \{ \alpha(T) \} \cup \bigcup \{N_n(T') : T' \in \pi_2(\alpha(T)) \}.
\]

We define the set of nodes \( N(T) \) to be \( \{ N_n(T) : n \in \mathbb{N} \} \). Let \( \text{Prf} \) be the set of (names of) proof rules of WS1 and \( \text{Seq} \) the set of sequents of WS1.

**Definition** An infinitary analytic proof of WS1 is an infinitary proof using only the core rules of WS1. Formally, this is an element \( T \in I = \text{Prf} \times \text{Seq} \) such that for each node \( (((P_x, X; \Theta \vdash \Gamma), c) \in N(T) \) we have \(|c| = \text{ar}(P_x) \) and if \((\pi_2 \circ \pi_1 \circ \alpha)(c_i) = X_i; \Theta_i \vdash \Gamma_i \) then the following is a valid core rule of WS1:

\[
p_x \frac{X_1; \Theta_1 \vdash \Gamma_1 \quad \ldots \quad X_{|c|}; \Theta_{|c|} \vdash \Gamma_{|c|}}{X; \Theta \vdash \Gamma}
\]

We let \( I_{\Gamma} \) denote the set of infinitary analytic proofs of \( \vdash \Gamma \).

Let \( \{ A_{X; \Theta \vdash \Gamma} : X; \Theta \vdash \Gamma \in \text{Seq} \} \) be family of sets indexed by sequents. We can construct a family of maps \( A_{X; \Theta \vdash \Gamma} : I_{X; \Theta \vdash \Gamma} \to T_{X; \Theta \vdash \Gamma} \) by giving, for each \( X; \Theta \vdash \Gamma \) and \( a \in A_{X; \Theta \vdash \Gamma} \), a proof rule that concludes \( X; \Theta \vdash \Gamma \) from \( X_1; \Theta_1 \vdash \Gamma_1 \), \ldots, \( X_n; \Theta_n \vdash \Gamma_n \) and for each \( i \) an element \( a_i \in A_{X_i; \Theta_i \vdash \Gamma_i} \).

7.6.2. Infinitary Proofs as a Limit of Paraproofs

We can consider an alternative approach for presenting our infinitary analytic proofs. We consider partial proofs, that may “give up” in the style of [12].

**Definition** An analytic paraproof of WS1 is a proof made up of the core proof rules of WS1, together with a daemon rule that can prove any sequent:

\[
\Phi \vdash \Gamma \quad p_x
\]
7.6.3. Semantics of Infinitary Analytic Proofs

We next describe semantics of infinitary analytic proofs via the semantics of analytic paraproofs.

We can interpret analytic paraproofs as partial strategies. We interpret paraproofs of \( X; \Theta \vdash \Gamma \) in \( G^{M_\infty} \). For the rules other than \( P_5 \), we use the fact that \( G^{M_\infty} \) is a WS!-category. We interpret \( P_5 \) as the strategy \( \{\epsilon\} \) where \( \epsilon \) denotes the empty play on any game. We can hence interpret a analytic paraproof of \( \Gamma \) as a strategy on \( \ell_1 \).

The category \( G^{M_\infty} \) is cpo-enriched, with \( \sigma \subseteq \tau \) if for each \( A, \sigma_A \subseteq \tau_A \) as a set of plays. The bottom element is the uniform strategy that is \( \{\epsilon\} \) at each component. Composition, pairing and currying are continuous maps of hom sets; as are the operations used in the first-order structure.

**Proposition 7.8.** If \( p \) and \( q \) are analytic paraproofs of \( \Gamma \) and \( p \sqsubseteq q \) then \( [p] \sqsubseteq [q] \).

**Proof** A simple induction on the proof rules for WS1, using the fact that composition, pairing and currying are monotonic operations. Note that \( [-] \) is also strict, as \( [P_5] = \{\epsilon\} \).

Hom sets of \( G^{M_\infty} \) are algebraic domains: each strategy is the limit of its compact (finite) approximants. Our monotonic map \( C_\Gamma \rightarrow [X; \Theta \vdash \Gamma] \) thus extends uniquely to a continuous map \( D_\Gamma \rightarrow [X; \Theta \vdash \Gamma] \). By construction this agrees with the semantics given above for analytic paraproofs in \( D_\Gamma \). Given any infinitary analytic proof \( p \) if \( p \downarrow \) is the set of analytic paraproofs less than \( p \) then \( [p] = \bigsqcup [p] \downarrow \) using the cpo structure in \( G^{M_\infty} \).

We can show that this really does capture the intended semantics of infinitary analytic proofs.

**Proposition 7.9.** The equations for the semantics of analytic proofs given in Figures 4, 12, and 13 hold for infinitary analytic proofs.

**Proof** We use the fact that the constructs used in the semantics of the core proof rules are continuous. We proceed by case analysis on the proof rule.

We just give an example. In the case of \( P_\otimes \), note that \( [P_\otimes(p,q)] = \bigsqcup [\{r\} : r \subseteq P_\otimes(p,q)] = \bigsqcup [\{p',q'\} : p' \subseteq p \land q' \subseteq q] = \bigsqcup \{\Gamma\}^{-1} \circ \text{dist}_{-\Gamma} \circ \{p',[q']\} : p' \subseteq p \land q' \subseteq q\} = \{\Gamma\}^{-1} \circ \text{dist}_{-\Gamma} \circ \{\{p' : p' \subseteq p\} \cup \{q' : q' \subseteq q\}\} = \bigsqcup \{\Gamma\}^{-1} \circ \text{dist}_{-\Gamma} \circ \{[p],[q]\} \) as required. All other cases are similar.

7.6.4. Totality

We need to show that given \( p \in I_\Gamma \), \( [p] \) is a total uniform strategy. Note that this is not true of arbitrary paraproofs in \( D_\Gamma \), nor is it true for infinite derivations in full WS1 (for example, one could repeatedly apply the \( P_\text{sym} \) rules forever).

To show this fact, we first introduce some auxiliary notions.
**Definition** Let \( \sigma : N \) be a strategy on a negative game. We say that \( \sigma \) is \( n \)-total if whenever \( s \in \sigma \land |s| \leq n \land s \in P_N \Rightarrow \exists p. sop \in \sigma \). A uniform strategy is \( n \)-total if it is pointwise \( n \)-total.

It is clear that a strategy is total if and only if it is \( n \)-total for each \( n \).

**Proposition 7.10.** The following hold:

1. If \( \sigma \) is \( n \)-total and \( \tau \) is an isomorphism then \( \tau \circ \sigma \) is \( n \)-total. If \( \sigma \) is \( n \)-total and \( \tau \) is an isomorphism then \( \sigma \circ \tau \) is \( n \)-total.
2. If \( \sigma : A \rightarrow B \) and \( \tau : A \rightarrow C \) are \( n \)-total then \( \langle \sigma, \tau \rangle \) is also \( n \)-total. If \( \sigma : A_i \rightarrow B \) is \( n \)-total then \( \sigma \circ \pi_i : A_1 \times A_2 \rightarrow B \) is \( n \)-total.
3. If \( \sigma : A \otimes B \rightarrow C \) is \( n \)-total then \( \Lambda(\sigma) \) is \( n \)-total. If \( \sigma : A \rightarrow B \) is \( n \)-total then \( \sigma \circ \text{id} : (B \rightarrow o) \rightarrow (A \rightarrow o) \) is \((n+2)\)-total.
4. If \( \sigma \) and \( \tau \) are \( n \)-total, then so is \([\sigma, \tau]_{\mathcal{C}, \mathcal{D}}, \sigma \circ \tau\). If \( \sigma \) is \( n \)-total, then \( n \) is \( \hat{\sigma} \).

**Proof** Simple verification. \( \square \)

**Proposition 7.11.** Given any infinitary analytic proof \( p \) of \( X; \Theta \vdash \Gamma \), \( [p] \) is total.

**Proof** We show that \( [p] \) is \( n \)-total for each \( n \). We proceed by induction on a compound measure.

- Define \( \text{tl}^+(A, \Gamma) \) to be the length of \( \Gamma \) as a list if \( A = \top \) or \( \infty \) otherwise.
- Define \( \text{hd}^+(A, \Gamma) \) to be \( |A| \) if \( A \) is positive or \( \infty \) otherwise.
- Define \( \text{tl}^-(A, \Gamma) \) to be the length of \( \Gamma \) as a list if \( A = \bot \) or \( \infty \) otherwise.
- Define \( \text{hd}^-(A, \Gamma) \) to be \( |A| \) if \( A \) is negative or \( \infty \) otherwise.

We proceed by induction on \( f(n, X, \Theta, \Gamma) = (n, \text{tl}^+(\Gamma), \text{hd}^+(\Gamma), \text{tl}^-(\Gamma), \text{hd}^-(\Gamma), L(X, \Theta)) \).

We proceed by case analysis on \( p \). If \( p = \mathcal{P}_\otimes(p_1, p_2) \) then \( [\mathcal{P}_\otimes(p_1, q_2)] = [\Gamma] \downarrow (\text{dec}^{-1}) \circ \text{dist}^{-1} \circ \langle [p_1], [p_2] \rangle \). By Proposition 7.10 \([\mathcal{P}_\otimes(p, q)] = [p] \) is \( n \)-total. The remaining cases work in an entirely analogous way. For \( \mathcal{P}^+ \) we must use the fact that currying is continuous and preserves \( n \)-totality. For termination:

- If \( \Theta \) is not lean, in the call to the inductive hypothesis the first five measures do not increase, and the fifth measure \( L \) decreases.
- In the case of \( \mathcal{P}_\otimes, \mathcal{P}_\wedge, \mathcal{P}_\lor \), the first three measures \( (n, \text{tl}^+(\Gamma), \text{hd}^+(\Gamma)) \) stay the same and either the fourth measure \( \text{tl}^-(\Gamma) \) decreases, or the fourth measure stays the same and the fifth measure \( \text{hd}^-(\Gamma) \) decreases.
- In the case of \( \mathcal{P}^\otimes, \mathcal{P}^\wedge, \mathcal{P}^\lor \), the first three measures stay the same and the fourth measure decreases.

56
In the cases of $P^+, P_\perp, P_\oplus, P?$ the first measure $n$ stays the same and either the second measure $t(\Gamma)$ decreases, or the second measure stays the same and the third measure $h(\Gamma)$ decreases.

In the case of $P^\otimes, P_\top, P^+\otimes$ the first measure stays the same and the second measure decreases.

In the case of $P^-, P_\top, P^+\top$ the first measure decreases.

In particular, $\lbrack P^+\rbrack(q) = \text{unit}_\otimes \circ (\lbrack q \rbrack \circ \text{id})$. By induction $\lbrack q \rbrack$ is $(n-2)$-total, and so $\lbrack q \rbrack \circ \text{id}$ is $n$-total, and so $\lbrack p \rbrack$ is $n$-total by Proposition 7.10.

Note that there are infinitary analytic proofs that denote strategies that are total, but not winning. For example, there is an infinitary analytic proof of $\vdash \bot$, $?(\top \triangleleft \bot)$ given by $P_\top(P_\top(P^+(h))))$. But there are no winning strategies on this game.

7.6.5. Reification of Total Strategies as Infinitary Analytic Proofs

We next show that any total strategy $\sigma$ on the denotation of a sequent is the interpretation of a unique infinitary analytic proof $\text{reify}(\sigma)$.

We first define $\text{reify}$ for winning strategies. We have seen that we can construct a family of maps $A_{X; \Theta \vdash \Gamma} \rightarrow I_{X; \Theta \vdash \Gamma}$ by giving, for each $X; \Theta \vdash \Gamma$ and $a \in A_{X; \Theta \vdash \Gamma}$, a proof rule that concludes $X; \Theta \vdash \Gamma$ from $X_1; \Theta_1 \vdash \Gamma_1, \ldots, X_n; \Theta_n \vdash \Gamma_n$ and for each $i$ an element $a_i \in A_{X_i; \Theta_i \vdash \Gamma_i}$.

$A_{\Gamma} \xrightarrow{f} (\text{Prf} \times \text{Seq}) \times (\sum_{X; \Theta \vdash \Gamma} A_{X; \Theta \vdash \Gamma})^*$

$\zeta f \zeta f \alpha \rightarrow (\text{Id} \times \zeta f \zeta f)^* \rightarrow (\text{Prf} \times \text{Seq}) \times I^*$

Note that our reification function $\text{reify}$ defined in Figure 14 is exactly of this shape. In this case $A_{X; \Theta \vdash \Gamma}$ is the set of uniform winning strategies on $[X; \Theta \vdash \Gamma]$. The function specifies, for each strategy, the root-level proof rule and the derived strategies that are given as input to $\text{reify}$ coinductively. In the case that $\sigma$ is bounded, we have seen that the process terminates and $\text{reify}(\sigma)$ is a finite proof.

In fact, we note that this family of maps are still well defined if $A_{X; \Theta \vdash \Gamma}$ is the set $\text{Tot}_{X; \Theta \vdash \Gamma}$ of uniform total strategies on $[X; \Theta \vdash \Gamma]$. In particular, the composition of a total strategy and an isomorphism is a total strategy; the composition of a total strategy and a projection is a total strategy; and the completeness axioms in Section 7.3 hold with respect to total strategies. This procedure provides, for each total strategy on $X; \Theta \vdash \Gamma$, a proof rule $P_x$ concluding $X; \Theta \vdash \Gamma$ from $X_1; \Theta_1 \vdash \Gamma_1, \ldots, X_n; \Theta_n \vdash \Gamma_n$ and
total strategies on each \([X; \Theta \vdash \Gamma]\). We write this map as \(\text{reif}_{X; \Theta \vdash \Gamma}\).

\[
\sum_{X; \Theta \vdash \Gamma \in \text{Seq}} \text{Tot}_{X; \Theta \vdash \Gamma} \xrightarrow{\text{reif}} (\text{Prf} \times \text{Seq}) \times (\sum_{X; \Theta \vdash \Gamma \in \text{Seq}} \text{Tot}_{X; \Theta \vdash \Gamma})^*
\]

Thus we can take the anamorphism of this map yielding a map from total strategies on \([X; \Theta \vdash \Gamma]\) to \(I_{X; \Theta \vdash \Gamma}\), as required.

### 7.6.6. Soundness and Uniqueness

We can show that given any winning strategy \(\sigma\), \(\text{reify}(\sigma)\) is the unique infinitary analytic proof \(p\) such that \(\text{reify}(p) = \sigma\).

For soundness, we first introduce some auxiliary notions.

**Definition** Let \(\sigma\) and \(\tau\) be strategies on \(A\). We say that \(\sigma =_n \tau\) if each play in \(\sigma\) of length at most \(n\) is in \(\tau\), and each play in \(\tau\) of length at most \(n\) is in \(\sigma\).

It is clear that \(=_n\) is an equivalence relation, and \(\sigma = \tau\) if and only if \(\sigma =_n \tau\) for each \(n \in \mathbb{N}\). We can lift the relation \(=_n\) to uniform total strategies pointwise.

**Proposition 7.12.**

1. If \(\sigma =_n \tau\) and \(\rho\) is an isomorphism then \(\sigma \circ \rho =_n \tau \circ \rho\).

2. If \(\sigma =_n \tau\) and \(\rho =_n \delta\) then \(\sigma, \rho =_n (\tau, \delta)\). If \(\sigma =_n \tau\) then \(\sigma \circ \pi_i =_n \tau \circ \pi_i\).

3. If \(\sigma =_n \tau\) then \(\Lambda(\sigma) =_n \Lambda(\tau)\). If \(\sigma =_n \tau\) then \(\sigma \circ \text{id} =_n \tau \circ \text{id}\).

4. If \(\sigma_1 =_n \sigma_2\) and \(\tau_1 =_n \tau_2\) then \([\sigma_1, \tau_1]_{C, D} = [\sigma_2, \tau_2]_{C, D}\). If \(\sigma_1 =_n \sigma_2\) then \(\sigma_1 \circ H =_n \sigma_2 \circ H\). If \(\sigma_1 =_n \sigma_2\) then \(\hat{\sigma}_1 =_n \hat{\sigma}_2\).

**Proof** Simple verification.

**Proposition 7.13.** For every uniform total strategy \(\sigma: [\vdash \Gamma]\), \(\text{reify}(\sigma) = \sigma\).

**Proof** We show that for each \(n\), \([\text{reify}(\sigma)] =_n \sigma\). The structure of the induction follows that of Proposition [7.11] lexicographically on

\([n, \text{tl}^+(\Gamma), \text{hd}^+(\Gamma), \text{tl}^-(\Gamma), \text{hd}^-(\Gamma), L(X, \Theta)]\).

In each particular case, the reasoning follows the proof of Proposition [7.6] using \(=_n\) in the inductive hypothesis rather than \(=\), and propagating this to the main equation using Proposition [7.12]. In the case of \(\Gamma = \top, N\) we use the inductive hypothesis with a smaller \(n\), using the final clause in Proposition [7.12].

**Proposition 7.14.** Given any infinitary analytic proof \(p\), \(\text{reify}(\lfloor p \rfloor) = p\).
Proof Since id = α ⊆, we know that id is the unique morphism $f$ such that:

$$
\begin{array}{ccc}
\mathcal{I}_\Gamma & \xrightarrow{\alpha} & (\text{Prf} \times \text{Seq}) \times \mathcal{I}^* \\
\downarrow f & & \downarrow \text{id} \times f^* \\
\mathcal{I}_\Gamma & \xrightarrow{\alpha} & (\text{Prf} \times \text{Seq}) \times \mathcal{I}^*
\end{array}
$$

Thus to show that $\text{reify} \circ \preceq = \text{id}$ it is sufficient to show that $\alpha \circ \text{reify} \circ \preceq = \text{id} \times (\text{reify} \circ \preceq)^* \circ \alpha$, i.e. that for each infinitary analytic proof $p$ we have $\alpha(\text{reify}(\llbracket p \rrbracket)) = (\text{id} \times (\text{reify} \circ \preceq)^*) (\alpha(p))$.

- For binary rules $P_x$ we must show that
  $$
  \text{reify}(\llbracket P_x(p_1, p_2) \rrbracket) = P_x(\text{reify}(\llbracket p_1 \rrbracket), \text{reify}(\llbracket p_2 \rrbracket)).
  $$

- For unary rules $P_x$ we must show that $\text{reify}(\llbracket P_x(p) \rrbracket) = P_x(\text{reify}(\llbracket p \rrbracket))$.

- For nullary rules $P_x$ we must show that $\text{reify}(\llbracket P_x \rrbracket) = P_x$.

For each proof rule, we have already shown this in the proof of Proposition 7.7. Proposition 7.9 ensures that the proof applies in this setting.

7.6.7. Full Completeness and Normalisation

We have thus shown:

**Theorem 7.15.** Each total strategy $\sigma$ on $\vdash \Gamma$ is the denotation of a unique infinitary analytic proof $\text{reify}(\sigma)$.

We hence have a bijection between infinitary analytic proofs of a formula, and total strategies on the denotation of that formula, via the semantics. Since any proof in WS1 can be given semantics as a winning strategy, and winning strategies are total, we may reify the semantics of a WS1 proof to generate its infinitary normal form $\text{reify}(\llbracket p \rrbracket)$.

**Theorem 7.16.** For each WS1 proof $p$, there is a unique infinitary analytic proof $q$ such that $\llbracket p \rrbracket = \llbracket q \rrbracket$.

**Proof** Let $q = \text{reify}(\llbracket p \rrbracket)$. Then $\llbracket q \rrbracket = \text{reify}(\llbracket p \rrbracket) = \llbracket p \rrbracket$ by Proposition 7.13. If $q'$ is an infinitary analytic proof with $\llbracket q' \rrbracket = \llbracket p \rrbracket$ then $\llbracket q' \rrbracket = \llbracket q \rrbracket$ and so $\text{reify}(\llbracket q' \rrbracket) = \text{reify}(\llbracket q \rrbracket)$ and Proposition 7.14 ensures that $q' = q$. □

While infinitary analytic proofs may denote strategies that are not winning, any infinitary analytic proof generated as a result of the above normalisation denotes a winning strategy. The above result also ensures that proofs $p_1$ and $p_2$ in WS1 denote the same strategy if and only if their normal forms (as infinitary analytic proofs) are identical.
8. 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. [30]). 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 [10])?

- Extension with propositional variables (and potentially, second-order quantification) would allow generic “copycat 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 $\mu LJ$ [8] would extend WS1 to a rich collection of datatypes (including finite and infinite lists, for example).

Acknowledgements. The authors would like to thank Pierre-Louis Curien, Alessio Guglielmi, Pierre Clairambault and anonymous reviewers for earlier comments on this work. This work was supported by the (UK) EPSRC grant EP/HO23097.

References


61