Game semantics of Martin-Löf type theory

Norihiro Yamada

This is a preprint submitted to Research in the Mathematical Sciences (RMS).

Abstract

We present new game semantics of Martin-Löf type theory (MLTT) equipped with One-, Zero-, N-, Pi-, Sigma- and Id-types. Our game semantics interprets MLTT more accurately than existing ones. Another advantage of our game semantics over existing ones is its interpretation of Sigma-types that is direct and compatible with the game semantics of product types. Besides, its mathematical structure is novel and useful; e.g., the category of our games has all finite limits, which is a key step to an extension of the present work to homotopy type theory, and our games interpret subtyping on dependent types for the first time as game semantics. Finally, we provide a new, game-semantic proof of the independence of Markov’s principle from MLTT, which demonstrates an advantage of our game semantics over extensional models of MLTT such as the effective topos.

Keywords: game semantics; Martin-Löf type theory; constructive mathematics

Contents

Abstract 1

1 Introduction 2

1.1 Martin-Löf type theory and the meaning explanation 2

1.2 Game semantics 2

1.3 Main results 3

1.4 Related work and our contributions 4

1.5 Concluding remarks 5

1.6 The structure of the present article 5

2 Games and strategies for simple type theories 6

2.1 Arenas and legal positions 6

2.2 Games and strategies 9

2.3 Constructions on games and strategies 11

3 Predicate games 13

3.1 Predicate games 15

3.2 Cartesian closed categories of predicate games 17

4 Game semantics of Martin-Löf type theory 21

4.1 Dependent predicate games 21

4.2 Pi on dependent predicate games 22

4.3 Sigma on dependent predicate games 23
1 Introduction

1.1 Martin-Löf type theory and the meaning explanation

Martin-Löf type theory (MLTT) [1, 2] is one of the best-known formal systems for constructive mathematics [3], which is comparable to set theory [4, 5] for classical mathematics. MLTT is also a programming language [6] that is a generalisation of the simply-typed lambda-calculus (STLC) [7] along the generalisation of (intuitionistic) propositional logic to predicate logic under the Curry-Howard isomorphisms [8]. By this computational nature, MLTT and similar formal systems enable computer formalisations of mathematics and its applications to programming [9, 10].

Like set theory is explained informally by sets, the conceptual foundation of MLTT is computations in an informal sense. That is, the fundamental idea of MLTT is to regard objects and proofs in constructive mathematics uniformly as computations, and MLTT is a syntactic formalisation of this foundational idea [6]. Hence, objects and proofs in MLTT are unified into terms, where formulas are called types. This standard, informal semantics of MLTT is called the meaning explanation [11, §5].

However, MLTT is not always the best formalisation of this conceptual foundation of constructive mathematics since it is an intricate formal system that inevitably contains superficial syntactic details.[1] In other words, the intuition behind MLTT is often blurred by the complexity and the syntactic nature of MLTT. In addition, the syntactic complexity makes it difficult to study the meta-theory of MLTT.

Accordingly, mathematical semantics [12] of MLTT that faithfully formalises the meaning explanation is strongly desired since such semantics would accurately and directly (or non-inductively) describe the intuition behind MLTT, abstracting the syntactic details. It would not only deepen our understanding of MLTT in this way but also suggest improvements and extensions of MLTT like coherence spaces by Girard [13] led to linear logic [13], and the groupoid model by Hofmann and Streicher [14] to homotopy type theory (HoTT) [10]. Besides, mathematical semantics has been highly effective for the meta-theoretic study of MLTT; e.g., see [15].

1.2 Game semantics

Game semantics [16, 17] is a particular class of mathematical semantics of logic and computation that models types and terms by games and strategies, respectively.

[1]A syntactic formalisation is also unsatisfactory from the syntax-first-view, i.e., the view that semantic concepts come first, and syntax merely provides notations.
Its strong point is its conceptual naturality: ‘Logic is the study of reasoning’ [18, p. 1], where one can regard ‘reasoning’ as dialogical arguments between Player (or a mathematician) and Opponent (or an oracle), and game semantics formalises this intuition. This game-semantic view on logic is also in harmony with the meaning explanation since dialogical arguments are a certain kind of computations.

Another strong advantage of game semantics is its precision in modelling syntax as various full completeness/abstraction results [19] in the literature demonstrate. This precision is due to its intensionality: Game semantics captures the processes themselves underlying terms rather than their extensions such as functions.

Finally, the concrete nature of game semantics enables its algorithmic applications to program analysis and verification (i.e., meta-theories of programs) [20].

1.3 Main results
To summarise the points so far, mathematical semantics of MLTT that advances our understanding of MLTT, promotes its improvements and extensions, and/or clarifies its meta-theory is strongly desired, and game semantics seems perfect for this role by its conceptual naturality, harmony with the meaning explanation, precision in modelling syntax and algorithmic applications. Also, game semantics models effects [21] and linear logic [13] in a highly systematic way (see [22] for the details); thus, game semantics of MLTT, if any, may lead to MLTT with effects and linear typing.

However, although game semantics of various logics and computations has been given, it is quite difficult to establish game semantics of MLTT. The main challenge in game semantics of MLTT is to model the extensional type dependency in Sigma-types by intensional processes in games; see the beginning of §3. In fact, this problem had been open for more than twenty years, and even today its definitive solution is yet to emerge though a few candidates have arisen recently [23, 24]; see §1.4.

Hence, we aim to provide another candidate for game semantics of MLTT with the hope that it sheds new light on this problem. Motivated in this way, we prove:

**Theorem** (Game semantics of MLTT) There is new game semantics of MLTT with One-, Zero-, N-, Pi-, Sigma- and Id-types (§4.4–4.5).

Our key idea for this theorem is to generalise games so that they can model Sigma-types, while we keep strategies unchanged so that we retain the advantages of game semantics such as intensionality (§1.2). See the beginning of §3 for the outline of this idea. Also, see §1.4 for the advantages of this method over existing ones [23, 24].

We also illustrate the utility of our game semantics by giving a new proof of:

**Corollary** (Independence of Markov’s principle [25]) Markov’s principle is independent from MLTT equipped with the types listed in Theorem 1.3 (§4.7).

The method used by Mannaa and Coquand [25] is syntactic, while our semantic approach provides a new, intuitive argument on why the independence holds. This corollary also illustrates an advantage of game semantics over other computational models since, e.g., the effective topos [27] cannot show the independence. Moreover, by the non-inductive nature of game semantics, the present method would be easily applied to the independence of Markov’s principle from various extensions of MLTT.

Finally, the novel mathematical structure of our game semantics enables:
Corollary (Game semantics of subtyping on dependent types) The game semantics given by Theorem 1.3 models subtyping [28, §15] on dependent types (§4.8).

1.4 Related work and our contributions

Abramsky et al. have established the first game semantics of MLTT equipped with One-, Pi-, Sigma-, Id- and finite inductive types [23, 29]. Its significance is that it is the first intensional model of MLTT, and thus it stands in sharp contrast with other computational models such as realizability and domain models [15, 30], which are extensional. Their main result is a certain kind of full completeness. However, they only interpret Sigma-types indirectly and inductively by a list construction and the adjunction between Pi- and Sigma-types. Specifically, they interpret a Sigma-type $\vdash \Sigma_x C \Downarrow D(x)$ type [2] by the list $(C, D)$ of the game $C$ that models the simple type $\vdash C$ type and the family $D = \langle D(x) \rangle_{x : C}$ of games $D(x)$ indexed by strategies $x$ on $C$ that models the dependent type $x : C \vdash D(x)$ type, and interpret a term of the form $\vdash (c, d) : \Sigma x : C D(x)$ by the list $(c, d)$ of the strategies $c$ and $d$ that respectively model the terms $\vdash c : C$ and $\vdash d : D(c)$, and a term of the form $z : \Sigma x : C D(x) \vdash e(z) : E(z)$ by identifying it with the one $x : C, y : D(x) \vdash e((x, y)) : E((x, y))$. As a result, they interpret types and terms by the lists of (families of) games and strategies, respectively; this method merely simulates the term model [31, §2.4] and drops the non-inductive nature of game semantics. Another undesirable feature of this approach is that it identifies the types $\Pi x \Sigma z A(x) C(x)$ and $\Pi x A \Pi y B \Sigma (x, y)$. Finally, for composing winning strategies, they use the O-sat operation [29, Remark 4.5]; however, it generates a significant gap between MLTT and their model. In fact, this method only works for a very specific class of finite inductive types [29, Figure 7]; see §4.3 on this point.

Another related work is the denotational model of MLTT [24] by Blot and Laird based on concrete date structures and sequential algorithms [32]. They interpret Boolean-, Pi- and Sigma-types and a universe though their interpretation of the universe is not by game semantics but by domain theory. Their main results are certain full completeness/abstraction. Notably, they directly interpret Sigma-types without the list construction, overcoming the problem of the preceding work. However, it is possible to play on both sides of their interpretation of a Sigma-type within a single play, which is far from the game semantics [33, 34, 35] (and even the graph game semantics [36]) of product types. Thus, it is arguable if their model properly captures the generalisation of product types to Sigma-types. As a more pragmatic disadvantage, their model admits control operators or classical reasonings, but the logical part of MLTT is intuitionistic; i.e., there is a gap between MLTT and their model. Besides, their model does not achieve the linear decomposition of function types [13] or the characterisation of effects by constraints on strategies, which are both strong advantages of game semantics [22]. Last but not least, their interpretation of Id-types by finite tuples of Boolean-type sketched in [24, §9] does not work in the presence of N-type since the set $\mathbb{N}$ of all natural numbers is unbounded.

Thus, each of the existing approaches to game semantics of MLTT has pros and cons, and we have not reached a consensus on which option should be a definitive solution. In this context, we offer the third method with the novel features listed below, hoping that it would eventually lead to a definitive solution in the future.

\[\text{For simplicity, here we only consider the empty context.}\]
First, we achieve the first game semantics of both N- and Id-types. The interpretation of these types enables us to show the independence of Markov’s principle.

Second, our games are a modest generalisation of a standard variant, McCusker’s games [35], and we model types and terms by such games and (ordinary) strategies, not lists of them, respectively. As a result, we retain the syntax-independence and the non-inductive nature of game semantics. Besides, we directly model Sigma-types by our games, where a play occurs only in either side. Hence, we overcome the main shortcomings of the preceding methods. For the basic idea, see the beginning of §3.

Third, our method inherits the linear decomposition of function types and the characterisation of effects by constraints on strategies in McCusker’s one [22].

Fourth, the novel mathematical structure of our games enables us to dispense with the O-sat operation unlike Abramsky et al. so that we can interpret the type dependency of a standard class of Pi- and Sigma-types more accurately; see §4.3.

Finally, the mathematical structure of our games is novel and useful. For instance, the category of our games has all finite limits (Corollary 3.14), while that of existing games does not. This novel structure enables us to internalise a certain notion of ∞-groupoids in the category of our games, which is a key step to extend the present work to HoTT [37]. Moreover, we accomplish the first game semantics of subtyping on dependent types (§4.8). On the other hand, in order to focus on the main idea of the present work, we leave it to another article to interpret universes.

1.5 Concluding remarks
Due to the novel mathematical structure of our games, one might misunderstand that our semantics is close to extensional models such as realizability and domain models. However, since our strategies are just the ordinary ones (§1.4), our method inherits the advantages of standard game semantics such as intensionality (§1.2). In fact, the intensional features of the preceding game semantics [23, §1] are mostly valid in ours (§4.6); e.g., both refute function extensionality. Also, our model refutes Markov’s principle by its intensionality (§4.7), while the effective topos does not.

Last but not least, we do not prove full completeness for the following reasons. First, the standard syntax of MLTT, specifically N-type, is not very suited to fully complete game semantics. For instance, the full completeness results [23, 29, 24] are on modifications of MLTT, and they exclude N-type. However, our main topic is MLTT itself (§1.1), and thus we leave full completeness on a modification of MLTT as future work. Second, our priority is more on an interpretation of N-type than full completeness without N-type since our motivation comes from foundations of mathematics (§1.1), for which N-type is crucial. Finally, one of our aims is to provide tools for the study of MLTT (§1.1), but full completeness is not necessarily the most important result for this aim. For example, the fully complete model [24] cannot show the independence of Markov’s principle since it admits classical reasonings. Hence, we instead show the utility of our model by proving the independence (§4.7).

1.6 The structure of the present article
The rest of this article proceeds as follows. We first recall McCusker’s games and strategies in §2 and generalise the games in §3. We then interpret MLTT by the generalised games and strategies in §4, where we analyse the intensionality of our game semantics in §4.6, give a game-semantic proof of the independence of Markov’s principle from MLTT in §4.7, and interpret subtyping on dependent types in §4.8.
2 Games and strategies for simple type theories

We first recall McCusker’s games and strategies for simple type theories [35], which the present work is based on. We select this variant for the following reasons. First, it combines the strong points of the two best-known variants: the linear decomposition of function types [13] achieved by AJM-games [33] and the characterisation of effects by constraints on strategies [22] that utilises pointers in HO-games [34] (originally introduced in [38]). Our games inherit these advantages so that they would shed new light on the problem of combining MLTT and linear logic and/or effects. Second, pointers enable us to refine game semantics into a model of computation [39], which is highly desirable as a mathematical foundation of constructive mathematics (§1.1).

We assume that the reader is familiar with McCusker’s games and strategies, and leave more expositions and examples to the gentle introduction [22]. We henceforth call McCusker’s games and strategies respectively games and strategies.

We first recall two preliminary concepts in §2.1, and then games and strategies in §2.2. We finally recall standard constructions on games and strategies in §2.3.

Notation We use the following notations throughout the present article:

- We use bold small letters \( s, t, u, v \), etc. for sequences, in particular \( e \) for the empty sequence, and small letters \( a, b, m, n, x, y \), etc. for elements of sequences;
- We define \( \mathbb{N} := \{ 1, 2, \ldots, n \} \) for each \( n \in \mathbb{N}^+ := \mathbb{N} \setminus \{ 0 \} \), and \( \emptyset := \emptyset \);
- We write \( x_1 x_2 \ldots x_{|s|} \) for \( s = (x_1, x_2, \ldots, x_{|s|}) \), where \( |s| \) is the length of \( s \), define \( s(i) := x_i \ (i \in |s|) \) and write \( a \in s \) if \( a = s(j) \) for some \( j \in |s| \);
- A concatenation of sequences \( s \) and \( t \) is represented by their juxtaposition \( st \) (or \( s t \)), but we often write \( a s, tb, u cv \) for \( (a)s, t(b), u(c)v \), and so on;
- We write Even(\( s \)) (resp. Odd(\( s \))) if \( s \) is of even- (resp. odd-) length, and given a set \( S \) of sequences and \( P \in \{ \text{Even}, \text{Odd} \} \), we define \( Sp := \{ s \in S \mid P(s) \} \);
- We write \( s \preceq t \) if \( s \) is a prefix of \( t \), and given a set \( S \) of sequences, Pref(\( S \)) for the set of all prefixes of sequences in \( S \), i.e., Pref(\( S \)) := \{ \( s \mid \exists t \in S. \ s \preceq t \} \);
- We use informal ‘tags’ (\( \_i \)) (\( i \in \mathbb{N} \)) for clarity (e.g., see Definition 2.14).

2.1 Arenas and legal positions

A game is a certain kind of a rooted directed acyclic graph, whose paths from a root represent possible developments or positions in a ‘game in the ordinary sense’ (e.g., chess). These positions are finite sequences of vertices or moves, and a play in the game proceeds as its participants alternately perform moves along a position. It is conventional to identify each game with the set of all positions in the game. We focus on standard two-person games between Player (\( P \)) (or a mathematician) and Opponent (\( O \)) (or an oracle), in which \( O \) always starts a play.

Technically, games are based on two preliminary concepts: arenas and legal positions. An arena defines the basic components of a game, which in turn induces legal positions of the arena that specify the basic rules of the game in the sense that each position of the game must be legal. Let us first recall these two concepts.

Definition 2.1 (Moves) Let us fix, throughout the present work, arbitrary pairwise distinct symbols \( O, P, Q \) and \( A \), and call them labels. A move is any triple \( m^{xy} := (m, x, y) \) such that \( x \in \{ O, P \} \) and \( y \in \{ Q, A \} \). We usually abbreviate moves \( m^{xy} \) as \( m \), and instead define \( \lambda(m) := xy, \lambda_{OP}(m) := x \) and \( \lambda_{QA}(m) := y \).
We call a move \( m \) an **O-move** if \( \lambda^{\text{OP}}(m) = \text{O} \), a **P-move** if \( \lambda^{\text{OP}}(m) = \text{P} \), a **question** if \( \lambda^{\text{QA}}(m) = \text{Q} \), and an **answer** if \( \lambda^{\text{QA}}(m) = \text{A} \).

**Definition 2.2** (Arenas [34, 35]) An arena is a pair \( G = (M_G, \vdash_G) \) such that
- \( M_G \) is a set of moves;
- \( \vdash_G \) is a subset of the cartesian product \((\star) \cup M_G \times M_G\), where \( \star \) (or represented more precisely by \( \star_G \)) is an arbitrarily fixed element such that \( \star \not\in M_G \), called the **enabling relation**, that satisfies
  - (E1) If \( \star \vdash_G m \), then \( \lambda(m) = \text{OQ} \);
  - (E2) If \( m \vdash_G n \) and \( \lambda^{\text{QA}}(n) = \text{A} \), then \( \lambda^{\text{QA}}(m) = \text{Q} \);
  - (E3) If \( m \vdash_G n \) and \( m \neq \star \), then \( \lambda^{\text{OP}}(m) \neq \lambda^{\text{OP}}(n) \).

We call moves \( m \in M_G \) **initial** if \( \star \vdash_G m \), and set \( M_G^{\text{Init}} := \{ m \in M_G \mid \star \vdash_G m \} \).

An arena \( G \) is **well-founded (w.f.)** if \( \vdash_G \) is well-founded, i.e., there is no sequence \( (m_i)_{i \in \mathbb{N}} \) of moves \( m_i \in M_G \) such that \( \star \vdash_G m_0 \) and \( m_i \vdash_G m_{i+1} \) for all \( i \in \mathbb{N} \).

**Remark** In the original article [35], an arena is a triple \( G = (M_G, \lambda_G, \vdash_G) \), where labels are assigned to moves by the **labelling function** \( \lambda_G : M_G \rightarrow \{\text{O, P}\} \times \{\text{Q, A}\} \). Instead, we embed labels into moves (Definition 2.1); this modification is convenient, e.g., when we take unions of games, since it makes labels on moves unambiguous without underlying arenas. Besides, the axiom E1 in [35] further requires \( n \vdash_G m \Leftrightarrow n = \star \) whenever \( \star \vdash_G m \). We discard this condition too again for unions of games.

An arena \( G \) specifies moves in a game, each of which is O’s/P’s question/answer, and which move \( n \) can be performed for each move \( m \) during a play in the game by the relation \( m \vdash_G n \) (cf. Definition 2.3), where \( \star \vdash_G m \) means that O can initiate a play by \( m \) in the game. The axioms E1, E2 and E3 are then to be read as follows:
- E1 sets the convention that an initial move must be O’s question;
- E2 states that an answer must be performed for a question;
- E3 says that an O-move must be performed for a P-move, and vice versa.

We next review legal positions, a certain class of finite sequences of moves equipped with **pointers** from later to earlier occurrences in the sequences. The idea is that each non-initial occurrence in a legal position must be made for a specific previous occurrence, and pointers specify such pairs of occurrences. Technically, pointers enable us to distinguish similar yet different plays [22, §2.4] and define **views** (Definition 2.4). Views play crucial roles when we define constraints on strategies (Definition 2.9).

We call a finite sequence of moves together with a pointer a **justified (j-) sequence**. A legal position is then a particular kind of a j-sequence.

**Definition 2.3** (Justified sequences [38, 35]) An **occurrence** in a finite sequence \( s \) is a pair \((s(i), i)\) such that \( i \in \overline{s} \). A **justified (j-) sequence** is a pair \( s = (s, \mathcal{J}_s) \) of a finite sequence \( s \) of moves and a map \( \mathcal{J}_s : \overline{s} \rightarrow \{0\} \cup \{0, \overline{s} - 1\} \) such that \( 0 \leq \mathcal{J}_s(i) < i \) for all \( i \in \overline{s} \), called the **pointer** of the j-sequence. An occurrence \((s(i), i)\) is **initial** in \( s \) if \( \mathcal{J}_s(i) = 0 \). We say that the occurrence \((s(\mathcal{J}_s(i)), \mathcal{J}_s(i))\) is the **justifier** of a non-initial one \((s(i), i)\) in \( s \), and \((s(i), i)\) is **justified** by \((s(\mathcal{J}_s(i)), \mathcal{J}_s(i))\) in \( s \).

A j-sequence is **in an arena** \( G \) if its elements are moves in \( G \), and its pointer respects the enabling relation \( \vdash_G \) in \( G \), i.e., \( \forall i \in \overline{s} \). \((s(\mathcal{J}_s(i)), \mathcal{J}_s(i)) \Rightarrow \left( \mathcal{J}_s(i) \neq 0 \Rightarrow s(\mathcal{J}_s(i)) \vdash_G s(i) \right) \land \left( \mathcal{J}_s(i) \neq 0 \Rightarrow s(\mathcal{J}_s(i)) \vdash_G s(i) \right) \). We write \( \mathcal{J}_G \) for the set of all j-sequences in \( G \).
A justified \((j\text{-})\) subsequence of a \(j\text{-sequence} \, s\) is a \(j\text{-sequence} \, t\), written \(t \sqsubseteq s\), such that \(t\) is a subsequence of \(s\), and \(J_s(i) = j\) if and only if \(J_s^+(i) = j\) for some \(n \in \mathbb{N}^+\) with the occurrences \((s(J_s^+(i)), J_s^+(i))\) for \(k = 1, 2, \ldots, n - 1\) deleted in \(t\).

**Remark** Unlike the original formulation [35], we define \(j\text{-sequences}\) in such a way that they make sense without underlying arenas. This reformulation is convenient as it enables us to manipulate \(j\text{-sequences}\) without calculating underlying arenas.

**Convention** Henceforth, we are casual about the distinction between moves and occurrences, and by abuse of notation, we frequently keep the pointer \(J_s\) of each \(j\text{-sequence} \, s = (s, J_s)\) implicit since it is mostly obvious, and abbreviate occurrences \((s(i), i)\) in \(s\) as \(s(i)\). Besides, we sometimes write \(J_s(s(i)) = s(j)\) if \(J_s(i) = j > 0\).

**Definition 2.4** (Views [38, 34, 35]) The **P-view** \([s]\) and the **O-view** \([s]\) of a \(j\text{-sequence} \, s\) are the \(j\)-subsequences of \(s\) defined by the following induction:

- \([\epsilon] := \epsilon;\)
- \([sm] := [s].m\) if \(m\) is a P-move;
- \([sm] := m\) if \(m\) is initial;
- \([smn] := [s].mn\) if \(n\) is an O-move such that \(m\) justifies \(n;\)
- \([\epsilon] := \epsilon;\)
- \([sm] := [s].m\) if \(m\) is an O-move;
- \([smn] := [s].mn\) if \(n\) is a P-move such that \(m\) justifies \(n.;\)

The idea on views is as follows. Given a nonempty \(j\text{-sequence} \, sm\) such that \(m\) is a P- \((\text{resp. O-})\) move, the P-view \([s]\) \((\text{resp. O-view} \, [s])\) is the currently ‘relevant part’ of the previous occurrences in \(s\) for P \((\text{resp. O}).\) I.e., P \((\text{resp. O})\) is concerned only with the last occurrence of an O- \((\text{resp. P-})\) move, its justifier and that justifier’s P- \((\text{resp. O-})\) view, which then recursively proceeds. See [38] for an explanation of views in terms of their counterparts in logical calculi, and [40] in lambda-calculi.

**Definition 2.5** (Legal positions [22]) A **legal position** is a \(j\text{-sequence} \, s\) such that

- (Alternation) If \(s = s_1mn_2s_2\), then \(\lambda^{op}(m) \neq \lambda^{op}(n);\)
- (Visibility) If \(s = tmu\) with \(m\) non-initial, then \(J_s(m)\) occurs in the P-view \([t]\) if \(m\) is a P-move, and in the O-view \([t]\) otherwise.

A legal position is **in an arena** \(G\) if it is a \(j\text{-sequence} \, s\) in \(G\) (Definition 2.3). We write \(\mathcal{L}_G\) for the set of all legal positions in \(G\).

As already noted, legal positions in an arena are to specify the basic rules of a game in the sense that positions in the game must be legal (Definition 2.6) so that

- During a play in the game, O makes the first move by a question (by E1), and then P and O alternately make moves (by alternation), where each non-initial move is made for a specific previous occurrence, viz., its justifier;
- The justifiers of non-initial occurrences are in the ‘relevant part’ (by visibility).

[3] Since the initial element \(s(1)\) of a legal position \(s\) in an arena \(G\) is subject to the equation \(J_s(1) = 0\), we have \(s(1) = L(G, s, 0)\). Hence, the axiom E1 on \(G\) implies \(\lambda(s(1)) = \text{OQ}\).

[4] Again, since we focus on a legal position \(s\) in an arena \(G\), the justifier of each P-move occurring in \(s\) is an O-move, and vice versa, by the axiom E3 on \(G\). In addition, the justifier of each answer occurring in \(s\) is a question by the axiom E2 on \(G\).
2.2 Games and strategies
We are now ready to recall the central concepts of games and strategies. For technical convenience, we slightly modify the original definition of games in [35, 22]:

Definition 2.6 (Games [35, 22]) A game is a set $G$ of legal positions such that
1. $G$ is nonempty and prefix-closed (i.e., $sm \in G \Rightarrow s \in G$);
2. $\text{Arn}(G) := (M_G, \rightarrow_G)$ is an arena, where $M_G := \{ s(i) \mid s \in G, i \in \overline{s} \}$ and $\rightarrow_G := \{ (s(j)) \mid s \in G, \mathcal{J}_s(j) = 0 \} \cup \{ (s(i), s(j)) \mid s \in S, \mathcal{J}_s(j) = i > 0 \}$.

It is well-founded (w.f.) if so is $\text{Arn}(G)$, and well-opened (w.o.) if each of its elements has at most one initial occurrence (i.e., the conjunction of $sm \in G$ and $m \in M_G^{\text{init}}$ implies $s = e$). We call elements of $G$ (valid) positions in $G$.

A subgame of $G$ is a game $H \subseteq G$, and $\text{sub}(G) := \{ H \mid H$ is a subgame of $G \}$.

Remark The original article [35, p. 27] also imposes thread-closure on each game $G$: The thread $s \mid I \subseteq s$ of a position $s \in G$ with respect to a given set $I$ of initial occurrences in $s$, which consists of occurrences hereditarily justified\(^5\) by elements in $I$, must be in $G$. This axiom is to ensure that positions in $G$ are in the exponential $!G$ (Definition 2.12), i.e., $G \subseteq !G$, which matches the intuition on exponential $!$ [13].

However, $!G$ is well-defined even if $G$ is not thread-closed. Also, we later focus on w.o. games (for the identities in the categories of games to be well-defined), which are trivially thread-closed. For these reasons, we omit thread-closure in Definition 2.6.

Each game $G$ is nonempty and prefix-closed because conceptually each nonempty position or ‘moment’ in $G$ must have the previous ‘moment.’ Note that positions in $G$ are automatically legal in the arena $\text{Arn}(G)$; i.e., as noted before, the legality is the basic or minimal requirement on positions in games. We later focus on w.o., w.f. games because the identities in the categories of such games behave well (§2.3).

The tuple $\mathcal{M}(G) := (M_G, \lambda_G, \rightarrow_G, G)$, where $\lambda_G := m \mapsto \lambda(m)$, forms a game in the sense of [35]\(^6\) whose labels are embedded into moves, which we call an MC-game.

The MC-game $\mathcal{M}(G)$ satisfies: Each move $m \in M_G$ occurs in some position in $G$, and each pair $s \rightarrow_G m$ or $m_1 \rightarrow_G m_2$ is used in some position in $G$. Conversely, given an MC-game $H$ that satisfies these two conditions, for which we call $H$ economical, the set of all positions in $H$ forms a game. Besides, these constructions are inverses to each other. Hence, $\mathcal{M}$ is a bijection between games and economical MC-games.

Since the economical axioms only exclude unused structures, our simplification of MC-games into games is harmless. We may further dispense with arenas by directly axiomatising the effects of arenas on legal positions (Footnotes 3–4), but we do not since arenas are convenient for defining constructions on games (Definition 2.12).

Definition 2.7 (Strategies [35, 22]) A strategy on a game $G$ is a subset $\sigma \subseteq G^{\text{Even}}$, written $\sigma : G$, that is nonempty, even-prefix-closed (i.e., $smn \in \sigma \Rightarrow s \in \sigma$) and deterministic (i.e., $smn, smn' \in \sigma \Rightarrow smn = smn'$). Let $\text{st}(G) := \{ \sigma \mid \sigma : G \}$.

\(^5\)An occurrence $n$ in a j-sequence $s$ is hereditarily justified by another occurrence $m$ in $s$ if $\mathcal{J}_s(n) = m$ for some $i \in \mathbb{N}^+$ [35, p. 22].

\(^6\)Except that the axiom $E1$ is slightly weakened, and the thread-closure is omitted; see the remarks after Definitions 2.2 and 2.6, respectively.
We define the closure of a strategy \( \sigma : G \) with respect to another game \( H \) to be the subgame \( \overline{\sigma}_H := \{ \epsilon \} \cup \{ sm \in H^{\text{Odd}} \mid s \in \overline{\sigma}_H \} \cup \{ tlr \in \sigma \mid tl \in \overline{\sigma}_H \} \subseteq \sigma \cup H. \)

The idea is that a strategy \( \sigma : G \) describes for \( P \) how to play on the game \( G \) by the computation \( sm \in G^{\text{Odd}} \Rightarrow smn \in \sigma \), which is deterministic by the determinacy of \( \sigma \), and in general partial since there can be no \( smn \in \sigma \) for some \( sm \in G^{\text{Odd}} \).

We use the closure operation in §3. We leave it to the reader to verify by induction that the equation \( \overline{\sigma}_G = \sigma \cup \{ sm \in G \mid s \in \sigma \} \) holds for all strategies \( \sigma : G \).

**Example 2.8** The terminal game \( T := \{ \epsilon \} \) only has the strategy \( \top := \{ \epsilon \} \).

The flat game on a set \( S \) is the game \( \text{flat}(S) := \text{Pref}(\{ q^{\text{OQ}}, m^{\text{PA}} \mid m \in S \}) \), where \( q \) is an arbitrarily fixed element such that \( q \notin S \), and \( q^{\text{OQ}} \) justifies \( m^{\text{PA}} \). It has strategies \( \bot := \{ \epsilon \} \) and \( m := \{ \epsilon, q m \} \) for each \( m \in S \). Consider, for instance, the empty game \( 0 := \text{flat}(\emptyset) \) and the natural number game \( N := \text{flat}(\mathbb{N}) \).

Next, recall that not every strategy corresponds to a proof. For instance, the empty game \( 0 \) models falsity, and thus the strategy \( \bot : 0 \) should not be an interpretation of a proof. We therefore carve out strategies for proofs as winning ones:

**Definition 2.9** (Constraints on strategies [38, 41, 35, 22]) A strategy \( \sigma : G \) is

- **Total** if it always responds: \( \forall s \in \sigma, sm \in G. \exists smn \in \sigma \); 
- **Innocent** if it only depends on P-views: \( \forall smn \in \sigma, tl \in G. [sm] = [tl] \Rightarrow \exists tlr \in \sigma. [smn] = [tlr] \); 
- **Noetherian** if there is no strictly increasing (with respect to the prefix relation \( \preceq \)) infinite sequence of elements in the set \( [\sigma] := \{ [s] \mid s \in \sigma \} \); 
- **Winning** if it is total, innocent and noetherian;
- **Well-bracketed (w.b.)** if its ‘question-answering’ in P-views is in the ‘last-question-first-answered’ fashion: If \( sqta \in \sigma \), where \( \lambda^{\text{QA}}(q) = Q \), \( \lambda^{\text{QA}}(a) = A \) and \( J_{sqta}(a) = q \), then each question occurring in \( t' \), where the P-view \( [sqt] \) is of the form \( [sqt] = [sq].t' \) by visibility, justifies an answer occurring in \( t' \).

**Example 2.10** The strategies \( \top : T \) and \( \emptyset : N \) for all \( n \in \mathbb{N} \) are winning and w.b., while the strategies \( \bot : 0 \) and \( \bot : N \) are not even total, let alone winning.

We think of winning strategies as proofs in classical logic as follows. First, proofs should not get ‘stuck,’ and so strategies for proofs must be total. Next, imposing innocence on strategies corresponds to excluding stateful terms [22, §2.9]. Since logic is concerned with truths, which are independent of ‘passage of time,’ proofs should not depend on ‘states of arguments.’ Hence, we impose innocence on strategies for proofs. In addition, we need noetherianity to handle infinite plays: If a play by an innocent, noetherian strategy keeps growing infinitely, then it cannot be \( P \)’s ‘intention,’ and so the play must be win for \( P \). Technically, noetherianity is crucial for the closure of winning strategies under composition (Definition 2.14) [38].

Further, well-bracketing bans classical reasoning or control operators [22, §2.10]. Hence, we regard winning, w.b. strategies as proofs in intuitionistic logic.
2.3 Constructions on games and strategies

In this section, we briefly recall constructions on games and strategies. Since they are standard in the literature, we leave expositions and examples to [22, §3.2].

Convention. We omit ‘tags’ for disjoint union \( \sqcup \). For instance, we write \( x \in A \sqcup B \) if \( x \in A \) or \( x \in B \); given relations \( R_A \subseteq A \times A \) and \( R_B \subseteq B \times B \), we write \( R_A \sqcup R_B \) for the relation on \( A \sqcup B \) such that \((x, y) \in R_A \sqcup R_B \) if \((x, y) \in R_A \) or \((x, y) \in R_B \).

Definition 2.11 (Constructions on arenas [35]) Given arenas \( A \) and \( B \), we define

1. \( A \sqcup B := (M_A \sqcup M_B, \vdash_A \sqcup \vdash_B) \);
2. \( A \multimap B := (\{a^{(x \to y)} \mid a^{xy} \in M_A\} \sqcup M_B, \vdash_{A \multimap B}), \quad O^0 := P, \; P^0 := O, \; * \vdash_{A \multimap B} m : \iff * \vdash_B m \) and \( m \vdash_{A \multimap B} n : \iff m \vdash_A n \lor m \vdash_B n \lor (\forall \vdash_B m \land * \vdash_A n) \).

Definition 2.12 (Constructions on games [35]) Given games \( G \) and \( H \), we define

1. \( G \otimes H := \{ s \in \mathcal{L}(G) \sqcup \mathcal{L}(H) \mid \forall X \in \{G, H\}, s \vdash X \subseteq X \} \), called the tensor of \( G \) and \( H \), where \( s \vdash X \subseteq s \) consists of occurrences of moves in \( X \);
2. \( G^! := \{ s \in \mathcal{L}(G) \mid \forall i \in \text{occ}(s). J_a(i) = 0 \Rightarrow s \vdash \{ (s(i), i) \} \in G \}, \) called the exponential of \( G \), where \( s \vdash \{ (s(i), i) \} \subseteq s \) consists of occurrence in \( s \) hereditarily justified (Footnote 5) by the initial occurrence \((s(i), i)\) in \( s \);
3. \( G \otimes H := \{ s \in \mathcal{L}(G) \sqcup \mathcal{L}(H) \mid \forall s. J_a(i) = 0 \Rightarrow s \vdash \{ (s(i), i) \} \in G \}, \) called the product of \( G \) and \( H \);
4. \( G \multimap H := \{ s \in \mathcal{L}(G) \multimap \mathcal{L}(H) \mid s \vdash G^! \in G, s \vdash H \subseteq H \} \), also written \( H^G \), where \( s \vdash G^! \) is obtained from \( s \vdash G \) by modifying all the moves \( m^{(x \to y)} \) occurring in \( s \vdash G \) into \( m^x^y \), called the linear implication from \( G \) to \( H \);
5. \( G \Rightarrow H := !G \multimap H \), called the implication from \( G \) to \( H \).

Notationally, exponential \(!\) precedes other constructions on games, while tensor \( \otimes \) and product \( \otimes \) & do linear implication \( \multimap \) and implication \( \Rightarrow \).

Lemma 2.13 (Well-defined constructions on games) Games and w.f. games are closed under \( \otimes \), \( \& \), \( ! \) and \( \multimap \), and w.o. ones under \( \& \), \( \Rightarrow \) and \( \Rightarrow \).

Proof. See [35] for the closure of MC-games under these constructions. The proof is essentially the same for games. The preservation of w.f. (resp. w.o.) ones is clear.

We leave it to the reader to verify that these constructions on games correspond to those on economical MC-games [35] under the bijection \( \mathcal{M} \) (except that we have to exclude unused moves and enabled pairs from an MC-game \( G \Rightarrow H \) if \( H = T \)).

Definition 2.14 (Constructions on strategies [35]) Given strategies \( \phi : A \multimap B \), \( \sigma : C \multimap D \), \( \tau : A \multimap C \), \( \psi : B \multimap C \) and \( \theta : !A \multimap B \), we define

1. \( \text{cp}_A := \{ s \in (A_{[0]} \multimap A_{[1]})^{\text{Even}} \mid \forall t \leq s. \text{Even}(t) \Rightarrow t \vdash A_{[0]} = t \vdash A_{[1]} \} \), called the copy-cat on \( A \);
2. \( \text{der}_A := \{ s \in (!A \multimap A)^{\text{Even}} \mid \forall t \leq s. \text{Even}(t) \Rightarrow t \vdash !A^! = t \vdash A \} \), called the dereliction on \( A \);
3. \( \phi \circ \sigma := \{ s \in A \otimes C \multimap B \otimes D \mid s \vdash A, B \in \phi, s \vdash C, D \in \sigma \} \), called the tensor of \( \phi \) and \( \sigma \), where \( s \vdash A, B \subseteq s \) (resp. \( s \vdash C, D \subseteq s \)) consists of occurrences of moves in \( A \) or \( B \) (resp. \( C \) or \( D \));


• \langle \phi, \tau \rangle := \{ s \in A \to B \& C \mid (s \mid A, B \in \phi \& s \mid C = \epsilon) \lor (s \mid A, C \in \tau \& s \mid B = \epsilon) \}, \text{called the pairing of } \phi \text{ and } \tau; \\
• \phi; \psi := \{ s \mid A, C \mid s \in \phi \parallel \psi \}, \text{called the composition of } \phi \text{ and } \psi, \text{where } \phi \parallel \psi := \{ s \in \mathcal{J} \mid s \mid A, B_{[0]} \in \phi, s \mid B_{[1]} \in \psi, s \mid B_{[0]} \perp B_{[1]} \in \mathcal{CP}_{B} \}, \\
\mathcal{J} := \mathcal{J}^{\text{Arm}(((A \to B_{[0]}) \to B_{[1]}) \Rightarrow C)}; s \mid B_{[0]} \perp B_{[1]} \text{ is obtained from } s \mid B_{[0]}, B_{[1]} \text{ by applying the operation } (\cdot)^{\perp} \text{ on all the moves, and } \phi; \psi \text{ is also written } \psi \circ \phi; \\
• \theta^{\dagger} := \{ s \in (A \to !B)_{\text{Even}} \mid \forall i \in |s|, \mathcal{J}_{s}(i) = 0 \Rightarrow s \mid \{(s(i), i)\} \in \theta \}, \text{called the promotion of } \theta.

For the dereliction \text{der}_A \text{ to be well-defined, we have to focus on } w.o. \text{ games } A; \text{ see [35, pp. 42–43]. Although } w.o. \text{ games are not closed under exponential }, \text{ it does not matter for us, like [35, p. 43], since we only need cartesian closure, not exponential! itself, and } w.o. \text{ games are closed under implication } \Rightarrow \text{ and product } \& (\text{Lemma 2.13}).

**Lemma 2.15 (Well-defined constructions on strategies)**  If \( \phi : A \to B, \sigma : C \to B, \tau : A \to C, \psi : B \to C \) and \( \theta : !A \to B \), then \( \mathcal{CP}_{A} : A \to A, \phi \otimes \sigma : A \otimes C \to B \otimes D, \mathcal{CP} \otimes \mathcal{CP} \) are winning \( w.o. \) strategies \( \phi; \psi : A \to C \) and \( \theta^{\dagger} : !A \to !B; \) also, \( \text{der}_B : B \Rightarrow B \) if \( B \) is \( w.o. \). Moreover, \( \mathcal{CP}_{A} \) (resp. \( \text{der}_B \)) is winning \( w.b. \) if \( A \) is \( w.f. \) (resp. if \( B \) is \( w.o. \) and \( w.f. \)), and \( \otimes, (\cdot \wedge), \circ \) and \( (\cdot)^{\dagger} \) preserve winning \( \text{ and wellBracketing!} \)

**Proof** The only nontrivial point not shown in [35] is that \( \mathcal{CP}_{A} \) is noetherian if \( A \) is w.f. (the case for \( \text{der}_B \) is the same). Note that \( \mathcal{CP}_{A} \) is total, innocent and \( w.b. \) even if \( A \) is not w.f. Given \( \text{smm} \in \mathcal{CP}_{A} \), we see by induction on \( |s| \) that the P-view \( \text{view } [\text{sm}] \) is of the form \( m_{1}m_{2}m_{3}\ldots m_{k}m_{k+} \), and therefore there is a sequence \( * \vdash A m_{1} \vdash A m_{2} \cdots \vdash A m_{k-1} \vdash A m_{k} \vdash A m_{k+} \). Hence, \( \mathcal{CP}_{A} \) is noetherian if \( A \) is w.f. 

**Definition 2.16 (Categories of games [35, 22])** The category \( \mathcal{G} \) consists of

- W.o. games as objects;
- Strategies on the implication \( A \Rightarrow B \) as morphisms \( A \to B; \)
- The composition \( \psi \parallel \phi := \psi \circ \phi^{\dagger} : A \Rightarrow C \) of strategies as the composition of morphisms \( \phi : A \to B \) and \( \psi : B \to C; \)
- The dereliction \( \text{der}_A \) as the identity on each object \( A \).

The subcategory \( \mathcal{LG} \) (resp. \( \mathcal{WG} \)) of \( \mathcal{G} \) consists of w.f., w.o. games as objects, and winning (resp. winning, w.b.) strategies as morphisms.

Games in \( \mathcal{G} \) (resp. \( \mathcal{LG} \) and \( \mathcal{WG} \)) are w.o. (resp. w.o. and w.f.) for the identities to be well-defined (Lemma 2.15). Strategies in \( \mathcal{G} \) embody unconstrained, general computations. In contrast, strategies in \( \mathcal{LG} \) (resp. \( \mathcal{WG} \)) are winning (resp. winning and w.b.), embodying proofs in classical logic (resp. proofs in intuitionistic logic).

These categories are cartesian closed, where a terminal object, product and exponential objects are the terminal game \( T \), product \& and implication \( \Rightarrow \), respectively. Since our morphisms are the same as those in the cartesian closed categories of MC-games [22], they satisfy the equational axioms on cartesian closure in the same way.

Therefore, by Lemmata 2.13 and 2.15, we conclude:

**Theorem 2.17 (Well-defined cartesian closed categories of games)** The structures \( \mathcal{G}, \mathcal{LG} \) and \( \mathcal{WG} \) form cartesian closed categories.
Notation  We employ the following notations:

- Given a strategy \( \sigma : G \), we write \( \sigma^T : T \rightarrow G \) and \( \sigma'^T : T \Rightarrow G \) for the evident strategies that coincide with \( \sigma \) up to ‘tags’;
- Given strategies \( \phi : T \rightarrow G \) and \( \phi' : T \Rightarrow G \), we write \( \phi_T, \phi'_T : G \) for the evident strategies that coincide with \( \phi \) and \( \phi' \) up to ‘tags’ respectively;
- Given strategies \( \psi : A \rightarrow B \) and \( \alpha : A \), we define \( \psi \circ \alpha := (\psi \circ \alpha^T) : B \);
- Given strategies \( \alpha : A \) and \( \beta : B \), we define \( \alpha \otimes \beta := ((\alpha^T \otimes \beta^T) \circ \iota) : A \otimes B \),
- where \( \iota \) is the unique strategy on \( T \rightarrow T \otimes T \), and \( \langle \alpha, \beta \rangle := \langle \alpha^T, \beta^T \rangle_T : A & B ;
- Given a strategy \( \alpha : A \), we define \( \alpha^T := ((\alpha^T)^T)^T : !A \).

3 Predicate games

Having reviewed games and strategies in §2, let us now initiate our contributions. Before going into details, we sketch our idea in the following paragraphs. In short, our main challenge is to interpret Sigma-types without destroying the non-inductive nature of game semantics or the additive nature of product & (§1.4), and we achieve it by generalising games. On the other hand, we keep strategies unchanged so that our method retains the advantages of game semantics such as intensionality (§1.2).

Naively, we can interpret each dependent type \( x : C \vdash D(x) \) type by a family \( D = (D(\sigma))_{\sigma \in C} \) of games \( D(\sigma) \) indexed by strategies \( \sigma \) on the game \( C \) that models the simple type \( C \). In the presence of Sigma-types, dependent types with only a single variable cover those with more than one variable, and so we focus on the former.

In light of product & (Definition 2.12), which models a particular kind of Sigma-types, viz., product types, it seems a natural idea to model the Sigma-type \( \Sigma_{\alpha \in C}D(\alpha) \) by a subgame \( \Sigma(C, D) \subseteq C \& \bigcup_{\alpha \in C} D(\sigma) \) such that strategies on \( \Sigma(C, D) \) are the pairings \( \langle \sigma, \tau \rangle \) of \( \sigma : C \) and \( \tau : D(\sigma) \). However, this idea does not work since

1. Each game \( G \), by definition, determines the set \( st(G) \) of all strategies on \( G \);
2. It is impossible for \( P \), when playing on such a game \( \Sigma(C, D) \), if any, to fix a strategy \( \sigma : C \), let alone a game \( D(\sigma) \), at the beginning of a play.

As an example of the first problem, consider a dependent type \( x : N \vdash N_b(x) \) type such that the canonical terms of the simple type \( N_b(k) \) \((k \in N)\) are numerals \( \underline{n} \) such that \( n \leq k \), and assume that we model \( N_b \) by the family \( N_b = (N_b(\sigma))_{\sigma \in N} \) of games \( N_b(\sigma) \) defined by \( N_b(k) := \text{Pref}(\{ qn \mid n \leq k \}) \) \((k \in N)\) and \( N_b(\underline{\bot}) := N \). However, there is no subgame \( G \subseteq N \& \bigcup_{\sigma \in N} N_b(\sigma) = N \& N \) such that \( \underline{\bot} : G \) if and only if \( \underline{\bot} : N_b(k) \) for all \( k \in N \) since if such a game \( G \) existed, then \( \langle \underline{\bot}, \underline{\bot} \rangle : G \), which implies \( \langle \underline{\bot}, \underline{\bot} \rangle : G \) by the definition of strategies on a game (Definition 2.7), a contradiction. Hence, no game can properly model the Sigma-type \( \Sigma_{x \in N}N_b(x) \).

Let us next give an example of the second problem. Let \( x : N \vdash \text{List}_N(x) \) type be a dependent type such that the canonical terms of the simple type \( \text{List}_N(k) \) \((k \in N)\) are the \( k \)-lists of numerals, and assume that we model \( \text{List}_N \) by the family \( \text{List}_N = (\text{List}_N(\sigma))_{\sigma \in N} \) of games \( \text{List}_N(\sigma) \) such that \( \text{List}_N(k) \) \((k \in N)\) is the \( k \)-ary tensor \( \otimes \) on \( N \), where \( \text{List}_N(\underline{\bot}) := T \), and \( \text{List}_N(\underline{\bot}) := \bigcup_{k \in N} \text{List}_N(k) \). If there were a subgame \( H \subseteq N \& \bigcup_{\sigma \in N} \text{List}_N(\sigma) \) that models the Sigma-type \( \Sigma_{x \in N}\text{List}_N(x) \), then the pairings \( \langle k, n_1 \otimes n_2 \otimes \cdots \otimes n_k \rangle \) for all \( k, n_1, n_2, \ldots, n_k \in N \) would be total on \( H \) since strategies for proofs must be winning (§2.2). However, there is no such \( H \) since \( O \) may select, by his first move, e.g., the \((k + 1)\)st component of \( \text{List}_N(k + 1) \).

We have seen the two fundamental limitations of games in modelling Sigma-types. We solve this problem by generalising games to pairs \( \Gamma = (|\Gamma|, ||\Gamma||) \) of a game \(|\Gamma|\)
and a family $\|\Gamma\| = (\Gamma(\gamma))_{\gamma \in \Gamma}$ of subgames $\Gamma(\gamma) \subseteq \Gamma$, called predicate (p-) games, and define strategies $\gamma$ on $\Gamma$, written $\gamma : \Gamma$, to be those $\gamma : \|\Gamma\|$ satisfying $\Gamma(\gamma) \subseteq \Gamma$. That is, a p-game $\Gamma$ is a game $\|\Gamma\|$ equipped with the specification $\|\Gamma\|$ for strategies $\gamma : \|\Gamma\|$ to be on $\Gamma$: The restriction of $\gamma$ to $\Gamma(\gamma)$, i.e., $\Gamma(\gamma)$, is a strategy on $\Gamma(\gamma)$.

A play in a p-game $\Gamma$ then proceeds as follows. First, Judge (J) asks P a question $q_1$ (‘What is your strategy?’) and P answers it by a strategy $\gamma : \Gamma$ (‘It is $\gamma$!’). After this initial protocol between J and P, an ordinary play on the game $\Gamma(\gamma)$ between P and O follows, in which P must employ the declared strategy $\gamma$ restricted to $\Gamma(\gamma)$, i.e., $\Gamma(\gamma) : \Gamma(\gamma)$. Accordingly, $\gamma : \Gamma$ is winning (resp. w.b.) if so is $\Gamma(\gamma) : \Gamma(\gamma)$.

This generalisation of games to p-games solves the first problem as follows. Let us define a p-game $\Sigma(N, N_b)$ by $|\Sigma(N, N_b)| := N \& N$ and $\Sigma(N, N_b)(\langle \sigma, \tau \rangle) := \begin{cases} N \& N_b(k) & \text{if } \sigma = k \text{ for some } k \in \mathbb{N} \\ N \& N & \text{otherwise} \end{cases}$ for all $\langle \sigma, \tau \rangle : |\Sigma(N, N_b)|$. Then, observe that strategies on the p-game $\Sigma(N, N_b)$ are the pairings $\langle \sigma, \tau \rangle : N \& N$ such that $\tau : N_b(k)$ if $\sigma = k$. For instance, typical plays by the strategy $\langle 7, 3 \rangle : \Sigma(N, N_b)$ are $\Sigma(N, N_b)(7, 3) \vdash q_7$ where J first asks P the question $q_\Sigma(N, N_b)$ (‘What is your strategy?’), and P answers it by $\langle 7, 3 \rangle : \Sigma(N, N_b)$ (‘It is $\langle 7, 3 \rangle$!’); then, an ordinary play between P and O on the game $\Sigma(N, N_b)(\langle 7, 3 \rangle) = N \& N_b(7)$ follows, where P must play by $\overline{\Sigma(N, N_b)(N \& N_b(7))} = \langle 7, 3 \rangle$. The arrows in the diagram represent pointers in j-sequences (Definition 2.3).

Although the declaration of a strategy is not necessary in this example, it is clear why P cannot play by, e.g., $\langle 0, 1 \rangle$ on $\Sigma(N, N_b)$ (‘It is $\langle 0, 1 \rangle$!’); then, an ordinary play between P and O on the game $\Sigma(N, N_b)(\langle 0, 1 \rangle) = N \& N_b(0)$ follows, where P must play by $\overline{\Sigma(N, N_b)(N \& N_b(0))} \nsubseteq \Sigma(N, N_b)(\langle 0, 1 \rangle)$. In this way, the specification $|\Sigma(N, N_b)|$ solves the first problem by filtering strategies. Moreover, the declaration of a strategy solves the second problem: The p-game $\Sigma(N, \text{List}_N)$ defined by $\Sigma(N, \text{List}_N)(\langle k \rangle) := N \& (N \& \cdots \& N)$ for all $k \in \mathbb{N}$, $\Sigma(N, \text{List}_N)(\bot) := \bigcup_{k \in \mathbb{N}} \Sigma(N, \text{List}_N)(\langle k \rangle)$ and $|\Sigma(N, \text{List}_N)| := |\Sigma(N, \text{List}_N)(\bot)|$ models the Sigma-type $\Sigma_{\mathbb{N}} \text{List}_N(\langle x \rangle)$. Typical plays in $\Sigma(N, \text{List}_N)$ look like $\Sigma(N, \text{List}_N)(\langle 2, 1 \otimes 3 \rangle) \vdash q_7$ where the declaration of the strategy $\langle 2, 1 \otimes 3 \rangle : \Sigma(N, \text{List}_N)$ fixes the underlying game $N \& (N \& N)$. Thus, $\overline{\Sigma(N, \text{List}_N)(\langle 2, 1 \otimes 3 \rangle)} = \langle 2, 1 \otimes 3 \rangle$ is total on $\Sigma(N, \text{List}_N)$.

In this way, we achieve game semantics of Sigma-types directly without the list construction of Abramsky et al. [23, 29] (§1.4). Let us add another important point that their method cannot properly interpret the Sigma-type $\Sigma(N, \text{List}_N)$ since total strategies on their interpretation of this Sigma-type are the lists $\langle k, n_1 \otimes n_2 \otimes \ldots \rangle$ of strategies, where the second component is an infinite iteration of tensor $\otimes$, due to the O-sat operation [29, Remark 4.5]; we come back to this point in §4.3.
Besides, our interpretation of Sigma-types retains the additive nature of product & because a play in the interpretation occurs in either side, not both, which is in contrast with the interpretation of Sigma-types by Blot and Laird [24] (§1.4).

Finally, we justify p-games, in particular the use of J, as a generalisation of games as follows. First, J is also assumed at least implicitly in conventional games as well since there must be someone other than P or O to check if j-sequences played by P and O are valid positions in the underlying game. Hence, the use of J is not a big departure from games. However, this argument is only conceptual, and the following mathematical arguments matter much more. Second, even in the literature of game semantics, P always plays by a fixed strategy as well. Besides, this predetermination of a strategy does not lose generality since each position s in a game G is the result of a play by some σ : G, viz., \( σ := \text{Pref}(\{s\})^{\text{Even}} \). Third, the initial two elements played in p-games are by J and P, not O and P, so that p-games inherit the intensionality of games: Strategies for P are revealed to O only gradually along the development of a play. I.e., the declaration of a strategy by P is hidden from O, so that O can see the strategy only gradually via his play against the strategy (as in the case of games). Technically, we implement this idea by excluding the initial two elements from ordinary plays (especially from O-views). Dually, O’s strategy on the domain \( Γ \) of a linear implication \( Γ \vdash Δ \) between p-games is revealed only gradually to P via plays. Consequently, the extensions of strategies on \( Γ \vdash Δ \) are continuous maps as in the case of games. More generally, since our strategies are just the ordinary ones, our method inherits the intensionality of standard game semantics; see §4.6–4.7.

In summary, since the declarations of strategies by P and O are invisible to each other, plays in p-games proceed just like those in games, and our approach inherits the intensionality of game semantics. We emphasise this intensionality since due to the declarations the reader may mistake our model to be close to extensional ones.

The rest of this section proceeds as follows. We first define p-games in §3.1, and then generalise constructions on games (§2.3) to p-games in §3.2. In these sections, we slightly modify and formalise the aforementioned interpretation of dependent types (Definition 3.3) as well as the examples List\(_N\) and \( N_b \) (Example 4.2).

### 3.1 Predicate games

We reformulate the idea of p-games \( Γ = (|Γ|, ||Γ||) \) sketched above in a handier way as follows. First, recall that the closure \( \overline{γ} |Γ| \) of a strategy \( γ : |Γ| \) with respect to the ambient game \( |Γ| \) is characterised by a simple form \( \overline{γ} |Γ| = γ \cup \{ sm ∈ |Γ| | s ∈ γ \} \).

Second, the central condition \( \overline{γ} |Γ| \cdot Γ(γ) \) is equivalent to another relation \( \overline{γ} |Γ| ≤ Γ(γ) \), where the symbol \( ≤ \) denotes Chroboczek’s beautiful liveness ordering:

**Definition 3.1 (Liveness ordering [42])** The liveness ordering is a partial order \( ≤ \) between games [42, Definition 8 and Theorem 9], which defines \( G ≤ H \) to mean that O (resp. P) is less (resp. more) restricted in \( G \) than in \( H \), i.e., they satisfy

1. If \( s ∈ (G ∩ H)^{\text{Even}} \) and \( sm ∈ H^{\text{Odd}} \), then \( sm ∈ G^{\text{Odd}} \);
2. If \( tl ∈ (G ∩ H)^{\text{Odd}} \) and \( tlr ∈ G^{\text{Even}} \), then \( tlr ∈ H^{\text{Even}} \).

**Proposition 3.2 (Liveness characterisation)** Assume \( σ : G \) and \( H ∈ \text{sub}(G) \).

1. \( \overline{σ}_H : H \) if and only if \( \overline{σ}_G ≤ H \);
2 If $\varphi_G \leq H$, then $\varphi_H^{\text{Even}} = \sigma \cap H$.

Proof We focus on the first clause since it is just a routine to verify the second one.

First, it is straightforward to see that the relation $\varphi_H^{\text{Even}} : \Gamma$ is equivalent to the subset relation $\varphi_H^{\text{Even}} \subseteq H^{\text{Even}}$. Hence, it suffices to show $\varphi_H^{\text{Even}} \subseteq H^{\text{Even}}$ if and only if $\varphi_G \leq H$. The sufficiency ($\Rightarrow$) is shown straightforwardly by the induction on the lengths of positions, which we leave to the reader; we focus on the necessity ($\Leftarrow$).

Assume $\varphi_G \not\leq H$; it remains to show $\varphi_H^{\text{Even}} \not\subseteq H^{\text{Even}}$. By $\varphi_G \not\leq H$, there is some $\text{sm} \in \varphi_G^{\text{Even}}$ such that $\text{sm} \in \varphi_G \cap H$ and $\text{sm} \not\in H$. Note that $\text{sm} \in \varphi_G^{\text{Even}}$ implies $\text{sm} \in \sigma$. We also see by induction on the lengths of positions that $H \not\subseteq \varphi_H^{\text{Even}}$ holds, whence $\text{sm} \not\in \varphi_H$. Then, $\text{sm} \in \varphi_H^{\text{Even}}$ follows from $\text{sm} \in \varphi_H$ and $\text{sm} \in \sigma$. We have shown $\varphi_H^{\text{Even}} \not\subseteq H^{\text{Even}}$ since $\text{sm} \in \varphi_H^{\text{Even}}$ and $\text{sm} \not\in H^{\text{Even}}$. $\square$

Proposition 3.2 reduces the relation $\gamma_{\Gamma(\gamma)}: \Gamma(\gamma)$ to another $\gamma_{[\Gamma]} \leq \Gamma(\gamma)$, where $\gamma_{[\Gamma]}$ is often easier to handle than $\gamma_{\Gamma(\gamma)}$ by its simple form $\gamma_{[\Gamma]} = \gamma \cup \{ \text{sm} \in [\Gamma] \mid \text{s} \in \gamma \}$.

Finally, if $\gamma_{[\Gamma]} \leq \Gamma(\gamma)$, then Proposition 3.2 simplifies $\gamma_{\text{Even}(\gamma)}$ by $\gamma_{\text{Even}(\gamma)} = \gamma \cap \Gamma(\gamma)$.

We are now ready to introduce the central concept of the present work:

Definition 3.3 (Predicate games) A predicate (p-) game is a pair $\Gamma = ([\Gamma], ||\Gamma||)$ of a game $[\Gamma]$ and a family $||\Gamma|| = (\Gamma(\gamma))_{\gamma:[\Gamma]}$ of subgames $\Gamma(\gamma) \subseteq [\Gamma]$. It is well-founded (w.f.) (resp. well-opened (w.o.f.)) if so is the game $[\Gamma]$.

A strategy on $\Gamma$, written $\gamma : \Gamma$, is a strategy $\gamma : [\Gamma]$ that satisfies $\gamma_{[\Gamma]} \leq \Gamma(\gamma)$. It is total (resp. innocent, noetherian, w.b.) if so is the strategy $\gamma \cap \Gamma(\gamma) : \Gamma(\gamma)$.

Let $\text{st}(\Gamma) := \{ \gamma \mid \gamma : \Gamma \}$ and $\varphi_{\Gamma} := \varphi_{\Gamma(\gamma)}(\gamma : \Gamma)$. A position in $\Gamma$ is a prefix of a sequence $q \cdot \gamma \cdot s$ such that $\gamma : [\Gamma]$ and $s \in \varphi_{\Gamma}$, where $q \cdot \gamma$ is an arbitrarily fixed element such that $q \cdot \gamma \not\in M_{[\Gamma]}$, $q \cdot \gamma$ is an initial protocol, and $s$ is an actual position.

A play in a p-game $\Gamma$ proceeds as follows. First, Judge (J) asks P a question $q \cdot \gamma$ (‘What is your strategy?’), and P answers it by a strategy $\gamma : \Gamma$ (‘It is $\gamma$!’). After this initial protocol $q \cdot \gamma \cdot s$, an ordinary play on the game $\Gamma(\gamma)$) between O and P follows, in which P must play by $\gamma$ restricted to $\Gamma(\gamma)$, i.e., $\gamma \cap \Gamma(\gamma) : \Gamma(\gamma)$. The crucial points of p-games are that P can only select a strategy $\gamma : [\Gamma]$ that satisfies $\gamma_{[\Gamma]} < \leq \Gamma(\gamma)$ in the initial protocol, and the declaration of $\gamma : \Gamma$ fixes the underlying game $\Gamma(\gamma)$.

In essence, the game-semantic counterpart of the path from STLC to MLTT (§1.1) is the family $||\Gamma||$ added to the game $[\Gamma]$, which brings these strategy filtering and game fixing abilities to $[\Gamma]$. As we have seen, the first and the second abilities address the first and the second problems listed at the beginning of §3, respectively.

If initial protocols were part of ordinary plays or visible to O, then by duality those on the domain of each linear implication would be visible to P, so that the extensions of strategies on linear implication may not be continuous. This extensionality would be undesirable for the intensional nature of game semantics. We address this point by excluding initial protocols from actual positions. In particular, initial protocols are out of the scope of P- and O-views since pointers are only on actual positions.

Example 3.4 Given a game $G$, we define the p-game $\mathcal{P}(G) := (G, \kappa_G)$, where $\kappa_G$ is the constant family at $G$. Clearly, $\text{st}(\mathcal{P}(G)) = \text{st}(G)$. We abbreviate $\mathcal{P}(T)$, $\mathcal{P}(0)$ and $\mathcal{P}(N)$ as $T$, $0$ and $N$, and call them the terminal p-game, the empty p-game and the natural number p-game, respectively (cf. Example 2.8).
3.2 Cartesian closed categories of predicate games

Next, we lift constructions on games (§2.3) to p-games. The cases of product &-, tensor \( \otimes \) and exponential \( ! \) are simple since we can construct them pointwisely:

**Notation**  Let \( G \) be a game, \( s \in \Gamma G \) and \( i \in \mathbb{N} \). We write \( s \upharpoonright i \) for the j-subsequence of \( s \) that consists of occurrences hereditarily justified by the \((i+1)\text{st}\) initial occurrence in \( s \). For instance, if \( s = q2qlq0 \in \mathbb{N} \), then \( s \upharpoonright 0 = q2 \), \( s \upharpoonright 1 = q1 \) and \( s \upharpoonright 2 = q0 \).

**Convention**  Given a strategy \( \sigma \) on the tensor \( G_0 \otimes G_1 \) of games \( G_i \) \((i = 0, 1)\), let

\[
\sigma \upharpoonright G_i := \begin{cases} 
\sigma_i & \text{if } \sigma = \sigma_0 \otimes \sigma_1 \text{ for (necessarily unique) } \sigma_0 : G_0 \text{ and } \sigma_1 : G_1; \\
\uparrow & \text{otherwise}, \end{cases}
\]

where \( \uparrow \) means being undefined.

Similarly, given a strategy \( \tau \) on the exponential \( !G \) of a game \( G \) and \( j \in \mathbb{N} \), let

\[
\tau \upharpoonright j := \begin{cases} 
\{ s \upharpoonright j \mid s \in \tau \} & \text{if } \{ s \upharpoonright k \mid s \in \tau \} : G \text{ for all } k \in \mathbb{N}; \\
\uparrow & \text{otherwise}.
\end{cases}
\]

Given a p-game \( \Gamma \), we define the value \( \Gamma(\uparrow) \) to be undefined, and the constructions \( \otimes, \rightarrow, \& \) and \( ! \) on undefined games to be undefined. Finally, we extend the relation \( \tau \upharpoonright G \subseteq \Gamma(\gamma) \) by defining that it does not hold if the game \( \Gamma(\gamma) \) is undefined.

**Definition 3.5** (Product and tensor on predicate games)  The **product** of p-games \( \Gamma \) and \( \Delta \) is the p-game \( \Gamma \& \Delta \) defined by \( |\Gamma \& \Delta| := |\Gamma| |\Delta| \) and \( (\Gamma \& \Delta)((\gamma, \delta)) := \Gamma(\gamma) \& \Delta(\delta) \) for all \( (\gamma, \delta) : |\Gamma \& \Delta| \), and their **tensor** is the p-game \( \Gamma \otimes \Delta \) defined by \( |\Gamma \otimes \Delta| := |\Gamma| |\Delta| \) and \( (\Gamma \otimes \Delta)(\sigma) := \Gamma(\sigma | \Gamma|) \otimes \Delta(\sigma | \Delta|) \) for all \( \sigma : |\Gamma \otimes \Delta| \).

**Definition 3.6** (Countable tensor)  The **countable tensor** of a family \( (G_i)_{i \in \mathbb{N}} \) of subgames \( G_i \subseteq H \) is the subgame \( \otimes_{i \in \mathbb{N}} G_i := \{ s \in !H \mid \forall j \in \mathbb{N}. s \upharpoonright j \in G_j \} \subseteq !H \).

**Definition 3.7** (Exponential of predicate games)  The **exponential** of a p-game \( \Gamma \) is the p-game \( \Gamma ! \) defined by \( |\Gamma !| := |\Gamma| \) and \( (\Gamma !)(\sigma) := \otimes_{i \in \mathbb{N}} \Gamma(\sigma | i) \) for all \( \sigma : |\Gamma| \).

Hence, strategies on \( \Gamma \& \Delta \) are the pairings \( (\gamma, \delta) \) of \( \gamma : \Gamma \) and \( \delta : \Delta \). Besides, by the above convention, strategies on \( \Gamma \otimes \Delta \) are the tensors \( \gamma \otimes \delta \) of \( \gamma : \Gamma \) and \( \delta : \Delta \), and strategies on \( \Gamma ! \) are those \( \sigma : |\Gamma| \) such that \( \{ s \upharpoonright i \mid s \in \sigma \} : \Gamma \) for all \( i \in \mathbb{N} \).

In contrast, we cannot apply the pointwise method to linear implication \( \rightarrow \): If we define \( |\Delta \Gamma| := |\Delta|^{|\Gamma|} \) for the linear implication \( \Delta \Gamma \), then it is unclear how to decompose strategies \( \phi : |\Delta \Gamma| \) into those on \( |\Gamma| \) and \( |\Delta| \). We solve this problem based on the elegant game semantics of universal quantification [43, p. 17]:

**Definition 3.8** (Linear implication and implication between predicate games)  The **linear implication** between p-games \( \Gamma \) and \( \Delta \) is the p-game \( \Gamma \rightarrow \Delta \) (also denoted by \( \Delta \Gamma \)) defined by \( |\Delta \Gamma| := |\Delta|^{|\Gamma|} \) and for all \( \phi : |\Delta \Gamma| \)

\[
(\Delta \Gamma)(\phi) := \{ e \cup \{ sm \in |\Delta \Gamma|^{\text{Odd}} \mid s \in (\Delta \Gamma)(\phi), \exists \gamma : \Gamma. sm \in (\phi \circ \gamma)^\Gamma \} \cup \{ tlr \in |\Delta \Gamma|^{\text{Even}} \mid tl \in (\Delta \Gamma)(\phi), \forall \gamma : \Gamma. tl \in (\phi \circ \gamma)^\Gamma \Rightarrow tlr \in (\phi \circ \gamma)^\Gamma \},
\]

and the **implication** between \( \Gamma \) and \( \Delta \) is the linear implication \( \Gamma \Rightarrow \Delta := !\Gamma \rightarrow \Delta \).
The first clause of the inductive definition of the subgame \((\Delta^\Gamma)(\phi) \subseteq |\Delta^\Gamma|\) is the base case. Then, the second one specifies one of the two inductive steps: At an even-length position \(s \in (\Delta^\Gamma)(\phi)^{\text{Even}}\), O can make a move \(m\) as in \(\Delta(\phi \circ \gamma)^{Tr} \subseteq |\Delta^\Gamma|\) for any \(\gamma : \Gamma \not\approx \text{not excluded}\), i.e., \(s \in \Delta(\phi \circ \gamma)^{Tr}\). Finally, the third one stipulates the other inductive step: At an odd-length position \(tl \in (\Delta^\Gamma)(\phi)^{\text{Odd}}\), the next move \(r\) by \(\phi\) must be as in \(\Delta(\phi \circ \gamma)^{Tr} \subseteq |\Delta^\Gamma|\) for any \(\gamma : \Gamma \not\approx \text{not excluded}\), i.e., \(tl \in \Delta(\phi \circ \gamma)^{Tr}\).

The basic idea is that in the subgame \(\Delta^\Gamma(\phi) \subseteq |\Delta^\Gamma|\) O can play as in any subgame \(\Delta(\phi \circ \gamma)^{Tr} \subseteq |\Delta^\Gamma|\) not yet excluded; then, since \(P\) or \(\phi\) should see what \(\gamma : \Gamma\) is only via plays, \(\Delta^\Gamma(\phi)\) only allows \(P\) to play as in \(\Delta(\phi \circ \gamma)^{Tr}\) for all \(\gamma\) not yet excluded. This is crucial for composition of strategies between p-games to be well-defined (Lemma 3.11); see Footnote 9. This additional subtlety is the main difference from the game semantics of universal quantification [43].

For instance, a strategy \(\phi : \Sigma(N_{[0]}, (\text{List}_N)_{[1]}) \rightarrow |N_{[2]}|\) containing \(q_{[2]}m_{[1]}\) is not on \(\Sigma_{(N_{[0]}, (\text{List}_N)_{[1]}) \rightarrow N_{[2]}\}\) whatever \(m\) is since at the odd-length position \(q_{[2]}\) the strategy \(\langle q_{[2]}, T \rangle : \Sigma(N, (\text{List}_N))\) is not excluded, and \(\Sigma(N, (\text{List}_N))(\langle q_{[2]}, T \rangle) = N \& T\).

As emphasised before, strategies \(\phi : \Delta^\Gamma\) are indeed ordinary ones (Definition 2.7), which just satisfy some additional axiom. In particular, \(\phi\) can see O’s strategy \(\gamma : \Gamma\) on the domain \(\Gamma\) only gradually via plays; i.e., the declaration of \(\gamma\) by O is invisible to \(\phi\). In this way, our approach retains the intensionality of game semantics.

Lemma 3.9 (Well-defined constructions on predicate games) P-games and w.f. p-games are closed under \&, \(\otimes\), ! and \(\not\circ\), and \(\rightarrow\), and w.o. ones under \&, \(\not\circ\) and \(\rightarrow\).

Proof Straightforward and left to the reader. □

Lemma 3.10 (Well-defined copy-cats and derelictions between predicate games)
Suppose that \(\Gamma\) is a p-game, and \(\Delta\) is a w.o. p-game.
1. The copy-cat \(\text{cp}_{|\Gamma|}\) is a w.b. strategy on \(\Gamma \rightarrow \Gamma\), and winning if \(\Gamma\) is w.f.;
2. The dereliction \(\text{der}_{|\Delta|}\) is a w.b. strategy on \(\Delta \Rightarrow \Delta\), and winning if \(\Delta\) is w.f.

Proof We focus on the first clause since the second one is similar, where we require \(\Delta\) to be w.o. for the same reason as the case of derelictions between games (§2.3).

We only show \((\text{cp}_{|\Gamma|})_{|\Gamma|\mid|\Gamma|} \equiv \Gamma^\Gamma(\text{cp}_{|\Gamma|})\) as it is the only nontrivial point. We do it by induction on the lengths of positions. The base case and the inductive step on odd-length positions are trivial. For the other inductive step, let \(tlr \in \text{cp}_{|\Gamma|}\) and \(tl \in (\text{cp}_{|\Gamma|})_{|\Gamma|\mid|\Gamma|} \cap \Gamma^\Gamma(\text{cp}_{|\Gamma|})\); we have to prove \(tlr \in \Gamma^\Gamma(\text{cp}_{|\Gamma|})\). By \(tl \in \Gamma^\Gamma(\text{cp}_{|\Gamma|})\) and \(tlr \in \text{cp}_{|\Gamma|}\), it follows that \(tlr\) satisfies the inductive condition for \(tlr \in \Gamma^\Gamma(\text{cp}_{|\Gamma|})\). □

Lemma 3.11 (Well-defined constructions on strategies between predicate games)
Given strategies \(\phi : \Gamma \rightarrow \Delta\), \(\psi : \Delta \rightarrow \Theta\), \(\sigma : \Theta \rightarrow \Xi\), \(\tau : \Gamma \rightarrow \Theta\) and \(\theta : \Gamma \rightarrow \Delta\) between p-games, by applying Definition 2.14 to them, we obtain strategies \(\psi \circ \phi : \Gamma \rightarrow \Theta\), \(\phi \otimes \sigma : \Gamma \otimes \Theta \rightarrow \Delta \otimes \Xi\), \((\phi, \tau) : \Gamma \rightarrow \Delta & \Theta\) and \(\theta : \Gamma \rightarrow \Delta\) between p-games, and these constructions preserve winning and well-bracketing.\[8\]

\[8\]Definition 2.14 is valid here since \(\phi : |\Delta|^{\mid|\Gamma|\}}\), \(\psi : \Theta^{\mid|\Delta|\}}\), \(\sigma : \Theta^{\mid|\Xi|\}}\), \(\tau : \Delta^{\mid|\Xi|\}}\) and \(\theta : \Delta^{\mid|\Gamma|\}}\).

\[8\] We go into details in the proof since this lemma is foundational for the rest of the present work. After proving the lemma, we are more sketchy on similar arguments.
Proof. We focus on the tensor $\phi \otimes \sigma$ and the composition $\psi \circ \phi$ because the other constructions are simpler to verify. First, let us show $\phi \otimes \sigma : \Gamma \otimes \Theta \to \Delta \otimes \Xi$, for which it suffices to verify $(\phi \otimes \sigma)_{\Delta \otimes \Xi} \subseteq (\Delta \otimes \Xi)_{\Gamma \otimes \Theta}(\phi \otimes \sigma)$. We prove it by induction on the lengths of positions. The base case and the inductive step on odd-length positions are trivial. For the other inductive step, let $s\!m\!n \in \phi \otimes \sigma$ and $s\!m \in (\phi \otimes \sigma)_{\Delta \otimes \Xi}(\phi \otimes \sigma)$; we have to show $s\!m\!n \in (\Delta \otimes \Xi)_{\Gamma \otimes \Theta}(\phi \otimes \sigma)$.

By $s\!m \in (\Delta \otimes \Xi)_{\Gamma \otimes \Theta}(\phi \otimes \sigma)$, it suffices to show $s\!m\!n \in (\Delta \otimes \Xi)((\phi \otimes \sigma) \circ \varphi)_{\Gamma \otimes \Theta}$. Fix such $\varphi$. Note that $\varphi = \gamma \otimes \vartheta$ for unique $\gamma : \Gamma$ and $\vartheta : \Theta$, and $(\phi \otimes \sigma) \circ \varphi = (\phi \circ \gamma) \otimes (\sigma \circ \vartheta)$. Hence, it suffices to show $s\!m\!n \upharpoonright \Delta^t \in (\Delta \otimes \Xi)(\phi \circ \gamma)^{\mathcal{R}}$ and $s\!m\!n \upharpoonright \Xi^s \in \Xi(\phi \circ \vartheta)^{\mathcal{R}}$. Assume $n$ in $\phi$; the other case is similar. Then, we have $s\!m \upharpoonright \Delta^t \in (\Delta \otimes \Xi)(\phi \circ \gamma)^{\mathcal{R}}$ and $s\!m \upharpoonright \Xi^s = s\!m \upharpoonright \Xi^s \in \Xi(\phi \circ \vartheta)^{\mathcal{R}}$. It remains to show $s\!m\!n \upharpoonright \Delta^t \in (\Delta \otimes \Xi)(\phi \circ \gamma)^{\mathcal{R}}$, but it follows from $\phi : \Delta^t$, $s\!m \upharpoonright \Delta^t \in (\Delta \otimes \Xi)(\phi \circ \gamma)^{\mathcal{R}}$. Assume $n \in \phi \otimes \sigma$, and $s\!m \upharpoonright \Delta^t \in \Xi(\phi \circ \gamma)^{\mathcal{R}}$, where $s\!m \upharpoonright \Delta^t \in (\Delta \otimes \Xi)(\phi \circ \gamma)^{\mathcal{R}}$. It remains to show $s\!m\!n \upharpoonright \Delta^t \in (\Delta \otimes \Xi)(\phi \circ \gamma)^{\mathcal{R}}$, but it follows from $s\!m\!n \upharpoonright \Delta^t \in (\Delta \otimes \Xi)(\phi \circ \gamma)^{\mathcal{R}}$. Assume $l \in \Gamma$; the other case is analogous. Note that $t\!l\!r = (w \upharpoonright |\Gamma|, |\Theta|)l$ for some unique $w \in \phi \parallel \psi$ (Definition 2.14) by the covering lemma [16, p. 12]. The computation of $\phi \parallel \psi$ on $w\!l\!r$ produces $r$ after playing a finite fragment $u$ of a position in the intermediate game $\text{Pref}(cp_{|\Delta|})$; see [16, p. 11]. Crucially, $\phi : \Delta^t$ and $\psi : \Theta^t$ imply by induction that $u$ is a suffix of a position in $(\text{cp}_{|\Delta|})_{\Delta(\phi \circ \gamma)(\sigma \circ \vartheta)}$. Hence, $t\!l\!r \in \Theta(\psi \circ \phi \circ \gamma)^{\mathcal{R}}$. Finally, the same inductive argument verifies that $\psi \circ \phi$ is winning if so are $\phi$ and $\psi$, and it trivially follows from Lemma 2.15 that $\psi \circ \phi$ is w.b. if so are $\phi$ and $\sigma$. □

We are now ready to summarise the present section by:

**Definition 3.12 (Categories of predicate games)** The category $\mathcal{PG}$ consists of

- W.o. p-games as objects;
- Strategies on the implication $\Gamma \Rightarrow \Delta$ as morphisms $\Gamma \Rightarrow \Delta$;
- The composition $\psi \circ \phi : \Gamma \Rightarrow \Theta$ of strategies as the composition of morphisms $\phi : \Gamma \to \Delta$ and $\psi : \Delta \to \Theta$;
- The dereliction $\text{der}_{|\Gamma|} : \Gamma \Rightarrow \Gamma$ as the identity $\text{id}_{\Gamma}$ on each object $\Gamma$.

The subcategory $\mathcal{LPG}$ (resp. $\mathcal{WPG}$) of $\mathcal{PG}$ consists of w.f., w.o. p-games as objects, and winning (resp. winning, w.b.) strategies as morphisms.

**Remark** As in the case of games [35, §3.3], there are the linear counterparts of the categories $\mathcal{PG}$, $\mathcal{LPG}$ and $\mathcal{WPG}$, in which morphisms $\Gamma \Rightarrow \Delta$ are strategies on the linear implication $\Gamma \Rightarrow \Delta$. We skip them since they are not central in this article. [35, §3.3] In other words, the contact point $\subseteq |\Delta|$ between $\phi$ and $\psi$ matches at each moment.
Nevertheless, our game semantics of MLTT (§4) together with these six categories implies that some combinations of dependent types and linearity/effects are already there in game semantics. We leave it as future work to study such combinations.

Just like the categories of games (Definition 2.16), p-games in PG (resp. LPG and WPG) are w.o. (resp. w.f. and w.o.) for the identities to be well-defined.

**Theorem 3.13** (Well-defined cartesian closed categories of predicate games) The structures PG, LPG and WPG form cartesian closed categories.

**Proof** By Lemma 3.11, it suffices to focus on PG. The composition is well-defined by Lemma 3.11, and so are the identities by Lemma 3.10. Since morphisms in PG are a certain class of those in G, and the composition and the identities in PG are those in G, the associativity and the unit law on PG follow from those on G (Theorem 2.17). The cartesian closure of PG is by Example 3.4 on T and Lemmata 3.9 and 3.11, where the required equations on morphisms again follow from those on G. □

**Convention** We write PG(Γ), LPG(Γ) and WPG(Γ) for the hom-sets PG(T, Γ), LPG(T, Γ) and WPG(T, Γ), respectively, for each object Γ, and do not distinguish Γ and PG(Γ); e.g., we say that γ : Γ is winning and w.b. if and only if γ ∈ WPG(Γ).

Finally, let us show a categorically pleasing feature of p-games (Corollary 3.14). We focus on WPG for brevity, but the following argument is valid in LPG and PG as well. First, we define an equivalence relation ≃ between morphisms ϕ, ϕ′ : Γ → Δ by ϕ ≃ ϕ′ :⇔ ϕ ◦ γ = ϕ′ ◦ γ : Δ for all γ : Γ; i.e., ϕ ≃ ϕ′ means that ϕ and ϕ′ are extensionally equal. Next, recall that a morphism ϕ : Γ → Δ is strict if m^{x^w}n^{z^w} ∈ ϕ implies n^{(z^+)^w} ∈ Γ [44]. We then define a category WPG_{≃} out of WPG, in which objects are those of WPG, and morphisms Γ → Δ are the equivalence classes [ϕ] of strict morphisms ϕ : Γ → Δ in WPG modulo ≃. The composition of morphisms [ϕ] : Γ → Δ and [ψ] : Δ → Θ in WPG_{≃} is given by [ψ] ◦ [ϕ] :≡ [ψ ◦ ϕ], and the identities in WPG_{≃} by id_{Γ} :≡ [der_{Γ}] for all Γ ∈ WPG_{≃}. We leave it to the reader as an easy exercise to prove that WPG_{≃} forms a cartesian closed category.

**Corollary 3.14** (Game-semantic finite limits) WPG_{≃} has all finite limits.

**Proof** It suffices to establish the equaliser of given morphisms [ϕ₁], [ϕ₂] : Γ ⇒ Δ. Let us then show that the required equaliser is given by the pair (Θ, [der_{Θ}]), where Θ is the p-game defined by |Θ| :≡ |Γ| and Θ(θ) :≡ \begin{cases} \Gamma(\theta) & \text{if } \phi₁ \bullet θ = \phi₂ \bullet θ; \\ T & \text{otherwise,} \end{cases}

θ : |Θ|. The p-game Θ is well-defined since it does not depend on the choice of the representatives φᵢ of the morphisms [ϕᵢ] (i = 1, 2). Note that γ : Θ if and only if φ₁ • γ = φ₂ • γ or γ = {ε} for all γ : Γ. Besides, we have φ₁ • {ε} = φ₂ • {ε} because of the strictness of φᵢ (i = 1, 2). Hence, it follows that [ϕ₁] • [der_{Θ}] = [ϕ₂] • [der_{Θ}].

Next, given an object Θ' and a morphism [ϕ] : Θ' → Γ with [ϕ₁] • [ϕ] = [ϕ₂] • [ϕ], we have [ϕ] : Θ' → Θ and [der_{Θ'}] • [ϕ] = [ϕ], where we leave the details to the reader. Finally, if another morphism [ϕ'] : Θ' → Θ satisfies [der_{Θ}] • [ϕ'] = [ϕ], then [ϕ'] = [der_{Θ}] • [ϕ'] = [ϕ]. We have shown the universal property of (Θ, [der_{Θ}]). □
We focus on strict strategies in $\mathbb{WPG}^\dagger_\leq$ for Corollary 3.14 since, e.g., there is no equaliser of the non-strict strategies $0, 1 : N \rightarrow N$. We further take the extensional quotient $[\phi]$ of strict strategies $\phi$ in $\mathbb{WPG}^\dagger_\leq$ since it seems that otherwise the category would not be finitely complete (though we have not shown this negative result).

This categorical structure is quite novel. For instance, if we apply the proof of Corollary 3.14 to $[\kappa_0], [\gamma] : N & N \rightarrow N$ such that $\kappa_0$ is any strict strategy such that $\kappa_0 \cdot \sigma = \emptyset$ for all $\sigma : N & N$, and $\gamma$ is any strict strategy such that $\gamma \cdot (k, n) = \emptyset$ if and only if $k \geq n$ for all $k, n \in \mathbb{N}$, then their equaliser is the p-game that models the Sigma-type $\Sigma_{x \in N} N_b(x)$ (see the beginning of §3). Therefore, this construction is impossible for games (even if we focus on strict strategies and take their quotient).

Corollary 3.14 is also useful: It enables us to internalise a certain notion of $\infty$-groupoids in $\mathbb{WPG}^\dagger_\leq$, which is a key step to extend this work to HoTT [37]. Note that we do not focus on strict strategies or take their quotient when we interpret HoTT, and so it is a proper extension of our game semantics of MLTT to HoTT.

4 Game semantics of Martin-Löf type theory

We are now ready to present our game semantics of MLTT. Concretely, we show that the CCC $\mathbb{WPG}$ forms abstract semantics of MLTT: a category with families (CwF) [45]. CwFs are closer to the syntax than other abstract semantics, so that we can directly see the semantic counterparts of MLTT. In fact, we even regard a CwF as another presentation of MLTT as Clairambault and Dybjer [46] do, and so we only show that $\mathbb{WPG}$ forms a CwF, leaving how a CwF models MLTT to [31].

Specifically, we prove that the CCC $\mathbb{WPG}$ gives rise to a CwF equipped with semantic type formers [31] for One-, Zero-, N-, Pi-, Sigma- and Id-types, establishing game semantics of MLTT equipped with these types.

The rest of this section proceeds as follows. We model dependent types in §4.1, Pi-types in §4.2, and Sigma-types in §4.3. We then show that $\mathbb{WPG}$ forms a CwF in §4.4 and equip it with all the semantic type formers in §4.5. Finally, we analyse the intensionality of our game semantics in §4.6, prove the independence of Markov’s principle from MLTT in §4.7, for which we use $\mathbb{WPG}$, not PG or LPG, and extend the game semantics of MLTT to subtyping on dependent types in §4.8.

4.1 Dependent predicate games

First, we interpret dependent types by w.o., w.f. dependent p-games:

**Definition 4.1** (Dependent predicate games) A linearly dependent predicate (p-) game over a p-game $\Gamma$ is a pair $L = ([L], \|L\|)$ of a game $|L|$ and a family $\|L\| = (L(\gamma_0))_{\gamma_0 \in \mathbb{WPG}(\Gamma)}$ of p-games $L(\gamma_0)$ such that $|L(\gamma_0)| = |L|$. It is well-opened (w.o.) (resp. well-founded (w.f.)) if so is the game $|L|$.

The extension of the family $\|L\|$ is the family $L^* = (L^*(\gamma))_{\gamma, \Gamma}$ of p-games $L^*(\gamma)$ defined by $L^*(\gamma) := \begin{cases} L(\gamma) & \text{if } \gamma \in \mathbb{WPG}; \\ \mathcal{P}(\|L\|) & \text{otherwise (cf. Example 3.4).} \end{cases}$

A dependent predicate (p-) game over $\Gamma$ is a linearly dependent one over $\Gamma$.

**Notation** We write $\mathcal{D}(\Gamma)$ (resp. $\mathcal{D}^w(\Gamma)$) for the set of all linearly dependent p-games (resp. w.o., w.f. ones) over $\Gamma$, and $\{\Gamma\}_\Gamma$ or $\{\Gamma\}'_\Gamma$ for the constant one at $\Gamma'$, i.e., $\{\Gamma\}'_\Gamma := (\Gamma', \gamma : \Gamma \rightarrow \Gamma')$. We define $\mathcal{D}(\Gamma) := \mathcal{D}(\Gamma) \cup \mathcal{D}^w(\Gamma)$.
We often write $\gamma_0^1$ for an arbitrary element of $\text{WPG}(!\Gamma)$, where $\gamma_0 \in \text{WPG}(\Gamma)$, since elements of $\text{WPG}(!\Gamma)$ are all innocent and so promotions of elements of $\text{WPG}(\Gamma)$.

We explain Definition 4.1 as follows. First, we define $p$-games $U$ to model universes and encode w.o., w.f. dependent $p$-games $A$ over a $p$-game $\Gamma$ by morphisms $\phi_A : \Gamma \to U$ in $\text{WPG}$ in a forthcoming article. Thus, naively $||A||$ would be the map $\gamma : !\Gamma \to \text{El}(\phi_A \circ \gamma)$, where $\text{El}(\mu) \in \text{WPG}$ is the $p$-game encoded by each $\mu \in \text{WPG}(U)$. However, $\phi_A \circ \gamma$ can be invalid as an encoding, i.e., $\phi_A \circ \gamma \notin \text{WPG}(U)$, for instance when $\phi_A \circ \gamma$ is partial, where the value $A(\gamma) = \text{El}(\phi_A \circ \gamma)$ is undefined. This suggests us to restrict $||A||$ to $\text{WPG}(!\Gamma)$, so that $\text{El}(\phi_A \bullet \gamma_0) \in \text{WPG}$ for all $\gamma_0^1 \in \text{WPG}(!\Gamma)$. Accordingly, $||A||$ is trivially continuous since elements of $\text{WPG}(!\Gamma)$ are all total.\[10\]

Then, however, we also need the ambient game $|A|$. For instance, if $A = \{N\}_{0\uparrow}$, then the family $|A|$ is empty, and so $||A||$ cannot retain the natural number $p$-game $N$. We address this problem by adding the game $|A|$ and extending $||A||$ to $A^\ast$.

**Example 4.2** We slightly modify and formalise the examples List$_N$, $N_b \in \mathcal{P}^\omega(N)$ by List$_N(\underline{k}) := N \otimes N \otimes \cdots \otimes N$ for all $k \in \mathbb{N}$ and $|\text{List}_N| := \bigcup_{k \in \mathbb{N}} \text{List}_N(\underline{k})$, as well as $|N_b| := N$ and $N_b(\underline{k}) := \text{Pref}(\{qn \mid n \leq k\})$ for all $k \in \mathbb{N}$.

### 4.2 Pi on dependent predicate games

We next interpret $\Pi$-types. Our idea is best explained by the set-theoretic analogy as follows. Given a dependent type $x : C \vdash D(x)$, the $\Pi$-type $\Pi_{x : C} D(x)$ is something like the set of all functions $f$ from $C$ to $\bigcup_{x \in C} D(x)$ such that $f(x) \in D(x)$ for all $x \in C$, called dependent maps from $C$ to $D$, where recall that the set-theoretic semantics interprets simple types $C$ and terms $x : C$ as sets $C$ and elements $x \in C$, respectively, and dependent types $D$ over $C$ as families $D = (D(x))_{x \in C}$ of sets $D(x)$.

Hence, in light of implication $\Rightarrow$ between $p$-games (Definition 3.8), it is now clear how to model $\Pi$-types. We interpret $\Pi$-types by the following pi $\Pi$:

**Definition 4.3** (Linear-pi and pi) Let $L$ be a linearly dependent $p$-game over a $p$-game $\Gamma$, and $A$ a dependent $p$-game over $\Gamma$. The linear-pi from $\Gamma$ to $L$ is the $p$-game $\Pi_{\ell}(\Gamma, L)$ defined by $||\Pi_{\ell}(\Gamma, L)|| := |L||\Gamma|$, and for all $\phi : ||\Pi_{\ell}(\Gamma, L)||$

\[
\Pi_{\ell}(\Gamma, L)(\phi) := \{\epsilon\} \cup \{sm \in ||\Pi_{\ell}(\Gamma, L)|| | \text{Odd} \mid s \in \Pi_{\ell}(\Gamma, L)(\phi), \exists \gamma : \Gamma. sm \in L^*(\gamma)(\phi \circ \gamma)^{\text{C}}\}
\]

\[
\cup \{ tlr \in ||\Pi_{\ell}(\Gamma, L)|| | \text{Even} \mid t \in \Pi_{\ell}(\Gamma, L)(\phi), \forall \gamma : \Gamma. tl \in L^*(\gamma)(\phi \circ \gamma)^{\text{FC}} \Rightarrow tlr \in L^*(\gamma)(\phi \circ \gamma)^{\text{FC}}\},
\]

and the $\Pi$ from $\Gamma$ to $A$ is the linear-pi $\Pi(\Gamma, A) := \Pi_{\ell}(\Gamma, A)$.

The idea of linear-pi is that it is linear implication between $p$-games except that it also satisfies type dependency. Specifically, the codomain of a linear-pi $\Pi_{\ell}(\Gamma, L)$ is the $p$-game $L(\gamma)$ if the strategy $\gamma : \Gamma$ on the domain satisfies $\gamma \in \text{WPG}(\Gamma)$, and the constant one $\mathcal{P}(|L|)$ otherwise. We then define $\pi$ out of linear-pi and exponential in the same way as we define implication out of linear implication and exponential.

Accordingly, linear-pi (resp. $\pi$) generalises linear implication (resp. implication): Given $p$-games $\Gamma$ and $\Gamma'$, $\Pi_{\ell}(\Gamma, \{\Gamma'\}_\|) = \Gamma \to \Gamma'$ (resp. $\Pi(\Gamma, \{\Gamma'\}_\|) = \Gamma \Rightarrow \Gamma'$).

Essentially the same proof as the one of Lemma 3.9 on linear implication shows:

\[\text{Remark 5}\] This point matches the remark by Abramsky et al. [29, Footnote 5] that the continuity of their interpretation of dependent types does not play any roles.
Theorem 4.4 (Well-defined linear-pi) Given a (w.o., w.f.) linearly dependent p-game L over a (w.o., w.f.) p-game Γ, the linear-pi $\Pi_4(\Gamma, L)$ is a (w.o., w.f.) p-game.

This theorem also implies that, given a (w.o., w.f.) dependent p-game A over a (w.o., w.f.) p-game Γ, the pi $\Pi(\Gamma, A)$ is a (w.o., w.f.) p-game. However, we have to handle the case where Γ is a dependent p-game. We address this point in §4.5.1.

Example 4.5 A strategy $\varsigma : \Pi(N, \text{List}_N)$ plays as the dependent map $n \in N \mapsto (0, 0, \ldots, 0) \in \mathbb{N}^n$ as follows. If O makes the first move $q[k]$ ($k \in \mathbb{N}^+$) on the codomain $|\text{List}_N|$, where $(\_)[k]$ is the ‘tag’ for tensor $N[1] \otimes N[2] \otimes \ldots$, then $\varsigma$ asks a question $q[0]$ on the domain $!N[0]$, where $(\_)[0]$ is another ‘tag’ for clarity. Finally, if O plays by $q[k]q[0] \mapsto n[0]$ for any $n \in \mathbb{N}^+$, then $\varsigma$ plays by $q[k]q[0]n[0] \mapsto 0[k]$. If $k \leq n$, then $\mathbf{N}_k \in \text{WPG}(!N)$ on the domain is not yet excluded; $\varsigma$ is compatible with this possibility since its computation so far is within the subgame $N \Rightarrow \text{List}_N(\mathbf{N}_k) \subseteq \Pi(N, \text{List}_N)$.

At this point, O can further play on the subgame $\Pi(N, \text{List}_N)$. For instance, O can make another move $q[k']$ on the codomain such that $k \neq k'$; then, $\varsigma$ again makes the move $q[0]$ on the domain. If O answers it by $n'[0]$ for any $n' \in \mathbb{N}^+$, then $\varsigma$ plays $0[k']$ on the codomain. If $n \neq n'$, $k > n$ or $k' > n$, then $\varsigma$ no longer has to play within the subgame $N \Rightarrow \text{List}_N(\mathbf{N}_k)$. In contrast, if $n = n'$, $k \leq n$ and $k' \leq n$, then $\varsigma$ still has to play on the subgame $N \Rightarrow \text{List}_N(\mathbf{N}_k)$, and its computation still satisfies this condition. In this way, a play by $\varsigma$ proceeds. Hence, we indeed have $\varsigma : \Pi(N, \text{List}_N)$.

4.3 Sigma on dependent predicate games

We next interpret Sigma-types. Recall that by the set-theoretic analogy the Sigma-type $\Sigma_{x \in D(x)}$ type represents the set of all pairs $(c, d)$ such that $c \in C$ and $d \in D(c)$, called dependent pairs on $C$ and $D$. Hence, we model Sigma-types by:

Definition 4.6 (Sigma) The sigma of a p-game $\Gamma$ and a dependent p-game $A$ over $\Gamma$ is the p-game $\Sigma(\Gamma, A)$ defined by $|\Sigma(\Gamma, A)| := |\Gamma| \& |A|$ and $\Sigma(\Gamma, A)(\langle \gamma, \alpha \rangle) := \Gamma(\gamma) \& A^*(\gamma^\dagger)(\alpha)$ for all $\langle \gamma, \alpha \rangle : |\Sigma(\Gamma, A)|$.

The basic idea of a sigma $\Sigma(\Gamma, A)$ is that strategies on $\Sigma(\Gamma, A)$ must be the pairings $\langle \gamma, \alpha \rangle : |\Gamma| \& |A|$ that satisfy $\gamma : \Gamma$ and $\alpha : A(\gamma^\dagger)$ whenever $\gamma \in \text{WPG}(\Gamma)$. The second condition matches the constraint on the pi $\Pi(\Gamma, A)$ as we shall see shortly.

When $A$ is a constant dependent p-game $\{\Gamma'\}_\Gamma$, the sigma $\Sigma(\Gamma, \{\Gamma'\})$ coincides with the product $\Gamma \& \Gamma'$. Thus, sigma $\Sigma$ generalises product $\&$ on p-games.

Theorem 4.7 (Well-defined sigma) Given a (w.o., w.f.) dependent p-game $A$ over a (w.o., w.f.) p-game $\Gamma$, the sigma $\Sigma(\Gamma, A)$ is a (w.o., w.f.) p-game.

Proof Straightforward and left to the reader. □

Let $\Gamma \in \text{WPG}$ and $A \in \mathcal{G}^w(\Gamma)$. Then, $\Sigma(\Gamma, A) \in \text{WPG}$, and it attains the following nontrivial point. First, the categorical view on semantics of MLTT (Definition 4.5) tells us that there must be a bijection between strategies $\psi \in \text{WPG}(\Delta, \Sigma(\Gamma, A))$ and pairs $(\phi, \hat{\alpha})$ of strategies $\phi \in \text{WPG}(\Delta, \Gamma)$ and $\hat{\alpha} \in \text{WPG}(\Pi(\Delta, A(\phi)))$, where
the dependent p-game $A(\phi) \in \mathcal{P}^w(\Delta)$ is defined by $|A(\phi)| := |A|$ and $A(\phi)(\delta) := A(\phi \circ \delta) (\delta \in \mathcal{WPG}(\Delta))$. Next, the constraint on the codomain $A(\phi)$ of $\Pi(\Delta, A(\phi))$ is graduably revealed along the gradual specification of a strategy $\delta$ on the domain $\Delta$. However, since $\Sigma(\Gamma, A)$ itself cannot refer to $\Delta$, unlike $\Pi(\Delta, A(\phi))$, the challenge is to define $\Sigma(\Gamma, A)$ in such a way that strategies $\psi \in \mathcal{WPG}(\Delta, \Sigma(\Gamma, A))$ achieve the bijection. Then, the sigma $\Sigma(\Gamma, A)$ in fact meets this requirement (Theorem 4.11).

**Example 4.8**  The sigmas $\Sigma(N, \text{List}_N)$ and $\Sigma(N, N_b)$ respectively formalise the two examples given at the beginning of §3.

The pairings $(\bar{k}, n_1 \otimes n_2 \otimes \cdots \otimes n_k)$, where $k, n_1, n_2, \ldots, n_k \in \mathbb{N}$, are all winning on $\Sigma(N, \text{List}_N)$ since the declaration of a strategy in a p-game can control odd-length positions. Besides, given a morphism $\phi : \Sigma(N, \text{List}_N) \rightarrow \Delta$ in $\mathcal{WPG}$, the composition $T (\bar{k}, n_1 \otimes n_2 \otimes \cdots \otimes n_k) \circ \phi \Delta$ is well-defined and winning (Lemma 3.11).

In contrast, the list $(\bar{k}, n_1 \otimes n_2 \otimes \cdots \otimes n_k)$ is partial on the list $(N, \text{List}_N)$ in Abramsky et al. [23, 29], where we regard $N$ and the components of List$_N$ as games in the evident way, since there is no way in their method to prevent $O$ from playing arbitrarily on the ambient game $\text{List}_N$. For instance, $(\emptyset, T)$ is partial on $(N, \text{List}_N)$.

Note that the underlying list of (families) of games for $(\bar{k}, n_1 \otimes n_2 \otimes \cdots \otimes n_k)$ cannot be $(N, \{\text{List}_N(\bar{k})\})$ as the composition of their morphisms $(\bar{k}, n_1 \otimes n_2 \otimes \cdots \otimes n_k) : T \rightarrow (N, \{\text{List}_N(\bar{k})\})$ and $\psi : (N, \text{List}_N) \rightarrow \Theta$ is ill-defined. Also, we cannot replace the domain of $\psi$ by $(N, \{\text{List}_N(\bar{k})\})$ either if it cannot handle the other values of $k \in \mathbb{N}$. Hence, the codomain of $(\bar{k}, n_1 \otimes n_2 \otimes \cdots \otimes n_k)$ must be $(N, \text{List}_N)$.

For this problem, they instead take the lists $(\bar{k}, \tau)$ of winning strategies $\bar{k}$ of $N$ and $\tau : \text{List}_N$. However, $\tau$ is redundant as it is total on $\text{List}_N$; e.g., $(\emptyset, T)$ is replaced with any $(\emptyset, n_1 \otimes n_2 \otimes \cdots)$ such that $n_i \in \mathbb{N}$ for all $i \in \mathbb{N}$ in their approach, where $n_1 \otimes n_2 \otimes \cdots$ is an infinite iteration of tensor, but neither of them is canonical. Consequently, their approach cannot properly model the Sigma-type $\Sigma(N, \text{List}_N)$.

Technically, Abramsky et al. employ the $O$-sat operation [29, Remark 4.5 and Theorem 5.6], which allows $O$ to ignore the type dependency of Pi- and Sigma-types, and impose winning on strategies against such unrestricted plays by $O$. However, this method generates a significant gap between their model and MLTT since terms $\vdash (a, b) : \Sigma(A, B)$ satisfy $\vdash a : A$ and $\vdash b : B(a)$, not $x : A \vdash b : B(x)$. How does their full completeness hold then? The answer is that they focus on a very specific class of finite inductive types [29, Figure 7] for which the problem just disappears. However, as we have just seen, their method is unsuited for more standard types such as $\text{List}_N$, which is generated by the elimination rule of $N$-type in the presence of universes.

In summary, the novel mathematical structure of p-games enables us to not only dispense with the list construction but also accurately model Pi- and Sigma-types.

Finally, recall that the interpretation of Sigma-types by Blot and Laird [24] does not preserve the additive nature of product $\&$ on games (§1.4). In contrast, sigma $\Sigma$ preserves the additive nature. The challenge we have overcome is that the additive nature requires that a play in $\Sigma(\Gamma, A)$ is either a play in $\Gamma$ or $A$, but then we have to specify the constraint on plays in $A$ for $\Sigma(\Gamma, A)$ even without playing on $\Gamma$.

---

[11] This problem is vacant in Abramsky et al. [23, 29] as they define morphisms $\Delta \rightarrow \Sigma(\Gamma, A)$ to be the pairs $(\phi, a)$ at the cost of the non-inductive nature of game semantics.
Again, however, this construction of sigma \( \Sigma \) is not general enough for the same reason as the case of pi \( \Pi \) (§4.2); we generalise it in §4.5.2.

### 4.4 A game-semantic category with families

We are now ready to present our game-semantic CwF. Let us first recall the general definition of CwFs introduced by Dybjer [45]:

**Definition 4.9 (CwFs [45, 31])** A category with families (CwF) is a tuple \( C = (C, \text{Ty}, \text{Tm}, \langle \_ \rangle, T, \_\_\_ p, \_\_\_ v, \langle \_ ,\_ \rangle) \), where

- \( C \) is a category with a terminal object \( T \in C \);
- \( \text{Ty} \) assigns, to each object \( \Gamma \in C \), a set \( \text{Ty}(\Gamma) \) of types in the context \( \Gamma \);
- \( \text{Tm} \) assigns, to each pair \( (\Gamma, A) \) of an object \( \Gamma \in C \) and a type \( A \in \text{Ty}(\Gamma) \), a set \( \text{Tm}(\Gamma, A) \) of terms of type \( A \) in the context \( \Gamma \);
- To each \( \phi : \Delta \to \Gamma \in C \), \( \langle \_ \rangle \) assigns a map \( \langle \phi \rangle : \text{Ty}(\Gamma) \to \text{Ty}(\Delta) \), called the *substitution on types*, and a family \( \langle \phi \rangle_A \) of maps \( \langle \phi \rangle : \text{Tm}(\Gamma, A) \to \text{Tm}(\Delta, A(\phi)) \), called the *substitutions on terms*;
- \( \_\_\_ \) assigns, to each pair \( (\Gamma, A) \) of a context \( \Gamma \in C \) and a type \( A \in \text{Ty}(\Gamma) \), a context \( \Gamma \cdot A \in C \), called the *comprehension* of \( A \);
- \( p \) (resp. \( v \)) associates each pair \( (\Gamma, A) \) of a context \( \Gamma \in C \) and a type \( A \in \text{Ty}(\Gamma) \) with a morphism \( p_A : \Gamma \cdot A \to \Gamma \) in \( C \) (resp. a term \( v_A \in \text{Tm}(\Gamma \cdot A, A(p_A)) \)), called the *first projection* on \( A \) (resp. the *second projection* on \( A \));
- \( \langle \_ ,\_ \rangle \) assigns, to each triple \( (\phi, A, \alpha) \) of a morphism \( \phi : \Delta \to \Gamma \) in \( C \), a type \( A \in \text{Ty}(\Gamma) \) and a term \( \alpha \in \text{Tm}(\Delta, A(\phi)) \), a morphism \( \langle \phi, \alpha \rangle_A : \Delta \to \Gamma \cdot A \) in \( C \), called the *extension* of \( \phi \) by \( \alpha \),

that satisfies, for any \( \Theta \in C \), \( \varphi : \Theta \to \Delta \) and \( \alpha \in \text{Tm}(\Gamma, A) \), the equations

- \((\text{Ty-Id})\) \( A\{\text{id}_\Gamma\} = A \);
- \((\text{Ty-Comp})\) \( A\{\phi \circ \varphi\} = A\{\phi\}\{\varphi\} \);
- \((\text{TM-Id})\) \( \alpha\{\text{id}_\Gamma\} = \alpha \);
- \((\text{TM-Comp})\) \( \alpha\{\phi \circ \varphi\} = \alpha\{\phi\}\{\varphi\} \);
- \((\text{Cons-L})\) \( p_A \circ \langle \phi, \alpha \rangle_A = \phi \);
- \((\text{Cons-R})\) \( v_A\{\phi, \alpha\} = \alpha \);
- \((\text{Cons-Nat})\) \( \langle \phi, \alpha \rangle_A \circ \varphi = \langle \phi \circ \varphi, \alpha\{\varphi\} \rangle_A \);
- \((\text{Cons-Id})\) \( \langle p_A, v_A \rangle = \text{id}_{\Gamma \cdot A} \).

Roughly, judgements of MLTT are interpreted in a CwF \( C \) by

\[
\Gamma \vdash \text{ctx} \to [\Gamma] \in C \quad \Gamma \vdash A \text{ type} \to [A] \in \text{Ty}([\Gamma]) \quad \Gamma \vdash a : A \to [a] \in \text{Tm}([\Gamma], [A])
\]

\[
\Gamma = \Delta \vdash \text{ctx} \to [\Gamma] = [\Delta] \quad \Gamma \vdash A = B \text{ type} \to [A] = [B] \quad \Gamma \vdash a = a' : A \to [a] = [a'],
\]

where \([\_]\) denotes the *semantic map* or *interpretation*. See [31] for the details.

Let us now turn to introducing our game-semantic CwF:

**Definition 4.10 (A game-semantic CwF)** We define a CwF \( WPG \) as follows:

- The category \( WPG \) is given in Definition 3.12, and \( T \in WPG \) in Example 3.4;
- \( \text{Ty}(\Gamma) := \mathcal{P}^\times(\Gamma) \cdot (\Gamma \in WPG) \) and \( \text{Tm}(\Gamma, A) := WPG(\Pi(\Gamma, A)) \cdot (A \in \mathcal{P}^\times(\Gamma)) \);
- Given \( \phi : \Delta \to \Gamma \) in \( WPG \), define \( \langle \phi \rangle : \text{Ty}(\Gamma) \to \text{Ty}(\Delta) \) by \( |A\{\phi\}| := |A| \) and \( A\{\phi\}(\delta_0) := A(\phi^\dagger \cdot \delta_0) \) for all \( A \in \text{Ty}(\Gamma) \) and \( \delta_0 \in WPG(!\Delta) \), and define \( \langle \phi \rangle_A : \text{Tm}(\Gamma, A) \to \text{Tm}(\Delta, A(\phi)) \) by \( \alpha\{\phi\} := \alpha \cdot \phi \) for all \( \alpha \in \text{Tm}(\Gamma, A) \).
Theorem 4.11 (A well-defined game-semantic CwF) The category \( \mathcal{WPG} \) together with the structures defined in Definition 4.10 gives rise to a CwF.

Proof We focus on substitution of terms, second projections and extensions since the other structures of \( \mathcal{WPG} \) are straightforward to check (e.g., the equational axioms on morphisms and terms in \( \mathcal{WPG} \) simply follow from Theorem 3.13). Let \( \Gamma, \Delta \in \mathcal{WPG}, A \in \mathcal{S}^w(\Gamma) \), \( \phi \in \mathcal{WPG}(\Delta, \Gamma) \), \( \alpha \in \mathcal{WPG}(\Gamma, A) \) and \( \check{\alpha} \in \mathcal{WPG}(\Delta, A\{\phi\}) \).

On \( \alpha\{\phi\} = \alpha \bullet \phi \in \mathcal{WPG}(\{\Delta\}, |A|) \),[12] we see by induction on the lengths of positions that \( (\alpha\{\phi\})_{\Pi(\Delta, A\{\phi\})} \equiv \Pi(\Delta, A\{\phi\})(\alpha\{\phi\}) \) follows from \( \phi \in \mathcal{WPG}(\Delta, \Gamma) \) and \( \alpha \in \mathcal{WPG}(\Gamma, A) \). The same induction also shows that \( \alpha\{\phi\} \cap \Pi(\Delta, A\{\phi\})(\alpha\{\phi\}) : \Pi(\Delta, A\{\phi\})(\alpha\{\phi\}) \) is winning and w.b., proving \( \alpha\{\phi\} \in \mathcal{WPG}(\Delta, A\{\phi\}) \).

The same induction proves \( v_A = \text{der}_{|A|} \in \mathcal{WPG}(\Sigma(\Gamma, A), A\{p_A\}) \), where the point is that the pi \( \Pi(\Gamma, A) \) and the sigma \( \Sigma(\Gamma, A) \) are defined by the same family \( A^* \).

Finally, on \( \langle \phi, \check{\alpha} \rangle \in \mathcal{WPG}(\{\Delta\}, |\Gamma| \& |A|) \), the same induction shows that \( \langle \phi, \check{\alpha} \rangle \in \mathcal{WPG}(\Delta, \Sigma(\Gamma, A)) \) follows from \( \phi \in \mathcal{WPG}(\Delta, \Gamma) \) and \( \check{\alpha} \in \mathcal{WPG}(\Delta, A\{\phi\}) \), where the fixed \( \phi \) in \( \langle \phi, \check{\alpha} \rangle \) plays a crucial role even if a play in \( \Pi(\Delta, \Sigma(\Gamma, A)) \) starts in \( A \).

It is notable that terms in the CwF \( \mathcal{WPG} \) are simply a certain class of strategies, and constructions on terms are those on strategies. This point is quite desirable since it means that the results/techniques on existing game semantics are also available for our game semantics of MLTT. It also implies that our game semantics inherits the advantages of existing game semantics (§1.2) such as intensionality (§4.6).

4.5 Game-semantic type formers

Nevertheless, CwFs only model the fragment of MLTT common to all types. Hence, in this section, we equip the CwF \( \mathcal{WPG} \) with semantic type formers [31] that model One-, Zero-, Pi-, Sigma- and Id-types.

4.5.1 Game semantics of Pi-types

We begin with \( \Pi \)-types. Recall first their semantic type former in an arbitrary CwF:

Definition 4.12 (CwFs with \( \Pi \)-types [31]) A CwF \( \mathcal{C} \) supports \( \Pi \)-types if

- \( (\Pi\text{-FORM}) \) Given \( \Gamma \in \mathcal{C}, A \in \text{Ty}(\Gamma) \) and \( B \in \text{Ty}(\Gamma, A) \), there is a type \( \Pi(\Gamma, A, B) \in \text{Ty}(\Gamma) \), where we also write \( A \Rightarrow B \) for \( \Pi(\Gamma, A, B) \) if \( B\{\text{id}_1, \alpha\} = B\{\text{id}_1, \alpha'\} \) \in \text{Ty}(\Gamma) \) for all \( \alpha, \alpha' \in \text{Ty}(\Gamma, A) \);
- \( (\Pi\text{-INTRO}) \) Given \( \beta \in \text{Ty}(\Gamma, A, B) \), there is a term \( \lambda_{A,B}(\beta) \in \text{Ty}(\Gamma, \Pi(\Gamma, A, B)) \);
- \( (\Pi\text{-ELIM}) \) Given \( \kappa \in \text{Ty}(\Gamma, \Pi(\Gamma, A, B)) \) and \( \alpha \in \text{Ty}(\Gamma, A) \), there is a term \( \text{App}_{A,B}(\kappa, \alpha) \in \text{Ty}(\Gamma, B\{\overline{\alpha}\}) \), where \( \overline{\alpha} := \langle \text{id}_1, \alpha \rangle \) \in \Gamma \Rightarrow \Gamma, A \);
- \( (\Pi\text{-COMP}) \) \( \text{App}_{A,B}(\lambda_{A,B}(\beta), \alpha) = \beta\{\overline{\alpha}\} \);

[12]: The following argument is based on the proof of Lemma 3.11 on composition.
• (II-SUBST) Given $\Delta \in C$ and $\phi : \Delta \to \Gamma$ in $C$, $\Pi(A, B)\{\phi\} = \Pi(A\{\phi\}, B\{\phi_A^+\})$, where $\phi_A^+ := (\phi \circ p_A(\phi), v_A(\phi))_A : \Delta.A\{\phi\} \to \Gamma.A$.

• (\lambda-SUBST) $\lambda_{A, B}(\beta)(\phi) = \lambda_{A\{\phi\}, B\{\phi_A^+\}}(\beta(\phi_A^+)) \in \text{Tm}(\Delta, \Pi(A\{\phi\}, B\{\phi_A^+\}))$;

• (APP-SUBST) $\text{App}_{A, B}(\kappa, \alpha)(\phi) = \text{App}_{A\{\phi\}, B\{\phi_A^+\}}(\kappa\{\phi\}, \alpha\{\phi\}) \in \text{Tm}(\Delta, B(\overline{\alpha}))\{\phi\}$.

Furthermore, $C$ strictly supports Pi-types if it additionally satisfies

• (\lambda-UNIQ) $\lambda_{A, B} \circ \text{App}_{A\{p_A\}, B\{(p_A^+\}_{A\{p_A\})}}(\kappa\{p_A\}, v_A) = \kappa$.

Pi-types (with $\eta$-rule) are modelled in a CwF that (strictly) supports Pi-types [31]. Let us now proceed to show that the CwF WPG strictly supports Pi-types.

**Lemma 4.13 (Currying)** Given $\Gamma \in \text{WPG}$, $A \in \mathcal{P}^w(\Gamma)$ and $B \in \mathcal{P}^w(\Sigma(\Gamma, A))$, there is a bijection $\lambda_{A, B} : \text{WPG}(\Sigma(\Gamma, A), B) \xrightarrow{\sim} \text{WPG}(\Gamma, \Pi(A, B))$, where $\Pi(A, B) \in \mathcal{P}^w(\Gamma)$ is defined by $\Pi(A, B) := |A| \Rightarrow |B|$ and $\Pi(A, B)(\gamma^0) := \Pi(A\{\gamma^0\}, B_{\gamma^0})$ for all $\gamma^0 \in \text{WPG}(\Gamma)$, and $B_{\gamma^0} \in \mathcal{P}^w(A\{\gamma^0\})$ by $|B_{\gamma^0}| := |B|$ and $B_{\gamma^0}(\alpha^0) := B(\langle \gamma^0, \alpha^0 \rangle)$ for all $\alpha^0 \in \text{WPG}(\Gamma(A\{\gamma^0\}))$.

**Proof** Let $\phi \in \text{WPG}(\Sigma(\Gamma, A), B)$; we first construct $\phi' \in \text{WPG}(\Gamma, \Pi(A, B))$ from $\phi$ as follows. By the isomorphism $|!\Sigma(\Gamma, A)| = |!|\Gamma| & |A| | \cong !|\Gamma| \otimes !|A|$, we adjust ‘tags’ or curry $\phi$ with respect to the adjunction between tensor $\otimes$ and linear implication $\to$ [35], obtaining another strategy $\phi' : \Gamma \Rightarrow (|A| \Rightarrow |B|)$.

Next, let us verify $\mathcal{P}_{\Pi(\Gamma, \Pi(A, B))} \leq \Pi(\Gamma, \Pi(A, B))(\phi')$ by induction on the length of positions, where again the only nontrivial case is the inductive step on even-length positions. This inductive step goes through by $\phi \in \text{WPG}(\Sigma(\Gamma, A), B)$ thanks to the correspondence between the constraints on $\Pi(\Sigma(\Gamma, A), B)$ of $\phi$ and $\Pi(\Gamma, \Pi(A, B))(\phi')$.

The same argument also shows that $\phi' \cap \Pi(\Gamma, \Pi(A, B))(\phi') : \Gamma \Rightarrow (|A| \Rightarrow |B|)$ is winning and w.b., and hence we have proven $\lambda_{A, B}(\phi) := \phi' \in \text{WPG}(\Gamma, \Pi(A, B))$.

Finally, the inverse of this construction $\lambda_{A, B}$ is given in the same vein.

**Example 4.14** Define $N^+_b := \mathcal{P}^w(\Sigma(N, N_b))$ by $|N^+_b| := N$ and $N^+_b(\langle k, n \rangle^+) := N_b(k+n)$ for all $k, n \in \mathbb{N}$ such that $k \geq n$. Then, Lemma 4.13 allows us to obtain $\lambda_{N_b, N^+_b}(\text{der}_N) \in \text{WPG}(N, \Pi(N_b, N^+_b))$ from $\text{der}_N = v_{N^+_b} \in \text{WPG}(\Sigma(N, N_b, N^+_b))$.

**Theorem 4.15 (Game semantics of Pi-types)** WPG strictly supports Pi-types.

**Proof** Let $\Gamma \in \text{WPG}$, $A \in \mathcal{P}^w(\Gamma)$, $B \in \mathcal{P}^w(\Sigma(\Gamma, A))$ and $\beta \in \text{WPG}(\Sigma(\Gamma, A), B)$.

• (II-FORM) $\Pi(A, B) \in \mathcal{P}^w(\Gamma)$ is defined in Lemma 4.13.

• (II-INTRO) By Lemma 4.13, we obtain $\lambda_{A, B}(\beta) \in \text{WPG}(\Gamma, \Pi(A, B))$. We often omit the subscripts $(\_)_{A, B}$ on $\lambda_{A, B}$ and the inverse $\lambda_{A, B}^{-1}$.

• (II-ELIM) $\text{App}_{A, B}(\kappa, \alpha) := \lambda_{A, B}^{-1}(\kappa \{\alpha\})$ for all $\kappa \in \text{WPG}(\Gamma, \Pi(A, B))$ and $\alpha \in \text{WPG}(\Gamma, A)$. We indeed have $\text{App}_{A, B}(\kappa, \alpha) = \lambda_{A, B}^{-1}(\kappa \{\alpha\}) \in \text{WPG}(\Gamma, B(\{\alpha\})$ by Theorem 4.11. We often omit the subscripts $(\_)_{A, B}$ on $\text{App}_{A, B}$.

• (II-COMP) $\text{App}_{A, B}(\lambda_{A, B}(\beta), \alpha) = \lambda_{A, B}^{-1}(\lambda_{A, B}(\beta) \{\alpha\}) = \beta(\{\alpha\})$.

• (II-SUBST) Let $\Delta \in \text{WPG}$ and $\phi \in \text{WPG}(\Delta, \Gamma)$. For the first components, we have $|\Pi(A, B)\{\phi\}| = |\Pi(A, B)| = |A| \Rightarrow |B| = |A\{\phi\}| \Rightarrow |B\{\phi_A^+\}| = \ldots$
\[ |\Pi(A(\phi), B(\phi^+))| \]. For the second components, we have

\[
\Pi(A, B)\{\phi\} = (\Pi(A(\phi^1 \circ \delta^1_0), B(\phi^+))_{\delta^1_0}^{\text{wpg}(\Delta)}
\]

\[
= (\Pi(A)\{\delta^1_0\}, B(\phi^+))_{\delta^1_0}^{\text{wpg}(\Delta)}
\]

\[
= \Pi(A)\{\phi\}, B(\phi^+)
\]

where the second equation holds since for all \(\phi^1_0 \in \text{wpg}(A(\phi^1 \circ \delta^1_0))\) we have

\[
B(\phi^+_{\delta^1_0}) = B(\phi^+_{\delta^1_0})((\delta_0, \hat{\delta}_0))^1
\]

\[
= B((\phi \cdot p\{A\{\phi\}, v\{A\{\phi\}\})) \cdot (\delta_0, \hat{\delta}_0)^1
\]

\[
= B((\phi \cdot \delta_0, \hat{\delta}_0)^1)
\]

\[
= B_{\phi^1 \circ \delta^1_0}(\hat{\delta}_0).
\]

We have shown \(\|\Pi(A, B)\{\phi\}\| = \|\Pi(A)\{\phi\}, B(\phi^+))\|^1\) as well.

- (\(\lambda\)-Subst) By the definition of \(\lambda\), we have \(\lambda_{A,B}(\beta)\{\phi\} = \lambda_{A(\phi), B(\phi^+)}(\beta^1_{\phi^+})\).

- (App-Subst) We have

\[
\text{App}_{A,B}(\kappa, \alpha)\{\phi\} = \lambda_{A,B}^{-1}(\kappa, \alpha) \cdot \phi
\]

\[
= \lambda_{A,B}^{-1}(\kappa, \alpha) \cdot \phi
\]

\[
= \lambda_{\phi_B}^{-1}(\kappa, \alpha) \cdot \phi
\]

\[
= \lambda_{A,B}^{-1}(\phi_B) \cdot \alpha(\phi)
\]

\[
= \lambda_{\phi_B}^{-1}(\kappa, \alpha) \cdot \alpha(\phi)
\]

(by \(\lambda\)-Subst)

\[
= \text{App}_{A(\phi), B(\phi^+)}(\kappa, \alpha) \cdot \phi).
\]

- (\(\lambda\)-Uniq) We have

\[
\lambda_{A,B}(\text{App}_{A(pA), B(pA^+)}(\kappa\{pA\}, vA)) = \lambda_{A,B}(\lambda_{A(pA), B(pA^+)}(\kappa\{pA\}))\{vA\})
\]

\[
= \lambda_{A,B}(\lambda_{A(pA), B(pA^+)}(\kappa\{pA\}))\{vA\})
\]

(by \(\lambda\)-Subst)

\[
= \lambda_{A,B}(\lambda_{A(pA), B(pA^+)}(\kappa\{pA\}))\{vA\})
\]

\[
= \lambda_{A,B}(\lambda_{A(pA), B(pA^+)}(\kappa\{pA\}))
\]

\[
= \lambda_{A,B}(\lambda_{A(pA), B(pA^+)}(\kappa\{pA\}))
\]

\[
= \lambda_{A,B}(\lambda_{A(\phi), B(\phi^+)})(\kappa\{\phi\}, \alpha(\phi))
\]

which completes the proof. \(\square\)

### 4.5.2 Game semantics of Sigma-types

Next, we consider \(\Sigma\)-types. Again, let us first recall their general semantic type former in an arbitrary CwF:

**Definition 4.16** (CwFs with \(\Sigma\)-types [31]) A CwF \(C\) *supports \(\Sigma\)-types* if

- (\(\Sigma\)-Form) Given \(\Gamma \in C\), \(A \in \text{Ty}(\Gamma)\) and \(B \in \text{Ty}(\Gamma, A)\), there is a type \(\Sigma(A, B) \in \text{Ty}(\Gamma)\), where we also write \(A \times B\) for \(\Sigma(A, B)\) if \(B(\text{id}_\Gamma, \alpha)) = B(\{\text{id}_\Gamma, \alpha^\prime\}) \in \text{Ty}(\Gamma)\) for all \(\alpha, \alpha^\prime \in \text{Tm}(\Gamma, A)\);
• (Σ-INTRO) There is a morphism Pair_{A,B} : Γ.A.B → Γ.Σ(A, B) in C;
• (Σ-ELIM) Given P ∈ Ty(Γ.Σ(A, B)) and ρ ∈ Tm(Γ.A.B, P{Pair_{A,B}}), there is a term \( R^\Sigma_{A,B,P}(\rho) \in Tm(Γ.Σ(A, B), P) \);
• (Σ-COMP) \( R^\Sigma_{A,B,P}(\rho) \{Pair_{A,B}\} = ρ \);
• (Σ-SUBST) Given Δ ∈ C and φ : Δ → Γ in C, \( Σ(A, B)\{φ\} = Σ(A(φ), B(φ^+_A)) \), where \( φ^+_A := φ \circ p_A(φ), v_A(φ) : Δ.A(φ) → Γ.A \);
• (PAIR-SUBST) \( p_{Σ(A,B)} \circ Pair_{A,B} = p_A \circ p_B \) and \( φ^+_Σ(A,B) \circ Pair_{A,B}(φ^+_A) = Pair_{A,B} \circ φ^+_Σ(A,B) \), where \( φ^+_Σ(A,B) := φ^+_A : Δ.A(φ).B(φ^+_A) → Σ(A,B) \);
• (RΣ-SUBST) \( R^\Sigma_{A,B,P}(p) \{φ^+_Σ(A,B)\} = R^Σ_{A(φ),B(φ^+_A)}(p(φ^+_Σ(A,B))) \).

In addition, C strictly supports Sigma-types if it also satisfies
• (RΣ-UNIQ) \( \bar{ρ} = R^\Sigma_{A,B,P}(ρ) \) if \( \bar{ρ} ∈ Tm(Γ.Σ(A, B), P) \) and \( \bar{ρ} \{Pair_{A,B}\} = ρ \).

Sigma-types (with η-rule) are modelled in a CwF that (strictly) supports Sigma-types [31]. Now, we show that our CwF WPG strictly supports Sigma-types:

**Theorem 4.17** (Game semantics of Sigma-types) **WPG strictly supports Sigma-types.**

**Proof** Let \( Γ ∈ WPG, A ∈ D^w(Γ), B ∈ D^w(Σ(Γ, A)) \) and \( P ∈ D^w(Σ(Γ, Σ(A, B))) \).

• (Σ-FORM) Similarly to pi II, \( Σ(A, B) := (|A| ≤ |B|, (Σ(A(γ^+_A)), B_{γ^+_A})) \) ∈ WPG(Γ).

• (Σ-INTRO) By the evident bijection \( Σ(Σ(Γ, A), B) ≃ Σ(Γ, Σ(A, B)) \), define \( Pair_{A,B} : Σ(Σ(Γ, A), B) → Σ(Γ, Σ(A, B)) \) in WPG to be the deleruction up to ‘tags,’ i.e., \( Pair_{A,B} := \{p_A \bullet p_B, (v_A\{p_B\}, v_B)\} \). Note that the inverse \( Pair^{-1}_{A,B} \) is \( \{⟨Σ_{Σ(Γ,A),B}, ω^A_1, ω^A_2⟩, Σ_{Σ(Γ,A),B} → A\{Σ_{Σ(Γ,A),B}\}\) and \( ω^A_1, ω^A_2 : Σ(Γ, Σ(A, B)) → B\{p_{Σ(Γ,A),B}, ω^A_{Σ(Γ,A),B}\} \) are the deleructions up to ‘tags.’

• (Σ-ELIM) Given \( ρ ∈ WPG(Σ(Γ, A), B), P\{Pair_{A,B}\} \), define \( R^Σ_{A,B,P}(ρ) := ρ\{Pair^{-1}_{A,B}\} ∈ WPG(Σ(Γ, Σ(A, B)), P\{Pair_{A,B}\}) \).

• (Σ-COMP) We have

\[
R^Σ_{A,B,P}(ρ) \{Pair_{A,B}\} = ρ \{Pair^{-1}_{A,B}\} \{Pair_{A,B}\}
\]
\[
= ρ \{Pair^{-1}_{A,B} \bullet Pair_{A,B}\}
\]
\[
= ρ \{idΣ(Σ(Γ,A),B)\}
\]
\[
= ρ.
\]

• (Σ-SUBST) Similar to the case of pi II (Theorem 4.15).

• (PAIR-SUBST) \( p_{Σ(A,B)} \bullet Pair_{A,B} = p_{Σ(A,B)} \bullet (p_A \bullet p_B, (v_A\{p_B\}, v_B)) = p_A \bullet p_B \)

and

\[
φ^+_Σ(A,B) \bullet Pair_{A(φ),B(φ^+_A)} = (φ \bullet p_{Σ(A,B)}(φ), v_{Σ(A,B)}(φ)) \bullet Pair_{A(φ),B(φ^+_A)}
\]
\[
= (φ \bullet p_{Σ(A,B)}(φ, v_{Σ(A,B)}(φ)) \bullet Pair_{A(φ),B(φ^+_A)}(Pair_{A(φ),B(φ^+_A)}))
\]
\[
= (φ \bullet (p_A(φ), p_B(φ^+_A)), v_{Σ(A,B)}(Pair_{A(φ),B(φ^+_A)}))
\]
\[
= (φ \bullet (p_A(φ), (v_A(p_B(φ^+_A), v_B(φ^+_A)) \bullet (φ \bullet p_{Σ(Γ,A),B}(φ), v_{Σ(Γ,A),B}(φ))
\]
\[
= (φ \bullet (p_A(φ), (v_A(p_B(φ^+_A), v_B(φ^+_A)) \bullet (φ \bullet p_{Σ(Γ,A),B}(φ), v_{Σ(Γ,A),B}(φ))
\]
\[
= Pair_{A,B} \bullet (φ^+_Σ(A,B), v_{Σ(Γ,A),B}(φ^+_A))
\]
\[
= Pair_{A,B} \bullet φ^+_Σ(A,B),
\]
• \((\mathcal{R}^\Sigma\text{-SUBST})\) We have
\[
\mathcal{R}_{A,B,P}^\Sigma(\phi|_{\Sigma(A,B)}) = \rho(\text{Pair}^{-1}_{\Sigma}) \cdot (\phi \cdot \text{Pair}_{\Sigma(A,B)}(\cdot, \cdot))
\]
\[
= \rho((\text{Pair}_{\Sigma(A,B)}(\cdot, \cdot)) \cdot \text{Pair}^{-1}_{\Sigma}) \cdot (\phi \cdot \text{Pair}_{\Sigma(A,B)}(\cdot, \cdot))
\]
\[
= \rho((\text{Pair}_{\Sigma(A,B)}(\cdot, \cdot)) \cdot \text{Pair}^{-1}_{\Sigma}) \cdot (\phi \cdot \text{Pair}_{\Sigma(A,B)}(\cdot, \cdot))
\]
\[
= \rho((\text{Pair}_{\Sigma(A,B)}(\cdot, \cdot)) \cdot \text{Pair}^{-1}_{\Sigma}) \cdot (\phi \cdot \text{Pair}_{\Sigma(A,B)}(\cdot, \cdot))
\]
\[
= \mathcal{R}_{A,B,P}^\Sigma(\phi|_{\Sigma(A,B)})
\]

• \((\mathcal{R}^\Sigma\text{-UNIQ})\) Given \(\rho \in \mathcal{WF}_G(\Sigma(\Gamma, \Sigma(A,B)), P)\) with \(\rho(\text{Pair}_{A,B}) = \rho\), we have
\[
\rho = \rho(\text{id}_{\Sigma(\Gamma, \Sigma(A,B))}) \cdot \rho(\text{Pair}_{A,B}) \cdot (\text{Pair}_{\Sigma(A,B)}^{-1}) \cdot \rho \cdot \text{Pair}_{\Sigma(A,B)\cdot P} = \mathcal{R}_{A,B,P}^\Sigma(\rho),
\]
which completes the proof. \(\square\)

### 4.5.3 Game semantics of N-type

We next present our game semantics of \(N\text{-type}\). Again, we first recall the general semantic type former for \(N\text{-type}\) in an arbitrary CwF:

**Definition 4.18 (CwFs with N-type [31])** A CwF \(C\) supports \(N\text{-type}\) if

- \((N\text{-FORM})\) Given \(\Gamma \in C\), there is a type \(N[\Gamma] \in \text{Ty}(\Gamma)\). We abbreviate it as \(N\).
- \((N\text{-INTRO})\) There are a term \(\overline{\rho}_r \in \text{Ty}(\Gamma, N)\) and a morphism \(\text{succ} : \Gamma.N \rightarrow \Gamma.N\) in \(C\) that satisfy for any morphisms \(\phi : \Delta \rightarrow \Gamma\) and \(\psi : \Delta.N \rightarrow \Gamma\) in \(C\)

\[
\overline{\rho}_r(\phi) = \overline{\rho}_\Delta \in \text{Ty}(\Delta, N) \quad \quad \text{p}_N \circ \text{succ} = \text{p}_N : \Gamma.N \rightarrow \Gamma.
\]
\[
\text{succ} \circ \langle \psi, \text{v}_N \rangle = \langle \psi, \text{v}_N \{\text{succ} \Delta \} \rangle : \Delta.N \rightarrow \Gamma.N,
\]

where the last equation makes sense since \(\text{succ} \circ \langle \psi, \text{v}_N \rangle, \langle \psi, \text{v}_N \{\text{succ} \Delta \} \rangle : \Delta.N \rightarrow \Gamma.N\) by \(N\text{-Subst}\) given below. We henceforth skip the same remark.

**Notation** Define \(\text{zero} := \langle \text{id}_\Gamma, \overline{\rho}_0 \rangle : \Gamma \rightarrow \Gamma.N\) for each \(\Gamma \in C\); it satisfies \(\text{zero} \circ \phi = \langle \phi, \overline{\rho}_\Delta \rangle : \Delta \rightarrow \Gamma.N\) for any \(\phi : \Delta \rightarrow \Gamma\) in \(C\). We often omit the subscript \((\cdot)_\Gamma\) on \(\overline{\rho}_r\) zero and succ. Define \(\overline{\rho}_r \in \text{Ty}(\Gamma, N)\) for each \(n \in \mathbb{N}\) by: \(\overline{\rho}_r\) is already given, and \(\overline{\rho}_{r+1} := \text{v}_N \{\text{succ} \circ (\text{id}_\Gamma, \overline{\rho}_r)\} : \Delta.N \rightarrow \Gamma.N\).

- \((N\text{-ELIM})\) Given a type \(P \in \text{Ty}(\Gamma, N)\), and terms \(c_0 \in \text{Ty}(\Gamma, P\{\text{zero}\})\) and \(c_1 \in \text{Ty}(\Gamma, P\{\text{succ} \circ \text{p}_P\})\), there is a term \(\mathcal{R}_P^N(c_0, c_1) \in \text{Ty}(\Gamma, N, P)\);
- \((N\text{-COMP})\) We have

\[
\mathcal{R}_P^N(c_0, c_1)\{\text{zero}\} = c_0 \in \text{Ty}(\Gamma, P\{\text{zero}\})\;
\]
\[
\mathcal{R}_P^N(c_0, c_1)\{\text{succ}\} = c_1(\text{id}_\Gamma, \mathcal{R}_P^N(c_0, c_1)\{\text{p}\}) \in \text{Ty}(\Gamma, N, P\{\text{succ}\});
\]
- \((N\text{-SUBST})\) \(N[\Gamma] \{\phi\} = N[\Delta] \in \text{Ty}(\Delta);\)
\( (\mathcal{R}^N.\text{-}\text{SUBST}) \mathcal{R}^N_P(c_z, c_s)\{\phi^+_N\} = \mathcal{R}^N_{P(\phi^+_N)}(c_z(\phi), c_s\{\phi^+_N,p\}) \in Tm(\Delta.N, P\{\phi^+_N\}) \)

where \( \phi^+_N := (\phi \circ p_N, v_N)_N : \Delta.N \rightarrow \Gamma.N \) and \( \phi^{++}_N, p := (\phi^+_N)_P : \Delta.N.P\{\phi^+_N\} \rightarrow \Gamma.N.P \).

Let us now present our game semantics of N-type, which is based on the standard game semantics of PCF [22]:

**Theorem 4.19 (Game semantics of N-type)** \( \text{WPG supports N-type.} \)

**Proof** Let \( \Gamma, \Delta \in \text{WPG}, \phi \in \text{WPG}(\Delta, \Gamma) \) and \( \psi \in \text{WPG}(\Sigma(\Delta, \{N\}!\Delta), \Gamma) \).

- (N-FORM) \( \mathcal{N}[\mathcal{F}] \) is the constant one \( \{N\}_\Gamma \in \mathcal{O}(\Gamma) \) at \( N \) (Example 3.4).

- (N-INTRO) \( \mathcal{N}_\Gamma \in \text{WPG}(\Gamma, \{N\}) \) is \( \mathcal{Q} : N \) (Examples 2.8 and 3.4) up to ‘tags,’ and \( \text{succ}_\Gamma := (p, \text{sc}_\Gamma) \in \text{WPG}(\Sigma(\Gamma, \{N\}), \Sigma(\Gamma, \{N\})) \), where we define \( \text{sc}_\Gamma := \text{Pref}(\{q_1[q_0[n_0[n_1 + 1]]| n \in \mathbb{N}\})^\text{Even} \in \text{WPG}(\Sigma(\Gamma, \{N[0]\}), \{N[1]\}) \).

Clearly, we have \( \mathcal{N}_\Gamma \cdot \phi = \mathcal{Q}_\Delta \) and \( \text{sc}_\Gamma \cdot \langle \psi, v_{(\Gamma)\Delta} \rangle = \text{sc}_\Delta = v_{(N)\Delta} \{\text{sc}_\Delta\} \), and therefore the required three equations hold.

- (N-ELIM) Given \( P \in \mathcal{O}(\Sigma(\Gamma, \{N\})), c_z \in \text{WPG}(\Gamma, P\{\text{zero}\}) \) and \( c_s \in \text{WPG}(\Sigma(\Gamma, \{N\}), P), P\{\text{succ}\cdot p_P\} \), there are two terms

\[
\bar{c}_z \in \text{WPG}(\Sigma(\Pi(\Sigma(\Gamma, \{N\}), P), \Sigma(\Gamma, \{N\}))), P\{\text{zero}\cdot p \cdot v\};
\]

\[
\bar{c}_s \in \text{WPG}(\Sigma(\Pi(\Sigma(\Gamma, \{N\}), P), \Sigma(\Gamma, \{N\}))), P\{\text{succ}\cdot v\},
\]

where note that \( v \in \text{WPG}(\Sigma(\Pi(\Sigma(\Gamma, \{N\}), P), \Sigma(\Gamma, \{N\}))), \Sigma(\Gamma, \{N\})\{p\} \) is just \( \pi_2 \in \text{WPG}(\Pi(\Gamma \& N), P) \& (\Gamma \& N), \Gamma \& N) \), defined respectively by

\[
\bar{c}_z := (\Pi(\Sigma(\Gamma, \{N\}), P) \& \Sigma(\Gamma, \{N\}) \xrightarrow{v} \Sigma(\Gamma, \{N\}) \xrightarrow{p} \Gamma \xrightarrow{c_z} \{P(\text{zero})\});
\]

\[
\bar{c}_s := (\Pi(\Sigma(\Gamma, \{N\}), P) \& \Sigma(\Gamma, \{N\}) \xrightarrow{(v, \text{ev}_P)} \Sigma(\Gamma, \{N\}) \& \{P\} \xrightarrow{c_s} \{P(\text{succ}\cdot p)\}.
\]

where \( \text{ev}_P \in \text{WPG}(\Sigma(\Pi(\Sigma(\Gamma, \{N\}), P), \Sigma(\Gamma, \{N\}))) \), \( P\{v\} \) is the evaluation [16], i.e., \( \text{ev}_P := \lambda^{-1}(\text{der}_{\Sigma(\Pi(\Sigma(\Gamma, \{N\}), p))}). \)

Next, we define \( \text{pred}_\Gamma := (p, \text{pd}_\Gamma) \in \text{WPG}(\Sigma(\Gamma, \{N\}), \Sigma(\Gamma, \{N\})) \), where

\[
\text{pd}_\Gamma := \text{Pref}(\{q_1[q_0[n_0[n_1 + 1]]| n \in \mathbb{N}\})^\text{Even} \in \text{WPG}(\Sigma(\Gamma, \{N[0]\}), \{N[1]\}) \), and often omit the subscript \( \langle\_\rangle \) on pred. Besides, let \( \text{cond}_P \in \text{WPG}(\Sigma(\Sigma(\Sigma(\Gamma, \{N\}), P\{\text{zero}\cdot p\}), P\{\text{succ}\cdot \text{pred}\cdot p\}), P\{p\cdot p\}) \)

be the game semantics of conditionals [33, 34, 22]. Given any initial move, it asks the number \( n \) on \( N \), and plays as the dereliction between \( P\{\text{zero}\cdot p\} \) and \( P\{p\cdot p\} \) if \( n = 0 \), and between \( P\{\text{succ}\cdot \text{pred}\cdot p\} \) and \( P\{p\cdot p\} \) otherwise. We then define \( \mathcal{F}_P^N(c_z, c_s) \in \text{WPG}(\Pi(\Sigma(\Gamma, \{N\}), P), \Pi(\Sigma(\Gamma, \{N\}), P)) \) by

\[
\mathcal{F}_P^N(c_z, c_s) := \lambda_{(\Sigma(\Gamma, \{N\})), \{P(v)\}}(\text{cond}_P\{(\langle v, \bar{c}_s \rangle, \bar{c}_s\{\langle p, \text{pred}\cdot v \rangle\})\})), \tag{3}
\]

which is well-defined thanks to the equation \( (\text{succ}\cdot v)\cdot (p, \text{pred}\cdot v) = \text{succ}\cdot \text{pred}\cdot v = (\text{succ}\cdot \text{pred}\cdot p)\cdot (v, \bar{c}_z) \).

Finally, let \( \mathcal{R}_P^N(c_z, c_s) \in \text{WPG}(\Sigma(\Gamma, \{N\}), P) \) be the least upper bound of the chain \( (\mathcal{R}_P^N(c_z, c_s)_n : \Pi(\Sigma(\Gamma, \{N\}), P))_{n \in \mathbb{N}} \) given by

\[
\mathcal{R}_P^N(c_z, c_s)_0 := \{\epsilon\} \quad \mathcal{R}_P^N(c_z, c_s)_{n+1} := \mathcal{F}_P^N(c_z, c_s) \cdot \mathcal{R}_P^N(c_z, c_s)_n.
\]
The difference from the game semantics of fixed-point combinator \([33, 34, 22]\) is that the strategy pred occurring in \(F_P^N(c_x, c_s)\) (3) decreases the parameter number on \(N\) at every recursive call by \(F_P^N(c_x, c_s)\), so that \(R_P^N(c_x, c_s)\) is total.

- \((N\text{-COMP})\) By the definition of \(R_P^N(c_x, c_s)\), we clearly have

\[
R_P^N(c_x, c_s)\{\text{zero}\} = c_x \in \text{WPG}(\Gamma, P\{\text{zero}\});
\]

\[
R_P^N(c_x, c_s)\{\text{succ}\} = c_s(\{\text{der}_{\Sigma(\Gamma, \{N\})}, R_P^N(c_x, c_s)\}) \in \text{WPG}(\Sigma(\Gamma, \{N\}), P\{\text{succ}\}).
\]

- \((N\text{-SUBST})\) It is clear that \(\{N\}!\{\phi\} = \{N\}_!\Delta\) holds.
- \((R^N\text{-SUBST})\) Finally, by the definition of \(R_P^N(c_x, c_s)\), we clearly have

\[
R_P^N(c_x, c_s)\{\phi_N^+\} = R_P^N(c_x, c_s)\{\phi_N^+\}(c_x, c_s;\phi_N^+; p) \in \text{WPG}(\Sigma(\Gamma, \{\phi_N\}), P\{\phi\}).
\]

(or alternatively show \(R_P^N(c_x, c_s)\{\phi_N^+\} = R_P^N(c_x, c_s)\{\phi_N^+\}(c_x, c_s;\phi_N^+; p)\); for all \(n \in \mathbb{N}\) by induction on \(n\) so that the above equation holds), which completes the proof.  

\[4.5.4\text{ Game semantics of One- and Zero-types}\]

Next, we interpret One- and Zero-types. Since it is trivial to interpret these types, we only sketch the proof. See \([31]\) for the semantic type formers for these types.

**Theorem 4.20** (Game semantics of One- and Zero-types) \(\text{WPG}\) supports the semantic type formers for One- (in the strict sense) and Zero-types.

**Proof (sketch)** We model One- and Zero-types by the constant dependent p-games at the terminal p-game \(T\) and the empty p-game \(0\), respectively (Example 3.4).

The only point on One-type is that there is only the trivial one \(\{\epsilon\} \in \text{WPG}(\Gamma, \{T\})\) for any \(\Gamma \in \text{WPG}\), so that we can interpret all the rules of One-type.

Similarly, given \(A \in \mathcal{G}(\Sigma(\Gamma, \{0\}))\) and \(\zeta \in \text{WPG}(\Gamma, \{0\})\), the only point on Zero-type is that for the elimination rule we may obtain \(R_A^0(\zeta) \in \text{WPG}(\Gamma, A\{\text{id}_T, \zeta\})\) from \(\zeta\) by case distinction: Let \(R_A^0(\zeta)\) be the one obtained from \(\zeta\) by replacing the unique first move \(q\) of \(0\) in \(\zeta\) with those of \(|A|\) if any, and \(R_A^0(\zeta) := \{\epsilon\}\) otherwise.  

\[4.5.5\text{ Game semantics of Id-types}\]

Let us proceed to present our game semantics of Id-types. Again, we first review the general semantic type former for Id-types in an arbitrary CwF:

**Definition 4.21** (CwFs with Id-types [31]) A CwF \(C\) supports Id-types if

- \((\text{Id-Form})\) Given \(\Gamma \in C\) and \(A \in \text{Ty}(\Gamma)\), there is a type \(\text{Id}_A \in \text{Ty}(\Gamma.A.A^+)\), where \(A^+ := A\{p_A\} \in \text{Ty}(\Gamma.A)\);
- \((\text{Id-Intro})\) There is a morphism \(\text{Refl}_A : \Gamma.A \to \Gamma.A.A^+.\text{Id}_A\) in \(C\) that satisfies \(p_{\text{Id}_A} \circ \text{Refl}_A = \nu_A : \Gamma.A \to \Gamma.A.A^+,\) where \(\nu_A := \text{id}_{\Gamma.A, v_A}\);
- \((\text{Id-Elim})\) Given \(B \in \text{Ty}(\Gamma.A.A^+.\text{Id}_A)\) and \(\beta \in \text{Tm}(\Gamma.A, B\{\text{Refl}_A\})\), there is a term \(\text{R}_{\text{Id}}^A_{\beta}(\beta) \in \text{Tm}(\Gamma.A.A^+.\text{Id}_A, B);\)
- \((\text{Id-Comp})\) \(\text{R}_{\text{Id}}^A_{\beta}(\beta)\{\text{Refl}_A\} = \beta;\)
- \((\text{Id-Subst})\) \(\text{Id}_A(\phi_{\Delta,A^+}) = \text{Id}_A(\phi) \in \text{Ty}(\Delta.A\{\phi\}.A\{\phi\}^+)\) for all \(\Delta \in C\) and \(\phi : \Delta \to \Gamma\) in \(C\), where \(\phi^+ := \langle \phi \circ p_{A(\phi)}, v_{A(\phi)} \rangle_A : \Delta.A\{\phi\} \to \Gamma.A\) and \(\phi_{\Delta,A^+}^+ := (\phi_A^+)_{\Delta,A^+}^+ : \Delta.A\{\phi\}.A^+\{\phi_A^+\} \to \Gamma.A.A^+;\)
Then, we present our game semantics of Id-types, which is essentially the same as the interpretation of Id-types by Abramsky et al. [23, 29]:

**Theorem 4.22 (Game semantics of Id-types)** \( \text{WPG supports Id-types.} \)

**Proof** Let \( \Gamma \in \text{WPG}, A \in \mathcal{D}^w(\Gamma) \) and \( B \in \mathcal{D}^w(\Sigma(\Sigma(\Sigma(\Gamma, A), A^+), \text{Id}_A)) \).

- (ID-FORM) Let 1 be the flat game flat(\( \{ \sqrt{\cdot} \} \)) (Example 2.8), where \( \{ \sqrt{\cdot} \} \) is an arbitrarily fixed singleton set. Define \( \text{Id}_A \in \mathcal{D}^w(\Sigma(\Sigma(\Gamma, A), A^+)) \) by \( \text{Id}_A := 1 \) and \( \text{Id}_A((\langle \gamma_0, \alpha_0 \rangle, \alpha'_0)^\dagger) := \begin{cases} (1, \kappa_1) & \text{if } \alpha_0 = \alpha'_0; \\ (1, \kappa_0) & \text{otherwise,} \end{cases} \) \( \text{WPG}(!\Sigma(\Sigma(\Gamma, A), A^+)), \) where \( \kappa_X \) is the constant family at \( X = 1, 0; \)

- (ID-INTRO) Define \( \text{Refl}_A := (\nabla A, \text{refl}_A) \in \text{WPG}(\Sigma(\Sigma(\Gamma, A), \Sigma(\Sigma(\Sigma(\Gamma, A), A^+), \text{Id}_A))), \) where \( \text{refl}_A \in \text{WPG}(\Sigma(\Sigma(\Gamma, A), \text{Id}_A(\nabla A))) \) is \( \sqrt{1} \) (Example 2.8) up to 'tags.' Note that \( \text{refl}_A \) is well-defined since its codomain is always the game 1.

- (ID-ELIM) Given \( \beta \in \text{WPG}(\Sigma(\Gamma, A), B\{\text{Refl}_A\}), \) let us construct the strategy \( \mathcal{R}^\text{Id}_{A,B}(\beta) \in \text{WPG}(\Sigma(\Sigma(\Sigma(\Gamma, A_{[0]}), A^+_{[1]}), \text{Id}_A), B) \) such that given any initial move in \( B \) it makes the move \( q \) in \( \text{Id}_A \), and if \( O \) plays there by \( \sqrt{\cdot} \), then it plays as \( \beta \) between \( \Sigma(\Gamma, A_{[0]}) \) and \( B \) in the rest of the play. Since there is the evident bijection \( \text{WPG}(\Sigma(\Gamma, A)) \cong \text{WPG}(\Sigma(\Sigma(\Sigma(\Gamma, A), A^+), \text{Id}_A)) \) thanks to the definition of \( \text{Id}_A \), \( \mathcal{R}^\text{Id}_{A,B}(\beta) \) can play as \( \beta \) after the initial two moves.

Having gone through the detailed proofs of Theorems 4.15 and 4.17, it is a routine to verify Id-Comp, Id-Subst and Refl-Subst. Also, essentially the same interpretation of Id-types is given by Abramsky et al. [23, 29]. Hence, we leave it to the reader. \( \square \)

### 4.6 Intensionality

We next show the intensionality of our game semantics. In this section, we focus on the components of \( \text{WPG definable} \) by MLTT with One-, Zero-, N-, Pi-, Sigma- and Id-types (see [31] for the interpretation) and formulate type-theoretic principles in \( \text{WPG} \). In the following, let us also fix an arbitrary CwF \( C \) that supports One-, Zero-, N-, Pi-, Sigma- and Id-types, and assume \( \Gamma \in C, A \in Ty(\Gamma) \) and \( B \in Ty(\Gamma.A) \).

**Equality reflection.** Equality reflection [11] states that propositionally equal terms are judgmentally equal: \( \alpha = \alpha' \in \text{Ty}(\Gamma, A) \) if \( \text{Ty}(\Gamma, \text{Id}_A(\{\langle \text{id}, \alpha, \alpha' \rangle \}) \) is inhabited.

Equality reflection fails in our game semantics: \( N \& \textbf{0} \not\leq N \) and \( N \& \textbf{0} \not\geq N \) are different definable terms in \( \text{WPG}(\Sigma(N, \{0\}), \{N\}), \) but the elimination rule on Zero-type gives a definable one \( \rho \in \text{WPG}(\Sigma(N, \{0\}), \text{Id}_N(\{\langle \text{id}, \pi_1 \bullet \text{succ}, \pi_1 \rangle \})). \)

Recall that \( \rho \) is the strategy that makes, after the initial move \( q \) on the codomain is made by \( O \), the move \( q \) in \( \textbf{0} \) on the domain (see the proof of Theorem 4.20).
Function extensionality. Function extensionality [10] postulates that if terms of a Pi-type are propositionally equal as a function, then they are propositionally equal, i.e., the inhabitation of the type \( \Pi(A, \text{Id}_B \{ (\text{id}, \text{App}(\psi\{p\}, v)), \text{App}(\psi'\{p\}, v) \}) \Rightarrow \text{Id}_{\Pi(A, B)}\{ (\text{id}, \psi, \psi') \} \in \text{Ty} (\Gamma) \) for all terms \( \psi, \psi' \in \text{Ty} (\Gamma, \Pi(A, B)) \).

Our game semantics refutes function extensionality since terms \( \psi \in \mathsf{WPG} (\Gamma, A) \) are in general not completely specified by their extensions \( \gamma : \Gamma \Rightarrow \phi \circ \gamma \). To see this point closely, consider, e.g., the terms \( 0 \neq \psi \in \mathsf{WPG} (T, \Pi (\{ (N[0], \{ N[1] \})}) \) defined by
\[
0 := \text{Pref} (\{ q[1] q[0] n[0] n[0][1] | n \in N \})^\text{Even} \quad \text{and} \quad \psi := \text{Pref} (\{ q[1] 0[1] \})^\text{Even}.
\]
They are definable in MLTT by the introduction and the elimination rules of \( N \)-type. Then, for the reason mentioned above, the function extensionality on these terms fails.

Criteria of intensionality. There are Streicher’s Criteria of Intensionality [47]:

1. \( v\{p \circ p\} \neq v\{p\} \in \text{Ty} (\Gamma, A, A^+, \text{Id}_A, A\{p \circ p \circ p\}) \);
2. \( B\{p \circ p \circ p, v\{p \circ p\}\} \neq B\{p \circ p \circ p, v\{p\}\} \in \text{Ty} (\Gamma, A, A^+, \text{Id}_A) \);
3. If \( \text{Ty} (T, \text{Id}_S\{ (\text{id}_T, \sigma, \sigma') \}) \) is inhabited, then \( \sigma = \sigma' \in \text{Ty} (T, S) \), for all \( \Gamma \in C, A \in \text{Ty} (\Gamma), B \in \text{Ty} (\Gamma, A), S \in \text{Ty} (T) \) and \( \sigma, \sigma' \in \text{Ty} (T, S) \).

Our game semantics satisfies the first criterion. For example, let \( \Gamma := T, A := \{ N \}_{T \Rightarrow} \) and \( B := \{ N \}_{1 \Rightarrow (T, (N))} \). Then, \( v\{p \circ p\} \) is \( T \& N \& N \& \Sigma \& N \& N \& \Sigma \& N \), and \( v\{p\} \) is \( T \& N \& N \& \Sigma \& N \& N \& \Sigma \& N \& N \), and they are distinct.

In contrast, our game semantics does not satisfy the second criterion as dependent \( p \)-games are indexed by winning strategies: Any \( \tau \in \mathsf{WPG} (\Sigma (\Sigma (\Sigma (\Gamma, A), A^+), \text{Id}_A)) \) must be of the form \( \tau = (\langle \langle \gamma_0, \alpha_0 \rangle, \alpha_0', \sqrt{\gamma} \rangle)^1 \) such that \( \alpha_0 = \alpha_0' \). In other words, our game semantics of dependent types is extensional as opposed to that of terms.

Finally, our model satisfies the third criterion: Any \( \rho \in \mathsf{WPG} (T, \text{Id}_S\{ (\text{id}_T, \sigma, \sigma') \}) \) must be the strategy \( \sqrt{\gamma} \) (up to ‘tags’) since it is winning, and hence \( \sigma = \sigma' \).

4.7 Independence of Markov’s principle

Markov’s principle (MP) [26] is a subtle principle in constructive mathematics and computability theory. In fact, it depends on the school of constructive mathematics whether MP should be regarded as constructive. Roughly, MP postulates that if it is impossible that there is no natural number \( n \in N \) such that \( f(n) = 0 \) for a given function \( f : N \to N \), then there is a natural number \( n' \in N \) such that \( f(n') = 0 \).

We can formulate MP in MLTT as the type

\[
f : N \Rightarrow N \vdash \neg \Sigma x. N \text{Id}_N (f(x), \emptyset) \Rightarrow \Sigma y. N \text{Id}_N (f(y), \emptyset) \text{ type,}
\]

where \( \neg A \) abbreviates \( A \Rightarrow 0 \) for any type \( A \), and \( 0, N, \Rightarrow, \Pi, \Sigma \) and \( \text{Id}_N \) are Zero-, \( N \)-, implication, Pi-, Sigma- and Id-types, respectively.

Manna and Coquand [25] have shown that MP is independent from MLTT; i.e., there is no term of the type \( (4) \) in MLTT. Their proof of this result is syntactic.

However, even representative computational models of MLTT such as the effective topos [27] validate MP. Besides, it is easy to see that the model by Blot and Laird [24] validates MP too (though this is hardly surprising as their model admits classical reasonings [24, §8]). In other words, there is a gap between MLTT and these models.

In this context, we illustrate the precision of our game semantics of MLTT by:
Corollary 4.23 (Game semantics refutes Markov’s principle) There is no game-semantic term on the interpretation of (4) in WPG. Hence, the game semantics of MLTT in WPG (§4.4–4.5) implies that MP is independent from MLTT.

Proof (sketch) Assume for a contradiction that there is a term on the interpretation of (4) in WPG, which is the p-game

\[
\Pi(N_0) \Rightarrow N_1, (\Sigma(N_2), \Id_N (\langle \text{App}(\pi_1, \pi_2), \varnothing \rangle)_{[3]} \Rightarrow 0_{[4]} \Rightarrow 0_{[5]} \Rightarrow \Sigma(N_6), \Id_N (\langle \text{App}(\pi_1, \pi_2), \varnothing \rangle)_{[7]} \Rightarrow \varphi).
\]

where we omit the terminal p-game T and the bracket \{\} for constant dependent p-games; e.g., we write N for T\{N\}. We call \Sigma(N_6), \Id_N (\langle \text{App}(\pi_1, \pi_2), \varnothing \rangle)_{[7]} the \textit{codomain}, and N_0 \Rightarrow N_1 and (\Sigma(N_2), \Id_N (\langle \text{App}(\pi_1, \pi_2), \varnothing \rangle)_{[3]} \Rightarrow 0_{[4]} \Rightarrow 0_{[5]}) the outer and the inner \textit{domains} of (5). We write \langle \phi, \psi \rangle for the assumed term.

Assume first that there are \textit{total} input strategies \varphi^† \Rightarrow !(N_0) \Rightarrow N_1 and (\langle \_ \_ \_ \rangle^† \Rightarrow !(N_2), \Id_N (\langle \text{App}(\pi_1, \pi_2), \varnothing \rangle)_{[3]} \Rightarrow \varphi) on which \phi eventually makes a P-move \varphi^* \in N_6 when \Omega begins a play in N_0. If \varphi \bullet \varphi^* \neq 0 for some of these \varphi and \varphi^*, and \Omega plays by them, then \Id_N (\langle \text{App}(\pi_1, \pi_2), \varnothing \rangle)_{[7]} can be the empty p-game 0, and \psi plays on the domains \textit{forever}, contradicting its \textit{noetherianity}. If \varphi \bullet \varphi^* = 0 for all \varphi and \varphi^*, then \phi is strict since otherwise \varphi^* \varphi^* is the same even when \Omega slightly changes \varphi so that \varphi \bullet \varphi^* \neq 0, a contradiction; hence, \phi is strict and answers the question in N_6, yet with no answer to the question in 0_{[4]}, contradicting the \textit{well-bracketing} of \phi.

Thus, given any total inputs \varphi^† and (\langle \_ \_ \_ \rangle^†, \varphi) does not make a P-move in N_6. Similarly to the case of \psi, however, this contradicts the \textit{noetherianity} of \phi. \square

4.8 Game semantics of subtyping on dependent types

Finally, we show that p-games enable us to interpret \textit{subtyping on dependent types}. The subtyping relation between types is something like the subset relation between sets, and it is mainly to ensure that if \alpha is a term of a type \alpha, and \alpha is a subtype of a type B, then \text{coe}(\alpha) is a term of B that computes in the same way as \alpha, where \text{coe} is an operation on terms, called \textit{coercion}. Therefore, subtyping has a pragmatic importance since it ensures \textit{reusability} of terms in different types.

Let us first formulate a subtyping relation between dependent types in CwFs:

Definition 4.24 (CwFs with subtyping) A CwF \mathcal{C} that supports One-, Pi- and Sigma-types \textbf{supports subtyping} (on One-, Pi- and Sigma-types) if it is equipped with a partial order \leq_\Gamma on Ty(\Gamma) for each \Gamma \in \mathcal{C} that satisfies

1. \( (A \leq_\Gamma A' \wedge \alpha \in \text{Ty}(\Gamma, A)) \Rightarrow \alpha \in \text{Ty}(\Gamma, A'); \)
2. \( (A \leq_\Gamma A' \wedge \phi \in \mathcal{C}(\Delta, \Gamma)) \Rightarrow (\alpha \leq_\Delta A' \{\phi\}; \)
3. \( \forall \Gamma \in \mathcal{C}, A \in \text{Ty}(\Gamma). A \leq_\Gamma 1; \)
4. \( (A \leq_\Gamma A' \wedge B \leq_\Gamma A', B') \Rightarrow (\Pi(A', B) \leq_\Gamma \Pi(A, B') \wedge \Sigma(A, B) \leq_\Gamma \Sigma(A', B')); \)

The first axiom requires the reusability of terms. In addition, for the \textit{compositional} nature of denotational semantics, the other axioms require that constructions on types preserve \leq_\Gamma. Note that the last axiom tacitly assumes \( B, B' \in \text{Ty}(\Gamma, A) \), and the order between \( A \) and \( A' \) is \textit{flipped} in Pi-types as fewer morphisms can take more inputs [12, p. 419]. We define game semantics of subtyping based on Definition 3.1:
**Definition 4.25** (Predicate liveness ordering) The *predicate (p-) liveness ordering* is a partial order \( \preceq \) between p-games \( \Gamma \) and \( \Delta \) defined by \( \Gamma \preceq \Delta :\iff |\Gamma| = |\Delta| \land \forall \gamma : |\Gamma|, \Gamma(\gamma) \preceq \Delta(\gamma) \), and lifted to the one \( \preceq_{\Gamma} \) between linearly dependent p-games \( L \) and \( R \) over \( \Gamma \) by \( L \preceq_{\Gamma} R :\iff |L| = |R| \land \forall \gamma : \Gamma. L(\gamma) \preceq R(\gamma) \).

It is easy to see that the p-liveness orderings \( \preceq \) and \( \preceq_{\Gamma} \) are partial orders since so is the liveness ordering \( \preceq \) [42, Theorem 9]. Besides, the p-liveness ordering \( \preceq_{\Gamma} \) clearly satisfies the second and the third axioms of Definition 4.24, where \( 1 = \{ T \} \).

Moreover, if \( \gamma : \Gamma \) and \( \Gamma \preceq \Delta \), then \( \gamma : \Delta \) since \( \gamma : |\Gamma| = |\Delta| \) and \( \forall |\Delta| = \forall |\Gamma| \preceq \Gamma(\gamma) \preceq \Delta(\gamma) \). This observation shows that \( \preceq_{\Gamma} \) also satisfies the first axiom:

**Lemma 4.26** (Preservation of linear typing under predicate liveness ordering) Let \( L \) and \( R \) be linearly dependent p-games over a p-game \( \Gamma \) such that \( L \preceq_{\Gamma} R \).

1. If \( \psi : \Pi(\Gamma, L) \), then \( \psi : \Pi(\Gamma, R) \).
2. If \( \psi : \Pi(\Gamma, L) \) is winning (resp. w.b.), then so is \( \psi : \Pi(\Gamma, R) \).

Further, \( \preceq_{\Gamma} \) satisfies the last axiom as well by:

**Lemma 4.27** (Preservation of predicate liveness ordering) If \( \Delta \preceq \Delta' \) and \( \Gamma \preceq \Gamma' \),

then \( \Delta \preceq \Gamma \preceq \Delta' \preceq \Gamma' \), \( \Delta \preceq \Delta' \) and \( \Delta \preceq \Delta' \). Moreover, if \( A \preceq_{\Gamma} A' \) and \( B \preceq_{\Gamma, A': B'} \), then \( \Pi(A', B) \preceq_{\Gamma} \Pi(A, B') \) and \( \Sigma(A, B) \preceq_{\Gamma} \Sigma(A', B') \).

(N.b., strictly, the pair \( \langle |B|, \|B\| \upharpoonright \Gamma.A \rangle \), where \( \|B\| \upharpoonright \Gamma.A \) is the restriction of \( \|B\| \) to \( \mathrm{st}(\Gamma.A) \subseteq \mathrm{st}(\Gamma.A') \), not \( B \) itself, is an element of \( \mathcal{P}(\Gamma.A) \), and similarly for \( B' \).)

**Proof** For the first part, we focus on linear implication \( \rightarrow \) since the cases of the other constructions are simpler. First, observe \( |\Delta' \rightarrow \Gamma| = |\Delta'| \rightarrow |\Gamma| = |\Delta| \rightarrow |\Gamma'| = |\Delta| \rightarrow |\Gamma'| \). Next, let \( \phi : |\Delta' \rightarrow \Gamma| \); it remains to show \( (\Delta' \rightarrow \Gamma)(\phi) \preceq (\Delta \rightarrow \Gamma') \). However, it follows from \( \Delta \preceq \Delta' \) and \( \Gamma \preceq \Gamma' \) by induction on the lengths of positions, where the symmetry (between \( O \) and \( P \)) of the liveness ordering \( \preceq \) is crucial.

Finally, we show the second part by essentially the same way as the first part. 

Hence, we have finally shown:

**Corollary 4.28** (Game semantics of subtyping) \( \text{WPG} \) supports subtyping.

**Acknowledgements**

The author acknowledges the financial support from the Funai Foundation, and he is grateful to fruitful discussions with Samson Abramsky and Thierry Coquand.

**References**

1. Martin-Löf, P.: An Intuitionistic Theory of Types: Predicative Part. Studies in Logic and the Foundations of Mathematics 80, 73–118 (1975)
2. Martin-Löf, P.: An Intuitionistic Theory of Types. Twenty-five years of constructive type theory 36, 127–172 (1998)
3. Troelstra, A.S., van Dalen, D.: Constructivism in mathematics. Vol. I, volume 121 of. Studies in Logic and the Foundations of Mathematics, 26 (1988)
4. Zermelo, E.: Untersuchungen über die grundlagen der mengenlehre. i. Mathematische Annalen 65(2), 261–281 (1908)
5. Fraenkel, A.: Zu den grundlagen der cantor-zermeloschen mengenlehre. Mathematische annalen 86(3-4), 230–237 (1922)
6. Martin-Löf, P.: Constructive Mathematics and Computer Programming. Studies in Logic and the Foundations of Mathematics 104, 153–175 (1982)
7. Church, A.: A formulation of the simple theory of types. The Journal of Symbolic Logic 5(02), 56–68 (1940)  
8. Sørensen, M.H., Urzyczyn, P.: Lectures on the Curry-Howard Isomorphism vol. 149. Elsevier, ??? (2006)  
9. Constable, R., Allen, S., Bromley, H., Cleaveilad, W., Cremer, J., Harper, R., Howe, D., Knoblock, T., Mendler, N., Panangaden, P., et al.: Implementing mathematics with the Nuprl proof development system (1986)  
10. Univalent Foundations Program, T.: Homotopy Type Theory: Univalent Foundations of Mathematics. https://homotopytypetheory.org/book, Institute for Advanced Study (2013)  
11. Dybjer, P., Palmgren, E.: Intuitionistic type theory. Stanford Encyclopedia of Philosophy (2016)  
12. Amadio, R.M., Curien, P.-L.: Domains and Lambda-Calculi vol. 46. Cambridge University Press, Cambridge (1998)  
13. Girard, J.Y.: Linear logic. Theoretical Computer Science 50(1), 1–101 (1987)  
14. Hofmann, M., Streicher, T.: The Groupoid Interpretation of Type Theory. Twenty-five years of constructive type theory (Venice, 1995) 36, 63–111 (1998)  
15. Streicher, T.: Semantics of Type Theory: Correctness, Completeness and Independence Results. Springer, ??? (2012)  
16. Abramsky, S., et al.: Semantics of interaction: An introduction to game semantics. Semantics and Logics of Computation 14, 1–31 (1997)  
17. Hyland, M.: Game semantics. In: Semantics and Logics of Computation vol. 14, p. 131. Cambridge University Press, New York (1997)  
18. Shoenfield, J.R.: Mathematical Logic vol. 21. Addison-Wesley, Reading (1967)  
19. Curien, P.-L.: Definability and full abstraction. Electronic Notes in Theoretical Computer Science 172, 301–310 (2007)  
20. Abramsky, S.: Algorithmic game semantics. In: Proof and System-Reliability, pp. 21–47. Springer, ??? (2002)  
21. Plotkin, G., Power, J.: Computational effects and operations: An overview. Electronic Notes in Theoretical Computer Science 73, 149–163 (2004)  
22. Abramsky, S., McCusker, G.: Game semantics. In: Computational Logic: Proceedings of the 1997 Marktoberdorf Summer School, pp. 1–55. Springer, Berlin, Heidelberg (1999)  
23. Abramsky, S., Jagadeesan, R., Vákár, M.: Games for dependent types. In: Automata, Languages, and Programming, pp. 31–43. Springer, Berlin, Heidelberg (2015)  
24. Blot, V., Laird, J.: Extensional and intensional semantic universes: A denotational model of dependent types. In: Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, pp. 95–104 (2018). ACM  
25. Manna, B., Coquand, T.: The independence of markov’s principle in type theory. Logical Methods in Computer Science 13 (2017)  
26. Markov, A.A.: On constructive mathematics. Trudy Matematicheskogo Instituta imeni VA Steklova 67, 8–14 (1962)  
27. Hyland, J.M.E.: The effective topos. In: Studies in Logic and the Foundations of Mathematics vol. 110, pp. 165–216. Elsevier, ??? (1982)  
28. Pierce, B.C., Benjamin, C.: Types and Programming Languages. MIT press, ??? (2006)  
29. Vákár, M., Jagadeesan, R., Abramsky, S.: Game semantics for dependent types. Information and Computation 261, 401–431 (2018)  
30. Palmgren, E., Stoltenberg-Hansen, V.: Domain interpretations of martin-lof’s partial type theory. Annals of Pure and Applied Logic 4(2), 135–196 (1990)  
31. Hofmann, M.: Syntax and Semantics of Dependent Types. In: Extensional Constructs in Intensional Type Theory, pp. 13–54. Springer, ??? (1997)  
32. Berry, G., Curien, P.-L.: Sequential algorithms on concrete data structures. Theoretical Computer Science 20(3), 265–321 (1982)  
33. Abramsky, S., Jagadeesan, R., Malacaria, P.: Full abstraction for PCF. Information and Computation 163(2), 409–470 (2000)  
34. Hyland, J.M.E., Ong, C.-H.: On full abstraction for PCF: I, II, and III. Information and Computation 163(2), 295–406 (2000)  
35. McCusker, G.: Games and Full Abstraction for a Functional Metalanguage with Recursive Types. Springer, London (1998)  
36. Hyland, M., Schalk, A.: Games on graphs and sequentially realizable functionals. extended abstract. In: Proceedings 17th Annual IEEE Symposium on Logic in Computer Science, pp. 257–264 (2002). IEEE  
37. Yamada, N.: Game Semantics of Homotopy Type Theory. Talk presented at homotopy type theory electronic seminar.ta. J (HoTTTEST), Western University, Ontario, Canada, ??? (2021)  
38. Coquand, T.: A semantics of evidence for classical arithmetic. The Journal of Symbolic Logic 60(1), 325–337 (1995)  
39. Yamada, N.: A game-semantic model of computation. Research in the Mathematical Sciences 6(1), 3 (2019)  
40. Curien, P.-L.: Abstract Böhm trees. Mathematical Structures in Computer Science 8(06), 559–591 (1998)  
41. Laird, J.: Full abstraction for functional languages with control. In: Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science, pp. 58–67 (1997). IEEE  
42. Chroboczek, J.: Game semantics and subtyping. In: Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No. 99CB36332), pp. 192–203 (2000). IEEE  
43. Abramsky, S., Jagadeesan, R.: A game semantics for generic polymorphism. Annals of Pure and Applied Logic 133(1), 3–37 (2005)  
44. Laurent, O.: Polarized games. In: Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science, pp. 265–274 (2002). IEEE  
45. Dybjer, P.: Internal Type Theory. In: Types for Proofs and Programs, pp. 120–134. Springer, ??? (1996)  
46. Clairambault, P., Dybjer, P.: The Bi-continuity of Locally Cartesian Closed Categories and Martin-Löf Type Theories. Mathematical Structures in Computer Science 24(05), 240501 (2014)  
47. Streicher, T.: Investigations Into Intensional Type Theory, (1993)