Game semantics of universes in Martin-Löf type theory

Norihiro Yamada

CMUC, University of Coimbra, Department of Mathematics
3000-143 Coimbra, Portugal
norihiro@mat.uc.pt

Abstract

We extend game semantics of Martin-Löf type theory to a cumulative hierarchy of *universes*. This extension fulfils game semantics of all standard types in Martin-Löf type theory for the first time in the literature. More broadly, its contribution to mathematical semantics, constructive mathematics and foundations of mathematics is that it is the first *intensional* model of universes, showing that it is possible to interpret universes by *finitary* computational steps. In contrast, extensional models of universes, *e.g.*, realisability and domain models, were given more than 30 years ago. As a result, the powerful combinatorial method of game semantics becomes available for the study of universes and types generated by them. We illustrate this advantage by applying the game semantics to show the independence of Markov's principle from Martin-Löf type theory with universes. A challenge in obtaining game semantics of universes comes from a *conflict* between identity types and universes: Naive game semantics of the encoding of identity types by universes yields a decision procedure on the equality between functions, contradicting recursion theory. We conquer this challenge by novel games for universes whose strategies encode games for identity types *without deciding the equality*. In this way, we encode extensionally undecidable types *effectively* by intensional computations.

Keywords: Game semantics, universes, Martin-Löf type theory, constructive mathematics
2020 MSC: 03B70, 03B38, 03F55, 03B16, 03F50

1. Introduction

The present work establishes the first *game semantics of universes* in Martin-Löf type theory in the literature[1] and illustrates its utility by applying it to provide a novel proof of the independence of Markov's principle from Martin-Löf type theory equipped with universes. For this introduction, we assume familiarity with Martin-Löf type theory and universes, but not game semantics.

1.1. Martin-Löf type theory and the meaning explanation

On the one hand, *formal systems* [Sho67] are a class of syntactic formalisations of mathematics, and *constructive mathematics* [TvD88] is a family of computational or *constructive* schools in (foundations of) mathematics. On the other hand, a *model* or *semantics* [Tar54, Sco70] of a formal system is an assignment of mathematical objects to syntactic objects of the formal system, where the former serves as the *interpretation* of the latter.

[1] Blot and Laird [BL18] interpret a universe extensionally by a domain, not intensionally by a game.

Martin-Löf type theory (MLTT) [ML75, ML84, ML98] is one of the best-known formal systems for constructive mathematics, and it is comparable to axiomatic set theory [Zer08, Fra22] for classical mathematics. In addition, it is also a higher-order functional programming language [ML82], which generalises the simply-typed λ-calculus (STLC) [Chu40] along the generalisation of (intuitionistic) propositional logic to higher-order (intuitionistic) predicate logic under the Curry-Howard isomorphisms [SU06]. By this computational nature, MLTT underlies the computer formalisations of mathematics and their applications to functional programming [CAB$^+$86, Uni13].

A fundamental idea of MLTT is to regard (mathematical) objects and proofs in constructive mathematics uniformly as *computations* in an informal sense; MLTT is a syntactic formalisation of this beautiful idea due to Martin-Löf [ML82]. Accordingly, objects and proofs in MLTT are unified into programs or *terms*, where formulae are called *types*. This standard yet informal semantics of MLTT, which interprets terms as computations (and types as collections of the computations), is known as the *meaning explanation* of MLTT [DP16, §5]; it can be seen as an extension of the *BHK interpretation* of intuitionistic logic [Bro54, Hey31, Kol32] to constructive mathematics.

For illustrating the meaning explanation, let us recall that types in MLTT are written

$$\Gamma \vdash A \text{ type},$$

abbreviated as A, where Γ is an assumption or *context*, and terms of A as

$$\Gamma \vdash a : A,$$

abbreviated as $a : A$ or a. Let us also recall that the generalisation of STLC to MLTT is by allowing a type A to contain variables in the ambient context Γ, which corresponds under the Curry-Howard isomorphisms to the path from propositions to predicates. For instance, an *identity (Id-)type*,

$$\Gamma, x : A, y : A \vdash \mathrm{Id}_A(x, y) \text{ type},$$

in MLTT corresponds under the Curry-Howard isomorphisms to the predicate that holds when the objects x and y are equal. In this way, types in MLTT may depend on the contents of variables, so they are said to be *dependent*. A dependent type is said to be *constant* if it does not contain a variable; those constant ones are equivalent to *simple* types in STLC.

Then, for example, function types $\Rightarrow$ in STLC are generalised in MLTT to *Pi-types* Π, whose typing rules include

$$(\Pi\text{-Form})\ \frac{\Gamma, x : A \vdash B \text{ type}}{\Gamma \vdash \Pi_{x:A} B \text{ type}} \qquad (\Pi\text{-Intro})\ \frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash \lambda x.\, b : \Pi_{x:A} B}$$

$$(\Pi\text{-Elim})\ \frac{\Gamma \vdash f : \Pi_{x:A} B \qquad \Gamma \vdash a : A}{\Gamma \vdash \mathrm{app}(f, a) : B\{a/x\}}$$

where $B\{a/x\}$ is the result of *substituting* a for x in B, and we omit the computation and the uniqueness rules. The difference between Pi- and function types is that B in these rules may contain variables, especially x, unlike the corresponding rules on functions types. Under the Curry-Howard isomorphisms, the formation rule Π-Form corresponds to the formation of *universal quantification* $\forall_{x:A} B$, where the type $\Pi_{x:A} B$ is identified with the formula $\forall_{x:A} B$, and the introduction rule Π-Intro and the elimination rule Π-Elim to the introduction and the elimination rules for universal quantification in natural deduction, respectively. According to the meaning explanation, the term

$$\Gamma \vdash \lambda x.\, b : \Pi_{x:A} B$$

yielded in the rule Π-Intro represents a computation that transforms a proof x of A to that b of B. This computation of the Pi-type $\Pi_{x:A}B$ makes sense intuitively as a *(constructive) proof* of the universal quantification $\forall_{x:A}B$. In addition, the meaning explanation interprets the term

$$\Gamma \vdash \mathrm{app}(f, a) : B\{a/x\}$$

generated in the rule Π-Elim as the result of applying a proof f of $\Pi_{x:A}B$ to that a of A. This computational procedure corresponds to the *instantiation* of a proof of the formula $\forall_{x:A}B$.

Another example is *Sigma-types* Σ in MLTT, which generalise product types $\times$ in STLC. The typing rules on Sigma-types include

$$(\Sigma\text{-Form})\ \frac{\Gamma, x : A \vdash B\ \mathrm{type}}{\Gamma \vdash \Sigma_{x:A}B\ \mathrm{type}} \qquad (\Sigma\text{-Intro})\ \frac{\Gamma, x : A \vdash B\ \mathrm{type} \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : B\{a/x\}}{\Gamma \vdash \langle a, b\rangle : \Sigma_{x:A}B}$$

$$(\Sigma\text{-Elim})\ \frac{\Gamma, z : \Sigma_{x:A}B \vdash C\ \mathrm{type} \qquad \Gamma, x : A, y : B \vdash g : C\{\langle x, y\rangle/z\} \qquad \Gamma \vdash p : \Sigma_{x:A}B}{\Gamma \vdash R^{\Sigma}([z : \Sigma_{x:A}B]C, [x : A, y : B]g, p) : C\{p/z\}}$$

The formation rule Σ-Form corresponds under the Curry-Howard isomorphisms to the formation of *existential quantification* $\exists_{x:A}B$, where the type $\Sigma_{x:A}B$ is seen as the formula $\exists_{x:A}B$; the introduction rule Σ-Intro and the elimination rule Σ-Elim to the introduction and the elimination rules for existential quantification in natural deduction, respectively. Again, it is straightforward to give these rules the meaning explanation; see the standard articles [ML82, DP16] for the details.

Nevertheless, the meaning explanation cannot act as a mathematically firm ground to analyse, justify or develop MLTT since it does not formulate the central concept of *computation* in a mathematically precise fashion. Moreover, MLTT is an intricate formal system, which contains superficial syntactic details. This problem makes it technically challenging to study meta-theoretic properties of MLTT such as independence and consistency.

1.2. Game semantics of Martin-Löf type theory

Due to these fundamental problems, one calls for *mathematical semantics* of MLTT that formalises the meaning explanation in a mathematically precise manner, yet abstracting the superfluous syntactic details, and advances the meta-theoretic study of MLTT. Motivated in this way, Yamada [Yam23] has established game semantics of MLTT that satisfies these criteria.

Game semantics [Hyl97, A$^+$97] is a class of mathematical semantics that interprets types by *games* between *Player* (or an agent) and *Opponent* (or an oracle), and terms by *strategies* on how to *play* games. A game is a certain rooted forest together with a set of *positions*, where the vertices of the rooted forest are called *moves*, and the positions are a class of finite sequence of the moves starting with the roots. A strategy on a game is Player's *algorithm* on how to walk on (or *play*) the game alternately with Opponent, and said to be *winning* if it always leads to Player's *win*.

We represent walks or *plays* in a game by a class of potentially infinite sequences of positions,

$$\boldsymbol{\epsilon}, m_1, m_1m_2, m_1m_2m_3, \ldots,$$

where $\boldsymbol{\epsilon}$ is the empty sequence or position, and the finite sequences $m_1m_2 \ldots m_k$ of moves m_i are nonempty positions. By convention, the first move m_1 is always made by Opponent, and Player and Opponent alternately make moves. Hence, odd-indexed moves m_{2i+1} are made by Opponent, and even-indexed ones m_{2i} by Player. Yamada's work is based on McCusker's games and strategies [McC98], and we are recalling this variant here, suppressing *pointers* (Appendix A.1) for brevity.

We represent a strategy σ on a game G, written $\sigma : G$, by a partial function,

$$m_1 \mapsto m_1 m_2, \qquad m_1 m_2 m_3 \mapsto m_4, \qquad \dots \qquad m_1 m_2 \dots m_{2i+1} \mapsto m_{2i+2}, \qquad \dots,$$

from odd-length positions $m_1 m_2 \dots m_{2i+1}$ to moves m_{2i+2} for Player in G (*i.e.*, the strategy tells Player how to play in the game when it is her turn to make a move) such that the concatenations

$$m_1 m_2 \dots m_{2i+1} m_{2i+2}$$

are positions. Equivalently, a strategy can also be given as the set of these concatenations, and we use the both presentations interchangeably. A strategy is said to be *total* if so is it as a partial map.

Yamada's game semantics of MLTT formalises the meaning explanation *mathematically* and *syntax-freely* by modelling terms as winning strategies or *computations on the truths of formulae*. It is also an effective tool for the study of MLTT; *e.g.*, it verifies the *independence of Markov's principle* [Yam23, Corollary 4.7.1], which is impossible by most other mathematical semantics of MLTT such as the *effective topos* [Hyl82]. The point here is that game semantics is unique in its interpretation of terms by strategies or *intensional processes*, computing in a *step-by-step* fashion,

$$m_1 \mapsto m_1 m_2, \qquad m_1 m_2 m_3 \mapsto m_4, \qquad \dots \qquad m_1 m_2 \dots m_{2i+1} \mapsto m_{2i+2}, \qquad \dots,$$

while other mathematical semantics interprets terms by extensional objects such as functions. Because terms are also intensional objects, the game semantics achieves a very tight correspondence between terms and strategies, which makes itself a powerful tool for the study of MLTT.

1.3. Universes in Martin-Löf type theory

We can extend MLTT by a 'type of types' or *universe* [ML75]. The universe enables MLTT to significantly expand its realm of constructive mathematics. For instance, the elimination rule of natural number (N-)type with respect to the universe generates *infinitely* indexed dependent types such as the type of finite lists of natural numbers (Example 3.2.5).

The power of the universe is greatly increased as soon as it is combined with *well-founded tree (W-)types* [ML82]. For instance, MLTT together with the universe and W-types interprets Aczel's constructive set theory [Acz86], and the combination of the universe and W-types provides MLTT with a high proof-theoretic strength among constructive formal systems [Set93, GR94].

Note, however, that the universe does not accommodate itself since otherwise it would lead to an inconsistency [Gir72]. To overcome this problem, Martin-Löf [ML75, ML84] instead introduced a cumulative *hierarchy* $(\mathcal{U}_k)_{k\in\mathbb{N}}$ of universes $\mathcal{U}_k$. The first universe $\mathcal{U}_0$ does not accommodate itself, but the second one $\mathcal{U}_1$ does. Analogously, the second universe $\mathcal{U}_1$ does not accommodate itself, but the third one $\mathcal{U}_2$ does, and so forth. This hierarchy is *cumulative*: If $i < j$, then the universe $\mathcal{U}_j$ accommodates everything contained by the smaller one $\mathcal{U}_i$ plus $\mathcal{U}_i$ itself. In this way, the hierarchy collectively accommodates *each* type, including the universes themselves, without an inconsistency.

1.4. Main results

For the significant roles of universes in MLTT and constructive mathematics, we aim to extend Yamada's game semantics of MLTT to universes so that game semantics becomes available as a tool for the study of universes and types generated by them. (We explain why we do not employ other semantics in §1.5.) As clarified in §2, this is by no means an easy task, but:

Theorem 1.4.1 (effective game semantics of universes)**.** *Yamada's game semantics of MLTT is extendable to the cumulative hierarchy of universes, in which strategies are all effectively computable in the standard sense of recursion theory.*

This theorem in turn extends Yamada's independence proof:

Corollary (independence of Markov's principle)**.** *Markov's principle [Mar62] is independent from MLTT equipped with the cumulative hierarchy of universes.*

This result illustrates a strong advantage of game semantics: Its combinatorial reasoning such as the independence proof remains valid *even if it is extended to new types*. As a result, when the game semantics of MLTT has been extended to new types, its meta-theoretic results on MLTT such as the independence of Markov's principle are *automatically* extended to those types too.

This advantage makes the game semantics a powerful tool for the study of MLTT. In contrast, the proof by Mannaa and Coquand [MC17], for instance, does not have such a *modular* property: An extension of MLTT may invalidate their syntactic, inductive reasoning.

1.5. Our contributions and related work

Our main contribution is the establishment of the first game semantics of the cumulative hierarchy of universes in the literature. More broadly, it is the first *intensional* semantics of universes (§4.2), while extensional variants of computational models of universes, *e.g.*, domains [Pal93] and realisability [Str91], were already given in early 1990's. Note that none of the existing intensional semantics of MLTT [AJV15, VJA18, BL18] models universes by intensional computations. We solve this long-standing problem in constructive mathematics, type theory and mathematical semantics.

As explained in §2.3, a main technical challenge in obtaining game semantics of universes is to encode games for Id-types by strategies on universes. We overcome this challenge by the novel idea to encode games by strategies that consist of ordinary computations and additional symbolic ones as sketched in §2.4, while we permit the decoding El to be extensionally uncomputable (without sacrificing the effectivity of strategies). This idea in turn requires a nontrivial recursive definition of games for universes, and one of our technical contributions is to realise it (Definition 3.1.2).

Another contribution of ours is to provide a new proof of the independence of Makov's principle from MLTT equipped with the cumulative hierarchy of universes. This result illustrates the *modularity* of the game semantics: A meta-theoretic result on MLTT given by the game semantics is automatically extended to new types as soon as the interpretation is extended to the types.

As related work, we mention that Abramsky *et al.* [AJV15] have obtained the first intensional semantics of MLTT. While this is a significant achievement, it interprets Sigma-types indirectly by formal lists of (families of) games, not games, which makes an interpretation of universes hopeless. Also, their method is valid only for a specific class of types [VJA18, Figure 7], which excludes a list type. Because a list type is constructible from N-type and a universe (Example 3.2.5), this also implies that their method cannot model universes. Thus, we instead adopt Yamada's semantics.

Blot and Laird [BL18] also model a universe [BL18, Table 3], for which they write $\Gamma \vdash_{\mathcal{E}} \mathcal{I}$ type, extensionally by a domain, not intensionally by a game. Although they do not interpret Id-types, they sketch how to do it *via* finite tuples of Boolean type [BL18, §9]; however, this method does not work in the presence of N-type since the set $\mathbb{N}$ of all natural numbers is unbounded. While their semantics is a valuable contribution to the literature, we do not employ it for these reasons.

McCusker [McC98] and Clairambault [Cla09] have established game semantics of *recursive types*. One may then wonder if their methods are applicable to universes. However, the answer is negative:

Whilst a variety of type constructions encoded by universes can be made into the form of recursive types, their methods cannot interpret dependent types. For instance, Id-types cannot be presented as recursive types because they take terms as parameters. This point illustrates the novelty of the present work. In addition, their interpretations of recursive types are extensional, while our game semantics of universes is intensional. Again, this novel intensionality is our main contribution.

Lastly, Manna and Coquand [MC17] have shown the independence of Markov's principle from MLTT equipped with a universe for the first time in the literature. Their proof is syntactic, which stands in contrast to our game-semantic proof. As we have mentioned in §1.4, their syntactic proof is not necessarily straightforward to extend to other types, while our game-semantic proof is.

1.6. Concluding remarks

Similarly to Yamada [Yam23], we do not give a full completeness result for the following reasons. First of all, the syntax of MLTT, specifically N-type, is not suited to full completeness. For instance, the full completeness theorem given by Abramsky *et al.* [AJV15, VJA18] is on a *modification* of MLTT, which excludes N-type. Because our main subject of study is MLTT itself, we leave a full completeness result on some modification of MLTT as future work. Secondly, one of our aims is to provide a tool for the study of MLTT, but a full completeness theorem is not necessarily essential for this aim. For example, the fully complete semantics by Blot and Laird [BL18] cannot verify the independence of Markov's principle because it admits classical reasoning. Hence, we instead show the *utility* of our game semantics by applying it to some meta-theoretic results on MLTT in §4.

1.7. Article structure

The remainder of this article proceeds as follows. We first informally describe a main challenge in obtaining game semantics of universes and sketch our solution to this problem in §2. We then present our result in §3: game semantics of the cumulative hierarchy of universes. We next show some corollaries of this result in §4 such as the independence of Markov's principle from MLTT equipped with the hierarchy. Appendix A summarises the game semantics of MLTT [Yam23].

Notation. Throughout the present article, we use the following notations:

- We use bold letters $\boldsymbol{s}$, $\boldsymbol{t}$, $\boldsymbol{u}$, $\boldsymbol{v}$ etc. for sequences, in particular $\boldsymbol{\epsilon}$ for the *empty sequence*, and small letters a, b, m, n, x, y, etc. for the elements of sequences;
- Let $\overline{n} := \{\, 1, 2, \ldots, n \,\}$ for each positive integer $n \in \mathbb{N}_+ := \mathbb{N} \setminus \{0\}$, and $\overline{0} := \emptyset$;
- We also write $x_1 x_2 \ldots x_{|\boldsymbol{s}|}$ for a finite sequence $\boldsymbol{s} = (x_1, x_2, \ldots, x_{|\boldsymbol{s}|})$, where $|\boldsymbol{s}|$ is the *length* of $\boldsymbol{s}$, and define $\boldsymbol{s}(i) := x_i$ for each $i \in \overline{|\boldsymbol{s}|}$;
- The *concatenation* of finite sequences $\boldsymbol{s}$ and $\boldsymbol{t}$ is represented by the juxtaposition $\boldsymbol{st}$ (or $\boldsymbol{s}.\boldsymbol{t}$), but we often write $a\boldsymbol{s}$, $\boldsymbol{t}b$ and $\boldsymbol{u}c\boldsymbol{v}$ for $(a)\boldsymbol{s}$, $\boldsymbol{t}(b)$ and $\boldsymbol{u}(c)\boldsymbol{v}$, respectively;
- We write $\mathrm{Even}(\boldsymbol{s})$ (respectively, $\mathrm{Odd}(\boldsymbol{s})$) if a sequence $\boldsymbol{s}$ is of even- (respectively, odd-)length, and for a set S of sequences and a predicate $\mathrm{P} \in \{\mathrm{Even}, \mathrm{Odd}\}$ define $S^{\mathrm{P}} := \{\, \boldsymbol{s} \in S \mid \mathrm{P}(\boldsymbol{s}) \,\}$;
- We write $\boldsymbol{s} \preceq \boldsymbol{t}$ if $\boldsymbol{s}$ is a *prefix* of a sequence $\boldsymbol{t}$, and for a set S of sequences, $\mathrm{Pref}(S)$ for the set of all the prefixes of the sequences in S.

2. Our challenge and solution in a nutshell

Before going into the details of our game semantics of universes in §3, we first informally recall games and strategies in §2.1, and based on them Yamada's game semantics of MLTT in §2.2. We next explain a challenge in extending the game semantics to universes in §2.3. Finally, we sketch how to overcome the challenge in §2.4, and how to extend it to the hierarchy of universes in 2.5.

2.1. Games and strategies

To understand our main challenge explained in §2.3, we begin with an informal introduction to games and strategies *à la* McCusker [McC98, AM99b]. According to the original definition, a *game* consists of a certain type of finite rooted forest, called an *arena*, and a class of finite sequences or *positions* of the vertices or *moves* of the arena starting with the roots. For technical convenience, Yamada [Yam23] recasts this definition in such a way that a game is identified with a certain set of finite sequences, which are to be regarded as positions, and an arena is *derived* from the positions. He also defines a *subgame* of a game G to be a game G' such that $G' \subseteq G$. In the following, we adopt Yamada's reformulation; the details are collected in Appendix A.1.

For instance, the simplest game is the *empty game*

$$T := \{\boldsymbol{\epsilon}\},$$

which has no moves to play. Thus, it has only the trivial strategy

$$\bot := \{\boldsymbol{\epsilon}\} : T.$$

As another, more substantial example, consider the game

$$N := \{\boldsymbol{\epsilon}, q\} \cup \{\, qn \mid n \in \mathbb{N} \,\}$$

of natural numbers. In this game, a play starts with Opponent's move q ('What is your number?') and ends with Player's move n ('My number is n!'). This *natural number game* N is not always the most preferred one because it is not very different from the set $\mathbb{N}$ of all natural numbers, and there is another, much more intensional game $\mathscr{N}$ for natural numbers [Yam19]. Nevertheless, the game N is simpler and suffices for our aim, so we adopt it in this article.

Moreover, the subset

$$2N := \{\boldsymbol{\epsilon}, q\} \cup \{\, q(2n) \mid n \in \mathbb{N} \,\} \subseteq N$$

is a subgame of N. Then, a strategy $\underline{n} : N$ ($n \in \mathbb{N}$) is just the map $q \mapsto n$, and it corresponds to the number n. Note that, *e.g.*, the strategy $\underline{7}$ is *not* valid on the game $2N$.

Next, there is a binary construction & on games, called *product*, which is a product in a category of games and strategies. The product $G \,\&\, H$ of games G and H is simply the disjoint union

$$G \,\&\, H := G \uplus H$$

of G and H. In other words, a position in $G \,\&\, H$ is a one in G or H. For instance, a maximal position of the product $N \,\&\, N$ of the game N and itself is either[2]

$N_{[0]}$	&	$N_{[1]}$
$q_{[0]}$		
$n_{[0]}$		

$N_{[0]}$	&	$N_{[1]}$
		$q_{[1]}$
		$m_{[1]}$

[2]The diagrams are only to make it explicit which component game each move belongs to; the positions in the diagrams are just the finite sequences $q_{[0]}n_{[0]}$ and $q_{[1]}m_{[1]}$.

where $n, m \in \mathbb{N}$, and the subscripts $(\_)_{[i]}$ $(0 \leqslant i \leqslant 1)$ are *tags* for the disjoint union to distinguish between the two copies of N. We often omit the tags if it does not bring confusion. We write $\langle \underline{n}, \underline{m} \rangle$ for the strategy on $N \,\&\, N$ that plays as in the diagrams; it is the *pairing* of the strategies $\underline{n}, \underline{m} : N$.

Another binary construction $\multimap$, called *linear implication*, accommodates the strategies playing as *linear functions* in the sense of linear logic [Gir87], *i.e.*, the functions that consume exactly one input to produce an output. A position in the linear implication $G \multimap H$ between games G and H is defined to be an interleaving mixture of positions in G and H such that

1. The first element of the position is a move in H;
2. Each change of the GH-parity in the position is by Player, *i.e.*, at an even-indexed move.

For example, a typical position in the linear implication $N \multimap N$ looks like

$N_{[0]}$	$\multimap$	$N_{[1]}$
		$q_{[1]}$
$q_{[0]}$		
$n_{[0]}$		
		$m_{[1]}$

which can be read as follows:

1. Opponent's question $q_{[1]}$ for an output ('What is your output?');
2. Player's question $q_{[0]}$ for an input ('Wait, what is your input?');
3. Opponent's answer $n_{[0]}$ to $q_{[0]}$ ('OK, here is an input n.');
4. Player's answer $m_{[1]}$ to $q_{[1]}$ ('Alright, the output is then m.').

This position corresponds to any linear function $\mathbb{N} \to \mathbb{N}$ that maps $n \mapsto m$. Then, for instance, the strategy $\mathrm{succ} : N \multimap N$ is the function that maps

$$q_{[1]} \mapsto q_{[0]}, \qquad\qquad q_{[1]} q_{[0]} n_{[0]} \mapsto n + 1_{[1]}$$

for all $n \in \mathbb{N}$, or diagrammatically it plays as

$N_{[0]}$	$\overset{\mathrm{succ}}{\multimap}$	$N_{[1]}$
		$q_{[1]}$
$q_{[0]}$		
$n_{[0]}$		
		$n + 1_{[1]}$

By the way, the play $q_{[1]} \mapsto m_{[1]}$ in $N \multimap N$, computing as a *constant* linear function that maps $x \mapsto m$ for all $x \in \mathbb{N}$, is possible too. Thus, strictly speaking, the linear implication $G \multimap H$ is the game for *affine maps* from G to H, but we follow the convention to call $\multimap$ linear implication.

Now, note that the linear implication $N \,\&\, N \multimap N$, where by convention & precedes $\multimap$, does not admit strategies that compute *binary* maps such as addition because its maximal positions are

N	&	N	$\multimap$	N
				q
q				
n				
				m

N	&	N	$\multimap$	N
				q
		q		
		n		
				m

N	&	N	$\multimap$	N
				q
				m

and these positions may contain *at most one* input on the domain $N \,\&\, N$.

The unary construction ! on games, called *exponential* or *of-course*, addresses this problem by defining a desired game $G \Rightarrow H$ for ordinary (not necessarily linear) functions from G to H by

$$G \Rightarrow H := !G \multimap H,$$

where by convention ! precedes $\multimap$ and &. This idea comes from linear logic [Gir87].

A position in the exponential $!G$ is defined to be an interleaving mixture of a finite number of positions in G in which each switch between positions in different copies of G is made by Opponent, *i.e.*, at an odd-indexed move. For instance, the exponential $!(N \,\&\, N)$ has the positions

$!(N \;\&\; N)$
q
n
q
m

$!(N \;\&\; N)$
q
n
q
m

so that there are strategies $\mathrm{add}, \mathrm{add}' : N \,\&\, N \Rightarrow N$ for addition that compute respectively as

$N \;\&\; N \overset{\mathrm{add}}{\Rightarrow} N$
q
q
n
q
m
$n + m$

$N \;\&\; N \overset{\mathrm{add}'}{\Rightarrow} N$
q
q
n
q
m
$n + m$

Although these strategies both implement addition, their computational processes are slightly different, illustrating the *intensional* nature of game semantics.

We next recall the *composition*

$$\psi \bullet \phi : G \Rightarrow K$$

of strategies $\phi : G \Rightarrow H$ and $\psi : H \Rightarrow K$. For instance, consider those $\mathrm{succ}, \mathrm{double} : N \Rightarrow N$ (where succ is not on the game $N \multimap N$, but its computation remains to be the same) computing as

$N_{[0]} \overset{\mathrm{succ}}{\Rightarrow} N_{[1]}$
$q_{[1]}$
$q_{[0]}$
$m_{[0]}$
$m + 1_{[1]}$

$N_{[2]} \overset{\mathrm{double}}{\Rightarrow} N_{[3]}$
$q_{[3]}$
$q_{[2]}$
$n_{[3]}$
$2n_{[2]}$

The composition $\mathrm{double} \bullet \mathrm{succ} : N \Rightarrow N$ is then defined as follows. First, we take the *promotion* $\mathrm{succ}^\dagger : !N_{[0]} \multimap !N_{[1]}$ of $\mathrm{succ} : !N_{[0]} \multimap N_{[1]}$, which computes as succ itself for each position in $!N_{[0]} \multimap N_{[1]}$ inside $!N_{[0]} \multimap !N_{[1]}$. Thus, a position played by $\mathrm{succ}^\dagger$ looks like as in Figure 1.

Next, we *synchronise* $\mathrm{succ}^\dagger$ and double *via* the codomain $!N_{[1]}$ of $\mathrm{succ}^\dagger$ and the domain $!N_{[2]}$ of double, for which Player also plays the role of Opponent in the component games $!N_{[1]}$ and $!N_{[2]}$ by copying her last moves as in Figure 2, where the moves for the synchronisation are marked by square boxes for clarity. Importantly, Opponent plays on the *external game* $N_{[0]} \Rightarrow N_{[3]}$, seeing only the moves in the component games $!N_{[0]}$ and $N_{[3]}$. This play is to be read as follows:

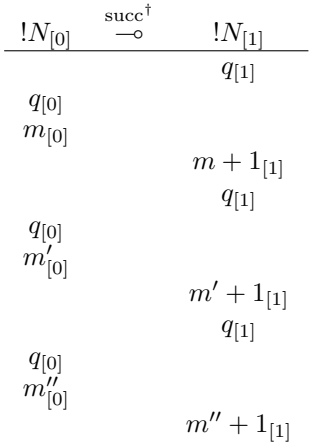

Figure 1: An example of promotion

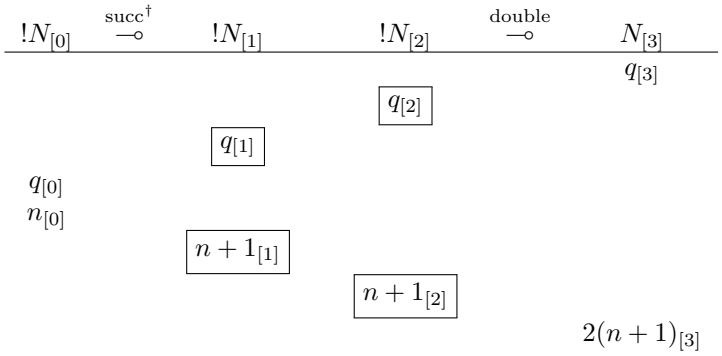

Figure 2: An example of an interaction between strategies

1. Opponent's question $q_{[3]}$ for an output in $!N_{[0]} \multimap N_{[3]}$ ('What is your output?');
2. Player's question $\boxed{q_{[2]}}$ by double for an input in $!N_{[2]} \multimap N_{[3]}$ ('Wait, what is your input?');
3. $\boxed{q_{[2]}}$ then triggers the question $\boxed{q_{[1]}}$ for an output in $!N_{[0]} \multimap !N_{[1]}$ ('What is your output?');
4. Player's question $q_{[0]}$ by $\mathrm{succ}^\dagger$ for an input in $!N_{[0]} \multimap !N_{[1]}$ ('Wait, what is your input?');
5. Opponent's answer $n_{[0]}$ to $q_{[0]}$ in $!N_{[0]} \multimap !N_{[3]}$ ('Here is an input n.');
6. Player's answer $\boxed{n+1_{[1]}}$ to $\boxed{q_{[1]}}$ by $\mathrm{succ}^\dagger$ in $!N_{[0]} \multimap !N_{[1]}$ ('The output is then $n+1$.');
7. $\boxed{n+1_{[1]}}$ then triggers the answer $\boxed{n+1_{[2]}}$ to $\boxed{q_{[2]}}$ in $!N_{[2]} \multimap N_{[3]}$ ('Here is the input $n+1$.');
8. Player's answer $2 \cdot (n+1)_{[3]}$ to $q_{[3]}$ by double in $!N_{[0]} \multimap N_{[3]}$ ('The output is then $2(n+1)$!').

Finally, we *hide* all the moves enclosed with the square boxes from the play, resulting in the strategy $\mathrm{double} \bullet \mathrm{succ} : N \Rightarrow N$ for the function $n \mapsto 2(n+1)$ as expected:

$N_{[0]}$	$\overset{\mathrm{double}\bullet\mathrm{succ}}{\Rightarrow}$	$N_{[3]}$
		$q_{[3]}$
$q_{[0]}$		
$n_{[0]}$		
		$2(n+1)_{[3]}$

The category of games and strategies has games as objects, strategies $\phi : G \Rightarrow H$ as morphisms $G \to H$, and the composition of strategies as the categorical composition. For each object G, the identity $\mathrm{id}_G : G \to G$ in the category is the strategy that simply *copy-cats* Opponent's last moves:

G	$\overset{\mathrm{id}_G}{\Rightarrow}$	G
		m_1
m_1		
m_2		
		m_2
		m_3
m_3		
	$\vdots$	

One can also compose strategies $\gamma : G$ and $\phi : G \Rightarrow H$ in the same vein, yielding a strategy $\phi \bullet \gamma : H$. For instance, we have the composition $\mathrm{double} \bullet \underline{n} = \underline{2n}$ for all $n \in \mathbb{N}$. Alternatively, the strategy $\phi \bullet \gamma : H$ can be recasted as the ordinary composition $\phi \bullet \gamma : T \Rightarrow H$ of $\gamma : T \Rightarrow G$ and $\phi : G \Rightarrow H$ thanks to the evident isomorphism $G \cong (T \Rightarrow G)$ in the category, where we do not distinguish between strategies on G and $T \Rightarrow G$ because they are essentially the same.

The *pairing* $\langle \phi, \psi \rangle : K \to G \,\&\, H$ of morphisms $\phi : K \to G$ and $\psi : K \to H$ plays by ϕ if the first move is in G, and by ψ otherwise. This generalises the example $\langle \underline{n}, \underline{m} \rangle : (N \,\&\, N) \cong (T \Rightarrow (N \,\&\, N))$.

We have seen that strategies are algorithms computing in a *step-by-step, finitary* fashion. This unique *intensionality* distinguishes game semantics from other mathematical semantics.

2.2. Game semantics of Martin-Löf type theory

We are now ready to sketch the main ideas of Yamada's game semantics of MLTT [Yam23]. We leave the details to Appendix A.2.

First, it is not a problem to interpret dependent types in MLTT since we can simply interpret a dependent type $x : C \vdash D(x)$ type by a family

$$D = \{D(\sigma)\}_{\sigma:C}$$

of games $D(\sigma)$ indexed by strategies σ on the game C that interprets the simple type $\vdash C$ type, abbreviated as C. Here, by abuse of notation, we omit the semantic bracket $[\![\_]\!]$ and notationally do not distinguish between types and games; we employ this convention as long as it does not bring confusion. The dependent type has just one variable, but in the presence of One- and Sigma-types (which is the case for this work) dependent types with a single variable cover all dependent types.

Next, in light of product & on games, which interprets a particular class of Sigma-types Σ, *viz.*, product types $\times$, it seems to be a natural idea to interpret the Sigma-type $\Sigma_{x:C}D(x)$ by a subgame

$$\Sigma(C, D) \subseteq C \,\&\, \bigcup_{\sigma:C} D(\sigma)$$

such that strategies on this hypothetical game $\Sigma(C, D)$ are the pairings $\langle \sigma, \tau \rangle$ of strategies $\sigma : C$ and $\tau : D(\sigma)$. However, this idea does not work due to the following two problems:

1. Each game G already determines the set $\mathrm{St}(G)$ of all strategies on G;
2. It is impossible for Player, 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

$$x : N \vdash N_b(x) \text{ type}$$

such that canonical terms of the simple type $N_b(\underline{k})$ for each $k \in \mathbb{N}$ are the numerals $\underline{n}$ such that $n \leqslant k$, and assume that we interpret this dependent type N_b by the family

$$N_b = \{N_b(\sigma)\}_{\sigma:N}$$

of games $N_b(\sigma)$ defined by

$$N_b(\underline{k}) := \{\boldsymbol{\epsilon}, q\} \cup \{\, qn \mid n \leqslant k \,\}, \qquad\qquad N_b(\bot) := N,$$

where $\mathrm{St}(N) = \{\, \underline{k} \mid k \in \mathbb{N} \,\} \cup \{\bot\}$. However, there is no subgame

$$G \subseteq N \,\&\, \bigcup_{\sigma:N} N_b(\sigma) = N \,\&\, N$$

such that $\langle \underline{k}, \underline{n} \rangle : G$ if and only if $\underline{n} : N_b(\underline{k})$, *i.e.*, $n \leqslant k$, for all $k, n \in \mathbb{N}$ since if such a game G exists, then we would have

$$\langle \underline{0}, \underline{0} \rangle, \langle \underline{1}, \underline{1} \rangle : G,$$

which (together with the definition of a strategy on a game) implies

$$\langle \underline{0}, \underline{1} \rangle : G,$$

a contradiction. Hence, no game can interpret the Sigma-type $\Sigma_{x:N}N_b(x)$.

We next give an example of the second problem. Consider a dependent type

$$x : N \vdash \mathrm{List}_N(x) \text{ type}$$

such that canonical terms of the simple type $\mathrm{List}_N(\underline{k})$ for each $k \in \mathbb{N}$ are k-lists of numerals, and assume that this dependent type List_N is modelled by the family

$$\mathrm{List}_N = \{\mathrm{List}_N(\sigma)\}_{\sigma:N}$$

of games $\mathrm{List}_N(\sigma)$ such that $\mathrm{List}_N(\underline{k})$ is the k-ary product & on N, where

$$\mathrm{List}_N(\underline{0}) := T, \qquad \mathrm{List}_N(\bot) := \bigcup_{k\in\mathbb{N}} \mathrm{List}_N(\underline{k}).$$

If there is a subgame

$$H \subseteq N \,\&\, \bigcup_{\sigma:N} \mathrm{List}_N(\sigma)$$

that interprets the Sigma-type $\Sigma_{x:N}\mathrm{List}_N(x)$, then the pairing

$$\langle \underline{k}, \langle \cdots \langle \underline{n_1}, \underline{n_2} \rangle, \cdots, \underline{n_k} \rangle \rangle$$

for all $k, n_1, n_2, \ldots, n_k \in \mathbb{N}$ would be *total* on H since strategies modelling proofs are total [Yam23]. Yet, Opponent may select, by his first move, *e.g.*, the $(k+1)$st component of $\mathrm{List}_N(\underline{k+1})$, for which the above pairing has no next move; *i.e.*, it is *not* total. Thus, there is no such a game H.

We have observed two fundamental limitations of games. Yamada [Yam23] has overcome these limitations by generalising a game to a pair

$$\Gamma = (|\Gamma|, \|\Gamma\|)$$

of a game $|\Gamma|$ and a family $\|\Gamma\| = \{\Gamma(\gamma)\}_{\gamma:|\Gamma|}$ of subgames $\Gamma(\gamma) \subseteq |\Gamma|$, called a *predicate (p-)game*, and defining a strategy γ *on* Γ, written $\gamma : \Gamma$, to be a one $\gamma : |\Gamma|$ *compatible* with $\Gamma(\gamma)$ in the sense that positions in $\Gamma(\gamma)$ are closed under the computations $\boldsymbol{so} \mapsto \boldsymbol{sop}$ of γ. In other words, a p-game Γ is a game $|\Gamma|$ together with a *specification* $\|\Gamma\|$ for strategies $\gamma : |\Gamma|$ to be valid on Γ.

Crucially, a play in a p-game Γ depends on Player's choice of a strategy $\gamma : \Gamma$ in the sense that her choice γ specifies the game $\Gamma(\gamma)$ to play. To depict this idea vividly, Yamada introduces *Judge*; a play in Γ then proceeds as follows. First, Judge asks Player a question q_Γ ('What is your strategy?'), and then Player answers it by a strategy $\gamma : \Gamma$ ('It is γ!'). After this *initial protocol*, an ordinary play in $\Gamma(\gamma)$ between Player and Opponent follows, where Player must play by γ yet restricted to $\Gamma(\gamma)$. In this fashion, Player declares her strategy γ, which is observed by Judge yet *not by Opponent*, and it predetermines the game $\Gamma(\gamma)$ for a play between Opponent and Player.

Strictly speaking, Judge and initial protocols are *informal* devices to explain this idea. Neither is part of p-games; it suffices to assign games $\Gamma(\gamma)$ to strategies $\gamma : \Gamma$ by the family $\|\Gamma\|$. However, we often add an initial protocol informally to the beginning of a play for *pedagogical* reasons.

Then, the generalisation of games to p-games solves the first problem as follows. Let $\Sigma(N, N_b)$ be the p-game defined by $|\Sigma(N, N_b)| := N \,\&\, N$ and

$$\Sigma(N, N_b)(\langle \sigma, \tau \rangle) := \begin{cases} N \,\&\, N_b(\underline{k}) & \text{if } \sigma = \underline{k} \text{ with } k \in \mathbb{N}; \\ N \,\&\, N & \text{otherwise} \end{cases}$$

for all $\langle \sigma, \tau \rangle : |\Sigma(N, N_b)|$. Hence, strategies on $\Sigma(N, N_b)$ are the pairings $\langle \sigma, \tau \rangle : N \,\&\, N$ such that $\tau : N_b(\underline{k})$ if $\sigma = \underline{k}$. For instance, some positions played by the strategy $\langle \underline{7}, \underline{3} \rangle : \Sigma(N, N_b)$ are

$\Sigma(N,$		$N_b)$
	$q_{\Sigma(N,N_b)}$	
	$\langle \underline{7}, \underline{3} \rangle$	
q		
7		

$\Sigma(N,$		$N_b)$
	$q_{\Sigma(N,N_b)}$	
	$\langle \underline{7}, \underline{3} \rangle$	
		q
		3

where Judge first asks Player the question $q_{\Sigma(N,N_b)}$ ('What is your strategy?'), and Player answers it by the strategy $\langle \underline{7}, \underline{3} \rangle : \Sigma(N, N_b)$ ('It is $\langle \underline{7}, \underline{3} \rangle$!'); then, a play between Player and Opponent on the game $\Sigma(N, N_b)(\langle \underline{7}, \underline{3} \rangle) = N \,\&\, N_b(\underline{7})$ follows, where Player must play by the one $\langle \underline{7}, \underline{3} \rangle$. Although the predetermination of a strategy is not strictly necessary in this example, Player cannot play by an *invalid* strategy, say, $\langle \underline{0}, \underline{1} \rangle$, on $\Sigma(N, N_b)$ because it is not compatible with the game $\Sigma(N, N_b)(\langle \underline{0}, \underline{1} \rangle)$. In this way, the specification $\|\Sigma(N, N_b)\|$ solves the first problem by *filtering* strategies.

On the other hand, the predetermination of a strategy in a p-game plays a crucial role in solving the second problem: The p-game $\Sigma(N, \mathrm{List}_N)$ defined by

$$\Sigma(N, \mathrm{List}_N)(\underline{k}) := N \,\&\, (\underbrace{N \,\&\, N \,\&\, \ldots \,\&\, N}_{k}) \quad (k \in \mathbb{N}), \qquad \Sigma(N, \mathrm{List}_N)(\bot) := \bigcup_{k \in \mathbb{N}} \Sigma(N, \mathrm{List}_N)(\underline{k}),$$

$$|\Sigma(N, \mathrm{List}_N)| := \Sigma(N, \mathrm{List}_N)(\bot)$$

interprets the Sigma-type $\Sigma_{\mathrm{x:N}}\mathrm{List_N(x)}$. Then, some typical positions in $\Sigma(N, \mathrm{List}_N)$ are

$\Sigma(N,$		$\mathrm{List}_N)$
	$q_{\Sigma(N,\mathrm{List}_N)}$	
	$\langle \underline{2}, \langle \underline{1}, \underline{3} \rangle \rangle$	
q		
2		

$\Sigma(N,$				$\mathrm{List}_N)$
	$q_{\Sigma(N,\mathrm{List}_N)}$			
	$\langle \underline{2}, \langle \underline{1}, \underline{3} \rangle \rangle$			
			q	
		1	or	3

where the predetermination of the strategy $\langle \underline{2}, \langle \underline{1}, \underline{3} \rangle \rangle$ *fixes* the underlying game $N \,\&\, (N \,\&\, N)$. As a result, this strategy is *total* on the p-game $\Sigma(N, \mathrm{List}_N)$ (again in the sense of partial maps).

In Appendix A.2, we recall how p-games interpret other type constructions in MLTT.

2.3. Problem: how to encode games for identity types by strategies

However, it turns out to be a challenge to realise game semantics of a universe in MLTT, and it has been open in the long history of game semantics (since the pioneering work [AJ94]).[3]

Specifically, the challenge is

how to encode games that interpret Id-types by strategies.

To see what it really means, let us recall the rules on a universe $\mathcal{U}$ in MLTT:

[3] Again, Blot and Laird [BL18] interpret a universe [BL18, Table 3], but it is extensionally by domain theory.

- The formation rule postulates for each context Γ the universe

$$\Gamma \vdash \mathcal{U} \text{ type};$$

- The introduction rule *encodes* each type $\Gamma \vdash A$ type by a term

$$\Gamma \vdash \mathrm{En}(A) : \mathcal{U};$$

- The elimination rule is embodied by a type

$$x : \mathcal{U} \vdash \mathrm{El}(x) \text{ type}$$

  via the substitution

$$(\Gamma \vdash u : \mathcal{U}) \mapsto (\Gamma \vdash \mathrm{El}(u) \text{ type})$$

  of a term u for the variable x in the type $\mathrm{El}(x)$;

- The computation rule requires the equation

$$\Gamma \vdash \mathrm{El}(\mathrm{En}(A)) = A \text{ type}.$$

Note that the universe itself is a *simple* type, so we shall interpret it by a *constant* p-game, *i.e.*, a game together with a constant family of its subgames, which is identified with a single game in the evident way. Hence, in order to interpret the above rules, we have to provide

- A game (or constant p-game) $\mathcal{U}$ that interprets the formation rule,
- A strategy $\mathrm{En}(A) : \Gamma \Rightarrow \mathcal{U}$ for each family A of games that interprets the introduction rule,
- A family $\mathrm{El} = \{\mathrm{El}(\mu)\}_{\mu:\mathcal{U}}$ of games $\mathrm{El}(\mu)$ that interprets the elimination rule and for all $\gamma : \Gamma$ satisfies the equation $\mathrm{El}(\mathrm{En}(A) \bullet \gamma) = A(\gamma)$ for the computation rule.

Having sketched what is necessary for game semantics to interpret a universe, we now explain the challenge. Let A be the Id-type

$$f : N \Rightarrow N, g : N \Rightarrow N \vdash \mathrm{Id}_{N\Rightarrow N}(f, g) \text{ type}$$

on the function type $N \Rightarrow N$. Yamada [Yam23] interprets this Id-type by a family

$$\mathrm{Id}_{N\Rightarrow N} = \{\mathrm{Id}_{N\Rightarrow N}(\langle f, g\rangle)\}_{\langle f,g\rangle:(N\Rightarrow N)\&(N\Rightarrow N)} = \{\mathrm{Id}_{N\Rightarrow N}(\langle f, g\rangle)\}_{f,g:N\Rightarrow N}$$

of (extremely simple) games $\mathrm{Id}_{N\Rightarrow N}(\langle f, g\rangle)$, whose details are left to Appendix A.2.22. Thus, the game semantics has to interpret the term

$$f : N \Rightarrow N, g : N \Rightarrow N \vdash \mathrm{En}(\mathrm{Id}_{N\Rightarrow N}(f, g)) : \mathcal{U},$$

which encodes the Id-type, by a strategy

$$\mathrm{En}(\mathrm{Id}_{N\Rightarrow N}) : ((N \Rightarrow N) \,\&\, (N \Rightarrow N)) \Rightarrow \mathcal{U} \tag{1}$$

that satisfies the equation

$$\mathrm{El}(\mathrm{En}(\mathrm{Id}_{N\Rightarrow N}) \bullet \langle f, g\rangle) = \mathrm{Id}_{N\Rightarrow N}(\langle f, g\rangle)$$

for all $f, g : N \Rightarrow N$. Since each game $\mathrm{Id}_{N\Rightarrow N}(\langle f, g\rangle)$ depends on whether the equation $f = g$ holds (Appendix A.2.22), the strategy $\mathrm{En}(\mathrm{Id}_{N\Rightarrow N}) \bullet \langle f, g\rangle$ *must vary over the cases* $f = g$ *and* $f \neq g$.

Thus, the strategy (1) is an algorithm that effectively decides if f and g are equal for all inputs $f, g : N \Rightarrow N$, a contradiction to a well-known fact in recursion theory [Rog67]. In other words, the problem is that the family $\mathrm{Id}_{N\Rightarrow N}$ of games is *(extensionally) not effectively computable*. This uncomputability corresponds, in game semantics, to the problem that the strategy (1) can learn about only a *finite* number of input-output pairs of f and g; *i.e.*, it can never decide if the equation $f = g$ holds, as illustrated by the following position played by the strategy (1)

$$\begin{array}{ccccccccccc}
(N & \overset{f}{\Rightarrow} & N) & \& & (N & \overset{g}{\Rightarrow} & N) & \overset{\mathrm{En}(\mathrm{Id}_{N\Rightarrow N})}{\Rightarrow} & \mathcal{U} \\
\hline
 & & & & & & & & q \\
 & & q & & & & & & \\
q & & & & & & & & \\
n & & & & & & & & \\
 & & f(n) & & & & & & \\
 & & & & & & q & & \\
 & & & & q & & & & \\
 & & & & n' & & & & \\
 & & & & & & g(n') & & \\
 & & & & & & & & \vdots \\
 & & & & & & & & ?
\end{array}$$

in which the strategy never collects the complete information about f or g.

Let us see more concretely how the following naive method fails due to this issue. We assign a number $\#(C) \in \mathbb{N}$ to the game C that interprets a simple type $\vdash C$ type except the universe $\mathcal{U}$, where the assignment $\#$ is injective, and define a game $\mathcal{U}$ whose maximal positions are of the form

$$\begin{array}{c}
\mathcal{U} \\
\hline
q \\
\#(C)
\end{array}$$

Intuitively, the initial move q is Opponent's question 'What is your game?', and the second one #(C) is Player's answer 'My game is C!'. If a strategy

$$\mathrm{En}(\mathrm{Id}_{N\Rightarrow N}) : \Gamma \Rightarrow \mathcal{U} \tag{2}$$

encodes the family $\mathrm{Id}_{N\Rightarrow N}$, then it would decide the number $\#(\mathrm{Id}_{N\Rightarrow N}(\langle f, g\rangle))$, depending on if

the equation $f = g$ holds, which is impossible (indicated by $\not\Rightarrow$ below) as illustrated by the play

$$\begin{array}{cccccccccc}
(N & \overset{f}{\Rightarrow} & N) & \& & (N & \overset{g}{\Rightarrow} & N) & \overset{\mathrm{En}(\mathrm{Id}_{N\Rightarrow N})}{\not\Rightarrow} & \mathcal{U} \\
\hline
 & & & & & & & & q \\
 & & q & & & & & & \\
q & & & & & & & & \\
n & & & & & & & & \\
 & & f(n) & & & & & & \\
 & & & & & & q & & \\
 & & & & q & & & & \\
 & & & & n' & & & & \\
 & & & & & & g(n') & & \\
 & & & & & & & \vdots & \\
 & & & & & & & & \#(\mathrm{Id}_{N\Rightarrow N}(\langle f, g\rangle))
\end{array}$$

because the strategy (2) never collects the complete information about f or g. In the rest of the present article, we forget about this naive definition of the game $\mathcal{U}$.

2.4. Solution: encoding without deciding

A key observation behind our solution to the problem just sketched in §2.3 is that the strategy

$$\mathrm{En}(\mathrm{Id}_{N\Rightarrow N}) \bullet \langle f, g\rangle : \mathcal{U}$$

does not have to decide whether the equation $f = g$ holds in a finitary fashion, *e.g.*, by the *one-step* computation $q \mapsto \#(\mathrm{Id}_{N\Rightarrow N}(\langle f, g\rangle))$; instead, the strategy may encode the game $\mathrm{Id}_{N\Rightarrow N}(\langle f, g\rangle)$ by *potentially infinite* plays in the game $\mathcal{U}$.

This observation leads to our solution sketched below. First, arbitrarily fix distinct numbers

$$\#(1), \#(0), \#(N), \#(\Pi), \#(\Sigma), \#(\mathrm{Id}) \in \mathbb{N}.$$

We shall define a game $\mathcal{U}$ for the universe such that there is a strategy $\mathrm{En}(\mathrm{Id}_{N\Rightarrow N})$ of the form (1) that first computes by $q \mapsto \#(\mathrm{Id})$ (indicating that it encodes an Id-type) and then, depending on the next move by Opponent, plays as the one $\mathrm{En}(N \Rightarrow N)$ (indicating that the Id-type is on the type $N \Rightarrow N$) or *copy-cats* the inputs $f, g : N \Rightarrow N$ given by Opponent in a *step-by-step, finitary* fashion (indicating that the Id-type is between f and g) *without fully detecting what f or g is*.

Such a strategy $\mathrm{En}(\mathrm{Id}_{N\Rightarrow N})$ encodes the family $\mathrm{Id}_{N\Rightarrow N}$ of games without sacrificing its *effectivity*: The copy-cat of f and g is trivially effective, while *potentially infinite* plays by the composition $\mathrm{En}(\mathrm{Id}_{N\Rightarrow N}) \bullet \langle f, g\rangle$ encode f and g entirely. Meanwhile, the strategy never completes the encoding of f or g within a *finite* play. We illustrate this point below, where we sketch the game $\mathcal{U}$ as well.

We now sketch the game $\mathcal{U}$ (Definition 3.1.2), which realises the above idea, and give examples of plays in $\mathcal{U}$. First, we encode the base case, *i.e.*, the games 1, 0 and N that model One-, Zero- and N-types, respectively [Yam23], by strategies on the game $\Gamma \Rightarrow \mathcal{U}$. Thus, $\mathcal{U}$ has the positions

$$\begin{array}{ccc}
\mathcal{U} & \mathcal{U} & \mathcal{U} \\
\hline
q & q & q \\
\#(1) & \#(0) & \#(N)
\end{array}$$

so that there are strategies $\mathrm{En}(1), \mathrm{En}(0), \mathrm{En}(N) : \Gamma \Rightarrow \mathcal{U}$ that compute by

$$\mathrm{En}(1) : q \mapsto \#(1), \qquad \mathrm{En}(0) : q \mapsto \#(0), \qquad \mathrm{En}(N) : q \mapsto \#(N).$$

We next consider the inductive step that encodes Pi- and Sigma-types. Suppose that a family $A = \{A(\gamma)\}_{\gamma:\Gamma}$ of games $A(\gamma)$ interprets a type $\Gamma \vdash A$ type, and a strategy $\mathrm{En}(A) : \Gamma \Rightarrow \mathcal{U}$ the encoding $\Gamma \vdash \mathrm{En}(A) : \mathcal{U}$. For simplicity, let Γ be the empty context; thus, Γ is the empty game T that has only the trivial strategy $\bot$, and A is identified with a game. Assume further that a family $B = \{B(\alpha)\}_{\alpha:A}$ of games $B(\alpha)$ interprets a type $x : A \vdash B$ type, and a strategy $\mathrm{En}(B) : A \Rightarrow \mathcal{U}$ the encoding $x : A \vdash \mathrm{En}(B) : \mathcal{U}$. Recall that Yamada interprets the Pi- and the Sigma-types

$$\vdash \Pi(A, B) \text{ type}, \qquad \vdash \Sigma(A, B) \text{ type}$$

by (the singleton families of) games

$$\Pi(A, B), \qquad \Sigma(A, B),$$

respectively, constructed out of A and B (Appendix A.2.19 and Appendix A.2.20). Then, there must be strategies

$$\mathrm{En}(\Pi(A, B)), \mathrm{En}(\Sigma(A, B)) : (T \Rightarrow \mathcal{U}) \cong \mathcal{U}$$

that encode these games. For this reason, the game $\mathcal{U}$ also has positions of the form

$\mathcal{U}$	$\mathcal{U}$	$\mathcal{U}$	$\mathcal{U}$
q	q	q	q
$\#(\Pi)$	$\#(\Pi)$	$\#(\Sigma)$	$\#(\Sigma)$
a_1	b_1	a_1	b_1
a_2	b_2	a_2	b_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$

where $a_1 a_2 \ldots$ are moves played by the strategy $\mathrm{En}(A) : \mathcal{U}$, and $b_1 b_2 \ldots$ by the one $\mathrm{En}(B) : A \Rightarrow \mathcal{U}$. We postulate these positions in $\mathcal{U}$ so that we can define the strategy $\mathrm{En}(\Pi(A, B))$ to be the pairing

$$\langle \mathrm{En}(A), \mathrm{En}(B) \rangle : \mathcal{U} \,\&\, (A \Rightarrow \mathcal{U})$$

prefixed by the moves $q.\#(\Pi)$, and similarly for the one $\mathrm{En}(\Sigma(A, B))$. In this manner, the game $\mathcal{U}$ encodes the games $\Pi(A, B)$ and $\Sigma(A, B)$ by its strategies.

Note that the games $\mathcal{U}$ and $A \Rightarrow \mathcal{U}$ for the positions $a_1 a_2 \ldots$ and $b_1 b_2 \ldots$, respectively, contain the game $\mathcal{U}$ itself. In particular, $A \Rightarrow \mathcal{U}$ is not $\mathcal{U}$ but the function game from A to $\mathcal{U}$. Our idea thus necessitates a nontrivial *recursive* definition of $\mathcal{U}$. One of our main achievements is to realise such a nontrivial game $\mathcal{U}$, subsuming the general case where Γ can differ from the empty one T.

Finally, there must be another strategy

$$\mathrm{En}(\mathrm{Id}_A) : (A \,\&\, A) \Rightarrow \mathcal{U}$$

that encodes the family

$$\mathrm{Id}_A = \{\mathrm{Id}_A(\langle \alpha, \alpha' \rangle)\}_{\langle \alpha, \alpha' \rangle : A \& A} = \{\mathrm{Id}_A(\langle \alpha, \alpha' \rangle)\}_{\alpha, \alpha' : A}$$

of games $\mathrm{Id}_A(\langle \alpha, \alpha' \rangle)$, where this family Id_A is the interpretation of the Id-type

$$x : A, x' : A \vdash \mathrm{Id}_A(x, x') \text{ type}$$

in Yamada [Yam23] (Appendix A.2.22). The required equation

$$\mathrm{El}(\mathrm{En}(\mathrm{Id}_A) \bullet \langle \alpha, \alpha' \rangle) = \mathrm{Id}_A(\langle \alpha, \alpha' \rangle)$$

implies that the strategy $\mathrm{En}(\mathrm{Id}_A) \bullet \langle \alpha, \alpha' \rangle : \mathcal{U}$ encodes the game $\mathrm{Id}_A(\langle \alpha, \alpha' \rangle)$. In the light of this encoding of $\mathrm{Id}_A(\langle \alpha, \alpha' \rangle)$, we add the positions

□	$\mathcal{U}$	□	□		□	$\mathcal{U}$	□	□		□	$\mathcal{U}$	□	□
	q					q					q		
	$\#(\mathrm{Id})$					$\#(\mathrm{Id})$					$\#(\mathrm{Id})$		
a_1							c_1						c'_1
a_2							c_2						c'_2
$\vdots$							$\vdots$						$\vdots$

to the game $\mathcal{U}$, where the moves $a_1 a_2 \dots$ are played by the strategy $\mathrm{En}(A) : \mathcal{U}$, those $c_1 c_2 \dots$ by the one $\alpha : A$, and those $c'_1 c'_2 \dots$ by the one $\alpha' : A$, and we write auxiliary squares □ on the top row to clarify the column to which each move belongs. That is, we postulate these positions in $\mathcal{U}$ so that we can define the encoding $\mathrm{En}(\mathrm{Id}_A(\langle \alpha, \alpha' \rangle))$ to be the pairing

$$\langle \mathrm{En}(A), \langle \alpha, \alpha' \rangle \rangle : \mathcal{U} \,\&\, (A \,\&\, A)$$

prefixed by the moves $q.\#(\mathrm{Id})$.

Then, based on this strategy, we can further define a strategy

$$\mathrm{En}(\mathrm{Id}_A) : (A \,\&\, A) \Rightarrow \mathcal{U}$$

that satisfies the equation

$$\mathrm{En}(\mathrm{Id}_A) \bullet \langle \alpha, \alpha' \rangle = \mathrm{En}(\mathrm{Id}_A(\langle \alpha, \alpha' \rangle))$$

as sketched below. Instead of the general case, we focus on its instance

$$\mathrm{En}(\mathrm{Id}_{N \Rightarrow N}) : ((N \Rightarrow N) \,\&\, (N \Rightarrow N)) \Rightarrow \mathcal{U} \tag{3}$$

in the rest of this section since this suffices for explaining our idea. The upshot is that we define the strategy (3) to be what plays in either of the ways in Figure 3, depending on plays by Opponent.

In the first two patterns of the figure, the strategy (3) encodes the underlying (constant) family $A = \{N \Rightarrow N\}$ of games by playing as the pairing

$$\langle \mathrm{En}(N), \mathrm{En}(N) \rangle : \mathcal{U} \,\&\, \mathcal{U}$$

prefixed by the moves $q.\#(\Pi)$, where $\Pi(N, N) = N \Rightarrow N$. In the last two patterns of the figure, what the strategy (3) does is essentially to *copy-cat* the input strategies f or g given by Opponent. This computation is trivially *effective*, but also its (potentially infinite) plays have all the information about f and g, in particular *if $f = g$ or not*. In this way, we solve the problem sketched in §2.3.

Now, recall that the family $\mathrm{Id}_{N \Rightarrow N}$ of games is (extensionally) uncomputable, and the extension

$$\langle f, g \rangle \mapsto \mathrm{El}(\mathrm{En}(\mathrm{Id}_{N \Rightarrow N}) \bullet \langle f, g \rangle) = \mathrm{Id}_{N \Rightarrow N}(\langle f, g \rangle)$$

of the strategy $\mathrm{En}(\mathrm{Id}_{N \Rightarrow N})$, combined with the decoding El, coincides with the uncomputable family $\mathrm{Id}_{N \Rightarrow N}$. This, however, does *not* contradict the effectivity of the strategy $\mathrm{En}(\mathrm{Id}_{N \Rightarrow N})$ because it does not decide if the equation $f = g$ holds in a finitary way; it only collects more and more yet incomplete, finite information about f and g, which never halts.

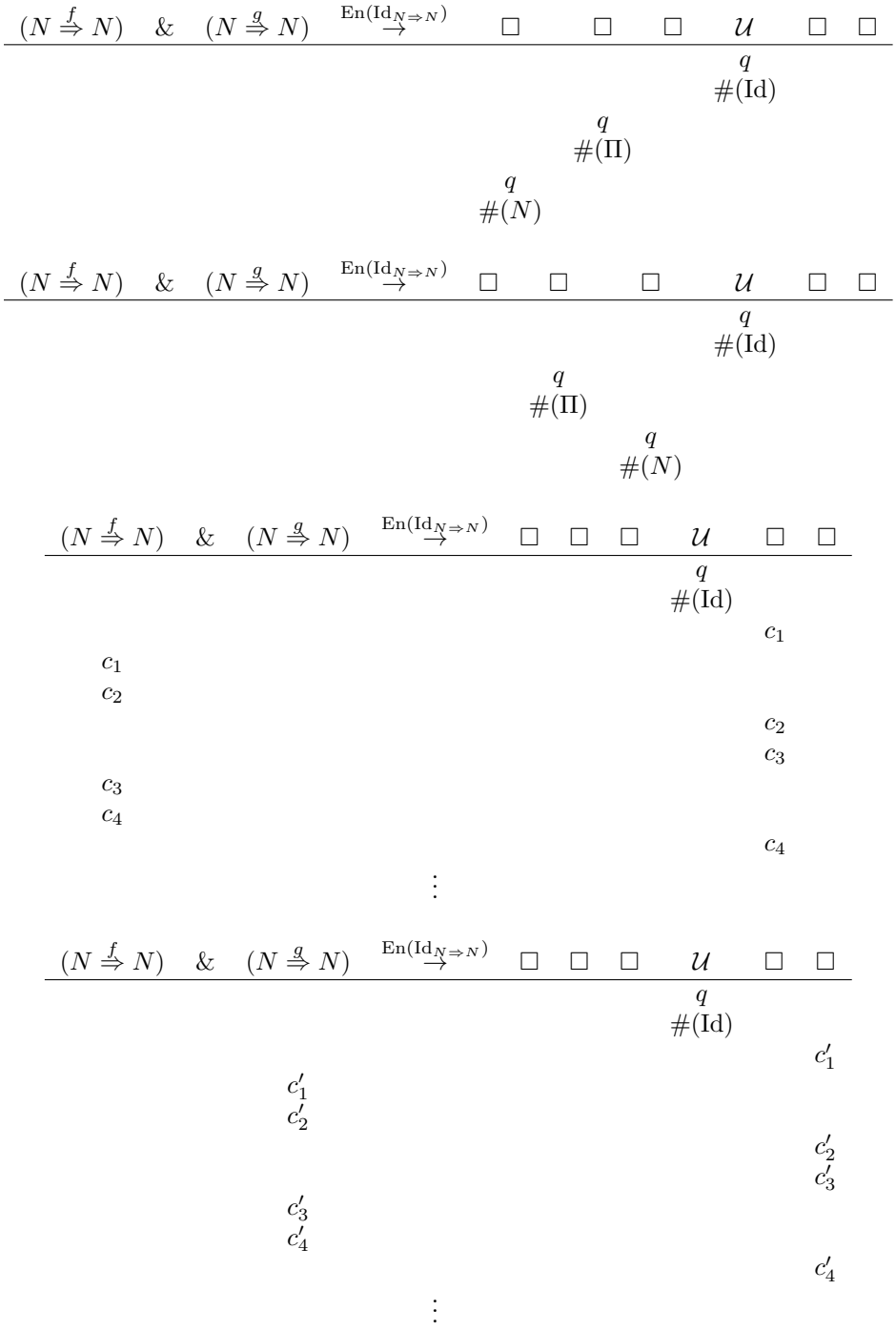

Figure 3: An illustration of the strategy on the encoding of an Id-type between functions

2.5. Lifting to a cumulative hierarchy of universes

The universe $\mathcal{U}$ does not have its own code since otherwise the code $\Gamma \vdash \mathrm{En}(\mathcal{U}) : \mathcal{U}$ leads to inconsistency known as *Girard's paradox* [Gir72]. To overcome this issue, Martin-Löf [ML75, ML84] excluded the code and proposed a *cumulative hierarchy* $(\mathcal{U}_k)_{k\in\mathbb{N}}$ of universes $\mathcal{U}_k$. The first universe $\mathcal{U}_0$ does not have its own code $\mathrm{En}(\mathcal{U}_0)$, but the second one $\mathcal{U}_1$ does. Analogously, the second universe $\mathcal{U}_1$ does not contain its own code $\mathrm{En}(\mathcal{U}_1)$, but the third one $\mathcal{U}_2$ does, and so on. This hierarchy is *cumulative*: If $i < j$, then the larger universe $\mathcal{U}_j$ contains all codes in the smaller one $\mathcal{U}_i$ plus $\mathrm{En}(\mathcal{U}_i)$. In this way, the hierarchy collectively encodes *every* type, including the universes themselves, by a code in some universe $\mathcal{U}_k$. Here, the original universe $\mathcal{U}$ can be identified with the first one $\mathcal{U}_0$.

Having established the game $\mathcal{U}$ for the universe, it is straightforward to lift the game semantics to the cumulative hierarchy of universes:

1. For the base case, we define the first universe game $\mathcal{U}_0$ by $\mathcal{U}_0 := \mathcal{U}$;
2. For the inductive step, we obtain the higher universe game $\mathcal{U}_{k+1}$ by adding new pairwise distinct moves $\#(\mathcal{U}_i) \in \mathbb{N}$ for $i = 0, 1, \ldots, k$ to the definition of the game $\mathcal{U}$.

3. Game semantics of universes

This section presents our main contribution: game semantics of universes. Recall that Yamada [Yam23] achieves game semantics of MLTT by showing that the category $\mathbb{WPG}_!$ of p-games and strategies (Appendix A.2.11) forms standard categorical semantics of MLTT, known as a *category with families (CwF)* [Dyb96]. Thus, our task is to equip the CwF $\mathbb{WPG}_!$ (Appendix A.2.18) with the *semantic type-former* [Hof97, §3.3] for universes (which is a categorical generalisation of the game semantics of universes sketched in §2.3). To this end, let us first recall CwFs:

Definition 3.0.1 (categories with families [Dyb96])**.** A *category with families (CwF)* is a tuple

$$\mathcal{C} = (\mathcal{C}, \mathrm{Ty}, \mathrm{Tm}, \_\{\_\}, T, \_.\_, \mathrm{p}, \mathrm{v}, \langle \_, \_ \rangle_\_)$$

such that

- $\mathcal{C}$ is a category with a terminal object $T \in \mathcal{C}$;
- Ty assigns, to each object $\Gamma \in \mathcal{C}$, a set $\mathrm{Ty}(\Gamma)$ of *types* in the *context* Γ;
- Tm assigns, to each pair (Γ, A) of an object $\Gamma \in \mathcal{C}$ and a type $A \in \mathrm{Ty}(\Gamma)$, a set $\mathrm{Tm}(\Gamma, A)$ of *terms* of type A in the context Γ;
- $\_\{\_\}$ assigns, to each morphism $\phi : \Delta \to \Gamma$, a map $\_\{\phi\} : \mathrm{Ty}(\Gamma) \to \mathrm{Ty}(\Delta)$, called the *substitution on types*, and a family $\{\_\{\phi\}_A\}_{A\in\mathrm{Ty}(\Gamma)}$ of maps $\_\{\phi\}_A : \mathrm{Tm}(\Gamma, A) \to \mathrm{Tm}(\Delta, A\{\phi\})$, called the *substitution on terms*;
- $\_.\_$ assigns, to each pair (Γ, A) of a context $\Gamma \in \mathcal{C}$ and a type $A \in \mathrm{Ty}(\Gamma)$, a context $\Gamma.A \in \mathcal{C}$, called the *comprehension* of A;
- p (respectively, v) associates each pair (Γ, A) of a context $\Gamma \in \mathcal{C}$ and a type $A \in \mathrm{Ty}(\Gamma)$ with a morphism $\mathrm{p}_A : \Gamma.A \to \Gamma$ (respectively, a term $\mathrm{v}_A \in \mathrm{Tm}(\Gamma.A, A\{\mathrm{p}_A\})$), called the *first projection* (respectively, the *second projection*) on A;

- $\langle \_, \_ \rangle_\_$ associates each triple $(\phi, A, \check{\alpha})$ of a morphism $\phi : \Delta \to \Gamma$, a type $A \in \mathrm{Ty}(\Gamma)$ and a term $\check{\alpha} \in \mathrm{Tm}(\Delta, A\{\phi\})$ with a morphism $\langle \phi, \check{\alpha} \rangle_A : \Delta \to \Gamma.A$, called the *extension* of ϕ by $\check{\alpha}$,

that satisfies, for any object $\Theta \in \mathcal{C}$, morphism $\varphi : \Theta \to \Delta$ and term $\alpha \in \mathrm{Tm}(\Gamma, A)$, the equations

- (Ty-Id) $A\{\mathrm{id}_\Gamma\} = A$,
- (Ty-Comp) $A\{\phi \circ \varphi\} = A\{\phi\}\{\varphi\}$,
- (Tm-Id) $\alpha\{\mathrm{id}_\Gamma\}_A = \alpha$,
- (Tm-Comp) $\alpha\{\phi \circ \varphi\}_A = \alpha\{\phi\}_A\{\varphi\}_{A\{\phi\}}$,
- (Cons-L) $\mathrm{p}_A \circ \langle \phi, \check{\alpha} \rangle_A = \phi$,
- (Cons-R) $\mathrm{v}_A\{\langle \phi, \check{\alpha} \rangle_A\} = \check{\alpha}$,
- (Cons-Nat) $\langle \phi, \check{\alpha} \rangle_A \circ \varphi = \langle \phi \circ \varphi, \check{\alpha}\{\varphi\}_{A\{\phi\}} \rangle_A$,
- (Cons-Id) $\langle \mathrm{p}_A, \mathrm{v}_A \rangle_A = \mathrm{id}_{\Gamma.A}$,

where we sometimes write $\mathrm{Ty}_\mathcal{C}$, $\mathrm{Term}_\mathcal{C}$ and so on if we want to emphasise the underlying CwF $\mathcal{C}$.

Roughly speaking, judgements in MLTT are interpreted in a CwF $\mathcal{C}$ by

$$\vdash \Gamma \text{ ctx} \mapsto [\![\Gamma]\!] \in \mathcal{C}, \qquad \Gamma \vdash \mathrm{A} \text{ type} \mapsto [\![\mathrm{A}]\!] \in \mathrm{Ty}([\![\Gamma]\!]), \qquad \Gamma \vdash \mathrm{a} : \mathrm{A} \mapsto [\![\mathrm{a}]\!] \in \mathrm{Tm}([\![\Gamma]\!], [\![\mathrm{A}]\!]),$$

$$\vdash \Gamma = \Delta \text{ ctx} \Rightarrow [\![\Gamma]\!] = [\![\Delta]\!], \qquad \Gamma \vdash \mathrm{A} = \mathrm{B} \text{ type} \Rightarrow [\![\mathrm{A}]\!] = [\![\mathrm{B}]\!], \qquad \Gamma \vdash \mathrm{a} = \mathrm{a}' : \mathrm{A} \Rightarrow [\![\mathrm{a}]\!] = [\![\mathrm{a}']\!],$$

where the square bracket $[\![\_]\!]$ denotes the *semantic map* or *interpretation* [Hof97].

Nevertheless, CwFs only interpret the core fragment of MLTT common to all types. To interpret individual type constructions such as Pi-, Sigma-, One-, N- and Id-types, CwFs must be equipped with their *semantic type formers* [Hof97, §3.3]. We recall these semantic type-formers for the CwF $\mathbb{WPG}_!$ in Appendix A.2; in the following, we focus on the semantic type former for universes:

Definition 3.0.2 (categorical universes [Hof97])**.** A CwF $\mathcal{C}$ is said to have *(a cumulative hierarchy of) universes* if

- (U-Form) Given an object $\Gamma \in \mathcal{C}$, there is a type

  $$\mathcal{U}_k^{[\Gamma]} \in \mathrm{Ty}(\Gamma)$$

  for each $k \in \mathbb{N}$, called the $(k+1)$*st universe* in Γ, where we often omit the superscript $(\_)^{[\Gamma]}$ (when Γ is obvious) and/or the subscript $(\_)_k$ (when k is unimportant);

- (U-Intro) Given a type $A \in \mathrm{Ty}(\Gamma)$, there is a term

  $$\mathrm{En}_k(A) \in \mathrm{Tm}(\Gamma, \mathcal{U}_k)$$

  for some $k \in \mathbb{N}$, subsuming the one

  $$\mathrm{En}_k(\mathcal{U}_k^{[\Gamma]}) \in \mathrm{Tm}(\Gamma, \mathcal{U}_{k+1}^{[\Gamma]})$$

  for each $k \in \mathbb{N}$, where we often omit the subscript $(\_)_k$;

- (U-Elim) Each term $\psi \in \mathrm{Tm}(\Gamma, \mathcal{U}_k)$ induces a type

$$\mathrm{El}_k(\psi) \in \mathrm{Ty}(\Gamma),$$

  where we often omit the subscript $(\_)_k$;
- (U-Comp) $\mathrm{El}(\mathrm{En}(A)) = A$;
- (U-Cumul) If $\psi \in \mathrm{Tm}(\Gamma, \mathcal{U}_k)$, then $\psi \in \mathrm{Tm}(\Gamma, \mathcal{U}_{k+1})$;
- (U-Subst) $\mathcal{U}_k^{[\Gamma]}\{\phi\} = \mathcal{U}_k^{[\Delta]} \in \mathrm{Ty}(\Delta)$ for each morphism $\phi : \Delta \to \Gamma$;
- (En-Subst) $\mathrm{En}(A)\{\phi\} = \mathrm{En}(A\{\phi\}) \in \mathrm{Tm}(\Delta, \mathcal{U})$.

Note that the axiom U-Cumul requires the hierarchy $(\mathcal{U}_k)_{k\in\mathbb{N}}$ of universes $\mathcal{U}_k$ to be *cumulative*. For achieving game semantics of a cumulative hierarchy of universes, it suffices to equip the game-semantic CwF $\mathbb{WPG}_!$ with this semantic type-former because then the semantic type-former (by its *soundness*) automatically forms game semantics of the cumulative hierarchy of universes [Hof97].

In the rest of the present section, we first define p-games that interpret universes in §3.1 and then show that they form an instance of the semantic type-former of universes in §3.2.

3.1. Universe predicate games

As a preparation, let us recall the interpretation of Id-types in CwFs:

Definition 3.1.1 (categorical identity types [Hof97])**.** A CwF $\mathcal{C}$ is said to have *Id-types* if

- (Id-Form) Given an object $\Gamma \in \mathcal{C}$ and a type $A \in \mathrm{Ty}(\Gamma)$, there is a type

$$\mathrm{Id}_A \in \mathrm{Ty}(\Gamma.A.A^+),$$

  where $A^+ := A\{\mathrm{p}_A\} \in \mathrm{Ty}(\Gamma.A)$;
- (Id-Intro) There is a morphism

$$\mathrm{Refl}_A : \Gamma.A \to \Gamma.A.A^+.\mathrm{Id}_A$$

  that satisfies the equation

$$\mathrm{p}_{\mathrm{Id}_A} \circ \mathrm{Refl}_A = \overline{\mathrm{v}_A} : \Gamma.A \to \Gamma.A.A^+,$$

  where $\overline{\mathrm{v}_A} := \langle \mathrm{id}_{\Gamma.A}, \mathrm{v}_A \rangle$;
- (Id-Elim) Given a type $B \in \mathrm{Ty}(\Gamma.A.A^+.\mathrm{Id}_A)$ and a term $\beta \in \mathrm{Tm}(\Gamma.A, B\{\mathrm{Refl}_A\})$, there is a term

$$\mathcal{R}^{\mathrm{Id}}_{A,B}(\beta) \in \mathrm{Tm}(\Gamma.A.A^+.\mathrm{Id}_A, B);$$

- (Id-Comp) $\mathcal{R}^{\mathrm{Id}}_{A,B}(\beta)\{\mathrm{Refl}_A\} = \beta$;
- (Id-Subst) $\mathrm{Id}_A\{\phi^{++}_{A,A^+}\} = \mathrm{Id}_{A\{\phi\}} \in \mathrm{Ty}(\Delta.A\{\phi\}.A\{\phi\}^+)$ for all $\Delta \in \mathcal{C}$ and $\phi : \Delta \to \Gamma$, where $A\{\phi\}^+ := A\{\phi\}\{\mathrm{p}\} \in \mathrm{Ty}(\Delta.A\{\phi\})$, $\phi^+_A := \langle \phi \circ \mathrm{p}, \mathrm{v} \rangle_A : \Delta.A\{\phi\} \to \Gamma.A$ and $\phi^{++}_{A,A^+} := (\phi^+_A)^+_{A^+} : \Delta.A\{\phi\}.A\{\phi\}^+ \to \Gamma.A.A^+$;

- (Refl-Subst) $\mathrm{Refl}_A \circ \phi_A^+ = \phi^{+++}_{A,A^+,\mathrm{Id}_A} \circ \mathrm{Refl}_{A\{\phi\}} : \Delta.A\{\phi\} \to \Gamma.A.A^+.\mathrm{Id}_A$, where $\phi^{+++}_{A,A^+,\mathrm{Id}_A} := (\phi^{++}_{A,A^+})^+_{\mathrm{Id}_A} : \Delta.A\{\phi\}.A^+\{\phi^+\}.\mathrm{Id}_{A\{\phi\}} \to \Gamma.A.A^+.\mathrm{Id}_A$;
- ($\mathcal{R}^{\mathrm{Id}}$-Subst) $\mathcal{R}^{\mathrm{Id}}_{A,B}(\beta)\{\phi^{+++}_{A,A^+,\mathrm{Id}_A}\} = \mathcal{R}^{\mathrm{Id}}_{A\{\phi\},B\{\phi^{+++}_{A,A^+,\mathrm{Id}_A}\}}(\beta\{\phi_A^+\})$.

Then, for technical convenience, we employ the following recast Id' of the semantic Id-types in an arbitrary CwF $\mathcal{C}$. First, note that the Id-type $\mathrm{Id}_A \in \mathrm{Ty}(\Gamma.A.A^+)$ is equivalent to the family

$$\{\mathrm{Id}'_A(\alpha,\alpha')\}_{\alpha,\alpha' \in \mathrm{Tm}(\Gamma,A)}$$

of types $\mathrm{Id}'_A(\alpha,\alpha') \in \mathrm{Ty}(\Gamma)$: The former is recovered from the latter by

$$\mathrm{Id}_A := \mathrm{Id}'_{A^{++}}(\mathrm{v}\{\mathrm{p}\},\mathrm{v}),$$

where $A^{++} := A^+\{\mathrm{p}\} \in \mathrm{Ty}(\Gamma.A.A^+)$, and the latter from the former by

$$\mathrm{Id}'_A(\alpha,\alpha') := \mathrm{Id}_A\{\langle\langle \mathrm{id}_\Gamma, \alpha\rangle, \alpha'\rangle\}.$$

Next, the axiom Id-Subst implies the equation

$$\begin{aligned}
\mathrm{Id}'_A(\alpha,\alpha')\{\phi\} &= \mathrm{Id}_A\{\langle\langle \mathrm{id}_\Gamma, \alpha\rangle, \alpha'\rangle\}\{\phi\} \\
&= \mathrm{Id}_A\{\langle\langle \phi, \alpha\{\phi\}\rangle, \alpha'\{\phi\}\rangle\} \\
&= \mathrm{Id}_A\{\langle\langle \phi \circ \mathrm{p}, \mathrm{v}\rangle \circ \mathrm{p}, \mathrm{v}\rangle\}\{\langle\langle \mathrm{id}_\Gamma, \alpha\{\phi\}\rangle, \alpha'\{\phi\}\rangle\} \\
&= \mathrm{Id}_A\{\phi^{++}_{A,A^+}\}\{\langle\langle \mathrm{id}_\Gamma, \alpha\{\phi\}\rangle, \alpha'\{\phi\}\rangle\} \\
&= \mathrm{Id}_{A\{\phi\}}\{\langle\langle \mathrm{id}_\Gamma, \alpha\{\phi\}\rangle, \alpha'\{\phi\}\rangle\} \quad \text{(by Id-Subst)} \\
&= \mathrm{Id}'_{A\{\phi\}}(\alpha\{\phi\}, \alpha'\{\phi\})
\end{aligned}$$

for all $\alpha,\alpha' \in \mathrm{Tm}(\Gamma,A)$ and $\phi : \Delta \to \Gamma$, which we call the axiom Id'-Subst. Conversely, this axiom Id'-Subst implies the original axiom Id-Subst because

$$\begin{aligned}
\mathrm{Id}_A\{\phi^{++}_{A,A^+}\} &= \mathrm{Id}'_{A^{++}}(\mathrm{v}\{\mathrm{p}\},\mathrm{v})\{\langle \phi^+_A \circ \mathrm{p}, \mathrm{v}\rangle\} \\
&= \mathrm{Id}'_{A^{++}\{\langle\langle \phi\circ\mathrm{p},\mathrm{v}\rangle\circ\mathrm{p},\mathrm{v}\rangle\}}(\mathrm{v}\{\mathrm{p}\}\{\langle \phi^+_A \circ \mathrm{p}, \mathrm{v}\rangle\}, \mathrm{v}\{\langle \phi^+_A \circ \mathrm{p}, \mathrm{v}\rangle\}) \quad \text{(by Id'-Subst)} \\
&= \mathrm{Id}'_{A\{\phi\circ\mathrm{p}\circ\mathrm{p}\}}(\mathrm{v}\{\langle \phi \circ \mathrm{p} \circ \mathrm{p}, \mathrm{v}\{\mathrm{p}\}\rangle\}, \mathrm{v}) \\
&= \mathrm{Id}'_{A\{\phi\}^{++}}(\mathrm{v}\{\mathrm{p}\},\mathrm{v}) \\
&= \mathrm{Id}_{A\{\phi\}},
\end{aligned}$$

where $A\{\phi\}^{++} := A\{\phi\}^+\{\mathrm{p}\} \in \mathrm{Ty}(\Delta.A\{\phi\}.A\{\phi\}^+)$.

If we focus on the CwF $\mathbb{WPG}_!$ (Appendix A.2), then the axiom Id'-Subst implies the equation

$$\mathrm{Id}'_A(\alpha,\alpha')\{\gamma_0^\dagger\} = \mathrm{Id}'_{A(\gamma_0^\dagger)}(\alpha \bullet \gamma_0, \alpha' \bullet \gamma_0) \tag{4}$$

for all $\gamma_0^\dagger \in \mathbb{WPG}_!(!\Gamma)$. We leave it as a straightforward exercise to recast the remaining axioms of the semantic Id-types Id in such a way that they are adapted to this reformulation Id'. From now on, we write $\mathrm{Id}(\alpha,\alpha')$ for $\mathrm{Id}'(\alpha,\alpha')$ because we henceforth focus on the reformulation.

Then, as sketched in §2.4, the idea of our game semantics of universes is centred around the following *universe p-games*. We henceforth assume familiarity with the contents of Appendix A.

Definition 3.1.2 (universe predicate games)**.** Fix an arbitrary injection

$$\#_0 : \{1, 0, N, \Pi, \Sigma, \mathrm{Id}\} \rightarrowtail \mathbb{N}.$$

For each $k \in \mathbb{N}$, let $\{\mathcal{U}_k^{(i)}\}_{i\in\mathbb{N}}$ be the family of p-games $\mathcal{U}_k^{(i)}$ together with an arbitrary injection

$$\#_k : \{\, 1, 0, N, \Pi, \Sigma, \mathrm{Id}\,\} \uplus \{\, \mathcal{U}_j \mid j < k \,\} \rightarrowtail \mathbb{N}$$

that conservatively extends the one $\#_{k-1}$ defined inductively (on $i \in \mathbb{N}$) as follows:

1. (Base case) We define the p-game

$$\mathcal{U}_k^{(0)} := \mathscr{P}\big(\mathrm{Pref}(\{\, q^{\mathrm{OQ}}_{(0)}.\#_k(X)^{\mathrm{PA}}_{(0)} \mid X \in \{1, 0, N, \mathcal{U}_j\}, j < k \,\})\big),$$

where the first move $q^{\mathrm{OQ}}_{(0)}$ justifies the second one $\#_k(X)^{\mathrm{PA}}_{(0)}$,[4] together with the auxiliary map

$$\begin{aligned}\mathrm{El}_k^{(0)} : \mathbb{WPG}_!(!\mathcal{U}_k^{(0)}) &\to \mathrm{Ob}(\mathbb{WPG}_!)\\ \underline{\#_k(X)_{(0)}}^{\dagger} &\mapsto X \quad \text{(see Appendix A.1.10 for the notation } \underline{\#_k(X)_{(0)}}).\end{aligned}$$

Abusing notation, we lift this map to a dependent p-game $\mathrm{El}_k^{(0)} \in \mathscr{D}(\mathcal{U}_k^{(0)})$ by

$$|\mathrm{El}_k^{(0)}| := \bigcup_{\underline{\#_k(X)_{(0)}}^{\dagger} \in \mathbb{WPG}_!(!\mathcal{U}_k^{(0)})} \mathrm{El}_k^{(0)}(\underline{\#_k(X)_{(0)}}^{\dagger}), \qquad \|\mathrm{El}_k^{(0)}\| : \underline{\#_k(X)_{(0)}}^{\dagger} \mapsto \mathrm{El}_k^{(0)}(\underline{\#_k(X)_{(0)}}^{\dagger}).$$

Abusing notation again, we write $\mathcal{U}_k^{(0)}$ for the constant dependent p-game $\{\mathcal{U}_k^{(0)}\}$. Moreover, for each $\Gamma \in \mathbb{WPG}_!$, we lift the dependent p-game $\mathrm{El}_k^{(0)}$ to the map

$$\begin{aligned}\mathrm{El}_{k,\Gamma}^{(0)} : \mathbb{WPG}_!(\Gamma, \mathcal{U}_k^{(0)}) &\to \mathscr{D}(\Gamma)\\ \psi &\mapsto \mathrm{El}_{k,\Gamma}^{(0)}(\psi),\end{aligned}$$

where the dependent p-game $\mathrm{El}_{k,\Gamma}^{(0)}(\psi) \in \mathscr{D}(\Gamma)$ is given by

$$|\mathrm{El}_{k,\Gamma}^{(0)}(\psi)| := \bigcup_{\gamma_0^{\dagger} \in \mathbb{WPG}_!(!\Gamma)} \mathrm{El}_k^{(0)}(\psi \bullet \gamma_0), \qquad \|\mathrm{El}_{k,\Gamma}^{(0)}(\psi)\| : \gamma_0^{\dagger} \mapsto \mathrm{El}_k^{(0)}(\psi \bullet \gamma_0).$$

This map $\mathrm{El}_{k,\Gamma}^{(0)}$ generalises the dependent p-game $\mathrm{El}_k^{(0)}$ by the isomorphism $\mathrm{El}_{k,T}^{(0)} \cong \mathrm{El}_k^{(0)}$. The map is necessary for the inductive step given below. We often omit the subscript $(\_)_\Gamma$ in $\mathrm{El}_{k,\Gamma}^{(0)}$ when it does not bring confusion.

2. (Inductive step) We define the p-game

$$\begin{aligned}\mathcal{U}_k^{(i+1)} := \mathscr{P}\big(\mathcal{U}_k^{(i)} \cup \mathrm{Pref}(\{\, q^{\mathrm{OQ}}_{(i+1)}.\#_k(Y)^{\mathrm{PQ}}_{(i+1)}.\boldsymbol{s}\{\mathrm{OA/OQ}\} \mid Y \in \{\Pi, \Sigma\}, \boldsymbol{s} \in \Sigma(\mathcal{U}_k^{(i)}, \mathrm{El}_k^{(i)} \Rightarrow \mathcal{U}_k^{(i)}) \,\})\\ \cup \mathrm{Pref}(\{\, q^{\mathrm{OQ}}_{(i+1)}.\#_k(\mathrm{Id})^{\mathrm{PQ}}_{(i+1)}.\boldsymbol{t}\{\mathrm{OA/OQ}\} \mid \boldsymbol{t} \in \Sigma(\mathcal{U}_k^{(i)}, \mathrm{El}_k^{(i)} \,\&\, \mathrm{El}_k^{(i)}) \,\})\big),\end{aligned}$$

[4] For clarity, we add the subscripts $(\_)_{(i)}$ on moves in this definition though we do not in the introduction.

where the first move $q^{\mathrm{OQ}}_{(i+1)}$ justifies the second ones $\#_k(Y)^{\mathrm{PQ}}_{(i+1)}$ and $\#_k(\mathrm{Id})^{\mathrm{PQ}}_{(i+1)}$, the j-sequences $\boldsymbol{s}\{\mathrm{OA}/\mathrm{OQ}\}$ and $\boldsymbol{t}\{\mathrm{OA}/\mathrm{OQ}\}$ are obtained respectively from those $\boldsymbol{s}$ and $\boldsymbol{t}$ by the replacement of the labels OQ on moves with the ones OA, the second moves $\#_k(Y)^{\mathrm{PQ}}_{(i+1)}$ and $\#_k(\mathrm{Id})^{\mathrm{PQ}}_{(i+1)}$ justify the moves $q^{\mathrm{OA}}_{(i)}$ in $\boldsymbol{s}\{\mathrm{OA}/\mathrm{OQ}\}$ and $\boldsymbol{t}\{\mathrm{OA}/\mathrm{OQ}\}$, respectively, and the justifier $q^{\mathrm{OQ}}_{(i)}$ of all moves of the forms $\#_k(X)^{\mathrm{PQ}}_{(j)}$, $\#_k(Y)^{\mathrm{PQ}}_{(j)}$ and $\#_k(\mathrm{Id})^{\mathrm{PQ}}_{(j)}$ in $\boldsymbol{s}\{\mathrm{OA}/\mathrm{OQ}\}$ and $\boldsymbol{t}\{\mathrm{OA}/\mathrm{OQ}\}$ with $0 \leqslant j \leqslant i$ are replaced with the first move $q^{\mathrm{OQ}}_{(i+1)}$,[5] together with the map

$$\begin{aligned}
\mathrm{El}^{(i+1)}_k : \mathbb{WPG}_!(!\mathcal{U}^{(i+1)}_k) &\to \mathrm{Ob}(\mathbb{WPG}_!)\\
\underline{\#_k(X)_{(i+1)}}^{\dagger} &\mapsto X\\
q_{(i+1)}.\#_k(Y)_{(i+1)}.\langle \mu, \psi\rangle^{\dagger} &\mapsto Y(\mathrm{El}^{(i)}_k(\mu), \mathrm{El}^{(i)}_k(\psi))\\
q_{(i+1)}.\#_k(\mathrm{Id})_{(i+1)}.\langle \mu, \langle \alpha, \alpha'\rangle\rangle^{\dagger} &\mapsto \mathrm{Id}_{\mathrm{El}^{(i)}_k(\mu)}(\mathrm{El}^{(i)}_k(\alpha), \mathrm{El}^{(i)}_k(\alpha')),
\end{aligned}$$

where for any strategy σ and moves q and a we define

$$q.a.\sigma := \mathrm{Pref}(\{\, q.a.\boldsymbol{v} \mid \boldsymbol{v} \in \sigma \,\})^{\mathrm{Even}}$$

(with the evident justifications). Again, we lift this function $\mathrm{El}^{(i+1)}_k$ to a dependent p-game $\mathrm{El}^{(i+1)}_k \in \mathscr{D}(\mathcal{U}^{(i+1)}_k)$, and further to a map $\mathrm{El}^{(i+1)}_k : \mathbb{WPG}_!(\Gamma, \mathcal{U}^{(i+1)}_k) \to \mathscr{D}(\Gamma)$ ($\Gamma \in \mathbb{WPG}_!$) as in the case of $\mathrm{El}^{(0)}_k$; it is for the next inductive step. Abusing notation again, let $\mathcal{U}^{(i+1)}_k$ be the constant dependent p-game $\{\mathcal{U}^{(i+1)}_k\}$ and apply the notations for $\mathrm{El}^{(0)}_k$ to $\mathrm{El}^{(i+1)}_k$.

Finally, the $(k+1)$*st universe predicate (p-)game* is the constant p-game

$$\mathcal{U}_k := \mathscr{P}(|\mathcal{U}_k|)$$

on the game $|\mathcal{U}_k| := \bigcup_{i\in\mathbb{N}} |\mathcal{U}^{(i)}_k|$ equipped with the injection

$$\# := \bigcup_{k\in\mathbb{N}} \#_k : \{\, 1, 0, N, \Pi, \Sigma, \mathrm{Id} \,\} \uplus \{\, \mathcal{U}_j \mid j \in \mathbb{N} \,\} \rightarrowtail \mathbb{N}.$$

Convention. For simplicity, we henceforth omit the subscripts $\_{(i)}$ on moves in the universe p-games unless they are strictly necessary.

The inductive step in Definition 3.1.2 realises our idea on how to encode game semantics of Pi-, Sigma- and Id-types by strategies as sketched in §2.4. Specifically, we define the universe p-game $\mathcal{U}_k$ inductively through the smaller p-games $\mathcal{U}^{(i)}_k$ ($i \in \mathbb{N}$) along the simultaneous construction of the auxiliary map $\mathrm{El}^{(i)}_k$. This is a main technical highlight of the present work.

[5] By this adjustment of justifiers, the P-moves of the form $\#_k(\_)$ are all justified by the first move $q^{\mathrm{OQ}}_{(i+1)}$ in the p-game $\mathcal{U}^{(i+1)}_k$, so the p-game $\mathcal{U}_k$ is *well-founded* (Appendix A.1.7). The adjustment alone makes strategies on $\mathcal{U}_k$ *ill-bracketed* (Appendix A.2.11), but we fix this issue by the replacement of the labels OQ with those OA.

3.2. Game semantics of the cumulative hierarchy of universes

We need one more preparation for game semantics of universes. The axiom U-Intro (Definition 3.0.2) requires that each type A has its encoding $\mathrm{En}(A)$. As indicated in §2.4, however, we define the encoding inductively along the construction of types. Thus, we have to restrict types in the CwF $\mathbb{WPG}_!$ to those freely generated by the type constructions of MLTT, leading to:

Definition 3.2.1 ($\mathbb{UPG}_!$)**.** Let $\mathbb{UPG}_! \hookrightarrow \mathbb{WPG}_!$ be the substructural CwF of $\mathbb{WPG}_!$ such that

- The underlying category $\mathbb{UPG}_!$ is the category $\mathbb{WPG}_!$;
- The types of $\mathbb{UPG}_!$ are inductively constructed from the atomic dependent p-games 1, 0, N and $\mathcal{U}_k$ for all $k \in \mathbb{N}$ by the constructions Π, Σ and Id;
- The terms of $\mathbb{UPG}_!$ are given by

$$\mathrm{Tm}_{\mathbb{UPG}_!}(\Gamma, A) := \mathrm{Tm}_{\mathbb{WPG}_!}(\Gamma, A)$$

  for all $\Gamma \in \mathbb{UPG}_!$ and $A \in \mathrm{Ty}_{\mathbb{UPG}_!}(\Gamma)$.

Remark. We can make the CwF $\mathbb{UPG}_!$ *democratic* [CD14] by restricting its objects and morphisms to those inductively constructed from its types and terms, respectively. This democracy, however, takes complex mutual recursion as in the syntax of MLTT; for simplicity, we do not take this option.

Corollary 3.2.2 (well-defined $\mathbb{UPG}_!$)**.** *The structure $\mathbb{UPG}_!$ gives rise to a well-defined CwF that has One-, Zero-, N-, Pi-, Sigma- and Id-types in the same way as the CwF $\mathbb{WPG}_!$ (Appendix A.2).*

Proof. This corollary follows from Appendix A.2.18 (where the only nontrivial point is the closure of types $\mathbb{UPG}_!$ under substitution, but it is easily shown by induction on the types). □

In addition, this CwF $\mathbb{UPG}_!$ also has the cumulative hierarchy of universes:

Theorem 3.2.3 (game semantics of universes)**.** *The CwF $\mathbb{UPG}_!$ has universes.*

Proof. Let $\Delta, \Gamma \in \mathbb{UPG}_!$, $A \in \mathrm{Ty}_{\mathbb{UPG}_!}(\Gamma)$ and $\phi \in \mathbb{UPG}_!(\Delta, \Gamma)$.

- (U-Form) We have $\mathcal{U}_k^{[\Gamma]} \in \mathscr{D}(\Gamma)$ for each $k \in \mathbb{N}$ (Definition 3.1.2).
- (U-Intro) By the definition of $\mathbb{UPG}_!$, the type A is given inductively, so we can yield a term $\mathrm{En}(A) \in \mathrm{Tm}_{\mathbb{UPG}_!}(\Gamma, \mathcal{U}_{k(A)})$ for some $k_A \in \mathbb{N}$ inductively along the construction of A:
  1. If A is 1, 0 or N, then
  $$\mathrm{En}(A) := \underline{A} \in \mathrm{Tm}_{\mathbb{UPG}_!}(\Gamma, \mathcal{U}_0);$$
  2. If A is $\mathcal{U}_i$ for some $i \in \mathbb{N}$, then
  $$\mathrm{En}(\mathcal{U}_i) := \underline{\mathcal{U}_i} \in \mathrm{Tm}_{\mathbb{UPG}_!}(\Gamma, \mathcal{U}_{i+1});$$
  3. If A is $Y(B, C)$, where Y is Π or Σ, then
  $$\mathrm{En}(Y(B,C)) := q^{\mathrm{OQ}}.\#(Y)^{\mathrm{PA}}.\langle \mathrm{En}(B), \lambda \circ \mathrm{En}(C)\rangle \in \mathrm{Tm}_{\mathbb{UPG}_!}(\Gamma, \mathcal{U}_{\max(k_B, k_C)}),$$
  where λ is the *currying* for the introduction rule on Pi-types [Yam23, Lemma 4.5.1.2];

4. If A is $\mathrm{Id}_D(\delta,\delta')$, then

$$\mathrm{En}(\mathrm{Id}_D(\delta,\delta')) := q^{\mathrm{OQ}}.\#(\mathrm{Id})^{\mathrm{PA}}.\langle \mathrm{En}(D), \langle \delta,\delta'\rangle\rangle \in \mathrm{Tm}_{\mathbb{UPG}_!}(\Gamma, \mathcal{U}_{k_D}).$$

- (U-Elim) We define $\mathrm{El}_k : \mathrm{Tm}_{\mathbb{UPG}_!}(\Gamma, \mathcal{U}_k^{[\Gamma]}) \cong \mathbb{WPG}_!(\Gamma, \mathcal{U}_k) \to \mathscr{D}(\Gamma)$ to be the union

$$\mathrm{El}_k := \bigcup_{i\in\mathbb{N}} \mathrm{El}_k^{(i)} : \mathbb{WPG}_!(\Gamma, \mathcal{U}_k) \to \mathscr{D}(\Gamma)$$

up to the isomorphism

$$\mathrm{Tm}_{\mathbb{UPG}_!}(\Gamma, \mathcal{U}_k^{[\Gamma]}) \cong \mathbb{WPG}_!(\Gamma, \mathcal{U}_k),$$

where $\mathrm{El}_k^{(i)} : \mathbb{WPG}_!(\Gamma, \mathcal{U}_k^{(i)}) \to \mathscr{D}(\Gamma)$ is given in Definition 3.1.2. Note that the one $\mathrm{El}_k(\psi) \in \mathscr{D}(\Gamma)$ for each $\psi \in \mathrm{Tm}_{\mathbb{UPG}_!}(\Gamma, \mathcal{U}_k^{[\Gamma]})$ is given by

$$|\mathrm{El}_k(\psi)| := \bigcup_{\gamma_0^\dagger \in \mathbb{WPG}_!(!\Gamma)} \mathrm{El}_k(\psi \bullet \gamma_0), \qquad \|\mathrm{El}_k(\psi)\| : \gamma_0^\dagger \mapsto \mathrm{El}_k(\psi \bullet \gamma_0) := \Big(\bigcup_{i\in\mathbb{N}} \mathrm{El}_k^{(i)}\Big)(\psi \bullet \gamma_0).$$

- (U-Comp) We show the equation $\mathrm{El}(\mathrm{En}(A)) = A$ by induction on the type A, where we focus on the cases of $A = \Pi(B,C)$ and $A = \mathrm{Id}_D(\delta,\delta')$ since the other cases are similar or trivial:
  1. Assume $A = \Pi(B,C)$. The dependent p-game

$$\mathrm{El} \circ \mathrm{En}(\Pi(B,C)) = \mathrm{El}(q.\#(\Pi).\langle \mathrm{En}(B), \lambda \circ \mathrm{En}(C)\rangle)$$

consists of the underlying p-game

$$\begin{aligned}
|\mathrm{El} \circ \mathrm{En}(\Pi(B,C))| &= |\mathrm{El}(q.\#(\Pi).\langle \mathrm{En}(B), \lambda \circ \mathrm{En}(C)\rangle)| \\
&= \bigcup_{\gamma_0^\dagger \in \mathbb{UPG}_!(!\Gamma)} |\Pi(\mathrm{El}(\mathrm{En}(B) \bullet \gamma_0), \mathrm{El}(\lambda \circ \mathrm{En}(C) \bullet \gamma_0))| \\
&= \bigcup_{\gamma_0^\dagger \in \mathbb{UPG}_!(!\Gamma)} (|\mathrm{El}(\mathrm{En}(B) \bullet \gamma_0)| \Rightarrow |\mathrm{El}(\lambda \circ \mathrm{En}(C) \bullet \gamma_0)|) \\
&= \bigcup_{\gamma_0^\dagger \in \mathbb{UPG}_!(!\Gamma)} |\mathrm{El}(\mathrm{En}(B) \bullet \gamma_0)| \Rightarrow \bigcup_{\gamma_0^\dagger \in \mathbb{UPG}_!(!\Gamma)} |\mathrm{El}(\lambda \circ \mathrm{En}(C) \bullet \gamma_0)| \\
&= |\mathrm{El} \circ \mathrm{En}(B)| \Rightarrow |\mathrm{El} \circ \mathrm{En}(C)| \\
&= |B| \Rightarrow |C| \quad \text{(by the induction hypothesis)} \\
&= |\Pi(B,C)|
\end{aligned}$$

and the function

$$\begin{aligned}
\|\mathrm{El} \circ \mathrm{En}(\Pi(B,C))\| : \gamma_0^\dagger \in \mathbb{UPG}_!(!\Gamma) &\mapsto \mathrm{El}(q.\#(\Pi).\langle \mathrm{En}(B) \bullet \gamma_0, \lambda \circ \mathrm{En}(C) \bullet \gamma_0\rangle) \\
&= \Pi(\mathrm{El}(\mathrm{En}(B) \bullet \gamma_0), \mathrm{El}(\lambda \circ \mathrm{En}(C) \bullet \gamma_0)) \\
&= \Pi(\mathrm{El} \circ \mathrm{En}(B)(\gamma_0^\dagger), \mathrm{El} \circ \mathrm{En}(C)_{\gamma_0^\dagger}) \\
&= \Pi(B(\gamma_0^\dagger), C_{\gamma_0^\dagger}) \quad \text{(by the induction hypothesis)} \\
&= \Pi(B,C)(\gamma_0^\dagger).
\end{aligned}$$

Hence, we have shown

$$\mathrm{El} \circ \mathrm{En}(\Pi(B, C)) = \Pi(B, C).$$

2. Assume $A = \mathrm{Id}_D(\delta, \delta')$. The dependent p-game

$$\mathrm{El} \circ \mathrm{En}(\mathrm{Id}_D(\delta, \delta')) = \mathrm{El}(q.\#(\mathrm{Id}).\langle \mathrm{En}(D), \langle \delta, \delta' \rangle \rangle)$$

consists of the underlying p-game

$$\begin{aligned} |\mathrm{El} \circ \mathrm{En}(\mathrm{Id}_D(\delta, \delta'))| &= |\mathrm{El}(q.\#(\mathrm{Id}).\langle \mathrm{En}(D), \langle \delta, \delta' \rangle \rangle)| \\ &= \bigcup_{\gamma_0^\dagger \in \mathbb{UPG}_!(!\Gamma)} |\mathrm{Id}_{\mathrm{El}(\mathrm{En}(D) \bullet \gamma_0)}(\delta \bullet \gamma_0, \delta' \bullet \gamma_0)| \\ &= T' \quad \text{(see Appendix A.2.22)} \\ &= |\mathrm{Id}_D(\delta, \delta')| \end{aligned}$$

and the function

$$\begin{aligned} \|\mathrm{Id}_D(\delta, \delta')\| : \gamma_0^\dagger \in \mathbb{UPG}_!(!\Gamma) &\mapsto \mathrm{El}(q.\#(\mathrm{Id}).\langle \mathrm{En}(D) \bullet \gamma_0, \langle \delta \bullet \gamma_0, \delta' \bullet \gamma_0 \rangle \rangle) \\ &= \mathrm{Id}_{\mathrm{El}(\mathrm{En}(D) \bullet \gamma_0)}(\delta \bullet \gamma_0, \delta' \bullet \gamma_0) \\ &= \mathrm{Id}_{\mathrm{El} \circ \mathrm{En}(D)(\gamma_0^\dagger)}(\delta \bullet \gamma_0, \delta' \bullet \gamma_0) \\ &= \mathrm{Id}_{D(\gamma_0^\dagger)}(\delta \bullet \gamma_0, \delta' \bullet \gamma_0) \quad \text{(by the induction hypothesis)} \\ &= \mathrm{Id}_D(\delta, \delta')(\gamma_0^\dagger) \quad \text{(by the equation (4)).} \end{aligned}$$

Hence, we have shown

$$\mathrm{El} \circ \mathrm{En}(\mathrm{Id}_D(\delta, \delta')) = \mathrm{Id}_D(\delta, \delta').$$

- (U-Cumul) By construction, $\psi \in \mathrm{Tm}_{\mathbb{UPG}_!}(\Gamma, \mathcal{U}_k)$ implies $\psi \in \mathrm{Tm}_{\mathbb{UPG}_!}(\Gamma, \mathcal{U}_{k+1})$.
- (U-Subst) By construction, the equation $\mathcal{U}_k^{[\Gamma]}\{\phi\} = \mathcal{U}_k^{[\Delta]} \in \mathrm{Ty}(\Delta)$ holds.
- (En-Subst) We see that the equation $\mathrm{En}(A)\{\phi\} = \mathrm{En}(A\{\phi\}) \in \mathrm{Tm}(\Delta, \mathcal{U})$ holds by induction on the type A, where again we focus on the cases of $A = \Pi(B, C)$ and $A = \mathrm{Id}_D(\delta, \delta')$:
  1. Assume $A = \Pi(B, C)$. We have the equation

$$\begin{aligned} \mathrm{En}(\Pi(B, C))\{\phi\} &= q.\#(\Pi).\langle \mathrm{En}(B) \bullet \phi, \lambda \circ \mathrm{En}(C) \bullet \phi \rangle \\ &= q.\#(\Pi).\langle \mathrm{En}(B\{\phi\}), \lambda \circ \mathrm{En}(C\{\phi_B^+\}) \rangle \quad \text{(by the induction hypothesis)} \\ &= \mathrm{En}(\Pi(B\{\phi\}, C\{\phi_B^+\})) \\ &= \mathrm{En}(\Pi(B, C)\{\phi\}) \quad \text{(by the equation (A.1)).} \end{aligned}$$

  2. Assume $A = \mathrm{Id}_D(\delta, \delta')$. We have the equation

$$\begin{aligned} \mathrm{En}(\mathrm{Id}_D(\delta, \delta'))\{\phi\} &= q.\#(\mathrm{Id}).\langle \mathrm{En}(D) \bullet \phi, \langle \delta \bullet \phi, \delta' \bullet \phi \rangle \rangle \\ &= q.\#(\mathrm{Id}).\langle \mathrm{En}(D\{\phi\}), \langle \delta\{\phi\}, \delta'\{\phi\} \rangle \rangle \quad \text{(by the induction hypothesis)} \\ &= \mathrm{En}(\mathrm{Id}_{D\{\phi\}}(\delta\{\phi\}, \delta'\{\phi\})) \\ &= \mathrm{En}(\mathrm{Id}_D(\delta, \delta')\{\phi\}) \quad \text{(by Id'-Subst).} \end{aligned}$$

We have verified all the required axioms, completing the proof. □

Remark. An Id-type $\mathrm{Id}_A(\alpha, \alpha')$ is the 'trivially true' one 1 if $\alpha = \alpha'$, and the 'trivially false' one 0 otherwise (Appendix A.2.22), but the strategy α or α' is not part of $\mathrm{Id}_A(\alpha, \alpha')$. One may thus wonder if the encoding $\mathrm{En}(\mathrm{Id}_A(\alpha, \alpha'))$ might not be well-defined. Nevertheless, it *is* well-defined, where the point is that En is *not* a function, and $\mathrm{En}(\mathrm{Id}_A(\alpha, \alpha'))$ is assigned *along the inductive construction of* $\mathrm{Id}_A(\alpha, \alpha')$.[6] The general definition of universes in a CwF (Definition 3.0.2) does not require En to be a function, so there is no problem in the definition of our encoding En.

Example 3.2.4. Let us consider the interpretation of the encoding

$$f : N \Rightarrow N, g : N \Rightarrow N \vdash \mathrm{En}_0(\mathrm{Id}_{N\Rightarrow N}(f, g)) : \mathcal{U}_0$$

of the Id-type discussed in §2.3. The strategy

$$\psi := \mathrm{En}_0(\mathrm{Id}_{N\Rightarrow N}(\pi_1, \pi_2)) : (N \Rightarrow N)\,\&\,(N \Rightarrow N) \to \mathcal{U}_0,$$

which interprets this encoding of the Id-type, plays as in Figure 4. We note that Figure 4 is just a slightly more precise description of Figure 3.

Example 3.2.5. Let us recall the rules for N type:

$$(N\text{-Form})\ \frac{}{\Gamma \vdash N\ \mathrm{type}} \qquad (N\text{-IntroZ})\ \frac{}{\Gamma \vdash \mathrm{zero} : N} \qquad (N\text{-IntroS})\ \frac{\Gamma \vdash t : N}{\Gamma \vdash \mathrm{succ}(t) : N}$$

$$(N\text{-Elim})\ \frac{\Gamma, x : N \vdash C\ \mathrm{type} \qquad \Gamma \vdash c_z : C\{\mathrm{zero}/x\} \qquad \Gamma, x : N, y : C \vdash c_s : C\{\mathrm{succ}(x)/x\} \qquad \Gamma \vdash t : N}{\Gamma \vdash R^N(C, c_z, c_s, t) : C\{t/x\}}$$

$$(N\text{-CompZ})\ \frac{\Gamma, x : N \vdash C\ \mathrm{type} \qquad \Gamma \vdash c_z : C\{\mathrm{zero}/x\} \qquad \Gamma, x : N, y : C \vdash c_s : C\{\mathrm{succ}(x)/x\}}{\Gamma \vdash R^N(C, c_z, c_s, \mathrm{zero}) = c_z : C\{\mathrm{zero}/x\}}$$

$$(N\text{-CompS})\ \frac{\Gamma, x : N \vdash C\ \mathrm{type} \qquad \Gamma \vdash c_z : C\{\mathrm{zero}/x\} \qquad \Gamma, x : N, y : C \vdash c_s : C\{\mathrm{succ}(x)/x\} \qquad \Gamma \vdash t : N}{\Gamma \vdash R^N(C, c_z, c_s, \mathrm{succ}(t)) = c_s\{t/x\}\{R^N(C, c_z, c_s, t)/y\} : C\{\mathrm{succ}(t)/x\}}$$

The elimination rule N-Elim with respect to a universe generates *transfinite* dependent types. For instance, the encoding of the dependent type

$$x : N \vdash \mathrm{List}_N(x)\ \mathrm{type}$$

of finite lists of natural numbers, which satisfies

$$\mathrm{List}_N(\underline{0}) = 1, \qquad \mathrm{List}_N(\underline{n+1}) = \mathrm{List}_N(\underline{n}) \times N,$$

is defined by applying N-Elim, where $\Gamma := \boldsymbol{\epsilon}$ (*i.e.*, the empty context) and $C := \mathcal{U}$, to the terms

$$\vdash c_z := \mathrm{En}(1) : \mathcal{U}, \qquad x : N, y : \mathcal{U} \vdash c_s := \mathrm{En}(\mathrm{El}(y) \times N) : \mathcal{U}.$$

Then, the strategy

$$\psi' := \mathscr{R}^N_{\mathcal{U}_0}(\mathrm{En}(1), \mathrm{En}(\mathrm{El}(\pi_2)\,\&\,N)) : N \to \mathcal{U}_0,$$

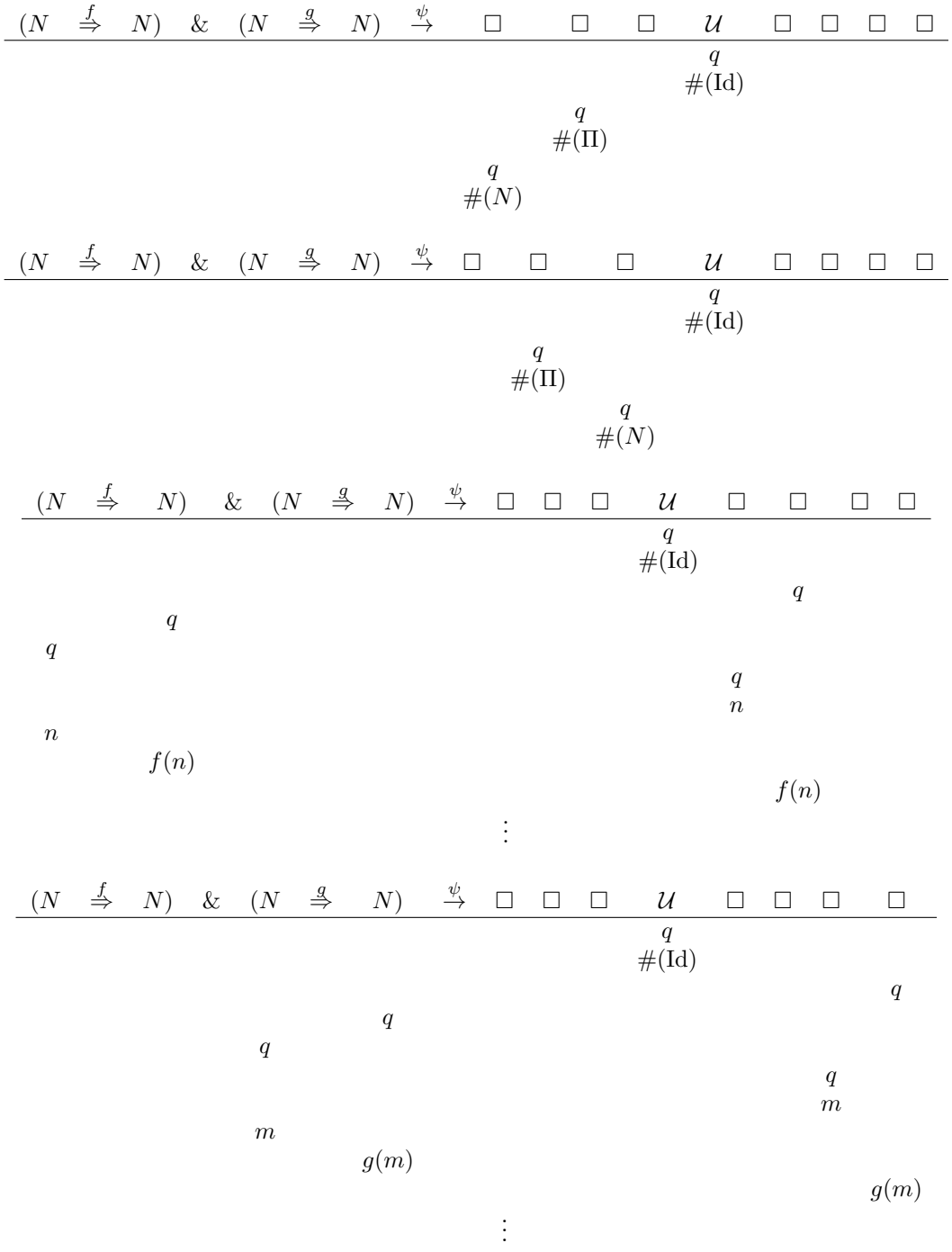

Figure 4: The strategy on the encoding of the Id-type between functions

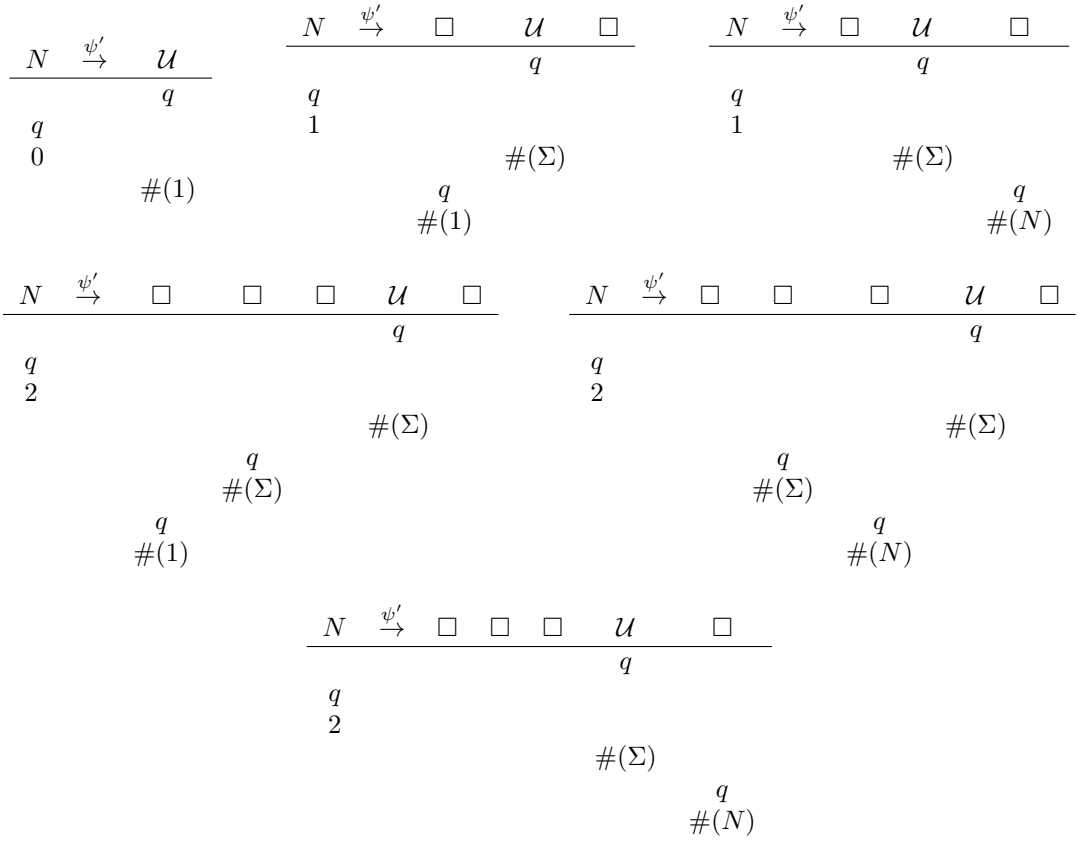

Figure 5: The strategy on the encoding of the list type

which interprets the encoding of the list type, plays as in Figure 5, where $\mathscr{R}^N$ is the game-semantic constructor that interprets the elimination rule N-ELIM [Yam23, Theorem 4.5.3.3].

This list type is out of the scope of Abramsky *et al.* [AJV15, VJA18] since their semantics is limited to *finite inductive types* [VJA18, Figure 7], which exclude the list type [Yam23, §4.3]. This implies that their approach cannot interpret the combination of universes and N-type.

4. Corollaries

This last section presents some corollaries of our game semantics of universes (Theorem 3.2.3). The first corollary is the *effective computability* of the game semantics (§4.1), the second one is the independence of the principle of *universe reflection* from MLTT equipped with universes (§4.2), and the last one is the independence of *Markov's principle* from MLTT together with universes (§4.3). We emphasise that these corollaries come from the unique *intensionality* of the game semantics.

4.1. Effective computability

We first show the *effective computability* of our game semantics of universes. Recall that strategies in the CwF $\mathbb{UPG}_!$ are just the conventional ones (Appendix A.1) that are *winning* and *well-bracketed* (Appendix A.2.11). Recall also that more unrestricted strategies that interpret terms in the higher-order functional programming language PCF [Sco93, Plo77] are all effective or *recursive*; see the original articles [AJM00, §5] and [HO00, §5.6] for the details. Thus, terms and morphisms in $\mathbb{UPG}_!$ are a class of winning, well-bracketed strategies in the game semantics of PCF that satisfy the additional condition of *strategy filtering* imposed by p-games (§2.2). Therefore, the definition of recursive strategies is directly applicable to terms and morphisms in $\mathbb{UPG}_!$.

Roughly, if moves are all encodable by natural numbers, which is clearly the case for Yamada's game semantics of MLTT, then a strategy is said to be *recursive* if its computational steps

$$\boldsymbol{s}a \mapsto \boldsymbol{s}ab$$

at the level of positions are all computable (with respect to the encoding of moves) in the standard sense [Rog67]. Again, see the aforementioned references for the details. We then define:

Definition 4.1.1 ($\mathbb{UPG}_!^{\mathrm{eff}}$)**.** Let $\mathbb{UPG}_!^{\mathrm{eff}} \hookrightarrow \mathbb{UPG}_!$ be the wide substructural CwF of $\mathbb{UPG}_!$ whose terms and morphisms are all recursive.

Because the strategies in $\mathbb{UPG}_!$ modelling terms in MLTT are more restricted than those modelling terms in PCF, it is just straightforward[7] to verify:

Corollary 4.1.2 (effective game semantics of universes)**.** *The CwF $\mathbb{UPG}_!^{\mathrm{eff}}$ is well-defined and has One-, Zero-, N-, Pi-, Sigma- and Id-types as well as the cumulative hierarchy of universes in the same way as the CwF $\mathbb{UPG}_!$ (Theorems 3.2.3 and Appendix A.2). This in particular establishes effective game semantics of universes.*

[6] For analogy, an *interpretation* of terms is given along the inductive construction of terms, and the interpretation is *a priori* not a map defined on terms themselves. The inductive construction of terms does not add any data to terms, but it makes the interpretation well-defined. In this sense, the interpretation is similar to the assignment En.

[7] Again, the point is that our strategies are just the conventional ones, so the existing arguments on the recursive nature of strategies interpreting PCF such as [AJM00, §5] and [HO00, §5.6] are applicable here.

4.2. Independence of universe reflection

Next, recall the principle of *universe reflection* [ML75, Pal98]: Given terms $\psi, \psi' \in \mathrm{Tm}(\Gamma, \mathcal{U})$, if

$$\mathrm{El}(\psi) = \mathrm{El}(\psi') \in \mathrm{Ty}(\Gamma),$$

then

$$\psi = \psi'.$$

Let us show that this principle is *independent* from MLTT. To this end, a key observation is that, by the *intensionality* of our game semantics, there can be more than one term that encodes the same type in the CwF $\mathbb{UPG}_!^{(\mathrm{eff})}$. In other words, the game semantics *refutes* universe reflection.

For instance, the term $\mathrm{En}(1) \in \mathrm{Tm}(T.N, \mathcal{U})$, which encodes One-type $1 \in \mathrm{Ty}(T.N)$ in $\mathbb{UPG}_!^{(\mathrm{eff})}$, plays by

$$\begin{array}{ccc} T.N & \overset{\mathrm{En}(1)}{\to} & \mathcal{U} \\ \hline & & q \\ & & \#(1) \end{array}$$

while another term $\psi \in \mathrm{Tm}(T.N, \mathcal{U})$, which plays by

$$\begin{array}{ccc} T.N & \overset{\psi}{\to} & \mathcal{U} \\ \hline & & q \\ q & & \\ n & & \\ & & \#(1) \end{array}$$

for all $n \in \mathbb{N}$, also encodes the same type. We note that the latter term ψ can be given by the interpretation $\mathscr{R}^N$ of the elimination rule N-Elim of N-type in $\mathbb{UPG}_!^{(\mathrm{eff})}$ applied to the constant terms $\underline{\#(1)} \in \mathrm{Tm}(T, \mathcal{U})$ and $\underline{\#(1)} \in \mathrm{Tm}(T.N.\mathcal{U}, \mathcal{U})$ in $\mathbb{UPG}_!^{(\mathrm{eff})}$.

This argument together with Theorem 3.2.3 immediately implies:

Corollary 4.2.1 (independence of universe reflection)**.** *The principle of universe reflection is independent from MLTT equipped with One-, Zero-, N-, Pi-, Sigma- and Id-types as well as the cumulative hierarchy of universes.*

The *intensionality* of strategies plays a vital role for this argument, while it is not available for other computational models such as those based on domains [Pal93, BL18] and realisability [Str91].

4.3. Independence of Markov's principle

Finally, recall that Yamada [Yam23, §4.7] proved that *Markov's principle* [Mar62] is invalid in his game semantics, so the principle is *independent* from MLTT equipped with One-, Zero-, N-, Pi-, Sigma- and Id-types. Markov's principle is a well-known one in constructive mathematics, and it depends on the school of constructive mathematics whether the principle is to be regarded as *constructive*. The principle postulates that if it is impossible that there is no natural number $n \in \mathbb{N}$ with $f(n) = 0$ for a map $f : \mathbb{N} \to \mathbb{N}$, then there *is* a natural number $n_0 \in \mathbb{N}$ with $f(n_0) = 0$.

To see the subtlety of Markov's principle, we note that Markov's principle forms an instance of the law of *double negation elimination*, which is available in classical logic but not in intuitionistic logic [TS00]. In this sense, the principle appears *non-constructive*. On the other hand, Markov

described an algorithm that validates the principle within his *recursive* school of constructive mathematics [TvD88, §4.5]. In this way, it depends on the choice of a school of constructive mathematics whether or not one accepts Markov's principle as constructive. In this context, it is an interesting problem to see if the principle is independent from (some extensions of) MLTT.

Clearly, the independence proof due to Yamada is also valid for our game semantics *without any modification*. This extends the independence result to universes:

Corollary 4.3.1 (independence of Markov's principle from universes)**.** *Markov's principle is independent from MLTT equipped with One-, Zero-, N-, Pi-, Sigma- and Id-types as well as the cumulative hierarchy of universes.*

Proof. It is instructive to recall Yamada's proof of the independence result [Yam23, Corollary 4.7.1] below, where we replace the CwF $\mathbb{WPG}_!$ with the one $\mathbb{UPG}_!^{(\mathrm{eff})}$.

First, recall that the p-game that interprets Markov's principle is

$$\Pi(N_{[0]} \Rightarrow N_{[1]}, D \Rightarrow C), \tag{5}$$

where

$$\begin{aligned} C &:= \Sigma(N_{[6]}, \mathrm{Id}_N\{\langle \mathrm{app}(\pi_1, \pi_2), \underline{0}\rangle\}_{[7]})\{\mathrm{p}\}, \\ D &:= \big(\Sigma(N_{[2]}, \mathrm{Id}_N\{\langle \mathrm{app}(\pi_1, \pi_2), \underline{0}\rangle\}_{[3]}) \Rightarrow 0_{[4]}\big) \Rightarrow 0_{[5]}, \end{aligned}$$

and omit the terminal p-game T and the bracket $\{\_\}$ for constant dependent p-games (*e.g.*, write N for $T.\{N\}$). We call C the *codomain* of the p-game (5), $N_{[0]} \Rightarrow N_{[1]}$ the *outer domain*, and D the *inner domain*. Assume for a contradiction that there is a term $\langle \phi, \psi\rangle$ on this p-game in $\mathbb{UPG}_!^{(\mathrm{eff})}$.

We then proceed by the following case analysis. Assume that there are *total* input strategies

$$\varphi^\dagger : !(N_{[0]} \Rightarrow N_{[1]}), \qquad \langle \underline{n}, \underline{\surd}\rangle^\dagger : !\Sigma(N_{[2]}, \mathrm{Id}_N\{\langle \mathrm{app}(\pi_1, \pi_2), \underline{0}\rangle\}_{[3]})$$

on which ϕ eventually makes a P-move n'_φ in $N_{[6]}$ if Opponent begins a play in $N_{[6]}$.

1. If $\varphi \bullet \underline{n'_\varphi} \neq \underline{0}$ for *some* of φ and n, and Opponent plays by them, then $\mathrm{Id}_N\{\langle \mathrm{app}(\pi_1, \pi_2), \underline{0}\rangle\}_{[7]}$ can be the empty one 0, and ψ plays on the domains *forever*, contradicting its *noetherianity*;
2. If $\varphi \bullet \underline{n'_\varphi} = \underline{0}$ for *all* φ and n, then ϕ is *strict*[8] since otherwise n'_φ would be the same even if Opponent changes φ so that $\varphi \bullet \underline{n'_\varphi} \neq \underline{0}$, a contradiction; thus, ϕ is strict and answers the question in $N_{[6]}$ with no answer to the question in $0_{[4]}$, contradicting the *well-bracketing* of ϕ.

Hence, we conclude that, given any total inputs $\varphi^\dagger$ and $\langle \underline{n}, \underline{\surd}\rangle^\dagger$, ϕ does not make a P-move in $N_{[6]}$. Similarly to the case of ψ, however, this contradicts the *noetherianity* of ϕ. □

Again, this proof takes advantages of the *intensional* nature of the game semantics, which is not available for other computational models of MLTT such as domain theory and realisability. The *intuitionistic* nature of the game semantics realised by the well-bracketing of strategies is also crucial in the proof; it is unavailable in the denotational semantics due to Blot and Laird [BL18].

Mannaa and Coquand [MC17] established the independence of Markov's principle from MLTT equipped with a universe for the first time in the literature. Their proof is *syntactic*, which stands in

[8] A morphism $\phi : \Gamma \to \Delta$ is said to be *strict* if $mn \in \phi$ implies $n \in \Gamma$ [Lau02, p. 88].

contrast to our game-semantic proof. As we have already mentioned, their proof is not automatically extendable to other types, and an extension can be nontrivial. In contrast, our reasoning is *modular*: A meta-theoretic result on MLTT given by our game semantics is automatically extended to new types as soon as the game semantics is extended to the types. This is a strong advantages of our game-semantic approach to the study of type theory and constructive mathematics.

Acknowledgements

The author acknowledges financial support from the Centre for Mathematics of the University of Coimbra (under the FCT project UID/00324) and from FCT (contract 2022.06122.CEECIND).

References

[A+97] Samson Abramsky et al., *Semantics of interaction: An introduction to game semantics*, Semantics and Logics of Computation **14** (1997), 1–31.

[Acz86] Peter Aczel, *The type theoretic interpretation of constructive set theory: inductive definitions*, Studies in Logic and the Foundations of Mathematics, vol. 114, Elsevier, 1986, pp. 17–49.

[AHM98] Samson Abramsky, Kohei Honda, and Guy McCusker, *A fully abstract game semantics for general references*, Logic in Computer Science, 1998. Proceedings. Thirteenth Annual IEEE Symposium on, IEEE, 1998, pp. 334–344.

[AJ94] Samson Abramsky and Radha Jagadeesan, *Games and full completeness for multiplicative linear logic*, The Journal of Symbolic Logic **59** (1994), no. 02, 543–574.

[AJM00] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria, *Full abstraction for PCF*, Information and Computation **163** (2000), no. 2, 409–470.

[AJV15] Samson Abramsky, Radha Jagadeesan, and Matthijs Vákár, *Games for dependent types*, Automata, Languages, and Programming, Springer, Berlin, Heidelberg, 2015, pp. 31–43.

[AM97] Samson Abramsky and Guy McCusker, *Linearity, sharing and state: a fully abstract game semantics for Idealized Algol with active expressions*, Algol-like languages, Springer, 1997, pp. 297–329.

[AM99a] ———, *Full abstraction for Idealized Algol with passive expressions*, Theoretical Computer Science **227** (1999), no. 1, 3–42.

[AM99b] ———, *Game semantics*, Computational Logic: Proceedings of the 1997 Marktoberdorf Summer School (Berlin, Heidelberg), Springer, 1999, pp. 1–55.

[BL18] Valentin Blot and Jim Laird, *Extensional and intensional semantic universes: A denotational model of dependent types*, Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, ACM, 2018, pp. 95–104.

[Bro54] LEJ Brouwer, *Points and spaces*, Canadian Journal of Mathematics **6** (1954), 1–17.

[CAB+86] RL Constable, SF Allen, HM Bromley, WR Cleaveland, JF Cremer, RW Harper, DJ Howe, TB Knoblock, NP Mendler, P Panangaden, et al., *Implementing mathematics with the nuprl proof development system*.

[CD14] Pierre Clairambault and Peter Dybjer, *The biequivalence of locally cartesian closed categories and martin-löf type theories*, Mathematical Structures in Computer Science **24** (2014), no. 6, e240606.

[CH10] Pierre Clairambault and Russ Harmer, *Totality in arena games*, Annals of pure and applied logic **161** (2010), no. 5, 673–689.

[Chr00] Juliusz Chroboczek, *Game semantics and subtyping*, Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No. 99CB36332), IEEE, 2000, pp. 192–203.

[Chu40] Alonzo Church, *A formulation of the simple theory of types*, The journal of symbolic logic **5** (1940), no. 2, 56–68.

[Cla09] Pierre Clairambault, *Least and greatest fixpoints in game semantics*, International Conference on Foundations of Software Science and Computational Structures, Springer, 2009, pp. 16–31.

[Coq95] Thierry Coquand, *A semantics of evidence for classical arithmetic*, The Journal of Symbolic Logic **60** (1995), no. 1, 325–337.

[DP16] Peter Dybjer and Erik Palmgren, *Intuitionistic type theory*, Stanford Encyclopedia of Philosophy (2016).

[Dyb96] Peter Dybjer, *Internal Type Theory*, Types for Proofs and Programs, Springer, 1996, pp. 120–134.

[Fra22] Adolf Fraenkel, *Zu den grundlagen der cantor-zermeloschen mengenlehre*, Mathematische annalen **86** (1922), no. 3-4, 230–237.

[Gir72] Jean-Yves Girard, *Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur*, Ph.D. thesis, Éditeur inconnu, 1972.

[Gir87] ———, *Linear logic*, Theoretical Computer Science **50** (1987), no. 1, 1–101.

[GR94] Edward Griffor and Michael Rathjen, *The strength of some martin-löf type theories*, Archive for Mathematical Logic **33** (1994), no. 5, 347–385.

[Hey31] Arend Heyting, *Die intuitionistische grundlegung der mathematik*, Erkenntnis **2** (1931), no. 1, 106–115.

[HO00] J Martin E Hyland and C-HL Ong, *On full abstraction for PCF: I, II, and III*, Information and Computation **163** (2000), no. 2, 285–408.

[Hof97] Martin Hofmann, *Syntax and Semantics of Dependent Types*, Extensional Constructs in Intensional Type Theory, Springer, 1997, pp. 13–54.

[Hyl82] J Martin E Hyland, *The effective topos*, Studies in Logic and the Foundations of Mathematics, vol. 110, Elsevier, 1982, pp. 165–216.

[Hyl97] Martin Hyland, *Game semantics*, Semantics and Logics of Computation, vol. 14, Cambridge University Press, New York, 1997, p. 131.

[Kol32] Andrej Kolmogoroff, *Zur deutung der intuitionistischen logik*, Mathematische Zeitschrift **35** (1932), no. 1, 58–65.

[Lai97] James Laird, *Full abstraction for functional languages with control*, Logic in Computer Science, 1997. LICS'97. Proceedings., 12th Annual IEEE Symposium on, IEEE, 1997, pp. 58–67.

[Lau02] Olivier Laurent, *Polarized games*, Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science, IEEE, 2002, pp. 265–274.

[Mar62] Andrei Andreevich Markov, *On constructive mathematics*, Trudy Matematicheskogo Instituta imeni VA Steklova **67** (1962), 8–14.

[MC17] Bassel Mannaa and Thierry Coquand, *The independence of markov's principle in type theory*, Logical Methods in Computer Science **13** (2017).

[McC98] Guy McCusker, *Games and full abstraction for a functional metalanguage with recursive types*, Springer Science & Business Media, London, 1998.

[ML75] Per Martin-Löf, *An Intuitionistic Theory of Types: Predicative Part*, Studies in Logic and the Foundations of Mathematics **80** (1975), 73–118.

[ML82] _____, *Constructive Mathematics and Computer Programming*, Studies in Logic and the Foundations of Mathematics **104** (1982), 153–175.

[ML84] _____, *Intuitionistic Type Theory: Notes by Giovanni Sambin of a series of lectures given in Padova, June 1980*, 1984.

[ML98] _____, *An Intuitionistic Theory of Types*, Twenty-five years of constructive type theory **36** (1998), 127–172.

[Pal93] Erik Palmgren, *An information system interpretation of martin-löf's partial type theory with universes*, Information and Computation **106** (1993), no. 1, 26–60.

[Pal98] _____, *On universes in type theory*, Twenty five years of constructive type theory (1998), 191–204.

[Plo77] Gordon D. Plotkin, *Lcf considered as a programming language*, Theoretical computer science **5** (1977), no. 3, 223–255.

[Rog67] Jr Rogers, Hartley, *Theory of recursive functions and effective computability*, vol. 5, McGraw-Hill, New York, 1967.

[Sco70] Dana Scott, *Outline of a mathematical theory of computation*, Oxford University Computing Laboratory, Programming Research Group Oxford, 1970.

[Sco93] Dana S Scott, *A type-theoretical alternative to iswim, cuch, owhy*, Theoretical Computer Science **121** (1993), no. 1-2, 411–440.

[Set93] Anton Setzer, *Proof theoretical strength of martin-löf type theory with w-type and one universe*, Ph.D. thesis, Uitgever niet vastgesteld, 1993.

[Sho67] Joseph R Shoenfield, *Mathematical logic*, vol. 21, Addison-Wesley, Reading, 1967.

[Str91] Thomas Streicher, *Semantics of Type Theory: Correctness, Completeness and Independence Results*, Springer Science & Business Media, 1991.

[SU06] Morten Heine Sørensen and Pawel Urzyczyn, *Lectures on the curry-howard isomorphism*, Elsevier, 2006.

[Tar54] Alfred Tarski, *Contributions to the theory of models. i*, Indagationes Mathematicae (Proceedings), vol. 57, Elsevier, 1954, pp. 572–581.

[TS00] Anne Sjerp Troelstra and Helmut Schwichtenberg, *Basic proof theory*, no. 43, Cambridge University Press, 2000.

[TvD88] Anne Sjerp Troelstra and Dirk van Dalen, *Constructivism in mathematics. two volumes*, NorthHolland, Amsterdam (1988).

[Uni13] The Univalent Foundations Program, *Homotopy type theory: Univalent foundations of mathematics*, `https://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.

[VJA18] Matthijs Vákár, Radha Jagadeesan, and Samson Abramsky, *Game semantics for dependent types*, Information and Computation **261** (2018), 401–431.

[Yam19] Norihiro Yamada, *A game-semantic model of computation*, Research in the Mathematical Sciences **6** (2019), no. 1, 3.

[Yam23] ———, *Game semantics of Martin-Löf type theory*, Mathematical Structures in Computer Science (2023), 1–42.

[Zer08] Ernst Zermelo, *Untersuchungen über die grundlagen der mengenlehre. i*, Mathematische Annalen **65** (1908), no. 2, 261–281.

Appendix A. Game semantics of Martin-Löf type theory

In this section, we review Yamada's game semantics of MLTT [Yam23]. To this end, we first recall McCusker's games and strategies [McC98, AM99b] (with the slight modifications made in *loc. cit.*) that interpret simple type theories [AM99b] in Appendix A.1 as *loc. cit.* is based on that variant of games and strategies. Yamada selects that variant since it has the *linear decomposition* of function types in the sense of linear logic [Gir87] and interprets *computational effects* [AM99b]; he hopes that his game semantics of MLTT will eventually solve the problem of combining MLTT with linear logic and/or computational effects [Yam23, §2]. We then review the basic definitions and the results of the game semantics of MLTT in Appendix A.2. Our exposition is *minimal*; see the original articles [AM99b, McC98, Yam23] for more explanations and examples.

Appendix A.1. Games and strategies

A *game* is a certain set of finite sequences or *positions*, which represent possible developments in the game. Each element of the position is called a *move* in the game, and a *play* in the game proceeds as the participants, *Player* and *Opponent*, alternately make moves along a growth of positions. By convention, Opponent always makes the first move.

These notions are centred around the structure of an *arena*, and positions are required to satisfy an axiom, called *legality*. Let us first recall these auxiliary concepts.

Definition Appendix A.1.1 (moves [Yam23])**.** Fix arbitrary pairwise distinct symbols O, P, Q and A, and call them *labels*. A *move* is a triple of the form

$$m^{xy} := (m, x, y)$$

such that $x \in \{\mathrm{O}, \mathrm{P}\}$ and $y \in \{\mathrm{Q}, \mathrm{A}\}$.

Convention. We usually abbreviate a move m^{xy} as m and instead define

$$\lambda(m) := xy, \qquad \lambda^{\mathrm{OP}}(m) := x, \qquad \lambda^{\mathrm{QA}}(m) := y,$$

and call a move m an *O-move* if $\lambda^{\mathrm{OP}}(m) = \mathrm{O}$, a *P-move* if $\lambda^{\mathrm{OP}}(m) = \mathrm{P}$, a *question* if $\lambda^{\mathrm{QA}}(m) = \mathrm{Q}$, and an *answer* if $\lambda^{\mathrm{QA}}(m) = \mathrm{A}$.

Definition Appendix A.1.2 (arenas [HO00, McC98])**.** An *arena* is a pair $A = (M_A, \vdash_A)$ of

- A set M_A of moves,
- A subset $\vdash_A$, called the *enabling relation*, of the cartesian product $(\{\star\} \cup M_A) \times M_A$, where $\star$ is an arbitrarily fixed element such that $\star \notin M_A$, that satisfies
  - (E1) If $\star \vdash_A m$, then $\lambda(m) = \mathrm{OQ}$;
  - (E2) If $m \vdash_A n$ and $\lambda^{\mathrm{QA}}(n) = \mathrm{A}$, then $\lambda^{\mathrm{QA}}(m) = \mathrm{Q}$;
  - (E3) If $m \vdash_A n$ and $m \neq \star$, then $\lambda^{\mathrm{OP}}(m) \neq \lambda^{\mathrm{OP}}(n)$.

We call a move m in an arena A *initial* in A if $\star \vdash_A m$, and define the set

$$M_A^{\mathrm{Init}} := \{\, m \in M_A \mid \star \vdash_A m \,\}$$

of all initial moves in A. An arena A is said to be *well-founded* if so is the relation $\vdash_A$, *i.e.*, there is no infinite sequence $(m_i)_{i \in \mathbb{N}}$ of moves $m_i \in M_A$ with $\star \vdash_A m_0$ and $m_i \vdash_A m_{i+1}$ for all $i \in \mathbb{N}$.

The idea is that an arena A specifies moves in a game, each of which is Opponent's/Player's question/answer, and the relation $m \vdash_A n$ defining that the move n can be made for the move m during a play in the game (Appendix A.1.3), where $\star \vdash_A m$ means that Opponent can *initiate* a play by the move 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 Opponent'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*.

Strictly speaking, a positions is a finite sequence together with a *pointer*. The idea is that each non-initial occurrence of a move in a position is made for a specific previous occurrence; a pointer defines these pairs of occurrences. We call a finite sequence together with a pointer a *j-sequence*:

Definition Appendix A.1.3 (j-sequences [Coq95, HO00, Yam23])**.** A *justified (j-)sequence* is a pair $\boldsymbol{s} = (\boldsymbol{s}, \mathcal{J}_{\boldsymbol{s}})$ of a finite sequence $\boldsymbol{s}$ of moves and a function

$$\mathcal{J}_{\boldsymbol{s}} : \overline{|\boldsymbol{s}|} \to \{0\} \cup \overline{|\boldsymbol{s}| - 1},$$

called the *pointer* of $\boldsymbol{s}$, such that $0 \leqslant \mathcal{J}_{\boldsymbol{s}}(i) < i$ for all $i \in \overline{|\boldsymbol{s}|}$.

- An *occurrence* in a finite sequence $\boldsymbol{t}$ refers to a pair $(\boldsymbol{t}(i), i)$ such that $i \in \overline{|\boldsymbol{t}|}$;
- An occurrence $(\boldsymbol{s}(i), i)$ is said to be *initial* in a j-sequence $\boldsymbol{s}$ if $\mathcal{J}_{\boldsymbol{s}}(i) = 0$;
- The occurrence $(\boldsymbol{s}(\mathcal{J}_{\boldsymbol{s}}(i)), \mathcal{J}_{\boldsymbol{s}}(i))$ such that $\mathcal{J}_{\boldsymbol{s}}(i) > 0$ is said to be the *justifier* of a non-initial one $(\boldsymbol{s}(i), i)$ in $\boldsymbol{s}$, and equivalently $(\boldsymbol{s}(i), i)$ is said to be *justified* by $(\boldsymbol{s}(\mathcal{J}_{\boldsymbol{s}}(i)), \mathcal{J}_{\boldsymbol{s}}(i))$ in $\boldsymbol{s}$;
- A j-sequence $\boldsymbol{s}$ is said to be *in an arena* A if the elements of its underlying sequence $\boldsymbol{s}$ are all moves in A, and its pointer $\mathcal{J}_{\boldsymbol{s}}$ respects the enabling relation $\vdash_A$, *i.e.*,

  $$\forall i \in \overline{|\boldsymbol{s}|}.\, \big(\mathcal{J}_{\boldsymbol{s}}(i) = 0 \Rightarrow \star \vdash_A \boldsymbol{s}(i)\big) \wedge \big(\mathcal{J}_{\boldsymbol{s}}(i) > 0 \Rightarrow \boldsymbol{s}(\mathcal{J}_{\boldsymbol{s}}(i)) \vdash_A \boldsymbol{s}(i)\big),$$

  where we write $\mathscr{J}_G$ for the set of all j-sequences in A.

Convention. Henceforth, we are casual about the distinction between moves and their occurrences in sequences; also, we often keep the pointer $\mathcal{J}_{\boldsymbol{s}}$ of a j-sequence $\boldsymbol{s}$ implicit (as it is mostly clear), and abbreviate an occurrence $(\boldsymbol{s}(i), i)$ in $\boldsymbol{s}$ as $\boldsymbol{s}(i)$. In addition:

- We write $\mathcal{J}_{\boldsymbol{s}}(\boldsymbol{s}(i)) = \boldsymbol{s}(j)$ if $\mathcal{J}_{\boldsymbol{s}}(i) = j > 0$;
- We extend the notation $\boldsymbol{s}(i)$ $(i \in |\boldsymbol{s}|)$ by $\boldsymbol{s}(0) := \star$ (so that the pair $(\star, 0)$ is seen as *justifying* initial occurrences in $\boldsymbol{s}$), and write $\mathcal{J}_{\boldsymbol{s}}(\boldsymbol{s}(i)) = \boldsymbol{s}(j)$ even if $\mathcal{J}_{\boldsymbol{s}}(i) = j = 0$, *i.e.*, $\boldsymbol{s}(j) = \star$.

This convention is convenient in practice. For instance, see the following:

Definition Appendix A.1.4 (j-subsequences [Yam23])**.** A *justified (j-)subsequence* of a j-sequence $\boldsymbol{s}$ is a j-sequence $\boldsymbol{t}$, written $\boldsymbol{t} \sqsubseteq \boldsymbol{s}$, whose underlying sequence is a subsequence of $\boldsymbol{s}$, and $\mathcal{J}_{\boldsymbol{t}}(n) = m$ if and only if there are occurrences $m_1, m_2, \ldots, m_k$ in $\boldsymbol{s}$ deleted in $\boldsymbol{t}$ such that

$$\mathcal{J}_{\boldsymbol{s}}(n) = m_1, \qquad \mathcal{J}_{\boldsymbol{s}}(m_1) = m_2, \qquad \ldots, \qquad \mathcal{J}_{\boldsymbol{s}}(m_{k-1}) = m_k, \qquad \mathcal{J}_{\boldsymbol{s}}(m_k) = m,$$

where note that m can be $\star$ (which is not a move) by the above convention.

We note that a j-subsequence of a j-sequence $\boldsymbol{s}$ is completely specified by a subsequence of the underlying sequence of $\boldsymbol{s}$. We are now ready to proceed to the last auxiliary concept for legality:

Definition Appendix A.1.5 (views [Coq95, HO00, McC98])**.** The *P-view* $\lceil \boldsymbol{s} \rceil$ and the *O-view* $\lfloor \boldsymbol{s} \rfloor$ of a j-sequence $\boldsymbol{s}$ are the j-subsequences of $\boldsymbol{s}$ defined respectively by

1. $\lceil \boldsymbol{\epsilon} \rceil := \boldsymbol{\epsilon}$;
2. $\lceil \boldsymbol{s}m \rceil := \lceil \boldsymbol{s} \rceil.m$ if m is a P-move, where the justifier of m is kept unchanged;

3. $\lceil \boldsymbol{s}m \rceil := m$ if m is initial;
4. $\lceil \boldsymbol{s}m\boldsymbol{t}n \rceil := \lceil \boldsymbol{s} \rceil.mn$ if n is an O-move such that m justifies n;
5. $\lfloor \boldsymbol{\epsilon} \rfloor := \boldsymbol{\epsilon}$;
6. $\lfloor \boldsymbol{s}m \rfloor := \lfloor \boldsymbol{s} \rfloor.m$ if m is an O-move, where the justifier of m is kept unchanged;
7. $\lfloor \boldsymbol{s}m\boldsymbol{t}n \rfloor := \lfloor \boldsymbol{s} \rfloor.mn$ if n is a P-move such that m justifies n.

Remark. Strictly speaking, the P-view $\lceil \boldsymbol{s} \rceil$ or the O-view $\lfloor \boldsymbol{s} \rfloor$ may not be a j-sequence because the justifier of m may be lost in the clause (2) or (6). Yet, this problem is insignificant as we later focus on *visible* j-sequences (Appendix A.1.6), for which this problem does not occur [McC98, pp. 19–20].

The idea on views is as follows. Given a nonempty j-sequence $\boldsymbol{s}m$ such that m is a P- (respectively, O-)move, the P-view $\lceil \boldsymbol{s} \rceil$ (respectively, O-view $\lfloor \boldsymbol{s} \rfloor$) is the currently 'relevant part' of the previous occurrences in $\boldsymbol{s}$ for Player (respectively, Opponent). In other words, Player (respectively, Opponent) is concerned only with the last occurrence of an O- (respectively, P-)move, its justifier and that justifier's P- (respectively, O-)view, which then recursively proceeds.

We are now ready to recall *legal positions* and then *games*:

Definition Appendix A.1.6 (legal positions [HO00, McC98, Yam23])**.** A *legal position* is a j-sequence $\boldsymbol{s}$ such that

- (Alternation) If $\boldsymbol{s} = \boldsymbol{s_1}mn\boldsymbol{s_2}$, then $\lambda^{\mathrm{OP}}(m) \neq \lambda^{\mathrm{OP}}(n)$;
- (Visibility) If $\boldsymbol{s} = \boldsymbol{t}m\boldsymbol{u}$ with m non-initial, then $\mathcal{J}_{\boldsymbol{s}}(m)$ occurs in the P-view $\lceil \boldsymbol{t} \rceil$ if m is a P-move, and in the O-view $\lfloor \boldsymbol{t} \rfloor$ otherwise.

A legal position is said to be *in an arena* A if it is a j-sequence in A (Appendix A.1.3). Let us write $\mathscr{L}_A$ for the set of all legal positions in A.

As already noted, legal positions *in an arena* are to specify the basic rules of a game in the sense that all positions in the game are legal so that

- During a play, Opponent makes the first move by a question (by E1),[9] and then Player and Opponent alternately make moves (by alternation), where each non-initial move is made for a specific one, *viz.*, its justifier;[10]
- The justifiers are in the 'relevant parts' or *views* (by visibility).

Definition Appendix A.1.7 (games [McC98, Yam23])**.** A *game* is a set G of legal positions that satisfies the following:

1. The set G is nonempty and *prefix-closed* (*i.e.*, $\boldsymbol{s}m \in G$ implies $\boldsymbol{s} \in G$);
2. The pair $\mathrm{Arn}(G) := (M_G, \vdash_G)$ is an arena, where
   (a) $M_G := \{\, \boldsymbol{s}(i) \mid \boldsymbol{s} \in G, i \in \overline{|\boldsymbol{s}|} \,\}$,
   (b) $\vdash_G := \bigcup_{\boldsymbol{s} \in G} (\{\, (\star, \boldsymbol{s}(j)) \mid \mathcal{J}_{\boldsymbol{s}}(j) = 0 \,\} \cup \{\, (\boldsymbol{s}(i), \boldsymbol{s}(j)) \mid \mathcal{J}_{\boldsymbol{s}}(j) = i > 0 \,\})$.

[9] Since the initial element $\boldsymbol{s}(1)$ of a legal position $\boldsymbol{s}$ *in an arena* A is subject to the equation $\mathcal{J}_{\boldsymbol{s}}(1) = 0$, we have $\star \vdash_A \boldsymbol{s}(1)$. Hence, the axiom E1 on A implies $\lambda(\boldsymbol{s}(1)) = \mathrm{OQ}$.

[10] Again, since we focus on a legal position $\boldsymbol{s}$ *in an arena* A, the justifier of each P-move occurring in $\boldsymbol{s}$ is an O-move, and *vice versa*, by the axiom E3 on A. In addition, the justifier of each answer occurring in $\boldsymbol{s}$ is a question by the axiom E2 on A.

We call elements of G *positions* in G. A *play* in G is a (finite or infinite) sequence of positions in G of the form

$$(\boldsymbol{\epsilon}, m_1, m_1m_2, m_1m_2m_3, \dots).$$

A game G is said to be *well-founded* if so is the arena $\mathrm{Arn}(G)$, and *well-opened* if each of its positions contains at most one initial move. A *subgame* of G is a game H such that $H \subseteq G$, where we write $\mathrm{Sub}(G)$ for the set of all subgames of G.

Example Appendix A.1.8. The simplest game is the *terminal game*

$$T := \{\boldsymbol{\epsilon}\},$$

which only has the empty position $\boldsymbol{\epsilon}$. The *flat game* on a given set S is the game

$$\mathrm{flat}(S) := \mathrm{Pref}(\{\, q^{\mathrm{OQ}}.m^{\mathrm{PA}} \mid m \in S \,\}),$$

where q is an arbitrarily fixed element such that $q \notin S$, and q^{OQ} justifies m^{PA}.

Consider, *e.g.*, the *empty game* 0 and the *natural number game* N defined respectively by

$$0 := \mathrm{flat}(\emptyset), \qquad\qquad N := \mathrm{flat}(\mathbb{N}).$$

The empty game interprets Zero-type, and the natural number game N-type (Appendix A.2).

Next, *strategies* on a game G are Player's algorithms to play in G:

Definition Appendix A.1.9 (strategies [McC98])**.** A *strategy* on a game G is a subset $\sigma \subseteq G^{\mathrm{Even}}$, written $\sigma : G$, that is nonempty, *even-prefix-closed* (*i.e.*, $\boldsymbol{s}mn \in \sigma$ implies $\boldsymbol{s} \in \sigma$) and *deterministic* (*i.e.*, $\boldsymbol{s}mn, \boldsymbol{s}mn' \in \sigma$ implies $\boldsymbol{s}mn = \boldsymbol{s}mn'$).

We write $\mathrm{St}(G)$ for the set of all strategies on a game G. The idea is that a strategy $\sigma : G$ describes for Player how to play in G by the computation

$$\boldsymbol{s}m \in G^{\mathrm{Odd}} \mapsto \boldsymbol{s}mn \in \sigma,$$

if any, which is *unique* by the determinacy of σ, and in general *partial* because there can be no output $\boldsymbol{s}mn \in \sigma$ for some input $\boldsymbol{s}m \in G^{\mathrm{Odd}}$.

Example Appendix A.1.10. The terminal game T only has the trivial strategy

$$\top := \{\boldsymbol{\epsilon}\},$$

and the flat game $\mathrm{flat}(S)$ on a set S has those

$$\bot := \{\boldsymbol{\epsilon}\}, \qquad\qquad \underline{m} := \{\boldsymbol{\epsilon}, qm\} \quad (m \in S).$$

Nevertheless, strategies are too unrestricted to correspond to proofs, and this problem motivates *winning* and *well-bracketing*: Winning strategies correspond to proofs in classical logic, and winning, well-bracketed ones to proofs in intuitionistic logic. Because the logic of MLTT is intuitionistic, we may obtain a tight correspondence between MLTT and game semantics by focusing on winning, well-bracketed strategies.

Definition Appendix A.1.11 (constraints on strategies [HO00, CH10, Coq95, Lai97])**.** A strategy $\sigma : G$ is said to be

- *Total* if it always responds: $\forall \boldsymbol{s} \in \sigma, \boldsymbol{s}m \in G.\, \exists \boldsymbol{s}mn \in \sigma$;
- *Innocent* if it only depends on P-views: $\forall \boldsymbol{s}mn \in \sigma, \boldsymbol{t} \in \sigma, \boldsymbol{t}l \in G.\, \lceil \boldsymbol{s}m \rceil = \lceil \boldsymbol{t}l \rceil \Rightarrow \exists \boldsymbol{t}lr \in \sigma.\, \lceil \boldsymbol{s}mn \rceil = \lceil \boldsymbol{t}lr \rceil$;
- *Noetherian* if there is no strictly increasing (with respect to the prefix relation $\preceq$) infinite sequence of elements in the set $\lceil \sigma \rceil := \{\, \lceil \boldsymbol{s} \rceil \mid \boldsymbol{s} \in \sigma \,\}$ of all P-views in σ;
- *Winning* if it is total, innocent and noetherian;
- *Well-bracketed* if its question-answering in P-views is in the last-question-first-answered fashion: If $\boldsymbol{sqt}a \in \sigma$, where $\lambda^{\mathrm{QA}}(q) = \mathrm{Q}$, $\lambda^{\mathrm{QA}}(a) = \mathrm{A}$ and $\mathcal{J}_{\boldsymbol{sqt}a}(a) = q$, then each question in $\boldsymbol{t'}$, where $\lceil \boldsymbol{sqt} \rceil$ is of the form $\lceil \boldsymbol{sqt} \rceil = \lceil \boldsymbol{sq} \rceil . \boldsymbol{t'}$ by visibility, justifies an answer in $\boldsymbol{t'}$.

Example Appendix A.1.12. The strategies $\top : T$ and $\underline{n} : N$ for all $n \in \mathbb{N}$ are winning and well-bracketed, while those $\bot : 0$ and $\bot : N$ are not even total.

We regard winning strategies as proofs in classical logic as follows. First, proofs must not get 'stuck,' so strategies playing as proofs are *total*. Next, imposing *innocence* on strategies corresponds to excluding *stateful* terms [AM97, AHM98, AM99a]. Since logic is concerned with *truths*, independently of 'passage of time,' proofs should not depended on 'states of arguments.' Thus, we impose innocence on strategies for proofs. Finally, we need *noetherianity* to handle infinite plays: If a play by an innocent, noetherian strategy keeps growing infinitely, then it cannot be Player's 'intention,' so the play must be her 'win.' Technically, noetherianity is crucial for the closure of winning strategies under composition (Appendix A.1.15).

In addition, *well-bracketing* bans classical reasoning or *control operators* [Lai97]. Thus, we see winning, well-bracketed strategies as proofs in intuitionistic logic.

Let us next recall standard constructions on games and strategies.

Convention. For readability, we omit *tags* for disjoint union $\uplus$. For instance, we write $x \in A \uplus B$ if $x \in A$ or $x \in B$; also, given relations $R_A \subseteq A \times A$ and $R_B \subseteq B \times B$, we write $R_A \uplus R_B$ for the relation on $A \uplus B$ such that $(x, y) \in R_A \uplus R_B :\Leftrightarrow (x, y) \in R_A \vee (x, y) \in R_B$.

Definition Appendix A.1.13 (constructions on arenas [HO00, McC98])**.** Given arenas A and B, we define the arenas

- $A \uplus B := (M_A \uplus M_B, \vdash_A \uplus \vdash_B)$;
- If $B \neq T$, then $A \multimap B := (\{\, a^{(x^\perp)y} \mid a^{xy} \in M_A \,\} \uplus M_B, \vdash_{A \multimap B})$, where

$$\mathrm{O}^\perp := \mathrm{P}, \qquad \mathrm{P}^\perp := \mathrm{O}, \qquad \star \vdash_{A \multimap B} m :\Leftrightarrow \star \vdash_B m,$$

$$m \vdash_{A \multimap B} n :\Leftrightarrow m \vdash_A n \vee m \vdash_B n \vee (\star \vdash_B m \wedge \star \vdash_A n);$$

- $A \multimap T := T$.[11]

[11] By distinguishing the case $A \multimap T$ from the one $A \multimap B$ with $B \neq T$, we save $A \multimap T$ from unused structures. In other words, it keeps games *economical*, *i.e.*, free from unused structures [Yam23, p. 9].

Definition Appendix A.1.14 (constructions on games [HO00, McC98])**.** Given games G and H, we define the games

- The *tensor* of G and H:

$$G \otimes H := \{\, \boldsymbol{s} \in \mathscr{L}_{\mathrm{Arn}(G) \uplus \mathrm{Arn}(H)} \mid \forall X \in \{G, H\}.\, \boldsymbol{s}{\restriction_X} \in X \,\},$$

  where the j-subsequence $\boldsymbol{s}{\restriction_X} \sqsubseteq \boldsymbol{s}$ consists of the occurrences of moves in X;

- The *exponential* of G:

$$!G := \{\, \boldsymbol{s} \in \mathscr{L}_{\mathrm{Arn}(G)} \mid \forall i \in |\boldsymbol{s}|.\, \mathcal{J}_{\boldsymbol{s}}(i) = 0 \Rightarrow \boldsymbol{s}{\restriction_{\{(\boldsymbol{s}(i), i)\}}} \in G \,\},$$

  where the j-subsequence $\boldsymbol{s}{\restriction_{\{(\boldsymbol{s}(i), i)\}}} \sqsubseteq \boldsymbol{s}$[12] consists of the occurrences in $\boldsymbol{s}$ *hereditarily justified*[13] by the initial occurrence $(\boldsymbol{s}(i), i)$ in $\boldsymbol{s}$;

- The *product* of G and H:

$$G \,\&\, H := \{\, \boldsymbol{s} \in \mathscr{L}_{\mathrm{Arn}(G) \uplus \mathrm{Arn}(H)} \mid (\boldsymbol{s}{\restriction_G} \in G \wedge \boldsymbol{s}{\restriction_H} = \boldsymbol{\epsilon}) \vee (\boldsymbol{s}{\restriction_G} = \boldsymbol{\epsilon} \wedge \boldsymbol{s}{\restriction_H} \in H) \,\};$$

- The *linear implication* from G to H:

$$G \multimap H := \{\, \boldsymbol{s} \in \mathscr{L}_{\mathrm{Arn}(G) \multimap \mathrm{Arn}(H)} \mid \boldsymbol{s}{\restriction_{G^\perp}} \in G, \boldsymbol{s}{\restriction_H} \in H \,\},$$

  also written H^G, where the j-sequence $\boldsymbol{s}{\restriction_{G^\perp}}$ is obtained from the one $\boldsymbol{s}{\restriction_G}$ by modifying all the moves $m^{(x^\perp)y}$ occurring in $\boldsymbol{s}{\restriction_G}$ into those m^{xy};

- The *implication* from G to H:

$$G \Rightarrow H := !G \multimap H.$$

Notation. Notationally, exponential ! precedes other constructions on games, while tensor $\otimes$ and product & precede linear implication $\multimap$ and implication $\Rightarrow$.

Definition Appendix A.1.15 (constructions on strategies [HO00, McC98])**.** Given strategies $\phi : G \multimap H$, $\sigma : K \multimap L$, $\tau : G \multimap K$, $\psi : H \multimap K$ and $\theta : !G \multimap H$, we define

- The *copy-cat* on G:

$$\mathrm{cp}_G := \{\, \boldsymbol{s} \in (G_{[0]} \multimap G_{[1]})^{\mathrm{Even}} \mid \forall \boldsymbol{t} \preceq \boldsymbol{s}.\, \mathrm{Even}(\boldsymbol{t}) \Rightarrow \boldsymbol{t}{\restriction_{G^\perp_{[0]}}} = \boldsymbol{t}{\restriction_{G_{[1]}}}, \mathrm{Init}_{G_{[0]}, G_{[1]}}(\boldsymbol{s}) \,\},$$

  where the predicate $\mathrm{Init}_{G_{[0]}, G_{[1]}}(\boldsymbol{s})$ means that every initial occurrence in the domain $G_{[0]}$ points to the last initial occurrence in the codomain $G_{[1]}$;

- The *dereliction* on G:

$$\mathrm{der}_G := \{\, \boldsymbol{s} \in (!G \multimap G)^{\mathrm{Even}} \mid \forall \boldsymbol{t} \preceq \boldsymbol{s}.\, \mathrm{Even}(\boldsymbol{t}) \Rightarrow \boldsymbol{t}{\restriction_{!G^\perp}} = \boldsymbol{t}{\restriction_G}, \mathrm{Init}_{!G, G}(\boldsymbol{s}) \,\};$$

[12] We abuse the notation $\restriction$ for the operations $\boldsymbol{s}{\restriction_X}$ and $\boldsymbol{s}{\restriction_{\{(\boldsymbol{s}(i), i)\}}}$, but it is not a problem in practice.

[13] An occurrence n in a j-sequence $\boldsymbol{s}$ is *hereditarily justified* by another occurrence m in $\boldsymbol{s}$ if $\mathcal{J}^i_{\boldsymbol{s}}(n) = m$ for some positive integer $i \in \mathbb{N}_+$ [McC98, p. 22].

- The *tensor* of ϕ and σ:

$$\phi \otimes \sigma := \{\, \boldsymbol{s} \in (G \otimes K) \multimap (H \otimes L) \mid \boldsymbol{s}\restriction_{G,H} \in \phi, \boldsymbol{s}\restriction_{K,L} \in \sigma \,\},$$

  where the j-subsequence $\boldsymbol{s}\restriction_{G,H} \sqsubseteq \boldsymbol{s}$ (respectively, $\boldsymbol{s}\restriction_{K,L} \sqsubseteq \boldsymbol{s}$) consists of the occurrences of moves in G or H (respectively, K or L);

- The *pairing* of ϕ and τ:

$$\langle \phi, \tau \rangle := \{\, \boldsymbol{s} \in G \multimap (H \,\&\, K) \mid (\boldsymbol{s}\restriction_{G,H} \in \phi \wedge \boldsymbol{s}\restriction_{K} = \boldsymbol{\epsilon}) \vee (\boldsymbol{s}\restriction_{G,K} \in \tau \wedge \boldsymbol{s}\restriction_{H} = \boldsymbol{\epsilon}) \,\};$$

- The *composition* of ϕ and ψ:

$$\phi; \psi := \{\, \boldsymbol{s}\restriction_{G,K} \mid \boldsymbol{s} \in \phi \parallel \psi \,\} \quad \text{(also written } \psi \circ \phi\text{)},$$

  where

$$\phi \parallel \psi := \{\, \boldsymbol{s} \in \mathscr{J}_{\mathrm{Arn}(((G \multimap H_{[0]}) \multimap H_{[1]}) \multimap K)} \mid \boldsymbol{s}\restriction_{G,H_{[0]}} \in \phi, \boldsymbol{s}\restriction_{H_{[1]},K} \in \psi, \boldsymbol{s}\restriction_{H_{[0]}^{\perp},H_{[1]}^{\perp}} \in \mathrm{cp}_H \,[14] \,\},$$

  and the j-sequence $\boldsymbol{s}\restriction_{H_{[0]}^{\perp},H_{[1]}^{\perp}}$ is obtained from the one $\boldsymbol{s}\restriction_{H_{[0]},H_{[1]}}$ by the application of the operation $(\_)^{\perp} : m^{xy} \mapsto m^{x^{\perp}y}$ (Appendix A.1.13) to all moves m^{xy};

- The *promotion* of θ:

$$\theta^{\dagger} := \{\, \boldsymbol{s} \in (!G \multimap !H)^{\mathrm{Even}} \mid \forall i \in |\boldsymbol{s}|.\, \mathcal{J}_{\boldsymbol{s}}(i) = 0 \Rightarrow \boldsymbol{s}\restriction_{\{(\boldsymbol{s}(i),i)\}} \in \theta \,\}.$$

Example Appendix A.1.16. The promotion $\mathrm{succ}^{\dagger} : !N \multimap !N$ of the strategy

$$\mathrm{succ} := \{\, q_{[1]}.q_{[0]}.n_{[0]}.n + 1_{[1]} \mid n \in \mathbb{N} \,\} : N_{[0]} \Rightarrow N_{[1]}$$

computes as sketched in the introduction (§2.1).

Let us summarise the present section by:

Definition Appendix A.1.17 (categories of games and strategies [McC98])**.** The category $\mathbb{G}_!$ consists of

- Well-opened games as objects;
- Strategies on the implication $G \Rightarrow H$ as morphisms $G \to H$;
- The composition $\psi \bullet \phi := \psi \circ \phi^{\dagger} : G \Rightarrow K$ of strategies as the composition of morphisms $\phi : G \to H$ and $\psi : H \to K$;
- The dereliction der_G as the identity on each object G.

The subcategory $\mathbb{LG}_!$ (respectively, $\mathbb{WG}_!$) of $\mathbb{G}_!$ consists of well-founded, well-opened games and winning (respectively, winning, well-bracketed) strategies.

[14] This condition $\boldsymbol{s}\restriction_{H_{[0]}^{\perp},H_{[1]}^{\perp}} \in \mathrm{cp}_H$ is to guarantee that the play in $H_{[0]}$ is the same as the one in $H_{[1]}$.

We focus on *well-opened* (respectively, *well-opened*, *well-founded*) games in the categories since otherwise the identities would not be well-defined [McC98, pp. 42–43]. Although one can permit ill-opened games by a standard method [McC98, §3.6], Yamada [Yam23] does not use it as it makes his game semantics more complex, and it is not strictly necessary for the interpretation of MLTT.

We use the subscript $(\_)_!$ to distinguish these categories from the *linear* ones [McC98, §3.3], in which morphisms $G \to H$ are strategies on the linear implication $G \multimap H$. We are not bothered about the distinction between strategies on a game G and those on the game $T \multimap G$ or $T \Rightarrow G$.

Appendix A.2. Game semantics of Martin-Löf type theory

Yamada [Yam23] establishes game semantics of MLTT. His idea is to generalise games into *predicate (p-)games*, which corresponds to the generalisation of simple types to dependent ones:

Definition Appendix A.2.1 (predicate games [Yam23])**.** A *predicate (p-)game* is a pair

$$\Gamma = (|\Gamma|, \|\Gamma\|)$$

of a game $|\Gamma|$ and a family $\|\Gamma\| = \{\Gamma(\gamma)\}_{\gamma:|\Gamma|}$ of subgames $\Gamma(\gamma) \subseteq |\Gamma|$, and said to be *well-founded* (respectively, *well-opened*) if so is the game $|\Gamma|$.

Example Appendix A.2.2. Given a game G, we have the p-game

$$\mathscr{P}(G) := (G, \kappa_G),$$

where κ_G is the constant family at G. Clearly, the game G and the p-game $\mathscr{P}(G)$ are essentially the same. We abbreviate the p-games $\mathscr{P}(T)$, $\mathscr{P}(0)$ and $\mathscr{P}(N)$ as T, 0 and N, and call them the *terminal p-game*, the *empty p-game* and the *natural number p-game*, respectively.

Before recalling strategies on p-games, we need a few preliminary concepts:

Definition Appendix A.2.3 (liveness ordering [Chr00])**.** The *liveness ordering* is a partial order $\preccurlyeq$ between games [Chr00, Definition 8 and Theorem 9], which defines $G \preccurlyeq H$ to mean that Opponent (respectively, Player) is less (respectively, more) restricted in G than in H, *i.e.*, they satisfy

1. If $\boldsymbol{s} \in (G \cap H)^{\mathrm{Even}}$ and $\boldsymbol{s}m \in H^{\mathrm{Odd}}$, then $\boldsymbol{s}m \in G^{\mathrm{Odd}}$;
2. If $\boldsymbol{t}l \in (G \cap H)^{\mathrm{Odd}}$ and $\boldsymbol{t}lr \in G^{\mathrm{Even}}$, then $\boldsymbol{t}lr \in H^{\mathrm{Even}}$.

Definition Appendix A.2.4 (closures of strategies [Yam23])**.** The *closure* of a strategy $\sigma : G$ with respect to another game H is the subgame defined inductively by

$$\overline{\sigma}_H := \{\boldsymbol{\epsilon}\} \cup \{\, \boldsymbol{s}m \in H^{\mathrm{Odd}} \mid \boldsymbol{s} \in \overline{\sigma}_H \,\} \cup \{\, \boldsymbol{t}lr \in \sigma \mid \boldsymbol{t}l \in \overline{\sigma}_H \,\} \subseteq \sigma \cup H.$$

We see by induction that the equation

$$\overline{\sigma}_G = \sigma \cup \{\, \boldsymbol{s}m \in G \mid \boldsymbol{s} \in \sigma \,\}$$

holds for all $\sigma : G$. Moreover, we have:

Proposition Appendix A.2.5 (liveness characterisation [Yam23])**.** *Let $\sigma : G$ and $H \in \mathrm{Sub}(G)$.*

1. *$\overline{\sigma}_H^{\mathrm{Even}} : H$ if and only if $\overline{\sigma}_G \preccurlyeq H$;*
2. *If $\overline{\sigma}_G \preccurlyeq H$, then $\overline{\sigma}_H^{\mathrm{Even}} = \sigma \cap H$.*

This proposition enables us to define strategies on p-games in a handy way:

Definition Appendix A.2.6 (strategies on p-games [Yam23])**.** A *strategy* on a p-game Γ, written $\gamma : \Gamma$, is a strategy $\gamma : |\Gamma|$ such that $\overline{\gamma}_{|\Gamma|} \preccurlyeq \Gamma(\gamma)$, where

$$\mathrm{St}(\Gamma) := \{\, \gamma \in \mathrm{St}(|\Gamma|) \mid \gamma : \Gamma \,\}, \qquad \overline{\gamma}_{\Gamma} := \overline{\gamma}_{\Gamma(\gamma)},$$

and said to be *total* (respectively, *innocent*, *noetherian*, *well-bracketed*) if so is the one $\gamma{\cap}\Gamma(\gamma) : \Gamma(\gamma)$.

A *position* in a p-game Γ is a one in a game $\overline{\gamma}_{\Gamma}$ for some strategy $\gamma : \Gamma$. A play in Γ then proceeds as follows: Before a play, Player fixes a strategy $\gamma : \Gamma$, and then an ordinary play on the game $\Gamma(\gamma)$ follows, where Player must play by the strategy γ yet restricted to $\Gamma(\gamma)$, *i.e.*, $\gamma{\cap}\Gamma(\gamma) = \overline{\gamma}^{\mathrm{Even}}_{\Gamma(\gamma)} : \Gamma(\gamma)$. This *predetermination* of a strategy is in accordance with conventional game semantics because in the literature game semantics has always focused on plays by a *fixed* strategy for an interpretation. In addition, the predetermination of a strategy does not lose generality since each position $\boldsymbol{s}$ in a game G is the result of a play by some strategy $\sigma : G$, *viz.*, $\sigma := \mathrm{Pref}(\{\boldsymbol{s}\})^{\mathrm{Even}}$.

By the generalisation of a game $|\Gamma|$ to a p-game $\Gamma = (|\Gamma|, \|\Gamma\|)$, we can *only* select a strategy $\gamma : |\Gamma|$ that satisfies the condition $\overline{\gamma}_{|\Gamma|} \preccurlyeq \Gamma(\gamma)$, and this choice $\gamma : \Gamma$ *fixes* the ambient game $\Gamma(\gamma)$; *i.e.*, the game-semantic counterpart of the generalisation of simple types to dependent ones is the family $\|\Gamma\|$, which brings these *strategy filtering* and *game fixing* abilities to the game $|\Gamma|$.

We next recall basic constructions on p-games:

Notation. If G is a game, $\boldsymbol{s} \in !G$ and $i \in \mathbb{N}$, then (again by abuse of the notation $\restriction$) let $\boldsymbol{s}\restriction_i$ be the j-subsequence of $\boldsymbol{s}$ that consists of the occurrences hereditarily justified by the $(i+1)$st initial occurrence in $\boldsymbol{s}$; *e.g.*, if $\boldsymbol{s} = q2q1q0 \in !N$, then $\boldsymbol{s}\restriction_0 = q2$, $\boldsymbol{s}\restriction_1 = q1$ and $\boldsymbol{s}\restriction_2 = q0$.

Given a strategy σ on the tensor $G_0 \otimes G_1$ of games G_i $(i = 0, 1)$, we define

$$\sigma\restriction_{G_i} := \begin{cases} \sigma_i & \text{if } \sigma = \sigma_0 \otimes \sigma_1 \text{ for (necessarily unique) } \sigma_0 : G_0 \text{ and } \sigma_1 : G_1; \\ \uparrow & \text{otherwise, where } \uparrow \text{ means being } \textit{undefined}. \end{cases}$$

Similarly, for a strategy τ on the exponential $!G$ of a game G and $j \in \mathbb{N}$, let

$$\tau\restriction_j := \begin{cases} \{\, \boldsymbol{s}\restriction_j \mid \boldsymbol{s} \in \tau \,\} & \text{if } \{\, \boldsymbol{s}\restriction_k \mid \boldsymbol{s} \in \tau \,\} : G \text{ for all } k \in \mathbb{N}; \\ \uparrow & \text{otherwise.} \end{cases}$$

For a p-game Γ, we let the value $\Gamma(\uparrow)$ to be *undefined*, and the operations $\otimes$, $\multimap$, $\&$ and $!$ on undefined games to be *undefined*. Lastly, we extend the relation $\overline{\gamma}_{|\Gamma|} \preccurlyeq \Gamma(\gamma)$ by defining that *it does not hold if the game* $\Gamma(\gamma)$ *is undefined*.

Definition Appendix A.2.7 (product and tensor on p-games [Yam23])**.** The *product* of p-games Γ and Δ is the p-game $\Gamma \,\&\, \Delta$ defined by

$$|\Gamma \,\&\, \Delta| := |\Gamma| \,\&\, |\Delta|, \qquad (\Gamma \,\&\, \Delta)(\langle \gamma, \delta \rangle) := \Gamma(\gamma) \,\&\, \Delta(\delta) \quad (\langle \gamma, \delta \rangle : |\Gamma \,\&\, \Delta|),$$

and their *tensor* is the p-game $\Gamma \otimes \Delta$ defined by

$$|\Gamma \otimes \Delta| := |\Gamma| \otimes |\Delta|, \qquad (\Gamma \otimes \Delta)(\sigma) := \Gamma(\sigma\restriction_{|\Gamma|}) \otimes \Delta(\sigma\restriction_{|\Delta|}) \quad (\sigma : |\Gamma \otimes \Delta|).$$

Definition Appendix A.2.8 (countable tensor [Yam23])**.** The *countable tensor* of a family $(G_i)_{i \in \mathbb{N}}$ of subgames $G_i \subseteq H$ is the subgame

$$\otimes_{i \in \mathbb{N}} G_i := \{\, \boldsymbol{s} \in !H \mid \forall j \in \mathbb{N}.\, \boldsymbol{s}\restriction_j \in G_j \,\} \subseteq !H.$$

Definition Appendix A.2.9 (exponential on p-games [Yam23])**.** The *exponential* of a p-game Γ is the p-game $!\Gamma$ defined by

$$|!\Gamma| := !|\Gamma|, \qquad (!\Gamma)(\sigma) := \otimes_{i\in\mathbb{N}}\Gamma(\sigma\restriction_i) \quad (\sigma : |!\Gamma|).$$

Definition Appendix A.2.10 (implication between p-games [Yam23])**.** The *linear implication* between p-games Γ and Δ is the p-game $\Gamma \multimap \Delta$ (also written Δ^Γ) defined by $|\Delta^\Gamma| := |\Delta|^{|\Gamma|}$ and for all $\phi : |\Delta^\Gamma|$

$$\begin{aligned}(\Delta^\Gamma)(\phi) := \{\boldsymbol{\epsilon}\} \cup \{\, \boldsymbol{s}m \in |\Delta^\Gamma|^{\mathrm{Odd}} \mid \boldsymbol{s} \in (\Delta^\Gamma)(\phi), \exists\gamma : \Gamma.\, \boldsymbol{s}m \in \Delta(\phi\circ\gamma)^{\overline{\gamma}_\Gamma} \,\}\\ \cup \{\, \boldsymbol{t}lr \in |\Delta^\Gamma|^{\mathrm{Even}} \mid \boldsymbol{t}l \in (\Delta^\Gamma)(\phi), \forall\gamma : \Gamma.\, \boldsymbol{t}l \in \Delta(\phi\circ\gamma)^{\overline{\gamma}_\Gamma} \Rightarrow \boldsymbol{t}lr \in \Delta(\phi\circ\gamma)^{\overline{\gamma}_\Gamma} \,\},\end{aligned}$$

and the *implication* between Γ and Δ is the linear implication

$$\Gamma \Rightarrow \Delta := !\Gamma \multimap \Delta.$$

The first clause of the inductive definition of the subgame $(\Delta^\Gamma)(\phi) \subseteq |\Delta^\Gamma|$ is the base case. The second clause specifies one of the two inductive steps: At an even-length position $\boldsymbol{s} \in (\Delta^\Gamma)(\phi)^{\mathrm{Even}}$, Opponent *can* make a move m as in $\Delta(\phi\circ\gamma)^{\overline{\gamma}_\Gamma} \subseteq |\Delta^\Gamma|$ for *any* $\gamma : \Gamma$ *not yet excluded*, *i.e.*, such that $\boldsymbol{s} \in \Delta(\phi\circ\gamma)^{\overline{\gamma}_\Gamma}$. Finally, the third clause stipulates the other inductive step: At an odd-length position $\boldsymbol{t}l \in (\Delta^\Gamma)(\phi)^{\mathrm{Odd}}$, the next move r by ϕ *must* be as in $\Delta(\phi\circ\gamma)^{\overline{\gamma}_\Gamma} \subseteq |\Delta^\Gamma|$ for *some* $\gamma : \Gamma$ *not yet excluded*, *i.e.*, such that $\boldsymbol{t}l \in \Delta(\phi\circ\gamma)^{\overline{\gamma}_\Gamma}$. The idea is that in the subgame $\Delta^\Gamma(\phi) \subseteq |\Delta^\Gamma|$ Opponent can play as in *any* subgame $\Delta(\phi\circ\gamma)^{\overline{\gamma}_\Gamma} \subseteq |\Delta^\Gamma|$ not yet excluded. Because Player or ϕ should be able to see what the input strategy $\gamma : \Gamma$ is *only via plays*, the subgame $\Delta^\Gamma(\phi) \subseteq |\Delta^\Gamma|$ only allows Player to play as in the game $\Delta(\phi\circ\gamma)^{\overline{\gamma}_\Gamma}$ for *all* γ not yet excluded.

Yamada [Yam23] emphasises that strategies $\phi : \Delta^\Gamma$ are the *ordinary* ones (Appendix A.1.9), which just satisfy an additional axiom. In particular, this means that ϕ can see Opponent's strategy $\gamma : \Gamma$ on the domain Γ *only gradually via plays*. In this way, his approach retains the *intensionality* of standard game semantics.

Besides, p-games give rise to categories just like games do (Appendix A.1.17):

Definition Appendix A.2.11 (categories of p-games [Yam23])**.** The category $\mathbb{PG}_!$ consists of

- Well-opened p-games as objects;
- Strategies on the implication $\Gamma \Rightarrow \Delta$ as morphisms $\Gamma \to \Delta$;
- The composition $\psi \bullet \phi := \psi \circ \phi^\dagger : \Gamma \Rightarrow \Theta$ of strategies as the composition of morphisms $\phi : \Gamma \to \Delta$ and $\psi : \Delta \to \Theta$;
- The dereliction $\mathrm{der}_{|\Gamma|} : \Gamma \Rightarrow \Gamma$ as the identity id_Γ on each object Γ.

Its subcategory $\mathbb{LPG}_!$ (respectively, $\mathbb{WPG}_!$) has well-founded, well-opened p-games and winning (respectively, winning, well-bracketed) strategies. Let

$$\mathbb{PG}_!(\Gamma) := \mathbb{PG}_!(T, \Gamma), \qquad \mathbb{LPG}_!(\Gamma) := \mathbb{LPG}_!(T, \Gamma), \qquad \mathbb{WPG}_!(\Gamma) := \mathbb{WPG}_!(T, \Gamma).$$

Again, the subscript $(\_)_!$ is to distinguish these categories from *linear* ones, and we focus on well-opened or well-opened, well-founded games for the identities to be well-defined. Since the logic of MLTT is intuitionistic, Yamada [Yam23] focuses on the category $\mathbb{WPG}_!$ and establishes game semantics of MLTT by showing that $\mathbb{WPG}_!$ induces a CwF (Definition 3.0.1).

In the following, we recall the additional structures on the category $\mathbb{WPG}_!$ that lift it to a CwF. First, types in the CwF $\mathbb{WPG}_!$ are *dependent p-games*:

Definition Appendix A.2.12 (dependent p-games [Yam23])**.** A *linearly dependent p-game* over a p-game Γ 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* (respectively, *well-founded*) if so is the game $|L|$. The *extension* of the family $\|L\|$ is the family $L^\star = \{L^\star(\gamma)\}_{\gamma:\Gamma}$, where

$$L^\star(\gamma) := \begin{cases} L(\gamma) & \text{if } \gamma \in \mathbb{WPG}_!(\Gamma); \\ \mathscr{P}(|L|) & \text{otherwise,} \end{cases}$$

and a *dependent p-game* over Γ is a linearly dependent p-game over the exponential $!\Gamma$.

Notation. Let $\mathscr{D}_\ell(\Gamma)$ (respectively, $\mathscr{D}^{\mathrm{w}}_\ell(\Gamma)$) be the class of all linearly dependent p-games (respectively, well-opened, well-founded ones) over a p-game Γ, $\{\Gamma'\}_\Gamma$ or $\{\Gamma'\}$ the *constant* one at Γ', *i.e.*, $\{\Gamma'\}_\Gamma := (\Gamma', (\gamma : \Gamma) \mapsto \Gamma')$, $\mathscr{D}(\Gamma) := \mathscr{D}_\ell(!\Gamma)$ and $\mathscr{D}^{\mathrm{w}}(\Gamma) := \mathscr{D}^{\mathrm{w}}_\ell(!\Gamma)$. We write $\gamma_0^\dagger$ for an arbitrary element of $\mathbb{WPG}_!(!\Gamma)$, where $\gamma_0 \in \mathbb{WPG}_!(\Gamma)$, since elements of $\mathbb{WPG}_!(!\Gamma)$ are all *innocent* and thus the promotions of elements of $\mathbb{WPG}_!(\Gamma)$. For the case of the CwF $\mathbb{UPG}_!^{(\mathrm{eff})}$, the class $\mathscr{D}(\Gamma)$ for each $\Gamma \in \mathbb{UPG}_!^{(\mathrm{eff})}$ is a *set* because dependent p-games in $\mathbb{UPG}_!^{(\mathrm{eff})}$ are inductively constructed.

Example Appendix A.2.13. A dependent p-game $\mathrm{List}_N \in \mathscr{D}^{\mathrm{w}}(N)$ for finite lists of natural numbers is defined by

$$\mathrm{List}_N(\underline{k}^\dagger) := \underbrace{N \,\&\, N \,\&\, \ldots \,\&\, N}_{k} \quad (k \in \mathbb{N}), \qquad\qquad |\mathrm{List}_N| := \bigcup_{k \in \mathbb{N}} \mathrm{List}_N(\underline{k}^\dagger).$$

Next, terms in the CwF $\mathbb{WPG}_!$ are winning, well-bracketed strategies on the following:

Definition Appendix A.2.14 (Pi [Yam23])**.** Let L be a linearly dependent p-game over a p-game Γ, and A be a dependent p-game over Γ. The *linear-Pi* from Γ to L is the p-game $\Pi_\ell(\Gamma, L)$ defined by $|\Pi_\ell(\Gamma, L)| := |L|^{|\Gamma|}$ and for all $\phi : |\Pi_\ell(\Gamma, L)|$

$$\begin{aligned} \Pi_\ell(\Gamma, L)(\phi) := \{\boldsymbol{\epsilon}\} &\cup \{\, \boldsymbol{s}m \in |\Pi_\ell(\Gamma, L)|^{\mathrm{Odd}} \mid \boldsymbol{s} \in \Pi_\ell(\Gamma, L)(\phi), \exists \gamma : \Gamma.\, \boldsymbol{s}m \in L^\star(\gamma)(\phi \circ \gamma)^{\overline{\gamma}_\Gamma} \,\} \\ &\cup \{\, \boldsymbol{t}lr \in |\Pi_\ell(\Gamma, L)|^{\mathrm{Even}} \mid \boldsymbol{t}l \in \Pi_\ell(\Gamma, L)(\phi), \forall \gamma : \Gamma.\, \boldsymbol{t}l \in L^\star(\gamma)(\phi \circ \gamma)^{\overline{\gamma}_\Gamma} \Rightarrow \boldsymbol{t}lr \in L^\star(\gamma)(\phi \circ \gamma)^{\overline{\gamma}_\Gamma} \,\}, \end{aligned}$$

and the *Pi* from Γ to A is the linear-Pi

$$\Pi(\Gamma, A) := \Pi_\ell(!\Gamma, A),$$

where we write $\Gamma \Rightarrow A$ for $\Pi(\Gamma, A)$ if A is constant.

The idea of linear-Pi is that it is linear implication between p-games except that it also satisfies *type dependency*. Here, the type dependency means that the codomain of a linear-Pi $\Pi_\ell(\Gamma, L)$ is the p-game $L(\gamma)$ if the input strategy $\gamma : \Gamma$ on the domain satisfies $\gamma \in \mathbb{WPG}_!(\Gamma)$, and the constant one $\mathscr{P}(|L|)$ otherwise. Similarly to linear implication (Appendix A.2.10), this type dependency on linear-Pi is imposed *only gradually (and often incompletely)* along the gradual (and often incomplete) disclosure of the input strategies by Opponent during a play; *i.e.*, linear-Pi is highly *intensional*. 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 (respectively, Pi) generalises linear implication (respectively, implication): Given p-games Γ and Γ', we have

$$\Pi_\ell(\Gamma, \{\Gamma'\}_\Gamma) = \Gamma \multimap \Gamma', \qquad\qquad \Pi(\Gamma, \{\Gamma'\}_{!\Gamma}) = \Gamma \Rightarrow \Gamma'.$$

Example Appendix A.2.15. There is a strategy $\zeta : \Pi(N, \mathrm{List}_N)$ that plays as the dependent function $n \in \mathbb{N} \mapsto (0, 0, \ldots, 0) \in \mathbb{N}^n$ as follows.

1. If Opponent makes the first move $q_{[k]}$ ($k \in \mathbb{N}_+$) on the codomain $|\mathrm{List}_N|$, where $(\_)_{[k]}$ is the tag on the kth component in the product $N_{[1]} \,\&\, N_{[2]} \,\&\, \ldots$, then ζ asks a question $q_{[0]}$ on the domain $!N_{[0]}$, where $(\_)_{[0]}$ is another tag;
2. Next, if Opponent plays by $q_{[k]}q_{[0]} \mapsto n_{[0]}$ ($n \in \mathbb{N}_+$), then ζ by $q_{[k]}q_{[0]}n_{[0]} \mapsto 0_{[k]}$. If $k \leqslant n$, then $\underline{n}^\dagger \in \mathbb{WPG}_!(!N)$ on the domain is not yet excluded; ζ is compatible with this possibility since its computation so far is within the subgame $N \Rightarrow \mathrm{List}_N(\underline{n}^\dagger) \subseteq |\Pi(N, \mathrm{List}_N)|$.

Finally, comprehensions in the CwF $\mathbb{WPG}_!$ are given by:

Definition Appendix A.2.16 (Sigma [Yam23])**.** The *Sigma* of a p-game Γ and a dependent p-game A over Γ is the p-game $\Sigma(\Gamma, A)$ defined by

- $|\Sigma(\Gamma, A)| := |\Gamma| \,\&\, |A|$,
- $\Sigma(\Gamma, A)(\langle \gamma, \alpha \rangle) := \Gamma(\gamma) \,\&\, A^\star(\gamma^\dagger)(\alpha)$ $(\langle \gamma, \alpha \rangle : |\Sigma(\Gamma, A)|)$,

where we write $\Gamma \,\&\, A$ for $\Sigma(\Gamma, A)$ if A is constant.

The idea is that strategies on the Sigma $\Sigma(\Gamma, A)$ are those $\langle \gamma, \alpha \rangle : |\Gamma| \,\&\, |A|$ that satisfy $\gamma : \Gamma$ and $\alpha : A(\gamma^\dagger)$ if $\gamma \in \mathbb{WPG}_!(\Gamma)$. When A is a constant one $\{\Gamma'\}_{!\Gamma}$, we have $\Sigma(\Gamma, \{\Gamma'\}_{!\Gamma}) \cong \Gamma \,\&\, \Gamma'$. Thus, Sigma generalises product on p-games.

Example Appendix A.2.17. Winning strategies on the Sigma $\Sigma(N, \mathrm{List}_N)$ are those $\langle \underline{k}, \tau \rangle$ with

$$k \in \mathbb{N}, \qquad \tau = \langle \ldots \langle \underline{n_1}, \underline{n_2} \rangle, \ldots, \underline{n_k} \rangle : \mathrm{List}_N(\underline{k}^\dagger), \qquad n_1, n_2, \ldots, n_k \in \mathbb{N},$$

which play as the dependent pairings $(k, (n_1, n_2, \ldots, n_k)) \in \mathbb{N} \times \mathbb{N}^k$. The *strategy filtering and game fixing* in p-games are crucial for the winning of these strategies $\langle \underline{k}, \tau \rangle$.

We are now ready to recall:

Theorem Appendix A.2.18 (a game-semantic CwF [Yam23])**.** *The category $\mathbb{WPG}_!$ gives rise to a CwF as follows:*

- *The terminal p-game $T \in \mathbb{WPG}_!$ (Appendix A.2.2) is a terminal object;*
- *We define*

$$\mathrm{Ty}(\Gamma) := \mathscr{D}^{\mathrm{w}}(\Gamma) \quad (\Gamma \in \mathbb{WPG}_!), \qquad \mathrm{Tm}(\Gamma, A) := \mathbb{WPG}_!(\Pi(\Gamma, A)) \quad (A \in \mathscr{D}^{\mathrm{w}}(\Gamma));$$

- *Given a morphism $\phi : \Delta \to \Gamma$, we define the map $\_\{\phi\} : \mathrm{Ty}(\Gamma) \to \mathrm{Ty}(\Delta)$ by*

$$|A\{\phi\}| := |A|, \qquad A\{\phi\}(\delta_0^\dagger) := A(\phi^\dagger \bullet \delta_0)$$

for all $A \in \mathrm{Ty}(\Gamma)$ and $\delta_0^\dagger \in \mathbb{WPG}_!(!\Delta)$, and the map $\_\{\phi\}_A : \mathrm{Tm}(\Gamma, A) \to \mathrm{Tm}(\Delta, A\{\phi\})$ by

$$\alpha\{\phi\}_A := \alpha \bullet \phi$$

for all $\alpha \in \mathrm{Tm}(\Gamma, A)$;

- *We define*

$$\Gamma.A := \Sigma(\Gamma, A), \qquad \mathrm{p}_A := \mathrm{der}_{|\Gamma|} : \Sigma(\Gamma, A) \to \Gamma, \qquad \mathrm{v}_A := \mathrm{der}_{|A|} : \Pi(\Sigma(\Gamma, A), A\{\mathrm{p}_A\}),$$

$$\langle \phi, \check{\alpha} \rangle_A := \langle \phi, \check{\alpha} \rangle : \Delta \to \Sigma(\Gamma, A) \quad (\check{\alpha} \in \mathrm{Tm}(\Delta, A\{\phi\})).$$

For any $\Gamma \in \mathbb{WPG}_!$ and $A \in \mathscr{D}^{\mathrm{w}}(\Gamma)$, let $\mathbb{WPG}_!(\Gamma, A) := \mathrm{Tm}(\Gamma, A)$. Strictly speaking, a CwF only interprets the core part of MLTT common to all types. For interpreting One-, Zero-, N-, Pi-, Sigma- and Id-types, we need to equip the CwF $\mathbb{WPG}_!$ with the *semantic type-formers* [Hof97, §3.3] that interpret these types. In the following, we only sketch the game-semantic type-formers on the CwF $\mathbb{WPG}_!$, leaving the general definition of semantic type-formers to Hofmann [Hof97, §3.3].

Fix objects $\Delta, \Gamma \in \mathbb{WPG}_!$ and types $A \in \mathscr{D}^{\mathrm{w}}(\Gamma)$ and $B \in \mathscr{D}^{\mathrm{w}}(\Sigma(\Gamma, A))$. For the semantic type-formers of One-, Zero-, N-, Pi- and Sigma-types, it suffices to lift the terminal p-game, the empty p-game, the natural number p-game, Pi and Sigma on dependent p-games, respectively:

Theorem Appendix A.2.19 (game semantics of Pi-types [Yam23])**.** $\mathbb{WPG}_!$ *has Pi-types, where*

- *(*Π*-Form) A dependent p-game* $\Pi(A, B) \in \mathscr{D}^{\mathrm{w}}(\Gamma)$ *is given by*

$$|\Pi(A, B)| := |A| \Rightarrow |B|, \qquad \Pi(A, B)(\gamma_0^\dagger) := \Pi(A(\gamma_0^\dagger), B_{\gamma_0^\dagger}) \quad (\gamma_0^\dagger \in \mathbb{WPG}_!(!\Gamma)),$$

where another dependent p-game $B_{\gamma_0^\dagger} \in \mathscr{D}^{\mathrm{w}}(A(\gamma_0^\dagger))$ *is given by*

$$|B_{\gamma_0^\dagger}| := |B|, \qquad B_{\gamma_0^\dagger}(\alpha_0^\dagger) := B(\langle \gamma_0, \alpha_0 \rangle^\dagger) \quad (\alpha_0^\dagger \in \mathbb{WPG}_!(!A(\gamma_0^\dagger))),$$

and we write $A \Rightarrow B$ *for* $\Pi(A, B)$ *if* $B_{\gamma_0^\dagger}$ *is constant for each* $\gamma_0^\dagger \in \mathbb{WPG}_!(!\Gamma)$*. Note that the equation (*Π*-Subst)*

$$\Pi(A, B)\{\phi\} = \Pi(A\{\phi\}, B\{\phi_A^+\}) \tag{A.1}$$

holds for each morphism $\phi : \Delta \to \Gamma$*, where*

$$\phi_A^+ := \langle \phi \bullet \mathrm{p}, \mathrm{v} \rangle : \Delta.A\{\phi\} \to \Gamma.A.$$

- *(*Π*-Intro) Given a term* $\beta \in \mathbb{WPG}_!(\Sigma(\Gamma, A), B)$*, a term*

$$\lambda_{A,B}(\beta) \in \mathbb{WPG}_!(\Gamma, \Pi(A, B)),$$

where the subscripts $(\_)_{A,B}$ *on* $\lambda_{A,B}$ *are often omitted, is obtained from* β *by adjusting tags with respect to the adjunction between tensor* $\otimes$ *and linear implication* $\multimap$ *[McC98]* via *the evident isomorphism*

$$|!\Sigma(\Gamma, A)| = !(|\Gamma| \,\&\, |A|) \cong !|\Gamma| \otimes !|A|.$$

- *(*Π*-Elim) We define the term*

$$\mathrm{app}_{A,B}(\kappa, \alpha) := \lambda_{A,B}^{-1}(\kappa)\{\overline{\alpha}\} \in \mathbb{WPG}_!(\Gamma, B\{\overline{\alpha}\})$$

for all $\kappa \in \mathbb{WPG}_!(\Gamma, \Pi(A, B))$ *and* $\alpha \in \mathbb{WPG}_!(\Gamma, A)$*, where the subscripts* $(\_)_{A,B}$ *on* $\mathrm{app}_{A,B}$ *are often omitted.*

Theorem Appendix A.2.20 (game semantics of Sigma-types [Yam23]). *The CwF* $\mathbb{WPG}_!$ *has Sigma-types, where*

- *(*Σ*-Form) We define the dependent p-game*

$$\Sigma(A,B) := (|A| \,\&\, |B|, \{\Sigma(A(\gamma_0^\dagger), B_{\gamma_0^\dagger})\}_{\gamma_0^\dagger \in \mathbb{WPG}_!(!\Gamma)}),$$

  where we write $A \,\&\, B$ *for* $\Sigma(A,B)$ *if* $B_{\gamma_0^\dagger}$ *is constant for each* $\gamma_0^\dagger \in \mathbb{WPG}_!(!\Gamma)$.

- *(*Σ*-Intro) By the evident bijection*

$$\Sigma(\Sigma(\Gamma,A),B) \cong \Sigma(\Gamma,\Sigma(A,B)),$$

  we define an isomorphism

$$\mathrm{Pair}_{A,B} := \langle \mathrm{p}_A \bullet \mathrm{p}_B, \langle \mathrm{v}_A\{\mathrm{p}_B\}, \mathrm{v}_B\rangle\rangle : \Sigma(\Sigma(\Gamma,A),B) \xrightarrow{\sim} \Sigma(\Gamma,\Sigma(A,B)).$$

- *(*Σ*-Elim) For a term* $\rho \in \mathbb{WPG}_!(\Sigma(\Sigma(\Gamma,A),B), P\{\mathrm{Pair}_{A,B}\})$*, let*

$$\mathcal{R}^{\Sigma}_{A,B,P}(\rho) := \rho\{\mathrm{Pair}^{-1}_{A,B}\} \in \mathbb{WPG}_!(\Sigma(\Gamma,\Sigma(A,B)),P).$$

Theorem Appendix A.2.21 (game semantics of atomic types [Yam23]). *The CwF* $\mathbb{WPG}_!$ *has One-, Zero- and N-types, where their formation rules are given by the constant dependent p-games at the terminal p-game* T*, the empty p-game* 0 *and the natural number p-game* N*, for which abusing notation we write* 1*,* 0 *and* N*, respectively.*

Finally, we recall the interpretation of Id-types:

Theorem Appendix A.2.22 (game semantics of Id-types [Yam23]). *The CwF* $\mathbb{WPG}_!$ *has Id-types, where*

- *(Id-Form) We define the p-game*

$$T' := \mathscr{P}(\mathrm{flat}(\{\surd\})) \quad \textit{(Appendix A.1.8)},$$

  where $\surd$ *is an arbitrarily fixed element, and the dependent p-game*

$$\mathrm{Id}_A \in \mathscr{D}^{\mathrm{w}}(\Sigma(\Sigma(\Gamma,A),A^+))$$

  by $|\mathrm{Id}_A| := T'$ *and*

$$\mathrm{Id}_A(\langle\langle\gamma_0,\alpha_0\rangle,\alpha_0'\rangle^\dagger) := \begin{cases} (T',\kappa_{T'}) & \textit{if } \alpha_0 = \alpha_0'; \\ (T',\kappa_0) & \textit{otherwise} \end{cases}$$

  for all $\langle\langle\gamma_0,\alpha_0\rangle,\alpha_0'\rangle^\dagger \in \mathbb{WPG}_!(!\Sigma(\Sigma(\Gamma,A),A^+))$*, where* $A^+ := A\{\mathrm{p}_A\} \in \mathrm{Ty}(\Gamma.A)$*, and* κ_X *is the constant family at a game* X*.*

- *(Id-Intro) We define the term*

$$\mathrm{Refl}_A := \langle \overline{\mathrm{v}_A}, \mathrm{refl}_A\rangle \in \mathbb{WPG}_!(\Sigma(\Gamma,A),\Sigma(\Sigma(\Sigma(\Gamma,A),A^+),\mathrm{Id}_A)),$$

  where $\mathrm{refl}_A \in \mathbb{WPG}_!(\Sigma(\Gamma,A),\mathrm{Id}_A\{\overline{\mathrm{v}_A}\})$ *is the strategy* $\underline{\surd} : T'$ *(Appendix A.1.8) up to tags.*

The game-semantic Id-type Id_A embodies 'trivially true' proposition if the two input strategies α_0 and α_0' are equal on the nose, and 'trivially false' one otherwise. Therefore, it is no surprise that those Id-types validate the principle of *uniqueness of identity proofs* [Yam23, §4.6]. It is left open to establish a nontrivial game-semantic interpretation of Id-types that refutes the principle.

Example Appendix A.2.23. The Id-type

$$f : N \Rightarrow N, g : N \Rightarrow N \vdash \mathrm{Id}_{N \Rightarrow N}(f, g) \text{ type}$$

is interpreted in the CwF $\mathbb{WPG}_!$ by the dependent p-game

$$\mathrm{Id}_{N \Rightarrow N} \in \mathscr{D}^{\mathrm{w}}(\Sigma(\Sigma(T, N \Rightarrow N), N \Rightarrow N)).$$

One may then wonder how the Pi

$$\Pi(\Sigma(\Sigma(T, N \Rightarrow N), N \Rightarrow N), \mathrm{Id}_{N \Rightarrow N})$$

works since one never completely knows what the input strategies on the domain given by Opponent are. Nevertheless, the Pi works because a component of the codomain of each Pi (Appendix A.2.14) has to be specified *only gradually* (and often *incompletely*) along the gradual (and often incomplete) disclosure of input strategies on the domain by Opponent in each play [Yam23].