

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/281144361>

Games for Dependent Types

Article · August 2015

DOI: 10.1007/978-3-662-47666-6_3 · Source: arXiv

CITATIONS

6

READS

35

3 authors, including:



[Samson Abramsky](#)

University of Oxford

213 PUBLICATIONS **8,097** CITATIONS

[SEE PROFILE](#)



[Radha Jagadeesan](#)

DePaul University

114 PUBLICATIONS **4,470** CITATIONS

[SEE PROFILE](#)

Games for Dependent Types

Samson Abramsky¹, Radha Jagadeesan² and Matthijs Vákár¹

¹ University of Oxford, Oxford, UK

² DePaul University, Chicago, USA

Abstract. We present a model of dependent type theory (DTT) with Π -, 1 -, Σ - and intensional Id -types, which is based on a slight variation of the category of AJM-games and history-free winning strategies. The model satisfies Streicher’s criteria of intensionality and refutes function extensionality. The principle of uniqueness of identity proofs is satisfied. We show it contains a submodel as a full subcategory which gives a faithful model of DTT with Π -, 1 -, Σ - and intensional Id -types and, additionally, finite inductive type families. This smaller model is fully (and faithfully) complete with respect to the syntax at the type hierarchy built without Id -types, as well as at the class of types where we allow for one strictly positive occurrence of an Id -type. Definability for the full type hierarchy with Id -types remains to be investigated.

1 Introduction

Dependent Type theory (DTT) can be seen as the extension of the simple λ -calculus along the Curry-Howard correspondence from a proof calculus for (intuitionistic) propositional logic to one for predicate logic. It forms the basis of many proof assistants, like NuPRL, LEGO and Coq, and is increasingly being considered as a more expressive type system for programming, as implemented in e.g. ATS, Cayenne, Epigram, Agda and Idris. [1] A recent source of enthusiasm in this field is homotopy type theory (HoTT), which refers to an interpretation of DTT into abstract homotopy theory [2] or, conversely, an extension of DTT that is sufficient to reproduce significant results of homotopy theory [3]. In practice, the latter means DTT with Σ -, Π -, Id -types, a universe satisfying the **univalence axiom**, and certain higher inductive types. The univalence axiom is an extensionality principle which implies, in particular, the axiom of function extensionality [3].

Game semantics provides a unified framework for intensional, computational semantics of various type theories, ranging from pure logics [4] to programming languages [5,6,7] with a variety of features (e.g. non-local control [8], state [9,10,11], non-determinism [12], probability [13], dynamically generated local names [14]) and evaluation strategies [15]. A game semantics for DTT has, surprisingly, so far been absent. Our hope is that such a semantics will provide an alternative analysis of the implications of the subtle shades of intensionality that arise in the analysis of DTT [16,17]. Moreover, the game semantics of DTT is based on very different, one might say orthogonal intuitions to those of the homotopical models: temporal rather than spatial, and directly reflecting the structure of computational processes. One goal, to which we hope this work will

be a stepping stone, is a game semantics of **HoTT** doing justice to both the spatial and temporal aspects of identity types. Indeed, such an investigation might even lead to a computational interpretation of the univalence axiom which has long been missing, although a significant step in this direction was recently taken by the constructive cubical sets model of **HoTT** [18].

Our game theoretic model of **DTT** is inspired in part by the domain model of **DTT** [19]. This model views a type family as a continuous function to a domain of domains, a witness of a Π -type $\Pi_{x:A}B$ as a continuous (set theoretic) dependent function and interprets identity types via a kind of intersection. We follow this recipe for modelling type families and identity types. We adapt the viewpoint of the game semantics of system **F** [7] to describe the Π -type to capture the intuitive idea that the specialisation of a term at type $\Pi_{x:A}B$ to a specific instance $B[a/x]$ is the responsibility solely of the context that provides the argument a of type A ; in contrast, any valid term of $\Pi_{x:A}B$ has to operate within the constraints enforced by the context. Our definition draws its power from the fact that in a game semantics, these constraints are enforced not only on completed computations, but also on the incomplete computations that arise when a term interacts with its context. Thus, while we follow the formal recipes of [19], the temporal character of game semantics results in strikingly different properties of the resulting model.

In the rest of this paper, we describe a game theoretic model of **DTT** with 1-, Σ -, Π - and intensional **Id**-types, where (lists of dependent) **AJM**-games interpret types and (lists of) history-free winning strategies on games of dependent functions interpret terms. We next specialize to the semantic type hierarchy formed by the 1-, Σ -, Π -constructions and substitution over a set of finite dependent games. We show that this gives of model of **DTT** which additionally supports finite inductive type families. Our two models have the following key properties.

- The place of the **Id**-types in the intensionality spectrum compares as follows with the domain semantics and with **HoTT**.

	Domains	HoTT	Games
Failure of Equality Reflection	✓	✓	✓
Streicher Intensionality Criteria (<i>I1</i>) and (<i>I2</i>)	✓	✓	✓
Streicher Intensionality Criterion (<i>I3</i>)	✗	✗	✓
Failure of Function Extensionality (FunExt)	✗	✗	✓
Failure of Uniqueness of Identity Proofs (UIP)	✗	✓	✗

- We show that the smaller model faithfully models the terms of a version of **DTT** with 1-, Σ -, Π - and **Id**-types and finite inductive type families. Moreover, it is fully complete at the types which do not involve **Id** in their construction or which involve one strictly positive **Id**-type as a subformula. In contrast, the domain theoretic model of [19] is not (fully) complete or faithful.

2 A Category of Games

The idea behind game semantics is to model a computation by an alternating sequence of interactions (the play) between a program (Player) and its envi-

ronment (Opponent), following some rules specified by its datatype (the game). In this translation, programs become Player strategies, while termination corresponds to a strategy being winning or beating all Opponents. The charm of this interpretation is that it not only fully captures the intensional aspects of a program but that it combines this with the structural clarity of a categorical model, thus interpolating between traditional operational and denotational semantics.

We assume the reader has some familiarity with the basics of categories of AJM-games and strategies, as described in [20], and only briefly recall the definitions. We define a category **Game** which has as objects AJM-games.

Definition 1 (Game). A *game* A is a tuple $(M_A, \lambda_A, P_A, \approx_A, W_A)$, where

- M_A is a countable set of **moves**;
- $M_A \xrightarrow{\lambda_A = \langle \lambda_A^{OP}, \lambda_A^{QA} \rangle} \{O, P\} \times \{Q, A\}$ is a function which indicates if a move is made by **Opponent** (O) or **Player** (P) and if it is a **Question** (Q) or an **Answer** (A), for which we write $\overline{O} = P$, $\overline{P} = O$ and $M_A^O := \lambda_A^{OP^{-1}}(O)$, $M_A^P := \lambda_A^{OP^{-1}}(P)$, $M_A^Q := \lambda_A^{QA^{-1}}(Q)$ and $M_A^A := \lambda_A^{QA^{-1}}(A)$;
- $P_A \subseteq M_A^{\oplus}$ is a non-empty prefix-closed set of **plays**, where M_A^{\oplus} is the set of finite sequences of uniquely occurring moves, with the properties
 - (p1) $s = at \Rightarrow a \in M_A^O$;
 - (p2) $\forall_i \lambda_A^{OP}(s_{i+1}) = \overline{\lambda_A^{OP}(s_i)}$, where we write s_i for the i -th move in s ;
 - (p3) $\forall_{t \leq s} |t|_{M_A^A} \leq |t|_{M_A^Q}$.
 Here, \leq denotes the prefix order and $|s|$ the length of a sequence. Write $j_{A,s}(m)$ for the last unanswered question preceding an answer m in a play s , which we say m answers. $j_{A,s}$ will be used to enforce **stack discipline**.
- \approx_A is an equivalence relation on P_A , satisfying
 - (e1) $s \approx_A t \Rightarrow \lambda_A^*(s) = \lambda_A^*(t)$;
 - (e2) $s \approx_A t \wedge s' \leq s \wedge t' \leq t \wedge |s'| = |t'| \Rightarrow s' \approx_A t'$;
 - (e3) $s \approx_A t \wedge sa \in P_A \Rightarrow \exists_b sa \approx_A tb$.
 Here, λ_A^* is the extension of λ_A to sequences.
- $W_A \subseteq P_A^\infty$ is a set of **winning plays**, where P_A^∞ is the set of infinite plays, i.e. infinite sequences of moves such that all their finite prefixes are in P_A , such that W_A is closed under \approx_A in the sense that

$$(s \in W_A \wedge t \notin W_A) \Rightarrow \exists_{s_0 \leq s, t_0 \leq t} |s_0| = |t_0| \wedge s_0 \not\approx_A t_0.$$

Our notion of morphism will be defined in terms of strategies on games.

Definition 2 (Strategy). A *strategy on* A is a subset $\sigma \subseteq P_A^{\text{even}}$ satisfying

- (Causal Consistency): $sab \in \sigma \Rightarrow s \in \sigma$;
- (Representation Independence): $s \in \sigma \wedge s \approx_A t \Rightarrow t \in \sigma$;
- (Determinacy): $sab, ta'b' \in \sigma \wedge sa \approx_A ta' \Rightarrow sab \approx_A ta'b'$.

We write $\text{str}(A)$ for the set of strategies on A . We sometimes identify σ with the subset of P_A that is obtained as its prefix closure. In fact, we restrict to history-free strategies, as we are modelling computation without mutable state.

Definition 3 (History-Free Strategy). We call a strategy $\sigma \in \text{str}(A)$ *history-free*, if there exists a non-empty causally consistent subset $\phi \subseteq \sigma$ (called a *history-free skeleton*) such that

- (Uniformization): $\forall_{sab \in \sigma} s \in \phi \Rightarrow \exists!_{b'} sab' \in \phi$;
- (History-Freeness 1): $sab, tac \in \phi \Rightarrow b = c$;
- (History-Freeness 2): $(sab, t \in \phi \wedge ta \in P_A) \Rightarrow tab \in \phi$.

Then, ϕ is induced by a partial function on moves and $\sigma = \{t \mid \exists_{s \in \phi} t \approx_A s\}$.

From now on, we assume strategies to be history-free. Winning conditions give rise to the notion of a winning strategy, the semantic equivalent of a normalising or total term. A winning strategy always has a response to any valid O -move. Furthermore, if the result of the interaction between a strategy and Opponent is an infinite play, then this is a member of the set of winning plays.

Definition 4 (Winning Strategy). A strategy $\sigma \in \text{str}(A)$ is *winning* if it satisfies

- (Finite Wins): If s is \leq -maximal in σ , then s is \leq -maximal in P_A .
- (Infinite Wins): If $s_0 \leq s_1 \leq \dots$ is an infinite chain in σ , then $\bigcup_i s_i \in W_A$.

We write $\text{wstr}(A)$ for the set of winning strategies on A . Next, we define some constructions on games, starting with their symmetric monoidal closed structure.

Definition 5 (Tensor Unit). We define the game $I := (\emptyset, \emptyset, \{\epsilon\}, \{(\epsilon, \epsilon)\}, \emptyset)$.

Definition 6 (Tensor). For games A, B , we define

$A \otimes B := (M_A + M_B = \Sigma_{i \in \{A, B\}} M_i, [\lambda_A, \lambda_B], P_{A \otimes B}, \approx_{A \otimes B}, W_{A \otimes B})$ with

- $P_{A \otimes B} = \{s \mid s \upharpoonright_A \in P_A \wedge s \upharpoonright_B \in P_B \wedge \text{fst}^*(j_{A \otimes B, s}^*(s \upharpoonright_{M_{A \otimes B}^A})) = \text{fst}^*(s \upharpoonright_{M_{A \otimes B}^A})\}$;
- $s \approx_{A \otimes B} t := s \upharpoonright_A \approx_A t \upharpoonright_A \wedge s \upharpoonright_B \approx_B t \upharpoonright_B \wedge \forall_{1 \leq i \leq |s|} s_i \in M_A \Leftrightarrow t_i \in M_A$;
- $W_{A \otimes B} := \{s \in P_{A \otimes B}^\infty \mid (s \upharpoonright_A \in P_A^\infty \Rightarrow s \upharpoonright_A \in W_A) \wedge (s \upharpoonright_B \in P_B^\infty \Rightarrow s \upharpoonright_B \in W_B)\}$.

Definition 7 (Linear Implication). For games A, B , we define

$A \multimap B := (M_A + M_B = \Sigma_{i \in \{A, B\}} M_i, [\overline{\lambda}_A, \lambda_B], P_{A \multimap B}, \approx_{A \multimap B}, W_{A \multimap B})$ with

- $P_{A \multimap B} = \{s \mid s \upharpoonright_A \in P_A \wedge s \upharpoonright_B \in P_B \wedge \text{fst}^*(j_{A \multimap B, s}^*(s \upharpoonright_{M_{A \multimap B}^A})) = \text{fst}^*(s \upharpoonright_{M_{A \multimap B}^A})\}$;
- $s \approx_{A \multimap B} t := s \upharpoonright_A \approx_A t \upharpoonright_A \wedge s \upharpoonright_B \approx_B t \upharpoonright_B \wedge \forall_{1 \leq i \leq |s|} s_i \in M_A \Leftrightarrow t_i \in M_A$;
- $W_{A \multimap B} := \{s \in P_{A \multimap B}^\infty \mid s \upharpoonright_A \in W_A \Rightarrow s \upharpoonright_B \in W_B\}$.

Note that the definitions of λ_- imply that in $A \otimes B$ only Opponent can switch between A and B , while in $A \multimap B$ only Player can. These definitions on objects extend to strategies, e.g. for (winning) strategies $\sigma \in \text{str}(A), \tau \in \text{str}(B)$, we can define a (winning) strategy $\sigma \otimes \tau = \{s \in P_{A \otimes B}^{\text{even}} \mid s \upharpoonright_A \in \sigma \wedge s \upharpoonright_B \in \tau\} \in \text{str}(A \otimes B)$. This gives us a model of multiplicative intuitionistic linear logic, with all structural morphisms consisting of appropriate variants of copycat strategies.

Theorem 1 (Linear Category of Games). We define a category **Game** by

- $\text{ob}(\mathbf{Game}) := \{A \mid A \text{ is an AJM-game}\};$
- $\mathbf{Game}(A, B) := \text{wstr}(A \multimap B);$
- $\text{id}_A := \{s \in P_{A \multimap A} \mid s \Vdash_{A(1)} \approx_A s \Vdash_{A(2)}\}$, the *copycat strategy* on A ;
- for $A \xrightarrow{\sigma} B \xrightarrow{\tau} C$, the composition (or *interaction*) $A \xrightarrow{\sigma;\tau} C$ is defined from parallel composition $\sigma \parallel \tau := \{s \in M_{(A \multimap B) \multimap C}^{\otimes} \mid s \Vdash_{A,B} \in \sigma \wedge s \Vdash_{B,C} \in \tau\}$ plus hiding: $\sigma; \tau := \{s \Vdash_{A,C} \mid s \in \sigma \parallel \tau\}.$

Then, $(\mathbf{Game}, I, \otimes, \multimap)$ is, in fact, a symmetric monoidal closed category.

To make this into a model of intuitionistic logic, a Cartesian closed category (ccc), through the (first) Girard translation, we need two more constructions on games, to interpret the additive conjunction $\&$ and exponential $!$, respectively. A play in $!A$ consists of any number of interleaved threads of plays in A .

Definition 8 (With). We define the game

$$A \& B := (M_A + M_B, [\lambda_A, \lambda_B], P_A + P_B, \approx_A + \approx_B, W_A + W_B).$$

Definition 9 (Bang). We define $!A := (\mathbb{N} \times M_A, \lambda_A \circ \text{snd}, P_{!A}, \approx_{!A}, W_{!A})$ with

- $P_{!A} = \{s \mid \forall i \in \mathbb{N} s \Vdash_i \in P_A \wedge \text{fst}^*(j_{!A,s}^*(s \Vdash_{M_{!A}^A})) = \text{fst}^*(s \Vdash_{M_{!A}^A})\};$
- $s \approx_{!A} t := \exists \pi \in S(\mathbb{N}) \forall i \in \mathbb{N} s \Vdash_{i \approx_A} t \Vdash_{\pi(i)} \wedge (\pi \circ \text{fst})^*(s) = \text{fst}^*(t);$
- $W_{!A} := \{s \in P_{!A}^\infty \mid \forall i s \Vdash_i \in P_A^\infty \Rightarrow s \Vdash_i \in W_A\}.$

Next, we note that $!$ can be made into a co-monad by defining, for $A \xrightarrow{\sigma} B$,

$$! \sigma := \{s \in P_{!A \multimap !B}^{\text{even}} \mid \exists \pi \in S(\mathbb{N}) \forall i \in \mathbb{N} s \Vdash_{(\pi(i), A), (i, B)} \in \sigma\},$$

and natural transformations

$$!A \xrightarrow{\text{der}_A} A := \{s \in P_{!A \multimap A}^{\text{even}} \mid \exists i \in \mathbb{N} s \Vdash_{!A} \Vdash_i \approx_A s \Vdash_A\} \quad \text{and}$$

$$!A \xrightarrow{\delta_A} !!A := \{s \in P_{!A \multimap !!A}^{\text{even}} \mid \exists p: \mathbb{N} \times \mathbb{N} \hookrightarrow \mathbb{N} \forall i, j \in \mathbb{N} s \Vdash_{!A} \Vdash_{p(i,j)} \approx_A s \Vdash_{!!A} \Vdash_i \Vdash_j\}.$$

This allows us to define the co-Kleisli category $\mathbf{Game}_!$, which has the same objects as \mathbf{Game} , while $\mathbf{Game}_!(A, B) := \mathbf{Game}(!A, B)$. We have a composition $(f, g) \mapsto f^\dagger; g$, where we write $f^\dagger := \delta_{\text{dom}(f)}; !f$, for which the strategies der_A serve as identities. We can define finite products in $\mathbf{Game}_!$ by I and $\&$ and write $\text{diag}_A := \{s \in P_{!A \multimap (A \& A)}^{\text{even}} \mid \exists i \in \mathbb{N} (s \Vdash_i \in \epsilon) \vee (s \Vdash_{!A} \Vdash_i \approx_A s \Vdash_{A(1)} \neq \epsilon) \vee (s \Vdash_{!A} \Vdash_i \approx_A s \Vdash_{A(2)} \neq \epsilon)\}$ for the diagonal $!A \longrightarrow A \& A$. Moreover, we have Seely-isomorphisms $!I \cong I$ and $!(A \& B) \cong !A \otimes !B$, so we obtain a linear-non-linear adjunction $\mathbf{Game} \rightleftarrows \mathbf{Game}_!$, hence a model of multiplicative exponential intuitionistic linear logic. In particular, by defining $A \Rightarrow B := !A \multimap B$, we get a ccc. We write $\text{comp}_{A,B,C}$ for the internal composition $((A \Rightarrow B) \& (B \Rightarrow C)) \longrightarrow A \Rightarrow C$ in $\mathbf{Game}_!$.

Theorem 2 (Intuitionist Category of Games). $(\mathbf{Game}_!, I, \&, \Rightarrow)$ is a ccc.

Note that for the hierarchy of intuitionistic types A that are formed by operations I , $\&$ and \Rightarrow from finite games, winning strategies are the total strategies - strategies which respond to any O -move - for which infinite chattering can only occur because Opponent opens infinitely many threads of the same game.

3 Dependent Games

The previous section sketched how $\mathbf{Game}_!$ models simple intuitionistic type theory. Next, we show how it comes equipped with a notion of dependent type. This leads to an indexed ccc $\mathbf{DGame}_!$ of dependent games and strategies.

We define a poset \mathbf{Game}_{\leq} of games with $A \leq B := (M_A = M_B) \wedge (\lambda_B|_{M_A} = \lambda_A) \wedge (P_A \subseteq P_B) \wedge (s \approx_A t \Leftrightarrow s \in P_A \wedge s \approx_B t) \wedge (W_A = W_B \cap P_A^\infty)$. Given a game C , we define the cpo $\mathbf{Sub}(C)$ as the poset of its \leq -subgames. We note that, for $A, B \in \mathbf{Sub}(C)$, $A \leq B \Leftrightarrow P_A \subseteq P_B$.

For a game A , we define the set $\mathbf{ob}(\mathbf{DGame}_!(A))$ of **games with dependency on A** as the set of continuous functions $\mathbf{str}(A) \xrightarrow{B} \mathbf{Sub}(\odot B)$ for some other game $\odot B$. We note that $\mathbf{ob}(\mathbf{DGame}_!(I))$ is the set of pairs $A = (A(\perp), \odot A)$ where $A(\perp) \leq \odot A$, of which $\mathbf{ob}(\mathbf{Game}_!)$ arises as the proper subset of diagonal elements (A, A) . We define more generally $\mathbf{ob}(\mathbf{DGame}_!(A)) := \mathbf{ob}(\mathbf{DGame}_!(\odot A))$. Writing $s \mapsto \bar{s}$ for the function $P_A \rightarrow \mathcal{P}(P_A)$ inductively defined on the empty play, Opponent moves and Player moves, respectively, as $\epsilon \mapsto \emptyset$, $s(i, a) \mapsto \bar{s}$, $s(i, a)(i, b) \mapsto \overline{s(i, a)} \cup \{t \mid \exists s' \in \bar{s} t \approx_A s'ab\}$, we define the dependent function space as follows.

Definition 10 (Π -Game). *Given $B \in \mathbf{ob}(\mathbf{DGame}_!(A))$, we define the Π -game $(\Pi_A B)(\perp) \leq \odot A \Rightarrow \odot B$ inductively as*

$$\{\epsilon\} \cup \{sa \mid s \in P_{(\Pi_A B)(\perp)}^{\text{even}} \wedge \exists \overline{sa|_{\odot A}} \subseteq \tau \in \mathbf{wstr}(A(\perp)) sa \in P_{A(\perp) \Rightarrow B(\tau)}\} \cup \{sab \mid sa \in P_{(\Pi_A B)(\perp)}^{\text{odd}} \wedge \forall \overline{sab|_{\odot A}} \subseteq \tau \in \mathbf{wstr}(A(\perp)) sa \in P_{A(\perp) \Rightarrow B(\tau)} \Rightarrow sab \in P_{A(\perp) \Rightarrow B(\tau)}\}.$$

We note that we can make $\mathbf{DGame}_!(A)$ into a ccc by defining I and $\&$ pointwise on dependent games B , while also performing the operation on $\odot B$, and by defining $\odot B \Rightarrow C := \odot B \Rightarrow \odot C$ and $P_{(B \Rightarrow C)(\sigma)} := \{s \in P_{B(\sigma) \Rightarrow C(\sigma)} \mid \exists \tau \in \mathbf{wstr}(B(\sigma)) s \upharpoonright_{B(\sigma)} \subseteq \tau\}$. This lets us define $\mathbf{DGame}_!(A)(B, C) := \mathbf{wstr}(\mathbf{O-sat}(\Pi_A(B \Rightarrow C)))$ with the obvious identities and composition, which we discuss later. Here, the game $\mathbf{O-sat}(A(\perp), \odot A)$ has plays $\{\epsilon\} \cup \{sa \in P_{\odot A}^{\text{odd}} \mid s \in P_{\mathbf{O-sat}(A(\perp), \odot A)}^{\text{even}}\} \cup \{sab \in P_{\odot A}^{\text{even}} \mid sa \in P_{\mathbf{O-sat}(A(\perp), \odot A)}^{\text{odd}} \wedge (sa \in P_{A(\perp)} \Rightarrow sab \in P_{A(\perp)})\}$. Explicitly, we have the **game $\mathbf{O-sat}(\Pi_A B)$ of dependent functions from A to B**

$$\{\epsilon\} \cup \{sa \mid s \in P_{\mathbf{O-sat}(\Pi_A B)}^{\text{even}}\} \cup \{sab \mid sa \in P_{\mathbf{O-sat}(\Pi_A B)}^{\text{odd}} \wedge \forall \overline{sab|_{\odot A}} \subseteq \tau \in \mathbf{wstr}(A(\perp)) sa \in P_{A(\perp) \Rightarrow B(\tau)} \Rightarrow sab \in P_{A(\perp) \Rightarrow B(\tau)}\}.$$

Following the mantra of game semantics for quantifiers [7], in $\mathbf{O-sat}(\Pi_A B)$, Opponent can choose a winning strategy τ on $A(\perp)$ while Player has to play in a way that is compatible with all choices of τ that have not yet been excluded. Similarly to the approach taken in the game semantics for polymorphism [7], we do not specify all of τ in one go, as this would violate “Scott’s axiom” of continuity of computation. Instead, τ is gradually revealed, explicitly so by playing in $! \odot A$ and implicitly by playing in $\odot B$. That is, unless Opponent behaves naughtily, in the sense that there is no winning history-free strategy τ on $A(\perp)$ which is consistent with her behaviour while $s \upharpoonright_{\odot B}$ obeys the rules of $B(\tau)$. In case of such a naughty Opponent, any further play in $\odot A \Rightarrow \odot B$ is permitted.

For an example, let $\text{days}(n) := \{m \mid \text{there are } > m \text{ days in the year } n\}$ and define $\widetilde{\text{days}}_*(\perp) = \emptyset_*$, $\widetilde{\text{days}}_*(n) := \text{days}(n)_*$ to obtain a game depending on \widetilde{N}_* (with $\text{days}(n) = \widetilde{N}_{<365*}$ or $\widetilde{N}_{<366*}$). Here, \widetilde{X}_* signifies the game with $P_{\widetilde{X}_*} = \{\epsilon, *\} \cup \{*x \mid x \in X\}$ and $\approx_X = \text{id}_X$. Then, the following are valid strategies.

$\widetilde{!N}_*$	\widetilde{days}_*	$\widetilde{!N}_*$	\widetilde{days}_*	$\widetilde{!N}_*$	\widetilde{days}_*	$\widetilde{!N}_*$	$\widetilde{!days}_*$	\widetilde{days}_*	
*	*	*	*	*	*	*	*	*	O
364	$(i, *)$			$(i, *)$			$(i, *)$		P
	$(i, 1984)$			$(i, 1984)$			(i, m)		O
		365		$(i + 1, *)$				m	P
				$(i + 1, 1985)$					O
					365				P

Fig. 1. Three strategies on $\text{O-sat}(\Pi_{\widetilde{\text{N}}_*}, \text{days}_*)$ and one on $\text{O-sat}(\Pi_{\widetilde{\text{N}}_*}, \text{days}_* \multimap \text{days}_*)$. The first as all years have > 364 days, the second as 1984 was a leap year, the third as Player can play any move in $\odot \text{days}_* = \widetilde{\text{N}}_{<366*}$ after Opponent has not played along a strategy on $\widetilde{\text{N}}_*$ and the fourth as Opponent makes the move m first, after which Player can safely copy it. In the paired moves, Player chooses an (irrelevant) index i .

The fourth example is especially important, as it generalises to a (derelicted) B -copycat on $\mathbf{O}\text{-sat}(\Pi_{!A}(B \multimap B))$ for arbitrary B , denoted $\mathbf{v}_{[A],[B]}$ in section 4. This motivates why Opponent can narrow down the fibre of B freely, while Player cannot. To see that Player should not be able to narrow down the fibre of B , note that we do not want $f := \{\epsilon, *365\}$ to define a strategy on $\mathbf{O}\text{-sat}(\Pi_{!N_*} \widetilde{\text{days}_*})$, as $1983; f = \{\epsilon, *365\} \notin \text{str}(\widetilde{\text{days}_*(1983)})$.

Theorem 3. *We obtain a strict indexed ccc $\mathbf{DGame}_! (I)^{op} \xrightarrow{(\mathbf{DGame}_!, -\{-\})} \mathbf{Cat}$ of dependent games, if we define*

- *fibrewise object sets* $\mathbf{ob}(\mathbf{DGame}_!(A)) := \{\mathbf{str}(\odot A) \xrightarrow{B} \mathbf{Sub}(\odot B) \mid \odot B \in \mathbf{ob}(\mathbf{Game}_!) \wedge B \text{ continuous}\};$
- *fibrewise hom-sets* $\mathbf{DGame}_!(A)(B, C) := \mathbf{wstr}(\mathbf{O-sat}(\Pi_{!A}(!B \rightarrow C)));$
- *fibrewise identities* $\mathbf{der}_B := \{s \in P_{\mathbf{O-sat}(\Pi_{!A}(!B \rightarrow B))} \mid \exists i s \upharpoonright !B \downarrow i \approx_B s \upharpoonright B\};$
- *if* $B \xrightarrow{\tau} C \xrightarrow{\tau'} D \in \mathbf{DGame}_!(A)$, $\tau^\dagger;_A \tau' := \mathbf{diag}_A^\dagger; \tau^\dagger \otimes \tau'; \mathbf{comp}_{\odot B, \odot C, \odot D};$
- *given* $f \in \mathbf{Game}_!(A', A)$, *we define the change of base functor* $-\{f\}: B\{f\} \in \mathbf{ob}(\mathbf{DGame}_!(A'))$ *where* $B\{f\}(\sigma) := B(!(\sigma); f)$ *and* $\odot B\{f\} := \odot B$ *and* $\tau\{f\} := f^\dagger; \tau.$

Seeing that $\mathbf{DGame}_!(I)$ additionally has a terminal object I to interpret the empty context, we are well on our way to producing a model of dependent type theory [21]: we only need to interpret context extension. This takes the form of the comprehension axiom for $\mathbf{DGame}_!$, which states that for each $A \in \mathbf{ob}(\mathbf{DGame}_!(I))$ and $B \in \mathbf{ob}(\mathbf{DGame}_!(A))$ the following presheaf is representable

$$x \mapsto \mathbf{DGame}_!(\text{dom}(x))(I, B\{x\}) : (\mathbf{DGame}_!(I)/A)^{op} \longrightarrow \mathbf{Set}.$$

Unfortunately, this fails, as $\mathbf{DGame}_!(I)$ does not yield a sound interpretation of dependent contexts. Essentially, the problem is that we do not have **additive** Σ -types, appropriate generalisations $\Sigma_A^{\&} B$ of $\&$ to interpret dependent context extension in $\mathbf{DGame}_!(I)$.

Theorem 4. $\mathbf{DGame}_!$ does not satisfy the comprehension axiom.

4 A Category with Families of Context Games

All is not lost, however. In fact, we have almost translated the structural core of the syntax of DTT into the world of games and strategies. The remaining generalisation, necessitated by the lack of additive Σ -types, is to dependent games depending on multiple (mutually dependent) games. We can produce a categorical model of DTT out of the resulting structure by applying a so-called **category of contexts (Ctxt) construction**, which is precisely how one builds a categorical model from the syntax of dependent type theory [17,22]. This can be seen as a way of making our indexed category satisfy the comprehension axiom, extending its base category by (inductively) adjoining (strong) Σ -types formally, analogous to the Fam-construction of [15] which adds formal co-products.

The problem which needs to be addressed is how to interpret dependent types and dependent functions of more variables. This is done through a notion of context game and a generalisation of the Π -game construction from section 3.

Definition 11 (Context Game). We define a **context game** $[X_i]_{1 \leq i \leq n}$ to be a list where X_i is a game with dependency on $[X_j]_{j < i}$, i.e. a continuous function $\text{str}(\odot X_1) \times \cdots \times \text{str}(\odot X_{i-1}) \xrightarrow{X_i} \text{Sub}(\odot X_i)$ for some game $\odot X_i$.

Definition 12 (Dependent Π -game). For a game X_{n+1} depending on $[X_i]_{i \leq n}$, we define the game $\Pi_{X_n} X_{n+1}$ depending on $[X_i]_{i \leq n-1}$ by $\odot \Pi_{X_n} X_{n+1} := \odot X_n \Rightarrow \odot X_{n+1}$ from which $(\Pi_{X_n} X_{n+1})(\sigma_1, \dots, \sigma_{n-1})$ is carved out as

$$\begin{aligned} & \{\epsilon\} \bigcup \{sa \mid s \in P_{(\Pi_{X_n} X_{n+1})(\sigma_1, \dots, \sigma_{n-1})}^{\text{even}} \wedge \exists \overline{sa \upharpoonright_{\odot X_n}} \subseteq \tau \in \text{wstr}(X_n(\sigma_1, \dots, \sigma_{n-1})) \\ & sa \in P_{X_n(\sigma_1, \dots, \sigma_{n-1}) \Rightarrow X_{n+1}(\sigma_1, \dots, \sigma_{n-1}, \tau)} \} \bigcup \{sab \mid sa \in P_{(\Pi_{X_n} X_{n+1})(\sigma_1, \dots, \sigma_{n-1})}^{\text{odd}} \\ & \wedge \forall \overline{sab \upharpoonright_{\odot X_n}} \subseteq \tau \in \text{wstr}(X_n(\sigma_1, \dots, \sigma_{n-1})) sa \in P_{X_n(\sigma_1, \dots, \sigma_{n-1}) \Rightarrow X_{n+1}(\sigma_1, \dots, \sigma_{n-1}, \tau)} \Rightarrow \\ & sab \in P_{X_n(\sigma_1, \dots, \sigma_{n-1}) \Rightarrow X_{n+1}(\sigma_1, \dots, \sigma_{n-1}, \tau)} \}. \end{aligned}$$

Consequently, the **game of dependent functions of multiple arguments** $\text{O-sat}(\Pi_{X_1} \cdots \Pi_{X_n} X_{n+1})$ is carved out in $\odot X_1 \Rightarrow \cdots \Rightarrow \odot X_n \Rightarrow \odot X_{n+1}$ as

$$\begin{aligned} & \{\epsilon\} \bigcup \{sa \mid s \in P_{\text{O-sat}(\Pi_{X_1} \cdots \Pi_{X_n} X_{n+1})}^{\text{even}} \} \bigcup \{sab \mid sa \in P_{\text{O-sat}(\Pi_{X_1} \cdots \Pi_{X_n} X_{n+1})}^{\text{odd}} \wedge \\ & \forall \overline{sab \upharpoonright_{\odot X_1}} \subseteq \tau_1 \in \text{wstr}(X_1(\perp)) \cdots \forall \overline{sab \upharpoonright_{\odot X_n}} \subseteq \tau_n \in \text{wstr}(X_n(\tau_1, \dots, \tau_{n-1})) \\ & sa \in P_{X_1(\perp) \Rightarrow \cdots \Rightarrow X_{n+1}(\tau_1, \dots, \tau_n)} \Rightarrow sab \in P_{X_1(\perp) \Rightarrow \cdots \Rightarrow X_{n+1}(\tau_1, \dots, \tau_n)} \}. \end{aligned}$$

For illustration, define a game $\widetilde{\text{RA}}_*$ depending on the context game $[\widetilde{\mathbb{N}}_*, \widetilde{\text{days}}_*]$ by $\text{RA}(n, m) := \{\text{Rick Astley lyrics from songs released before day } m \text{ of year } n\}$.

			Games for Dependent Types			
$\widetilde{\mathbb{N}}_*$	$\widetilde{\text{!days}}_*$	$\widetilde{\text{RA}}_*$	$\widetilde{\mathbb{N}}_*$	$\widetilde{\text{!days}}_*$	$\widetilde{\text{RA}}_*$	
	$(i, *)$	*	$(i, *)$			O
	$(i, m > 206)$		$(i, n > 1987)$			P
$(j, *)$				Never Gonna Let You Down		O
$(j, 1987)$						P
	Never Gonna Give You Up					O
						P

Fig. 2. Two examples of (partial) strategies on $\text{O-sat}(\Pi_{\widetilde{\mathbb{N}}_*} \Pi_{\widetilde{\text{!days}}_*} \widetilde{\text{RA}}_*)$.

We define a category $\text{Ctx}(\mathbf{DGame}_1)$ with objects context games and morphisms which are defined inductively as (dependent) lists of winning strategies on appropriate games of dependent functions. We show that this has the structure of a category with families (CwF) [17], a canonical notion of model of DTT. This gives a more concise presentation of the resulting indexed category with comprehension, where we also add formal Σ -types in the fibres.

Definition 13 (CwF). A CwF is a category \mathcal{C} with a terminal object \cdot , for all objects Γ a set $\text{Ty}(\Gamma)$, for all $A \in \text{Ty}(\Gamma)$ a set $\text{Tm}(\Gamma, A)$, for all $\Gamma' \xrightarrow{f} \Gamma$ in \mathcal{C} functions $\text{Ty}(\Gamma) \xrightarrow{-\{f\}} \text{Ty}(\Gamma')$ and $\text{Tm}(\Gamma, A) \xrightarrow{-\{f\}} \text{Tm}(\Gamma', A\{f\})$, such that

$$\begin{aligned} A\{\text{id}_\Gamma\} &= A & (Ty\text{-Id}) & & A\{g \circ f\} &= A\{g\}\{f\} & (Ty\text{-Comp}) \\ t\{\text{id}_\Gamma\} &= A & (Tm\text{-Id}) & & t\{g \circ f\} &= t\{g\}\{f\} & (Tm\text{-Comp}), \end{aligned}$$

for $A \in \text{Ty}(\Gamma)$ a morphism $\Gamma.A \xrightarrow{\text{p}_{\Gamma,A}} \Gamma$ of \mathcal{C} and $\mathbf{v}_{\Gamma,A} \in \text{Tm}(\Gamma.A, A\{\mathbf{p}_{\Gamma,A}\})$ and, finally, for all $t \in \text{Tm}(\Gamma', A\{f\})$ a morphism $\Gamma' \xrightarrow{\langle f, t \rangle} \Gamma.A$ such that

$$\begin{aligned} \mathbf{p}_{\Gamma,A} \circ \langle f, t \rangle &= f & (Cons\text{-}L) & & \mathbf{v}_{\Gamma,A}\{\langle f, t \rangle\} &= t & (Cons\text{-}R) \\ \langle \mathbf{p}_{\Gamma,A}, \mathbf{v}_{\Gamma,A} \rangle &= \text{id}_{\Gamma.A} & (Cons\text{-}Id) & & \langle f, t \rangle \circ g &= \langle f \circ g, t\{g\} \rangle & (Cons\text{-}Nat). \end{aligned}$$

Theorem 5. We have a CwF $(\text{Ctx}(\mathbf{DGame}_1), \text{Ty}, \text{Tm}, \mathbf{p}, \mathbf{v}, -, \circ, \langle -, - \rangle)$.

We define the required structures. All equations follow trivially from the definitions and the two claims stated. We define $\text{Ty}([X_i]_i)$ as the set of **context games with dependency** on $[X_i]_i$: $[Y_j]_j \in \text{Ty}([X_i]_i)$ iff $[X_i]_i.[Y_j]_j := [X_1, \dots, X_n, Y_1, \dots, Y_m]$ is a context game, while $\cdot := []$ is the terminal object.

Next, $\text{mor}(\mathcal{C})$ and $-\{-\}_{\text{Ty}}$ are defined with $\odot Y_j \{[f_k]_{k < j}\} = \odot Y_j$ and $\text{Ctx}(\mathbf{DGame}_1)([X_i]_{i \leq n}, [Y_j]_{j \leq m}) := \{[f_j]_{j \leq m} \mid f_j \in \text{wstr}(\text{O-sat}(\Pi_{X_1} \dots \Pi_{X_n} Y_j \{[f_k]_{k < j}\}))\}$ $Y_j \{[f_k]_{k < j}\}(\sigma_1, \dots, \sigma_n) := Y_j(\langle \sigma_1, \dots, \sigma_n \rangle^\dagger; f_1, \dots, \langle \sigma_1, \dots, \sigma_n \rangle^\dagger; f_{j-1})$.

Here, $\langle \sigma_1, \dots, \sigma_n \rangle^\dagger; f_j$ is defined as the usual composition of (winning) strategies on $\odot X_1 \& \dots \& \odot X_n$ and $\odot X_1 \Rightarrow \dots \Rightarrow \odot X_n \Rightarrow \odot Y_j$.

The identities are defined as lists of derelicted copycats. Let us define a strategy $\text{der}_{[X_j]_j, X_i}$ which plays the derelicted copycat on all of $\odot X_i$: $\text{der}_{[X_j]_j, X_i} := \{s \in P_{\text{O-sat}(\Pi_{X_1} \dots \Pi_{X_n} X_i)} \mid \exists k s \upharpoonright_{X_i} \downarrow_k \approx_{\odot X_i} s \upharpoonright_{X_i}\}$. We then define $\text{id}_{[X_i]_i} := [\text{der}_{[X_j]_j, X_i}]_i$ and $\mathbf{p}_{[X_i]_i, [Y_j]_j} := [\text{der}_{[X_i]_i, [Y_j]_j, X_k}]_k$. Let us define

$\text{Tm}([X_i]_{i \leq n}, [Y_j]_{j \leq m}) := \{[f_j]_j \mid [X_i]_i \xrightarrow{[\text{der}_{[X_i]_i, X_1}, \dots, \text{der}_{[X_i]_i, X_n}, f_1, \dots, f_m]} [X_i]_i.[Y_j]_j\}$. Then, we can define $\mathbf{v}_{[X_i]_i, [Y_j]_j} := [\text{der}_{[X_i]_i, [Y_j]_j, Y_k}]_k$. Note that these are well-defined because of the following claim.

Claim. $\text{der}_{[X_j]_j, X_i} \in \text{wstr}(\Pi_{X_1} \dots \Pi_{X_n} X_i \{[\text{der}_{[X_j]_j, X_k}]_{k \leq i-1}\})$.

We define $\langle [f_j]_{j \leq m}, [g_k]_{k \leq l} \rangle := [f_1, \dots, f_m, g_1, \dots, g_l]$. We inductively define the composition of $[X_i]_{i \leq n} \xrightarrow{[f_j]_j} [Y_j]_{j \leq m} \xrightarrow{[g_k]_k} [Z_k]_k$ in $\mathbf{Ctx}(\mathbf{DGame}_!)$ by

$$[f_j]_j; [g_k]_k := [\langle f_1, \dots, f_m \rangle^\dagger; g_k]_k,$$

using the usual (co-Kleisli) composition of (winning) strategies on $\odot X_1 \Rightarrow \dots \Rightarrow \odot X_n \Rightarrow (\odot Y_1 \& \dots \& \odot Y_m)$ and $\odot Y_1 \Rightarrow \dots \Rightarrow \odot Y_m \Rightarrow \odot Z_k$. We note that we can assign to this composition a more precise dependent function type.

Claim. $[f_j]_j; [g_k]_k$ is a list of winning strategies if $[g_k]_k$ and $[f_j]_j$ are.

Finally, for $[X_i]_i \xrightarrow{[f_j]_j} [Y_j]_j$ and $[g_k]_k \in \mathbf{Tm}([Y_j]_j, [Z_k]_k)$, we can define

$$[g_k]_k \{ [f_j]_j \} := [\langle f_1, \dots, f_m \rangle^\dagger; g_k]_k.$$

Remark 1. Note that, in $\mathbf{Ctx}(\mathbf{DGame}_!)$, $[A, B] \cong [A \& B]$ if A and B are games (without mutual dependency) and $[] \cong [I]$.

5 Semantic Type Formers

We show that our \mathbf{CwF} supports 1-, Σ -, Π -, and \mathbf{Id} -types. We characterise some properties of the \mathbf{Id} -types, marking their place in the intensionality spectrum.

Theorem 6. *Our \mathbf{CwF} supports 1-, Σ - and Π -type with their β - and η -rules.*

The 1-type is interpreted by the empty context $[]$ and Σ -types are just interpreted by concatenation of lists. We define a Σ -type $\Sigma_{[Y_j]_j} [Z_k]_k \in \mathbf{Ty}([X_i]_{i \leq n})$ as $[Y_j]_j \cdot [Z_k]_k$ for $[Z_k]_{k \leq l} \in \mathbf{Ty}([X_i]_{i \leq n} \cdot [Y_j]_{j \leq m})$.

We have already seen Π -types $\Pi_{[X_i]_{i \leq n}} [Y] := [\Pi_{I_{X_1}} \dots \Pi_{I_{X_n}} Y]$ of dependent games. What remains to be defined are Π -types $\Pi_{[X_i]_i} [Y_j]_j$ of general dependent context games, which can now be reduced to the former, as we have that $\Sigma_{f: \Pi_{x:A} B} \Pi_{x:A} C[f(x)/y]$ satisfies the rules for $\Pi_{x:A} \Sigma_{y:B} C$.

Corollary 1. *Note that this means that $\mathbf{Ctx}(\mathbf{DGame}_!)$ is in particular a ccc.*

We turn to identity types next, which are essentially defined as those of the domain semantics of DTT [19]. Interestingly, due to the more intensional nature of game semantics, they acquire a more intensional character, refuting \mathbf{FunExt} .

For $[Y_j]_j \in \mathbf{Ty}([X_i]_i)$, define $\mathbf{Id}_{[Y_j]_j} \in \mathbf{Ty}([X_i]_i \cdot [Y_j]_j \cdot [Y_{j'}]_{j'})$ through the intersection of subgames³ of $\odot Y_j$ for $1 \leq j \leq m$:

$$\mathbf{Id}_{[Y_j]_j}([\sigma_i]_i, [\tau_j]_j, [\tau'_j]_j) := [\mathbf{Id}_{Y_j}]_j([\sigma_i]_i, [\tau_j]_j, [\tau'_j]_j) := [\tau_j \cap \tau'_j]_j.$$

Here, $\odot \mathbf{Id}_{Y_j} := \odot Y_j$.

Theorem 7. *This definition satisfies the I-, E- and β -rules for \mathbf{Id} -types.*

³ Here, we identify a strategy σ on X with the subgame $\sigma \cup \{sa \in P_X \mid s \in \sigma\} \trianglelefteq X$.

For **ld-I**, $x : A \vdash \text{refl}_t : \text{ld}_B(t, t)$ can be interpreted as the list of strategies $\llbracket t \rrbracket$ but at $\Pi_{\llbracket A \rrbracket} \text{ld}_{\llbracket B \rrbracket}(\llbracket t \rrbracket, \llbracket t \rrbracket) \leq \Pi_{\llbracket A \rrbracket} \llbracket B \rrbracket$, where we write $\llbracket - \rrbracket$ for the interpretation of DTT in our model. We interpret **ld-E** by sending $H \in \text{Tm}(\llbracket A \rrbracket, \llbracket B \rrbracket, \llbracket C \rrbracket \{ \text{id}_{\llbracket A \rrbracket, \llbracket B \rrbracket}, \mathbf{v}_{\llbracket A \rrbracket, \llbracket B \rrbracket}, \text{refl}_{\llbracket B \rrbracket} \})$ to $J(H) \in \text{Tm}(\llbracket A \rrbracket, \llbracket B \rrbracket, \llbracket B \rrbracket, \llbracket \text{ld}_B \rrbracket, \llbracket C \rrbracket)$ by playing H between $\llbracket A \rrbracket$, $\llbracket \text{ld}_B \rrbracket$ and $\llbracket C \rrbracket$ rather than $\llbracket A \rrbracket$, $\llbracket B \rrbracket$ and $\llbracket C \rrbracket$.

In addition to being non-extensional (i.e. refuting the principle of equality reflection), these identity types can be said to be intensional in a positive sense.

Theorem 8. *Streicher’s Criteria of Intensionality [16] are satisfied, i.e.*

- (I1) *there exist $\vdash A$ type such that $x, y : A, z : \text{ld}_A(x, y) \not\vdash x \equiv y : A$;*
- (I2) *there exist $\vdash A$ type and $x : A \vdash B$ type such that $x, y : A, z : \text{ld}_A(x, y) \not\vdash B \equiv B[y/x]$ type;*
- (I3) *for all $\vdash A$ type, $\vdash p : \text{ld}_A(t, s)$ implies $\vdash t \equiv s : A$.*

(I1) relies on the interpretation of terms carrying intensionality. For instance, we can take $\llbracket A \rrbracket = \widetilde{\mathbb{B}_*}$, where $\mathbb{B} := \{\text{tt}, \text{ff}\}$, and evaluate the first and second projections on $\llbracket x \rrbracket = \llbracket z \rrbracket = \perp$ and $\llbracket y \rrbracket = \text{tt}$. (I2) relies on semantic types having intensional features. We can use $\llbracket B \rrbracket := (\perp, \text{ff} \mapsto I, \text{tt} \mapsto \widetilde{\mathbb{B}_*})$ on the data of (I1). (I3) follows as $[p_i]_i \in \text{Ctxt}(\mathbf{DGame!})(\llbracket \cdot \rrbracket, \text{ld}_{\llbracket X_i \rrbracket}([f_i]_i, [g_i]_i) := [f_i \cap g_i]_i)$ implies that $p_i = f_i = g_i$ for all i , as winning strategies are maximal.

The proofs of (I1) and (I2) also work for the domain model of DTT. (I3) relies on a crucial difference between the domain and games models: winning strategies are maximal, while to account for function types of domains with totality, we cannot assume that total domain elements are maximal. For similar reasons, **FunExt** is seen to fail in the games model: note that for strict and non-strict constantly **tt** functions f and g , we have $[f] \in \text{Tm}(\widetilde{\mathbb{B}_*}, \text{ld}_{\widetilde{\mathbb{B}_*}}([f], [g]))$, while $\text{Tm}(\llbracket \cdot \rrbracket, \text{ld}_{\Pi_{\widetilde{\mathbb{B}_*}} \widetilde{\mathbb{B}_*}}([f], [g])) = \emptyset$.

Theorem 9. ***FunExt** is refuted: for $\vdash f, g : \Pi_{x:A} B$, we do not generally have $z : \Pi_{x:A} \text{ld}_B(f(x), g(x)) \vdash \text{FunExt}_{f,g} : \text{ld}_{\Pi_{x:A} B}(f, g)$.*

On the other hand, it turns out that we have the principle of uniqueness of identity proofs **UIP**, by playing derelicted copycats between $\llbracket \text{ld}_A \rrbracket$ and $\llbracket \text{ld}_{\text{ld}_A} \rrbracket$.

Theorem 10. *We have $x, y : A, p, q : \text{ld}_A(x, y) \vdash \text{UIP}_A : \text{ld}_{\text{ld}_A(x, y)}(p, q)$.*

6 Ground Types and Completeness Results

We illustrate how our model of dependent games and winning strategies satisfies a completeness result with respect to the syntax of DTT with **1**-, Σ -, Π - and **ld**-types and finite inductive type families. The precise variant of DTT that these completeness results refer to can be found in the long version of this paper.

We describe a scheme for inductively defining finite type families. Let A be a type not containing any Π -constructors. Then, we specify a finite inductive definition of a type family $x : A \vdash B$ type by specifying finitely many closed terms $a_1, \dots, a_n : A$ and distinct symbols b_{ij} , $1 \leq i \leq n$, $1 \leq j \leq m_i$. The idea is that B is a type family, such that $B[a_i/x]$ contains precisely the distinct closed

terms $b_{i,1}, \dots, b_{i,m_i}$. These type families are more limited than general inductive definitions as they are freely generated by (finitely many) **closed** terms, while one would allow open terms in the general case. This means that we precisely get the inductive type families that have finitely many non-empty fibres which are all finite types. The prototypical example of such a type family is a calendar in which the type of days depends on the month and year we are in.

We interpret such a definition as specifying I - and E -rules for B :

$$\frac{}{\vdash b_{i,j} : B[a_i/x]} B\text{-}I_{i,j} \quad \frac{x : A, y : B \vdash C \text{ type}}{\vdash \text{case}_B : \prod_{x:A, y:B, z_{11}:C[a_1/x, b_{1,1}/y], \dots, z_{nm_n}:C[a_n/x, b_{n,m_n}/y]} C \text{ } B\text{-}E,$$

together with the β - and η -rules, commutative conversions and a rule⁴ defining a **ex falso** eliminator from $\text{ld}_B(b_{i,j}, b_{i',j'})$ for distinct constructors $b_{i,j}, b_{i',j'}$ of B .

Let $\text{Ctxt}(\mathbf{DGame}_!)\text{fin1}\Sigma\Pi\text{Id}$ be the full subcategory of $\text{Ctxt}(\mathbf{DGame}_!)$ on the hierarchy generated by 1 -, Σ -, Π -, and Id -types and finite dependent games (and substitution), as below. Then we have the following results.

Theorem 11 (Finite Dependent Games). *Finite inductive type families B in context $x : A$, where $B[a_i/x]$ is generated by $\{\widetilde{b_{i,j} \mid j}\}$, have a sound interpretation in $\text{Ctxt}(\mathbf{DGame}_!)\text{fin1}\Sigma\Pi\text{Id}$: $\llbracket B \rrbracket : \llbracket a_i \rrbracket \mapsto [\{\widetilde{b_{i,j} \mid j}\}_*]$, else $\mapsto [\emptyset_*]$.*

Theorem 12 (Id-free Full and Faithful Completeness). *All morphisms in $\text{Ctxt}(\mathbf{DGame}_!)(\llbracket A \rrbracket, \llbracket B \rrbracket)$ if A and B do not contain Id -constructors are faithfully definable in DTT.*

As our interpretation factors faithfully over that of a total finitary PCF, faithfulness follows from (a variation on) the corresponding result for PCF [6]. Definability is proved along the lines of the template of [23] and hinges on the decomposition lemma for PCF-games.

Although the completeness properties of the model at the hierarchy with Id -types remain to be studied in detail, we do have the following.

Theorem 13 (Full and Faithful Completeness for strictly positive Id-types). *All morphisms in $\text{Ctxt}(\mathbf{DGame}_!)(\llbracket \cdot \rrbracket, \llbracket \Pi_{x:A} \text{Id}_B(f, g) \rrbracket)$ for $x : A \vdash f, g : B$ are faithfully definable in DTT, if $\vdash A \text{ type}$ and $x : A \vdash B \text{ type}$ are types built without Id -constructors.*

This completeness result for Id -types relies on the faithfulness of the interpretation of DTT in our model.

7 Future Work

Ultimately, the main goal is a thorough intensional, computational analysis of HoTT [3]. Obvious concrete directions for future work are the following:

- breaking UIP, by considering higher dimensional ground types;
- examining the phenomena of function extensionality and univalence;
- study of universes and a more intensional notion of type family;

⁴ Note that this rule is derivable in presence of a universe.

- study of (higher) inductive type families and their definability results;
- establishing completeness results for the full type hierarchy with `Id`-types;
- constructing models of DTT with side effects;
- synthesising strategies from a dependently typed specification;
- study of a possible embedding of the model in the co-Eilenberg-Moore category **Game**¹, which might simplify its presentation.

Acknowledgements

Samson Abramsky was supported by the EPSRC, AFOSR and the John Templeton Foundation. Radha Jagadeesan acknowledges support from the NSF. Matthijs Vákár was supported by the EPSRC and the Clarendon Fund.

References

1. Altenkirch, T., McBride, C., McKinna, J.: Why dependent types matter. Manuscript, available at <http://www.cs.nott.ac.uk/~txa/publ/ydtm.pdf> (2005) 235
2. Awodey, S., Warren, M.A.: Homotopy theoretic models of identity types. *Mathematical Proceedings of the Cambridge Philosophical Society* **146**(01) (2009) 45–55
3. HoTTbaki, U.: Homotopy Type Theory: Univalent Foundations of Mathematics. <http://homotopytypetheory.org/book>, Institute for Advanced Study (2013)
4. Abramsky, S., Jagadeesan, R.: Games and full completeness for multiplicative linear logic. *The Journal of Symbolic Logic* **59**(02) (1994) 543–574
5. Hyland, J.M.E., Ong, C.H.: On full abstraction for PCF: I, II, and III. *Information and computation* **163**(2) (2000) 285–408
6. Abramsky, S., Jagadeesan, R., Malacaria, P.: Full abstraction for PCF. *Information and Computation* **163**(2) (2000) 409–470
7. Abramsky, S., Jagadeesan, R.: A game semantics for generic polymorphism. *Annals of Pure and Applied Logic* **133**(1) (2005) 3–37
8. Laird, J.: Full abstraction for functional languages with control. In: *Logic in Computer Science, 1997. LICS’97. Proceedings., 12th Annual IEEE Symposium on, IEEE* (1997) 58–67
9. Abramsky, S., McCusker, G.: Linearity, sharing and state: a fully abstract game semantics for idealized algol with active expressions. *Electronic Notes in Theoretical Computer Science* **3** (1996) 2–14
10. Abramsky, S., Honda, K., McCusker, G.: A fully abstract game semantics for general references. In: *Logic in Computer Science, 1998. Proceedings. Thirteenth Annual IEEE Symposium on, IEEE* (1998) 334–344
11. Murawski, A.S., Tzevelekos, N.: Game semantics for good general references. In: *Logic in Computer Science (LICS), 2011 26th Annual IEEE Symposium on, IEEE* (2011) 75–84
12. Harmer, R., McCusker, G.: A fully abstract game semantics for finite nondeterminism. In: *Logic in Computer Science, 1999. Proceedings. 14th Symposium on, IEEE* (1999) 422–430
13. Danos, V., Harmer, R.S.: Probabilistic game semantics. *ACM Transactions on Computational Logic (TOCL)* **3**(3) (2002) 359–382
14. Abramsky, S., Ghica, D.R., Murawski, A.S., Ong, C.H., Stark, I.D.: Nominal games and full abstraction for the nu-calculus. In: *Logic in Computer Science, 2004. Proceedings of the 19th Annual IEEE Symposium on, IEEE* (2004) 150–159

15. Abramsky, S., McCusker, G.: Call-by-value games. In Nielsen, M., Thomas, W., eds.: *Computer Science Logic*. Volume 1414 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (1998) 1–17
16. Streicher, T.: *Investigations into intensional type theory* (1993) <http://www.mathematik.tu-darmstadt.de/streicher/HabilStreicher.pdf>.
17. Hofmann, M.: *Extensional Constructs in Intensional Type Theory*. Springer (1997)
18. Bezem, M., Coquand, T., Huber, S.: A model of type theory in cubical sets. In: *19th International Conference on Types for Proofs and Programs (TYPES 2013)*. Volume 26. (2014) 107–128
19. Palmgren, E., Stoltenberg-Hansen, V.: Domain interpretations of Martin-Löf’s partial type theory. *Annals of Pure and Applied Logic* **48**(2) (1990) 135–196
20. Abramsky, S., Jagadeesan, R.: Game semantics for access control. *Electronic Notes in Theoretical Computer Science* **249** (2009) 135–156
21. Vákár, M.: A categorical semantics for linear logical frameworks. To appear in the proceedings of *FoSSaCS 2015* (2015) Online version at <http://arxiv.org/abs/1501.05016> .
22. Pitts, A.M.: Categorical logic. In Abramsky, S., Gabbay, D., Maibaum, T., eds.: *Handbook of Logic in Computer Science*, Volume 5. OUP (2000) 39–128
23. Abramsky, S.: Axioms for definability and full completeness. In: *Proof, Language and Interaction: Essays in Honour of Robin*. MIT Press (2000) 55–75