

Game Semantics for Dependent Types

Samson Abramsky, Radha Jagadeesan and Matthijs Vákár

Cork, 28 August, 2015

Game theoretic model of dependent type theory (DTT):

- refines model in domains and (total) continuous functions;
- call-by-name evaluation;
- faithful model of (total) DTT with Σ -, Π -, Id-types and finite inductive type families;
- fully complete if Id-types limited;

Game theoretic model of dependent type theory (DTT):

- refines model in domains and (total) continuous functions;
- call-by-name evaluation;
- faithful model of (total) DTT with Σ -, Π -, Id-types and finite inductive type families;
- fully complete if Id-types limited;
- Id-types more intensional than domain model: function extensionality fails;
- intensional in orthogonal way to HoTT (time vs space): UIP holds.

Game Semantics?

- Interpolates between operational and denotational semantics:
very intensional with structural clarity of categorical model;
- Unified framework for intensional, computational semantics:
 - PCF (HO, N, **AJM**);

Game Semantics?

- Interpolates between operational and denotational semantics: very intensional with structural clarity of categorical model;
- Unified framework for intensional, computational semantics:
 - PCF (HO, N, **AJM**);
 - references, non-local control, dynamically generated local names, probability, non-determinism, concurrency...;
 - various evaluation strategies;
 - recursive types, polymorphism;
 - propositional logic, impredicative 2^{nd} order quantification, external 1^{st} order quantification;

Game Semantics?

- Interpolates between operational and denotational semantics: very intensional with structural clarity of categorical model;
- Unified framework for intensional, computational semantics:
 - PCF (HO, N, **AJM**);
 - references, non-local control, dynamically generated local names, probability, non-determinism, concurrency...;
 - various evaluation strategies;
 - recursive types, polymorphism;
 - propositional logic, impredicative 2^{nd} order quantification, external 1^{st} order quantification;
 - **internal 1^{st} order quantification / dependent types surprisingly absent (and surprisingly hard!);**

Game Semantics?

- Interpolates between operational and denotational semantics: very intensional with structural clarity of categorical model;
- Unified framework for intensional, computational semantics:
 - PCF (HO, N, **AJM**);
 - references, non-local control, dynamically generated local names, probability, non-determinism, concurrency...;
 - various evaluation strategies;
 - recursive types, polymorphism;
 - propositional logic, impredicative 2^{nd} order quantification, external 1^{st} order quantification;
 - **internal 1^{st} order quantification / dependent types surprisingly absent (and surprisingly hard!)**;
- Tight correspondence with syntax (full abstraction, full faithful completeness): often unique semantics in this respect.

Computation / Logic	Games
data type / proposition	2-player game (duality!)
computational process / argument	play: alternating seq. of moves
program / proof	Player (P) strategy
environment / refutation	Opponent (O) strategy
variable declarations / axiom links	copycat strategies
substitution / cut	interaction
termination / correctness	winning
pure sequential functional behaviour (no state, parallelism, control ops) /..	history-freeness + determinism + well-bracketing of strategies

An example:

Player: $x : \mathbb{B}, y : (\mathbb{B} \Rightarrow \mathbb{B}) \vdash y(x) : \mathbb{B}$

Opponent: $- [\text{tt}/x, (\lambda_{z:\mathbb{B}} \neg z)/y]$

$\mathbb{B} \Rightarrow (\mathbb{B} \Rightarrow \mathbb{B}) \Rightarrow \mathbb{B}$	
	*
	*
	*
*	*
tt	*
	tt
	ff
	ff
	O
	P
	O
	P
	O
	P
	O
	P

An example:

Player: $x : \mathbb{B}, y : (\mathbb{B} \Rightarrow \mathbb{B}) \vdash y(x) : \mathbb{B}$
Opponent: $- [(\lambda_{z:\mathbb{B}} \text{ff})/y]$

$\mathbb{B} \Rightarrow (\mathbb{B} \Rightarrow \mathbb{B}) \Rightarrow \mathbb{B}$			
		*	<i>O</i>
	*		<i>P</i>
	ff		<i>O</i>
		ff	<i>P</i>

history-free!

Games and winning history-free strategies form a smcc **Game**:

- I : the game with one play of length 0;
- $A \otimes B$: playing A and B simultaneously, where only Opponent can switch games;
- $A \multimap B$: $\text{swap}_{O,P}(A)$ and B simultaneously, Player switches.

Games and winning history-free strategies form a smcc **Game**:

- I : the game with one play of length 0;
- $A \otimes B$: playing A and B simultaneously, where only Opponent can switch games;
- $A \multimap B$: $\text{swap}_{O,P}(A)$ and B simultaneously, Player switches.

Also model simple type theory (STT): have a ccc **Game**_!:

- $A \Rightarrow B := !A \multimap B$;
- $!A$: playing ω equivalent copies of A simultaneously, where only Opponent can switch games;
- Product $A \& B$: Opponent chooses to play A or B (unit: I).

Games and winning history-free strategies form a smcc **Game**:

- I : the game with one play of length 0;
- $A \otimes B$: playing A and B simultaneously, where only Opponent can switch games;
- $A \multimap B$: $\text{swap}_{O,P}(A)$ and B simultaneously, Player switches.

Also model simple type theory (STT): have a ccc **Game**_!:

- $A \Rightarrow B := !A \multimap B$;
- $!A$: playing ω equivalent copies of A simultaneously, where only Opponent can switch games;
- Product $A \& B$: Opponent chooses to play A or B (unit: I).

Ground types (finite inductive types): for a set X , game \tilde{X}_* with one Opponent move $*$, followed by any of the Player moves $x \in X$.

Use the term simple type theory (STT) to refer to a simple λ -calculus with binary products \times , function types \Rightarrow and finite inductive types $\{a_i \mid 1 \leq i \leq n\}$, or a total finitary PCF with binary products, with $\beta\eta$ -rules and PCF commutative conversions for case-constructs.

Straightforward consequence of AJM:

Theorem

*The interpretation of STT in $\mathbf{Game}_!$ is **fully and faithfully complete**.*

Dependent type theory (DTT)?

What is it?

- Curry-Howard for predicate logic: types with free (term) variables, constructions Σ, Π, Id on types.
- Judgements: $\vdash \Gamma \text{ ctxt}$, $\Gamma \vdash A \text{ type}$, $\Gamma \vdash a : A$, equations.
- Order in context matters!
- No clean separation syntax types and terms.

Dependent type theory (DTT)?

What is it?

- Curry-Howard for predicate logic: types with free (term) variables, constructions Σ, Π, Id on types.
- Judgements: $\vdash \Gamma \text{ ctxt}, \Gamma \vdash A \text{ type}, \Gamma \vdash a : A$, equations.
- Order in context matters!
- No clean separation syntax types and terms.

Why care?

- Move towards richer type systems: e.g. GADTs in Haskell.
- Types allowed to refer to data: e.g. $n : \mathbb{N} \vdash \text{List}(n)$ type.
- Specification by typing: certification by type checking.
- Proof assistants.
- Logical Frameworks.

A Faithful Translation to Simple Type Theory (STT)

Idea: DTT talks about same algorithms as STT but can assign them a more precise type/specification.

Formally: have translation of DTT into STT. Let DTT inherit the equational theory of STT to make translation faithful.

A Faithful Translation to Simple Type Theory (STT)

Idea: DTT talks about same algorithms as STT but can assign them a more precise type/specification.

Formally: have translation of DTT into STT. Let DTT inherit the equational theory of STT to make translation faithful.

$x : A \vdash (a_i \mapsto_i \{b_{i,j} \mid j\}) \text{ type} \quad \mapsto \quad \vdash \{b_{i,j} \mid i, j\} \text{ type}$

$x : A \vdash \Sigma_{y:B} C \text{ type} \quad \mapsto \quad \vdash B^T \times C^T \text{ type}$

$x : A \vdash \Pi_{y:B} C \text{ type} \quad \mapsto \quad \vdash B^T \Rightarrow C^T \text{ type}$

$x : A, y : B, y' : B \vdash \text{Id}_B(y, y') \text{ type} \quad \mapsto \quad \vdash B^T \text{ type}$

$x' : A' \vdash B[a/x] \text{ type} \quad \mapsto \quad \vdash B^T \text{ type}$

+ translation on terms.

The idea of our interpretation $\llbracket \cdot \rrbracket$ will be to construct a category **CtxtGame**_I of *dependent context games* with a functor $\odot(-)$ to **Game**_I, such that

$$\begin{array}{ccc}
 \text{Syntax}_{DTT} & \xrightarrow{\llbracket - \rrbracket} & \mathbf{CtxtGame}_I \\
 \downarrow (-)^T & & \downarrow \odot(-) \\
 \text{Syntax}_{STT} & \xrightarrow{\llbracket - \rrbracket} & \mathbf{Game}_I.
 \end{array}$$

Games model of DTT will therefore automatically be faithful.

Dependent Games and Strategies

Game $B \in \text{ob}(\mathbf{DGame}_I(A))$ with dependency on A :

- game $\odot(B) \in \text{ob}(\mathbf{Game}_I)$ (without dependency);
- continuous function $\text{str}(A) \xrightarrow{B} \text{Sub}(\odot(B))$ from strategies on A to subgames (\lesssim -closed subsets of plays) of $\odot(B)$.

Dependent Games and Strategies

Game $B \in \text{ob}(\mathbf{DGame}_!(A))$ with dependency on A :

- game $\odot(B) \in \text{ob}(\mathbf{Game}_!)$ (without dependency);
- continuous function $\text{str}(A) \xrightarrow{B} \text{Sub}(\odot(B))$ from strategies on A to subgames (\lesssim -closed subsets of plays) of $\odot(B)$.

Note: $\text{ob}(\mathbf{Game}_!) \subsetneq \text{ob}(\mathbf{DGame}_!(I))$. Indeed, a dependent game A in empty context is a pair $A(\perp) \trianglelefteq \odot(A)$, for empty strategy \perp .

Define $\text{ob}(\mathbf{DGame}_!(A)) := \text{ob}(\mathbf{DGame}_!(\odot(A)))$.

Can define I , $\&$ and \Rightarrow on games with dependency and make into ccc with homset $\mathbf{DGame}_!(A)(B, C) := \text{wstr}(\Pi_A(B \Rightarrow C))$.

We define a game $\Pi_A B \trianglelefteq \odot(A) \Rightarrow \odot(B)$ of dependent functions.

- Idea: the choice of a fibre $B[a/x]$ for the output of a dependent function $f : \Pi_A B$ is entirely the responsibility of the context that provides the argument a .

We define a game $\Pi_A B \trianglelefteq \odot(A) \Rightarrow \odot(B)$ of dependent functions.

- Idea: the choice of a fibre $B[a/x]$ for the output of a dependent function $f : \Pi_A B$ is entirely the responsibility of the context that provides the argument a .
- Opponent can determine fibre $B(\tau)$ of B :
 - explicitly, revealing winning history-free strategy τ on $A(\perp)$, by playing in $!\odot(A)$;
 - implicitly, by playing in $\odot(B)$;
- Player has to stay within $B(\tau)$ for all τ consistent with Opponent's behaviour.

We define a game $\Pi_A B \trianglelefteq \odot(A) \Rightarrow \odot(B)$ of dependent functions.

- Idea: the choice of a fibre $B[a/x]$ for the output of a dependent function $f : \Pi_A B$ is entirely the responsibility of the context that provides the argument a .
- Opponent can determine fibre $B(\tau)$ of B :
 - explicitly, revealing winning history-free strategy τ on $A(\perp)$, by playing in $!\odot(A)$;
 - implicitly, by playing in $\odot(B)$;
- Player has to stay within $B(\tau)$ for all τ consistent with Opponent's behaviour.
- That is, as long as there is such a τ ; otherwise, anything goes.
- Indeed, Opponent is totally free and might not play along a winning strategy, as $\odot(-)$ should be faithful (to match $(-)^T$).

Non-example of dependently typed algorithm: scheduling finance meetings

Player/Academic: $x : \text{months} \vdash 31 : \text{days}(x)$

Opponent/Education Finance

Business Manager Manager:

$\Pi(\widetilde{\text{months}}_*, \widetilde{\text{days}}_*)$	
	*
	31
	O
	P

Player chooses fibre: e.g. February doesn't have a 31st day. Mr Manager shouldn't allow that...

Example of dependently typed algorithm:

Academic/Player: $x : \text{months} \vdash \text{case}_{\text{days, months}}(x, \{31, 1, \dots, 1, 31\}) : \text{days}(x)$

Opponent/Education Finance
 Business Manager Manager: $- [\text{January}/x]$

$\Pi(\widetilde{\text{months}}_*, \widetilde{\text{days}}_*)$	
	O
$*$	P
January	O
31	P

Example (fibre-wise identities):

Player: $x : A, y : B(x) \vdash y : B(x)$

Opponent: choose your favourite

$\prod([A], [B] \Rightarrow [B])$		
	b	O
b		P
b'		O
	b'	P
	\vdots	

Theorem

Have strict indexed ccc $\mathbf{DGame}_!(I)^{op} \xrightarrow{\mathbf{DGame}} \mathbf{CCCat}$ of dependent games:

- $\text{ob}(\mathbf{DGame}_!(A)) := \{\text{continuous str}(\odot(A)) \xrightarrow{B} \text{Sub}(\odot(B))\}$
- *hom-sets* $\mathbf{DGame}_!(A)(B, C) := \text{wstr}(\Pi_A(B \Rightarrow C))$
- *identities as in example;*
- *composition: usual AJM-composition;*
- *change of base: usual AJM-composition (works out!).*

As we don't have (additive) Σ -types, this is not a model of DTT!

Theorem

Have strict indexed ccc $\mathbf{DGame}_!(I)^{op} \xrightarrow{\mathbf{DGame}} \mathbf{CCCat}$ of dependent games:

- $\text{ob}(\mathbf{DGame}_!(A)) := \{\text{continuous str}(\odot(A)) \xrightarrow{B} \text{Sub}(\odot(B))\}$
- *hom-sets* $\mathbf{DGame}_!(A)(B, C) := \text{wstr}(\Pi_A(B \Rightarrow C))$
- *identities as in example;*
- *composition: usual AJM-composition;*
- *change of base: usual AJM-composition (works out!).*

As we don't have (additive) Σ -types, this is not a model of DTT!

Theorem

Formally add them: get model $\mathbf{CtxtGame}_!$ of DTT with $\Sigma\Pi\text{Id}$ (the last through intersection) and fin. inductive type families!

Non-Example:

Player: $x : \mathbb{B}, y : \mathbb{B} \vdash p : \text{Id}_{\mathbb{B}}(x, y)$

Opponent : $- [\text{tt}/x, \text{ff}/y]$

$\Pi(\tilde{\mathbb{B}}_*, \Pi(\tilde{\mathbb{B}}_*, \text{Id}_{\tilde{\mathbb{B}}_*}))$		
		*
*		*
tt		*
	*	*
	ff	*
		tt
		P

...as tt does not lie in intersection $\text{tt} \cap \text{ff}$.

Example:

Player: $x : \mathbb{B} \vdash \text{refl}_x : \text{Id}_{\mathbb{B}}(x, x)$

Opponent:

$\prod(\tilde{\mathbb{B}}_* , \text{Id}_{\tilde{\mathbb{B}}_*} \{ \text{diag}_{\tilde{\mathbb{B}}_*} \})$	
	*
*	
x	
	x
	O
	P
	O
	P

...as x lies in $x \cap x$.

Place in intensionality spectrum Id-types:

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

Discrete ground types (0-types), but function hierarchy generates (open) propositional identities: observational equivalences.

Completeness Results

Summarising, we have

Theorem (Soundness and Faithfulness)

We have a faithful model of DTT with Σ -, Π -, Id-types and finite inductive type families: faithful functor

$$\text{Syntax}_{DTT} \xrightarrow{\llbracket - \rrbracket} \mathbf{CtxGame}_!$$

Completeness Results

Summarising, we have

Theorem (Soundness and Faithfulness)

We have a faithful model of DTT with Σ -, Π -, Id-types and finite inductive type families: faithful functor

$$\text{Syntax}_{DTT} \xrightarrow{\llbracket - \rrbracket} \mathbf{CtxGame}_!$$

Actually, the model has strong completeness properties.

Theorem (Full Completeness)

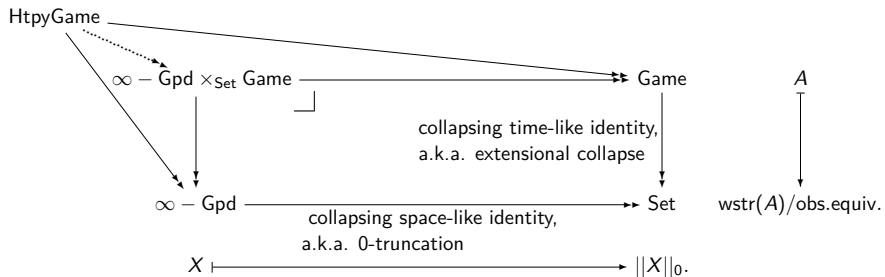
This interpretation is full when restricted to the types of the form A or $\Pi_A \text{Id}_B(f, g)$ with A and B built without Id-types:

$$\text{Syntax}_{DTT}^{\text{Restricted Types}} \xrightarrow{\llbracket - \rrbracket} \mathbf{CtxGame}_! \quad \text{full.}$$

Ultimate goal: intensional, computational analysis of HoTT.

- game semantics of higher inductive types / quotient types;
- examining function extensionality and univalence;
- universes and a more intensional notion of type family;
- infinite inductive type families and their definability results;
- examining completeness properties of the model for the complete type hierarchy, including Id-types;
- constructing models of DTT with side effects.

In particular, for the first item:



Bonus Example (Higher order dependent functions):

Player/Employer: $x : \text{years}, y : \Pi(x : \text{days}(\text{year}), \Pi(y : \text{holidays}, \mathbb{B})) \vdash \text{approval check1} : \mathbb{B}$
 Opponent/Employee: $- [2015/x, \text{holiday plans}/y]$

$\Pi(\widetilde{\text{years}}_*,$	$\Pi(\widetilde{\text{days}}_*,$	$\Pi(\widetilde{\text{holidays}}_*,$	$\widetilde{\mathbb{B}}_*))$,	$\widetilde{\mathbb{B}}_*$	
				*	O
				*	P
		*			O
		Holi			P
	*				O
*					P
2015					O
	65				P
			tt		O
				tt	P

(Note that Holi happens every year with variable Gregorian date and that Player gets to choose the day so can even specify the holiday before the day.)

Bonus Example (Higher order dependent functions):

Player/Employer: $x : \text{years}, y : \Pi(x : \text{days}(\text{year}), \Pi(y : \text{holidays}, \mathbb{B})) \vdash \text{approval check2} : \mathbb{B}$
 Opponent/Employee: $\neg[2015/x, \text{holiday plans}/y]$

$\Pi(\widetilde{\text{years}}_*, \Pi(\widetilde{\text{days}}_*, \Pi(\widetilde{\text{holidays}}_*,$	$\widetilde{\mathbb{B}}_*), \widetilde{\mathbb{B}}_*)$	
	*	O
	*	P
	*	O
International Talk Like a Pirate Day		P
	tt	O
	ff	P

(Note that ITLPD happens every year on fixed Gregorian date: 19 September!)