A Game-Theoretic Framework For Dependent Types

Pierre Clairambault
PPS – Université Paris 7

GaLoP 2009

0. Introduction

Why dependent types ?

Several motives:

- Understand better dependent types,
- Gives an interactive view of dependent type constructions (Σ,Π,Id),
- A framework robust to variations of constraints (Control features, references)...
- and to new constructions (inductive types, universes...)
- Try to bridge a gap in the community...

I. Dependent Types

Basic Framework

Types can depend on terms. Seven kind of judgements.

- $\vdash \Gamma$ ctxt
- $\Gamma \vdash A$ type
- $\Gamma \vdash M : A$
- $\Gamma = \Delta$ ctxt
- $\Gamma \vdash A = B$ type
- $\Gamma \vdash M = N : A$
- $\delta : \Delta \to \Gamma$

With all the rules for reasoning with equality over terms, types, contexts, and the rules for context and susbtitutions formation.

Intensional Identity Types

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash N : A}{\Gamma \vdash Id_A(M, N)\ type}$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash refl_A(M) : Id_A(M, M)}$$

$$\frac{\Gamma, z : A \vdash H : B[z/x, z/y, refl_A(z)/p] \qquad \Gamma \vdash P : Id_A(M, N)}{R^{Id}(H, M, N, P) : B[M/x, N/y, P/p]}$$

Extensionality

The two following rules makes typechecking undecidable.

$$\frac{\Gamma \vdash P : Id_A(M, N)}{\Gamma \vdash M = N : A}$$

$$\frac{\Gamma \vdash P : Id_A(M, N)}{\Gamma \vdash P = refl_A(M)}$$

Proved independent by the **groupoid model** of type theory (Hofmann&Streicher)

Categorical models

A (non exhaustive) list of categorical models:

- Locally cartesian closed categories (Seely, 1984) : Extensional type theory. Coherence problem.
- Display categories (Taylor, 1986)
- D-categories (fibrations) (Ehrhard, 1988)
- Categories with attributes (Cartmell, 1978), categories with families (Dybjer, 1996): modular, closer to syntax

Categorical models

A (non exhaustive) list of categorical models:

- Locally cartesian closed categories (Seely, 1984) : Extensional type theory. Coherence problem.
- Display categories (Taylor, 1986)
- D-categories (fibrations) (Ehrhard, 1988)
- Categories with attributes (Cartmell, 1978), categories with families (Dybjer, 1996): modular, closer to syntax

Categories with families

A CwF is given by the following data:

- A base category $\mathbb{C}$ with a terminal object 1.
- A functor $T : \mathbb{C}^{op} \rightarrow \mathit{Fam}$ (associates to each context a family of **terms** indexed by **types**)

  $$M \in T(\Gamma)_A \text{ is denoted by } M : \Gamma \vdash A$$

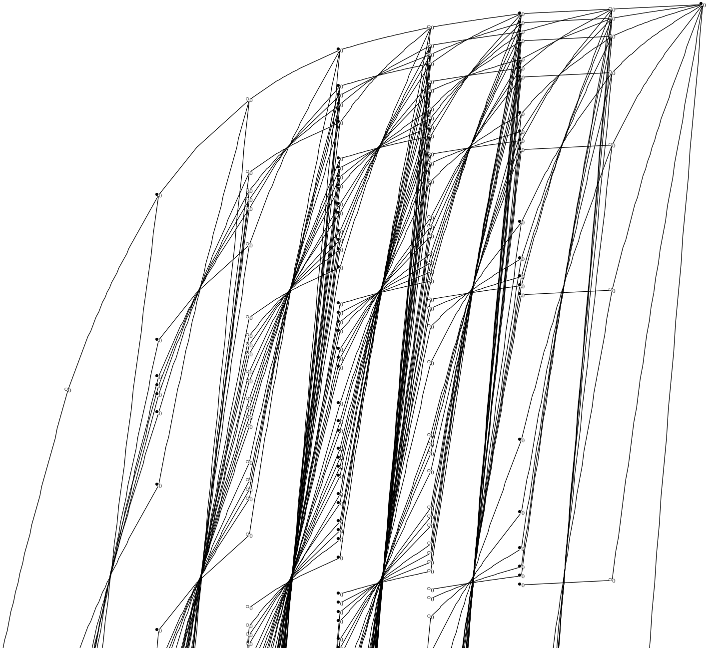
  The action of $T(\delta)$ (substitution) is denoted by $\_[\delta]$ on types and terms.
- A **context extension** operation. If $\Gamma \in \mathbb{C}$ and $A \in \mathit{Type}(\Gamma)$, $\Gamma \cdot A \in \mathbb{C}$, equiped with projections and pairing.

II. Dependent Games

Games and Totality

- A language for **proofs** is a **total** programming language,
- Hence, proofs are to be interpreted as **total** strategies,
- The class of total strategies is **not** closed under composition. . .

Interlude: the game-theoretic interaction of $\delta\delta$



Bounded total strategies

Definition

A strategy is **bounded** *when there is a bound on the size of its P-views.*

Theorem (Coquand, Clairambault&Harmer)

An interaction of bounded strategies is necessarily finite

Corollary

If $\sigma : A \Rightarrow B$ *and* $\tau : B \Rightarrow C$ *are total and bounded, so is* $\sigma ; \tau : A \Rightarrow C$*.*

Hence we get a CCC of arenas and total bounded strategies.

Dependent Games

Base idea: dependent games are usual games, but enriched with dependency information.

Definition

A **dependent game** *is a pair* (A, P_A)*, where:*

- *A is an arena,*
- *$P_A \subseteq \mathcal{L}_A$ is the set of* **valid plays**.

Dependent games will be the semantic counterpart of **contexts**.

Example

$[\![n : \texttt{nat}, l : \texttt{list}(n)]\!]$ is the pair (A, P_A) where:

- $A = \texttt{nat} \times \texttt{list}$
- P_A is the set of plays on A such that there is $n \in \texttt{nat}$:
  - $s_{\upharpoonright \texttt{nat}} \in [\![n]\!]$
  - $s_{\upharpoonright \texttt{list}} \in [\![\texttt{list}(n)]\!]$

Plays such as

$$
\begin{array}{ccc}
n : \texttt{nat} & \times & \texttt{list}(n) \\
q & & \\
1 & & \\
 & & q \\
 & & \texttt{Nil}
\end{array}
$$

are banned.

Valid strategies

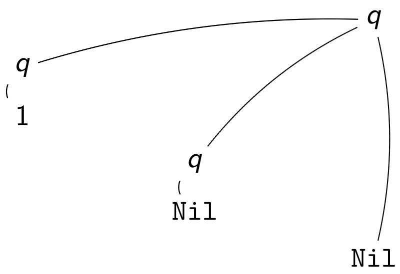
First (inaccurate) intuition:

Definition

σ is a valid strategy on A if $\sigma \subseteq P_A$.

Too strong: the following play should be accepted:

$$n : \mathtt{nat} \quad \vdash \quad \mathtt{list}(n) \Longrightarrow \mathtt{list}(n)$$



Valid strategies

Strategies are not forced to obey dependency if Opponent breaks it first.

Definition

A strategy is **valid** *on A if for any even-length $s \in \sigma$, if $sa \in P_A$, then there is $sab \in \sigma \cap P_A$.*

An analogous condition (skipped here) allows Player to break dependency if Opponent behaves non-innocently, *i.e.* uses side-effects in an obvious way.

Theorem

There is a cartesian closed category **Dep** *of dependent games and valid strategies.*

External dependency, 1: the informational preorder

- External dependencies will be modeled as **relations**
- These relations have to respect the **informational preorder**

Definition (Information)

Let $\sqsubseteq$ denote the prefix order on plays.

$$V(s) = \{\ulcorner s' \urcorner \mid s' \sqsubseteq s\}$$

$V(s)$ quantifies the **information on Player** contained in s.

Definition (Informational preorder)

$$s_1 \leq s_2 \Leftrightarrow V(s_1) \subseteq V(s_2)$$

$\leq$ also corresponds to $\sqsubseteq$ up to reordering of independent parts of the play.

Relations and external dependencies

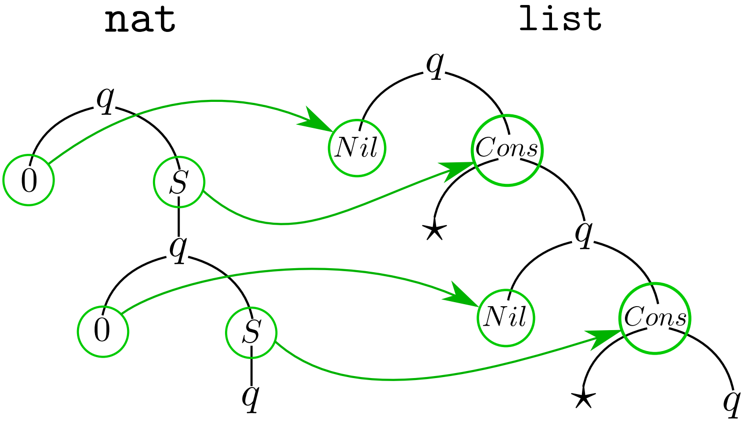
Definition

If $\Gamma \in \mathbf{Dep}$, *a game* **dependent over** Γ *will be a triple* $(A, P_A, \rhd_A)$ *where:*

- (A, P_A) *is a dependent game,*
- $\rhd_A \subseteq \mathcal{L}_\Gamma \times \mathcal{L}_A$, *satisfying*
- $\forall s \in \mathcal{L}_\Gamma,\ s \rhd_A \epsilon$
- $\forall s, s', t,\ s \rhd_A t \wedge s' \geq s \implies s' \rhd_A t$

The two last conditions are known as **monotonicity**. We denote by $Dep(\Gamma)$ the set of games dependent over Γ.

Paradigmatic example: `list`(n)



Substitution, 1: The relational functor

There is a functor

$$Rel : \mathbf{Dep} \to \mathbf{Rel}$$

- To any game A, Rel associates $\mathcal{L}_A$
- To any strategy $\sigma : A \Rightarrow B$, $Rel(\sigma) \subseteq \mathcal{L}_A \times \mathcal{L}_B$ is

$$\{(s_{\upharpoonright A}, s_{\upharpoonright_B}) \mid s \in \sigma\}$$

Substitution, 2: Composition and monotonic completion

Definition

If $A \in \mathbf{Dep}(\Gamma)$ *and* $\sigma : \Delta \Rightarrow \Gamma$, *then*

$$A[\delta] = (A, P_A, \overline{Rel(\delta)}; \rhd_A)$$

where $\overline{Rel(\delta)}$ *is the* **monotonic completion** *of* $Rel(\delta)$. *We check that* $A[\delta] \in Dep(\Delta)$

This construction is functorial, hence produces a functor

$$T : \mathbf{Dep}^{op} \longrightarrow Set$$

To get a Cwf, we still need terms and context comprehension.

Dependent game constructions

To build the Cwf structure, we will do the following:

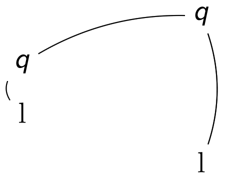
- For $A \in \mathit{Dep}(\Gamma)$, build a dependent game $\Gamma \vdash A$. Terms $\sigma \in \Gamma \vdash A$ will be strategies $\sigma : \Gamma \vdash A$.
- For $A \in \mathit{Dep}(\Gamma)$, build a dependent game $\Gamma \cdot A$, the **context extension**.

$\Gamma \vdash A$ and $\Gamma \cdot A$ are respectively special cases of Π-types and Σ-types.

Construction of $\Gamma \vdash A$

The base arena of $\Gamma \vdash A$ will be $\Gamma \Rightarrow A$. When is $s \in P_{\Gamma \vdash A}$?

$$n : \texttt{nat} \quad \cdot \quad \texttt{list}(n) \quad \vdash \quad \texttt{list}(n)$$



This play should be accepted, even if it is not in $\rhd_{\texttt{list}}$

Definition (Forcing)

$$s \Vdash_A t \Leftrightarrow \forall \alpha : \Gamma,\ s \in \alpha \implies \exists s' \in \alpha,\ s' \rhd_A t$$

Construction of $\Gamma \vdash A$

Definition

$\Gamma \vdash A = (\Gamma \Rightarrow A, P_{\Gamma \vdash A})$, *with*

$$P_{\Gamma \vdash A} = \{s \in P_{\Gamma \Rightarrow A} \mid s_{\restriction \Gamma} \Vdash_A s_{\restriction A}\}$$

Definition

Terms $\sigma \in \Gamma \vdash A$ *are simply valid strategies* $\sigma : \Gamma \vdash A$*. If* $\delta : \Delta \Rightarrow \Gamma$,

$$\sigma[\delta] = \delta; \sigma : \Gamma \vdash A[\delta]$$

With these definitions, the functor T extends to

$$T : \mathbf{Dep}^{op} \to \mathit{Fam}$$

Construction of $\Gamma \cdot A$

Let us look at some examples.

$$n : \texttt{nat} \cdot \texttt{list}(n)$$

$$q$$

$$0$$

$$q$$

$$\texttt{Nil}$$

must be naturally accepted, since it is in $\triangleright_{\texttt{list}}$.

Construction of $\Gamma \cdot A$

But if Opponent asks first right...

$$n : \texttt{nat} \cdot \texttt{list}(n)$$

q

Nil

q

1

We see that there is a retroaction from right to left, so the situation is not so simple.

Construction of $\Gamma \cdot A$

The appropriate definition is dual to forcing.

Definition (Coherence)

Let $\Gamma \in \mathbf{Dep}$*, and* $A \in Dep(\Gamma)$*. We set:*

$$s \frown_A t \Leftrightarrow \exists \alpha : \Gamma,\ s \in \alpha \wedge \exists s' \in \alpha,\ s' \triangleright_A t$$

Definition

$\Gamma \cdot A = (\Gamma \times A, P_{\Gamma \cdot A})$*, with*

$$\Gamma \cdot A = \{ s \in P_{\Gamma \times A} \mid s_{\restriction \Gamma} \frown_A s_{\restriction A} \}$$

Projections comes from the underlying cartesian product of $\Gamma \cdot A$, and all the required equations are satisfied. Hence $(\mathbf{Dep}, T)$ is a Cwf.

III. Intensional Identity Types

The basic idea

Let us consider $\sigma, \tau : A$. The type $Id_A(\sigma, \tau)$ will look as follows:

- Its base arena will be A
- $P_{Id_A(\sigma,\tau)}$ will be

$$P_{Id_A(\sigma,\tau)} = \{s \in P_A \mid s \in \sigma \ \wedge \ s \in \tau\}$$

Then, the existence of a total strategy $p : Id_A(\sigma, \tau)$ will be equivalent to $\sigma = \tau$.

Identity types

We define a game $Id_A \in Dep(\Gamma \cdot A_1 \cdot A_2[p])$ as follows:

- The base arena is A
- The set of valid plays is P_A
- We need a monotonic relation $\triangleright_{Id_A} \subseteq \mathcal{L}_{\Gamma \cdot A_1 \cdot A_2[p]} \times \mathcal{L}_A$:

$$s \triangleright_{Id_A} t \Leftrightarrow \begin{cases} t \leq s_{\upharpoonright A_1} \\ t \leq s_{\upharpoonright A_2} \end{cases}$$

Which satisfies the required properties.

Reflexivity

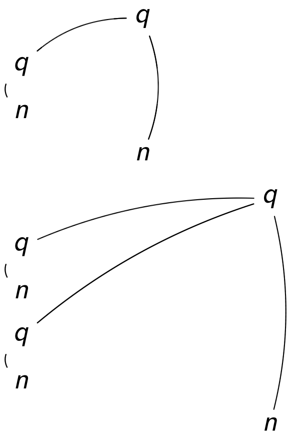
We need a strategy $refl_A : \Gamma \cdot A \vdash Id_A[\langle id, q \rangle]$.

- i.e. a strategy $refl_A : \Gamma \times A \Rightarrow A$, satisfying additional conditions.
- We define $refl_A$ as the copycat $\pi_2 : \Gamma \times A \rightarrow A$

$refl_A$ satisfies the required conditions, and is stable under substitution.

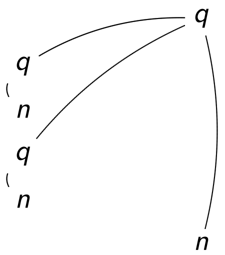
The model refutes extensionality

$$n : \mathtt{nat} \vdash \Sigma_{m_1:\mathtt{nat}}. \quad \Sigma_{m_2:\mathtt{nat}}. \quad Id_{\mathtt{nat},m_1,m_2}$$



The model refutes uniqueness of proofs

$$refl'_{\texttt{nat}} \quad : \quad \Gamma \quad \cdot \quad \texttt{nat} \quad \vdash Id_{\texttt{nat}}[\langle id, q\rangle]$$



IV. Conclusion

Achievements

- We've built a Cwf of games and strategies,
- It supports intensional identity types, but refutes both extensionality and uniqueness of proofs.

Not presented here are extensions to:

- Σ-types: no fundamental problem, they can be accomodated in this setting.
- Π-types: necessity to handle dependencies in **contravariant** position. External dependencies extended to $(\triangleright_A^P, \triangleright_A^O)$.
- Extensionality identity types: achieved after a (quite technical) extensional collapse.

Future work

Lots of things to consider.

- Find a (more) elegant formulation of the model with Π and Σ,
- Inductive types,
- Universes,
- Inductive-recursive definitions. . .

QUESTIONS ?