

Type Theory in Ludics

Eugenia Sironi

November 16, 2013

Ludics as game semantics

- **Ludics** may be seen as a non typed **game semantics** based on **interaction**.
- **Design**: an **innocent strategy** (it may also be seen as a proof-like sequent structure).
- **Actions**: **moves** with a **polarity** +,-.
 - +: when a player **talks**
 - -: when a player **receives** what the other player said
 - there is a special positive action, called the **daimon**: \times
→ when one of the two player uses the daimon, he “gives up” and the other player wins.
- **Chronicle**: **play**
an alternating sequence of **actions**.

- **Normalization** of $\{\mathcal{E}, \mathcal{F}\}$ denoted by $[[\mathcal{E}, \mathcal{F}]]$: the result of the interaction between \mathcal{E} and \mathcal{F} .
- **Interaction** between two designs: the travel which starts from the first positive action, moves to the corresponding negative one in the other design, moves upward to the (unique) positive action which follows, and so on.
 - If the corresponding negative action does not exist, normalization **fails**
 - if the action \star is reached, $[[\mathcal{E}, \mathcal{F}]] = \star$, the designs are said **orthogonal** $\mathcal{E} \perp \mathcal{F}$.
 - interaction may also be **infinite**.

Given a set of designs E ,

$$E^\perp = \{\mathcal{F} \mid \mathcal{F} \perp \mathcal{E}, \text{ for all } \mathcal{E} \in E\}$$

Interaction is untyped

Designs can always interact with each other
→ there is **only one arena** of all the possible moves.

- A set of designs closed by biorthogonal is called a **behavior** $\mathbf{G}^{\perp\perp} = \mathbf{G}$.
→ a set of strategies that “behave in the same way”.
- A design \mathcal{D} can belong to several behaviors $\mathbf{E}^{\perp\perp}, \mathbf{G}^{\perp\perp}, \dots$
 - When we consider \mathcal{D} in a behavior, there is a part of \mathcal{D} which is **minimal** w.r.t. the behavior in which we consider it and is still a design.
 - A design can be minimal in a behavior and not in another one.
 - We call **principal** the subset of a behavior whose elements are \star -free and minimal w.r.t. the behavior.

A constructive set theory based on Curry-Howard isomorphism
(proposition as types/sets, proofs as programs).

Types are defined by judgements of the form $\Gamma \vdash t : A$ where t is
a term of type A , w.r.t. a context Γ .

The type \mathbb{N} , the type *List*, the type **cartesian product**
 $(\prod x \in A)B(x)$, the type **disjoint union** $(\sum x \in A)B(x)$.

There are four kind of **rules** used to construct judgements:

- **Formation rule:** How to **form** a new **type A**
- **Introduction rule:** Explains what are the **canonical terms of type A**
- **Elimination rule:** How to define **functions** on the **terms of type A**
- **Equality rule:** How to define **functions** on the **canonical terms of type A**

In MLTT a proposition is characterized by the set of its proofs, in Ludics a behavior is characterized by the designs that generate it.

How to represent a Martin-Löf type in Ludics?

Designs are untyped objects, their type depends on the behavior in which we consider them.

TYPES \rightarrow BEHAVIORS

canonical terms of type $A \rightarrow \mathbb{A}$.

$$A \rightarrow \mathbb{A}^{\perp\perp}$$

Given a Martin-Löf **type** A , it is represented by a **behavior**

$\mathbf{A} = \mathbb{A}^{\perp\perp}$ generated by the **principal** set \mathbb{A} .

The designs of \mathbb{A} represent the canonical terms of type A .

The kind of **rules** used to construct judgements in Martin-Löf Type Theory are interpreted as follows:

- **Introduction rule** → define a set of designs which represent canonical terms
- **Formation rule** → prove that the set defined is principal
- **Elimination rule** → cut
- **Equality rule** → cut-elimination

In Ludics **terms come before type**, as to define a behavior we have to say what are the designs that belong to it.

We represent following the simple types:

- The type Nat with a principal set $\mathbb{N}at$ and the behavior that it generates.
- The type $Bool$ with a principal set $\mathbb{B}ool$ and the behavior that it generates.
- The type $List$ with a principal set $\mathbb{L}ist$ and the behavior that it generates.
- The type $List_n$ with a principal set \mathbb{L}_n and the behavior that it generates.

Natural numbers

A natural number $n \in \mathbb{N}$ is represented by a design \mathbf{n}_σ (σ is a fixed address):

$$\mathbf{0}_\sigma = \frac{-\emptyset}{\vdash \sigma}, \quad (\mathbf{n+1})_\sigma = \frac{\mathbf{n}_{\sigma 01}}{\sigma \mathbf{0} \vdash \vdash \sigma}.$$

$$\mathbb{N}at = \{\mathbf{n}_\sigma \mid n \in \mathbb{N}\}$$

Proposition

$\mathbb{N}at$ is principal.

Thus $\mathbb{N}at^{\perp\perp}$ is the behavior which represent the type of the natural numbers and $\mathbb{N}at$ represents the set of its canonical terms.

Lists

We represent the type of lists of length n following the recursive definition

$$\begin{aligned} \text{List}_{n+1} &::= \epsilon + \mathbb{N} \times \text{List}_n \\ \mathfrak{D}_{0,\xi}^\epsilon &= \frac{-\emptyset}{\vdash \xi \text{ is the empty list,}} \\ \mathfrak{D}_{n,\xi}^{\langle a_1, \dots, a_n \rangle} &= \frac{\frac{\mathfrak{A}_{\xi 0}^{a_1}}{\xi 0 \vdash} \quad \frac{\mathfrak{D}_{n-1,\xi 1}^{\langle a_2, \dots, a_n \rangle}}{\xi 1 \vdash}}{\vdash \xi} \quad \text{for } n > 0, \end{aligned}$$

We denote with \mathbb{L}_n the set of lists of a given length $n \in \mathbb{N}$.

Proposition

\mathbb{L}_n is principal.

Thus we represent with $\mathbb{L}_n^{\perp\perp}$, the type of lists of length n of natural numbers and with \mathbb{L}_n the set of its canonical terms.

The type \rightarrow

Let the types A, B be represented by $\mathbb{A}^{\perp\perp}, \mathbb{B}^{\perp\perp}$, then we represent the type $A \rightarrow B$ with the behavior $(\mathbb{A} \Rightarrow \mathbb{B})^{\perp\perp}$, where

$$\mathbb{A} \Rightarrow \mathbb{B} := \{\mathcal{D} \text{ minimal} \mid \forall \mathcal{A} \in \mathbb{A}, [[\mathcal{A}, \mathcal{D}]] \in \mathbb{B}\}.$$

Proposition

$\mathbb{A} \Rightarrow \mathbb{B}$ is principal.

Thus we represent the type $A \rightarrow B$ with the behavior $(\mathbb{A} \Rightarrow \mathbb{B})^{\perp\perp}$.

What are the elements of $\mathbb{A} \Rightarrow \mathbb{A}$, i.e., the canonical elements of type $A \rightarrow A$?

There is a design, called *Pseudo- $\mathfrak{F}\alpha\mathfrak{x}$* that corresponds to the **copycat strategy**: each move is the copy of the last move of the opponent.

The elements of $\mathbb{A} \Rightarrow \mathbb{A}$ are the *Pseudo- $\mathfrak{F}\alpha\mathfrak{x}$* which copy the branches of the elements of \mathbb{A} .

Simple types and functions on them

We represent following the simple types:

- The type Nat with a principal set $\mathbb{N}at$ and the behavior that it generates.
→ **successor**, the **predecessor** and the **sum**.
- The type $Bool$ with a principal set $\mathbb{B}ool$ and the behavior that it generates.
- The type $List$ with a principal set $\mathbb{L}ist$ and the behavior that it generates.
- The type $List_n$ with a principal set \mathbb{L}_n and the behavior that it generates.
→ **add an element in the head position** and **eliminate the element in the position i** of a list

The type $(\prod x \in A)B(x)$

Let A and a $(B(x))_{x \in A}$, respectively represented by means of the principal set \mathbb{A} and the family of principal sets $(\mathbb{B}(x))_{x \in \mathbb{A}}$.

We represent the type $(\prod x \in A)B(x)$ by the behavior generated by the set of designs

$$(\prod x \in \mathbb{A})\mathbb{B}(x) = \{\mathcal{D} \text{ minimal} \mid \forall \mathfrak{A} \in \mathbb{A}, [[\mathfrak{A}, \mathcal{D}]] \in \mathbb{B}(\mathfrak{A})\}.$$

Proposition

$(\prod x \in \mathbb{A})\mathbb{B}(x)$ is principal.

Thus $(\prod x \in \mathbb{A})\mathbb{B}(x)^{\perp\perp}$ represents the type $(\prod x \in A)B(x)$.

Proposition

$$((\prod x \in \mathbb{A})\mathbb{B}(x))^{\perp\perp} = \bigcap_{x \in \mathbb{A}} (x \Rightarrow \mathbb{B}(x))^{\perp\perp}$$

The type $(\sum x \in A)B(x)$

The disjoint union and the product are two operations dual to each other, then we want the behavior which represents $(\sum x \in A)B(x)$ to be the dual of the behavior which represents $(\prod x \in A)B(x)$.

Work in progress: can $(\sum x \in A)B(x)$ be represented by $\bigcup_{\mathbf{a} \in \mathbb{A}} \{(\mathbf{a}, \mathfrak{F}) \mid \mathfrak{F} \in \mathbb{B}(\mathbf{a})\}$?

Exploiting the usual duality between \cup and \cap through $(.)^\perp$.

THANK YOU!