Proofs and Dialogue : the Ludics view

Alain Lecomte

Laboratoire : “Structures formelles du langage”, Paris 8 Université

February, 2011
Tübingen
with collaboration of Myriam Quatrini
Table of Contents

1. Ludics as a pre-logical framework
   - A polarized framework
   - A localist framework

2. Designs as paraproofs
   - Rules
   - *Daimon* and *Fax*
   - Normalization

3. The Game aspect
   - Plays and strategies
   - The Ludics model of dialogue
Ludics is a theory elaborated by J-Y. Girard in order to rebuild logic starting from the notion of interaction. It starts from the concept of proof, as was investigated in the framework of Linear Logic:

- **Linear Logic** may be polarized (→ negative and positive rules)
- Linear Logic leads to the important notion of proof-net (→ being a proof is more a question of connections than a question of formulae to be proven) → loci
Polarization

Results on polarization come from those on \textit{focalization} (Andréoli, 1992)

- some connectives are \textit{deterministic} and \textit{reversible} (= \textit{negative} ones): their right-rule, which may be read in both directions, may be applied in a deterministic way:

\begin{align*}
&\vdash A, B, \Gamma \\
&\vdash A \otimes B, \Gamma [\otimes] \\
&\vdash A, \Gamma \quad \vdash B, \Gamma [\&] \\
&\vdash A \& B, \Gamma
\end{align*}
the other connectives are non-deterministic and non-reversible ( = positive ones) : their right-rule, which cannot be read in both directions, may not be applied in a deterministic way (from bottom to top, there is a choice to be made) :

Example

\[
\begin{align*}
& \vdash A, \Gamma \\
& \vdash B, \Gamma' \\
& \vdash A \otimes B, \Gamma, \Gamma' \quad [\otimes] \\
& \vdash A, \Gamma \\
& \vdash A \oplus B, \Gamma \quad [\oplus_g] \\
& \vdash B, \Gamma \\
& \vdash A \oplus B, \Gamma \quad [\oplus_d]
\end{align*}
\]
The Focalization theorem

- every proof may be put in such a form that:
  - as long as there are negative formulae in the (one-sided) sequent to prove, choose one of them at random,
  - as soon as there are no longer negative formulae, choose a positive one and then continue to focalize it
- we may consider positive and negative “blocks” → synthetic connectives
- convention: the negative formulae will be written as positive but on the left hand-side of a sequent → fork
Hypersequentialized Logic

Formulae:

\[ F = O|1|P|(F^\perp \otimes \cdots \otimes F^\perp) \oplus \cdots \oplus (F^\perp \otimes \cdots \otimes F^\perp) | \]

Rules:

- axioms:
  \[ \overline{\vdash P, \Delta} \quad \vdash 1, \Delta \quad \overline{\vdash O, \Delta} \]

- logical rules:
  \[ \vdash A_{11}, \ldots, A_{1n_1}, \Gamma \ldots \vdash A_{p1}, \ldots, A_{pn_p}, \Gamma \]
  \[ \frac{(A_{11}^\perp \otimes \cdots \otimes A_{1n_1}^\perp) \oplus \cdots \oplus (A_{p1}^\perp \otimes \cdots \otimes A_{pn_p}^\perp) \vdash \Gamma}{A_{i1} \vdash \Gamma_1 \ldots A_{in_i} \vdash \Gamma_p} \]
  \[ \vdash (A_{i1}^\perp \otimes \cdots \otimes A_{1n_1}^\perp) \oplus \cdots \oplus (A_{p1}^\perp \otimes \cdots \otimes A_{pn_p}^\perp), \Gamma \]

where \( \bigcup \Gamma_k \subset \Gamma^1 \) and, for \( k, l \in \{1, \ldots, p\} \), \( \Gamma_k \cap \Gamma_l = \emptyset \).

- cut rule:
  \[ \frac{A \vdash B, \Delta \quad B \vdash \Gamma}{A \vdash \Delta, \Gamma} \]
Remarks

- all propositional variables $P$ are supposed to be **positive**
- formulae connected by the positive $\otimes$ and $\oplus$ are **negative**
  (positive formulae are maximal positive decompositions)
- $\ldots (\ldots \otimes \ldots \otimes \ldots ) \oplus (\ldots \otimes \ldots \otimes \ldots ) \ldots \oplus (\ldots \otimes \ldots \otimes \ldots )$ is not a restriction because of distributivity
- $((A \oplus B) \otimes C \equiv (A \otimes C) \oplus (B \otimes C))$
Interpretation of the rules

- **Positive** rule :
  1. choose $i \in \{1, \ldots, p\}$ (a $\oplus$-member)
  2. then decompose the context $\Gamma$ into disjoint pieces

- **Negative** rule :
  1. nothing to choose
  2. simply enumerates all the possibilities

First interpretation, as **questions** :

- **Positive** rule : to choose a component where to answer
- **Negative** rule : the range of possible answers
The daimon

Suppose we use a rule:

\[
\Gamma \vdash \text{(stop!)}
\]

for any sequence \(\Gamma\), *that we use when and only when we cannot do anything else...*

- the system now “accepts” proofs which are not real ones
- if \((\text{stop!})\) is used, this is precisely because... the process does not lead to a proof!
- \((\text{stop!})\) is a **paralogism**
- the proof ended by \((\text{stop!})\) is a **paraproof**
- cf. (in classical logic) it could give a distribution of truth-values which gives a counter-example (therefore also: *counter-proof*)
A reminder of proof-nets

\[ \vdash A \perp \emptyset B \perp, (A \otimes B) \otimes C, C \perp \]

\[ \vdash A, A \perp \quad \vdash B, B \perp \]
\[ \vdash A \otimes B, A \perp, B \perp \quad \vdash C, C \perp \]
\[ \vdash (A \otimes B) \otimes C, A \perp, B \perp, C \perp \]
\[ \vdash A \perp, B \perp, (A \otimes B) \otimes C, C \perp \]
\[ \vdash A \perp \emptyset B \perp, (A \otimes B) \otimes C, C \perp \]

Alain Lecomte
Proofs and Dialogue: the Ludics view
Ludics as a pre-logical framework
Designs as paraproofs
The Game aspect

A polarized framework
A localist framework

Proofs and Dialogue: the Ludics view
We define a proof structure as any such a graph built only by means of these links such that each formula is the conclusion of exactly one link and the premiss of at most one link.
Criterion

**Definition (Correction criterion)**

Correction criterion A proof structure is a proof net if and only if the graph which results from the removal, for each \( \& \) link ("par" link) in the structure, of one of the two edges is connected and has no cycle (that is in fact a tree).
Ludics as a pre-logical framework
Designs as paraproofs
The Game aspect

A polarized framework
A localist framework

\[ B \perp \varnothing A \perp A \perp B \perp (A \otimes B) \otimes C \perp C \]
Ludics as a pre-logical framework
Designs as paraproofs
The Game aspect

A polarized framework
A localist framework

Alain Lecomte
Proofs and Dialogue: the Ludics view
Loci

Rules do not apply to **contents** but to **addresses**

**Example**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Contents</th>
<th>Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊢ (e \perp, c)</td>
<td>⊢ (e \perp)</td>
<td>⊢ (e \perp)</td>
</tr>
<tr>
<td>⊢ (e \perp, l)</td>
<td>⊢ (e \perp, c \oplus d)</td>
<td>⊢ (e \perp, c \oplus d)</td>
</tr>
<tr>
<td>⊢ (e \perp, l &amp; (c \oplus d))</td>
<td>⊢ (e \perp )</td>
<td>⊢ (e \perp )</td>
</tr>
<tr>
<td>⊢ (e \perp )</td>
<td>(l &amp; (c \oplus d))</td>
<td>(l &amp; (c \oplus d))</td>
</tr>
</tbody>
</table>
with only loci:

$$
\frac{
\frac{\xi \vdash}{\xi.3.1 \vdash \xi 1} \\
\vdash \xi 1, \xi 2
}{\vdash \xi 1, \xi 2, \xi 3} \\
\frac{\vdash \xi 1, \xi 2}{\vdash \xi 1, \xi 2, \xi 3}
$$

$$
\frac{
\frac{\xi \vdash}{\xi.3.2 \vdash \xi 1} \\
\vdash \xi 1, \xi 2
}{\vdash \xi 1, \xi 2, \xi 3} \\
\frac{\vdash \xi 1, \xi 2}{\vdash \xi 1, \xi 2, \xi 3}
$$
Rules

Definition

positive rule

\[ \ldots \quad \xi \star i \vdash \Lambda_i \quad \ldots \]
\[ \vdash \xi, \Lambda \quad (+, \xi, \Lambda) \]

- \( i \in I \)
- all \( \Lambda_i \)'s pairwise disjoint and included in \( \Lambda \)

Definition

negative rule

\[ \ldots \quad \vdash \xi \star J, \Lambda_J \quad \ldots \]
\[ \xi \vdash \Lambda \quad (-, \xi, \Lambda J) \]
# daimon

- $\text{Dai}$

$\text{Dai} \vdash \text{Λ}$

- it is a **positive** rule (something we choose to do)
- it is a **paraproof**
Is there a identity rule?

- No, properly speaking (since there are no longer atoms!)
- two loci cannot be identified
- there only remains the opportunity to recognize that two sets of addresses correspond to each other by displacement: \( Fax \)

\[
Fax_{\xi, \xi'} = \frac{\ldots Fax_{\xi_{i_1}, \xi'_{i_1}} \ldots}{\ldots \xi' \ast i \vdash \xi \ast i \ldots} \quad (\ast \xi'_{J_1} \ast \ldots) \\
\frac{\ldots \vdash \xi \ast J_1, \xi' \ldots}{\xi \vdash \xi'} \quad (\ast \xi, P_f(\mathbb{N}))
\]
Infinite proofs

- $\mathcal{Fax}$... is **infinite**! (cf. the directory $\mathcal{P}_f(\mathbb{N})$)
- it provides a way to explore any “formula” (a tree of addresses) at any depth
Definition

A design is a tree of forks $\Gamma \vdash \Delta$ the root of which is called the base (or conclusion), which is built only using:

- daimon
- positive rule
- negative rule
Example

\[
egin{array}{ll}
011 & 012 \rightarrow 02 \\
031 & 033 \rightarrow 01 \\
0 & \rightarrow (+,<>\{0\})
\end{array}
\]

- a negative step gives a fixed focus and a set of ramifications,
- on such a basis, a positive step chooses a focus and a ramification
An illustration

- **positive** rule: a question (where will you go next week?)
- **negative** rule: a scan of possible answers is provided, (Roma and Naples or Rome and Florence)
- in case of the choice 1: **positive** rule on the base ”Roma”, new questions (with whom? and by what means?)
- in case of choice 2: **positive** rule on the base ”Florence”, new questions (with whom? and how long will you stay?)
Normalization

- no explicit cut-rule in Ludics
- but an implicit one: the meeting of same addresses with opposite polarity
Example

\[
\begin{align*}
\vdash \xi_{11}, \xi_{12} & \\
\vdash \xi_{21} & \\
\vdash \xi_{22}, \xi_{23} & \\
\vdash \xi_{11} & \\
\vdash \xi_{2} & \\
\vdash \xi_{12} & \\
\vdash \xi & \\
\vdash \xi & \\
\end{align*}
\]
Ludics as a pre-logical framework
Designs as paraproofs
The Game aspect

Rules
*Daimon* and *Fax*
Normalization

\[
\begin{align*}
\vdash \xi_{11}, \xi_{12} & \quad \vdash \xi_{11} & \vdash \xi_{1}, \xi_{2} \\
\xi_{1} & \vdash & \vdash \xi_{11} & \vdash \xi_{2} & \vdash \xi_{21} & \vdash \xi_{22}, \xi_{23} \\
& & \vdash \xi_{1}, \xi_{2} & & \vdash \xi_{2} &
\end{align*}
\]
which is rewritten in:

\[
\vdash \xi_{12} \vdash \xi_{12}, \xi_{11} \quad \vdash \xi_{11} \vdash \xi_{2} \quad \vdash \xi_{21} \quad \vdash \xi_{22}, \xi_{23}
\]

And so on ...
When the interaction meets the *daimon*, it converges. The two interacting designs are said **orthogonal**
Otherwise the interaction is said **divergent**.
Normalization, formally - 1- Closed nets

Namely, a **closed net** consists in a cut between the two following designs:

\[
\begin{align*}
\mathcal{D} & \quad \mathcal{E} \\
\vdash \xi & \quad \xi \vdash (\xi, \mathcal{N}) \\
\end{align*}
\]
Orthogonality

- if $\kappa$ is the daimon, then the normalized form is:

$$
\vdash \quad \kappa
$$

(this normalised net is called dai)

- if $\kappa = (\xi, I)$, then if $I \notin \mathcal{N}$, normalization fails,

- if $\kappa = (\xi, I)$ and $I \in \mathcal{N}$, then we consider, for all $i \in I$ the design $\mathcal{D}_i$, sub-design of $\mathcal{D}$ of basis $\xi \star i \vdash$, and the sub-design $\mathcal{E}'$ of $\mathcal{E}$, of basis $\vdash \xi \star I$, and we replace $\mathcal{D}$ and $\mathcal{E}$ by, respectively, the sequences of $\mathcal{D}_i$ and $\mathcal{E}'$. 
In other words, the initial net is replaced by:

\[
\begin{align*}
&D_{i_1} \quad \vdash \quad \mathcal{E}' \quad \vdash \quad D_{i_n} \\
&D_{i_1} \quad \vdash \quad \xi \star i_1 \quad \vdash \quad \xi \star i_1, \ldots, \xi \star i_n \quad \xi \star i_n \vdash
\end{align*}
\]

with a cut between each \(\xi \star i_j \vdash\) and the corresponding "formula" \(\xi \star i_j\) in the design \(\mathcal{E}'\)
An example of normalization which does not yield *dai*

\[ \mathcal{F}ax_{\xi \vdash \rho} \text{ against a design } \mathcal{D} \text{ of basis } \vdash \xi \]

Let \( \mathcal{D} \) the design:

\[
\begin{array}{c}
\mathcal{D}_1 \\
\xi \star 1 \vdash \quad \mathcal{D}_2 \\
\xi \star 2 \vdash \\
\vdash \xi
\end{array}
\]

Normalization selects first the slice corresponding to \( \{1, 2\} \), after elimination of the first cut, it remains:

\[
\begin{array}{c}
\mathcal{D}_1 \\
\xi \star 1 \vdash \\
\mathcal{D}_2 \\
\xi \star 2 \vdash \\
\mathcal{F}ax \\
\rho \star 1 \vdash \xi \star 1 \\
\mathcal{F}ax \\
\rho \star 2 \vdash \xi \star 2 \\
\vdash \xi \star 1, \xi \star 2, \rho
\end{array}
\]

and finally:

Alain Lecomte
Proofs and Dialogue : the Ludics view
where, in $D'_1$ and $D'_2$, the address $\xi$ is systematically replaced by $\rho$. 

$D'_1$ \[ \rho \ast 1 \vdash \] $D'_2$ \[ \rho \ast 2 \vdash \] $\vdash \rho$
The separation theorem

**Theorem**

If $D \neq D'$ then there exists a counterdesign $E$ which is orthogonal to one of $D, D'$ but not to the other.

Hence the fact that: the objects of ludics are completely defined by their interactions

- a design $D$ inhabits its *behaviour* (= like its *type*)
- a *behaviour* is a set of designs which is stable by bi-orthogonality ($G = G^\perp\perp$)
The game aspect

A slight change of vocabulary:
step in a proof        action
positive step          positive action $(+, \xi, I)$
negative step          negative action $(-, \zeta, J)$
branch of a design     play in a game chronicle
design                 strategy
design (dessein)       as a set of chronicles
Example

\[
\begin{align*}
011 \vdash & \quad 012 \vdash 02 \\
\quad \vdash & \quad 01, 02 \\
\quad \vdash & \quad 01, 03 \\
0 \vdash & \quad (+, <>, \{0\}) \\
\vdash <&> \\
\end{align*}
\]

Example

\[
(+, <>, 0), (−, 0, \{1, 2\}), (+, 01, \{1, 2\}) \\
(+, <>, 0), (−, 0, \{1, 3\}), (+, †)
\]
Dialogue in Ludics

The archetypal figure of interaction is provided by two intertwined processes the successive times of which, alternatively positive and negative, are opposed by pairs.

<table>
<thead>
<tr>
<th>Ludics</th>
<th>Dialogue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive rule</td>
<td>performing an intervention</td>
</tr>
<tr>
<td></td>
<td>or committing oneself (Brandom)</td>
</tr>
<tr>
<td>Negative rule</td>
<td>recording or awaiting</td>
</tr>
<tr>
<td></td>
<td>or giving authorization</td>
</tr>
<tr>
<td>Daïmon</td>
<td>giving up or ending an exchange</td>
</tr>
</tbody>
</table>
Ludics as a pre-logical framework
Designs as paraproofs
The Game aspect

Plays and strategies
The Ludics model of dialogue

\[ \vdash P, \Delta \]

I commit myself to speak of \( P_1, P_2, P_3 \) provided by interlocutor

among authorizations provided by interlocutor
The daimon rule

\[ \vdash \top \]

\[ \vdash \Delta \]

- In proof reading this represents the fact to abandon your proof search or your counter-model attempt.
- This represents the fact to close a dialogue (by means of some explicite signs: “well”, “OK”, . . . or implicitly because it is clear that an answer was given, an argument was accepted and so on . . .).
Convergence and divergence

- Convergence in dialogue holds as long as commitments of one speaker belong to authorizations provided by the other speaker (pragmatics: “Be relevant!” replaced by “Keep convergent!”)
- orthogonality = private communication
- non-orthogonality: normalization may yield side effects: public results of communication
Example of two elementary dialogues:

Example

The first one is well formed:
- Have you a car?
- Yes,
- Of what mark?

\[
Fax_{\xi_{010}, \sigma} \\
\xi_{010} \vdash \sigma \\
\vdash \xi_{01}, \sigma \\
\{\emptyset, \{1\}\} \\
\xi_0 \vdash \sigma \\
\vdash \xi, \sigma \quad \text{VS} \\
\vdash \xi_{010} \\
\xi_{01} \vdash \xi_0 \\
\xi \vdash
\]

\[\xi \vdash \]
Examples

The locus $\sigma$ is a place for recording the answer:

Example

- Have you a car?
- Yes,
- Of what mark?
- Honda.

$$\text{Fax}_{\xi_{010},\sigma}$$

\[\xi_{010} \vdash \sigma\]

\[\vdash \xi_{01}, \sigma\]

\[\xi_0 \vdash \sigma\]

\[\vdash \xi, \sigma\]

VS

$$\xi_{010k} \vdash \xi_{010}$$

$$\xi_{01} \vdash \xi_0$$

$$\xi \vdash$$
The interaction reduces to:

Example

\[
\sigma k \vdash \\
\vdash \sigma
\]

The mark of the car is “Honda”.
This “assertion” is recorded by the speaker.

*It is the function of* \( F\text{ax} \)* to interact in such a way that the design anchored on* \( \xi_{010} \)* is transferred to the address \( \sigma \), *thus* providing the answer.*
The second dialogue is ill-formed: - **Have you a car?**
- * **No, I have no car**
- * **Of what mark?**

\[
Fax_{\xi_{010}, \sigma} \\
\xi_{010} \vdash \xi_{01}, \sigma \\
\vdash \xi_{01}, \sigma \\
\vdash {\{1\}} \\
\xi_0 \vdash \sigma \\
\vdash \xi, \sigma
\]

\[
Fax_{\xi_{010}, \sigma} \\
\xi_{010} \vdash \sigma \\
\vdash \xi_{010}, \sigma \\
\vdash \xi_{01}, \sigma \\
\vdash {\emptyset, \{1\}} \\
\xi_0 \vdash \sigma \\
\vdash \xi, \sigma
\]

\[
\xi_0 \vdash \sigma \\
\vdash \xi_{010}, \sigma \\
\vdash \xi_{01}, \sigma \\
\vdash {\emptyset, \{1\}} \\
\xi_0 \vdash \sigma \\
\vdash \xi, \sigma
\]
### Modelling dialogue

<table>
<thead>
<tr>
<th>Intervention of $S$</th>
<th>Current state</th>
<th>Intervention of $A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$S_1$</td>
<td>$A_2$</td>
</tr>
<tr>
<td></td>
<td>$E_1 = S_1$</td>
<td></td>
</tr>
<tr>
<td>$S_2 = [E_1, A_2]$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_3$</td>
<td>$E_2 = [E_1, S_3]$</td>
<td>$A_2$</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$E_3 = [E_2, S_3]$</td>
<td>$A_2$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Rebuilding Logic

- behaviours
- operations on behaviours

Example

Additives:
- if $G$ and $H$ are two disjoint negative behaviours:
  \[ G \& H = G \cap H \]
- if they are positive $G \oplus H = G \sqcup H (= (G \cup H)\perp\perp)$
Rebuilding Logic-2

Example

Multiplicatives:

- Let us take two positive designs $\mathcal{D}$ and $\mathcal{D}'$ starting from respectively $(+, \xi, I)$ and $(+, \xi, J)$, we may make a new design starting from $(+, \xi, I \cup J)$. The problem is: what to do with $I \cap J$?
  - we may introduce a priority $\rightarrow$ non-commutative $\otimes$
  - or we may stop those branches by $\mathcal{D}ai$ (a special design ended by $\dagger$) $\rightarrow \otimes$
Further developments

- K. Terui’s c-designs: **computational designs**
  - from *absolute* addresses to relative addresses: *variables* of designs
  - ramifications replaced by *named actions with an arity*
  - finite objects: *generators*, in case of infinite designs
  - c-designs are *terms* which generalize λ-terms (simultaneous and parallel reductions via several channels)
  - inclusion of *exponentials* (authorizes replay)

The introduction of variables allows to deal with designs with variables which correspond to designs with partial information (the whole future may stay unknown)
Conclusion

- usually, the logician lives in a dualist universe:
  - proof vs (counter) - model
- with ludics, he lives in a monist universe
  - proof vs counter - proof
- proofs (dessins) and strategies (desseins) are two faces of the same objects
- formulae (= types) are behaviours
- behaviours can be decomposed by means of $\&$, $\oplus$, $\otimes$, thus providing the analogues of formulae of Linear (or Affine?) Logic
- no atoms: such decompositions may be infinite!
- this opens the field to considering very ancient conceptions of Logic (Nāgārjuna) for which there are no grounded foundations of our assertions
Conclusion

- usually, the logician lives in a **dualist** universe:
  - proof *vs* (counter) - model
- with ludics, he lives in a **monist** universe
  - proof *vs* counter - proof
- proofs (**dessins**) and strategies (**desseins**) are two faces of the same objects
- formulae (= types) are **behaviours**
- behaviours can be decomposed by means of &, ⊕, ⊗, thus providing the analogues of formulae of Linear (or Affine?) Logic
- **no atoms** : such decompositions may be **infinite**!
- this opens the field to considering very ancient conceptions of Logic (**Nāgārjuna**) for which there are no grounded foundations of our assertions
Conclusion

- usually, the logician lives in a dualist universe:
  - proof vs (counter) - model
- with ludics, he lives in a monist universe
  - proof vs counter - proof
- proofs (dessins) and strategies (desseins) are two faces of the same objects
- formulae (= types) are behaviours
  - behaviours can be decomposed by means of $\&$, $\oplus$, $\otimes$, thus providing the analogues of formulae of Linear (or Affine?) Logic
  - no atoms: such decompositions may be infinite!
  - this opens the field to considering very ancient conceptions of Logic (Nāgārjuna) for which there are no grounded foundations of our assertions
Conclusion

usually, the logician lives in a dualist universe:

- proof vs (counter) - model

with ludics, he lives in a monist universe

- proof vs counter - proof

proofs (dessins) and strategies (desseins) are two faces of the same objects

formulae (= types) are behaviours

behaviours can be decomposed by means of & , ⊕ , ⊗ , thus providing the analogues of formulae of Linear (or Affine?) Logic

no atoms : such decompositions may be infinite!

this opens the field to considering very ancient conceptions of Logic (Nāgārjuna) for which there are no grounded foundations of our assertions
Conclusion

- usually, the logician lives in a **dualist** universe:
  - proof *vs* (counter) - model
- with ludics, he lives in a **monist** universe
  - proof *vs* counter - proof
- proofs (**dessins**) and strategies (**desseins**) are two faces of the same objects
- **formulae** (= **types**) are **behaviours**
- behaviours **can be decomposed** by means of &, ⊕, ⊗, thus providing the analogues of formulae of Linear (or Affine?) Logic
- **no atoms** : such decompositions may be **infinite**!
- this opens the field to considering very ancient conceptions of Logic (**Nāgārjuna**) for which **there are no grounded foundations of our assertions**
Conclusion

- usually, the logician lives in a **dualist** universe:
  - proof vs (counter) - model
- with ludics, he lives in a **monist** universe
  - proof vs counter - proof
- proofs (**dessins**) and strategies (**desseins**) are two faces of the same objects
- **formulae** (= **types**) are **behaviours**
- behaviours **can be decomposed** by means of $\&, \oplus, \otimes$, thus providing the analogues of formulae of Linear (or Affine?) Logic
- **no atoms**: such decompositions may be **infinite**!
- this opens the field to considering very ancient conceptions of Logic (**Nāgārjuna**) for which **there are no grounded foundations of our assertions**