Ludics and Rhetorics

Abstract

In this paper, we give some illustrations of the expressive power of Ludics with regards to some well known problems often regrouped under the label of Rhetorics. Nevertheless our way of considering Rhetorics encompasses many questions which have been put nowadays in Semantics and Pragmatics.

1 Introduction

Language is mainly interaction. It may even be said, following famous cognitivist S. Pinker that *language emerges from human minds interacting with one another*. The main interest of Ludics, for the study of natural language, resides in the possibility it offers for expressing this interaction.

There are many indices of this interaction in language itself and even in syntax, as witnessed by the presence of many small words which have essentially a rhetoric or pragmatic impact, like ”*even*”, ”*nevertheless*” or ”*but*”. Like French linguist Oswald Ducrot said during the eighties, these words cannot be understood simply in truth-conditionnal terms. ”*But*” is not simply ”*and*” for instance, and if, for a truth-conditionnal viewpoint, there is not a big difference between ”*few*” and ”*a few*”, it remains that from a pragmatic side, it is quite obvious that sentences containing these two words cannot be pursued in the same way. If I say *Peter has read few books by Virginia Woolf*, this can be continued by *therefore he does not know her well as a writer*, and if I say *Peter has read a few books by Virginia Woolf*, this can be continued by *therefore he knows her as a writer a little*, and the converse discourse cannot be pronounced. This can be interpreted if we assume that the speaker answers an implicit question which could be : *Does Peter know Virginia Woolf well as a writer?*, and that ”*few*” is a negative item, while ”*a few*” is a positive one. Vague quantifiers, like *”few”, ”a few”, ”many”, ”a lot”*... denote, as is well known, (approximative) positions on a scale, but specific vague quantifiers have the property of orienting this scale in a direction or in another. ”*But*” has a similar property: taking two propositions as inputs, it does not only provide us with a coordination of them, something which can be done by a simple ”*and*”, it also creates a scale which can be scanned in two opposite directions: one proposition is supposed to be oriented towards one end and the other one towards the other end. This is particularly visible when ”*but*” is used to coordinate two quantified expressions, thus requiring they have opposite directions of variation (cf. *he has many relatives but few friends* vs **he has many relatives but many friends*).

We may also study the phenomenon of presupposition and see about it that it is as if a

dialectical structure was at stake. If A says to B : "I don't regret to have been watching the movie", A not only says something about his/her feelings (that s/he does not regret) but also restricts the ways B can react to this assertion because B is supposed to know that A went to see the movie, or if s/he did not know, is required to include in his/her knowledge database that of course A watched the movie. All these facts are well known. They all assume a model of conversation where each speaker takes into account not only the assertions made by the other but also (and perhaps mainly) the *set of expectations* that each speaker has concerning the reactions of the other.
Ludics exactly provides a framework where the "actions" of one speaker, seen as "positive", not only depend on the actions of the other of the same polarity, but also on its expected answers, that is "negative" actions.
In this paper, we shall develop some concepts of Ludics for language study. The body of the paper will present them in a rather intuitive setting, while the Appendix gives more precise definitions. Section 2 presents the ludical notions of *designs* and *normalization*, with their interpretations in proof-theoretic terms and in strategic ones : Ludics formalizes moves in a game as well as in the research of a proof. Section 3 applies these notions to Rhetorics and Eristics. It makes use of an extension of designs (*c-designs*) due to K. Terui which has two main advantages : it provides a linearized formulation of designs which makes them to look like λ-terms, and it includes *cuts*. Section 4 tries to give an account of *semantics*, as it is generally understood, that is the context-free study of meanings, but instead of staying stuck to a truth-conditional conception of meaning, we envisage it as something built in interactions. There are two ways of looking at meaning : as a simple set of justifications (when we take in consideration all the possible objections which can be made to a sentence), or as a *behaviour*, that is the set of all designs which give the same results for all the interactions (in terms of *convergence / divergence*). Complex articulated sentences may be viewed as combinations of more elementary behaviours. After a completeness theorem, the behaviours generated by such operations are complete, in the sense that they contain no other design than those obtained by the operations. Because a behaviour is ordered by a relation of refinement, and because behaviours may be ordered by inclusion, it is possible to associate *a family* of behaviours with a sentence, each of them reflecting some degree of refinement. Because, for a sentence, there is in principle no end to the process of its examination, there are no "atoms" properly speaking. In the conclusion, we come back to the main features of our approach and emphasize its proximity with some neuro-cognitivist views on meaning (Changeux).

2 Ludics for Language Moves

2.1 Designs

Let us call *language move* every move that a speaker makes during a conversation. A positive move consists in an explicit assertion or question. A negative one consists in a way of collecting the content of the other's utterrance and reacting to it in a purely mental way. If somebody says to me:

(1) *Are you still smoking?*

I will store in my short term memory that *this is a question* and that it is assumed that *I was a smoker (and even that perhaps I am still one)*. I also know of course that the speaker who says (1) has some informations on me and, in particular that s/he knows (or believes) that I was a smoker. Therefore when s/he asked his or her question, s/he had in mind some of the states in which I can be. In the absolute, these states are combinations of the following:

I was a smoker	vs	*I was not a smoker*
I presently smoke	vs	*I don't presently smoke*

The point here is that by (1), the speaker *a priori* eliminates two states, those the commun feature of which is *I was not a smoker*.

Let us represent the various possible elementary states by integers $0/1$, which are also called *bias*. A *move* is a sequence of such bias. Many exchanges are such that they have only moves of length 1: in such a case, the player who moves (either negatively or positively) simply adds a bias to a previous sequence which summarizes the history of the exchange, but sometimes s/he adds three bias at a time or even perhaps more. Let us suppose a positive move of a player is an *elementary* question. By asking it, s/he adds a bias, let us call it 0. After this positive move, s/he makes a negative one, which consists in expecting an answer 0 or 1 (if there are only two possible answers, like it is the case for dichotomic questions) : these are precisely the *loci* where the other player could play if there would be no presupposition (case of *did you ever smoke?*, for instance). But by asking a complex question like (1) (that Aristotle named a *multiple question*), two elementary questions are combined (to each of which we assign a bias 0) so that in fact, the only "loci" the speaker provides to his or her opponent are 0000 and 0001 (and neither 0100 nor 0101). Let us therefore assume that the speaker starts from scratch (a situation which is in reality never the case!), we represent that by the empty locus $< >$. By asserting (1), the speaker directly jumps to the set of possible answers $\{0000, 0001\}$. If it happens that, in reality, I never smoked, I have no locus to answer: we say that the conversation locally diverges. Only a "meta"-game allows us to fix the interaction. This "meta"-game uses pieces of interaction which are "ready made", for instance one of them consists in forcing the speaker to *retract* oneself. This is done by *erasing* the whole interaction and *replacing* it by a more fine-grained interaction, according to which:

1. the speaker gives at first the alternatives 00 and 01
2. and plans, if the other speaker answers by 00, to give another set of alternatives 0000 and 0001

There is another kind of move, the one in which the speaker decides s/he has got enough information, for instance by means of an answer from the other speaker which satisfies him or her. There is no bias added in that situation, only a signal indicating that the exchange is over. We note † such a move, which is a positive one. Generally, no speaker's viewpoint can end up on an indefinite wait for a positive action[1].

We may represent this interaction in the following way:

[1] except in case of a partial design, the last positive rule being symbolized by Ω which precisely means the absence of rule.

The Speaker's viewpoint (S)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash 0000}\dagger \quad \dfrac{}{\vdash 0001}\dagger}{000 \vdash}(-,000,\{\{0\},\{1\}\})}{\vdash 00}(+,00,\{0\}) \qquad \dfrac{}{\vdash 01}\dagger}{0 \vdash}(-,0,\{\{0\},\{1\}\})}{\vdash < >}(+,< >,\{0\})$$

"My" viewpoint A

$$\dfrac{\dfrac{\dfrac{\dfrac{0000 \vdash}{\vdash 000}(+,000,\{0\})}{00 \vdash}(-,00,\{\{0\}\})}{\vdash 0}(+,0,\{0\})}{< > \vdash}(-,< >,\{\{0\}\}) \qquad or \qquad \dfrac{\dfrac{\dfrac{\dfrac{0001 \vdash}{\vdash 000}(+,000,\{1\})}{00 \vdash}(-,00,\{\{0\}\})}{\vdash 0}(+,0,\{0\})}{< > \vdash}(-,< >,\{\{0\}\})$$

A third issue is the one for which I object that I never smoked, which would be, on my viewpoint:

$$\dfrac{\dfrac{01 \vdash}{\vdash 0}(+,0;\{1\})}{< > \vdash}(-,< >,\{\{0\}\})$$

We may contrast this kind of exchange with the one in which there is *presupposition* (or multiple question). The Speaker's viewpoint is replaced by:

$$\dfrac{\dfrac{\dfrac{}{\vdash 0000}\dagger \quad \dfrac{}{\vdash 0001}\dagger}{000 \vdash}(-,000,\{\{0\},\{1\}\})}{\vdash < >}(+,< >,\{0\});(-,0,\{\{0\}\});(+,00,\{0\})$$

A's viewpoint is replaced by:

$$\dfrac{\dfrac{0000 \vdash}{\vdash 000}(+,0,\{0\})}{< > \vdash}\mathfrak{C} \qquad or \qquad \dfrac{\dfrac{0001 \vdash}{\vdash 000}(+,0,1\})}{< > \vdash}\mathfrak{C}$$

Where $\mathfrak{C}$ is the chronicle $(-,< >,\{0\})(+,0,\{\{0\}\})(-,00,\{0\})$.

2.2 Normalization

Let us examine how the previous viewpoints (or *designs*) may interact.

1. at the bottom of each design, we have either $\vdash < >$ or $< > \vdash$, that is twice the same *locus* but with two different polarities. This is exactly similar to a situation where the cut-rule may apply in a sequent calculus. The difference is here that there will be no formulation of *cut* as a rule, but simply we shall consider a cut as a situation where two *loci* meet with two different polarities. In that case, this *cut* may be eliminated according to the standard technique. After the first elimination, what remains is:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash 0000}\dagger \quad \dfrac{}{\vdash 0001}\dagger}{000 \vdash}}{\vdash 00} \qquad \vdash 01}{0 \vdash} \qquad\qquad \dfrac{\dfrac{\dfrac{0000 \vdash}{\vdash 000}}{00 \vdash}}{\vdash 0}$$

2. a new cut appears after this first elimination: between $0 \vdash$ and $\vdash 0$. This exactly illustrates the fact that there is a minimal agreement between two speakers: the second agrees to record in his or her own mind the question asked by the first one, what remains is:

$$\dfrac{\dfrac{\dfrac{}{\vdash 0000}\dagger \quad \dfrac{}{\vdash 0001}\dagger}{000 \vdash}}{\vdash 00} \qquad\qquad \dfrac{\dfrac{0000 \vdash}{\vdash 000}}{00 \vdash}$$

3. again new cuts appear, the first one between $00 \vdash$ and $\vdash 00$, then between $000 \vdash$ and $\vdash 000$, and finally between $0000 \vdash$ and $\vdash 0000$, when this last cut is eliminated, what remains is :

$$\dfrac{}{\vdash}\dagger$$

   something we shall consider a *null* object.

Such a termination will be associated to a *convergence* case (see the Appendix). The two objects which have converged this way when put together are said to be *orthogonal designs*. It is now obvious that our so called third issue does not converge with the presuppositional question made by S since after the first and the second cut-eliminations, there is no cut situation any longer.

2.3 Strategies, proofs and designs

By paying attention to the previous example, we can observe that what we named ”viewpoints” were in fact exactly like *strategies* that players can have when playing together. In the Appendix, the reader can find the definition of *designs*. Intuitively speaking, designs are sequences of moves, each of which being associated with the application of a rule (positive or negative), they can therefore be seen at the same time as *deductions* or *proofs*. The dialectical interaction that we have between a speaker S and his or her co-speaker A may be seen either as the opposition of two strategies

in a game or as a tentative to build a proof against the objections (or the counter-proof) of the other speaker. One of the main differences with the usual games lies in the fact that there may be no winner in that kind of game : if we believe in Grice's Cooperation Principle, the goal is *not* to win against the other speaker but to reach together a situation in which there is an *agreement* on *expectations*. Such a situation is expressed in terms of convergence. Divergence, on the contrary, may be assigned to *failure*, like *presuppositional failure*.
Let us otherwise notice that, conceived this way, dialectical exchanges seem to occur not by opposing steps to steps, one by one, but by opposing a whole strategy to another one. It is as if each speaker had in his or her own mind, a whole plan, or as if s/he was *projecting* an entire design. This seems to be in agreement with present views in neurosciences, as attested by these words of Jean-Pierre Changeux (himself quoting works by Sperber and Wilson):

> ”Human communication generally takes place in a well defined context of knowledge in which speakers are informing each other [...] Aiming at maximizing the efficience of communication, each speaker tries to recognize and to infer the intention of the one who communicates. In other words, when communication begins, each partner has in his or her own mind the whole possible content of the speech, which constitutes a subset of all his or her knowledge on the world. [...] We may think that each speaker constantly tries to project his or her frame of thought into the mind of his or her co-speaker”.

3 Some Applications to the Argumentation Theory

3.1 Controversies

In the previous section, it was seen that by associating a design with a speaker's viewpoint in a dialogue, we may highlight its interactive features. In doing so, we give an account of the interactive content of a discourse in the following way : a speech turn is anchored on a locus created by the previous one, due to the other speaker, and creates new *loci* on which the interaction will continue. The design associated with a viewpoint can be more or less elaborated : in the simplest cases, it is only a positive action, but sometimes, like in a presuppositional case, it is a sequence of successive actions (or *chronicle*). Moreover, it may be still more elaborated in the case of *controversies*, which we will consider below.
Let us define a *controversy* as a language game in which there is a goal, consisting in getting a *winning* position in a debate[2]. Controversies are therefore a subset of the class of dialogues.

[2]Technically speaking, a strategy is winning if it does not use the daïmon. We retrieve the notion of *winning design* defined by J.-Y. Girard.

3.2 Fallacies and c-designs

3.2.1 c-designs

Controversies may contain figures of dialogue, called *fallacies*, which are mainly used to confound the interlocutor. We know that, in his *Sophistical Refutations* (or *De Sophistis Elenchis*), Aristotle was the first to show how to refute these dialectical tricks.
We will try to see in the following how it is possible to characterize fallacies by means of special properties of the designs which represent them. In order to do so, we will use a variant of Girard's designs, called "c-designs"[3], defined by K. Terui [Terui 08]. "c" comes from "computational" : c-designs extend ordinary designs in that they contain explicit interactions and moreover, they allow to define infinite designs by means of finite devices called their *generators*.
The reader will find a more complete presentation of c-designs in the Appendix. Let us only focalize on their main features. Instead of using the proof-like presentation, the c-designs may be roughly described as generalised λ-terms. In the term calculus which results, we have not a simple, unique application but many ones, in fact as many as there are elements in a signature set $\mathcal{A}$ consisting in a given set of pairs (a, n) where a is a name and n is an arity. The normal terms or cut-free c-designs are still sequences of alternated actions, but :
- positive actions are either constants: † (daïmon or abandon) and Ω (divergence or absence of rule), or proper and specific actions (denoted by $\overline{a}$ for a given name a);
- negative actions are either variables (x, y, z,...) or proper negative actions (denoted by $a(x_1, ..., x_n)$).
Then the terms (or c-designs) are defined as follows:
- a negative c-design is either a variable or a sum of negative actions applied with positive c-designs as operands: $a_0(\vec{x}_{a_0}).P_{a_0} + \cdots + a_k(\vec{x}_{a_k}).P_{a_k}$;
- a positive c-design is either a constant († or Ω) or an application which is denoted by $N_0||\overline{a} < N_1, \ldots, N_n >$ where $||$ indicates an interaction (or cut). Such an interaction is an application in the following sense: if N_0 contains a subterm $a(\vec{x}_a).P_a$ then we have to perform the application $(P_a)N_1 \ldots N_n$. Precisely, in such a case $N_0||\overline{a} < N_1, \ldots, N_n >$ reduces into $P_a[N_1/x_1, \ldots, N_n/x_n]$. Otherwise, if there is no subterm $a(\vec{x}_a).P_a$ in N_0 (or, equivalentely, if N_0 contains the subterm $a(\vec{x}_a).\Omega$) the interaction diverges. For instance, the design (i) below is translated into the term (ii):

$$(i)\qquad \dfrac{\dfrac{\dfrac{}{\vdash x.6.1, x.6.2}\dagger \qquad \dfrac{x.6.0.3 \vdash \quad x.6.0.4 \vdash}{\vdash x.6.0}(x.6.0, \{3,4\})}{x.6 \vdash}(x.6, \{\{1,2\},\{0\}\})}{\vdash x}(x, \{6\})$$

$$(ii)\qquad P = x||\overline{\{6\}} < \{1,2\}(x_1, x_2).\dagger + \{0\}(y).y||\overline{\{3,4\}} < \Omega^-, \Omega^- >>$$

where Ω^- is a notation for $\Sigma_{a\in\mathcal{A}} a(\vec{x_a}).\Omega$.
Since in P the first negative term is the variable x, there is no cut. In such a case we

[3] see more details in the Appendix.

use the term ”chanel” to talk about the variable x and indicates that it is the locus on which an interaction may be plugged in. This design, based on x, consists in a positive action (named $\overline{\{6\}}$) which gives access to two negative c-designs:

- the first one begins by a negative action $\{1, 2\}(x_1, x_2)$ which, in principle, binds the variables x_1 and x_2 in any c-design which follows (here †, which is a constant).
- the second one performs a negative action $\{0\}(y)$ which binds y inside the positive c-design: $y||\overline{\{3, 4\}} < \Omega^-, \Omega^- >$ which follows this negative action. The later positive design reduces to a simple positive action $\overline{\{3, 4\}}$ since in fact no (total) negative c-design follows it.

Let us note that c-designs (terms) where $||$ occurs between a mere channel and a subdesign correspond to Girard’s designs. Nevertheless, there are cases for which $||$ occurs between two subdesigns : in these cases, c-designs extend Girard’s designs, because they now involve *cuts*. $||$ is therefore interpreted as a cut-plugging. For instance, the cut-net (iii) below is translated into the c-design (iv) :

$$(iii)\qquad \dfrac{\dfrac{\dfrac{z.6.1\vdash \quad z.6.2\vdash}{\vdash z.6}}{z\vdash} \qquad \dfrac{\dfrac{\dfrac{}{\vdash z.6.1, z.6.2}\,\dagger \quad \dfrac{z.6.0.3\vdash \quad z.6.0.4\vdash}{\vdash z.6.0}}{z.6\vdash}}{\vdash z}}{L \qquad\qquad\qquad P}$$

$$(iv)\qquad P[L/x] = L||\overline{\{6\}} < N >$$

with:

$$L = \{6\}(z).z||\overline{\{1, 2\}} < \Omega^-, \Omega^- >$$
$$N = \{1, 2\}(x_1, x_2).\dagger + \{0\}(y).(y||\overline{\{3, 4\}} < \Omega^-, \Omega^- >$$

3.2.2 Petitio Principii

A typical fallacy is provided by *Petitio Principii*, translated by *begging the question* in English. Like this expression tells us, it is the rhetorical figure which consists in *smuggling the conclusion into the wording of the premises, thus begging or avoiding the question at issue in the argument* (Schipper and Schuh, quoted by [Hamblin 70]). In other words, a given argument depends on what it is trying to support, and as a result, the proposition is being used to prove itself.
We characterise such a fallacy in Ludics as a block where there are no loci on which a continuation of the interaction can be performed. The *loci* that the intervention should create are in fact never available. Like if, during a game, your turn was never given back. An example is provided by:
(2) *Your daughter became dumb because she lost the use of language*
Let us take the following notations:

- E , E_1, E_2 and E' respectively denote the utterances *your daughter became dumb because she lost the use of language*, *your daughter became dumb*, *she lost the use of language* and *your daughter lost the use of language because she became dumb*;

- e, e_1, e_2 and e' denote the names of actions which are respectively associated with them.

Let us describe below the design $\mathcal{E}$ associated with E:

- $\mathcal{E} = y||\overline{e_1} < \mathcal{N} >$ where:

  - y is the channel where an answer may be plugged in;

  - $\overline{e_1}$ is the first positive action and corresponds to the claim E_1: *your daughter became dumb*;

  - the design $\mathcal{N}$ is associated with the justification of E_1.

- to make $\mathcal{E}$ more explicit, we need to precise the design $\mathcal{N}$ which contains as a justification for E_1, the argument E_2. We have : $\mathcal{N} = e_2(x).\mathcal{E}'$. Indeed the negative action $e_2(x)$ gives an account of the position which is ready to support E_2: *she lost the use of language*. Moreover we may forecast that such a support would be the utterance E': *your daughter lost the use of the word because she became dumb*, with which is associated the design $\mathcal{E}'$.

- Clearly the design $\mathcal{E}'$ is equal to $\mathcal{E}$ except that the positions of e_1 and e_2 are exchanged. Precisely, $\mathcal{E}' = x||\overline{e_2} < \mathcal{N}' >$, where $\mathcal{N}'$ is $e_1(z).\mathcal{E}''$, and $\mathcal{E}'' = \mathcal{E}[z/y]$, and so on ...

Finally the design $\mathcal{E}$ associated with the utterance E: *your daughter became dumb because she lost the use of language* is:

$$\mathcal{E}_y = y||\overline{e_1} < e_2(x).x||\overline{e_2} < e_1(z).z||\overline{e_1} \cdots >>$$

The design $\mathcal{E}$ is infinite. Nevertheless, we may give a finite presentation of it, by means of a finite generator $\mathcal{G}$ (see in appendix):

$\mathcal{G} = (\{s_1^+, s_2^+\}, \{s_1^-, s_2^-\}, l, \{s_1^+\})$, where the function l is defined as follows:

$$\begin{array}{rcl} l(s_1^+) & = & y||\overline{e_1} < s_1^- > \\ l(s_1^-) & = & e_2(x).s_2^+ \\ l(s_2^+) & = & x||\overline{e_2} < s_2^- > \\ l(s_2^-) & = & e_2(y).s_1^+ \end{array}$$

The design-like presentation highlights the characteristics of “petitio principii”:

- the design is infinite; the loci to which the addressee could stick oneself are never available.

- the design’s generator gives an account of the circularity of the argument.

3.3 Transfering the premises from a locus to another one

A well known work on what is sometimes named *eristic*, according to the ancient Greek word *Eris* meaning ”wrangle” or ”strife”, is the famous *The Art of Always Being Right* written by Schopenhauer. In this book, the German philosopher gives several ”stratagems” according to which it is easy to win a controversy against any opponent. For instance, the ”fourth stratagem” is the following :

> *Make the opponent to admit the premises of a proposition, in a hidden way during the conversation. Once it is visible that your opponent has conceded all the necessary premises, play the sentence implied by these premises.*

Let us build the design associated with a speaker who argues in favour of his or her thesis by refering to premises already accepted by the other speaker.
Let us use the following notations :

- the speaker is referred to by ”player” or P
- the addressee by ”opponent” or O.
- the utterances corresponding to the premises already accepted by O are denoted A and B.
- we assume that the claim made by P by using the premises already accepted by O, is similar to the following E: *Since A and B (that you accepted) imply C, you will agree that C.*
- we denote by I the utterance A *and* B *imply* C.
- the names a, b, e, i are respectively associated with the utterances A, B, E et I, and their arities will be made precise below.

We associate with E the following design:

$$\mathcal{E} = y||\overline{e_1} < a(x_a).\mathcal{A}, b(x_b).\mathcal{B}, i(x_i).\mathcal{I} >$$

which is built as follows:

- y is the channel where an answer may be plugged in;

- $\overline{e_1}$ is the first positive action : it consists in the claim of the thesis C. This action is ternary since it suggests that the interaction may continue on each element which constitutes the logical argumentation: the two premises and the implication.

- then P is ready to continue the interaction on three channels, represented by three negative actions $a(x_a)$, $b(x_b)$, $i(x_i)$.

- the designs $\mathcal{A}$ and $\mathcal{B}$ respectively associated with the support of the premises A and B are already built.

Let us consider the following simplified case: during previous dialogues, the utterances A and B were asserted by P and immediatly accepted by O. We can then represente each of them by some design reduced to an elementary positive action, respectively : $x_a||\overline{a} < \Omega^- >$ and $x_b||\overline{b} < \Omega^- >$. We give an account of the transfer of these premises so that they become arguments of the thesis C by the fact that $\mathcal{A}$ et $\mathcal{B}$ are respectively obtained on the following way:

$$\mathcal{A} = \mathcal{F}ax_{x_a}||\overline{a} < \Omega^- > \text{ et } \mathcal{B} = \mathcal{F}ax_{x_b}||\overline{b} < \Omega^- >$$

where $\mathcal{F}ax$ is an infinite design generated by the following finite[4] generator:

$$(\{s_u\}_{u \in U}, \{s_N\}, l, s_N) \text{ where :}$$

$$l(s_N) = \Sigma_{u \in U} u(\vec{x}_u).s_u, \quad l(s_u) = y||\overline{u} < s_N, \ldots, s_N > \text{ when } y \notin \vec{x}_u.$$

Then Fax_{y_a} is the negative c-design:

$$\Sigma_{u \in U} u(x_1, \ldots, x_n).(y_a||\overline{u} < Fax_{x_1}, \ldots, Fax_{x_n} >)$$

with which the normalisation of a positive design $\mathcal{D} = x_a|| \ldots$ gives $\mathcal{D}[y_a/x_a]$.
Since O knows that the designs $x_a||\overline{a} < \Omega^- >$ and $x_b||\overline{b} < \Omega^- >$ are winning P's designs, his or her only possibility to successfully continue the dialogue is to use the channel $i(x_i)$. Indeed the subdesign $\mathcal{I}$ is still partial at this step[5]. If O has nothing to oppose to I, then s/he looses: s/he is obliged to play the daïmon.

Let us underline that such an interaction may be taken into account because we are able to express cuts inside the representations of the speech turns. These internalized cuts allow us to keep the information coming from previous exchanges and to be able to reuse it in the future.

4 Ludics and "semantics"

4.1 On Natural Language Semantics

In the previous sections, we were concerned by *rhetoric*, that is the way in which language is used according to a persuasion goal. Rhetoric is a question of *positions* that speakers occupy in their interlocutory space. In particular, we saw in the previous section how a speaker is bound to make use of the other speaker's expectations. In contrast with rhetoric (which takes place in fact inside what is nowadays called *pragmatics*, that is the study of the *use* of language in context), *semantics* is supposed to deal with the proper content of a sentence (more or less independantly of its context). Traditional Formal Semantics does as if there existed an objectivable sentence meaning which could even be reduced to truth conditions, according to Frege's program. Another viewpoint amounts to consider that :

- meaning is always decided in context, that is, more precisely, in dialectical exchanges,
- meaning is always determined according to reciprocal expectations coming from two speakers (or more) in a dialogue

[4]provided that U (a set of names associated with some utterances) is finite
[5]that is it has still some missing rule over it

Of course, when dealing with content, we are obliged to start from some primitive meanings associated with words (*lexical meaning*) and from primitive ways in which those contents may be combined. For instance, when utterring *there is a cat on the mat*, we refer to primitive concepts like those of `cat` and `mat`, and also on relational concepts like `being on`... At a first glance, we may ignore what there is exactly *inside* those concepts! Tarskian semantics would say : `cat` is the concept defined by the function which assigns 1 to every individual x which is a cat, and 0 to the other individuals... We don't think this kind of view bring anything to the comprehension of the semantics of natural language. If we try to reason more in accordance with modern neurocognitive views, we should prefer to say that `cat` is that part of the brain which reacts when the word is heard or when a real cat crosses the road in front of us... But we can also leave the door open to other conceptions : in fact, for us, `cat` will be... a set of designs, (or *behaviour*) seen as the set of all designs which interact the same way with regards to the other designs inside our situational representation of the world, at the moment we have an exchange about cats with other people. Of course `cat` as a notion may be more or less deepened in a given situation : it may suffice for us to identify a cat simply by a single feature (its miewing, its whiskers or else...), or we can be in a situation where we expect more, something like a *proof* that there is a true cat! Here the separation theorem is fundamental. It states that designs may be ordered inside the same behaviour. Very long designs may inhabit the behaviour associated with the notion, as well as shorter and more branching ones.

We propose here a conception of *interactive meaning* based on Ludics. At a metaphoric level, in the same way a design is defined by its orthogonal (according to the separation theorem), we postulate that the meaning of a sentence is given by its dual sentences : that is the sentences with which the interaction converges. Moreover, we claim that Ludics also offers a framework in which we may modelize the "meaning" of a sentence.

We follow below an example which illustrates a classical problem of ambiguity (*scope* ambiguity).

4.2 Meaning through dual sentences

Let us consider the statement (from now on denoted by S) :

(4) *Every linguist speaks some african language*

Usually two "logical forms" can be associated with such a sentence S, depending on whether *some* has the narrow or the wide scope. Namely:

$S_1 = \quad \forall x(L(x) \Rightarrow \exists y(A(y) \wedge P(x,y)))$

$S_2 = \quad \exists y(A(y) \wedge \forall x(L(x) \Rightarrow P(x,y)))$

where $L(x)$ means"x is a linguist" , $A(y)$ means "y is an african language" and $P(x,y)$ means "x speaks y".

When "some" has the narrow scope, we assume that the logical form converges with the LF of sentences like:

(1) There is a linguist who does not know any african language.

(2) Does even John, who is a linguist, speak an african language ?

(3) Which is the African language spoken by John ?

On the opposite, if ”some” has the wide scope, the logical form converges with :
(4) There is no african language which is spoken by all the linguists.
(5) Which african language every linguist speaks ?

4.3 Meaning as a set of justifications

In this section, for the sake of brevity, we shall discard the $L(x)$ part of the S-formulae, simply considering the following formulae:
$S'_1 = \quad \forall x \exists y (A(y) \wedge\ P(x,y))$
$S'_2 = \quad \exists y (A(y) \wedge \forall x\ P(x,y))$
We realize the idea according to which meaning is equated with a set of dual sentences by associating with S's meaning a set of designs. These designs will be seen as *justifications* for S, that is the supports of potential dialogues during which a speaker P can assert and justify the statement S against an adressee O who has several tests at his/her disposal.
Let us denote by $\mathcal{E}$ such a design. We can write it as :

$$\mathcal{E} = y||\overline{e_1} < \mathcal{N} > \quad \text{or} \quad \mathcal{E} = y||\overline{e_2} < \mathcal{M} >$$

where y is the channel by which the interaction is performed, e_1 (resp. e_2) is the name associated with S when ”some” has the narrow scope (resp. the wide scope) and $\mathcal{N}$ (resp. $\mathcal{M}$) corresponds to an expected interaction.
Let us focus on $\mathcal{N}$. It may be written :

$$\Sigma_{e_l \in L} e_l(\vec{x}).\mathcal{E}_l$$

where $\mathcal{E}_l$ is a justification of E_l : *the linguist l speaks an african language*, and the e_l's are the names associated with those utterances.
Let us then explore $\mathcal{E}_l$. If a is the name associated with the utterance A: *F is the african language that l speaks*, the designs $\mathcal{A}_1$ and $\mathcal{A}_2$ are justifications of, respectively, A_1: *F is an african language* and A_2: *l speaks F*. We may write :

$$\mathcal{E}_l = x||\overline{a} < \mathcal{A}_1, \mathcal{A}_2 >$$

Example : We may have: $\mathcal{N}_1$= $a_1(x_1).(x_1||\overline{\emptyset})$ and $\mathcal{N}_2$= $a_2(x_2).(x_2||\overline{\emptyset})$. In such a case the locutor justifies A_1 and A_2 by saying that there are *datas* which justify them.
Finally we obtain as a first example of S's justification:

$$\mathcal{E}_0 = y||\overline{e_1} < \Sigma_{e_l \in L} e_l(\vec{x}).(x||\overline{a} < a_1(x_1).(x_1||\overline{\emptyset}), a_2(x_2).(x_2||\overline{\emptyset}) >) >$$

which may be read as follows:

- P asserts e_1. The positive action is $\overline{e_1}$
- Then he is ready to listen objection for every linguist l (the negative actions $e_l(\vec{x})$).
- For every linguist l P is able to exhibit some language F, arguing that F is an african language and F speaks l. This is the positive action $\overline{a}$.

- Lastly, if O had still some doubts about one of these two claims (expressed by the negative actions $a_1(x_1)$ and $a_2(x_2)$), P could continue to give justifications, but here, s/he asserts that they are provided by mere datas ($\overline{\emptyset}$).

Remark : The design $\mathcal{E}_0$ is built in order to normalize with a design associated with an attempt to negate it : ”There is some linguist which doesn’t speak any african language”. We could also find other designs as justifications for S with its first reading, for instance O could ask to check if l really speaks F, if F is really an africal language and so on ...
Example :
The following design would also be convenient :

$$\mathcal{E} = y||\overline{e_1} < \Sigma_{e_l \in L} e_l(\vec{x}).(x||\overline{a} < \Omega^-, \Omega^- >) >$$

It differs from the previous one by the fact that it doesn’t plan to justify the statement A: *F is the african language that l speaks*. In such a case, a counter design may normalize only if it plays the daïmon againts P’s action $\overline{a}$.
The following one is still a design that could be convenient :

$$\mathcal{E}_1 = y||\overline{e_1} < \Sigma_{e_l \in L} e_l(\vec{x}).(x||\overline{a} < a_1(x_1).(x_1||\overline{\emptyset}), a_2(x_2).(x_2||\overline{g} < \mathcal{N}_1, \mathcal{N}_2) >) >$$

Here, instead of justifying a_2 by a data, P goes deeper and gives a more detailed justification, for example G: *l spent his childhood in Tunisia and went to a local school*, $\mathcal{N}_1$ and $\mathcal{N}_2$ (not detailed here) are the subdesigns associated with the underlying utterances of G.
A first and rough approximation of the meaning of the sentence could therefore be a set of such designs, which are all supports of potential dialogues. In this case, if $\mathbb{S}_1$ denotes the set of designs representing the first reading of S and if $\mathbb{S}_2$ denotes the set of designs representing the second one, the set of designs representing the meaning of S is the union of these two sets : $\mathbb{S} = \mathbb{S}_1 \cup \mathbb{S}_2$.

4.4 Meanings as behaviours (from designs to proofs)

The concept of *behaviour* (see the Appendix) will be the key concept. Actually, it is by its means that we may recover the notion of *formula*.
Let us recall from the Appendix that a behaviour is a *set of designs closed by biorthogonality*, that is a set of designs which have the same behaviour with regards to normalization with the other designs. Generally speaking, a behaviour is the orthogonal of some set of designs. The behaviour generated by a design $\mathcal{D}$ is $\mathcal{D}^{\perp\perp}$. By associating the meaning of an utterance with a behaviour, we get, in our interactive setting, a counterpart to the more usual notion of a ”logical form” associated with a sentence.
The entire meaning of the sentence (4) above is given an account as a behaviour *and* as a linear formula expressed in $HSLL$, the *hypersequentialized polarised linear logic* (see the Appendix). In $HSLL$, at a first glance, formulae S_1 and S_2 can be expressed as :
$S_1 = \quad \forall x\ L(x) \multimap \exists y(A(y) \otimes P(x,y)))$
$S_2 = \quad \exists y\ A(y) \otimes \forall x(L(x) \multimap P(x,y)))$

If we still concentrate ourselves on the reduced forms :

$S'_1 = \quad \forall x\, \exists y (A(y) \otimes P(x,y))$

$S'_2 = \quad \exists y\, A(y) \otimes \forall x\, P(x,y))$

and If we go back to the designs defined in the previous section, we see that $\mathcal{E}_0$, $\mathcal{E}_1$ and $\mathcal{E}$ share the same series of initial steps. We may be easily convinced that any design which could be a justification for S would begin the same way. This suggests that they belong to the same behaviour, or in other words, that each of them may be seen as an attempt to prove the same formula, namely the formula $S = S'_1 \oplus \downarrow S'_2$ where S'_1 and S'_2 are given above, and where $\downarrow$ is used to deal with the subformula S'_2 separately.
Here is an attempt to prove S'_1:

$\mathcal{E}$ =

$$\dfrac{\begin{array}{c}\mathcal{D}_{l'}\\ \vdots\end{array} \qquad \dfrac{\downarrow A^\perp(F)\vdash \qquad \downarrow P^\perp(l,F)\vdash}{\vdash \exists y(\uparrow A(y)\otimes \uparrow P(l,y))} \qquad \begin{array}{c}\mathcal{D}_{l''}\\ \vdots\end{array}}{(\forall x\exists y(\uparrow A(y)\otimes\uparrow P(x,y)))^\perp \vdash}$$

Because this proof-attempt can be viewed as a *design* (let us recall from the Appendix that a design can be seen, in a complementary way, in games terms as well as in proof terms), we are allowed to consider the *behaviour* that it generates. Let us call it $\mathbb{C}$. In the sequel, we will try to refine this behaviour, since for the time being its members converge with a lot of designs.
We have to specify the meaning of $A(y)$ and of $P(x,y)$. They will come from the *interactions* they have with other designs.
Incidentally, it may be interesting to notice what would be the behaviour generated by the whole formula S_1:

$\mathcal{E}'$ =

$$\dfrac{\begin{array}{c}\mathcal{D}_{l'}\\ \vdots\end{array} \qquad \dfrac{\downarrow A^\perp(F)\vdash \qquad \downarrow P^\perp(l,F)\vdash}{\vdash\downarrow L^\perp(l), \exists y(\uparrow A(y)\otimes \uparrow P(l,y))} \qquad \begin{array}{c}\mathcal{D}_{l''}\\ \vdots\end{array}}{(\forall x(\uparrow L(x) \multimap \exists y(\uparrow A(y)\otimes\uparrow P(x,y))))^\perp \vdash}$$

We leave to the reader to check that this $\mathcal{E}'$ converges with the following $\mathcal{D}$:

$$\dfrac{\dfrac{\overline{\vdash L(l)}\;\emptyset}{\downarrow L^\perp(l)\vdash} \qquad \dfrac{\overline{\vdash\downarrow A^\perp(F'),P^\perp(l,F')}\;\dagger \qquad \overline{\vdash\downarrow A^\perp(F),P^\perp(l,F)}\;\dagger \quad ...}{\forall y(\uparrow A(y) \multimap \downarrow P^\perp(l,y))^\perp\vdash}}{\vdash \exists x(\uparrow L(x)\otimes \forall y(\uparrow A(y)\multimap\downarrow P^\perp(x,y))}$$

that we may interpret in the following way:

- the opponent is ready to accept any argument from the proponent according to which some language is really an african language and the linguist s/he has chosen speaks it
- but s/he is not ready to discuss the fact that l is a linguist

Going back to the simplified version, we have a convergence of $\mathcal{E}$ simply with:

$$\mathcal{D}' = \cfrac{\cfrac{\cfrac{}{\vdash\downarrow A^{\perp}(F'), P^{\perp}(l, F')}\dagger \quad \cfrac{}{\vdash\downarrow A^{\perp}(F), P^{\perp}(l, F)}\dagger \quad \ldots}{\forall y(\uparrow A(y) \multimap \downarrow P^{\perp}(f, y))^{\perp} \vdash}}{\vdash \exists x(\forall y(\uparrow A(y) \multimap \downarrow P^{\perp}(x, y))}$$

Therefore, $\mathcal{E} \perp \mathcal{D}'$, and any design which is orthogonal to $\mathcal{D}'$ belongs to $\mathbb{C}$. We note that:

- $\mathcal{E}$ is minimal with regards to the justifications for atomic statements: it *stops before* exploring the components A and P. In an interaction as the one seen above, A and P are simply assumed to be true formulae, they could be replaced by the negative true formula $\mathtt{T}$[6]. Moreover, $\mathcal{E}'$ is more explicitly formulated as the bi-orthogonal of (the attempt of proving) $\forall x\ \exists y(\mathtt{T} \otimes \mathtt{T})$.

Then, we may decide to go deeper and to view the statements $A(F) = F$ *is an african language* and $P(l, F) = l$ *speaks* F as themselves formulae which are waiting for justifications. In this case, we use the shift operator $\uparrow$ (cf. the Appendix) in order to separate them in the course of the proof. They may be viewed as mere facts, that is kinds of *propositions*, or *datas* (which happens in the case of the design $\mathcal{E}_0$), or they may also be viewed as more elaborate representations which can be still decomposed (this is the case of the design $\mathcal{E}_1$). In both cases, $\downarrow A(y)$ and $\downarrow P(x, y)$ admit proofs (either trivial as if they were simple axioms, or more elaborate).
Seen as proofs, $\mathcal{E}_1$ and $\mathcal{E}_0$ obviously coincide on the initial steps (starting from the bottom) but differ above $P(l, F)$. Such initial steps are :

$$\cfrac{\cfrac{\begin{matrix}\mathcal{D}_{l'} \\ \vdots\end{matrix} \quad \cfrac{\cfrac{\cfrac{}{\vdash A(F)}}{\downarrow A^{\perp}(F) \vdash} \quad \cfrac{\cfrac{\vdots}{\vdash P(l, F)}}{\downarrow P^{\perp}(l, F) \vdash}}{\vdash \exists y(\uparrow A(y) \otimes \uparrow P(l, y))} \quad \begin{matrix}\mathcal{D}_{l''} \\ \vdots\end{matrix}}{(\forall x(\exists y(\uparrow A(y) \otimes \uparrow P(x, y))))^{\perp} \vdash}}{\vdash S}$$

The attempts to prove $A(F)$ and $P(l, F)$ give rise to two new *behaviours* (the bi-orthogonals of these attempts), that we shall denote respectively : $\mathbb{A}(F)$ and $\mathbb{P}(l, F)$. The designs $\mathcal{E}_0$ and $\mathcal{E}_1$ finally belong to the same behaviour: $\forall x(\exists y(\uparrow\ \mathbb{A}(y) \otimes \uparrow \mathbb{P}(x, y)))$[7], provided that, for instance, $\mathbb{A}(F) = \mathbf{1}$[8] and $\mathbb{P}(l, F) = \mathbf{1} \oplus \{\mathcal{E}_1\}^{\perp\perp}$ (the later is essentially the union of the two behaviours $\mathbf{1}$ and $\{\mathcal{E}_1\}^{\perp\perp}$).
Finally, going back to the complete statement S, its meaning is given by the behaviour: $\mathbb{S}$, which corresponds to the $HSLL$ formula (still denoted by S) :

[6] Let us recall that $\mathtt{T}$ is the neutral element of the negative additive $\&$

[7] Vertical arrows are shift operators - see the Appendix - which make the formulae negative ones, a condition that is necessary if we wish to respect the implicit convention in $HSLL$ according to which formulae are decomposed into maximal blocks of alternate polarities.

[8] The behaviour $\mathbf{1}$ associated with the positive linear constant 1 contains two designs: $\cfrac{}{\vdash \xi}\emptyset$ and $\cfrac{}{\vdash \xi}\dagger$.

$$S = (\forall x \uparrow L(x) \multimap (\exists y \uparrow A(y) \otimes \uparrow P(x,y)))) \oplus \exists y \uparrow A(y) \otimes (\uparrow L(x) \multimap \forall x \uparrow P(x,y))).$$

And therefore

$$\mathbb{S} = (\forall x \uparrow \mathbb{L}(x) \multimap (\exists y \uparrow \mathbb{A}(y) \otimes \uparrow \mathbb{P}(x,y)))) \oplus \exists y \uparrow \mathbb{A}(y) \otimes (\uparrow \mathbb{L}(x) \multimap \forall x \uparrow \mathbb{P}(x,y))).$$

Let us recall a behaviour may be viewed as a set of justifications. Of course, each justification of S may be found in the behaviour $\mathbb{S}$, provided that it is large enough.
Analyzing things more deeply, we discover that in fact, $\mathbb{S}$ may be viewed as a *family of behaviours*. This results from the fact that, if a behaviour, by itself, always contain the designs less defined as any of its designs, it does not contain generally more defined ones. Of course we have always the ability to consider those more defined designs, but each time we refine a design, we get a new behaviour, (which contains the previous ones). Larger behaviours support more justifications than smaller ones.

5 Conclusion

Intuitively speaking, Ludics allows us to develop a new viewpoint, according to which very various objects may be assigned to sentence (and word?) meaning. These objects are technically characterized as *behaviours* that is, as *stable* sets (with regards to bi-orthogonality). We don't need to know what is the *essence* of meaning (by entering into some kind of metaphysics), because those objects are defined by their reactions with regards to other ones they are interacting with. Moreover, we may make the depth of the characterization vary, according to a *separation* theorem. As pointed out by C. Faggian ([Faggian 2006]), only the properties of these objects that can be tested by means of interaction with objects of the same kind can be observed, they are the *observables*.
At the low level of sentence interpretation, there are atoms which are simply facts, or *datas*. They can be replaced by **1**, the neutral element of the $\otimes$ of the algebra of behaviours. In such a *reduction* of meaning, sentences the analysis of which ends up on those atoms are said to be *true*. Actually, a design which does not use the daïmon can be seen as *winning*. By the completeness theorem, that amounts to say that it can be translated into a proof of the statement it argues for, and therefore this statement can be seen as a true one. This provides us with a possible reduction to the *truth-conditional* paradigm.
At a higher level of interpretation, sentence meaning is seen as a *family* of behaviours, parametrized by the elementary behaviours that are involved in the sentence : this is in fact the counterpart of the traditional idea, in Possible Worlds Semantics, of a proposition as a set of possible worlds. Here, a proposition is a behaviour, which is itself defined according to elementary ones associated with atomic sentences. We may for instance define *necessity* as truth for any fine-grained exploration of the behaviours associated with atomic sentences. It is probably a notion of *necessity* slightly different from the usual one: *Necessarily p* does not mean that p is true "by essence", but that p could be defended against any kind of objection. Let us notice that this allows to grasp the difference between (5) and (6) below:

(5) *it is necessary that Bill be the culprit*

(6) *Bill might be the culprit*
For (5) : for any objections formulated against the idea that *Bill be the culprit* (that is, for any fine-grained behaviour associated with that idea), there is a winning design in favour of the idea. For (6), it suffices that there is a winning design against one current argument against the idea.
We conclude from that that the Ludics viewpoint is not totally orthogonal to the classical one, in that it should be possible to express in it most of the distinctions that the classical view is able to draw. Nevertheless, we must emphasize the following points:

- In Ludics, formulae are not the primitive objects. The meaning of a sentence is thus assigned an object which is not *a priori* closed : the meaning of a sentence may be *more and more refined*. In particular, the order between designs given by the separation theorem enables one to explore more and more precisely the argumentative potential of the sentence.

- Ludics is built with an explicit attention given to the ”logical frontier” (what falls inside logic *versus* what falls not). Logical concepts like formulae, proofs and connectives are defined in a world which is larger than the strict logical word (let us remember that we have *paralogisms* like the *daïmon*, and counter-proofs in that world!). This feature may be used to formalize aspects of meaning which don’t directly deal with Logic, like it is the case in pragmatics, dialectic and rhetoric, as seen in the first part of this paper.

- Ludics also refers to the possibility of playing on various interpretations of logical concepts : *localized* vs *delocalized* (or *spiritualist* in Girard’s sense), where we see an implementation of the well known distinction between *tokens* and *types* (see [Strawson 50]). The same sentence can be viewed (inside the same framework) as a token - when seen as an utterance made in a given context - as well as a type - when seen as delocalized and understood independently of any context. Normalization with $\mathcal{F}ax$ makes the communication possible between the two.

- Ludics also enables us to deal with *dynamics* for free, like we saw it in the first part of this paper. This feature is particularly highlighted in the use of c-designs, which include *cuts* and therefore a procedure of normalization which is similar to β-reduction. In this reduction of c-designs, arguments which were initially put at some locations may be displaced in order to play the role they are expected to have in an argumentation, thus ensuring the communication from tokens to types.

Finally, we point out the harmony there is between this conception and the neuro-cognivist views according to which meaning is a question of activation of neuron sets, which are always specific to each mind/brain but are such that transfer is always available from a specific brain to another one. Reproduction of meaning in the dialogue activity is thus made similar with (and perhaps even extends) reproduction of cells in the biological world.

6 Appendix A: a very short presentation of Ludics

6.1 Ludics in a nutshell

Ludics is a recent theory of Logic introduced by J.-Y. Girard in [Girard 01]. Here we don't give the entire definitions of the core concepts of Ludics but we just give an account of the objects we use in this paper.

6.1.1 Designs

At a first glance, *designs* look like *proofs*. In fact, they come from a deep study of proofs and their interaction. It was discovered already during the nineties that proofs of Linear Logic could be decomposed into blocks of opposite polarities (positive like for $\otimes$ and $\oplus$ steps, negative like for $\&$ and $\wp$ steps). That opened the field to a polarized and focalized logic. In such a frame, blocks of a given polarity are reduced to only one step : it is as if a synthetic connective (involving several premisses, not necessarily one or two) was used at each step. It is possible in such a frame to make confrontations between polarized objects : we can think of an attempt to prove a statement vs an attempt to prove the contrary. At the basis of each attempt, there is a sequent, but both bases have opposed polarities, that is one is positive (trying for instance to prove $\vdash P$) and the other negative (trying to prove $\vdash \neg P$ or $P \vdash$).
Because of the multiplicity of premisses, the calculus, using formulae and the usual connectives of Linear Logic is called *Hypersequentialized Linear Logic* ($HSLL$). This calculus contains a large number of rules, even if we may present it by using rule schemata. An overview is given infra.
When opposing two proofs one against the other, it may happen that one of the two be a real proof. In this case, the other one is of course not a proof but what we may call a *counter-proof*. Proofs and counter-proofs together are called *paraproofs*. Amongst paraproofs, there are of course singular objects which are real counter-proofs: they are defeated during the confrontation against a real proof. The prototype of these objects is the one step paraproof :

$$\frac{}{\vdash \Gamma}\dagger$$

where Γ is a sequence of formulae possibly empty, and † is the special positive rule called *Daïmon*. For a proof-searcher, to make this step in a proof amounts to admit his or her failure. In Ludics, this has the meaning *I'm giving up*. It happens that this is the only *paralogism* that Ludics allows.
$HSLL$ may be displayed in a standard way, using only positive formulae : the negative ones are simply put on the left-hand side of the sequent to prove (or to refute). All the sequents which enter the game are therefore of the general form $\Gamma \vdash \Delta$, where Γ and Δ may be empty and Γ contains at most a formula. These sequents are therefore called *forks*, and the negative part (the left-hand side) is called the *handle*. Elements of the right hand side are the *teeth*. If Γ is empty, the fork is said to be positive, if not, it is said to be negative.
Going to Ludics strictly speaking amounts to get rid of formulae in favour of only their adresses, called *loci*. These *loci* are simply sequences of integers (or *bias*). Forks are arrangements of loci, some being positive, others negative.

From now on, if we make exception of †, only two rules are necessary, one for the positive steps, the other for the negative ones. We have therefore :
Definition: *A* design *is a tree of forks* $\Gamma \vdash \Delta$*, built by means of the three following rules :*

- **Daïmon**

$$\frac{}{\vdash \Delta}\,\dagger$$

- **Positive rule**

$$\frac{\cdots \qquad \xi.i \vdash \Delta_i \qquad \cdots}{\vdash \Delta, \xi}\,(\xi, I)$$

where I may be empty and for every indexes $i, j \in I$ ($i \neq j$), Δ_i and Δ_j are disconnected and every Δ_i is included[9] in Δ.

- **Negative rule**

$$\frac{\cdots \qquad \xi.I \vdash \Delta_I \qquad \cdots}{\xi \vdash \Delta}\,(\xi, \mathcal{N})$$

where $\mathcal{N}$ is a possibly empty or infinite set of ramifications such that for all $I \in \mathcal{N}$, Δ_I is included in Δ.

Let us mention that is is usual to interpret the positive rule as a positive choice made by a player : s/he can make a "true" choice, like it is the case when we use the $\oplus$-rule, or s/he can keep several issues simultaneously, like we do when using the $\otimes$-rule. In any case, s/he selects a *locus*, considers it a *focus* (the focus of the action), and s/he selects a ramification, that is a set of adresses on which the focus is distributed.
Similarly, the negative rule is interpreted as a more passive step, since the *focus* is already determined (it is the only locus which occurs on the left-hand side of the negative fork). Moreover, the set associated to that rule is not a specific ramification, but a set of ramifications. In our pragmatic, or rhetorical, view, it is as if the player, after making an assertion (positive step) was waiting for an expected set of answers from his or her co-player. In terms of proofs: the proof makes a choice *and then* predicts the kinds of objections that can be made in the *counter-proof*. If the player wishes to achieve his or her proof, s/he has to continue the design for each branch, each corresponding to a possible refutation. We see here that negative steps are bifurcations in the proof-search.
Because these considerations can be held, a design may be seen also in games terms: each player sees the paraproof s/he is presently building as a a strategy in a game in which the goal could be *avoid the daïmon!*.
In this other view, we see a design as a set of *possible plays*. These plays are called

[9] Every rule where the union of the Δ_i is strictly included in Δ correspond to the weakening rule (respectively for negative rule when Δ_I is strictly included in Δ).

chronicles. A chronicle may be built from a design according to the first view. Starting from the bottom, we record all the branches and their sub-branches. A branch is necessarily a sequence of actions, some are positive and some negative (alternatively). In order to correspond to a true design, these chronicles must satisfy some conditions (coherence, propagation, positivity, totality).

6.1.2 Interaction

Interaction consists in a coincidence of two loci in dual position in the bases of two designs. This creates a dynamics of rewriting of the cut-net made of the designs, called, as usual, *normalisation*. We sum up this process as follows: the cut link is duplicated and propagates over all immediate *subloci* of the initial cut-*locus* as long as the action anchored on the positive fork containing the cut-locus corresponds to one of the actions anchored on the negative one. The process terminates either when the positive action anchored on the positive cut-fork is the *daïmon*, in which case we obtain a design with the same base as the starting cut-net, or when it happens that in fact, no negative action corresponds to the positive one. In the later case, the process fails (or *diverges*). The process may not terminate since designs are not necessarily finite objects.
When the normalization between two designs $\mathcal{D}$ and $\mathcal{E}$ (respectively based on $\vdash \xi$ and $\xi \vdash$) succeeds, the designs are said to be *orthogonal*, and we note: $\mathcal{D} \perp \mathcal{E}$. In this case, normalization ends up on the particular design :

$$\frac{}{\vdash}\ [\dagger]$$

Let $\mathcal{D}$ be a design, $\mathcal{D}^{\perp}$ denotes the set of all its orthogonal designs. It is then possible to compare two designs according to their counter-designs. We set $\mathcal{D} \prec \mathcal{E}$ when $\mathcal{D}^{\perp} \subset \mathcal{E}^{\perp}$.

The separation theorem [Girard 01] ensures that this relation of preorder is an order, so that a design is exactly defined by its orthogonal.

6.1.3 Behaviours

One of the main virtues of this ”deconstruction” is to help us rebuilding Logic.

- Formulae are now some sets of designs. They are exactly those which are closed (or stable) by interaction, that is those which are equal to their *bi-orthogonal*. Technically, they are called *behaviours*.
- The usual connectives of Linear Logic are then recoverable, with the very nice property of *internal completeness*. That is : the bi-closure is useless for all linear connectives. For instance, every design in a behaviour $\mathbf{C} \oplus \mathbf{D}$ may be obtained by taking either a design in $\mathbf{C}$ or a design in $\mathbf{D}$.
- Finally, *proofs* will be now designs satisfiying some properties, in particular that of not using the daïmon rule.

6.2 The c-designs

In [Terui 08] K. Terui proposes an alternative formulation of Ludics which is motivated by stakes of "developing a monistic, logical and interactive theory for computability and complexity". In order to follow such a program, K. Terui modifies and extends the formalism for Ludics.
We focus here on the notions of c-designs and generators that we use in our text and we propose a very simplified presentation of them.

6.2.1 c-Designs

Amongs the new features of the c-designs compared to the original ones of Girard let us underline the followings:

- Instead of objects with absolute address, the c-designs may be described using a term calculus approach. The absolute addresses are replaced by variable binding.
- The c-designs extend ordinary designs in that they contain explicit interactions.

We then focus on some technical modification into the designs building. The c-designs still contains sequences of alternated actions, but we may at first observe that we have a new notion of action. The c-designs are defined according to a signature set $\mathcal{A}$: a set of couples (a, n) where a is a name and n is its arity. And the positive actions are either constants: † (daïmon or abandon) and Ω (divergence or absence of positive rule), or proper and specific actions (denoted by $\overline{a}$ for a given name a) while the negative actions are either variables (x, y, z,$\ldots$) or proper negative actions (denoted by $a(x_1, ..., x_n)$). Secondly, designs contains also cuts which enables to consider applications in a term calculus approach. Let us underline that in such a term calculus , we do not have a unique application but as many applications as elements in a signature set $\mathcal{A}$. Then the terms or c-designs are co-inductivey defined:

- The positive c-designs are: $P = \quad \Omega \quad | \quad \dagger \quad | \quad N_0 || \overline{a} < N_1, \ldots, N_n >$
- The negative c-designs are: $N = \quad x \quad | \quad \Sigma_{a \in A} a(\vec{x}).P_a$

The positive designs really containing a cut are designs $N_0 || \overline{a} < N_1, \ldots, N_n >$ when N_0 is not a variable. In such a case the cut may be seen as an application in the following sense: if N_0 contains a subterm $a(\vec{x}_a).P_a$ then we have to perform the application $(P_a)N_1 \ldots N_n$. Precisely, in such a case $N_0 || \overline{a} < N_1, \ldots, N_n >$ reduces into $P_a[N_1/x_1, \ldots, N_n/x_n]$. Otherwise, if there is no subterm $a(\vec{x}_a).P_a$ in N_0 (or, equivalentely, if N_0 contains the subterm $a(\vec{x}_a).\Omega$) the interaction diverges.
When N_0 is a variable the c-design is said to be a cut-free design.

Let give as an instance of c-design the one corresponding to the $\mathcal{F}ax$. It is a negative c-design recursively defined as follows:

$$Fax_y = \Sigma_{a \in A} a(x_1, \ldots, x_n).(y || \overline{a} < Fax_{x_1}, \ldots, Fax_{x_n} >)$$

6.2.2 Generators

K. Terui introduces in [Terui 08] design generators that provide a means to finitely describe infinite designs.

A **generator** is a triple (S^+, S^-, l) where S^+ and S^- are disjoint sets of states and l is a function defined on $S = S^+ \cup S^-$ satisfying the following conditions:

- For $s^+ \in S^+$, $l(s^+)$ is either Ω, $\dagger$ or an expression of the form $s_0^- || \overline{a} < s_1^-, \dots, s_n^- >$ such that the s_i^-'s belong to S^-.
- For $s^- \in S^-$, $l(s^-)$ is either a variable x, or an expression on the form $\Sigma_{a \in A} a(\vec{x}).s_a^+$ such that the s_a^+'s belong to S^+.

A **pointed generator** is a quadruple (S^+, S^-, l, s_I) where (S^+, S^-, l) is a generator and $s_I \in S$.
We say that (S^+, S^-, l, s_I) generates a c-design called $design(S^+, S^-, l, s_I)$.

A c-design D is **finitely generated** if it is generated by a pointed generator which has finitely many states, and whenever $l(s^-) = \Sigma_{a \in \mathcal{A}} a(\vec{x}).s_a$, all but finitely many s_a have the label Ω.

Examples:
- the pointed generator $(\{s_\dagger\}, \{s\}, l, s_\dagger)$, with: $l(s_\dagger) = \dagger, \quad l(s) = \Sigma_{a \in A}.s_\dagger$ generates the negative daïmon: $\Sigma_{a \in A}.\dagger$.
- the pointed generator$(\{s_a\}_{a \in A}, \{s_N\}, l, s_N)$ with:

$$l(s_N) = \Sigma_{a \in A}(\vec{x}_a).s_a \text{ and } l(s_a) = y || \overline{a} < s_N, \dots, s_N > \text{ if } y \notin \vec{x}_a$$

generates the $\mathcal{F}ax$.
Remark: Provided that $\mathcal{A}$ is finite Dai^- and $\mathcal{F}ax$ are finitely generated.

7 Appendix B: an hypersequentialized linear calculus

We give here a short presentation of a hypersequentialized version of linear calculus, which enables one to manipule the designs as (para)proofs of a logical calculus.

7.1 Formulae and sequents

By means of polarity, we may simplify the calculus by keeping *only positive formulae*. Of course, there are still negative formulae... but they are simply put on the left-hand side after they have been changed into their negation. Moreover, in order to make paraproofs to look like sequences of alternate steps (like it is the case in ordinary games), we will make blocks of positive and of negative formulae in such a way that each one is introduced in only one step, thus necessarily using *synthetic connectives*. Such connectives are still denoted $\oplus$ and $\otimes$ but are of various arities. We will distinguish the case where both $\oplus$ and $\otimes$ are of arity 1 and denote it $\downarrow$.

- The only linear formulae which are considered in such a sequent calculus are built from the set P of linear constants and propositionnal variables according to the following schema :

$$F = P|(F^{\perp} \otimes \cdots \otimes F^{\perp}) \oplus \cdots \oplus (F^{\perp} \otimes \cdots \otimes F^{\perp})| \downarrow F^{\perp}$$

- The sequents are **denoted** $\Gamma \vdash \Delta$ where Δ is a multiset of formulae and Γ contains at most a formula.

7.2 Rules

- There are some axioms (logical and non logical axioms):

$$\overline{P \vdash P} \qquad \overline{\vdash 1} \qquad \overline{\vdash \downarrow T, \Delta} \qquad \overline{\vdash \Delta}^{\dagger}$$

where P is a propositionnal variable ; 1 and T are the usual linear constants (respectively positive and negative).

- The ”logical” rules are the following ones :

Negative rule

$$\frac{\vdash A_{11}, \ldots, A_{1n_1}, \Gamma \quad \ldots \quad \vdash A_{p1}, \ldots, A_{pn_p}, \Gamma}{(A_{11} \otimes \cdots \otimes A_{1n_1}) \oplus \cdots \oplus (A_{p1} \otimes \cdots \otimes A_{pn_p}) \vdash \Gamma}$$

Positive rule

$$\frac{A_{i1} \vdash \Gamma_1 \quad \ldots A_{in_i} \vdash \Gamma_p}{\vdash (A_{11} \otimes \cdots \otimes A_{1n_1}) \oplus \cdots \oplus (A_{p1} \otimes \cdots \otimes A_{pn_p}), \Gamma}$$

where $\cup \Gamma_k \subset \Gamma$ and for $k, l \in \{1, \ldots p\}$ the $\Gamma_k \cap \Gamma_l = \emptyset$.

7.3 Remarks on Shifts

Using the shift is a way to break a block of a given polarity. Separate steps may be enforced by using the *shift* operators $\downarrow$ and $\uparrow$ which change the negative (resp. positive) polarity into the positive (resp. negative) one. The rules introducing such shifted formulae are particular cases of the positive and the negative one:

$$\frac{A^{\perp} \vdash \Gamma}{\vdash \downarrow A, \Gamma}[+] \qquad \frac{\vdash A^{\perp}, \Gamma}{\downarrow A \vdash \Gamma}[-]$$

where A is a negative formula.
Example In a block like $A \otimes B \otimes C$ in principle, A, B and C are negative, but if we

don't want to deal with A, B, C simultaneously, we may change the polarity of $B \otimes C$ (which is positive) and make it negative by means of $\uparrow$. We write then $A \otimes \uparrow (B \otimes C)$. Compare the two following partial proofs, where (1) does not use any shifts and (2) uses one :

instead of (1): $\dfrac{A^{\perp} \vdash \quad B^{\perp} \vdash \quad C^{\perp} \vdash}{\vdash A \otimes B \otimes C}$ we get (2) : $\dfrac{A^{\perp} \vdash \quad \dfrac{\dfrac{B^{\perp} \vdash \quad C^{\perp} \vdash}{\vdash B \otimes C}}{\downarrow (B \otimes C)^{\perp} \vdash}}{\vdash A \otimes \uparrow (B \otimes C)}$

References

[Andréoli 92] J.-M. Andréoli *Logic Programming with Focusing Proofs in Linear Logic*, The Journal of Logic and Computation, 2, 3, pp. 297-347, 1992,

[Curien 2004] Pierre-Louis Curien *Introduction to linear logic and ludics, part I and II*, to appear, downloadable from http://www.pps.jussieu.fr/ curien/LL-ludintroI.pdf,

[Ducrot 1984] O. Ducrot, *Le dire et le dit*, Editions de Minuit, Paris, 1984.

[Faggian 2006] C. Faggian, *Ludics and interactive observability : te geometry of tests* Theoretical Computer Science, vol 350, n^{o}2, pp 213-233, 2006

[Fleury-Quatrini 04] M.-R. Fleury, M. Quatrini *First order in Ludics* Mathematical Structures in Computer Science 14,no 2, 189–213,

[Girard 99] J.-Y. Girard *On the Meaning of Logical Rules-I* in *Computational Logic*, U. Berger and H. Schwichtenberg eds. Springer-Verlag, 1999,

[Girard 01] J.-Y. Girard *Locus Solum* Mathematical Structures in Computer Science 11, pp. 301-506, 2001,

[Girard 03] J.-Y. Girard *From Foundations to Ludics* Bulletin of Symbolic Logic 09, pp. 131-168, 2003,

[Girard 06] J.-Y. Girard *Le Point Aveugle*, vol. I , II, Hermann, Paris, 2006,

[Hamblin 70] C.-L. Hamblin *Fallacies*, Vale Press, Newport News, republished in 2004

[Hintikka-Kulas 83] J. Hintikka, J. Kulas *The Game of Language: Studies in Game Theoretical Semantics and its Applications*, D. Reidel, 1983,

[Hintikka-Sandu 97] J. Hintikka, G. Sandu *Game Theoretical Semantics*, in J. Van Benthem and A. ter Meulen *Handbook of Logic and Language*, chap. 6, Elsevier, 1997,

[Martin-Löf 84] Per Martin-Löf *Intuitionistic Type Theory*, Bibliopolis, Naples, 1984, [Pietarinen 07] A-V. Pietarinen *Game Theory and Linguistic Meaning*, Elsevier, 2007,

[Ranta 94] A. Ranta *Type-Theoretical Grammar* Oxford University Press, 1994,

[Schopenhauer] A. Schopenhauer, *The Art of Always Being Right*.

[Strawson 50] P. F. Strawson, *On Referring*, Mind, vol. 59, 1959.

[Sundholm 86] G. Sundholm *Proof Theory and Meaning* in D. Gabbay and F. Guenthner eds. *Handbook of Philosophical Logic*, Vol III, pp. 471-506, D. Reidel, 1986,

[Terui 08] K.Terui, *Computational Ludics*, to appear in Theoretical Computer Science, J.-Y.Girard's Festschrift special issue, 2008.

[Tronçon 06] S. Tronçon *Dynamique des démonstrations et théorie de l'interaction*, PhD thesis, Université d'Aix-Marseille, 2006,

[Wittgenstein 53] Ludwig Wittgenstein, *Philosophische Untersuchungen*, Blackwell, 1953.