Ludics and Natural Language: First Approaches

Christophe Fouqueré, Myriam Quatrini

Ludics and Natural Language: First Approaches

Christophe FOUQUERÉ[1] and Myriam QUATRINI[2]

[1] LIPN, Université Paris 13 and CNRS, christophe.fouquere@lipn.univ-paris13.fr
[2] IML, Université d'Aix-Marseille and CNRS, quatrini@iml.univ-mrs.fr

Abstract. Ludics is a rebuilding of Linear Logic from the sole concept of interaction on objects called designs, that abstract proofs. Works have been done these last years to reconsider the formalization of Natural Language: a dialogue may be viewed as an interaction between such abstractions of proofs. We give a few examples taken from dialogue modeling but also from semantics or speech acts to support this approach. [3]

1 Introduction

Ludics is a rebuilding of Linear Logic from the sole concept of interaction on objects called designs that abstract proofs. Works have been done these last years to reconsider the formalization of Natural Language: a dialogue may be viewed as an interaction between such abstractions of proofs. Among the domains that have been explored, one may cite formal semantics [1] and pragmatism: speech acts [2] and figures of dialogue [3]. The approach follows what have been done initially for dialogues [4] where dialogues are modeled in proof theory using Ludics.

Ludics [5] is a logical framework developed by J.-Y. Girard around 2000. The motto underlying Ludics is: *interaction is a central concept in logic.* This motto may be considered as following the fruitful paradigm between logic and computer science, namely the Curry-Howard isomorphism. Recall that this isomorphism establishes a perfect correspondence between programs and their execution on one side, and formal proofs and cut elimination on the other side. A cut between conclusions of proofs enables an interaction between these proofs, when one of the formulas is the conclusion of one proof, whereas the other is a hypothesis of the other proof. Called modus ponens, it is the main ingredient of reasoning; one of the main results of proof theory states that it is always possible to *normalize* a proof, *i.e.* to transform a proof with cuts to an equivalent proof without cuts. The dynamic of computation may therefore be considered as the heart of logic.

After decades of work studying the properties of this dynamic and expanding its scope of relevance, Ludics reverses priorities of concepts. Traditionally, formulas and proofs are first set, the cut being one of the rules used in the definition of what is a correct proof. The cut elimination procedure completes the picture by adding a dynamic. Then, progressively, to ensure dynamic properties, formulas and proofs have been refined. Linear Logic [6] illustrates these changes: conjunction and disjunction connectives are each replaced by two versions, additive and multiplicative, and the framework of proofnets gives a geometrical presentation to proofs where the cut elimination property still appears as a property given a posteriori, *i.e.* as a reduction of such graphs. A contrario, in Ludics, cut, *i.e.* interaction, is a primitive concept. Neither formulas nor proofs are considered primitive but *designs*, whose sole purpose is that they carry on the interaction. In fact, interaction takes place between two designs as a step-by-step travel through two dual paths, one path in each

[3] This work was completed with the support of the ANR project LOCI number 0212 02

design. A design is then nothing else but a set of potential paths where interaction may take place. Moreover, a design is essentially defined by its counter-designs: those with whom it interacts.

The concrete basic steps of interaction (called *actions* in Ludics) correspond to the basic steps of cut elimination. They allow for the exploration of a design as one may explore a formula through its main connective to its subformulas, and so on. However, a design abstracts from the notion of formula: a design has only loci, *i.e.* adresses, where interaction goes through. When a set of designs is given, the space where interaction with these designs may take place is also given. This space is defined by all the counter-designs of each design of this set. This gives an external viewpoint on designs, from which one may observe regularities. Formulas may then be retrieved as sets of designs closed relatively to these counter-designs. Indeed, exploring a set of designs reduces to exploring each design of this set. When it is closed, a set of designs describes all the ways to explore the object it represents, until its undecomposable elements. This has to be related to the concept of a formula, defined inductively by its connectives and subformulas that compose it until obtaining propositional variables or constants. Moreover, and this is an essential property of Ludics, some designs associated with a formula may be proofs (of this formula): those designs that satisfy suitable properties, among which, precisely, the fact that the exploration can always continue until a suitable term. So, truthness of a formula requires the existence of some proof belonging to the set of designs associated to this formula.

The formalization of Natural Language uses these "primitive" objects of the interaction, upstream of a reconstruction of logic. The fact that interaction is the fundamental concept of Ludics justifies its relevance to the study of dialogues, hence also to Natural Language. Remark that Ludics has also been used in the same spirit for formalizing web processes [7].

Section 2 is devoted to a presentation of Ludics. We describe first what is an action and what conditions should satisfy a tree of actions to be a design. We give the definition of interaction between nets of designs and give the main steps for rebuilding logic. In particular, we recall the main theorems that Ludics satisfies, *i.e.* internal completeness. Then we present a sequent calculus that mimics the structure of designs. Finally, we summarize works that have been done to extend Ludics.

Section 3 is an introduction to the formalization of dialogues in Ludics. We consider short dialogues to show how Ludics may be used. In particular, we express how to represent a divergent dialogue, and how to represent presuppositions.

Works concerning the modeling of meaning are presented in section 4. Contrarily to standard semantics, the meaning of an utterance is given by the way this utterance may be justified, explored in dialogues. Hence a set of designs may be associated with an utterance. We show with one example that classical semantics may be retreived thanks to the reconstruction of logic available with Ludics. Finally we present a few schemas that may be used for representing simple speech acts.

2 Ludics: basic Notions

In this section we present the basic concepts of Ludics. The reader may find thorough presentations of Ludics obviously in [5] but also in [8,9,10,11].

2.1 Actions, Designs, Interaction: an informal Description

In Ludics, primitive elements of interaction are **actions**. Actions are polarized and appear as dual pairs: for each positive action (resp. negative) κ, there exists a dual negative action (resp. positive)

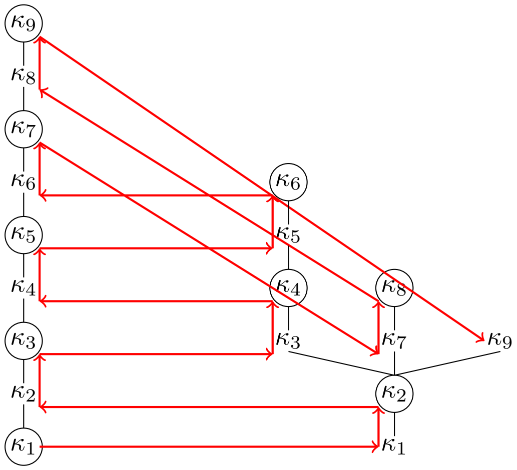
$\overline{\kappa}$ and $\overline{\overline{\kappa}} = \kappa$. As in game theory, a **justification** relation between actions is given. Actions are organized in alternate sequences called **chronicles** in Ludics. These sequences should satisfy the following conditions: (i) a positive action can be either *initial* or *justified* by a negative action that precedes it in the sequence, (ii) a negative action, except the first which may be initial, is justified by the positive action immediately preceding. Following the metaphor of games, these alternate sequences of actions can be seen as *plays* that can be grouped to form *strategies* named designs in Ludics [12]. Thus, **designs** are sets of chronicles that may be interleaved to give rise to what is travelled during an interaction. Not all sets of chronicles can be designs: in particular a design should be organized as a forest, with only one root when the root is a positive action.

An **interaction** occurs between two designs when each of them contains a path dual of the other path (*e.g.* figure 1 on the left). It is a travel through these two designs, which runs as follows:
- It starts with the object that contains as root a positive action (hence unique).
- Every time it goes through a positive action $\textcircled{$\kappa$}$ of one of the two designs, the travel continues in the other design on the negative action that is dual κ (when it exists), then continues on the only positive action that follows this negative action in the same design.

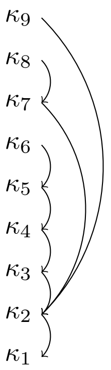
This process continues until positive actions that are visited have a dual negative action in the other design. If such a negative action is not found, the process *diverges*, *i.e.* fails.

The **trace** of an interaction is given by the sequence of pairs of dual actions followed during this interaction (*e.g.* figure 1 on the right). Conversely, it is possible, from the trace of an interaction, to recover the minimal designs that led to this trace.

Example 1. In figure on the left, circled actions are positive. In figure on the right, the name of the action is noted in place of the pair of dual actions for easiness of reading, an arrow from action κ to action κ' denotes that κ is justified by κ'.



Interaction between two designs



Trace of the interaction with justification between actions.

2.2 Actions, Designs, Interaction: a formal Presentation

The definition of actions in Ludics is precisely defined by means of the concept of *locus* or *address*. This concept plays a very special rôle in Ludics as their intended use consists in replacing logical formulas as designs should replace proofs.

Definition 1 (Action). *A* proper action κ *is a triple* (ϵ, ξ, I) *where:*
- $\epsilon \in \{+, -\}$ *is the* **polarity** *of* κ,
- *the finite sequence of integers* ξ *is the* **focus** *(the address) of* κ,
- *the finite set of integers* I *is the* **ramification** *of* κ.
Besides proper actions, there exists also a special positive action, the **daïmon**, *noted* †.

In the following, we note $\xi.i$ the sequence of integers ξ followed by the integer i. We note also $\overline{\epsilon}$ the dual polarity of ϵ, *e.g.* if $\epsilon = +$ then $\overline{\epsilon} = -$. Finally, if $I = \{i_1, i_2, \ldots, i_n\}$, then $\xi.I$ is the set of sequences of integers $\{\xi.i_1, \xi.i_2, \ldots, \xi.i_n\}$. The ramification of an action determines the loci that may be used as continuations during an interaction. Relations of *justification* and *duality* follow from the previous definition. An action $(\epsilon, \xi.i, J)$ is justified by the action $(\overline{\epsilon}, \xi, I)$ when $i \in I$. The dual action of the action (ϵ, ξ, I) is the action $(\overline{\epsilon}, \xi, I)$.

The main objects of Ludics are the designs, *i.e.* sets of *chronicles* satisfying specific conditions. A chronicle is a sequence of actions that satisfies also conditions to ensure in particular justification and linearity.

Definition 2 (Chronicle). *A* **chronicle** $\mathfrak{c}$ *is a non-empty, finite, alternate sequence of actions such that*

- Proper positive action: *A proper positive actionis either justified,* i.e. *its focus is built from one of the previous actions in the sequence, or is called an initial action.*
- Negative action: *A negative action may be initial, in that case it is the first action of the chronicle. Otherwise, this action is justified by the positive action that precedes it immediately.*
- Linearity: *Actions have distinct foci.*
- Daïmon: *If the action daïmon is present, this action is the last one of the chronicle.*

Definition 3 (Coherence between chronicles). *Two chronicles* $\mathfrak{c}_1$ *and* $\mathfrak{c}_2$ *are* **coherent**, i.e. $\mathfrak{c}_1 \frown \mathfrak{c}_2$, *when the following conditions are satisfied:*

- Comparability: *Either one chronicle extends the other, or they differ first on negative actions,* i.e. *if* $w\kappa_1 \frown w\kappa_2$ *then either* $\kappa_1 = \kappa_2$ *or* κ_1 *and* κ_2 *are negative actions.*
- Propagation: *If the two chronicles diverge on negative actions with distinct foci, then ulterior actions in one chronicle are distinct from ulterior actions in the other*i.e. *if* $w(-, \xi_1, I_1)w_1\sigma_1 \frown w(-, \xi_2, I_2)w_2\sigma_2$ *with* $\xi_1 \neq \xi_2$ *then* σ_1 *and* σ_2 *have distinct foci.*

A *base* is associated to each chronicle. A base is a sequent of loci denoted $\Gamma \vdash \Delta$ such that Δ is a finite set of loci and Γ contains at most one locus. Furthermore, loci of $\Gamma \cup \Delta$ are pairwise disjoint, *i.e.* there does not exist a locus that is a sub-locus of another one. If Γ is empty, the base is *positive*, otherwise it is *negative*. A chronicle $\mathfrak{c}$ has a **base** $\Gamma \vdash \Delta$ as soon as Γ is either empty and the locus in Γ is the focus of the first (negative) action of the chronicle, or Γ is empty and the first action of $\mathfrak{c}$ is positive and Δ contains all the foci of initial positive actions of the chronicle.

Definition 4 (Design, Net).

- *A* **design** $\mathfrak{D}$, *of base* $\Gamma \vdash \Delta$, *is a set of chronicles of base* $\Gamma \vdash \Delta$, *such that the following conditions are satisfied:*
  - Forest: *The set is prefix-closed.*
  - Coherence: *The set is a clique of chronicles,* i.e. $\forall \mathfrak{c}_1, \mathfrak{c}_2,\ \mathfrak{c}_1 \frown \mathfrak{c}_2$.

- Positivity: *A chronicle without extension in $\mathfrak{D}$ ends by a positive action.*
  - Totality: *$\mathfrak{D}$ is not empty when the base is positive, in that case each chronicle begins with a (unique) positive action.*
- *A* **net** *is a finite set of designs of disjoint bases.*

Designs may also be presented in a sequent-like style: it is a tree whose nodes are sequents made of loci in place of formulas. An action specifies the relation between a node and its daughters: loci that are present in nodes are subloci of ancestors in the tree. The root of the tree contains initial loci that may be used in an interaction. As the framework is linear, a locus that is initial cannot be a sublocus of other initial loci.

Definition 5 (Designs in a sequent-style).

- *A* **sequent** $\Gamma \vdash \Delta$ *is given with two finite sets of addresses or loci Γ and Δ,* i.e. *finite sequences of integers, such that Γ contains at most one element and there is no address of $\Gamma \cup \Delta$ is an initial sequence of another one.*
- *A* **sequent-design** *(simply called design), with base the sequent $\Gamma \vdash \Delta$ is a tree of sequents built from the three following rules:*
  - Daimon

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

  - Positive rule

$$\frac{\ldots \quad \xi.i \vdash \Delta_i \quad \ldots}{\vdash \Delta, \xi}\ (+, \xi, I)$$

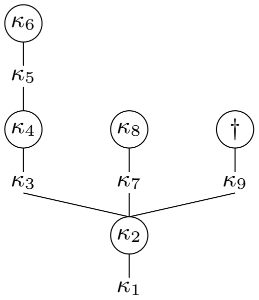
for $i \in I$, Δ_i are pairwise disjoint and included in Δ.

  - Negative rule

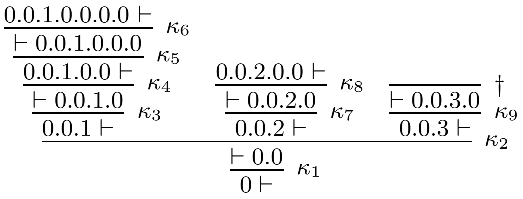
$$\frac{\ldots \quad \vdash \xi.I, \Delta_I \quad \ldots}{\xi \vdash \Delta}\ (-, \xi, \mathfrak{R})$$

$\mathfrak{R}$ is a set (that may be empty or infinite) of ramifications. For all $I \in \mathfrak{R}$, Δ_I, non necessarily disjoint, are included in Δ.

Example 2 (example 1 ctd).



a design



in a sequent-style

We provide in subsection 2.3 a brief presentation of a sequent calculus related to Ludics. We already may give intuitions to interpret logically a design. In a bottom-up reading, a design corresponds to a proof search. Ti recover a proof in Linear Logic, one tries to substitute formulas to addresses. The action specifies the principle generalized connective of the formula, combining connectives $\otimes$ and $\oplus$ for a positive action, or connectives $⅋$ and & for a negative action. Subformulas are found by iterating this operation on premisses of the rule. This process yields either to specific cases[4], without using the daïmon, then the result is an explicit proof of a formula.

Example 3.

- The simplest design consists in a unique positive action, the daimon. It is noted $\mathfrak{Dai}_+$: $\overline{\vdash \ \Gamma}^{\ \dagger}$. It is a positive design. We note later that it interacts with all negative designs of the same base: the action daïmon ends an interaction.
- The fax noted $\mathcal{F}ax_{\xi,\xi'}$ is a design recursively defined in the following way:

$$\dfrac{\dots \qquad \dfrac{\dots \quad \overset{\mathcal{F}ax_{\xi'.i,\xi.i}}{\xi'.i \vdash \xi.i} \quad \dots}{\vdash \xi.I,\xi'}\,(+,\xi',I) \qquad \dots}{\xi \vdash \xi'}\,(-,\xi,\mathcal{P}_f(\mathbb{N}))$$

  The set of finite sets of integers is noted $\mathcal{P}_f(\mathbb{N})$. We note later that it allows for delocalizing a design from a locus ξ to another locus ξ'.

Interaction, *i.e.* cut elimination, is the normalisation process os specific nets of designs called *cut nets*. In a cut net, addresses that appear in the bases of designs are either distinct, or present once in the positive part of a base and once in the negative part of a base, such pairs are the *cuts* of the net. Hence, in a cut net, the graph made of bases and cuts is acyclic and connex. We give below the definition of interaction in case of a *closed* cut net, *i.e.* addresses in bases are part of cuts. The reader may find a definition of interaction for general cut nets in [5]. Remark that in a closed cut net, there must exist a positive design, *i.e.* a design whose base has an empty left part. Such a design is called a *principal* design.

Definition 6 (Interaction for closed cut nets). *Let $\mathfrak{R}$ be a closed cut net, the result of the interaction is a design noted $[\![\mathfrak{R}]\!]$ and defined in the following way: let $\mathfrak{D}$ be the principal design of $\mathfrak{R}$, with first action κ,*

- *if κ is a daïmon, then $[\![\mathfrak{R}]\!] = \mathfrak{Dai}_+$,*
- *otherwise κ is a proper positive action $(+, \sigma, I)$ such that σ is part of a cut with another design, whose first action is $(-, \sigma, \mathcal{N})$ (where $\mathcal{N}$ aggregates the ramifications of the negative actions of same focus σ):*
  - *If $I \notin \mathcal{N}$, then interaction fails (diverges).*
  - *Otherwise, interaction continues with the connex part of the subdesigns we get with I and the remainder of $\mathfrak{R}$.*

Hence either an interaction of aclosed cut net diverges, or it does not end, or the result is the design $\mathfrak{Dai}_+$.

Example 4.

[4] empty sets of ramifications, empty ramifications, recursive configurations, ...

- Consider the design $\mathfrak{D}$ of base $\vdash \xi$ and with unique action $(+, \dagger)$, and $\mathfrak{E}$ of base $\xi \vdash$, then $[\![\mathfrak{D}, \mathfrak{E}]\!] = \mathfrak{D}$.
- Let $\mathfrak{D}$ be a design of base $\vdash \xi$ then normalization with the fax $\mathcal{F}ax_{\xi,\xi'}$ is the design $\mathfrak{D}'$ we get from $\mathfrak{D}$ by substituting ξ with ξ' in each chronicle of $\mathfrak{D}$. For example, if we consider that the first action of $\mathfrak{D}$ is $(+, \xi, I)$, the interaction goes as follows:

$$\dfrac{\dfrac{\overset{\mathfrak{D}_1}{\xi.1 \vdash} \quad \ldots \quad \overset{\mathfrak{D}_n}{\xi.n \vdash}}{\vdash \xi}\,(+,\xi,I) \qquad \dfrac{\ldots \quad \dfrac{\ldots \quad \overset{\mathcal{F}ax_{\xi'.i,\xi.i}}{\xi'.i \vdash \xi.i} \quad \ldots}{\vdash \xi.I,\xi'}\,(+,\xi',I) \quad \ldots}{\xi \vdash \xi'}\,(-,\xi,\mathcal{P}_f(\mathbb{N}))}{}$$

After two steps, we get:

$$\dfrac{\overset{[\![\mathfrak{D}_1,\mathcal{F}ax_{\xi'_1,\xi_1}]\!]}{\xi'.1 \vdash} \qquad \ldots \qquad \overset{[\![\mathfrak{D}_n,\mathcal{F}ax_{\xi'_n,\xi_n}]\!]}{\xi'.n \vdash}}{\vdash \xi'}\,(+,\xi',I)$$

The process follows in a recursive manner.

Orthogonality is defined in the following way:

Definition 7 (Orthogonal, Behaviour).

- *Let $\mathfrak{D}$ be a design of base $\xi \vdash \sigma_1, \ldots, \sigma_n$ (resp. $\vdash \sigma_1, \ldots, \sigma_n$), the net of designs $\mathfrak{R} = (\mathfrak{A}, \mathfrak{B}_1, \ldots, \mathfrak{B}_n)$ (resp. $\mathfrak{R} = (\mathfrak{B}_1, \ldots, \mathfrak{B}_n)$), where $\mathfrak{A}$ has the base $\vdash \xi$ and $\mathfrak{B}_i$ has the base $\sigma_i \vdash$, belongs to the orthogonal of $\mathfrak{D}$ noted $\mathfrak{D}^\perp$ if $[\![\mathfrak{D}, \mathfrak{R}]\!] = \mathfrak{Dai}$.*
- *Let E be a set of designs of same base, $\mathrm{E}^\perp = \bigcap_{\mathfrak{D} \in \mathrm{E}} \mathfrak{D}^\perp$.*
- E *is a* behaviour *if* $\mathrm{E} = \mathrm{E}^{\perp\perp}$. *A behaviour is* positive *(resp.* negative*) if the base of its designs is positive (resp. negative).*

Among remarkable facts of Ludics, let us note that designs are completely defined by their interactions, as affirmed by the *separation theorem*: two designs $\mathfrak{D}$ and $\mathfrak{D}'$ of same base are different iff there exists a design $\mathfrak{E}$ such that the results of interaction $[\![\mathfrak{D}, \mathfrak{E}]\!]$ and $[\![\mathfrak{D}', \mathfrak{E}]\!]$ differ. Other main properties are the following ones [5]:

Theorem 1. *Normalization is stable by intersection, is associative and monotonous with respect to the order induced by duals:*

- *If K is not empty and for all $k \in K$, $\mathfrak{R}_k \subset \mathfrak{R}$ then $[\![\bigcap_{k \in K} \mathfrak{R}_k]\!] = \bigcap_{k \in K} [\![\mathfrak{R}_k]\!]$.*
- *Let $\{\mathfrak{R}_0, \ldots, \mathfrak{R}_n\}$ be a net of nets, $[\![\mathfrak{R}_0 \cup \cdots \cup \mathfrak{R}_n]\!] = [\![[\![\mathfrak{R}_0]\!], \ldots, [\![\mathfrak{R}_n]\!]]\!]$.*
- *If $\mathfrak{D}_0^\perp \subset \mathfrak{E}_0^\perp, \ldots, \mathfrak{D}_n^\perp \subset \mathfrak{E}_n^\perp$ then $[\![\mathfrak{D}_0, \ldots, \mathfrak{D}_n]\!]^\perp \subset [\![\mathfrak{E}_0, \ldots, \mathfrak{E}_n]\!]^\perp$*

Operations on behaviours are defined hereafter. Two behaviours are considered *disjoint* when the sets of ramifications of their first action are disjoint. Two designs, two behaviours of same polarity are *alien* when the intersection of the unions of the ramifications of their first actions is empty. Finally two positive behaviours $\mathbf{G}$ and $\mathbf{H}$ are *independent* when, if I and I' are two ramifications of first actions of designs in $\mathbf{G}$ (resp. J and J' in $\mathbf{H}$) such that $I \cup J = I' \cup J'$ then $I = I'$ and $J = J'$.

Definition 8.

- *Let $\mathbf{G}_k$ be a family of behaviours of positive base pairwise disjoint, $\bigoplus_k \mathbf{G}_k = (\bigcup_k \mathbf{G}_k)^{\perp\perp}$*

- *Let* $\mathbf{G}_k$ *be a family of behaviours pairwise disjoint,* $\&_k \mathbf{G}_k = \bigcap_k \mathbf{G}_k$
- *Let* $\mathfrak{A}$ *and* $\mathfrak{B}$ *be two positive alien designs, one defines* $\mathfrak{A} \otimes \mathfrak{B}$ *in the following way: if* $\mathfrak{A}$ *or* $\mathfrak{B}$ *is* $\mathfrak{Dai}$*, then* $\mathfrak{A} \otimes \mathfrak{B} = \mathfrak{Dai}$*. Otherwise* $\mathfrak{A}$ *and* $\mathfrak{B}$ *have as first action respectively* $(+, \langle\rangle, I)$ *and* $(+, \langle\rangle, J)$*, let us define* $\mathfrak{A}'$ *by replacing in the chronicles of* $\mathfrak{A}$ *the first action* $(+, \langle\rangle, I)$ *by* $(+, \langle\rangle, I \cup J)$*, one defines in the same way* $\mathfrak{B}'$*, then* $\mathfrak{A} \otimes \mathfrak{B} = \mathfrak{A}' \cup \mathfrak{B}'$
- *Let* $\mathbf{G}$ *and* $\mathbf{H}$ *be two positive alien behaviours,* $\mathbf{G} \otimes \mathbf{H} = \{\mathfrak{A} \otimes \mathfrak{B} \ ; \ \mathfrak{A} \in \mathbf{G}, \mathfrak{B} \in \mathbf{H}\}^{\perp\perp}$
- *Let* $\mathbf{G}$ *and* $\mathbf{H}$ *be two negative alien behaviours,* $\mathbf{G} \parr \mathbf{H} = (\mathbf{H}^{\perp} \otimes \mathbf{G}^{\perp})^{\perp}$

Theorem 2.

- *(internal additive completeness) Let* $K \neq \emptyset$*,* $\bigoplus_{k \in K} \mathbf{G}_k = \bigcup_{k \in K} \mathbf{G}_k$
- *A behaviour of positive base (resp. negative base) is always decomposable as a* $\bigoplus$ *(resp. a* $\&$*) of connected behaviours.*
- *(adjunction) Let* $\mathfrak{F}$*,* $\mathfrak{A}$*,* $\mathfrak{B}$ *be three designs,* $\mathfrak{F}$ *negative,* $\mathfrak{A}$ *and* $\mathfrak{B}$ *positive, there exists a unique negative design* $(\mathfrak{F})\mathfrak{A}$ *(that does not depend on* $\mathfrak{B}$*) such that* $[\![\mathfrak{F}, \mathfrak{A} \otimes \mathfrak{B}]\!] = [\![(\mathfrak{F})\mathfrak{A}, \mathfrak{B}]\!]$*.*
- *(internal multiplicative completeness) Let* $\mathbf{G}$ *and* $\mathbf{H}$ *be two independent positive behaviours, then* $\mathbf{G} \otimes \mathbf{H} = \{\mathfrak{A} \otimes \mathfrak{B} \ ; \ \mathfrak{A} \in \mathbf{G}, \mathfrak{B} \in \mathbf{H}\}$

Theorem 3 (Full soundness and completeness). *Ludics is fully sound and complete with respect to second-order multiplicative-additive Linear Logic.*

2.3 Back to a Sequent Calculus

Hypersequentialized calculi have been proposed for various fragments or the full Linear Logic (*e.g.* [13,14,1]). Designs may there be considered as (para)proofs of such calculi. The key ingredient to specify what kind of structure should be given to sequents comes from works on focalization in logical programming: Andreoli states in [15] that a proof in Linear Logic can be organized in such a way that decomposition of clustered deterministic (say negative, *i.e.* $\parr$ and &) and non-deterministic (say positive, *i.e.* $\otimes$ and $\oplus$) connectives in a formula alternate: designs may then be viewed as abstracting such concrete and focalized proofs, and taking into account infinity, *e.g.* η-expansion, and failures, *i.e.* at most one of two dual formulas has a proof, represented in designs by the possibility of a daimon. Furthermore, although every negative formulas[5] may be concurrently and immediately decomposed, one positive formula may be chosen when a positive step occurs. Such a positive formula is the focus of the application of the rule. Following focalization, we can consider that a sequent contains at most one negative formula and we put its dual on the left part of the sequent (using implicitly De Morgan rules). Hence the right part of a sequent contains only positive formulas. A decomposition step, *i.e.* a bottom-up application of a rule in a sequent calculus, consists in

- either choosing a positive formula to be decomposed, to give rise to a set of (negative) subformulas, hence a set of sequents (one for each negative subformula),
- or decomposing the negative formula to give rise to a set of sets of (positive) subformulas, hence a set of sequents (one for each set of positive subformulas).

The calculus contains then three rules: one for the axiom, and one for the decomposition of each polarity, using synthetic connectives of various arity. A cut rule may be added that does not change

[5] A negative formula has a negative connective as the main one.

the calculus. Positive formulas which are considered in such a sequent calculus are built from a set X of positive atoms according to the following schema:

$$F := X \mid (F^\perp \otimes \cdots \otimes F^\perp) \oplus \cdots \oplus (F^\perp \otimes \cdots \otimes F^\perp)$$

The rules are the following ones:

Axiom rule $\dfrac{}{x \vdash x, \Gamma}$ where $x \in X$. **Cut rule** $\dfrac{B \vdash A, \Gamma \quad A \vdash \Delta}{B \vdash \Gamma, \Delta}$

Negative rule

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

Positive rule

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

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

The complete decomposition of connectives of the same polarity may be broken by using *shift* connectives $\downarrow$ and $\uparrow$, that correspond to unary versions of the previous connectives. Let $\downarrow$ (resp. $\uparrow$) change the negative (resp. positive) polarity into the positive (resp. negative) one. Then two following rules may be added to the system:

Shift rules $\dfrac{A \vdash \Gamma}{\vdash \downarrow A^\perp, \Gamma}$ $\qquad$ $\dfrac{\vdash A, \Gamma}{\downarrow A^\perp \vdash \Gamma}$

Example 5. Let A, B and C be negative formulas, formulas $A \otimes B \otimes C$ and $A {\otimes}{\uparrow}(B \otimes C)$ are positive. Their decomposition using the previous rules are the following ones:

$$\frac{A \vdash \quad B \vdash \quad C \vdash}{\vdash A \otimes B \otimes C} \qquad \frac{A \vdash \quad \dfrac{\dfrac{B \vdash \quad C \vdash}{\vdash B \otimes C}}{\downarrow (B \otimes C)^\perp \vdash}}{A {\otimes}{\uparrow}(B \otimes C)}$$

2.4 Extensions of Ludics

Works have been done to extend Ludics beyond the multiplicative-additive fragment of Linear Logic. These works are of particular interest as linearity or sequentiality is no more a constraint. In current use of Ludics in Natural Language, this has not been considered. However, it may be useful in some cases.

First, *Ludics nets* were developed as a game model for concurrent interaction by F. Maurel and C. Faggian [16] and more thoroughly analyzed by P.-L. Curien and C. Faggian in [13]: in some way, Ludics nets are abstract proof-nets, hence graphs, in the same way as designs are abstract proof trees.

K. Terui proposed in [17] a reformulation of Ludics more suitable from a computational point of view. *c-designs* are built as Girard's designs except that (i) variables may occur as generic addresses (possibly in an infinite number), (ii) positive nodes may be explicit internal cuts or serve to model divergence of interaction, (iii) c-designs may not be linear. The syntax of c-designs is such that they may be more easily used for programming purposes, and a grammar allows for a finite specification of them by means of generators. K. Terui uses this model for characterizing word languages where the acceptance relation between a word and a grammar is expressed w.r. to orthogonality between c-designs. Its main results relate automata and c-designs:
- Girard's finite designs recognize exactly regular languages.

- Cuts are necessary (hence the use of c-designs) to recover the full expressive power of Turing machines.

M. Basaldella and C. Faggian in [12] extended Ludics to be able to deal with exponentials. Contrarily to the works of Terui, they only extend the language with *neutral* actions to represent exponentials. They obtain a full completeness result with respect to an hypersequentialized calculus for Linear Logic.

3 Dialogues and argumentation

Ludics has been used in a series of papers for representing several aspects of Natural Language. This section is devoted to the representation of dialogues (*e.g.* [4,3]). Shortly speaking, a dialogue may be considered as the trace of the interaction between two designs, one for each locutor. In elementary cases, an utterance of a locutor is a positive action in her design that has to be present as a negative action in her interlocutor for the dialogue to continue[6]. Generally, an utterance is not as simple as a unique action: an utterance may be a complex sentence including several propositions, it may make use of presupposition, . . . , or it may correspond to a misunderstanding or an end of the dialogue. The formalization of dialogues in Ludics considers that an intervention in a dialogue conveys one or several **dialogue acts**, that Landragin [18] defines as "the minimal unit of communication in a dialogical context". A dialogue act is a communicational fact whose role is to fuel the dynamic and determine the shape of the dialogue. It may be explicit or implicit, verbal or not (*e.g.* an acknowledgment given as a gesture). It may appear as one or more propositions, but also as part of a proposition (word, adverb, . . .). It expresses an entitlement or a decision of the speaker, and also its acknowledgment by the addressee. In some sense, it is quite close to a speech act. However a speech act may correspond to several dialogue acts as shown in following examples. Dialogue acts are indeed more elementary than speech acts. They can be seen as the basic blocks from which one builds interpretation for dialogical interventions or even utterances. Formally, a dialogue act may be defined as an action in Ludics together with the expression that reveals the dialogue act in the intervention. Such an expression may be a proposition, a word (*e.g.* a single adverb, a noun), a prosodic feature, a non verbal sign (a nod, a shake, a slap, . . .). In trivial cases, an intervention is a unique dialogue act. Otherwise a turn of speech has to be decomposed into sequences of dialogue acts, hence may correspond to a complex design. Note that the representation of an utterance in terms of dialogue acts is dependent of the context of the dialogue, and in particular of past interventions that occurred.

Let us give a small example of a dialogue, its rudimentary interpretation in terms of dialogue acts and the way interaction is done with Ludics.

Example 6. Let us consider the following example between a traveller T and an employee E:
– T: *What time does the next train to Paris leave?*
– E: *7:45 p.m.*
– T: *Thanks.*

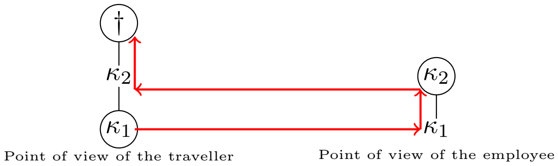
In a first approximation, each intervention is interpreted as a unique dialogue act:

- dual actions $\kappa_1/\overline{\kappa_1} = (+/-, \xi, \{0\})$ and an expression that is the proposition "*What time does the next train to Paris leave?*"; ξ is a locus arbitrarily chosen on which this act is localized; this act offers a unique opening on which E may anchor her answer.

[6] Note that the duality positive/negative does not correspond to a duality question/answer as in game semantics but to a duality production/reception in dialogue modelling.

- dual actions $\kappa_2/\overline{\kappa_2} = (+/-, \xi.0, \emptyset)$ and an expression that is the proposition "*the next train to Paris leaves at 7:45 p.m.*" ; $\xi.0$ is the locus of this dialogue act, justified by κ_1; there is no opening created by this act as it is simply a given fact.
- an action $\kappa_3 = (+, \dagger)$ with the expression "*Thanks*"; with this dialogue act, T informs E that the dialogue went well and is finished.

Actions κ_1 and κ_2 are positive from the point of view of the locutor that produces them: positive for T for the first, for E for the second, and negative for the locutor that receives them. The action κ_3, positive, is produced by T. The interaction between the two (really simple) designs is depicted in the following figure. It converges as it ends with †.



In the following subsections, we present first the approach by means of a simple dialogue extracted from a novel of C. Dickens. Then we illustrate specific forms of dialogues in terms of designs.

3.1 First Use in Dialogue

The following dialogue is extracted from the novel "David Copperfield" of C. Dickens.

Example 7. (C. Dickens, David Copperfield) The dialogue takes place between a coachman (**C**) and David (**D**), the coachman brings David to London:

C | I_1 : You are going through, sir?
D | I_2 : Yes, William. I am going to London.
I shall go down into Suffolk afterwards.
C | I_3 : Shooting, sir?
D | I_4 : I don't know.
C | I_5 : Birds is got wery shy, I'm told
D | I_6 : So I understand
C | I_7 : Is Suffolk your county, sir?
D | I_8 : Yes, Suffolk's my county.
C | I_9 : The dumplings is uncommon fine down there ...

As a first approximation, a dialogue act is associated with each utterance with actions $\kappa_1, \ldots, \kappa_9$ and respective espressions $I_1, \ldots, I_9$. The justification between actions is given by the figure on the right in example 1. The first action κ_1 is initial: with this dialogue act **C** initiates the dialogue. Actions κ_3, κ_7 and κ_9 are justified by the second intervention κ_2: in the three cases, the corresponding dialogue acts refer to '*Suffolk*', entity present in the second intervention that may introduce topics concerning hunting, native soil, gastronomy, ...

Ludics allows for rebuilding the designs whose interaction produces as a trace this alternate sequence of interventions. The dialogue acts may be given with a positive polarity for the interventions of **C**: in that way, the dialogical interaction is represented with the point of view of **C**. Reversing the polarities, one gets the point of view of **D**. This produces two designs that interact and whose trace is the dialogue. In figures given in example 1, the design on the left is the point of view of **C**, and the design on the right is the point of view of **D**.

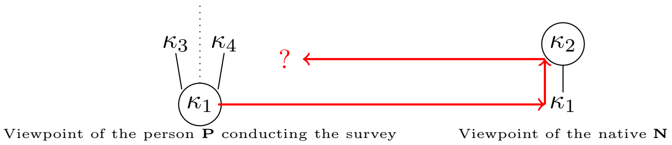
More generally, actions of Ludics are interpreted in terms of dialogue in the following way: a positive action in a design corresponds to an active rôle of the locutor when the design is her point of view. On the contrary, a negative action in the same design reports a passive rôle (*e.g.* receiving an intervention of her interlocutor). Viewing the dialogue as a trace makes explicit the fact that there are two points of view. Moreover it allows to observe the success or the failure of these two points of view, *i.e.* the fact that the dialogue may fail or end with a drop.

3.2 Divergent Dialogues, Presupposition in Dialogues

Dialogues may badly end because of misunderstandings, disagreements,... and it is necessary to be able to represent such situations. Ludics interaction distinguishes two cases. An interaction is convergent when it ends with a daimon, it is divergent when a positive focus has no dual counterpart. These two cases allow for interpreting two standard final situations in dialogues: either it finishes well or there is a misunderstanding between the two locutors. The following example is given by M. Chemillier [19] to illustrate the difficulties that arise when one wants to isolate the logical part of a dialogue in a field survey.

Example 8. A person **P** conducting a survey gives to a native the following informations: *All the Kpelle cultivate rice. Mister Smith does not cultivate rice*, the person **P** asks the following question to the native **N**:
- **P**: "Is Mister Smith a Kpelle?" (κ_1)
- **N**: "I do not know Mister Smith, I have never seen him." (κ_2)

Here we consider only the question and its answer that we represent each by a single dialogue act (resp. with actions κ_1 and κ_2). The person **P** expects a logically correct answer, *e.g. "No"*, with action κ_3. He plans also to receive an incorrect answer, *e.g. "Yes"* or *"It may be the case"*, with action κ_4. Hence the design for the person **P** may be either $\textcircled{\kappa_1}$ followed by κ_3, or $\textcircled{\kappa_1}$ followed by κ_4. The interaction in Ludics is given below: it is divergent as there is no negative action dual to κ_2 in what is expected in the design of **P**. The dialogue cannot continue.

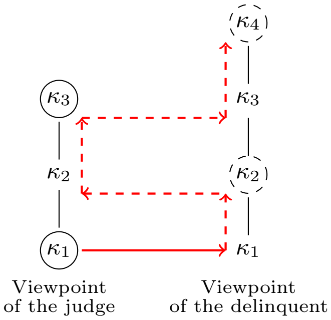

κ_3
κ_4
?
κ_2
κ_1
κ_1
Viewpoint of the person **P** conducting the survey
Viewpoint of the native **N**

The previous modeling is simplistic: a turn of speech may be more complex than a simple action. In fact, to each intervention may be associated a set of dialogue acts that complements the current design of the locutor. The result should still be a design. In particular, the first dialogue act should be anchored in some previous intervention: the dialogue continues normally if each person answers (in some way) a previous intervention. Furthermore, an intervention must be represented by a sequence of dialogue acts beginning and ending with a positive action: if the last action is not the daimon, the interlocutor may continue the dialogue. Dually, the design of the interlocutor is increased by what comes from the locutor. The result should be such that one of the chronicles ends with a negative action: the interlocutor initiates her own intervention by a dialogue act anchored on this negative action.

Dialogues with presuppositions are examples of such more complex situations. A presupposition is an implicit assertion concerning the world, whose validity is accepted in the dialogue.

Example 9. Let us consider this well known example due to Aristotle; a judge asks a young delinquent this question: *"Have you stopped beating your father?"*. Answering this question by '*Yes*' or '*No*' supposes that the answer of an implicit question is '*Yes*':
– *"Did you beat your father?"*
– *"Yes."*
– *"Have you stopped beating him?"*

The question asked by the judge, "*Have you stopped beating your father?*" exhibits three dialogue acts κ_1, κ_2 and κ_3 that refer respectively to interventions : "*Did you beat your father?*"; "*Yes*"; "*Have you stopped beating him?*". If the young adult continues the dialogue, he implicitly assumes that the interaction between the two (partial) designs do not diverge. Hence he accepts to justify his next intervention on the negative dialogue act dual to κ_3, this dialogue act being anchored on the positive dialogue act dual to κ_2: he assumes he would have answered "*Yes*" to the implicit question.



3.3 Argumentation

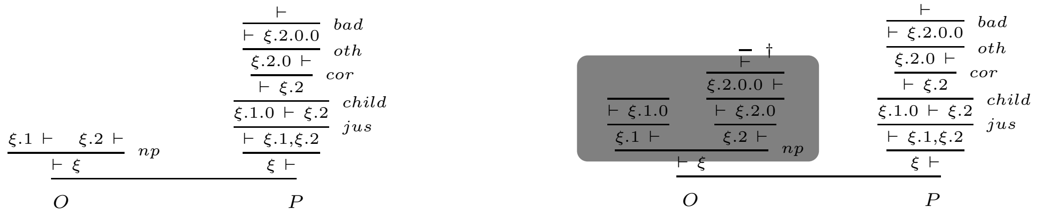
In *L'Art d'avoir toujours raison* [20], Schopenhauer means to define dialectics as "the art of winning controversies". Ideally, the easiest way to win a controversy is to have a strategy, winning against adversaries. This corresponds to a proof in Ludics. Several Schopenhauer's stratagems have been studied by Quatrini in [21], we reproduce below an example of *retorsio argumenti* or turning of the tables, by which your opponent's argument is turned against himself:

EXAMPLE **1** *The opponent O declares, for instance, "So-and-so is a child, you must make allowance for him." The proponent P retorts, "Just because he is a child, I must correct him; otherwise he will persist in his bad habits."*

The controversy between P and O is analyzed in the following way:

- O justifies his implicit thesis (*"You might be wrong correcting this child"*) by two premisses: *"He is a child"* and *"you must make allowance for children"*. This is represented by a unique dialogue act with action $(+, \xi, \{1, 2\})$, and expression noted *np*. The ramification contains two elements as the argumentation has two premisses.
- P accepts this intervention with an action dual to the previous one. Then he replies: he explicitly concedes the first premisse and contradicts implicitly the second premisse (*"you must make allowance for him"* = *"you must correct him"*) by giving a counter-argument *"otherwise he will persist in his bad habits"*. This gives rise to several dialogue acts (see figure 1 on the left):
  - The first sentence: *"Just because he is a child"* contains a concession : $(+, \xi.1, \{0\})(-, \xi.1.0, \emptyset)$. The first action is expressed with the word *"Just"* (noted jus) and the second action is expressed by *"he is a child"* (noted *child*).
  - The intervention of P continues with a negation of the assertion of O. This is represented by an action $(+, \xi.2, \{0\})$ expressed with *"I must correct him"* (noted *cor*);

- The counter-argumentation conveys two dialogue acts: the first one $(-, \xi.2.0, \{0\})$, expressed by *"otherwise"* (noted *oth*), lets P resuming his intervention, and the second one $(+, \xi.2.0.0, \emptyset)$ establishes his argument, expressed with *"he will persist in his bad habits"* (notê *bad*). The ramification of this last action is empty as this final argument cannot be contradicted.
- The situation after the opponent O accepts this intervention of P is depicted in figure 1 on the right, there are no more loci where O may continue the dialogue hence either he refuses it (divergence) or he "plays a daimon" (to keep the dialogue convergent). In any case, he looses.



Situation after the intervention of P Situation after O accepts the intervention of P

Fig. 1. Schopenhauer - stratagem 26

4 Meaning in Ludics

Resting on the formalization of dialogues in Ludics as it is presented in the previous section, propositions have been done for modeling various aspects of Natural Langage analysis, especially meaning. In this section we show how Ludics has been used for semantics and speech acts. Furthermore, we precise how inferential issues may be tackled with such a formalization. We illustrate these propositions by analysing further example 8 we reproduce below.

Example 10. (example 8 ctd)
- **P**: "All the Kpelle cultivate rice. Mister Smith does not cultivate rice. Is Mister Smith a Kpelle?"
- **N**: "I do not know Mister Smith, I have never seen him."

4.1 Semantics of utterances

In [1,4], Lecomte and Quatrini propose a conception of interactive meaning based on Ludics where the Ludics frame is at the same time used as a metaphor and as a formal device able to elaborate this metaphor. At a metaphoric level, a design being defined by its orthogonal[7], they postulate that the meaning of a sentence is given by its *dual* sentences. Moreover, they claim that Ludics offers also a framework to model the "meaning" of a sentence.

Precisely, the *dual* sentences of a given sentence u are utterances v which both:
- correctly interact with u: a speaker S claims the sentence u, her interlocutor expresses such an utterance v, and S is able to continue the dialogue.
- directly concern the contain of u: the utterance v is a question or a negation which may be considered as a test against u.

[7] according to the separation theorem.

In this way, a set of designs is associated with the meaning of the sentence u: such designs are the carriers of the dialogues during which a speaker S affirms and justifies the statement u against an interlocutor.

Let us look at the utterance $\mathbb{E}$ and some of its dual sentences:

($\mathbb{E}$) "All the Kpelle cultivate rice."

(a) "But Mr. Durand does not cultivate rice."
(b) "Even the children?"
(c) "Was it the case twenty years ago?"

Let us consider the formalization in Ludics of one of these starting dialogues. Suppose that a speaker S claims the utterance $\mathbb{E}$ and her addressee answers saying (a).
From the point of view of S, the first intervention is a positive dialogue act which corresponds to the assertion of $\mathbb{E}$ by S. The ramification of this dialogue act is a singleton: only one locus is created on which the interaction may continue (w.r. to this asserted sentence). Finally the expression of the dialogue act is the utterance $\mathbb{E}$.
The second intervention is represented also by a unique dialogue act. The speaker receives a counter-argument from her addressee. According to the view of S, this dialogue act is a negative one whose expression is the sentence (a). Its ramification contains two elements: the dialogue may continue questioning the fact that Mr. Durand is or is not a Kpelle, or the fact that Mr. Durand does or does not cultivate rice.
When she asserts the utterance $\mathbb{E}$, the speaker has to be ready to receive such counter-arguments that concerns individuals about whom the addressee may argue that they are Kpelle, but that they do not cultivate rice. The design associated with the project of dialogue of the speaker S when she asserts $\mathbb{E}$ must contain such possibilities. It is represented in figure 2.

$$\dfrac{\dfrac{\begin{matrix}\vdots & & \vdots & \\ \vdash \xi.0.1_d, \xi.0.2_d & \qquad & \vdash \xi.0.1_{d'}, \xi.0.2_{d'} & \qquad \dots\end{matrix}}{\xi.0 \vdash}}{\vdash \xi}$$

Fig. 2. The project of dialogue of a speaker asserting $\mathbb{E}$

After the dialogue act corresponding to the assertion of $\mathbb{E}$, this design contains as many branches as individual Kpelle (denoted by d, d' ...). Moreover, for each such Kpelle d, the speaker has to be ready to continue the dialogue by giving some argument to justify that d cultivates rice: each branch indexed by d may have ulterior actions. In fact, there is not a unique such design, but a more precise specification of these designs should be relative to contextual considerations. In order to interact with other utterances, for example with questions like (b) or (c), designs would have some longer branches in order to explicit which Kpelle are concerned by the claim (for example the adults) or in order to precise the historical context of the fact.

Nevertheless, we may observe that all such designs share the same starting actions. Then, interpreting actions as logical operations, we may associate with these designs the beginning of a logical proof, represented in figure 3.

Let us sum up the methodology that we sketched above. As "meaning" of an utterance, we associate with the utterance a set of designs, namely the designs which are the supports of dialogues that the claim of this utterance initiates. And we remark that, since these designs share their first

$$\dfrac{\dfrac{\begin{array}{ccc}\vdots & \vdots & \\ \vdash\downarrow K(d)^{\perp}, C(d) & \vdash\downarrow K(d')^{\perp}, C(d') & \ldots\end{array}}{(\forall_x(\uparrow K(x) \multimap C(x)))^{\perp} \vdash}}{\vdash\downarrow (\forall_x(\uparrow K(x) \multimap C(x)))}$$

Fig. 3. A logical reading of the assertion of $\mathbb{E}$

actions, we may organize them in a set of designs built by means of operations of Ludics. Namely, we associate as meaning of $\mathbb{E}$ the set:

$$\mathbf{E} = \downarrow (\&_d(\uparrow \mathbf{K}(d) \multimap \mathbf{C}(d)))$$

where $\mathbf{K}(d)$ and $\mathbf{C}(d)$ are also sets of designs. In particular, provided that $\mathbf{K}(d)$ and $\mathbf{C}(d)$ are behaviours associated with logical formulas, the set $\mathbf{E}$ is a behaviour associated with a logical formula. In this way, we retrieve the semantical notion of a "logical form", thus also its nice properties: vericonditional semantics, quantifiers scope ..., in the same time we retrieve a more proof theoretical approach of semantics, hence transposing to Natural Language semantics the following motto: *the meaning of a logical formula is the set of its proofs*.

At a first glance, this does not seem very different from the way formal semantics usually proceeds. Nevertheless, let us underline some points which are slightly different and new and which could favorably extend the usual models of semantics:

- Recall that formulas are not primitive objects in Ludics: interaction is first defined on designs. It may be relevant to associate as meaning of a sentence an object which is not *a priori* closed, *i.e.* a set of designs that is not necessarily a behaviour. It may also be relevant to associate designs that are not completely defined, *i.e.* that may be *more and more refined*. For example, the formula $K(d)$ may be either an atomic one but may also be decomposable, say in $k(d) \otimes A(d)$ to give an account that the Kpelle, the claim is about, are adults.
- Another result is that the logical interpretation of the quantifier '*All the*' is not *a priori* fixed. Depending on the context and on the kind of justifications of the claim $\mathbb{E}$, the quantifier may be either interpreted by a generalized additive conjunction in case the justification is relative to each individual, what was used to define $\mathbf{E}$, or interpreted by a universal first order quantifier, provided that the justifications are the same for each individual, *i.e.* do not depend on the individuals. We recall that, in a study on first-order quantifiers in Ludics [22], Fleury and Quatrini show that the family of designs associated with a first order universal quantifier has to satisfy a *uniformity* property: roughly speaking the designs should be in some way 'the same' (and should represent the same proof).

4.2 Speech acts

The framework brought by the formalization of dialogues in Ludics has also been used to give an account for speech acts. In [2], Fleury and Tronçon transpose the characterization of speech acts given by Searle [23] in ludical terms. A speech act may be associated with a design which is able to interact with designs associated with the preconditions of this speech act. The normal form we get after interaction gives an account of the effects of the speech act. In [3], Lecomte and Quatrini characterize some figures of dialogue according to the form of their associated designs. In particular they focus on elementary speech acts. Let us recall the cases of assertions and interrogations.

As proposed by Walton in [24], **asserting** is “willing to defend the proposition that makes up the content of the assertion, if challenged to do so”. This remark was the core for defining the meaning of an utterance (see previous subsection). When a speaker asserts some utterance, she must have in mind all justifications for predictable objections, hence the design associated with an assertion starts with a positive dialogue act, followed by a set of negative dialogue acts, each of them being the basis for justifications against predictable objections. Note that an assertion is not represented as a simple (logical) proposition. Indeed a design associated with a proposition takes into account just its logical behaviour, say an atomic proposition[8], a decomposable formula. Although the design associated with the assertion of this proposition contains also the dialogue act corresponding to the utterance of the assertion. This difference is pointed in figure 4 which gives the designs for ‘*Mr. Smith is/is not a Kpelle*’. Designs concerning the negative cases finish with a daimon to indicate that this position is given up: this enables to express a logical negation. Note finally that our representation is extremely simplified with respect to what could be a general assertion. We consider only the logical feature of utterances, and we suppose that its logical form is univocally fixed.

Propositions		Assertions	
$\dfrac{}{\vdash K(s)}\emptyset$	$\dfrac{\dfrac{\dfrac{}{\vdash}\dagger}{K(s)\vdash}\emptyset}{\vdash\downarrow K(s)^{\perp}}$	$\dfrac{\dfrac{\dfrac{}{\vdash K(s)}\emptyset}{P^{\perp}\vdash}}{\vdash\downarrow P}$	$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash}\dagger}{K(s)\vdash}\emptyset}{\vdash\downarrow K(s)^{\perp}}}{\downarrow P\vdash}}{\vdash\downarrow\uparrow P^{\perp}}$
is	**is not**	**is**	**is not**

Fig. 4. Proof-like designs associated with the propositions/assertions ‘*Mr. Smith is/is not a Kpelle*’.

Unlike other speech acts, it is relevant to associate with **questions** designs with more than one locus in the base. Indeed, viewed as a speech act, an interrogation consists in asking a question and being ready to *register* the answer. Thus it is necessary to associate with an interrogation a design with base $\vdash \tau, \sigma$, where τ is the locus of the question and σ is the locus where the answer is registered: such a design ends with a $\mathcal{F}ax$ so that the answer to the question is *moved to* σ when interacting with a dual design.

Let us consider for example the question: *“Is Mr. Smith a Kpelle?”*, that we denote by q. The design associated with this question is represented in figure 5. The speaker who is asking this question is ready to receive three answers:
- either an assertion, say ‘*Mr. Smith is/is not a Kpelle*’. In such a case, she registers the answer:

[8] In Ludics, one associates with it the design which is the proof of the logical constant 1.

the fax after the action on the locus $\tau.0.1$ is dedicated to copying this assertion (from $\tau.0.1.0$ to σ).
- or the word '*Yes*'. In such a case, the speaker rebuilds the information: it uses the structure of the design for the proposition '*Mr. Smith is a Kpelle*' (figure 4).
- or the word '*No*'. This case is similar to the previous one: it uses the structure of the design for the proposition '*Mr. Smith is not a Kpelle*' (figure 4).

$$\dfrac{\dfrac{\dfrac{\mathcal{F}ax}{\tau.0.1.0 \vdash \sigma}}{\vdash \tau.0.1, \sigma} \qquad \dfrac{}{\vdash \tau.0.2, \sigma}\,\emptyset \qquad \dfrac{\dfrac{\dfrac{}{\vdash \tau.0.3}\,\dagger}{\sigma.0 \vdash \tau.0.3}\,\emptyset}{\vdash \tau.0.3, \sigma}}{\dfrac{\tau.0 \vdash \sigma}{\vdash \tau, \sigma}}$$

Fig. 5. The design associated with the question q.

If the answer is '*Yes*', *i.e.* represented by the design $\dfrac{\dfrac{\tau.0.2 \vdash}{\vdash \tau.0}}{\tau \vdash}$, the normal form of the net defined by the designs associated to the question and the answer is then $\dfrac{}{\vdash \sigma}\,\emptyset$. In other words, the result is a design localized on σ that corresponds to the proof of the proposition '*Mr Smith is a Kpelle.*'.

4.3 Inferences

Designs that represent independent parts of a dialogue or independent dialogues may be used in order to do inferences: when they form a net of designs, they may normalize. We illustrate this point by considering again example 8 where we suppose that the Native answers '*Yes*'. We focus first on the overall intervention of the investigator:

– **She provides a first information '***All the Kpelle cultivate rice***'.** This utterance is given as a **true** proposition. Even more, the investigator suggests to her addressee to use it in a logical reasoning. That is a full proof is provided by her intervention. This part of the intervention is naturally associated with the design corresponding to this proof. Proof and design are both represented below. For simplification issue, we consider an axiom rule (and a fax in the design) in place of a development of a justification for the fact that a Kpelle cultivates rice, in other words being a Kpelle implies cultivating rice and also having other properties that characterize a Kpelle: we abstract from these other properties hence we get an axiom.

$$\dfrac{\dfrac{\dfrac{}{K(d) \vdash C(d)}\,ax}{\vdash\downarrow K(d)^{\perp}, C(d)} \qquad \dfrac{\dfrac{}{K(d') \vdash C(d')}\,ax}{\vdash\downarrow K(d')^{\perp}, C(d')} \qquad \ldots}{\dfrac{(\forall_x(\uparrow K(x) \multimap C(x)))^{\perp} \vdash}{\vdash\downarrow (\forall_x(\uparrow K(x) \multimap C(x)))}} \qquad \dfrac{\dfrac{\dfrac{\mathcal{F}ax}{\xi.0.1_d.0 \vdash \xi.0.2_d}}{\vdash \xi.0.1_d, \xi.0.2_d} \qquad \dfrac{\dfrac{\mathcal{F}ax}{\xi.0.1_{d'}.0 \vdash \xi.0.2_{d'}}}{\vdash \xi.0.1_{d'}, \xi.0.2_{d'}} \qquad \ldots}{\dfrac{\xi.0 \vdash}{\vdash \xi}}$$

- **She provides a second information: '***Mr. Smith does not cultivate rice***'.** This information is a priori independent from the previous one, hence is represented as another design:

$$\dfrac{\dfrac{}{\vdash}\,\dagger}{\beta \vdash}\,\emptyset \qquad \textit{i.e.}\text{ the proof:} \qquad \dfrac{\dfrac{}{\vdash}\,\dagger}{C(s) \vdash}\,\emptyset$$

- **She asks a question: '*Is Mr. Smith a Kpelle?*'.** This corresponds to the design given in the previous subsection in figure 5.

We have then four designs at disposal:

– the design associated to a proof of the proposition '*All the Kpelle cultivate rice*', based on $\vdash \xi$, that we will denote $\mathcal{D}_\xi$;
– the design associated to a proof of the proposition '*Mr Smith does not cultivate rice.*' based on $\beta \vdash$, that we will denote $\mathcal{D}_\beta$;
– the design noted $\mathcal{D}_\alpha$ obtained by means of a shift and a delocation (from σ to $\alpha.0$) of the design that results from the interaction of the question q and the answer '*Yes*' and which is associated to a proof of the proposition '*Mr Smith is a Kpelle.*' (end of last subsection). The design $\mathcal{D}_\alpha$ is then associated with a proof of $\downarrow K^{\perp}(s) \vdash$.
– The following design associated with a proof which enables one to perform the interaction between the proposition '*All the Kpelle cultivate rice*' and some propositions about one Kpelle, that we denote by $\mathcal{F}_{\xi,\alpha,\beta}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{F}ax}{\alpha.0 \vdash \xi.0.1_s.0}}{\vdash \xi.0.1_s.0, \alpha}}{\xi.0.1_s \vdash \alpha} \qquad \dfrac{\mathcal{F}ax}{\xi.0.2_s \vdash \beta}}{\dfrac{\vdash \xi.0, \alpha, \beta}{\xi \vdash \alpha, \beta}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{K(s) \vdash K(s)}}{\vdash K(s), \downarrow K^{\perp}(s)}}{\downarrow K^{\perp}(s) \vdash \downarrow K^{\perp}(s)} \qquad \dfrac{}{C(s) \vdash C(s)}}{\dfrac{\vdash \exists_x((\uparrow K(x))^{\perp} \otimes C(x), \downarrow K^{\perp}(s), C(s)}{\downarrow (\forall_x(\uparrow K(x) \multimap C(x))) \vdash \downarrow K^{\perp}(s), C(s)}}$$

Then, the cut-net $[\![\mathcal{F}_{\xi,\alpha,\beta}, \mathcal{D}_\xi, \mathcal{D}_\alpha, \mathcal{D}_\beta]\!]$ normalizes, its normal form is the design which encodes the logical contradiction $\dfrac{}{\vdash}\,\dagger$.

5 Conclusion

In the previous sections, we present Ludics and its current uses in modeling Natural Language. It should be clear that this is an ongoing research that changes largely the way Natural Language may be analyzed, as Ludics changes radically the point of view we may have on logics. In Ludics, the fundamental operation concerns interaction between objects called designs. Such objects may clearly be interpreted in a certain sense as "proofs". However, two main differences exist with what is generally called a proof that should be considered. First, a design may include daimon actions, this allows for considering proofs and counter-proofs in the same language. Second, a design may be infinite (depth as well as height), hence such an object may include enough information to interact with an infinite number of counter-objects. We recall also that Ludics is a rebuilding of Linear Logic: formulas may be denoted by closed sets of designs. In that way, one recovers standard concepts of logics, say truth, proof,...

If we summarize the modeling of language by means of Ludics, the most important point is that it is the interaction in dialogue (explicitly or not) that provides content to the language, and not a relation to some external reality, hence assuming an "inferentialist" position [25]. In the previous sections, we presented a few domains where this principle has begun to be applied, namely dialogue, speech acts, semantics. Works of Terui [17] about automata and formal language show that syntax may be also studied in that perspective. However, this approach has to be carried on to improve the propositions and to tackle other questions in Natural Language modeling.

References

1. Lecomte, A., Quatrini, M.: Ludics and its applications to natural language semantics. In Ono, H., Kanazawa, M., de Queiroz, R.J.G.B., eds.: WoLLIC. Volume 5514 of Lecture Notes in Computer Science., Springer (2009) 242–255
2. Fleury, M.R., Tronçon, S.: Ludics as a frame for the formalisation of speech acts. In Lecomte, A., Tronçon, S., eds.: Ludics, Dialogue and Interaction. Volume 6505 of Lecture Notes in Computer Science., Springer (2011) 1–25
3. Lecomte, A., Quatrini, M.: Figures of Dialogue: a View from Ludics. Synthese **183** (2011) 59–85
4. Lecomte, A., Quatrini, M.: Pour une étude du langage via l'interaction : dialogues et sémantique en Ludique. Mathématiques et sciences humaines **189**(1) (2010) 37–67
5. Girard, J.Y.: Locus solum: From the rules of logic to the logic of rules. Mathematical Structures in Computer Science **11**(3) (2001) 301–506
6. Girard, J.Y.: Linear logic. Theor. Comput. Sci. **50** (1987) 1–102
7. Fouqueré, C.: Ludics and web: Another reading of standard operations. In Lecomte, A., Tronçon, S., eds.: PRELUDE Project. Volume 6505 of Lecture Notes in Computer Science. Springer (2011) 58–77
8. Girard, J.Y.: From foundations to ludics. Bulletin of Symbolic Logic **9**(2) (2003) 131–168
9. Girard, J.Y.: Le Point Aveugle : vers l'imperfection. Volume 2 of Visions des Sciences. Hermann (2007)
10. Curien, P.L.: Introduction to linear logic and ludics, part i. CoRR **abs/cs/0501035** (2005)
11. Curien, P.L.: Introduction to linear logic and ludics, part ii. CoRR **abs/cs/0501039** (2005)
12. Basaldella, M., Faggian, C.: Ludics with repetitions (exponentials, interactive types and completeness). In: LICS, IEEE Computer Society (2009) 375–384
13. Curien, P.L., Faggian, C.: L-nets, strategies and proof-nets. In Ong, C.H.L., ed.: CSL. Volume 3634 of Lecture Notes in Computer Science., Springer (2005) 167–183
14. Fouqueré, C., Mogbil, V.: Rewritings for polarized multiplicative and exponential proof structures. Electr. Notes Theor. Comput. Sci. **203**(1) (2008) 109–121
15. Andreoli, J.M.: Logic programming with focusing proofs in linear logic. J. Log. Comput. **2**(3) (1992) 297–347
16. Faggian, C., Maurel, F.: Ludics nets, a game model of concurrent interaction. In: LICS, IEEE Computer Society (2005) 376–385
17. Terui, K.: Computational ludics. Theor. Comput. Sci. **412**(20) (2011) 2048–2071
18. Landragin, F.: Vers l'identification et le traitement des actes de dialogue composites. In: Traitement Automatique du Langage Naturel (TALN). (2008) 460–469
19. Chemillier, M.: Eléments pour une ethnomathématique de l'awélé. Mathématiques et sciences humaines **181**(Varia) (2008) 5–34
20. Schopenhauer, A.: L'art d'avoir toujours raison. Circé (1830)
21. Quatrini, M. In: Une relecture ludique des stratagèmes de Schopenhauer. Presses de la Sorbonne (to appear)
22. Fleury, M.R., Quatrini, M.: First order in ludics. Mathematical Structures in Computer Science **14**(2) (2004) 189–213
23. Searle, J.: Speech Acts. Cambridge University Press (1969)
24. Walton, D.: The place of dialogue theory in logic, computer science and communication studies. Synthese **123** (2000) 327–346
25. Brandom, R.: Articulating Reasons: An Introduction to Inferentialism. Harvard University Press (2000)