Speech Acts in Ludics*

Marie-Renée Fleury, Samuel Trončon

Institut de Mathématiques de Luminy - Aix-Marseille Université

Abstract. In this paper, we attempt to show that recent developments in proof theory, especially with ludics, are relevant for the study and the formalization of speech acts. This logical framework does not deal with truth values but with proofs, and this opens a new way for taking in charge the performative part of linguistic utterances. After having presented two models of speech acts and what theoretical elements we will hold as relevant for our own model, we introduce the ludical point of view by defining a speech acting conceptualization which renders some determinations not presented in the former models. We end by giving some examples of speech acts, presented in their ludical embedding, and we discuss what features the model provides.

Our aim is to show that recent developments in proof theory, especially with ludics, are relevant for the study and the formalization of speech acts. A common view about proof theory is that this logical framework does not deal with truth values but with proofs, and this opens a new way for taking in charge the performative part of linguistic utterances. Proofs are sometimes treated in the litterature as a syntactic objet, and beotians ignore what could be a semantic for proof theory. According to the cut elimination procedure due to Gentzen, and the related properties of convergence (the so called "*Hauptsatz*"[1]) and confluence (Church-Rosser), "*denotational semantics*" are defined only upon proofs with proof reduction. And a recent and genious extension of these ideas is given by the notion of "*encounter*" in Ludics, due to Girard (Girard 01), by which we can study convergences and divergences in the process of interaction between dialogical structures[2].

We present in the first section two models of speech acts and what theoretical elements we will hold as relevant for our own model. Secondly, we introduce the ludical point of view by defining a speech acting conceptualization which renders some determinations not presented in the former models. In the third section we present the ludic framework, not extensively detailed but focusing on the main points used in our formalization (designs, behaviors and the normalization

---

* This work was supported by the ANR project no. 06-BLAN-0032 "PRELUDE".

[1] The "*Hauptsatz*" is a theorem proved by Gentzen, which ensure the convergence of calculus (by the process of cut-elimination). This convergence is strong in the case of an intuitionnistic calculus, and weak for the classical one.

[2] For a good introduction to actual development of proof theory and the works of Girard, you can read recent works of Jean-Baptiste Joinet (17) and Samuel Trončon (28).

procedure). Last but not least, we give some examples of speech acts, presented in their ludical embedding, and we discuss what features the model provides.

1 Introducing speech acts

1.1 Classical view

First we observe that foundationnal remarks about speech acts and the classical separation between constative and performative assumptions, as given first by Austin and Searle, opens the way to a bridge with proof theory. For example, the meaning of a constative sentence like %`The weather is good` can be explained in terms of truth and falsity. But, in the case of a performative sentence like %`I wish you to grow old`, we can't say what would be the truth value. We just know that, in some cases depending of the context, the sentence would be understood as a real wish or not. For this reason, we can identify a whole class of natural sentences which are not truth valuable and for whose meaning determination require a contextual analysis : these utterances convey "*Speech Acts*", *i.e.* (linguistic) objects which realize some concrete (pragmatic) action. A short analysis of embodied speech acts shows that their realizability conditions are unhomogeneous. Some of them are very primitive (*for expl.* the presence of an addressee which can receipt messages), others are conventional (*for expl.* use of water for the baptism) and procedural (*for expl.* to get divorced is possible only after being married). We have also conditions justified by mental states (*for expl.* sincerity of the promiser), and behaviors or expectations accorded with actual speech act orientation (*for expl.* in the case of the promise, the adressee would expect for the promised object).

We could define the core problem of the speech acts theory as a problem of assignment of type (Gazdar, 1981) : what markers and processes do we use to recognize the category of a speech act ? In the tradition, a large place is given to illocutionary force markers (*for expl.* %`I promise ...` in a promise), because it is the most evident manner for typing a speech act. But, when considering actual speech acts, explicit and/or pure syntactic markers are not sufficient, and we would have to consider not explicit markers, pragmatic means and contexts to specify the type of a speech act.

In the most common view, we can define a speech act as a mean used by a locutor in order to produce an effect in its environment by its words. By this use, he wants to inform, incite, convince or demand something to its addressee.

We suppose that for each meaning M, there exist an expression E which is a relevant formulation of M, *i.e.* E is sufficient to transmit the meaning's intention M to the interlocutor. So, following the *founding fathers*[3], we could define a speech act as a quadruplet (I,Γ,B,Υ) given by a communicational intention (I), a set of pre-requisite conditions (Γ), a body which realize concretely the action (B), a set of effects (Υ). This definition states speech acts in their purest form, *i.e.* the most disembodied one, as we see in the following example.

[3] Evidently, Austin, Searle and their successors.

Example 1 (The promise). A speech act S will be considered of the type "promise" when four conditions are respected[4] :

- *Spk* has the intention of meaning some propositional content p
- *Spk* wants to make a commitment to *Add* about p
- *Spk* really wants to fulfil this commitment
- for *Spk* and *Add* , it is an evidence that without this speech act, *Spk* would not have done the promised action

Some remarks can be done at this stage. First, we do not know exactly how to consider and take in charge the speaker's **intention** of communicate something because the speech act is defined as if the intention was totally transparent.
Second, the **normativity** of speech acts is unclear, because it suggests that conditions are fully established before the realisation of the speech act. But we know that by convention we intend to justify the fact that speakers associate linguistics expressions with meanings they want to refers.
Last, the definition suffers from a lack of clarity and precision. It concerns notably the fact that we do not know exactly where a speech act begins : Is the speech act distinct from its effects ? What difference can we make between the effect and the intention ?
Considering these critical points, we consider that a good theory of speech act would have to define conventions and intentions not only as core notions but, evidently, as material objects upon which interactions between agents can be developed. Although these critical points, which we will take in charge in the last part of this paper, we find here the main connections with our proof-view in logics. The body of a speech act, *i.e.* the means which operates an effect, can be seen as a function responding to an expectation. This function, in order to be operational (*i.e.* to produce some effect), must interact with another function, some set of data and/or contextual elements : we would call all this bunch of unhomogeneous elements *functionnal environment*.

Speech Acts Theory	**Functions Theory**
intention	function's type
conditions	contextual datas required in a given type
body	function
effects	output datas produced in a given type

1.2 Inside games

One hypothesis about speech acts that we do not want to justify is about the *Literal Force Hypothesis* (18). It establishes a bi-univoque relation between a restricted set of illocutionary forces and a closed set of clause types. We agree

[4] Considering that the object of this article is not to introduce precise categorizations of speech acts but a reflexion on their formalization, we will not consider in our examples all the possible conditions, just their most common specificities.

with, for example, Beyssade and Marandin (4) which wants to solve some problems of assignement in speech acts without assumption about litteral force (9). For Beyssade and Marandin, we must enrich the semantics and take in charge in a more fine manner the process of communication. In other words, speech acts must be seen as actions committed (by a speaker) and accepted (by some addressee), which updates some databases about facts, things to do and commitments. This mirroring effect can be called the *game-play*, and the syncing effect between players is like a *gameboard* (as it is called in (3)). We can see the origin of this idea of *game-play* in the fact that sentences can produce dysimetrical effects on context. They engage a commitment of the speaker about something, and call on the addressee to take in charge some counterpart. With these ideas, Ginzburg opens the way to the formalization of speech acts as games, *i.e.* integrating the two players in the analysis.

Syntactic type	**Form**	**Speaker's committment**	**Call to addressee**
declaratives	direct	truth	truth
	interro-declarative	truth	question
interrogatives	direct	question	question
	interro-negative	falsity	question

So, the notion of commitment is modelized by the means of *gameboard* in the Ginzburgh view. It generates a sort of bookkeeping of actions, things to do, shared knowledge and all the things which must be known for ensuring the practicability of speech act and/or the reliability with what follows the speech act. In the classical style, we postulate that a speech act is defined by some propositional content associated with an illocutionary force, according to the fact that the content can be affected by the force. For example, (1) (2) and (3) have the same propositional content (p = '`we laugh`'), and three different illocutionary forces : respectively `assertion`, `question`, `command` :

(1) %`We laugh.`
(2) %`Are we laughing ?`
(3) %`Let us laugh !`

Gazdar said in (9) that the two main problems raised by this foundation are about the uniformity of the propositional contents transmitted (as if it was the same in all the variations), and the bijective relation between clause-types and illocutionary forces (as if it was possible to establish a strict and decidable correspondance between them). Ginzburg and Sag (11) are morever in favour of defining multiple types of propositional content accorded to types of speech act, as we see in table 1.2.

Syntactic Type	**Semantic Type**	**Pragmatic Type**
Declarative	Proposition	Being able to demonstrate the proposition
Interrogative	Question	Being interested in the answer
Imperative	Query	Being waiting the realization of a potential state of affairs

In the game view, like *for expl.* in Gazdar, we see the speech act as a program which operates on private and shared databases. For a speech act, the program updates essentially two types of entries : commitments of the player, call on its opponent. Beyssade and Marandin propose to extend the Gazdar's notion of commitment (9) defining it as a function operating on the environment. In this manner, we can see the following examples of speech act types :

Example 2 (Assertion). An assertion about the fact that F is a function which modifies the context in which the speaker is not committed about the knowledge of the truth of F, in a context in which the speaker is committed to demonstrate the truth of F.

So an actual road map of the speech acts theory would be clearly defined by the aim of taking in account the interactivity and the context dependance.
First, we have to show that speech acts we are interested in can be seen as a collaboratives results. For this reason, the function played by the addressee in the felicity of a speech act must attract all our attention.
Second, we must restore the dialogical dimension of speech act, notably in their evaluation. This point implies to relay the game view, concentrating our efforts on the chaining and the embedding of speech act.
The third objective is about the extension of the notion of commitment, viewing it like an *impact on an environment*. Now, we can define commitment as a registration, and context modifications as transformations on the data structure (like *for expl.* a database). By this way we hope to take account of different possible realizations for each type : perfect, indirect, partial, disturbed or asymetric...

1.3 Towards a dialogical view based on ludics

In the first subsection, we presented the common view about speech acts, as developped by Austin, Searle and their followers. With the second subsection, we introduced the view based on the "*speech acts as games*" interpretation, as developped in the vein of Gazdar, Ginzburg, Beyssade and Marandin. We identify in these positions two models, promoting different means for understanding speech acts situations and effects, which do not focuse on the same object. Our view constitute a third option, not rival but complementary, which focuses on the structural modifications in the speech act world and view the speech acts as constrained processii realized in context. In the following table we sum up these distinctions[5] :

	classical view	*game view*	*ludical view*
object	conditionnal	function	interactions
variability	expression	datas	[shared] contexts
invariance	speech act type	inscriptions	impacts

It is evident that with the ludical view the complexity rise against the two others systems. But this complexity is offset by the way opened to the formalisation

[5] We call 'object' the logical form given to speech acts in the corresponding model.

of complex and multiscaled architectures of speech acts elements. For example, in the game view, we got a game design for speech acts, in which we focuse on commitments by the way of functions modifying some data bases. And by the use of ludics we introduce also a parallel analysis of speech acts, in which speech acting structures (contexts and executables) are homogeneously considered, as well identified by their polarity. In fact, we take the interaction at its more primitive level, like a sort of "machine language" which "talks" both the language of executables structures and the contextual structures one.

For another example, the inscriptions, when considered in the game view, are notifications in a notebook managed by some interacting agent (a gameboard). It suggests that commitments are countable, and that for each commitment there is almost two inscriptions in two personal notebooks, or just one inscription in some big shared notebook. This hypothesis holds very well, but in the case of a negative condition, as in the promise *for expl.* , when we have to verify that the speaker is not already commited to do the promised thing, we do not know exactly what could be an inscription of the fact that "*someone is not committed*"... A possible approach would define it as the fact that there is no inscription about that in the available notebooks. Agents would have to scan notebooks, searching for an inscription, exiting with an error. In a *ludical view* we consider that, for a good approximation of the problem, we can define the negative condition as the absence of opposition of the present speakers to the presupossed fact contained in the speech act. As if the speaker was saying "*I'm not actually commited to do the thing I want to realize with this promise*" and nobody would have neither arguments nor the desire to refute this proposition. The interest in this point is that "absence of reaction" is a procedural definition, because there could be no known reaction at t time, and a contra-reaction at $t+1$: speech act evaluation became a real-time conceptualization and the model is self-contained. So, the basic scheme we assume about contexts is summed up by the following table :

	Game view	*Ludical view*
negative condition	absence of inscription	latence of reaction
positive condition	presence of inscription	actance of reaction

Latence means that the speaker knows that some reactions are possible, but there is no reaction in this branch of the design structure, or there is an explicit giving up. *Actance* means that some reactions are activated on a branch of the design structure. Evidently, we must explain how the agent can know something by himself. We define the process of "knowing by itself" exactly as the process of "knowing by somebody", but directed on itself. So, knowing that a condition is realized from my own view is reduced to the fact of testing possible reactions of myself about some proposition.

2 Speech acts in ludics

As we must justify the relevance of a new model about speech acts, we present here the most important points for understanding our view, in their most intuitive form. The further sections being devoted to present the technical framework and some exemples.

Before going into details of our conceptualisation of speech acts let us introduce some core notions, which are imported form ludics, and would be precisely defined in the next section.

2.1 Speech acting

While speech acts are viewed as stable structures, we introduce a "speech acting" description in which the interaction plays the main part.

Classical speech acts are presented in a synchronous view. But objectively, the procedurality of embodied speech acts oblige us to introduce some complexity. Speech acting is the fact of an asynchronous processing, and we will take in account this fact first in our conceptualization, secondly in our formal view.

First, we recount the interaction process as a parallel processing, realized by two speaking agents, forming their own actions on the basis of their interlocutor's actions.

Process 1 (Speaker's action)

$\star$ *Spk proposes a speech act* $\mathfrak{A}$*. It can be seen as a function defined on the context with values in a subset of the context containing the commitments of the interlocutors. The context must satisfy some conditions* $C_1, \cdots, C_n$ *in* C*. So we denote by* $\mathbb{C}$ *a closed*[6] *set filled with justifications of* C *(*$\mathbb{C}$ *will be called a behavior, i.e. a set of designs). As we just define* A*, we can write* $\mathfrak{A} \in \mathbb{C} \to \mathbb{E}$ *which means "there is a construction called* $\mathfrak{A}$ *which transforms the contextual elements in* $\mathbb{C}$ *to obtain contextual elements in* $\mathbb{E}$*".*

$\star$ *Spk thinks that he can access in the (open) context to some locus in which the condition* C *is justified by a design* $\mathfrak{D}$ *: knowing that, he is considering a set* $\mathbb{C}$ *of justifications of* C*. This design can be located in a private or a (preferably) shared part of the context.*

$\star$ *Spk feels himself justified to provoke the commitment* $\mathfrak{E}$ *which would be realized in the context* $\mathbb{E}$*.*

Process 2 (Addressee's reaction)

$\star$ *Add perceives the speech act*

$\star$ *Add does not have objectively the same view on the context that Spk had. For example, he can perceive differently the same situation, i.e. having more elements in* $\mathbb{C}$*, or even possessing arguments which contradicts a condition* $\mathbb{C}_i$*. He can also refuse arbitrarily the effectivity of a commitment even if the conditions were filled (for expl. in the order, Add can ignore the authority of Spk on him).*

$\star$ *Add responds by forecasting the speaker's actions or by acting in some locus, or by introducing some elements which were not taken in charge by the design of Spk .*

[6] The notion of closure will then be clarified in the next section.

The complexity of speech acting representation is dependent of two important problems :
Structures we present get the form of a plan of actions and reactions. In a schematic view, actions are done by the speaker, reactions are done by the addressee. But, this pattern masks the fact that these structures renders the subtle distinction between what is planned, what is added/modified in the interaction process, and what is finally realized. In our formalization, we present speech acts as realized, *modulo* further continuations which left unchanged the given basic structure. So we must take in charge the difference between the action plan made by some speaking agent (the potent), parts of the action plan which are invoked at this time in the speaking situation (the patent), and open branchs which are not constructed as well at this time (the latent).

	of *Spk* for *Spk*	**of *Spk* for *Add***
latent	planned but not invoked	not explored but anticipated
potent	opened but not planned	not anticipated
patent	invoked	explored

We see speech acting as a parallel process implying two agent activities : the observation ("*what parts of the context are selected by my speech act's conditions of realization*") and the action ("*how I want to modify the present context*").The observation corresponds to what is anticipated by a locutor from its interlocutor (parts of its plan which have an inverse polarity). And the action corresponds to what is forecasted by some locutor from its own actions (parts of its plan which have the same polarity). The situation is slightly different depending on whether we are in a perfect world or a real world :

	perfect world *all is transparent*	**real world** *there is some opacity*
***Spk* observation**	*Add* plan	negative steps of *Spk* plan
***Spk* action**	*Spk* plan	positive steps of *Spk* plan
***Add* observation**	*Spk* plan	positive steps of *Add* plan
***Add* action**	*Add* plan	negative steps of *Add* plan

2.2 Speech acts

Definition 1 (Ludical speech acts). *A "ludical speech act" is a sequence of reduction by which some executable (called the speech act) produce an impact when some conditions in the context are fulfilled.*

A type of "ludical speech act" is a class of executables which produces the same impact when observed in the same conditions.

The speech act is uttered in some determinated context which can contains facts, knowledges, past actions, mental states... Each speaking agent selects the parts of the context he considers as relevant, and acts by reference to these elements. For example, at the utterance of the speech act by *Spk* , *Add* will react by means of parts of the context he "perceives", and which could be different from those viewed by *Spk* . The difference between what views *Spk* and what uses *Add* plays a huge function in the realization of impacts, as *Spk* can feel himself so justified by contextual elements to obtain some effects which are refused by *Add* by means of some others contextual elements.

In the interaction, these contextual elements takes the form of possible actions of speaking agents in their plans. They are positive when invoked by *Add* as contexts (negatives for *Spk*), or invoked by *Spk* as conditions. They are negative when used by *Add* as an action (positives for *Spk*), or invoked by *Spk* as anticipation. Evidently, we take the classical conditions (preliminary and essential rules, commitments...) associated with the running speech act type as the main contextual elements invoked. In the case of the order, one of the preliminary conditions is that *Spk* had some authority on the actions done by *Add* , when he is for example his guru, his boss, his baby-sitter or his mother. Speaking agents refers to these elements essentially as argues in the dialogue, it is under this form that we will formalise them. For example, the set of actions associated with this first rule for the order contains the following utterances, which could be used in a real situation of speech acting dialogue :

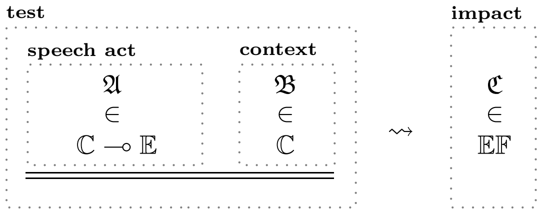
- for *Spk* : %`I have authority on` *`Add`* , %`I'm the boss of` *`Add`* , %*`Add`* `is the child whom I am the babysitter`, %*`Add`* `is one of my children...`
- for *Add* (respectively) : %`I choosen to follow the lead of this man`, %`I'm not working`, %`She is not my mother !`, %`ok mum !`

Definition 2 (Structure of ludical speech acts). *A ludical speech act is defined by three elements :*

- *the* **speech act** *: some competence of the Spk for impacting the context considering some actual conditions or anticipated reactions of its interlocutor. This competence is invoked in the situation as a structured system of actions.*
- *the* **test** *: the interactive situation by which we oppose the speech act and a complex structure which plays the role of context mixing contextual datas with interlocutor reactions*
- *the* **impact** *: which is the result of the interaction, i.e. a modification of the context (inscription, erasing)*

So, we decompose the speech act in three levels : the body of the speech act (i.e. the speech act as we find in the litterature), the speech act in interactive situation (i.e. taking in account the reaction of addressee) and the achieved act (main or peripheral effect). In the following design we assume that the active part take the form of a function modifying context to produce an effect, which is the basic form underlying our models[7]

[7] Please consider that it is not exactly the case for all the speech acts situations we will present in the last section, by the fact that it depends on the type of speech act and the *modus operandi* of the interaction. But, as we say in french mathematics, it is "*morally the same*".



Where $\mathfrak{A}$, $\mathfrak{B}$ and $\mathfrak{C}$ are some interactive trees (called "designs") structured interactively by alternation observation/action. In the case they are *negatives*, they are passive and represents a contextual structure (knowledges, mental states,facts...). When *positives*, they are active trees and represents the structure of an operation (fonctions, transformations...). Each design (like $\mathfrak{C}$) realizes a behavior (here $\mathbb{EF}$) which is a category possessed by the (some) speaking agents. So we can say that the realisation $\mathfrak{C}$ is in the behavior $\mathbb{EF}$. The sequence of transformations between the act and its effects is made by a rule of interaction (given in the following section) which modify the structures of trees.

The speech act strictly considered (*i.e.* seen as a program or a function) is represented here by a design $\mathfrak{A}$ in the behavior $\mathbb{C} \multimap \mathbb{E}$. More exactly, we would write $\bigotimes \mathbb{C}_i \multimap \mathbb{E}$ where the behaviors in $\bigotimes \mathbb{C}_j$ are the conditions associated to the effects $\bigotimes \mathbb{E}_k$.

We recall that a design $\mathfrak{F}$ in a behavior $\mathbb{A} \multimap \mathbb{B}$ is performing effects somehow as a fonction f defined in the set $A \rightarrow B$. If we present to it a design $\mathfrak{A} \in \mathbb{A}$, then the interactive situation $[\![\mathfrak{F} \mid \mathfrak{A}]\!]$ is normalizing in a design $\mathfrak{B} \in \mathbb{B}$, either it fails.

Main and board effects In our model, we could have consider that a failing speech act is formalized by an interaction wich diverges and produces no inscription. Thus, we consider that, even non felicitous, a speech act can produce some inscription, almost an inscription of the fact that it fails. So we propose to base differently our model, taking in charge two types of inscriptions and two types of evaluation.

	evaluation	**inscriptions**
convergence	realization of the speech act	inscription of main effects
divergence	failure of the speech act	inscription of null effect
stages of reduction	processing the speech act	incription of board effects

For example, for the promise, one of the conditions when something is promised by *Spk* is that *Add* prefers the promised to be realized rather it fails. The not-felicity of the speech act in this case will produce an inscription like `%Add don't prefer that ...`, `%Add don't like ...`. This new inscription is available for further speech act as a contextual element. This product, which has the form of a design is not, evidently, in the expected behavior $\mathbb{E}$, which represent the commitment of *Spk* for *Add* , that we call "*main effect*". This is one of the properties of designs which are defined out of the behaviors they belong to. The design $\mathfrak{A}$ of the behavior $\mathbb{C} \multimap \mathbb{E}$, can interacts with a design $\mathfrak{D}$ which does not belong to $\mathbb{C}$. This interaction, underdefined (untyped as we say in lambda-calculus), can performs a convergence with production of some main effect.

The modification of the context is done during the processing (board effects) and/or after the reduction (main effect). These effects take the form of designs representing some heterogeneous datas : commitments, responses, new knowledges...

3 Ludics in a nutshell

As previousy in Linear Logic, J.-Y. Girard (Girard 01) adopts a geometrical point of view of proofs and an internal approach of dynamics ; so Ludics can be sum up as a *interaction theory*.

The objects of the Ludics are no more formulas and proofs, but their geometrical representation, seen as an architectural object ; only what is needed for the interaction is kept. In order to perform this geometrical work, polarized formulas are taken in account. This leads us to create a link (8) between Ludics and Game Semantics which then is a good metaphor for a first approach of Ludics.

The central object of Ludics is the **design**. From the logical point of view, its conception is radically monist : syntaxically, it can be seen as the architecture of a proof (a paraproof), whereas its semantics is the result of its interactions against the others designs.

Instead of formulas we find the addresses (locus) where formulas and subformulas are stored. One of its, the focus, represents the locus where the interaction between designs takes place ; instead of proofs we find designs which can be seen as trees of addresses with rules for building designs. These rules specify either the offered possibilities in a point, to make an action, or the anchorage points that we consider as possible for the reaction. Everything is ready in view of the interaction.

Here we content ourself to give a basic survey of notions needed to understand our purpose. The reader concerned with more details on the mathematical notions and rich concepts of Ludics is recommanded to read the source texts (Girard 01), (14).

3.1 The Design

The **design** is the central object of Ludics ; it can be seen as an infinite tree. By means of the metaphor of Games, a design can be understood as a *strategy*, i.e. as a set of *plays* (**chronicles**), sequences of couples *action* (Player) - *reaction* (Opponent). In the description of a strategy, the point of view of Player is taken in account so that every positive move is the possible action of Player, and negative moves are anticipations of Opponent moves by Player.

In Ludics, the nodes of a design (seen as a tree) are labelled by two sets (Γ, Δ) of addresses (loci) denoted $\Gamma \vdash \Delta$; an address is a finite sequence of integers, for example[8]: $\xi * i$. Roughly speaking, $\Gamma \vdash \Delta$ is an organisation of positions from which the next move can be executed. The root is called the base of the design.

The designs are built by the means of only three rules schemes: two first schemes of rules are issued from logical rules (a positive one and a negative one)[9] and a new one called the damon ($\mathfrak{Dai}$), seen as a "Giving up". This rule does not arise from the logic, but is needed for taking in account the interaction.

Positive action : *to perform an action, to ask, to answer*
By a positive action, the player selects a branching locus and opens all its possible loci for continuing the interaction. He chooses to act in the place ξ and he opens to his interlocutor the range of actions $\xi * i_1 \vdash, \ldots, \xi * i_n \vdash$.

$$\frac{\xi * i_1 \vdash \Lambda_i \qquad \ldots \qquad \xi * i_n \vdash \Lambda_n}{\vdash \Lambda, \xi} \; (\xi, \{i_1, ..., i_n\})$$

Where $\cup \Lambda_i \subset \Lambda$.
REMARK: By playing an empty ramification, Player prevents Opponent from any reaction and so he blocks the continuation of the interaction.

Negative action : *to receive opponent action, to foresee*
Formally, a negative action specifies the ramifications of a directory. In our context, we can say that, in the place ξ, else after one of my action I get ready for receiving reactions of my interlocutor in a list that I have foreseen, or I receive a list of messages sent by my interlocutor.

$$\frac{\vdash \Lambda, \xi * i_1^1, \ldots, \xi * i_n^1 \qquad \ldots \qquad \vdash \Lambda, \xi * i_1^j, \ldots, \xi * j_m^j}{\xi \vdash \Lambda} \; \xi, \{R_1, ..., R_j\}$$

Where $R_k = \{i_1^k, \cdots, i_n^k\}$

Giving up : At any time in a positive context, the active player (the one who has to play a move) can prefer to give up ; so the immediate effect is to stop the interaction.

$$\frac{}{\vdash \Lambda} \maltese$$

Examples 1 ($\mathfrak{Dai}$ and $\mathfrak{Fa}x$).

[8] We will use $*$ as the sign for the concatenation operation.
[9] The underlying logic is the hypersequentialized linear logic which works with polarized synthetic connectives. So only two rules schemes are needed

- $\mathfrak{Dai}^+$ and $\mathfrak{Dai}^-$ *I decide to immediately stop, as soon as I have the hand.*

$$\mathfrak{Dai}^+ = \frac{}{\Delta}\maltese \qquad\qquad \mathfrak{Dai}^- = \cfrac{\cfrac{}{\vdash \xi * I, \Delta}\maltese}{\xi \vdash \Delta}(\xi, \mathcal{P}_f(\mathbb{N}))$$

- The design $\mathfrak{Fax}_{\xi,\xi'}$ will play a crucial role in the following sections. As it is suggested by its name, this design is a sort of echolalic design. $\mathfrak{Fax}_{\xi,\xi'}$ allows us by the means of interaction to move a design $\mathfrak{D}$ localised at the address (locus) ξ into a design $\mathfrak{D}$' localised at the address ξ' ; it is recursevely defined. Rougthly speaking, it imitates the two first actions of the design $\mathfrak{D}$ and so on infinitely. Here $\mathcal{P}_f(\mathbb{N})$ is the set of finite subsets of $\mathbb{N}$; it suggests that all the possible cases are avalaible for the design to be delocalised.

$$\cfrac{\ldots \quad \cfrac{\begin{array}{c}\mathfrak{Fax}_{\xi'*i,\xi*i}\\ \xi' * i \vdash \xi * i\end{array}}{\vdash \xi * I, \xi'}(\xi', I) \quad \ldots \quad \cfrac{\begin{array}{c}\mathfrak{Fax}_{\xi'*j,\xi*j}\\ \xi' * j \vdash \xi * j\end{array}}{\vdash \xi * J, \xi'}(\xi', J) \quad \ldots}{\xi \vdash \xi'}(\xi, \mathcal{P}_f(\mathbb{N}))$$

From now $\underset{\xi \vdash \Delta}{\nabla}$ will symbolize the design (seen as a tree) based on $\xi \vdash \Delta$.

3.2 The interaction

In Linear Logic, the interaction is represented by the cut elimination. In Ludics, the interaction is a meeting between two players strategies localised in a same locus[10]. The **dynamics of the interaction** is given by a process[11] which regulates this meeting. At any step, action-reaction of players are put in coincidence until one of the players gives up or until this coincidence failed or even endlessly continues.

Without entering the formalism, we can say that the interaction between designs is seen as a cut-net, *i.e.* an acyclic finite graph of designs pairwise connected by their bases. For example, let us consider the following cut-net of base[12] $\sigma \vdash \lambda, \rho$:

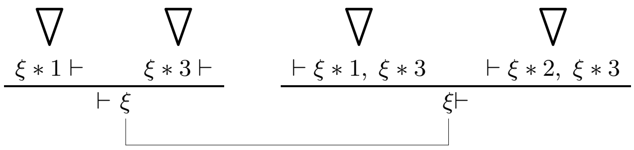
[10] The interaction is concretely translated by a coincidence of two loci in dual positions in the bases of two designs. For example a design of base $\sigma \vdash \xi$ can interact against a design of base $\xi \vdash \rho$.

[11] This process is called the “normalisation procedure” in lambda-calculus, or the “cut elimination procedure” in the sequent calculus, by logicians and computer scientists.

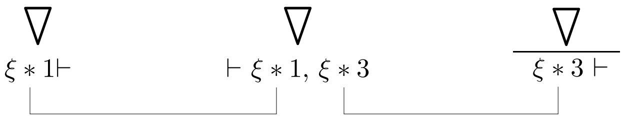
[12] The base of the cut-net is obtained by erasing the cut loci, *i.e.* the loci by which they are connected.

We don't give here a complete description of this procedure, as we just present two views about how it is processing :

Example 3. In this example the base of the cut-net[13] is $\vdash$. We can observe that in this example, the ramification $R = \{1, 3\}$ is in the directory $\mathcal{R} = \{\{1, 3\}, \{2, 3\}\}$



The first reduction step eliminates dead forks (the branches upon an unconnected node) and produce the following net:

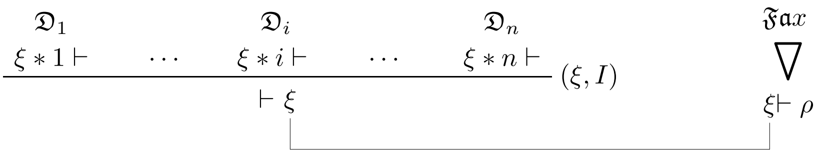


After n reduction steps, there's only three possible configurations :

- *Convergence* by "giving up": $\mathfrak{Dai}$ is one of the designs produced by the reduction, then all the net is reduced to the $\mathfrak{Dai}$.
- *Divergence* : the ramification R of the positive design is not in the directory $\mathcal{R}$ of the negative design.
- *Divergence* because of an infinite interaction : The exchange makes loops or continues infinitely.

Example 4. Here we present the interaction between a design $\mathfrak{D}$ of base $\vdash \xi$ and the $\mathfrak{Fax}$ of base $\xi \vdash \rho$. Let us observe that the base of this cut-net is not empty. In case of convergence, we then obtain a design $\mathfrak{D}'$ of base $\vdash \rho$; the resulting design in fact is similar to the design $\mathfrak{D}$.

To cut a design $\mathfrak{D}$ and the $\mathfrak{Fax}$ enables us to *delocalize* a design, to move it, to modify its place of anchoring.



Two reduction steps produce the design:

[13] such a cut-net is called a closed cut-net

$$\dfrac{\begin{matrix}[[\mathfrak{Fax}_{\rho*1,\xi*1}, \mathfrak{D}_1]] & & [[\mathfrak{Fax}_{\rho*i,\xi*i}, \mathfrak{D}_i]] & & [[\mathfrak{Fax}_{\rho*n,\xi*n}, \mathfrak{D}_n]] \\ \rho * 1 \vdash & \cdots & \rho * i \vdash & \cdots & \rho * n \vdash\end{matrix}}{\vdash \rho} (\xi, I)$$

Finally, let us introduce the notion of orthogonality. It will allow the handling of complete objects named behaviors and presented in the further section. indexobjetOrthogonality

Definition 1 (Orthogonality) *Two designs* $\mathfrak{D}$ *and* $\mathfrak{E}$ *are orthogonal if the reduction of the net* $[[\ \mathfrak{D}\ ;\ \mathfrak{E}\ ]]$ *terminates and produces* $\mathfrak{Dai}$.

3.3 Behaviors

Definition 2 (Behavior) *A behavior* $\mathbb{C}$ *is a set of designs of same base, closed by biorthogonal.*

$$\mathbb{C} = \mathbb{C}^{\perp\perp}$$

Then we can consider the following correspondence:

designs	(para-)proofs
$\mathfrak{D}$	π
behaviors	formulas
$\mathbb{C}$	A

At the abstract level, the behaviors can be composed by the means of the connectives of the LL:

$$\mathbb{C}_1 \otimes \mathbb{C}_2, \quad \mathbb{C}_1 \oplus \mathbb{C}_2, \quad \mathbb{C}_1 \multimap \mathbb{C}_2, \quad \mathbb{C}_1 \wp \mathbb{C}_2, \quad \mathbb{C}_1 \,\&\, \mathbb{C}_2, \quad \cdots$$

and then also be interpreted in interactive manner.

For example the behavior $\mathbb{C}_1 \multimap \mathbb{C}_2$ can be seen as a function which transforms a design in $\mathbb{C}_1$ into a design in $\mathbb{C}_2$. (remark: we also can speak of a design of type $\mathbb{C}_1$) We can also say that when a design $\mathfrak{E}_1 \in \mathbb{C}_1$ interact with a design $\mathfrak{D}$ in $\mathbb{C}_1 \multimap \mathbb{C}_2$, then the result is a design $\mathfrak{E}_2 \in \mathbb{C}_2$:

$$\begin{matrix}\mathfrak{E}_1 & \quad & \mathfrak{D} & & \mathfrak{E}_2 \\ \in & & \in & \rightsquigarrow & \in \\ \mathbb{C}_1 & & \mathbb{C}_1 \multimap \mathbb{C}_2 & & \mathbb{C}_2\end{matrix}$$

Example 5. Let us consider a record in a data base with three fields : coordonnate (cd), shape (sh) and color (col). It can be represented by the following design $\mathfrak{R}$ based on $\xi \vdash$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \xi * cd * 1 * 1}\emptyset}{\xi * cd * 1 \vdash}}{\vdash \xi * cd} \quad \dfrac{\dfrac{\dfrac{}{\vdash \xi * sh * sqr * sqr}\emptyset}{\xi * sh * sqr \vdash}}{\vdash \xi * sh} \quad \dfrac{\dfrac{\dfrac{}{\vdash \xi * col * yell * yell}\emptyset}{\xi * col * yell \vdash}}{\vdash \xi * col}}{\xi \vdash}$$