of truth-conditionals but by means of the construction of proofs that are *syntactical* in nature. In classical proof theory, the meaning of the sentence is the set of its proofs, or, more accurately, knowledge of the conditions of its assertion, which then counts as knowledge of what would count as a proof, and hence the meaning of the sentence. Thus, in determining the meaning of a sentence priority is given not to the notion of truth but to that of proof. Or, from the viewpoint of the pragmatic theory of meaning-as-use, the meaning of a sentence or formula is not explained by their truth, but by their use or consequences in a proof:

$$\frac{\vdash A \quad A \vdash \boldsymbol{B}}{\vdash \boldsymbol{B}}\,\text{cut} \rightsquigarrow \frac{\vdash A \quad A \vdash}{\vdash}\,\text{cut}$$

(Notice that A on the left and the right side of the consequence relation or turnstile can be seen as a duality: proofs of A versus proofs of $A^{\perp}$, where the linear negation $\perp$ signifies the switching of roles or, in this case, swapping the *place* of A with regard to the turnstile.)

The shortcoming of the classical meaning-as-proof paradigm is that the proof is conceived statically and monologically, in tandem with the mathematical interpretation of classical logic. For example, in this classical setup, given A (a formula, a logical expression, a piece of syntax, a proposition), if one has the proof—the meaning of A—then one also has the meaning of $\neg A$. That is to say, having the proof implies having the disproof by way of the classical negation that negates some unspecified or arbitrary iteration of A. But with the introduction into classical proof theory of interaction (i.e., dualities as the interchange of roles rather than as classical negation), the proof or determination of meaning takes on a different form. The meaning of A can only be determined through interaction or dialogue with its counter-proof or refuter, and vice versa. The determination of meaning or proof can only be achieved by stepping outside of the static-monological framework of proof into one where proof is the interaction between a prover (player) and a refuter (opponent), A and $\neg A$, or, more precisely, A and the model of $\neg A$.

Just like the environment in the classical Church-Turing paradigm of computation, where it was presupposed but never explicitly asserted, in its classical form $\neg A$ is rather a passive and extrinsic piece of information in relation to A. In the interactive framework, however, $\neg A$ is an intrinsic dimension of the proof or meaning of A. The proof of A rests on the proof of $\neg A$ in the context of an interaction or game played by the proponent P and the opponent $\neg A$. In this approach, however, classical negation is abandoned in favour of interactional dualities, with negation as the interchange of roles between the proofs of A and $A^{\perp}$ where the absurdity or contradiction or *falsum* sign $\perp$ is a linear negation that expresses duality (e.g., $A \multimap B = B^{\perp} \multimap A^{\perp}$):

Action of type A = Reaction of type $A^{\perp}$

In this fashion, even logical connectives can be expressed as moves in interaction games. For example, in the classical Lorenzenian version of game semantics,[264] conjunction ($p \wedge q$) and disjunction ($p \vee q$) can be expressed in terms of how the game proceeds when compound propositions constructed by conjunction or disjunction are attacked or defended, questioned or asserted. To challenge/question the conjunction, the opponent may select either conjunct, while the proponent can only defend/assert the conjunct that the opponent has selected, while to challenge the disjunction, the opponent may ask the proponent to select and defend one of the disjuncts. Similarly, to challenge the implication ($p \rightarrow q$) the opponent may assert p. Then the proponent must either challenge p or assert/defend q.

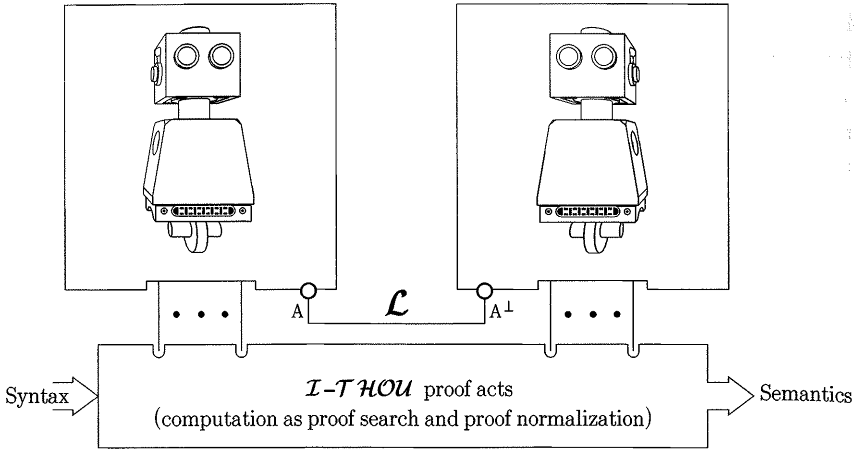
Reconceived within a dialogical framework, proof is no longer an object in the way it is classically understood, but an *act*. As an act, proof implies the confrontation of acts, i.e., interaction—in the computational sense introduced above—between dualities that can be represented as agents, processes, or strategies. What dialogical interaction affords is not merely the computation of meaning or semantic values, but also semantic information such as the contextuality and thematization necessary for the determination of meaning. In the dialogical paradigm of meaning as proof or use, semantic values and contexts are computations.

264 K. Lorenz and P. Lorenzen, *Dialogische Logik* (Darmstadt: WBG, 1978).

Depending on the specificity of a dialogical interaction, computation can be understood either as proof search or proof normalization. In the proof search schema, if the interaction game returns the proof of a sentence or formula, the meaning is computed. Otherwise, it searches forever and fails to obtain the proof (cf. the halting problem), i.e., the dialogue will not be able to determine the meaning. In the normalization schema, the proof or meaning can be computed if, by the elimination of unnecessary steps, transitions, or rules (useless detours) for a formula or sentence, the normal form or canonical proof can be obtained. This normal form is the simplest i.e. most irreducible proof obtainable without using unnecessary detours (cf. normal form in abstract rewriting as an object that cannot be transformed any further).[265] In this setting, two statements have the same meaning if their proofs or constructions can count as equivalent in so far as there is a reversible inferential relation or mapping between them.[266]

265 For a term A, some $x \in A$ is a normal or simplified form if no $y \in A$ exists such that $x \to y$ (i.e. x cannot be written further). Computability can be understood in terms of a rewrite system or process of normalization that terminates once it reaches a strongly normalized term, i.e., a term that cannot be reduced or simplified any further.

266 If we were—in accordance with the correspondences between proofs, programs, and mathematical structures—to take a proposition A as the topological space of proofs, then the proofs M and N can be thought of as points in this space. Consequently, the paths between these points can be considered as equivalence relationships. Suppose that both M and N are proofs of A, denoted as $M{=}N{:}A$. Now there is a mapping or path α that can deform M to N ($\alpha{:}M \rightsquigarrow N{:}A$) and vice versa. The existence of such a path can be understood as the existence of a piece of evidence—a particular mental object or *gegenstand*—that falls under both constructions M and N. In Kantian terms, M and N are *concepts* which cover the same object. Moreover, the existence of such a path or equivalence relationship implies the existence of equivalences of equivalences, or concepts of the same concept type. Such higher equivalences are called homotopies, which are basically the deformation maps of deformations between M and N. See Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics* (2013), <https://homotopytypetheory.org/book/>.



Toy meaning dispenser

In the universe of the automata, the interactive schema of meaning-as-proof can be thought of as a toy meaning-dispensing machine. The machine consists of two agents plugged into an interaction over a language $\mathcal{L}$. The interaction modelled on a two-person game—more specifically, an elementary I-Thou dialogue in which, from the perspective of one agent, it is the system and the other agent is the environment. Inside this interactive machine, there are algorithms which obtain proof either through normalization or search. Insert syntactic tokens or terms into this machine and the machine dispenses meaning. Composed of syntactic structures, the semantic value or meaning has a normal form and is invariant with respect to the interactive dynamics inside the machine. Give syntactic terms of the same type to the vending machine, and you should be able to get the same outcome.

A similar interaction machine can be imagined for transforming the minimal generative syntax of the kind our automata already have into the concrete syntax structures required for sentential composition. The configuration of the machine remains intact, but the emphasis of the machine operation or proof acts will be on resource-sensitivity in handling syntactic tokens. In this machine, raw chunks of syntax go in and the sentential syntactic units whose dependencies underlie semantic sentential dependencies come out.[267] The products of the machine are precisely those syntactic structures that support semantic compositionality (the fact that sentences can be decomposed into more elementary constituents). However, syntax at the level of sentential structure is deeply resource-sensitive. This is particularly the case for the syntactic binding of sentences where the meaning (i.e., referent) of an item such as a pronoun can only be determined by the connection it has at any instant with another item in the sentence. Such linked items whose meaning or reference cannot be determined by themselves—which is usually the case for ordinary lexical entries—are known as anaphoric elements. The need to keep track of anaphoric dependency relations between different uses of words in a sentence or between sentences is directly tied to the resource-sensitivity of syntactic relations.[268]

In the sentence, 'Kanzi saw a monkey touching the monolith, and it also heard the monkey screaming as it touched it,' the anaphoric use of the pronoun *it* is dependent on tracking its point of reference as well as its iterations within the sentence. Does 'it' refer to Kanzi, the monkey, or the monolith? This anaphoric binding is essentially resource-sensitive in the sense that it cannot be seen as a simple transition $Kanzi \rightarrow it$ in which 'it' can be obtained by repeated iterations of 'Kanzi' (i.e., all iterations of 'it' in the sentence do not necessarily refer to 'Kanzi'). Instead the transition should be seen as a resource-sensitive linear implication $Kanzi \multimap it$ where 'it' uses 'Kanzi' as its point of reference (resource) *exactly* once. After that, the syntactic-semantic relations between other iterations of 'it' and other antecedent nouns in the sentence will have to be checked to determine whether or not 'Kanzi' can be reused as point of reference or resource for 'it'. Such resource-sensitive connections require logical operators—such

267 For a proof-theoretic account of syntax at the level of syntactic structures of sentences see Lecomte, *Meaning, Logic and Ludics*, 33–52.

268 See G.-J. Kruijff and R. Oehrle (eds.), *Resource-Sensitivity, Binding and Anaphora* (Dordrecht: Springer, 2012).

as exponentials in linear logic—to control and keep track of the use of syntactic resources in sentences.[269]

KEEP IT IN FOCUS: A DIALOGUE IN EIGHT ACTS

Earlier in this chapter, the problem of the syntax-semantics interface was addressed. The solution to this problem was put forward in terms of dialogical interaction as a logico-computational condition for the possibility of meaning. But in order for our automata to have dialogues in which the rules of language are not set in advance, a new type of dialogue must be introduced—one in which rules emerge through the interaction itself. The form of such a dialogue corresponds with the general category of games G, discussed earlier. One may call such a dialogue an interaction with no preinstalled normative bells and whistles. Dialogue, in this sense, counts as the generalization of the logico-computational notion of interaction as the bridge between syntax and semantics. According to Samuel Tronçon et al., a dialogue has at least three essential functions—exchange of information (i.e., computation in the sense discussed earlier), construction of knowledge, and resolution of cognitive tensions (i.e., harnessing the behaviours of interlocutors toward new behaviours):

> First, at every stage, a speaker is giving a symbol, and this exchange is informative in three ways: it informs us about the discussed (some

269 Other than the resource-sensitive linear implication $A \multimap B$ (reads as 'A yields B and is consumed in the process') in which B is yielded by the use of exactly one iteration of A, there are logical operators in linear logic that directly handle the permissions on the use of resources or formulas. These exponentials are !! and ??, which are respectively called exponential conjunction (reads as 'of course' operator) and exponential disjunction (reads as 'why not' operator). In the interaction framework, roughly, $!A$ is defined as a permission for the system/player to use or access the iterations of the formula, hypothesis, or resource A without restriction. Whereas $?A$ is defined as a permission given by the environment to the system to use A: The system asks 'can I use A?', and the environment or the interlocutor may reply 'why not'. Subsequent use of or access to A would again require the permission of the environment/interlocutor.

> thesis), the subject that is speaking (his approach about this thesis), and the connection between a present intervention and some counter-interventions (upstream or downstream, actuals or virtuals). Second, running dialogues shows arguments interacting like machines built up to explore relevant opportunities of discussion according to some global strategy: *I argue in this way to reach this point, I open these branches to induce some reactions*.... So, dialogue is a sort of unfolding structure that represents some knowledge. Evidently, involving friendly but tenacious interlocutors ensures a good (exhaustive) exploration. Third, by the interaction, the locutors can extract some new information which is about the form of the interaction, contained in the result of the dialogue: what is stable, what is explored, what is new, what is in latence.[270]

This multilayered view of dialogue can be modelled on the general dynamics of interaction in such a way that the functions of dialogues mirror the deep properties of logic and computation. One example of such a model is Jean-Yves Girard's *ludics*, which at a paradigmatic level of interpretation reflects the logic of dialogue, a linguistic interaction in which not only can syntax be bridged with semantics, but the rules of language naturally emerge as part of the dialogue.[271]

Girard's ludics can be characterized as a pre- or proto-logical framework for analysing logical and computational phenomena at the most elementary level. Defined at the intersection between linear logic (a substructural logic capable of capturing the duality of interaction and the interchange of roles), proof theory, game semantics, and computer science, ludics introduces a logico-computational framework in which, at the *deepest level*, the distinction between syntax and semantics collapses. Yet this continuity

---

270 M.-R. Fleury, M. Quatrini, and S. Tronçon, 'Dialogues in Ludics', in *Logic and Grammar* (Dordrecht: Springer, 2011), 138–57.

271 For an exquisitely engaging introduction to Girard's project in a philosophical context see O.L. Fraser, *Go Back to An-Fang* (2014), <http://www.academia.edu/352702/Go_back_to_An-Fang>.

between syntax and semantics is established by deviating from a number of traditional approaches such as referential theories of meaning, monological conceptions of semantics, non-autonomous approaches to syntax, and the traditional forms of game semantics conditioned on game-theoretic concepts such as predetermined winning strategies, preference ranking, payoff functions, and referees. In ludics, these are replaced by an inferential theory of meaning, dialogical and operational semantics, an autonomy of minimal syntax, and a general notion of game devoid of any predetermined winning strategies or payoff functions.

Ludics shows that the continuity between syntax and semantics is naturally achieved through an interactive stance toward syntax in its most atomic and naked appearance: the trace of the sign's occurrence, the *locus* or place of its inscription. Semantics immanently unfolds through the dynamic impact of the most minimal appearance of syntax—its locus—stripped of all preformulated rules and metaphysical references. Different stances toward an atomic syntactic expression are represented as interacting strategies (called *designs* in ludics) that are tested not against a preestablished model, but against one another. In the process, the rules of logic (or alternatively, the forms of thought) emerge from the confrontation of strategies whose element of interaction is a positive or a negative locus rather than a proposition. The positive and negative locus refer to the address of the sign/formula in interaction, where an act is exercised either in response to, or as a request for, another act on the same locus but played according to the opposite strategy. The exchange via the relation between a locus and its polar counterpart corresponds with responses and demands between interlocutors in a dialogue, or players in a game.

Let us briefly examine the most basic elements of ludics as the logic of dialogue:[272]

- Designs: As the central objects of ludics, designs can be identified as strategies in a game, namely, sets of plays (or in ludics, *chronicles*)

272 See J.-Y. Girard, 'Locus Solum: From the Rules of Logic to the Logic of Rules', *Mathematical Structures in Computer Science* 11:3 (2001), 301–506.

distinguished by the answer or defence of the player against the attacking move or query made by the opponent. Less in traditional game terms and more in ludics terms, designs are alternating sequences of positive and negative rules which may either progress endlessly (as in endless dispute) or be closed by a rule called *daïmon*.

- Plays (dialogical equivalents of proofs) are alternating sequences of moves or actions.
  - Moves are defined in terms of a tuple $\langle p,l,r \rangle$ where
    - p is the local polarity of moves (positive polarity for the actions or moves of the player and negative polarity for the moves of the opponent).
    - l is a *locus* or fixed position to which a move is anchored. Loci can be thought of as memory cells or places in which formulas are stored. In Ludics, formulas or syntactic terms are replaced by the address of their locations in the interaction/dialogue. In other words, all that matters are the locations of syntactic terms or formulas—not formulas themselves—and how they are manipulated within a dialogue or proof.
    - r is a finite number of positions which can be reached in one step and which is called *ramification*.
  - Positions are *addresses* of loci encoded by finite sequence of integers and usually denoted by the letters ξ, ρ, σ, ... which stand for the threads and subthreads of the interaction (loci and subloci which represent thematization of a dialogue across different topics). Trees of addresses are equated with designs. For example, in an absurdist dialogue between Kanzi and Sue, addresses can be encoded in the following manner. In the contexts ξ and σ two trees of alternating sequences emerge:

Kanzi: Did you see that grey stuff? (ξ.0)
Sue: Yes. (ξ.0.1)
Kanzi: Have you noticed it makes noise whenever it finds something new? (ξ.0.1.0)
Sue: It often does. (ξ.0.1.0.1)
Kanzi: Why is that you are clicking too much today? (σ.0)
Sue: I have no idea what you are talking about. (σ.0.1)

— The starting positions, also known as *forks*, are denoted in the sequent form as $\Gamma \vdash \Delta$ where Γ and Δ are either singletons or empty sets. In this setting, when an element belongs to Γ every play on this element starts with the opponent/environment's move and the fork is said to be negative, whereas if the player/system starts with an element of its choice taken in Δ the fork is said to be positive.

In this form, a design can be represented as a tree whose nodes are forks $\Gamma \vdash \Delta$ and whose threefold root is comprised of the positive rule, the negative rule, and the *daïmon*.

- Rules: The only rules *necessary* for building designs or strategies are *daïmon* plus negative and positive rules, which are indexed by a property called focalization. Roughly, focalization can be thought of as proof search in an interactive framework: it allows for the grouping of a series of consecutive actions (i.e., plays as proof) of the same polarity as if they were a single logical action in a proof. That is to say, focalization enables the alternating dynamics of positive and negative actions. In ludics, focalization represents proofs as interactive processes between two players where the player/proponent can choose what to do (positive action) or receives and acknowledges the action or move performed by the opponent (negative action). From this perspective, focalization creates a proof tree in which rules or logical connectives are captured by alternating sequences of positive and negative actions which switch loci for subloci and vice versa across the consequence relation ($\vdash$): $\vdash\xi$, $\xi.0 \vdash$,

$\vdash \xi.0.1, \ldots$ The meaning or semantic value of rules is obtained from the interaction of such consequence relations qua *use* in the dialogue.

- The positive rule can be denoted in the sequent form as:

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

  where I is a finite set of integers (possibly an empty ramification) such that, for every pair of indices $(i, j) \in I$, Δ_i and Δ_j are disconnected and every Δ_i is included in Δ.

- Correspondingly, the negative rule can be denoted in the sequent form as:

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

  where $\mathcal{N}$ is a possibly empty or infinite set of finite set of integers (ramifications) such that instances of Δ_I—not necessarily disconnected—are included in Δ.

- In addition to positive and negative rules which range over loci and subloci, there is also a general axiomatic rule called *daïmon* symbolized by †, which can be understood as a paralogism. Whichever player or proof-process ($\vdash A$ or $\vdash A^{\perp}$) invokes the *daïmon*, the other player wins. In other words, the *daïmon* allows for identification of the winning strategy, the correct proof, or the validity of a proposition. A proposition can be said to be valid for a given design if that design never invokes *daïmon* before its opponent. As such, the *daïmon* can be identified with the conceding act of 'I give up' or 'Stop!' (terminating a design or closing off the branches of a proof tree) in a dialogue. Additionally, *daïmon* is a positive rule, meaning that either player can choose to invoke it. *Daïmon* is represented as a sequent with no premises:

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

Here, what we are interested in is the *dialogical interpretation* of the objects of ludics in a way that allows us to understand dialogue as a dynamic process—interactive proof or computation—in which rules and meanings spontaneously emerge throughout the course of the conversation (i.e., simply through the exchange of loci and subloci—addresses of syntactic terms or formulas—and in the presence of most elementary rules of negative and positive actions plus the *daïmon* or termination command). According to this dialogical interpretation, for a given dialogue, the conversation progresses from whatever locus the polar strategies (players) choose to focus on.

Again in an imaginary dialogue between Kanzi and Sue, this can be interpreted as the first move made by Kanzi—for example, an utterance regarding the presence of a fuzzy item (the focus $\xi.0$). If Sue acknowledges Kanzi's utterance as a speech act, it may reply with an utterance on the same focus ($\xi.0\,.1$). Then the dialogue can be said to be developing along a selected context Λ_0 in which the theme ξ (the presence of fuzzy grey item) is developed. Contextualization and topicalization in a dialogue can, therefore, be captured by alternating sequences of positive and negative actions.[273] For a given locus or address ξ (focus of the dialogue), the context Λ_0 can be written as $\xi = \tau * 0 * 0,\ \tau * 0 * 1,\ \ldots,\ * \tau * 0 * n$. If we view the design from a top-down perspective or in terms of pre-steps, the topic of the conversation can be said to be an operation that unifies the focus and the context, i.e. Kanzi's first move together with the acknowledgement of that move by Sue:

$$\cfrac{\cfrac{\vdash \tau * 0 * 0, \ldots, \tau * 0 * n = \Lambda}{\tau * 0 \vdash}(-, \tau * 0, \{I\})}{\vdash \tau}(<>, +, \{0\})$$

In this fashion, thematic variations (defined as a set of sets of loci) are developed in a dialogue as polar strategies give different addresses (loci) to an expression. A new locus can be accepted by the competing strategies or designs (corresponding with the agreement of interlocutors on a specific way of addressing the initial topic of the conversation) and introduced as

273 See A. Lecomte and M. Quatrini, *Dialogue and Interaction: The Ludics View* (2014), <http://iml.univ-mrs.fr/editions/publi2010/files/Quatrini_Lecomte-esslli.pdf>.

the focus of the exchange. This is the process of (bilateral) focalization in ludics, which maps the actions played in one strategy (the behaviour of a design) to its polar counterpart and projects them back. In doing so, focalization reveals logical constants and rules that are not a priori given, but adaptively emerge in the absence of any gain function or external referee.

Moreover, focalization progressively topicalizes the interaction and drives expressions into a state of context-sensitivity in which syntactic terms or formulas acquire semantic value by virtue of how they are thematized in the same context (focalization on the same set of addresses) as well as how the context is updated in the dialogue (sets of designs or proof acts exchanging the same locations and sub-locations). This is equivalent to Brandom's inferentialist-pragmatic theory of meaning where the contextualization and updating of the context of expressions by the inferential relations obtained between them count as the inferential articulation of expressions that genuinely confer conceptual content on them.[274] Like normative scorekeeping pragmatics, ludics adopts a conception of meaning that is not representational—i.e., it does not relate to a nonlinguistic item—but is instead conferred by the reciprocation between different competing strategies or interlocutors. Semantics immanently arises as the normalization of clashing strategies or processes executed at the level of syntax. In other words, the interaction between different stances toward an expression (a location) can be seen as the process of computing meanings or, in proof theoretic terms, as the process of locating proofs or meanings of expressions.

274 'The connection between the normative scorekeeping pragmatics and the inferentialist semantics is secured by the idea that the *consequential* scorekeeping relations among expression-repeatables needed to compute the significance updates can be generated by broadly *inferential* relations among those expression-repeatables. The theory propounded in *Making It Explicit* is that there are six consequential relations among commitments and entitlements that are *sufficient* for a practice exhibiting them to qualify as *discursive*, that is, as a practice of giving and asking for *reasons*, hence as conferring *inferentially* articulated, thus genuinely *conceptual* content on the expressions, performances, and statuses that have scorekeeping significances in those practices.' R. Brandom, 'Conceptual Content and Discursive Practice', in J. Langkau and C. Nimtz (eds), *New Perspectives on Concepts* (Amsterdam: Rodopi, 2010), 20.

As the logic of dialogue, ludics provides an interactive articulation of speech acts. But unlike the classical view of speech acts, where the speakers' intention to communicate a propositional content is normatively predefined, in ludics speech acts evolve naturally through interaction. Interacting moves or actions (partial functions in the computational game framework) attain the status of speech acts (such as assertion and questioning) once they produce, in a shared context, invariant impacts on the environment, i.e. the addressee from the perspective of the speaker and the speaker from the perspective of the addressee. Here, invariance means that the execution of such actions or dialogical interventions on variable expressions in the same shared context or condition (data from the computational game standpoint) always yields the same observed impact for both speaker and addressee. In this sense, ludics allows speech acts to be defined in terms of designs (strategies of the speaker), locations, positive and negative rules (respectively, executing dialogical interventions and recording or anticipating the interventions of the addressee/interlocutor). As elaborated by Tronçon and Fleury, a ludics definition of speech can be laid out in terms of three elements:[275]

1. The speech acting competence $\mathfrak{A}$ of the speaker to impact (denoted by a positive rule) the context (interactions on the same loci) given the anticipated reactions of the addressee (denoted by a negative rule).
2. The test, which is an interactive situation that contrasts the speech act with a complex structure representing the context $\mathfrak{B}$ that mixes or interacts contextual data with the addressee's reactions. Put simply, the test is the interaction between speech acts and context.
3. The impact $\mathfrak{C}$, which is the effect e of the interaction: the updating, modification, or erasing of a shared context c.

The speech act can then be understood as the interaction of two designs $\mathfrak{A}$ and $\mathfrak{B}$, resulting in a new design $\mathfrak{C}$. When negative or passive ($\mathfrak{B}$), actions

275 M.-R. Fleury and S. Tronçon, 'Speech Acts in Ludics', in A. Lecomte, S. Tronçon (eds.), *Ludics, Dialogue and Interaction* (Dordrecht: Springer, 2011), 1–24.

or observations represent a contextual structure c_i. And when positive ($\mathfrak{A}$), actions or observations represent active design-trees which are operations, functions, or transformations (modification or erasing) in the context c_i such that $c_i \multimap e$. The impact or significance of the speech act (the design $\mathfrak{C}$) can, therefore, be expressed as an invariant behaviour which, for every c_i, yields or produces the effect e in exactly that context. Thus the speech act is nothing but its impact in a dialogue, and this impact is essentially the normal form resulting from the normalization of two interacting designs $\mathfrak{A}$ and $\mathfrak{B}$ (proof and counter-proof).

With these preliminary remarks, we can see how ludics as interactive logic or the logic of dialogue is the generalization of what Brandom calls deontic scorekeeping, the game of giving and asking for reasons. Let us imagine a hypothetical dialogue in the style of ludics and deontic scorekeeping between Kanzi and the monkey. The monkey's noises are mapped onto Kanzi's sentences, and vice versa, in such a way that both agents can recognize what they say in virtue of the interlocutor's evidence for or against it.

Let Kanzi be the asserter/speaker *A*, and *B* its interlocutor, or the addressee. *A* is in possession of a set of markers (sentences) in such a way that playing a marker counts as claiming or asserting something. These markers are differentiated into two classes, those kept and those discarded. The record of such markers is kept either in a general notebook shared by both *A* and *B*, or in personal notebooks belonging to *A* and *B*. The markers kept and labelled by *A* count as the score of the speaker. If *A* plays a new marker—making an assertion—it changes its own score and maybe its interlocutor's. Now, provided that some possibilities have been presented by *B* as entitlements to commitments (i.e., reasons to believe that *p*), A may play a marker by asserting or endorsing a claim, e.g., 'This monolith is black'. Additionally, we assume that *A* keeps a database or directory of entitlements, i.e., it records the reactions of *B* toward its commitment 'The monolith is black'. In such a scenario, we can imagine a dialogue *focused* on the 'colour' of the monolith, and which consists of eight acts tagged by digits. In the simplest form, they are tagged by 0 and 1 denoting the negation and affirmation of the predicate. Other digits could be introduced, for example, to express modalities. The dialogical acts progress in the following manner:

Act 1: *A* chooses an object $\{j\}$—a singular term—among the set of many such objects (cf. the choice of playing an opening move from a set of possible alternative moves). The choice of this object has the pragmatic significance of being entitled by the interlocutor to make the assertion *p* or in this case, the entitlement to address some theme (the colour of the monolith).
Act 2: Thus having chosen to speak of some definite object, *A* entitles *B* to treat this object in terms of the range of properties that are generally associated with it (e.g., black as a colour, not of light colour, etc.).
Act 3: In return, *B* entitles *A* to choose and play a property $\{k\}$ from this range.
Act 4: On the permission of *B*, *A* now chooses a property and entitles B to regard or act on this property within a range of values (e.g., shiny, opaque, textured black).
Act 5: *B* entitles *A* to choose a value $\{c\}$.
Act 6: *A* picks up a value and entitles *B* to act on it within a set of modalities which can simply pertain to truth and falsity.
Act 7: *B* entitles *A* to pick up a modality.
Act 8: *A* chooses a modality and waits for *B*'s acknowledgement.

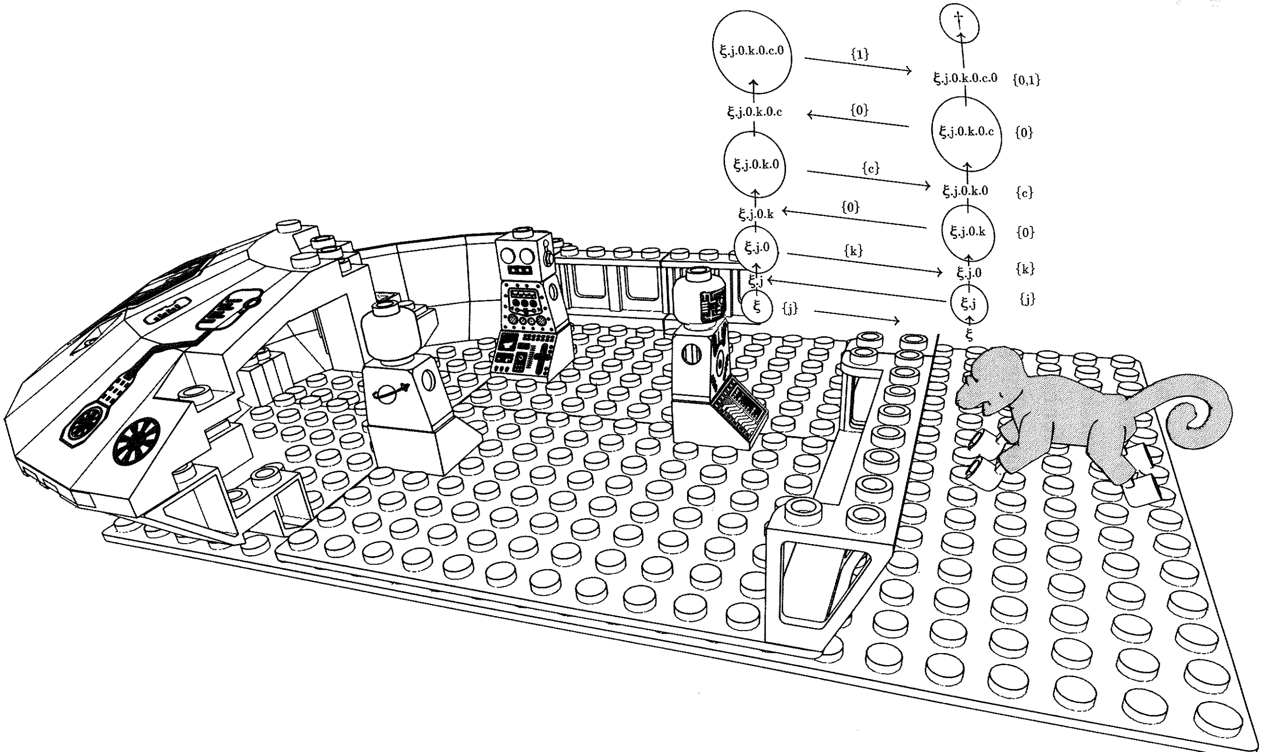
The dialogue may conclude once *B* invokes the daïmon. The entire conversation can be represented as interacting design-trees where all that is required to keep and update the scores of the speaker and the addressee so that their acts can be regarded as sufficiently discursive is the method of tracking how the locations or addresses of the object $\{j\}$, together with its range of properties and values, are exchanged and swapped from one side to another. The dialogue can be represented in tree form with the speaker on the left side and the addressee on the right side. The designs of *A* and *B* also can be written in sequent form as follows:

A:

$$\dfrac{\vdash \xi 1 \ldots, \qquad \dfrac{\dfrac{\ldots, \xi.j.0.k, \ldots}{\xi.j.0 \vdash}\,(-, \{\{1\}, \ldots, \{k\}, \ldots, \{m\}\})}{\vdash \xi j}\,(+, \xi.j, \{0\}) \qquad , \ldots, \xi n}{\xi \vdash}\,\mathcal{N}$$

B:

$$\dfrac{\dfrac{\dfrac{\xi.j.0.k \vdash}{\vdash \xi.j.0}\,(+, \xi.j.0, \{k\})}{\xi.j \vdash}\,(-, \{\{0\}\})}{\vdash \xi}\,(+, \xi, \{j\})$$



A dialogue in eight acts

Ludics provides a framework in which the logico-computational phenomena implicit in the pragmatic dimension of language as the bridge between syntax and semantics, symbol-pushing and inferential articulation, come to the foreground. However, unlike the pragmatic theory of meaning-as-use—even in its Brandomian version, which significantly differs from its original Wittgensteinian form—for ludics there is no preestablished space of reasons into which one graduates. The generation of rules and the capacity to reason are inconceivable without interaction, and are inseparable from the complex contexts that arise throughout its course. However, contra Sellars, this interaction is not a matter of acquaintance with norms as a matter of cultural evolution; and contra Brandom, it is not an index of a substantive sociality of reason. It is rather the very formal condition of language and meaning—a logico-computational dynamics that realizes the syntax-semantics interface. The substantive sociality of reason is built on this formal condition, not the other way around.

FORMALISM AND THE PURE AUTONOMY OF THE FORMAL

This formal condition in which computation, logic, and mathematics intersect is exactly what we can call the pure transcendental condition—that is, a formal dimension at last liberated from Kantian transcendental logic anchored in the provincialism of the apperceptive I and the particular transcendental types which shape and constrain the experience of the subject. And it is in fact this formal dimension—as captured implicitly by natural language and explicitly by interactive logics and artificial languages—that grounds transcendental logic as the science of pure understanding and rational cognition—or simply the concept of mind in relation to the world. The subject has experiences (i.e., it can access the *content* of its experiences) because there is a formal dimension—call it language-as-interaction-as-logic-as-computation—that permits the semantic structuration of such experiences. Without it, there would be no experience and no understanding.

Yet the qualifier 'formal' should not be construed as a reinscription of form versus content or, more precisely, the dichotomy of syntactic form and semantic content. *Both* form and content belong to language as a formal dimension. Moreover, the interactionist view of logical syntax is itself 'semantic in disguise'. While the form-content dichotomy can be seen as still operative in the context of the classical picture of formal languages—a problem that is often raised by proponents of language who see it solely in terms of natural languages—as we observed, it is not a tenable index for the characterization or distinguishing natural languages from formal languages, particularly in the wake of developments in the interactive paradigms in computation and logic. Formal languages are better candidates for the articulation of language qua logic as an organon than natural languages, not merely because they can possess more expressive powers by virtue of their unbound syntactic complexity (i.e., the Carnapian view of language as a general calculus), but more importantly because they explicate interaction as the formal condition of language. In this way, the specific picture of formal languages developed in the interactive paradigm averts the risk of grasping the nature of language and linguistic practices through a vaguely metaphysical account of sociality which results either in seeing linguistic practices as one generic social practice among others or in an inflationary social account of language and reason.

From this perspective, the designation 'artificial' for an artificial language founded on the logic of interaction no longer implies a transcendental lack in comparison to natural language—a lack that can supposedly be overcome through better imitations of natural linguistic behaviours. On the contrary, the artificial, as that which has the capacity to range over broader logical and computational behaviours, is what expresses the transcendental dimension—i.e., the autonomy of logic and language over experiential content—in its pure and unrestricted form. Said differently, the designation 'artificial' in 'artificial languages' implies the possibility of unbinding the formal as that which structures content.

Put simply, artificiality, in this sense, does not signify an inferiority to the natural or something that vainly attempts to mimic the behaviours of natural language. It is rather the case that the 'natural' in 'natural languages' designates a subset of the designation 'artificial' in 'artificial languages'. The behaviours of natural language, its capacity for the inferential articulation of content, of syntactic and semantic structuration, merely represent a narrow