Theoretical Computer Science 50 (1987) 1-102
North-Holland

LINEAR LOGIC*

Jean-Yves GIRARD

Équipe de Logique Mathématique, UA 753 du CNRS, UER de Mathématiques, Université de Paris VII, 75251 Paris, France

Communicated by M. Nivat
Received October 1986

A la mémoire de Jean van Heijenoort

Abstract. The familiar connective of negation is broken into two operations: linear negation which is the purely negative part of negation and the modality "of course" which has the meaning of a reaffirmation. Following this basic discovery, a completely new approach to the whole area between constructive logics and programmation is initiated.

Contents

* Because of its length and novelty this paper has not been subjected to the normal process of refereeing. The editor is prepared to share with the author any criticism that eventually will be expressed concerning this work.

"La secrète noirceur du lait..."

(*Jacques Audiberti*)

I. Introduction and abstract

Linear logic is a logic behind logic, more precisely, it provides a continuation of the constructivization that began with intuitionistic logic. The logic is as strong as the usual ones, i.e., intuitionistic logic can be translated into linear logic in a faithful way. Linear logic shows that the constructive features of intuitionistic operations are indeed due to the linear aspects of some intuitionistic connectives or quantifiers, and these linear features are put at the prominent place. Linear logic has two semantics:

(i) a Tarskian semantics, the *phase semantics*, based on the idea that truth makes an angle with reality; this 'angle' is called a phase, and the connectives are developed according to those operations that are of immediate interest in terms of phases.

(ii) a Heytingian semantics, the *coherent semantics*, which is just a cleaning of familiar Scott semantics for intuitionistic operations, followed by the decomposition of the obtained operations into more primitive ones.

The connectives obtained are

- the *multiplicatives* ($\otimes, \parr, \multimap$) which are bilinear versions of "and", "or", "implies";
- the *additives* ($\oplus, \&$) which are linear versions of "or", "and";
- the *exponentials* (!, ?) which have some similarity with the modals $\Box$ and $\Diamond$ and which are essential to preserve the logical strength.

All these connectives are dominated by the linear negation $(.)^\perp$, which is a constructive and involutive negation; by the way, linear logic works in a classical framework, while being more constructive than intuitionistic logic. The phase semantics is complete w.r.t. linear *sequent calculus* which is, roughly speaking, sequent calculus without weakening and contraction. Since the framework is classical, it is necessary, in order to get a decent proof-theoretic structure, to consider *proof-nets* which are the natural deduction of linear logic; i.e., a system of proofs with multiple conclusions, which works quite well.

One of the main outputs of linear logic seems to be in computer science:

(i) As long as the exponentials ! and ? are not concerned, we get a very sharp control on normalization; linear logic will therefore help us to improve the efficiency of programs.

(ii) The new connectives of linear logic have obvious meanings in terms of parallel computation, especially the multiplicatives. In particular, the multiplicative fragment can be seen as a system of communication without problems of synchronization. The synchronization is handled by proof-boxes which are typical of the additive level. Linear logic is the first attempt to solve the problem of parallelism *at the logical level*, i.e., by making the success of the communication process only dependent of the fact that the programs can be viewed as *proofs* of something, and are therefore sound.

(iii) There are other potential fields of interest, e.g., databases: the use of classical logic for modelling automatic reasoning has led to logical atrocities without any practical output. One clearly needs more subtle logical tools taking into account that it costs something to make a deduction, a guess, etc.; linear logic is built on such a principle (the phases) and could serve as a prototype for a more serious approach to this subject.

(iv) The change of logic could change the possibilities of logical programming since linear negation allows a symmetric clausal framework, free from the usual problems of classical sequent calculus.

II. Linear logic explained to a proof-theorist

There is no constructive logic beyond intuitionistic logic: *modal logic*, inside which intuitionistic connectives can be faithfully translated, is nonconstructive; various logics which have been considered in the philosophical tradition lack the seriousness that is implicitly needed (a clean syntax, cut-elimination, semantics, etc.) and, in the best case, they present themselves as impoverishments of intuitionism, while we are of course seeking something as strong as intuitionism but more subtle. The philosophical exegesis of Heyting's rules leaves in fact very little room for a further discussion of the intuitionistic calculus; but has anybody ever seriously tried? In fact, linear logic, which is a clear and clean extension of usual logic, can be reached through a more perspicuous analysis of the semantics of proofs (not very far from the computer-science approach and thus relegated to the next section), or by certain more or less immediate considerations about sequent calculus. These considerations are of immediate geometrical meaning but in order to understand them, one has to forget the intentions, remembering, with a Chinese leader, that it is not the colour of the cat that matters, but the fact it catches mice.

II.1. The maintenance of space in sequent calculus

When we write a sequent in classical logic, in intuitionistic or minimal logic, the only difference is the *maintenance of space*: in classical logic we have $n+m$ rooms separated by $\vdash$; in minimal logic $m=1$, while in intuitionistic logic $m=0$ or 1. Into the three cases we build particular connectives which belong to such and such tradition, but beyond any tradition we are extremely free, provided we respect an implicit symmetry that is essential for cut-elimination.

Now, what is the meaning of the separation $\vdash$? The classical answer is "to separate positive and negative occurrences". This is factually true but shallow; we shall get a better answer by asking a better question: what in the essence of $\vdash$ makes the two latter logics more constructive than the classical one? For this the answer is simple: take a proof of the existence or the disjunction property; we use the fact that the last rule used is an introduction, which we cannot do classically because of a possible contraction. Therefore, in the minimal and intuitionistic cases, $\vdash$ serves to mark a place where contraction (and maybe weakening too) is forbidden; classically speaking, the $\vdash$ does not have such a meaning, and this is why lazy people very often only keep the right-hand side of classical sequents. Once we have recognized that the constructive features of intuitionistic logic come from the dumping of structural rules on a specific place in the sequents, we are ready to face the consequences of this remark: the limitation should be generalized to the other rooms, i.e., weakening and contraction disappear. As soon as weakening and contraction have been forbidden, we are in linear logic.

II.2. Linear logic as a sequent calculus

Linear logic ignores the left/right-asymmetry of intuitionism; in particular, one can directly deal with right-handed sequents. The first thing to do is to try to rewrite familiar connectives in this framework where structural rules have been limited to exchange; in particular, the two possible traditions for writing the right ∧-rule:

$$\frac{\vdash A, C \quad \vdash B, D}{\vdash A \wedge B, C, D}\,M, \qquad \frac{\vdash A, C \quad \vdash B, C}{\vdash A \wedge B, C}\,A,$$

which are equivalent in classical sequent calculus modulo easy structural manipulations, now become two radically different conjunctions:

- rule (M) treats the contexts by juxtaposition and yields the *multiplicative* conjunction $\otimes$ (*times*);
- rule (A) treats the contexts by identification, and yields the *additive* conjunction & (*with*).

The vaultkey of the system is surely *linear negation* $(.)^{\perp}$. This negation, although constructive, is involutive! All the desirable De Morgan formulas can be written with $(.)^{\perp}$, without losing the usual constructive features. Through the De Morgan translations, the other elementary connectives are easily accessible: the multiplicative disjunction ⅋ (*par*) and the additive one $\oplus$ (*plus*) and linear implication $\multimap$ (*entails*). The last connective was the first to find its way into official life and it has given its name to the full enterprise.

There is a philosophical tradition of 'strict implication' amounting to Lewis. In some sense, linear implication agrees with this tradition: in a linear implication, the premise is used 'once', in the sense that weakening and contraction are forbidden on the premise (in fact 'once' means only something in multiplicative terms, i.e., w.r.t. juxtaposition: additively speaking, the premise can be used several times. $A \multimap A \otimes A$ is not derivable, but $A \multimap A \& A$ is). Even if this philosophical tradition has not been very successful, the existence of linear logic gives a retrospective justification to these attempts.

The most hidden of all linear connectives is *par*, which came to light purely formally as the De Morgan dual of $\otimes$ and which can be seen as the effective part of a classical disjunction. Typically, $A \multimap A$, which everybody understands, is literally the same as $A^{\perp}$ ⅋ A.

II.3. Strength of linear logic

Is linear logic strong enough? In other terms, is it possible to translate usual logic (especially intuitionistic logic) into linear logic? If we look at usual connectives, we discover that some laws belong to the multiplicative universe (e.g., $A \multimap A$), and others to the additive realm (e.g., the equivalence between A and $A \& A$). In fact, no translation works, for a very simple reason: since there are no structural rules, the proofs diminish in size during normalization! This feature, one of the most outstanding qualities of linear logic, definitely forbids any decent translation on the basis of the connectives so far written. The only solution is to allow weakening and contraction to some extent; in order to do so, the *exponentials* ! and ? are introduced; these modalities indicate the possibility of structural rules on the formula beginning with them. For linear proofs involving these modalities, a loss of control over the normalization times occurs, as expected. Roughly speaking, one can say that usual logic (which can be viewed as the adjunction of the modalities) appears as a passage to the limit in linear logic since, for $?A$ to be equivalent with $?A \parr ?A$, we have to do something like an infinite "par". The translation of intuitionistic logic into linear logic is defined by

$$A^0 = A \quad \text{for } A \text{ atomic},$$

$$(A \wedge B)^0 = A^0 \mathbin{\&} B^0, \qquad (A \vee B)^0 = !A^0 \oplus !B^0, \qquad (A \Rightarrow B)^0 = !A^0 \multimap B^0,$$

$$(\neg A)^0 = !A^0 \multimap \mathbf{0} \quad (\mathbf{0} \text{ is one of the constants of the system}),$$

$$(\forall x.A)^0 = \bigwedge x.A^0, \qquad (\exists x.A)^0 = \bigvee x.!A^0$$

($\bigwedge$ and $\bigvee$ are the linear quantifiers, about which there is little to say except that $\bigvee$ is effective, and the De Morgan dual of $\bigwedge!$).

II.4. Subtlety of linear logic

Let us give an example in order to show the kind of unexpected distinctions that linear logic may express; for this, we shall assume that the formula A is quantifier-free, and is represented in linear logic in such a way that "or" between instances of A becomes "$\parr$". We claim that linear logic can write a formula with this meaning: $\exists xAxy$ and this with a Herbrand expansion of length 2. Intuitionistic logic can handle the Herbrand expansion of length 1 and classical logic can handle the general case with no bound on the length, but the much more interesting case of a given length bound was not expressible in usual logics. People with some proof-theoretic background know that a Herbrand expansion of length 2 can be rewritten as a sequent $\vdash Atx, Aux'$, with x, x' not free in t, x' not free in u (midsequent theorem). Now, the formula

$$(\bigvee x \bigwedge yAxy) \parr (\bigvee x \bigwedge yAxy)$$

of linear logic will be cut-free provable exactly when a sequent of the above kind is provable.

II.5. The semantics of linear logic: phases

The sentence saying that usual logic is obtained from linear logic by a passage to the limit is reminiscent of the relation between classical and quantum mechanics. There is a Tarskian semantics for linear logic with some physical flavour: w.r.t. a certain monoid of *phases* formulas are true in certain situations. The set of all phases for which a formula may be true is called a *fact*, among which the fact of *orthogonal phases* plays a central role to represent the absurdity $\perp$. One easily defines the *orthogonal* $A^\perp$ of a fact and by restricting to facts which are themselves orthogonals, we get an involutive operation representing linear negation. There usual connectives are easily defined: & is the intersection, whereas $\multimap$ is the *dephasing*, i.e., $A \multimap B$ is the set of all p such that $p.A \subset B$, where the phase space, although commutative, is being written multiplicatively. A fact is said to hold when it contains the unit phase 1. This semantics is complete and sound w.r.t. linear logic. One of the wild hopes that this suggests is the possibility of a direct connection with quantum mechanics... but let's not dream too much!

III. Linear logic explained to a (theoretical) computer scientist

There are still people saying that, in order to make computer science, one essentially needs a soldering iron; this opinion is shared by logicians who despise computer science and by engineers who despise theoreticians. However, in recent years, the need for a logical study of programmation has become clearer and clearer and the linkage logic-computer-science seems to be irreversible.

(i) For computer science, logic is the only way to rationalize '*bricolage*'; for instance, what makes resolution work is not this or that trick, but the fundamental results on sequent calculus. More: if one wants to improve resolution, one has to respect the hidden symmetries of logic. In similar terms, the use of typed systems (e.g., the system F), is a safeguard against errors of various sorts, e.g., loops or bugs. Moreover, in domains where the methodology is a bit hesitating, as in parallelism, logic is here as the only milestone. In some sense, logic plays the same role as the one played by geometry w.r.t. physics: the geometrical frame imposes certain conservation results, for instance, the Stokes formula. The symmetries of logic presumably express deep conservation of information, in forms which have not yet been rightly conceptualized.

(ii) For logic, computer science is the first real field of application since the applications to general mathematics have been too isolated. The applications have a feedback to the domain of pure logic by stressing neglected points, shedding new light on subjects that one could think of as frozen into desperate staticism, as classical sequent calculus or Heyting's semantics of proofs. Linear logic is an illustration of this point: everything has been available to produce it since a very long time; in particular, retrospectively, the syntactic restriction on structural rules seems so obviously of interest that one can hardly understand the delay of fifty years in its study. Computer science prompted this subject through semantics: The idea behind Heyting's semantics of proofs is that proofs are functions; this idea has been put into a formal correspondence between proofs and functional systems by Curry, Howard, and De Bruijn. The tradition on Heyting's semantics is of little interest because it is full of theological distinctions of the style: "how do we prove that this a proof", etc.; computer scientists have attacked more or less the same question, but with the idea of concentrating on the material part of these proofs, or functions. This has led to Scott semantics, for long the reference concept in the subject; however, Scott's semantics is not free from big defects, specifically, where what is called *finite* in these domains is not finite in any reasonable sense (*noetherian* would have been a happier terminology) and in particular, the fact that arbitrary objects are approximable by finite ones only because we have abused the word 'finite'. In recent work in 1985, the present author has made essential simplifications of Scott's work: by use of a more civilized approach to function spaces one can work with qualitative semantics (rebaptized here to *coherent spaces*) in which finiteness is finiteness. Moreover, all basic definitions involving coherent spaces are very simple and a certain number of features that were hidden behind the heavy apparatus of Scott domains are brought to light, in particular, the discovery that *the arrow is not primitive.*

III.1. The semantics of linear logic: coherent spaces

A *coherent space* is a graph on a set (the *web* of the space). What counts as an object in this space is any subset of the graph made of pairwise compatible (we say: coherent) points. The interpretations made in 1984–85 gave very simple interpretations for the intuitionistic connectives in terms of coherent spaces as schemes for building new coherent spaces. Now, it turns out that there are intermediate constructions; typically, the function space $X \Rightarrow Y$ can be split into

- the formation of the 'repetition space' $!X = X'$;
- the formation of the linear implication $X' \multimap Y$.

Later, $\multimap$ can in turn be split as $X \multimap Y = X^{\perp} \mathbin{⅋} Y$, but this is a bit more dubious since $X \mathbin{⅋} Y = X^{\perp} \multimap Y$, so there is some subjectivity in deciding which among the two multiplicatives is more primitive (not to speak of $\otimes$!).

In fact, after some times, one arrives at the following stock of operations: $(.)^{\perp}$, ⅋, $\multimap$, $\otimes$, &, $\oplus$, !, ?; usual functional types admit decompositions according to the formulas in Section II.3. One can imagine other linear connectives (e.g., a primitive equivalence) which, although very natural, do not lead to any logical system. What is outstanding here is that the decomposition can also be carried out *at the logical level*; i.e., that these new connectives have all qualities of a logic. This logic has already been disclosed before, but we have to look again at its proof-system, as will be done in the following section.

III.2. Proof-nets: a classical natural deduction

In the beginning (say: up to the end of '85) things remained simple because it was not clear that the intuitionistic framework was an artificial limitation; at that moment, $(.)^{\perp}$, ⅋, and ? were still in statu nascendi. It was therefore possible to keep a functional notation for the proofs of this intuitionistic linear logic: for instance, the formation of the linear application tu was subject to the restriction $\mathrm{FV}(t) \cap \mathrm{FV}(u) = \emptyset$; for the additive pair $\&tu$, the restriction was $\mathrm{FV}(t) = \mathrm{FV}(u)$, for linear lambda abstraction $Lx.t$, the restriction was $x \in \mathrm{FV}(t)$, etc. This gave a reasonable system, still of interest, however with the defect of too many connectives involving commutative conversions ($\otimes$, $\oplus$, !). The conviction that the system should be relevant to parallelism and the rough analogy 'sequential = intuitionistic, parallel = classical' produced a shift to the 'classical' framework, where sequents are left/right-symmetric with bad consequences for normalization since it is well-known that no decent natural deduction exists in classical frameworks. However, linear logic is not so bad since it has been possible to define a notation of '*proof-net*' which is a proof with several conclusions. (It seems that the problem is hopeless in usual classical logic and the accumulation of several inconclusive attempts is here to back up this impression. In linear logic, we can distinguish between, for instance, the multiplicative features of disjunction, which deserve a certain treatment, and the additive ones, which deserve another treatment; trying to unify both treatments would lead to a mess.)

In fact, the core of proof-nets consists in studying the multiplicative fragment, which is handled in a very satisfactory way: starting with axioms $\overline{A \quad A^{\perp}}$ one develops proof structures by combination of two binary rules, or *links*,

$$\frac{A \quad B}{A \otimes B} \quad \text{and} \quad \frac{A \quad B}{A \parr B}.$$

Most of these structures are incorrect because the left-hand side may work in a way contradicting the right-hand side. Now the soundness criterion is as follows: to each multiplicative link is associated a *switch* with two positions "L" and "R"; an ideal particle (representing the flow of input/output inside the proof) tries to travel starting from a given formula, and in a given direction (upwards: question, or downwards: answer); at each moment, the particle knows (the switches are here to help) where to go, so it eventually makes a cyclic *trip*.

The soundness requirement is simple: for any positioning of the n switches, the trip is long, i.e., it goes twice through all formulas, once up, once down; in other terms, the whole structure has been visited, and every question has been answered, every answer has been questioned. Clearly, the most difficult result of the paper is to prove that this notion of proof-net is equivalent to a sequent calculus approach, i.e., can be sequentialized. The full calculus does not have such a system; it is handled through the concept of *proof-box*, which is a lazy device: moments of sequentialization are put into boxes and the boxes are interconnected in the perfect multiplicative way already explained.

The soundness conditions for proof-nets (absence of shorttrips) should not be misunderstood: it is not a condition that should be practically verified because, with n switches, we essentially have 2^n verifications to make. It must be seen as an *abstract* property of these parallel proofs that we are not supposed to check by means of a concrete algorithm. (In the same way, the fact that an object of type int$\Rightarrow$int of F when applied to an integer $\bar{n}$ yields, after normalization, an integer $\bar{m}$ need not be checked; this, however, is an important abstract property of F). However, we want to work with proof-nets as programs, and when we have one, we want to be sure by some means that it is actually a proof-net; this is why methods for generating proof-nets are very important, and we know the following ones:

(i) *Desequentialization* of a proof in linear sequent calculus: we can imagine this as the most standard way to obtain a proof-net. The user would produce a source program (in linear sequent calculus), and this program would be compiled as a proof-net, i.e., as a graph between occurrences of formulas.

(ii) *Logical operations* between proof-nets: for instance, a proof-net β with two conclusions A and B can be connected to a proof-net β' with conclusion $A^{\perp}$, by means of the cut-link in Fig. 1(a). Here, β has been seen as a (linear) function taking as arguments objects of type $A^{\perp}$ (proof-nets admitting $A^{\perp}$ as conclusion) and yielding an answer of type B. The function applies to the argument by means of CUT.

Fig. 1.

The picture is however wholly symmetrical; one can also apply β to an argument β'' of type $B^{\perp}$, and this yields a result of type A as shown in Fig. 1(b). One can also combine both ideas and apply β to both β' and β'' etc.; when we say that we 'apply' β to β', this is really to explain the things very elementarily because one could have said that β' is applied to β. There is a perfect function/argument symmetry in linear logic.

(iii) *Normalization*: There is a normalization procedure which eliminates the use of the cut-link. When we normalize a proof-net, then the result is still a proof-net, as we can easily *prove*. In the study of normalization, the structure

$$\frac{\overline{A \quad A^{\perp}}}{\text{CUT}}$$

plays an important role: first, this is not a proof-net, because there is the shorttrip $A^{\wedge}, A^{\perp}_{\vee}, A^{\wedge}$ etc.; second, it would be a 'black hole' in the normalization process, i.e., a structure hopelessly normalizing into itself. If we try to imagine what could be the meaning of this pattern, think of a variable of type A (denoting the address of a memory case) calling itself. To some extent, the conditions on trips are here just to forbid this 'black hole' and to forbid by the way all situations that would lead to such a pattern by normalization, etc.

Although we can deal with concrete proof-nets without a *feasible* characterization, alternative definitions of soundness are welcome. One can expect from them two kinds of improvements:

(i) feasibility, i.e., the possibility to use proof-nets directly that have been written on a screen for instance;

(ii) the extension of the advantages of the parallel syntax outside the multiplicative fragment; for instance, the weakening rules for ($\perp$) and (?) are handled in terms of boxes because our definition of rip excludes disconnected nets. Surely, some disconnected nets are however acceptable, but we have not found any simple soundness characterization for them.

III.3. Normalization for proof-nets

There is a radical and quick procedure for normalizing the multiplicative fragment of proof-nets. The full language is more delicate to handle because the cut-rule sometimes behaves badly w.r.t. boxes; in particular, it is difficult to define a terminating normalization process for the full calculus with a Church–Rosser property. Here we have to be a little more careful:

(i) Although the Church–Rosser property is lost, there is a semantic invariance of the normalization process which ensures that no serious divergence can be obtained.

(ii) One can even represent a proof-net by a family of *slices*, and this representation is Church–Rosser. Unfortunately, the status of slices is far from being well-understood and secured.

(iii) One can expect to be able to improve the syntax by reducing the number of boxes. The box of the rule (!) is an absolute one and one cannot imagine to remove it. But the boxes for $\perp$ and (W?), for (&) and for ($\bigwedge$) can perhaps be removed with various difficulties. The main problem is of course to find each time the ad hoc modification of the concept of trip and also to remain with a manageable condition.

Anyway, there is a strong normalization theorem for PN2 which generalizes the familiar result of [1] for *F*. The methodology combines the old idea of *candidat de réductibilité* (CR) with phase-like duality conditions; it is very instructive to remark that a CR defined by duality immediately has a lot of properties which, in the old works (e.g., [2]), we were forced to require explicitly!

Cut-elimination plays the role of the execution of a program; here we must recall a point that is very important: proof-nets generalize natural deduction, which in turn is isomorphic to lambda-calculi with types. Hence, proof-nets are a direct generalization of typed lambda-calculus and a proof-net is as much expressive as a functional notation. It is sometimes difficult for computer scientists to go through this point: they are shocked by the fact that PN2 does not look like a functional calculus. In fact, the earlier versions were functional, using typed linear abstraction, etc.; but the functional notation is unable to make us understand this basic fact, namely, that a (linear) term $t[x^A]$ (of type B) should be simultaneously seen as a term $u[y^{B^\perp}]$ (of type $A^\perp$). The functional notation is surely too expressive for our imagination to ever abolish its use; but at a very high degree of abstraction, it is clearly misleading and unadapted. But, once more, proof-nets are lambda expressions, simply written in the way best adapted to their structure. The fact that they are quite remote from our usual habits or prejudices should not be seen as an argument against their use in computer science: if there is ever any computer working with linear logic, it is expected that the proof-nets will mostly be seen by the compiler, the user however will use more traditional devices!

III.4. Relevance for computer science

Roughly speaking, proof-nets represent programs whose termination is guaranteed by the normalization theorem. The kind of programs obtained is quite different from those coming from intuitionistic logic; let us examine this in detail in the following sections.

III.4.1. Questions and answers

In typed lambda-calculus, a term $t[x_1, \ldots, x_n]$ of type $[\sigma_1, \ldots, \sigma_n]\tau$ represents a *communication* between *inputs* of types $\sigma_1, \ldots, \sigma_n$ and an *output* of type τ. The communication works as follows: outputs $u_1, \ldots, u_n$ of types $\sigma_1, \ldots, \sigma_n$ can be substituted for $x_1, \ldots, x_n$ yielding $t[u_1, \ldots, u_n]$; recall that substitution is the functional analogue of the cut-rule. Now, the terminology 'input/output', although correct, is a bit misleading because we have a tendency to say that t waits for the inputs $u_1, \ldots, u_n$, whereas it is preferable to realize that the inputs are in fact the abstract symbols for inputs, i.e., the variables $x_1, \ldots, x_n$. For that reason, the terminology *question* (for $x_1, \ldots, x_n$) and *answer* (for t) is more suited. In other terms, x_i is a question answered by u_i when doing the substitution. The functional approach to programmation is essentially asymmetric as the communication is always between several questions and one answer: this unique answer depends on the answers to the previous questions. For instance, the term x of type σ represents a very simple type of communication: the question is to find a term u of type σ and, from an answer to this question, we get the answer corresponding to the term "x", namely "u". The communication is therefore just recopying.

III.4.2. Towards parallelism

Parallelism will occur as soon as we are able to break the Q/A-asymmetry between questions and answers. The most radical solution is the ability to exchange roles; i.e., a question could be seen as an answer and conversely. Lineaments of this situation appear in usual typed lambda-calculus: the term $t[x]$, which is a communication between a question of type σ and an answer of type τ, yields a term $t'[y]$ which is a communication between a question of type $\tau \Rightarrow \rho$ and an answer of type $\sigma \Rightarrow \rho$: $t'[y] = \lambda x.y(t[x])$. The roles Q/A have been exchanged during this process; however, the transformation made is not reversible: the exchange of roles is hopelessly mixed with other things and we cannot proceed any longer in this framework.

In linear logic, this limitation is overcome: questions and answers play absolutely symmetric roles, which are interchanged by *linear negation.* A proof-net with conclusions $A_1, \ldots, A_n$ can be seen as a communication net between answers of respective types $A_1, \ldots, A_n$. But since we have

$$\textit{answer of type } A = \textit{question of type } A^{\perp},$$

we can put it as, say, a communication between questions of types $A_1^{\perp}, \ldots, A_{n-1}^{\perp}$ and an answer of type A_n, and this immediately sounds more familiar!

Typically, the axiom link $\overline{A \quad A^{\perp}}$, which is a communcation between two answers of types A and $A^{\perp}$, can be seen as the trivial communication between either a question of type $A^{\perp}$ and an answer of the same type, or between a question of type $A^{\perp\perp} = A$ and an answer of the type A. The cut-rule $\frac{A \quad A^{\perp}}{\text{CUT}}$, which puts together two questions of orthogonal types (or a question and an answer of the same type), is the basic way to make the communication effective.

III.4.3. Communication and trips

In spite of the equality between an answer and a question of the orthogonal type, the two things should be distinguished a little; for instance, in the basic sequential case (Section III.4.1) the communication is *from* the question (variables) *to* the answer (the term). This distinction is essentially the one used for *trips*: $A^{\wedge}$ stands for a question of type $A^{\perp}$, where $A_{\vee}$ stands for an answer of type A. In other terms, when we move upwards, we question; when we move downwards, we answer!

The way communication works is well explained by the trips:

(i) In a CUT-link $\frac{A \quad A^{\perp}}{\text{CUT}}$ the trip algorithm: 'from $A_{\vee}$ go to $A^{\perp\wedge}$ and from $A_{\vee}^{\perp}$ go to $A^{\wedge}$', is easily explained as: an answer of type A is replaced by a question of type A, and an answer of type $A^{\perp}$ is replaced by a question of type $A^{\perp}$: the questions met their answers.

(ii) In the axiom link $\overline{A \quad A^{\perp}}$, the interpretation is basically the same, the only difference being that, in the case of the cut, an *answer is changed into a question*, whereas in the case of the axiom, a *question becomes an answer.*

(iii) The multiplicatives $\otimes$ and $\parr$ are the two basic ways of living *concurrence*: in both cases, a question of type $(A \; m \; B)^{\perp}$ has to be seen as two separate questions of type $A^{\perp}$ and $B^{\perp}$; an answer of type $A \; m \; B$ has to be seen as two separate answers of types A and B. However,

- in the case of $\otimes$, there is no cooperation: if we start with $A^{\wedge}$, then we come back through $A_{\vee}$ before entering $B^{\wedge}$ after which we come back through $B_{\vee}$;
- in the case of $\parr$, there is cooperation: if we start again with $A^{\wedge}$, then we are expected through $B_{\vee}$, from which we go to $B^{\wedge}$ and eventually come back through $A_{\vee}$.

(iv) The additives & and $\oplus$ correspond to superposition, whose typical example is the instruction IF-THEN-ELSE:

- in terms of answers, the type $A \,\&\, B$ means one answer of each type;
- in terms of questions, the type $A \,\&\, B$ means either a question of type A or a question of type B. (The meaning of $\oplus$ is explained by duality.)

In particular, only one of the answers occurring in an answer of type $A \,\&\, B$ is of interest, but we do not know which until the corresponding question is clearly asked. This is why the rule (&) works with boxes in which we put the two alternative answers so that this superposition is isolated from the rest. This is a typical lazy feature.

(v) The exponentials ! and ? correspond to the idea of *storage*:
- an answer of type $!A$ corresponds to the idea of *writing* an answer of type A in some stable register so that it can be used *ad nauseam.*
- a question of type $!A$ corresponds to several questions of type A, 'several' being extremely unspecified (0 times: rule $(W?)$, 1 time: rule $(D?)$, etc.), i.e., is an iterated *reading.*

The fact that the rule (!) works with boxes expresses the idea of storage.

III.4.4. Work in progress

In the previous section we explained the general pattern of communication in linear logic; compared to extant work on parallelism, especially Milner's approach, the main novelty lies in the *logical approach*, i.e., parallelism should work for *internal*[1] logical reasons and not by chance. Systems like F, which are of extremely general interest for sequential computation, work because they are logically founded. (Remember that F is the system of proofs of second-order intuitionistic propositional calculus.) For similar reasons, the programs coming from linear logic are proofs within a very regular logic and this gives the good functioning of such programs (in terms of communication) at a theoretical level. At a more practical level, all these general ideas have to be put into more concrete proposals and this will be done in separate publications among which the following are expected quite soon:
- A paper with Yves Lafont, concentrating on the nonparallel aspects, in particular the new treatment of data suggested by linear logic; the formalism used is intuitionistic linear logic.
- A paper with Gianfranco Mascari, concentrating on the parallel aspects of linear logic; in particular, it will give some more details for a concrete implementation.

IV. Pons asinorum: from usual implication to linear implication[2]

What is the use of semantics of programming languages?

(1) Some people think that the purpose of a semantics is to interpret sets of equations in a complete way; in some sense, the semantics comes like the *official blessing* on our language, rules, etc.: they are perfect and cannot be improved. The success of this viewpoint lies in Gödel's completeness theorem which guarantees, for any consistent system of axioms, a complete semantics.

(2) A more controversial viewpoint consists in looking for *disturbing* semantics, which shed unexpected lights on the systems we know. Of course, this viewpoint is highly criticizable since there is no absolute certainty that anything will result from

[1] In this, our approach is radically different from the idea of an *external* logical comment to parallelism by means of modal, temporal, etc. logics.

[2] *Remark on notations*: the symbol ⅋ is not accessible on a word processor; I therefore propose to replace it by $\sqcup$ and simultaneously to replace & by the symbol $\sqcap$; it is important that the two symbols remind one of another.

such attempts. But the stakes are clearly higher than in the hagiographic viewpoint (1) and a change of syntax (as the one coming with linear logic) may occur from disturbing semantics.

Let us explain what is the starting disturbance that led to linear logic: the semantics of *coherent spaces* (qualitative domains in [4]), developed in Section 3 below, works as follows in the case of an implication.

IV.1. Definition. If X and Y are two coherent spaces, then a *stable function* from X to Y is a function F satisfying:

(S1) $a \subset b \to F(a) \subset F(b)$;

(S2) $a \cup b \in X \to F(a \cap b) = F(a) \cap F(b)$;

(S3) F commutes with directed unions.

IV.2. Theorem (see [4, 5]). *With any stable function F from X to Y, associate the set* $\mathrm{Tr}(F)$ *defined by*

$$\mathrm{Tr}(F) = \{(a, z)\,; a \textit{ finite object of } X,\ z \in F(a) \textit{ such that } z \in F(b) \textit{ and } b \subset a \to b = a\}.$$

Then, for any $c \in X$, we have

$$F(c) = \{z\,; \exists a \subset c \textit{ such that } (a, z) \in \mathrm{Tr}(F)\}. \tag{1}$$

Moreover, the set of all sets $\mathrm{Tr}(F)$, *when F varies through all stable functions from X to Y, is a coherent space $X \Rightarrow Y$ which can be independently defined by*

$$|X \Rightarrow Y| = X_{\mathrm{fin}} \times |Y|$$

$$(a, z) \bigcirc (b, t)[\mathrm{mod}\ X \Rightarrow Y] \textit{ iff } (1)\ a \cup b \in X \to z \bigcirc t[\mathrm{mod}\ Y],$$

$$(2)\ a \cup b \in X \textit{ and } a \neq b \to z \neq t.$$

IV.1. Interpretation of functional languages

Theorem IV.2 is enough to interpret λ-abstraction and application in typed languages; the formation of the trace, encoding a function by a coherent set, interprets λ-abstraction whereas application is defined by means of formula (1) above. Beta- and eta-conversion are immediately seen to be sound w.r.t. this semantics.

IV.2. The disturbance

If one forgets the fact that coherent spaces are extremely simpler than Scott domains, nothing unexpected has occurred: the semantics follows the same lines as Scott-style interpretation. The novelty is that, for the first time, we have a *readable* definition of function space. The disturbance immediately comes from a close inspection of this definition: why not consider independently:

(1) a space of repetitions $!X$,

(2) a more restricted form of implication; the *linear* one.

IV.3. Definition. Let X and Y be coherent spaces; a stable function F from X to Y is said to be *linear* when it enjoys the two conditions:

(L1) $a \cup b \in X \to F(a \cup b) = F(a) \cup F(b)$,

(L2) $F(\emptyset) = \emptyset$.

Equivalently, (L1), (L2), and (S3) can be put together in one condition (L):

(L) Assume that $A \subset X$ is a set of pairwise coherent subsets of X; then $F(\bigcup A) = \bigcup\{F(a); a \in A\}$.

IV.4. Theorem. *A stable function F is linear iff its trace is formed of pairs (a, z), where the components a are singletons.*

Proof. $(a, z) \in \mathrm{Tr}(F)$ when $z \in F(a)$: this forbids $a = \emptyset$ by (L2) and when a is minimal w.r.t. inclusion among such pairs (b, z). Write $a = a' \cup a''$; then, from $(a, z) \in \mathrm{Tr}(F)$, we get $z \in F(a) = F(a') \cup F(a'')$, so $z \in F(a')$ or $z \in F(a'')$. Now, the minimality of a implies that a is a singleton. □

IV.5. Definition. For linear maps, it is more suited to forget the singleton symbols; so we get the following characterization of the space $X \multimap Y$ formed of traces of linear maps from X to Y:

$$|X \multimap Y| = |X| \times |Y|;$$

$$(x, y) \frown\!\!\!\smile (x', y')\ [\mathrm{mod}\ X \multimap Y] \text{ iff } \begin{array}{l}(1)\ x \frown\!\!\!\smile x'\ [\mathrm{mod}\ X] \to y \frown\!\!\!\smile y'\ [\mathrm{mod}\ Y],\\ (2)\ x \frown x'\ [\mathrm{mod}\ X] \to y \frown y'\ [\mathrm{mod}\ Y].\end{array}$$

Formula (1) is now replaced by the formula of linear application:

$$F(b) = \{y; \exists x \in b \text{ such that } (x, y) \in \mathrm{Tr}(F)\}. \tag{2}$$

IV.6. Definition. If X is a coherent space, then $!X$ is defined as follows:

$$|!X| = X_{\mathrm{fin}},$$

$$a \frown\!\!\!\smile b \bmod !X \text{ iff } a \cup b \in X.$$

IV.3. The decomposition

We clearly have: $X \Rightarrow Y = !X \multimap Y$; in order to make this look like a decomposition, we have to decompose λ-abstraction and application; for linear functions, the trace and formula (2) will play the role of linear λ-abstraction and linear application. We need two other operations connected with !:

(i) when $x \in X$, the formation of $!x \in !X$:

$$!x = \{a; a \in x \text{ and } a \text{ finite}\};$$

(ii) the linearization of a function: if F is a stable map from X to Y, one can construct a linear stable map $\mathrm{LIN}(F)$ from $!X$ to Y simply by $\mathrm{Tr}(\mathrm{LIN}(F)) = \mathrm{Tr}(F)$.

Now, λ-abstraction can be viewed as two operations: first linearization, then linear λ-abstraction.

Application can be viewed as two operations too: applying a to b is the linear application of a to $!b$.

IV.4. Further questions

This already suggests to develop linear implication and ! for themselves. However, there are a lot of other disturbing facts; for instance, nothing prevents us from considering the result of replacing a coherent space X by its dual $X^\perp$ where the roles of coherence and incoherence have been exchanged. But then $X \multimap Y$ is the same as $Y^\perp \multimap X^\perp$, i.e., linear functions are reversible in some sense. This immediately suggests a classical framework (symmetry between inputs/outputs, or variables/results, or questions/answers). Deep syntactic modifications come from simple semantic identifications.

1. The phase semantics

Usual finitary logics (we mean classical and intuitionistic logics) involve a certain conception of semantics which may be described as follows: truth is considered as meaningful independently of the process of verification. In particular, this imposes strong limits on the propositional parts of such logics and, for instance, starting with the general idea of *conjunction* we are hopelessly led to 'A & B true iff A true and B true'. By admitting several possible worlds, the intuitionistic semantics (Kripke models) allows more subtle distinctions, promoting drastically new connectives (the intuitionistic disjunction) while being extremely conservative on others (conjunction).

The change of viewpoint occurring in linear logic is simple and radical: we have to introduce an *observer*; in order to verify a *fact* A, the observer has to do something, i.e., there are tasks $p, q, \ldots$ which verify A (notation: $p \models A$, $q \models A, \ldots$). These tasks can be seen as *phases* between a fact and its verification; phases form a monoid P and we shall consider that a fact is verified by the observer when there is no phase between him and the fact, i.e., when $1 \models A$. The idea of introducing phases makes a radical change on our possible connectives, which we shall explain by starting with the idea of a conjunction:

(i) "&": say that $p \models A \,\&\, B$ when $p \models A$ and $p \models B$;

(ii) "$\otimes$": say that $pq \models A \otimes B$ when $p \models A$ and $q \models B$.[3]

In the case of "&" the task p shares two verifications, while in the case of "$\otimes$", the verification is done by dispatching the total task pq between A and B: p verifies A and q verifies B. These two connectives have very different behaviours, as expected from their distinct semantics.

[3] The definition given here is slightly incorrect and has been simplified for pedagogic purposes.

By the way, observe that "$\otimes$" appears as a noncommutative operation. In order to keep the simplicity of the theory at this early stage of development, we shall make the hypothesis of the *commutativity* of P, i.e., we shall develop *commutative* linear logic. The noncommutative case is obviously more delicate and we have not accumulated enough materials in the commutative case to be able to foresee the general directions for a noncommutative linear logic; also we should have some real reason to shift to noncommutativity!

Semantically speaking, let us identify a fact A with the set $\{p; p \models A\}$ so that a fact appears as a set of phases. Among all facts, we shall distinguish the *absurd* fact $\perp$, which is an arbitrary fixed subset of P. Given any subset G of P, we define (w.r.t. $\perp$) its dual $G^\perp$; facts will be exactly those subsets G of P such that $G^{\perp\perp} = G$. The basic linear connectives are then easily developed as those operations leading from facts to facts.

1.1. Definition. A *phase space* P consists in the following data:

(i) a monoid, still denoted P, and whose elements are called *phases*; the monoid is supposed to be commutative, i.e.,

- neutrality: $1p = p1 = p$ for all $p \in P$,
- commutativity: $pq = qp$ for all $p, q \in P$,
- associativity: $(pq)r = p(qr)$ for all $p, q, r \in P$;

(ii) a set $\perp_P$ (often denoted $\perp$ for short), the set of *antiphases* of P. (One can also say "orthogonal phases".)

1.2. Definition. Assume that G is a subset of P; then we define its *dual* $G^\perp$ by

$$G^\perp = \{p \in P; \forall q(q \in G \rightarrow pq \in \perp)\}.$$

1.3. Definition. A *fact* is a subset G of P such that $G^{\perp\perp} = G$; the elements of G are called the *phases* of G; G is *valid* when $1 \in G$.

1.4. Immediate properties

(i) *For any* $G \subset P$, $G \subset G^{\perp\perp}$;

(ii) *For any* $G, H \subset P$, $G \subset H \rightarrow H^\perp \subset G^\perp$;

(iii) G *is a fact iff* G *is of the form* $H^\perp$ *for some subset* H *of* P.

1.5. Examples. (i) $\perp$ is a fact because $\perp = \{1\}^\perp$.

(ii) Define a fact $\mathbf{1}$ by $\mathbf{1} = \perp^\perp$; then, observe that $1 \in \mathbf{1}$; moreover, if $p, q \in \mathbf{1}$, then $pq \in \mathbf{1}$ (more generally, if $p \in \mathbf{1}$, and $q \in G$, then $pq \in G^{\perp\perp}$). Hence, $\mathbf{1}$ is a submonoid of P.

(iii) Define a fact $\top$ by $\top = \emptyset^\perp$; clearly, $\top = P$.

(iv) Define a fact $\mathbf{0}$ by $\mathbf{0} = \top^\perp$; then $\mathbf{0}$ is the smallest fact (w.r.t. inclusion). Most of the time, $\mathbf{0}$ will be void.

1.6. Proposition. *Facts are closed under arbitrary intersections.*

Proof. If $(G_i)_{i\in I}$ is a family of facts, then $\bigcap G_i = (\bigcup G_i^{\perp})^{\perp}$ and $\bigcap G_i$ are therefore facts. □

1.7. Definition. The connective *nil* (*linear negation*) is defined by duality: if G is a fact, then its linear negation is $G^{\perp}$. Things have been made so that nil is an involution, i.e., $G^{\perp\perp} = G$.

1.8. Definition. The multiplicative connectives (or *multiplicatives*) are the connectives *times*, *par* and *entails*; all three are definable from the operation of product of subsets of P: if $G, H \subset P$, then $GH = \{pq\,; p \in G \text{ and } q \in H\}$.

(i) If G and H are facts, their *parallelization* (connective *par*) $G ⅋ H$ is defined by $G ⅋ H = (G^{\perp}.H^{\perp})^{\perp}$.

(ii) If G and H are facts, their *tensorization* (connective *times*) $G\otimes H$ is defined by $G\otimes H = (G.H)^{\perp\perp}$.

(iii) If G and H are facts, their *linear implication* (connective *entails*) is defined by $G \multimap H = (G.H^{\perp})^{\perp}$.

1.9. Basic properties of multiplicatives. (i) *Any multiplicative can be defined from any other and* nil:

$$G\otimes H = (G^{\perp} ⅋ H^{\perp})^{\perp}, \qquad G \multimap H = G^{\perp} ⅋ H,$$

$$G ⅋ H = (G^{\perp}\otimes H^{\perp})^{\perp}, \qquad G \multimap H = (G\otimes H^{\perp})^{\perp},$$

$$G ⅋ H = G^{\perp} \multimap H, \qquad G\otimes H = (G \multimap H^{\perp})^{\perp}.$$

(ii) par *and* times *are commutative and associative; they admit neutral elements which are* $\perp$ *and* $\mathbf{1}$ *respectively*:

$$\perp ⅋ G = G, \qquad \mathbf{1}\otimes G = G,$$

$$(G ⅋ H) ⅋ K = G ⅋ (H ⅋ K), \qquad (G\otimes H)\otimes K = G\otimes(H\otimes K),$$

$$G ⅋ H = H ⅋ G, \qquad G\otimes H = H\otimes G.$$

(iii) entails *satisfies the following properties*:

$$\mathbf{1} \multimap G = G, \qquad G \multimap \perp = G^{\perp},$$

$$(G\otimes H) \multimap K = G \multimap (H \multimap K), \qquad G \multimap (H ⅋ K) = (G \multimap H) ⅋ K,$$

$$G \multimap H = H^{\perp} \multimap G^{\perp}.$$

Proof. (i) is immediate from the definitions; once (i) has been established, (ii) and (iii) can be reduced to the particular case of $\otimes$, i.e., associativity, commutativity and neutrality of $\mathbf{1}$ for this connective; among these properties only associativity deserves some attention: we shall prove that $(G\otimes H)\otimes K = (G.H.K)^{\perp\perp}$. This property is easily reduced to the following lemma.

1.9.1. Lemma. *If F, G are two subsets of P, then $(F.G)^{\perp\perp} \supset F^{\perp\perp}.G^{\perp\perp}$.*

Proof. Assume that $p \in F^{\perp\perp}$, $q \in G^{\perp\perp}$; we have to show that, given any $v \in (F.G)^{\perp}$, $pqv \in \perp$; the hypothesis $v \in (F.G)^{\perp}$, however, means that, for all $f \in F$ and $g \in G$, $vfg \in \perp$. In particular, if we fix $f \in F$, it follows that $vf \in G^{\perp} = G^{\perp\perp\perp}$, hence, $vfq \in \perp$. For similar reasons, $vq \in F^{\perp} = F^{\perp\perp\perp}$, so $vpq \in \perp$. □

1.10. Remark. There is the following illuminating alternative definition of $G \multimap H$: $p \in G \multimap H$ iff, for all $q \in G$, $pq \in H$.

1.11. Definition. The additive connectives (or *additives*) are the connectives *with* and *plus* which are obtained from the boolean operations:

(i) If G and H are facts, then their *direct product* (connective *with*) $G \& H$ is defined by $G \& H = G \cap H$.

(ii) If G and H are facts, then their *direct sum* (connective *plus*) $G \oplus H$ is defined by $G \oplus H = (G \cup H)^{\perp\perp}$.

1.12. Basic properties of additives. (i) *The additives are interrelated by a De Morgan principle*

$$G \& H = (G^{\perp} \oplus H^{\perp})^{\perp}, \qquad G \oplus H = (G^{\perp} \& H^{\perp})^{\perp}.$$

(ii) with *and* plus *are commutative and associative*; *they admit neutral elements which are* $\top$ *and* $\mathbf{0}$ *respectively*:

$$\top \& G = G, \qquad \mathbf{0} \oplus G = G,$$

$$G \& H = H \& G, \qquad G \oplus H = H \oplus G,$$

$$G \& (H \& K) = (G \& H) \& K, \qquad G \oplus (H \oplus K) = (G \oplus H) \oplus K.$$

1.13. Distributivity properties. (i) $\otimes$ *is distributive w.r.t.* $\oplus$, *and* $⅋$ *is distributive w.r.t.* $\&$.

$$G \otimes (H \oplus K) = (G \otimes H) \oplus (G \otimes K), \qquad G ⅋ (H \& K) = (G ⅋ H) \& (G ⅋ K),$$

$$\mathbf{0} \otimes G = \mathbf{0}, \qquad \top ⅋ G = \top.$$

This yields, for $\multimap$,

$$(G \oplus H) \multimap K = (G \multimap K) \& (H \multimap K), \qquad G \multimap (H \& K) = (G \multimap H) \& (G \multimap K),$$

$$\mathbf{0} \multimap G = \top, \qquad G \multimap \top = \top.$$

(ii) *Between* $\otimes$ *and* $\&$, *between* $⅋$ *and* $\oplus$, *there is only 'half-distributivity'*

$$G \otimes (H \& K) \subset (G \otimes H) \& (G \otimes K),$$

$$(G ⅋ H) \oplus (G ⅋ K) \subset G ⅋ (H \oplus K)$$

which yields, in the case of $\multimap$,

$$(G\multimap K)\oplus(H\multimap K)\subset(G\,\&\,H)\multimap K,$$

$$(G\multimap H)\oplus(G\multimap K)\subset G\multimap(H\oplus K).$$

Proof. Let us concentrate on the first distributivity property; the following lemma is of general interest.

1.13.1. Lemma. *If G is any subset of P, then $G^{\perp\perp}$ is the smallest fact containing G.*

Proof. There is a smallest fact containing G, namely the intersection of all facts containing G; let us call it S; clearly, $S\subset G^{\perp\perp}$, from which we get $G^{\perp\perp}\subset S^{\perp\perp}\subset G^{\perp\perp\perp\perp}$, and so $G^{\perp\perp}=S^{\perp\perp}=S$. □

Proof of Properties 1.13 (*continued*). Now, observe that $G.(H\cup K)^{\perp\perp}\subset(G.(H\cup K))^{\perp}$ (Lemma 1.9.1); but $G.(H\cup K)=G.H\cup G.K$, from which we get

$$G.(H\cup K)^{\perp\perp}\subset(G.H\cup G.K)^{\perp\perp}\subset(G\otimes H)\oplus(G\otimes K).$$

By Lemma 1.13.1, $G\otimes(H\oplus K)\subset(G\otimes H)\oplus(G\otimes K)$.

Conversely, starting with $G.H\subset G\otimes(H\oplus K)$ and $G.K\subset G\otimes(H\oplus K)$, we get, by Lemma 1.13.1, $G\otimes H\cup G\otimes K\subset G\otimes(H\oplus K)$, and, by Lemma 1.13.1 once more,

$$(G\otimes H)\oplus(G\otimes K)\subset G\otimes(H\oplus K). \qquad \square$$

Up to now we have described the propositional framework of linear logic. This part is the most important part of linear logic, the one which is the most radically different from usual logics. For this fragment, we shall now set up a formal system and prove a completeness theorem; afterwards, we shall have a quick overview of the modalities and quantifiers which have been deleted for reasons of clarity. These new operations, although important, will not disturb too much the ordinance of things that is now taking shape. Our first syntactical concern is economy: we essentially need a multiplicative, an additive, and nil. We get a very nice syntax by adopting the following conventions:

(i) The atomic propositions are $\mathbf{1},\perp,\top,\mathbf{0}$ together with propositional letters $a,b,c,\ldots$ and their duals $a^{\perp},b^{\perp},c^{\perp},\ldots$.

(ii) Propositions are constructed from atomic ones by applying the binary connectives $\otimes$, $\parr$, $\oplus$, $\&$.

In particular, linear negation of a proposition is defined as follows:

$$\mathbf{1}^{\perp}=\perp,\qquad \perp^{\perp}=\mathbf{1},\qquad \top^{\perp}=\mathbf{0},\qquad \mathbf{0}^{\perp}=\top,$$

$$a^{\perp\perp}=a\quad(a^{\perp}\text{ is already in the syntax}),$$

$$(A\otimes B)^{\perp}=A^{\perp}\parr B^{\perp},\qquad (A\parr B)^{\perp}=A^{\perp}\otimes B^{\perp},$$

$$(A\oplus B)^{\perp}=A^{\perp}\,\&\,B^{\perp},\qquad (A\,\&\,B)^{\perp}=A^{\perp}\oplus B^{\perp}.$$

This particular form of syntax without negation is not something deep; once more, this is just a way to avoid repetitions.

1.14. Definition. (i) A *phase* structure for the propositional language just described consists in a phase space $(P, \bot_P)$ and, for each propositional letter a, a fact a_S of P.

(ii) With each proposition we associate its interpretation, i.e., a fact of P, in a completely straightforward way: the interpretation of A in the structure S is denoted by A_S or $S(A)$.

(iii) A is valid in S when $1 \in S(A)$.

(iv) A is a *linear tautology* when A is valid in any phase structure S.

1.15. Definition. We develop here a sequent calculus for linear logic; as expected from the peculiarities of our syntax, the sequents will have no left-hand side, i.e., they will be of the form $\vdash A_1, \ldots, A_n$. The real change w.r.t. usual logics consists in *the dumping of structural rules* and, in fact, only the rule of exchange is left. Such a situation is more familiar than it seems at first sight since intuitionistic logic is built on such a restriction: the unique right-hand side formula in a special place in the sequent where contraction is forbidden. In linear logic this interdiction is extended to the whole sequent and weakening is forbidden as well.

Logical axioms:

$$\vdash A, A^\bot.$$

(It is enough to restrict it to the case of a propositional letter a.)

Cut rule:

$$\frac{\vdash A, \boldsymbol{B} \qquad \vdash A^\bot, \boldsymbol{C}}{\vdash \boldsymbol{B}, \boldsymbol{C}}\,\mathrm{CUT}.$$

Exchange rule:

$$\frac{\vdash \boldsymbol{A}}{\vdash \boldsymbol{B}}\,\mathrm{EXCH},$$

where $\boldsymbol{B}$ is obtained by permuting the formulas of $\boldsymbol{A}$.

Additive rules:

$$\vdash \top, \boldsymbol{A} \quad (\text{axiom, } \boldsymbol{A} \text{ arbitrary}) \quad (\text{no rule for } \mathbf{0}),$$

$$\frac{\vdash A, \boldsymbol{C} \quad \vdash B, \boldsymbol{C}}{\vdash A \,\&\, B, \boldsymbol{C}}\,\&, \qquad \frac{\vdash A, \boldsymbol{C}}{\vdash A \oplus B, \boldsymbol{C}}\,1\oplus \quad \frac{\vdash B, \boldsymbol{C}}{\vdash A \oplus B, \boldsymbol{C}}\,2\oplus.$$

Multiplicative rules:

$$\vdash \mathbf{1} \quad (\text{axiom}), \qquad \frac{\vdash \boldsymbol{A}}{\vdash \bot, \boldsymbol{A}}\,\bot,$$

$$\frac{\vdash A, \boldsymbol{C} \quad \vdash B, \boldsymbol{D}}{\vdash A \otimes B, \boldsymbol{C}, \boldsymbol{D}}\,\otimes, \qquad \frac{\vdash A, B, \boldsymbol{C}}{\vdash A \parr B, \boldsymbol{C}}\,\parr.$$

The additional possibilities of linear logic can be ascribed to the dumping of the structural rules: in presence of these rules, for instance the rule (&) and the rule (⊗) would amount to the same. Without structural rules, the *contexts* (i.e., the unspecified lists of formulas A, B, C, etc.) which occur in the rules take a greater importance. In particular, if we start with the idea of a conjunction, there are two reasonable ways to handle the contexts: by juxtaposition (connective ⊗) or by identification (connective &). By the way, observe that contexts behave like phases and this will be the basis of the completeness argument.

1.16. Proposition. *Linear sequent calculus is sound w.r.t. validity in phase structures.*

Proof. We have not defined the interpretation of a sequent $\vdash A$; but, obviously, $\vdash A$ must be read as the *par* of its components (and, when the list is void, as $\bot$). We prove that, whenever $\vdash A$ is provable, then it is valid in any phase structure $(P, \bot_P, (a_S))$. We argue by induction on a proof of A:

(i) The proof consists of the axiom $\vdash B, B^\perp$. $B \mathbin{\invamp} B^\perp = B \multimap B$ and, by Remark 1.10, $1 \in S(B \multimap B) = S(B) \multimap S(B)$.

(ii) The proof consists of the axiom $\vdash \top, A$. The interpretation is of the form $S(\top) \mathbin{\invamp} G$, with G as fact; equivalently, $S(\mathbf{0}) \multimap G$, but since $S(\mathbf{0})$ is the smallest fact of P, $\mathbf{0}_P$ is included in G and, by Remark 1.10, $1 \in S(\mathbf{0})$.

(iii) The proof consists of the axiom $\vdash \mathbf{1}$. The interpretation of $\mathbf{1}$ is $\mathbf{1}_P$ and we have already observed that $1 \in \mathbf{1}_P$ (Example 1.5(ii)).

(iv) The proof ends with a cut-rule. To prove the soundness of this rule, we have to show that, in case $1 \in G \mathbin{\invamp} H$ and $1 \in G^\perp \mathbin{\invamp} K$, then $1 \in H \mathbin{\invamp} K$. This is easily seen if one puts $G \mathbin{\invamp} H$, $G^\perp \mathbin{\invamp} K$, and $H \mathbin{\invamp} K$ under the forms $H^\perp \multimap G$, $G \multimap K$, and $H^\perp \multimap K$ and if one then applies Remark 1.10.

(v) The proof ends with an exchange rule: immediate.

(vi) The proof ends with (&). In order to prove the soundness of (&), we have to show that, in case $1 \in G \mathbin{\invamp} K$ and $1 \in H \multimap K$, then $1 \in (G \,\&\, H) \multimap K$. This is immediate from the distributivity law 1.13(i).

(vii) The proof ends with (1⊕) or (2⊕). In order to prove the soundness of these rules, we have for instance to show that, in case $1 \in G \mathbin{\invamp} K$, then $1 \in (G \oplus H) \mathbin{\invamp} K$, which is immediate from the half-distributivity of *par* w.r.t. *plus* (c.f. Property 1.13(ii)).

(viii) The proof ends with ($\bot$). In order to prove the soundness of ($\bot$), we have to show that, in case $1 \in G$, then $1 \in S(\bot) \mathbin{\invamp} G$; this is immediate from $\bot_P \multimap G = G$.

(ix) The proof ends with (⊗). In order to prove the soundness of (⊗), we have to show that, in case $1 \in F \mathbin{\invamp} G$, $1 \in H \mathbin{\invamp} K$, then $1 \in (F \otimes H) \mathbin{\invamp} G \mathbin{\invamp} H$. If these three facts are put as $G^\perp \multimap F$, $K^\perp \multimap H$, and $(G^\perp \otimes K^\perp) \multimap (F \otimes H)$, then we have, by hypothesis, $G^\perp \subset F$, $K^\perp \subset H$ from which one deduces $G^\perp \otimes K^\perp \subset F \otimes H$.

(x) The proof ends with (⅋): immediate. □

1.17. Theorem. *Linear sequent calculus is complete w.r.t. phase semantics.*

Proof. For the time of the proof, let us adopt the following convention: we are dealing with sequences of formulas $A, B, C, \ldots$ up to their order, i.e., we are dealing with *multisets* of formulas. Multisets of formulas form a commutative monoid, when equipped with concatenation $*$ and with $\emptyset$ as a neutral element. This defines a phase space M, and the antiphases of M will just be those multisets A such that $\vdash A$ is provable in linear sequent calculus (pedantically, in the variant without exchange, adapted to multisets of formulas). We claim the existence of a phase structure $(M, \bot_M, (a_S))$ such that, for all formulas A,

$$S(A) = \{B; \vdash A, B \text{ is provable in linear sequent calculus}\}. \tag{3}$$

Let us call the right-hand side of the equation (3) $\Pr(A)$.

A crucial observation is that sets $\Pr(A)$ are facts of M; more precisely, that $\Pr(A) = \Pr(A^\bot)^\bot$: if $B \in \Pr(A)$ and if $C \in \Pr(A^\bot)$, then $\vdash A, B$ and $\vdash A^\bot, C$ are both provable and, by CUT, so is $\vdash B, C$; hence, $B * C \in \bot_M$. This shows that $\Pr(A) \subset \Pr(A^\bot)^\bot$.

Conversely, if $B \in \Pr(A^\bot)^\bot$, observe that since $\vdash A, A^\bot$ is an axiom, $A \in \Pr(A^\bot)$; hence, $A * B \in \bot_M$, i.e., $\vdash A, B$ is provable; but then $B \in \Pr(A)$. It is therefore legitimate to define a phase structure S by setting $S(a) = \Pr(a)$ for any propositional letter a.

By induction on the formula A, we prove that $S(A) = \Pr(A)$:

(i) A is an atom a: by definition.

(ii) A is of the form $a^\bot$: because $S(a^\bot) = S(a)^\bot = \Pr(a)^\bot = \Pr(a^\bot)$.

(iii) A is $\bot$: observe that $\bot_M = \Pr(\bot)$ (immediate).

(iv) A is $\mathbf{1}$: as in (ii).

(v) A is $\top$: the axiom for $\top$ yields $S(\top) = M = \top_M$.

(vi) A is $\mathbf{0}$: as in (ii).

(vii) A is $B \,\&\, C$: it is easily seen that $\vdash B \,\&\, C, D$ is provable iff $\vdash B, D$ and $\vdash C, D$ are both provable. From this we get $\Pr(A \,\&\, B) = \Pr(A) \,\&\, \Pr(B)$ which is what is needed to go through this step.

(viii) A is $B \oplus C$: from $\Pr(A \,\&\, B) = \Pr(A) \,\&\, \Pr(B)$, we easily obtain $\Pr(A \oplus B) = \Pr(A) \oplus \Pr(B)$, etc.

(ix) A is $B \otimes C$: $\Pr(A).\Pr(B) \subset \Pr(A \otimes B)$ (by the $\otimes$-rule); hence, by Lemma 1.13.1, $\Pr(A) \otimes \Pr(B) \subset \Pr(A \otimes B)$. Conversely, if $C \in \Pr(A \otimes B)$ and $D \in (\Pr(A).\Pr(B))^\bot$, then $\vdash D, A^\bot, B^\bot$ is provable and, by the rule ($\parr$) and a cut, we get a proof of $\vdash D, C$. This shows the reverse inclusion, namely, $\Pr(A \otimes B) \subset \Pr(A) \otimes \Pr(B)$. From this we can easily conclude our statement.

(x) A is $B \parr C$: similar to (viii).

Now, assume that A is a linear tautology; then A is valid in S, which means that $\emptyset \in S(A) = \Pr(A)$; but then $\vdash A$ is provable in linear sequent calculus. $\square$

1.18. Remarks. (i) Usually, a completeness is stated, not for logical calculi, but for theories. We have given absolutely no meaning to the concept of 'linear logical theory'. The reason is that if we were allowing an axiom, say A, to say that B is provable with the help of A, this has nothing to do with the provability of $A \multimap B$;

in fact, this is equivalent to the provability of $!A \multimap B$, where "!" is the connective 'of course', which has not yet been introduced.

(ii) The phase spaces P have been supposed to be commutative. If P were not commutative and if we restrict our attention to *distinguished* subsets of P (i.e., $p.G = G.p$ for all $p \in P$) (in particular, $\perp_P$ must be distinguished), then one can develop the same theory as we did, and the connectives $\otimes$ and $\parr$ are still commutative. Of course, when we were thinking of a possible 'noncommutative linear logic', this is not the straightforward generalization that we have in mind, but something for which the multiplicatives would be noncommutative.

Now that the main features of linear logic have been disclosed, let us complete the language to obtain a predicate calculus with the same expressive power as the usual ones. The first effort has to be made on the propositional part: one must be able to recover somewhere the structural rules. Let us call $?A$ a formula obtained from A which is 'A-saturated w.r.t. the structural rules'. The fact that $?A$ comes from A can be expressed as $A \subset ?A$; the fact that one can weaken on $?A$ can be expressed as $\perp \subset ?A$, and the fact that one can contract on $?A$ as $?A \parr ?A \subset ?A$. But there are also other properties of $?A$ and everything will be summarized in the following definition.

1.19. Definition. A *topolinear space* $(P, \perp, \mathbb{F})$ consists in a phase space $(P, \perp)$ together with a set $\mathbb{F}$ of facts of $(P, \perp)$, the *closed facts*; the following conditions are required:

(i) $\mathbb{F}$ is closed under arbitrary *with*; in particular, $P \in \mathbb{F}$.

(ii) $\mathbb{F}$ is closed under finite *par*; in particular, $\perp \in \mathbb{F}$.

(iii) $\perp$ is indeed the smallest fact of $\mathbb{F}$: $\perp \subset F$ for all $F \in \mathbb{F}$.

(iv) *par* is idempotent on $\mathbb{F}$: $F \parr F = F$ for all $F \in \mathbb{F}$.

The linear negations of closed facts are called *open facts.*

1.20. Definition. The exponential connectives or *modalities* are the connectives *of course* and *why not*; both of them are definable in topolinear spaces:

(i) If G is a fact, then its *affirmation* (connective *of course*) $!G$ is defined as the greatest (w.r.t. inclusion) open fact included in G (the *interior* of G).

(ii) If G is a fact, then its *consideration* (connective *why not*) $?G$ is defined as the smallest (w.r.t. inclusion) closed fact containing G (the *closure* of G).

We now adapt our syntax so that we can handle the two modalities; first, the *definition* of linear negation is extended to the modalities by

$$(!A)^\perp = ?(A^\perp), \qquad (?A)^\perp = !(A^\perp).$$

Then we have to give sequent rules for the modalities. In the definition below, $?C$ is short for a sequence $?C_1, \ldots, ?C_n$.

1.21. Definition. Linear sequent calculus is enriched with the following rules for modalities:

Exponential rules:

$$\frac{\vdash A, \mathbf{B}}{\vdash ?A, \mathbf{B}}D? \qquad \text{(dereliction)},$$

$$\frac{\vdash \mathbf{B}}{\vdash ?A; \mathbf{B}}W? \qquad \text{(weakening)}, \qquad \frac{\vdash A, ?\mathbf{B}}{\vdash !A, ?\mathbf{B}}!,$$

$$\frac{\vdash ?A, ?A, \mathbf{B}}{\vdash ?A, \mathbf{B}}C? \qquad \text{(contraction)}.$$

1.22. Theorem. *The calculus as extended in Definition* 1.21 *is sound and complete w.r.t. validity in topolinear structures, i.e., the obvious analogue of phase structures where phase spaces have been replaced by topolinear spaces.*

Proof. (i): *soundness.* The soundness of dereliction comes from $F \subset ?F$. The soundness of weakening comes from $\bot \subset ?F$ and the soundness of contraction from $?F = ?F \parr ?F$. The soundness of (!) is the fact that if $F \subset ?G_1 \parr \cdots \parr ?G_n$, then $?F \subset ?G_1 \parr \cdots \parr ?G_n$. This is because $?G_1 \parr \cdots \parr ?G_n$ is a closed fact and because $?F$ is the smallest closed fact containing F.

(ii): *completeness.* We define the phase space M as in the proof of Theorem 1.17. Now we define a topolinear structure $\mathbb{F}$ on M by saying that the closed facts are just arbitrary intersections (i.e., *with*) of facts of the form $S(?A)$.

1.22.1. Lemma. *The set* $\mathbb{E}$ *of all facts of the form* $S(?A)$ *satisfies properties* (ii)–(iv) (*Definition* 1.19) *of topolinear spaces.*

Proof. (ii): The axiomatic system just described proves the linear equivalence $?(A\oplus B) \multimapboth (?A) \parr (?B)$ ($C \multimapboth D$ is short for $(C \multimap D) \,\&\, (D \multimap C)$):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash A^\perp, A}{\vdash A^\perp, A\oplus B}1\oplus}{\vdash A^\perp, ?A\oplus B}D?}{\vdash !A^\perp, ?A\oplus B}! \quad \dfrac{\dfrac{\dfrac{\vdash B^\perp, B}{\vdash B^\perp, A\oplus B}2\oplus}{\vdash B^\perp, ?A\oplus B}D?}{\vdash !B^\perp, ?A\oplus B}!}{\vdash !A^\perp \otimes !B^\perp, ?A\oplus B, ?A\oplus B}\otimes}{\vdash !A^\perp \otimes !B^\perp, ?A\oplus B}C?}{\vdash (?A \parr ?B) \multimap ?(A\oplus B)}\parr \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash A^\perp, A}{\vdash A^\perp, ?A}D?}{\vdash A^\perp, ?A, ?B}W? \quad \dfrac{\dfrac{\vdash B^\perp, B}{\vdash B^\perp, ?B}D?}{\vdash B^\perp, ?A, ?B}W?}{\vdash A^\perp \,\&\, B^\perp, ?A, ?B}\&}{\vdash !A^\perp \,\&\, B^\perp, ?A, ?B}!}{\vdash !A^\perp \,\&\, B^\perp, ?A \parr ?B}\parr}{\vdash (?(A\oplus B)) \multimap ?A \parr ?B}\parr}{}$$

$$\dfrac{\vdash (?A \parr ?B) \multimap ?(A\oplus B) \qquad \vdash (?(A\oplus B)) \multimap ?A \parr ?B}{\vdash (?A \parr ?B) \multimapboth ?(A\oplus B)}\&$$

From this linear equivalence, it is plain that $S(?A) \parr S(?B) = S(?A \parr ?B) = S(?(A\oplus B))$, so $\mathbb{E}$ is closed by *par*.

(iii): The equivalence $\perp \multimap\!\circ\ ?0$ is easily proved. From this, $\perp \in \mathbb{E}$; also the weakening rule shows that $M(\perp) \subset M(?A)$.

(iv): From the equivalence $A \multimap\!\circ\ A \oplus A$, it is easy to obtain $?A \multimap\!\circ\ ?(A \oplus A)$, and so $?A \multimap\!\circ\ (?A \parr ?A)$. From this we get $S(?A) = S(?A) \parr S(?A)$. □

Proof of Theorem 1.22 (*continued*). Now, $\mathbb{F}$ is made of arbitrary intersections of elements of $\mathbb{E}$. From this it is immediate that $\mathbb{F}$ satisfies properties (i) and (iii) of topolinear spaces.

Now, if we observe that $\parr$ is distributive w.r.t. arbitrary intersections, it is not difficult to get (ii) and (iv) for $\mathbb{F}$ from Lemma 1.22.1. For instance,

$$\left(\bigcap S(?A_i)\right) \parr \left(\bigcap S(?A_i)\right) = \bigcap S(?A_i) \parr S(?A_i)$$

because $S(?A_i) \parr S(?A_j) \supset S(?A_i)$, which in turn follows from $S(?A_j) \supset \perp$, etc.

We have established that $\mathbb{F}$ is a topolinear structure. Now, $\mathbb{F}$ is defined in such a way that, in $\mathbb{F}$,

$$?S(A) = \bigcap\{S(?B); S(A) \subset S(?B)\} = \bigcap\{S(?B); \vdash A \multimap ?B \text{ is provable}\}.$$

But, by rule (!), $\vdash A \multimap ?B$ is provable iff $\vdash ?A \multimap ?B$ is; i.e., we finally find out that $?S(A) = S(?A)$ and this is enough to finish the completeness argument for Theorem 1.22. □

1.23. Remark. In spite of the obvious analogies with the modalities of modal logic, it is better to keep distinct notations, for at least two reasons:

(ii) ! and ? are modalities within linear logic, while □ and ◇ are modalities inside usual predicate calculus; in particular, the rule (!) which looks like the introduction rule of □ is perhaps not so close since the sequents are handled in a completely different way.

(ii) The adoption of the modal symbolism would convey the idea of 'yet another modal system', which is not quite the point of linear logic....

1.24. Definition. The *quantifiers* are semantically defined as infinite generalizations of the additives; these quantifiers are $\bigwedge$ (*any*) and $\bigvee$ (*some*). The semantic definitions are straightforward and boring and are left to the reader. The syntactic rules are

Quantifier rules:

$$\frac{\vdash A, B}{\vdash \bigwedge x A, B}\bigwedge, \qquad \frac{\vdash A[t/x], B}{\vdash \bigvee x A, B}\bigvee.$$

($\bigwedge$) is subject to the familiar restriction on variables: x not free in B. If the reader has written the boring definition of the semantics, he can in turn prove the following boring theorem.

1.25. Theorem. *Phase semantics is sound and complete w.r.t. linear predicate calculus.*

1.26. Remark. $\bigwedge$ and $\bigvee$ are De-Morgan dual; $\bigvee$ is effective, which does not prevent it from being a linear negation!

2. Proof-nets

To understand what is going on in this section, think of the two formalisms for the intuitionistic fragment ($\wedge, \rightarrow, \forall$): NJ (natural deduction) and LJ (sequent calculus). In LJ, the portion of proof from $A, C, E \vdash F$ to $A \wedge B, C \wedge D, E \vdash F$ can be written in two different ways, whereas in NJ there is only one way. According to Prawitz, the proofs in NJ should be viewed as true ones, while those in LJ are no longer primitive, i.e., a proof in LJ gives us instructions enabling us to recover a real proof in NJ. The issue is important, especially if we think of equality of normal forms and related questions. Unfortunately, already in the case of intuitionistic logic, something goes wrong with NJ when we consider the full language: the proofs in NJ cannot any longer be viewed as primitive since one needs additional identifications between them (*commutation rules*). The attempts to fix these defects by means of 'multiple conclusion logics' were never convincing.

Linear logic is faced with the same question with a greater acuteness: on the one hand, since linear logic is constructive, linear proofs must be seen as programs whose execution involves a normal-form theorem. On the other hand, linear logic is 'classical' to some extent, which means that the commutation problems are bigger than they are in the intuitionistic case. These constraints made it necessary to have a second look at this question of 'multiple conclusion logic', but now making use of the linear connectives which enable us to make distinctions that were not accessible to people working on the same subject with usual logics.

Proof-nets, which we are going to develop in this section, are the result of these investigations. The core of the theory is the multiplicative fragment, which works in an extremely satisfactory way. The extension of the theory to the full language is not so good, but good enough to keep a reasonable fragment of the calculus free from any criticism. We first start with the *multiplicative fragment*: by this we mean the formulas built from atoms $a, b, \ldots, a^{\perp}, b^{\perp}$, by means of the connectives $\otimes$ and ⅋. There are three basic ingredients that will be used to produce proof-nets.

(i) Axioms:

$$\overline{A \quad A^{\perp}},$$

(ii) $\otimes$-rules:

$$\frac{A \quad B}{A \otimes B},$$

(iii) ⅋-rules:

$$\frac{A \quad B}{A ⅋ B}.$$

(The cut-rule, which has exactly the same geometrical structure as the ⊗-rule is omitted from this study.)

Using these three patterns, we can construct (in a way not difficult to imagine) *proof-structures* with several conclusions $A_1, \ldots, A_n$. We would like to make sure that such structures are sound, i.e., that they prove the *par* of their conclusion, i.e., $A_1 \parr \cdots \parr A_n$. With usual proofs (written as trees), the soundness condition is local (i.e., one checks each rule separately), while, in a pattern with several conclusions, the left-hand side cannot act in contradiction with the right-hand side as typified in Fig. 2. Fig. 2(a) is sound, since $\vdash A \otimes B,\ A^\perp \parr B^\perp$ is provable, but Fig. 2(b) is not. In fact, (a) is sound because the two rules used are distinct. But what is the general global criterion of correctness?

(a) (b)

Fig. 2.

2.1. Definition. A *proof-structure* consists of the following objects:

(i) Occurrences of formulas (i.e., to be pedantic, pairs (A, i), where A is the formula and the second component i serves to distinguish between two occurrences, etc.); there must be at least one formula occurring.

(ii) A certain number of *links* between these (occurrences of) formulas. These links, which must be viewed as binary or ternary relations, are of the three kinds already mentioned:

Axiom link: $\mathrm{AX}(A, i; A^\perp, j)$ between an occurrence of A and an occurrence of $A^\perp$. This link is considered to be symmetric, i.e., it is the same as $\mathrm{AX}(A^\perp, j; A, i)$. This link has no *premise* and two *conclusions*, namely (A, i) and $(A^\perp, j)$.

Times link: $\otimes(A, i; B, j; A\otimes B, k)$. This link is not the same as $\otimes(B, j; A, i; A\otimes B, k)$; i.e., the link is not symmetric. The *premises* of the link are (A, i) and (B, j), and its *conclusion* is $(A\otimes B, k)$.

Par link: $\parr(A, i; B, j; A \parr B, k)$, whose premises are (A, i) and (B, j) and whose conclusion is $(A \parr B, k)$. This link is not symmetric.

(iii) We require that

- every occurrence of a formula in the structure is the conclusion of one and only one link;
- every occurrence of a formula in the structure is the premise of at most one link.

2.2. Examples. What we have pedantically defined as a proof-structure corresponds to the rough idea of starting with the axiom links and then developing the proof in a tree-like way by using the ⊗- and ⅋-rules. In particular, Fig. 2(a) and (b) are equally satisfactory as proof-structures. One may imagine more pathological examples, e.g., nonconnected proof-structures:

$$\overline{A \quad A^{\perp}} \qquad \overline{B^{\perp} \quad B}.$$

(In the sequel, proof-nets will be connected.)

We shall content ourselves with a graphical representation for proof-structures of the kind already used. Such a representation is rarely ambiguous and so easy to understand. Let us however recall that in an axiom link, the left formula is here for convenience (symmetry of the link), while in the two other types of link, left and right have a real meaning. We shall not define obvious notions such as being above/below a formula, or being connected. By the way, we have already started with our last abuse: we speak of a formula instead of an occurrence, any time this is not ambiguous.

2.1. The concept of trip

The problem is to determine which among the proof-structures are logically sound; such structures will be called *nets*. For this we shall define the concept of *trip*, which is better understood within a temporal imagery and the idea of something (a particle, information, etc.) travelling through the proof-structure.

(i) The 'time' will be *cyclic*, discrete and finite. In general, the trips will be scheduled according to some Z/kZ, but the choice of the origin of time will be just a convention with no meaning. The problem with cyclic time is that 'after' and 'before' are not defined globally. However, if t and u are two distinct times, the *interval* $[t, u]$ makes sense: choose a determination r between 0 and $k-1$ for $u-t$; then, $[t, u]$ consists of the time moments $t, t+1, \ldots, t+r=u$.

(ii) Each formula will be viewed as a box (see Fig. 3) in which our particle may travel. The idea is that the particle will enter A by the gate marked "↥" and will

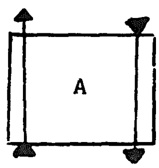


Fig. 3.

immediately go out through the gate marked "↑". These two operations are performed in the same unit of time $t^{\wedge}$; $t^{\wedge}$ is used only for this purpose; i.e., at $t^{\wedge}$ the particle is in A between "↥" and "↑" and nowhere else. The particle will re-enter (but there is no 'before', no 'after') A by "↧" and will immediately go out through "↓", all this at another moment $t_{\vee}$. Now the particle exits through a gate with a direction; at the moment just after this exit, it will enter another formula by using the links of the proof-structure. Sometimes, the trip can only be done in one way, but very often, there are several ways of continuing, and these ways are selected by *switches*.

(1) *Axiom link*: The picture in Fig. 4 is clear: immediately after exiting through gate $A\uparrow$, the particle enters $A^{\perp}\downarrow$; immediately after exiting through gate $A^{\perp}\uparrow$ the particle enters $A\downarrow$. In other terms, $t(A^{\perp}_{\vee}) = t(A^{\wedge})+1$ and $t(A_{\vee}) = t(A^{\perp\wedge})+1$.

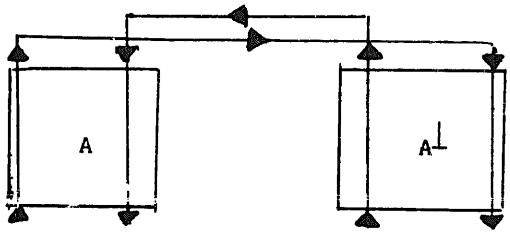


Fig. 4.

(2) *Terminal formula*: This is the case of a formula which is not a premise (see Fig. 5), i.e., just after exiting through $A\downarrow$, the particle re-enters through $A\uparrow$. In other terms, $t(A^{\wedge}) = t(A_{\vee})+1$.

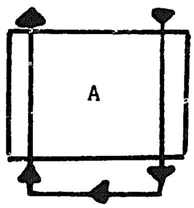
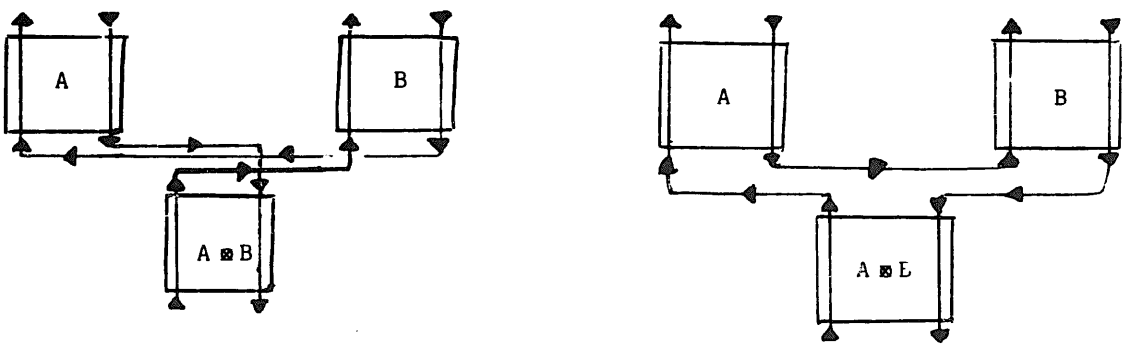


Fig. 5.

(3) *Times link*: Associated with such a link is a *switch* with two positions "L" and "R". The switch is set on one of these two positions for the whole trip, independently of the positions of other switches. According to the position of the switch, the particle moves as shown in Fig. 6.



(a) (b)

Fig. 6. (a) switch on "L". (b) switch on "R".

Looking carefully, we see that the only difference between "L" and "R" is the interchange between A and B. The terminology "L" comes from the fact that we reach the conclusion ($A \otimes B\downarrow$) through the left premise ($A\downarrow$) when the switch is on

"L". The same reason is behind the terminology for the *par* link.

$$\text{“L”:}\quad t(B^{\wedge})=t(A\otimes B^{\wedge})+1,\qquad t(A^{\wedge})=t(B_{\vee})+1,\qquad t(A\otimes B_{\vee})=t(A_{\vee})+1;$$

$$\text{“R”:}\quad t(A^{\wedge})=t(A\otimes B^{\wedge})+1,\qquad t(B^{\wedge})=t(A_{\vee})+1,\qquad t(A\otimes B_{\vee})=t(B_{\vee})+1.$$

(4) *Par link* (see Fig. 7): Associated with such a link is a switch with two positions "L" and "R"; here too, the switch is set once and for all for the trip, independently

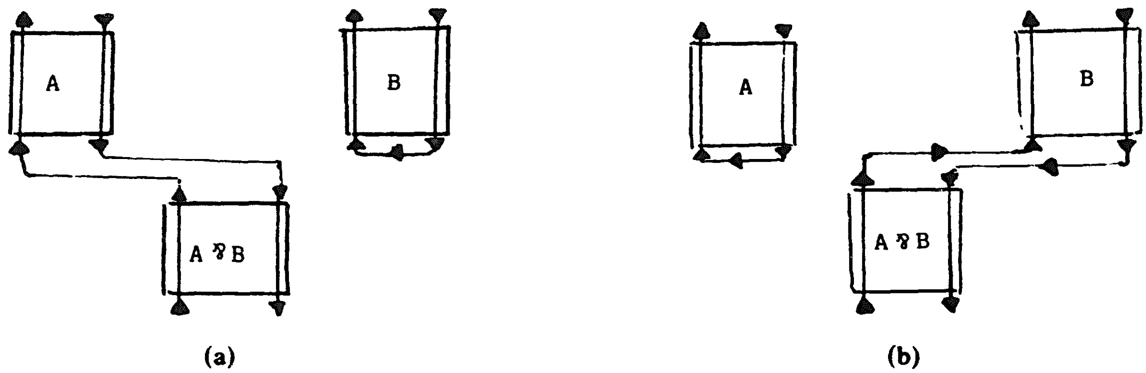


Fig. 7. (a) switch on "L". (b) switch on "R".

from other switches. The two positions of the switch lead to symmetric travels of the particle.

$$\text{“L”:}\quad t(A^{\wedge})=t(A\parr B)^{\wedge}+1,\qquad t(A\parr B_{\vee})=t(A_{\vee})+1,\qquad t(B^{\wedge})=t(B_{\vee})+1;$$

$$\text{“R”:}\quad t(B^{\wedge})=t(A\parr B)^{\wedge}+1,\qquad t(A\parr B_{\vee})=t(A_{\vee})+1,\qquad t(A^{\wedge})=t(A_{\vee})+1.$$

(iii) Everything is now ready for the trip, which can be made as follows: we set the switches of all *par* and *times* links on arbitrary positions (so there are 2^n possibilities if n is the number of switches). We select an arbitrary formula A and an exit gate ($A\uparrow$ or $A\downarrow$) at time 0. Then there are clear, unambigous conditions to go on forever. Since the whole structure is finite, the trip is eventually periodic, but since the same remark can be made if one inverts the sense of time, this concretely forces the existence of a strictly positive integer k such that, at time k, the particle enters through the dual gate ($A\underline{\downarrow}$ or $A\overline{\uparrow}$ respectively); the number k with this property can be chosen minimal and then we have obtained a *trip* parameterized by Z/kZ. Now, two possibilities arise:

(a) $k<2p$, where p is the number of formulas of the structure; the trip is called a *shorttrip*.

(b) $k=2p$, with p as above; the trip is called a *longtrip*.

2.3. Examples. Go back to the examples in Fig. 2 and, in both cases, set all switches on "L": in Fig. 2(a) we get a longtrip, namely,

$$A^{\wedge}, A^{\perp}_{\vee}, A^{\perp}\parr B^{\perp}_{\vee}, A^{\perp}\parr B^{\perp\wedge}, A^{\perp\wedge}, A_{\vee}, A\otimes B_{\vee}, A\otimes B^{\wedge}, B^{\wedge}, B^{\perp}_{\vee}, B^{\perp\wedge}, B_{\vee}, A^{\wedge}, \ldots .$$

Here, each formula has been passed in both senses. In Fig. 2(b), however, we get

$$A^{\wedge}, A^{\perp}_{\vee}, A^{\perp}\otimes B^{\perp}_{\vee}, A^{\perp}\otimes B^{\perp\wedge}, B^{\perp\wedge}, B_{\vee}, A^{\wedge}, \ldots,$$

a typical shorttrip. These examples are enough to give some evidence for the following definition.

2.4. Definition. A proof-structure is said to be a *proof-net* when it admits no shorttrip.

2.5. Reformulation. We reformulate the definition of proof-net in more usual terms: A proof structure with p formulas and n switches is said to be a *proof-net* when, for any position of the switches, there is a bijection $t(.)$ between $Z/2pZ$ and the $2p$ distinct exits of the structure such that, for any two exits e, e', $t(e') = t(e)+1$ iff e' comes immediately after e in the travel process that has been exposed in full details in the introduction of Section 2.1 (w.r.t. the positions of the switches, of course).

2.6. Remark. Checking that a proof-structure has no shorttrip requires looking at 2^n different cases. This number can be decreased to 2^{n-1}: let (1) and (2) be two positions of the switches, (2) being obtained from (1) by commuting all $\otimes$-switches to the other position. It is easily checked that (2) is the same as (1) *but with time reversed* and, in particular, it suffices to check only one of them. Anyway, the soundness condition is not feasible. However, it is not part of our intentions to *check* soundness by concrete means. The proof-nets we shall deal with in practice will all come from sequent calculus or will be obtained from other proof-nets by means of transitions preserving the soundness condition. Hence, the soundness condition is an abstract notion (just like, say, semantic soundness), whose importance lies in its relation to linear sequent calculus. We shall now prove that proof-nets are the 'natural deduction of linear sequent calculus'.

2.7. Theorem. *If π is a proof in linear sequent calculus of $\vdash A_1, \ldots, A_n$ (in fact, in the multiplicative fragment without cut), then we can naturally associate with π a proof-net π^- whose terminal formulas are exactly (one occurrence of) $A_1, \ldots,$ (one occurrence of) A_n.*

Proof. The proof-net π^- is defined by induction on π as follows.

Case 1: π is an axiom $\vdash A, A^{\perp}$; obviously, one must define π^- as

$$\overline{A \qquad A^{\perp}}$$

which is a proof-net: there is no switch, and only one longtrip $A^{\wedge}$, $A^{\perp}_{\vee}, A^{\perp\wedge}, A_{\vee}, A^{\wedge}, \ldots.$

Case 2: π is obtained from λ by an exchange rule; take $\pi^- = \lambda^-$.

Case 3: π is obtained from λ by ($\parr$):

$$\frac{\overset{\lambda}{\vdash A, B, C}}{\vdash A \parr B, C}.$$

By induction hypothesis, we have obtained λ^-, in which we can individualize A and B and π^- is obtained as follows

$$\frac{\overset{\lambda^-}{A \quad B}}{A \parr B}.$$

We have to prove the soundness of π^-: Setting all switches on arbitrary positions in π^- and assuming that the switch of the new link is on, say "L", we can start a trip at $t=0$ and $A^\wedge$; since λ^- is sound, at time $2n-1$ (where n is the number of formulas of λ^-) we arrive at $A_\vee$; then the trip is easily finished by adding $A \parr B_\vee, A \parr B^\wedge$ at times $2n, 2n+1$. We have obtained a longtrip, so π^- is correct.

Case 4: π is obtained from λ and μ by ($\otimes$):

$$\frac{\overset{\lambda}{\vdash A, C} \qquad \overset{\mu}{\vdash B, D}}{\vdash A \otimes B, C, D}.$$

By induction hypothesis, we have obtained λ^- and μ^-, in which we can respectively individualize A and B. π^- is obtained as follows:

$$\frac{\overset{\lambda^-}{A} \qquad \overset{\mu^-}{B}}{A \otimes B}.$$

Again, set all switches of π^- for a trip, and let us say that the last switch is on "R". Assume that there are n formulas in λ^-, m formulas in μ^-: starting with $A^\wedge$ at time $t=0$, we arrive at $A_\vee$ at time $t=2n-1$ (soundness of λ^-); then, at time $t=2n$, we move at $B^\wedge$ and so, since μ^- is sound, we arrive at $B_\vee$ at time $t=2n+2m-1$. Then, at times $2n+2m$ and $2n+2m+1$, we visit $A \otimes B_\vee$ and $A \otimes B^\vee$, therefore ending a longtrip. Hence, π^- is sound. □

2.8. Remark. The transformation $\pi \rightsquigarrow \pi^-$ identifies proofs which differ by the order of rules. For instance, the proofs

$$\dfrac{\dfrac{\dfrac{\vdash A, A^\perp \quad \vdash B, B^\perp}{\vdash A \otimes B, A^\perp, B^\perp}\otimes}{\vdash A \otimes B, A^\perp \parr B^\perp}\parr \quad \vdash C, C^\perp}{\vdash (A \otimes B) \otimes C, A^\perp \parr B^\perp, C^\perp}\otimes \qquad \dfrac{\dfrac{\dfrac{\vdash A, A^\perp \quad \vdash B, B^\perp}{\vdash A \otimes B, A^\perp, B^\perp}\otimes \quad \vdash C, C^\perp}{\vdash (A \otimes B) \otimes C, A^\perp, B^\perp, C^\perp}\otimes}{\vdash (A \otimes B) \otimes C, A^\perp \parr B^\perp, C^\perp}\parr$$

are the same, up to the order of the rules. Both are represented by the same proof-net, shown in Fig. 8. The question is to prove the converse of Theorem 2.7, namely that

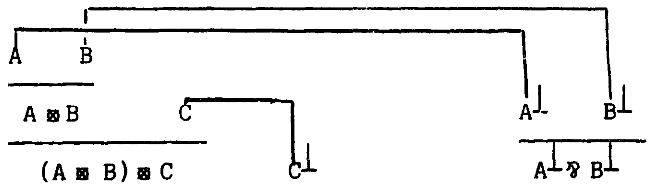


Fig. 8.

a proof-net can always be written as π^- for an appropriate π. The difficulty is that several π's are possible, and the converse is a very subtle result.

2.9. Theorem. *If β is a proof-net, one can find a proof π in sequent calculus such that $\beta = \pi^-$.*

Proof. By induction on the number of links in β:

(i) If β has exactly one link, then β must be of the form $\overline{A \quad A^\perp}$: in that case, the claim is proved by taking as π the axiom $\vdash A, A^\perp$.

(ii) If β has more than one link, then one of these links must be a *par* or a *times* link (otherwise, β is not connected and it is immediate that, in a nonconnected proof-structure, all trips are short). Hence, there is a terminal formula which is the conclusion of a *par* or a *times* link. In this case, we state the hypothesis that one can find such a terminal formula as the conclusion of a *par* link: write β as

$$\begin{array}{c}\beta' \\ \dfrac{A \quad B}{A ⅋ B},\end{array}$$

where β' is the *proof-structure* obtained by restriction. β' has one link less than β. Furthermore, β' is a proof-net: setting all switches for a trip in β', and setting the additional switch of β on "L", we get a longtrip

$$A^\wedge, \ldots, B_\vee, B^\wedge, \ldots, A_\vee, A ⅋ B_\vee, A ⅋ B^\wedge, A^\wedge, \ldots$$

in β. But this shows the existence of a longtrip

$$A^\wedge, \ldots, B_\vee, B^\wedge, \ldots, A_\vee, A^\wedge, \ldots$$

in β', and this established the soundness of β'.

The induction hypothesis has built π' such that $\pi'^- = \beta'$. Then, for π one can take the proof:

$$\begin{array}{c}\pi' \\ \dfrac{\vdash A, B, C}{\vdash A ⅋ B, C}\,⅋\end{array}$$

(exchanges have not been indicated).

(iii) In this case we assume that there is more than one link, but that no terminal formula is the conclusion of a *par* link. The temptation is to split the proof-net as follows:

$$\frac{\overset{\beta'}{A} \quad \overset{\beta''}{B}}{A \otimes B},$$

where β' and β'' are disjoint subproof-nets of β. Of course, we can try to make such a splitting for each terminal conclusion of a *times* link in β and in simple examples it turns out that if we choose the wrong terminal formula, then the splitting does not work. For instance, the proof-net in Fig. 9 has three terminal ⊗-links but one

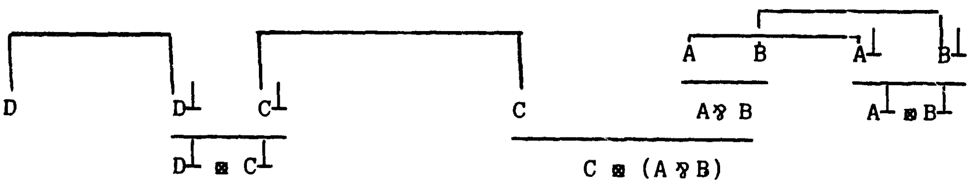


Fig. 9.

of them (the one ending with $A^\perp \otimes B^\perp$) cannot be split. The proof-net can be split at the two other terminal links, and the non-unicity of the solution of splitting problems makes it even more complicated. We shall now try to prove the existence of a splitting from which the last case of the theorem will easily follow. Let us remark however that the hypothesis 'no terminal ⅋-link' is needed since the net in Fig. 10 cannot be split.

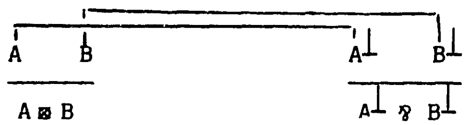
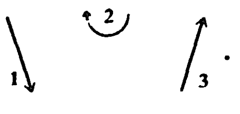


Fig. 10.

2.9.1. Lemma. *Let β be a proof-net and consider (w.r.t. a certain position of the switches) the respective times of passage of the particle through $A_\vee$, $A\otimes B^\wedge$, $B_\vee$, say t_1, t_2, t_3, in a given ⊗-link of β; then $t_2 \in [t_1, t_3]$. In other terms, in case "L", the particle travels as follows:*

and not as:



Proof. Assume, for a contradiction, that $t_3 \in [t_1, t_2]$ and assume that the switch of our ⊗-link is set on, say "L". Then the trip runs as follows:

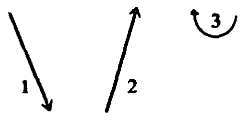
$$A_\vee, A\otimes B_\vee, \ldots, B_\vee, A^\wedge, \ldots, A\otimes B^\wedge, B^\wedge, \ldots, A_\vee.$$

Now commute the switch to "R" and observe that we get a shorttrip

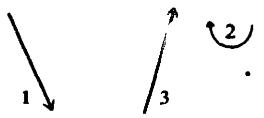
$$A\otimes B_\vee, B_\vee, \ldots, A\otimes B_\vee, \ldots.$$

(The portions in dots in the original trip are common to both trips.) □

2.9.2. Lemma. *Let β be a proof-net and consider (w.r.t. a certain position of the switches) the respective times t_1, t_2, t_3 of passage of the particle through the gates $A_\vee$, $A^\wedge$ and $B_\vee$ of a given $\parr$-link; then $t_2 \in [t_1, t_3]$. In other terms, in case "L", the order of passage is*



and not:



Proof. Very similar to the proof of Lemma 2.9.1. Assume, for a contradiction, that $t_3 \in [t_1, t_2]$. Then (if our $\parr$-switch is on "L"), the trip runs as follows:

$$A_\vee, A \parr B_\vee, \ldots, B_\vee, B^\wedge, \ldots, A \parr B^\wedge, A^\wedge, \ldots, A_\vee.$$

Setting the switch on the position "R", we get the shorttrip $A^\wedge, \ldots, A_\vee, A^\wedge$. □

2.9.3. Definition. Assume that $\frac{A \quad B}{A\otimes B}$ is a given $\otimes$-link in a proof-net β. We define the *empires* eA and eB of A and B as follows: eA consists of those formulas C such that, for any trip leaving $A^\wedge$ at time t_1 and returning at $A_\vee$ at time t_2 and passing through $C^\wedge$ and $C_\vee$ at times u_1 and u_2, we have $u_1, u_2 \in [t_1, t_2]$. The empire eB is defined in a symmetric way.

2.9.4. Facts

(i) $A \in eA$.

(ii) $eA \cap eB = \emptyset$.

(iii) *If $C \in eA$, and C is linked to $C^\perp$ by $\overline{C \quad C^\perp}$, then $C^\perp \in eA$.*

(iv) *If $C \otimes D \in eA$ is the conclusion of a link $\frac{C \quad D}{C\otimes D}$, then $C, D \in eA$.*

(v) *If $C \parr D \in eA$ is the conclusion of a link $\frac{C \quad D}{C \parr D}$, then $C, D \in eA$.*

(vi) *If $\frac{C \quad D}{C\otimes D}$ is a $\otimes$-link distinct from the link $\frac{A \quad B}{A\otimes B}$ and if $C \in eA$ (respectively $D \in eA$), then $C \otimes D \in eA$.*

(vii) *If $\frac{C \quad D}{C \parr D}$ is a $\parr$-link and $C, D \in eA$, then $C \parr D \in eA$.*

Proof. (i): self-evident.

(ii): By Lemma 2.9.1, the intervals $A^\wedge, \ldots, A_\vee$ and $B^\wedge, \ldots, B_\vee$ in a given trip are necessarily disjoint, etc.

(iii): This is because the gates of $C^\perp$ are crossed just one step before or after the gates of C.

(iv): Take a trip with the $C \otimes D$-switch on "L". Then, since $C_\vee$ and $D^\wedge$ are passed at one step from $C \otimes D_\vee$ and $C \otimes D^\wedge$, $C_\vee$ and $D^\wedge$ belong to the interval $A^\wedge, \ldots, A_\vee$. Now, imagine that the two consecutive times of passage through $D_\vee$ and $C^\wedge$ are outside the interval $A^\wedge, \ldots, A_\vee$: then the interval $B^\wedge, \ldots, D_\vee$ (or $A \otimes B_\vee, \ldots, D_\vee$) is performed without passing through any of the two links $\frac{A \quad B}{A \otimes B}$ and $\frac{C \quad D}{C \otimes D}$. Now, let us commute the switch $\frac{C \quad D}{C \otimes D}$ to "R". It is immediate (for reasons of symmetry) that $D_\vee$ belongs to the interval $A^\wedge, \ldots, A_\vee$ of this new trip. However, the part in dots $B^\wedge, \ldots, D_\vee$ (or $A \otimes B_\vee, \ldots, D_\vee$) is common to both trips, and does not contain A; a contradiction. We arrive at the conclusion that the four exits of C and D are passed between $A^\wedge$ and $A_\vee$.

(v): Assume that the switch of our ⅋-link is on "L". Then the times of passage through $C_\vee$ and $C^\wedge$ being one step from the times of passage through $C \otimes D_\vee$ and $C \otimes D^\wedge$, it follows that $C_\vee, C^\wedge \in A^\wedge, \ldots, A_\vee$. Now, assume that the consecutive times of passage through $D_\vee$ and $D^\wedge$ are outside the interval $A^\wedge, \ldots, A_\vee$. Then we conclude that the interval $B^\wedge, \ldots, D_\vee$ (or $A ⅋ B_\vee, \ldots, D_\vee$) is performed without passing through the links $\frac{A \quad B}{A \otimes B}$ and $\frac{C \quad D}{C ⅋ D}$. Form another trip by setting our ⅋-switch to "R" and derive a contradiction as in (iii).

(vi): Consider a trip with the switch $\frac{C \quad D}{C \otimes D}$ on "L". Then, by hypothesis, the times of passage through $C_\vee$ and $C^\wedge$ are within the interval $A^\wedge, \ldots, A_\vee$. We have the following possibilities, using Lemma 2.9.1:

$$A^\wedge, \ldots, C_\vee, C \otimes D_\vee, \ldots, C \otimes D^\wedge, D^\wedge, \ldots, D_\vee, C^\wedge, \ldots, A_\vee,$$

$$A^\wedge, \ldots, C \otimes D^\wedge, D^\wedge, \ldots, D_\vee, C^\wedge, \ldots, C_\vee, C \otimes D_\vee, \ldots, A_\vee,$$

$$A^\wedge, \ldots, D_\vee, C^\wedge, \ldots, C_\vee, C \otimes D_\vee, \ldots, C \otimes D^\wedge, D^\wedge, \ldots, A_\vee.$$

There is no other possibility since $A_\wedge, \ldots, C^\wedge, \ldots, A^\wedge$ and $A_\vee, \ldots, C \otimes D^\wedge, \ldots, A^\wedge$ are easily refuted by commuting our switch to "R". From this we get the result.

(vii): Consider a trip (with our ⅋-switch on "L"). Then, by Lemma 2.9.2, the order is as follows:

$$C^\wedge, \ldots, D_\wedge, D^\wedge, \ldots, C_\vee, C ⅋ D_\vee, \ldots, C ⅋ D^\wedge, C^\wedge$$

and we still have to determine the respective places of $A^\wedge$ and $A_\vee$ in this pattern. If $A^\wedge$ is between $C ⅋ D_\vee$ and $C ⅋ D^\wedge$, then the only way to ensure that $C \in eA$ is to have

$$A^\wedge, \ldots, C ⅋ D^\wedge, C^\wedge, \ldots, C_\vee, C ⅋ D_\vee, \ldots, A_\vee, \ldots,$$

i.e., $A_\vee$ is also in the same interval. If $A^\wedge$ is between $C^\wedge$ and $D_\vee$, then the only way to have $C \in eA$ is

$$A^\wedge, \ldots, D_\vee, \ldots, C_\vee, \ldots, C^\wedge, \ldots, A_\vee, \ldots,$$

i.e., $A_\vee$ is also in the same interval. Using $D \in eA$ we can conclude in the case where $A^\wedge$ is between $D^\wedge$ and $C_\vee$ that $A_\vee$ is also in the same interval. In these three cases, the two exits of $C ⅋ D$ are between $A^\wedge$ and $A_\vee$. □

2.9.5. Theorem (Trip Theorem). *There is a position of the switches such that eA is exactly the interval* $A^{\wedge}, \ldots, A_{\vee}$.

Proof. Each time there is a ⅋-link $\frac{C \quad D}{C ⅋ D}$ with exactly one premise in eA, set the corresponding switch to "R" if $C \in eA$ and to "L" if $D \in eA$. The other switches are set arbitrarily. Now we start our trip with $A^{\wedge}$, i.e., with an element of eA, and we check that we stay in eA up to the point $A_{\vee}$: this can be easily proved by induction since the only way to exit eA would be going from a premise to a conclusion in a ⅋-link. In that case, however, only one premise is in eA and the switches have been set such as to move backwards, i.e., to pass from $E_{\vee}$ to $E^{\wedge}$ and thus to stay in eA. So the whole portion betwen $A^{\wedge}$ and $A_{\vee}$ is in eA and since it contains eA, we are done. □

2.9.6. Corollary. *If* $C \otimes D$ *is 'above'* A, *i.e., is an hereditary premise of* A, *and if* $C \otimes D$ *is the conclusion of a* $\otimes$*-link* $\frac{C \quad D}{C \otimes D}$, *then* $eC \cup eD \subset eA$.

Proof. Set the switches so that eA is performed between $A^{\wedge}$ and $A_{\vee}$; since all formulas 'above' A belong to eA, we are free to set the switches so that, after $A^{\wedge}$, the particle goes from conclusion to premise up to $D^{\wedge}$. Then the trip can be visualized as

$$\ldots, A^{\wedge}, \ldots, D^{\wedge}, \ldots, D_{\vee}, \ldots, A_{\vee}, \ldots.$$

From this, $eD \subset D^{\wedge}, \ldots, D_{\vee} \subset A^{\wedge}, \ldots, A_{\vee} = eA$. For similar reasons, $eC \subset eA$. □

2.9.7. Theorem (Splitting Theorem). *Let* β *be a proof-net with at least one terminal* $\otimes$*-link and no terminal* ⅋*-link. Then it is possible to find a terminal* $\otimes$*-link* $\frac{A \quad B}{A \otimes B}$ *such that* β *is the union of* eA, eB, *and of the formula* $A \otimes B$.

Proof. Choose a link $\frac{A \quad B}{A \otimes B}$ such that $eA \cup eB$ is maximal w.r.t. inclusion. If $A \otimes B$ is not terminal, then below $A \otimes B$ there is a terminal link and, by hypothesis, this link must be of the form $\frac{C \quad D}{C \otimes D}$. (Say that, for instance, $A \otimes B$ is above D.) Then, by Corollary 2.9.6, $eA \cup eB \subset eD$, contradicting the assumption of maximality. Now, assume, for a contradiction, that $eA \cup eB \cup \{A \otimes B\} \neq B$. Then we claim the existence of a link $\frac{C \quad D}{C ⅋ D}$ with either $C \in eA$ and $D \in eB$, or $C \in eB$ and $D \in eA$. This link exists because, otherwise, by setting the switches as we did in the proof of the Trip Theorem 2.9.5, we can simultaneously realize $eA = A^{\wedge}, \ldots, A_{\vee}$ and $eB = B^{\wedge}, \ldots, B_{\vee}$. In the remaining two steps, the particle is in $A \otimes B$, so $\beta = eA \cup eB \cup \{A \otimes B\}$: a contradiction.

Now the formula $C ⅋ D$ is above a terminal $\otimes$-link $\frac{E \quad F}{E \otimes F}$ and we may assume that $C ⅋ D$ is indeed above, say F. Set the switches for a trip such that $F^{\wedge}, \ldots, F_{\vee} = eF$. In Corollary 2.9.6 it has already been observed that we are free to set the switches as we want above F, and we do it so that, immediately after $F^{\wedge}$, the particle runs up to $C^{\wedge}$: in particular, our ⅋-switch is on "L". Now the trip looks as follows (using

Lemma 2.9.2)

$$\ldots, F^{\wedge}, \ldots, C ⅋ D^{\wedge}, C^{\wedge}, \ldots, D_{\vee}, D^{\wedge}, \ldots, C_{\vee}, C ⅋ D_{\vee}, \ldots, F_{\vee}, \ldots.$$

Using the fact that $D \in eB$ and $C \in eA$, the only room for $B^{\wedge}$ in this picture is between $C^{\wedge}$ and $D_{\vee}$; the only possible room for $B_{\vee}$ in this picture is between $D^{\wedge}$ and $C_{\vee}$. But this shows that $eB \subset eF$. By interchanging C and D, we obtain that $eA \subset eF$. But then $eA \cup eB$ cannot be maximal any longer. We have therefore contradicted the assumption $eA \cup eB \cup \{A \otimes B\} \neq \beta$. □

Proof of Theorem 2.9 (*continued*). The remaining case is just the case where Theorem 2.9.7 applies, so choose a terminal link $\frac{A \quad B}{A \otimes B}$ with the splitting property. We claim that, once the terminal link has been removed, the subsets eA and eB are not linked together. The only possible linkage would be through a ⅋-link $\frac{C \quad D}{C ⅋ D}$ with one premise in eA and the other in eB, but then $C ⅋ D$ would be nowhere. So it makes sense to speak of the respective restrictions β' and β'' of β to eA and eB respectively. The fact that β' and β'' are in turn proof-nets is a trifle compared with the complexity of the proof of the Splitting Theorem and is therefore omitted! If we associate, using the induction hypothesis, with β' and β'' proofs π' and π'' in linear sequent calculus of $\vdash A, C$ and $\vdash B, D$ respectively such that $\pi'^{-} = \beta'$ and $\pi''^{-} = \beta''$, then we can define π as

$$\frac{\begin{matrix}\pi' \\ \vdash A, C\end{matrix} \qquad \begin{matrix}\pi'' \\ \vdash B, D\end{matrix}}{\vdash A \otimes B, C, D}\otimes$$

and, clearly, $\pi^{-} = \beta$. □

2.2. *The cut-rule*

The cut-rule has been omitted because, *geometrically speaking*, it is very close to the ⊗-rule. The cut-rule can be written as a link

$$\underline{A \qquad A^{\perp}}$$

with two premises and no conclusion. Trips through this link proceed in the obvious way: after $A_{\vee}$ go to $A^{\perp\wedge}$, after $A^{\perp}_{\vee}$ go to $A^{\wedge}$. Now this is (up to inessential details) as if one were working with a link

$$\frac{A \qquad A^{\perp}}{A \otimes A^{\perp}}$$

with $A \otimes A^{\perp}$ terminal (the position of the corresponding switch does not matter). We decide to write the cut-rule as

$$\frac{A \qquad A^{\perp}}{\text{CUT}},$$

where CUT is a symbol (not a formula) which is used so that the cut-rule looks formally like a ⊗-link, the only difference being that since CUT is not a formula, CUT is necessarily terminal. Then this link is treated exactly as a ⊗-link and our main theorem immediately extends to the calculus with cut. In particular, the phenomenon of splitting (Theorem 2.9.7) involves either a ⊗-link of a CUT-link.

2.3. *Cut-elimination*

Take β, a proof-net in the multiplicative fragment with cut and select a particular CUT-link $\frac{A \quad A^\perp}{\mathrm{CUT}}$. We define a *contractum* β' of β (w.r.t. this particular link) as follows:

(i) If A and $A^\perp$ are the respective conclusions of dual multiplicative links, i.e., if β ends with

$$\dfrac{\dfrac{\begin{matrix}\vdots & \vdots \\ B & C\end{matrix}}{B\,m\,C} \qquad \dfrac{\begin{matrix}\vdots & \vdots \\ B^\perp & C^\perp\end{matrix}}{B^\perp\, m^\perp\, C^\perp}}{\mathrm{CUT}}$$

with m, $m^\perp$ dual multiplicatives, then replace this part of β by

$$\dfrac{\begin{matrix}\vdots & \vdots \\ B & B^\perp\end{matrix}}{\mathrm{CUT}} \qquad \dfrac{\begin{matrix}\vdots & \vdots \\ C & C^\perp\end{matrix}}{\mathrm{CUT}}.$$

(ii) If A is the conclusion of an axiom-link, i.e., β ends with

$$\begin{matrix} \overline{A^\perp \qquad A} & \hspace{-1em}\begin{matrix}\vdots \\ A^\perp\end{matrix} \\ \vdots & \hspace{-3em}\overline{\mathrm{CUT}} \end{matrix}$$

unify the two occurrences of $A^\perp$ so that one gets

$$\begin{matrix}\vdots \\ A^\perp. \\ \vdots\end{matrix}$$

(iii) If $A^\perp$ is the conclusion of an axiom-link, we define β' symmetrically. When both A and $A^\perp$ are conclusions of axiom-links, i.e., in the situation

$$\begin{matrix} \overline{A^\perp \qquad A} & \overline{A^\perp \qquad A} \\ \vdots \qquad\qquad & \hspace{-4em}\overline{\mathrm{CUT}} \qquad\quad \vdots\,, \end{matrix}$$

then we have a conflict between (ii) and (iii). However, both cases lead to

$$\begin{matrix} \overline{A^\perp \qquad A} \\ \vdots \qquad\quad \vdots\,. \end{matrix}$$

2.10. Proposition. (1) *If β is a proof-net and β' is a contractum of β, then β' is a proof-net.*

(2) *β' is strictly smaller than β, i.e., has strictly less formulas.*

Proof. We prove (2) first, then (1).

(2): The *size* $s(\beta)$ of a proof-net is the number of formulas of β including the symbol CUT. The inclusion of these cut-symbols is the most elegant solution. In case (i) the size is diminished by 1; in cases (ii) and (iii) it is diminished by three units. There is however a hidden trap in this apparently trivial proof: when A is the conclusion of an axiom-link, who tells us that the two occurrences of $A^\perp$ are distinct ? In fact, they can be the same if β is the proof-structure

$$\frac{\overline{A \qquad A^\perp}}{\text{CUT}}$$

in which case β' is β. But this particular proof-structure is obviously not a proof-net since it has a shorttrip. This shows the extreme importance of being a proof-net.

(1): Assume that we are in case (i) (cases (ii) and (iii) are immediate) and that the multiplicative m is $\otimes$. Set the switches of β' for a trip. This induces a switching of β (our two multiplicative links being switched on "L"). Then the trip works as follows (cf. Lemmas 2.9.1 and 2.9.2):

$$B_\vee, B\otimes C_\vee, B^\perp ⅋ C^{\perp\wedge}, B^{\perp\wedge}, \ldots, C^\perp_\vee, C^{\perp\wedge}, \ldots, B^\perp_\vee, B^\perp ⅋ C^\perp_\vee,$$
$$B\otimes C^\wedge, C^\wedge, \ldots, C_\vee, B^\wedge, \ldots, B_\vee.$$

Hence, in β', the particle travels as follows:

$$B_\vee, B^{\perp\wedge}, \ldots, C^\perp_\vee, C^\wedge, \ldots, C_\vee, C^{\perp\wedge}, \ldots, B^\perp_\vee, B^\wedge, \ldots, B_\vee$$

which is a longtrip. □

2.11. Proposition. *Say that β reduces to β' (notation: β red β') when β' is a hereditary contractum of β. Then, reduction satisfies the Church-Rosser property.*

Proof. Essentially, this proposition says that if we contract two distinct cuts of β, then the result does not depend on the order of application of the contractions. This is practically immediate. □

2.12. Theorem (Strong Normalization). *A proof-net of size n normalizes into a cut-free proof-net in less than n steps; the result, which does not depend on the order of application of the contractions, is called the* normal form *of our proof-net.*

2.13. Remark. The '*n* steps' of the theorem suggest a sequential procedure for the normalization process. It is, of course, more natural to think of a parallel procedure, namely to work independently on each CUT. One of the astounding properties of the multiplicative fragment is that everything works perfectly well, i.e., that there is not the slightest problem of synchronization.

The problem is now to extend the theory of proof-nets to the full language. At the present moment (September 1986) there is no extant extension of the same quality as what exists for the multiplicative fragment. Moreover, we doubt that anything really convincing might ever be done for the additives. For the rest of the language, what we present below can surely be improved....

The general pattern we shall now introduce is that of a *proof-box*. Roughly speaking, proof-boxes are synchronization marks in the proof-net. They can also be seen as moments where we restore the sequent (i.e., the sequential!) structure. Their use is therefore a bridle to parallelism and, for that reason, one must try to limit their use. In particular, reasonable improvements of the concept of proof-net showing that some rules can be written without boxes will be of great practical interest (however, we believe that (&) cannot be written without boxes).

How to *make* a box depends on the particular syntax, the particular rules we want to express, and is of no interest right now. What concerns us now is how to *use* a box. When made, a box looks as shown in Fig. 11, i.e., a black 'thing' with

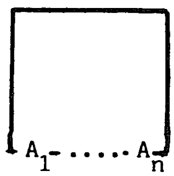


Fig. 11.

n outputs/inputs, $n \neq 0$. What is inside the square is the building process of the box, which is irrelevant: boxes are treated in a perfectly modular way: we can use the box $\mathbb{B}$ in β without knowing its contents, i.e., another box $\mathbb{B}'$ with exactly the same *n doors* $A_1, \ldots, A_n$ would do as well. This is the principle of the *black box*: in order to check the correctness of a proof-net involving a box $\mathbb{B}$, we have to check the proof-net without knowing anything about $\mathbb{B}$ and check $\mathbb{B}$ itself. Now, what is the criterion for the correctness of a proof-net involving a box $\mathbb{B}$? Simply, the box is treated as any potential proof-net with conclusions $A_1, \ldots, A_n$.

2.14. Definition. The *multiplicative fragment with proof-boxes* admits, besides the links already acknowledged, a new kind of link shown in Fig. 12, where $A_1, \ldots, A_n$ are arbitrary formulas and $n \neq 0$. With such a box is associated a switch: a position of the switch consists in a permutation of n which is cyclic, i.e., $1, \sigma(1), \sigma^2(1), \ldots, \sigma^{n-1}(1)$ are all distinct. The trip is executed as follows: after $A_i^\wedge$ go to $A_{\sigma(i)^\vee}$.

2.15. Theorem. *Theorems* 2.7 *and* 2.9 *can be extended to the calculus with boxes*; *the*

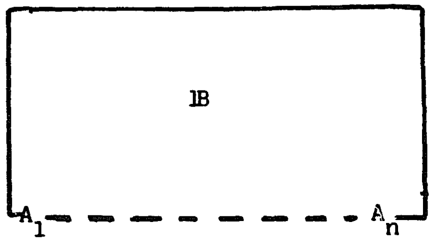


Fig. 12.

use of a proof-box with doors $A_1, \ldots, A_n$ in a proof-net corresponds to the use of the axiom $\vdash A_1, \ldots, A_n$.

Proof. The extension of the results to this new context offers no difficulty. In fact, proof-boxes have more or less the same structure as the logical axiom $\overline{A \quad A^{\perp}}$. For instance, let us look at the analogue of Fact 2.9.4(iii): if one door A_k of $\mathbb{B}$ belongs to eA, then any other door A_l of $\mathbb{B}$ does so too. (*Proof*: First observe that a trip is performed by putting together slices $A_{i\vee}, \ldots, A_i^{\wedge}$ because if outside the box one could have something like $A_{i\vee}, \ldots, A_j^{\wedge}$ with $j \neq i$, then, by appropriately choosing σ ($\sigma(j) = i$), one would get the shorttrip $A_{i\vee}, \ldots, A_j^{\wedge}, A_{i\vee}$. Now, if $A^{\wedge}$ and $A_{\vee}$ are located in $A_{i\vee}, \ldots, A_i^{\wedge}$ and $A_{j\vee}, \ldots, A_j^{\wedge}$ respectively, we claim that $i = j$; suppose otherwise, then take $\sigma(i) = j$: the two doors of A_k are not in $A_{i\vee}, \ldots, A_j^{\wedge}$. Finally, the only possible location of $A^{\wedge}$ and $A_{\vee}$ within $A_{i\vee}, \ldots, A_i^{\wedge}$ is $A_{i\vee}, \ldots, A_{\vee}, \ldots, A^{\wedge}, \ldots, A_i^{\wedge}$ and from this, we easily conclude that all the doors of $\mathbb{B}$ are in eA.) Apart from this analogue, there is very little novelty in this generalization. □

2.4. The system PN1

The system PN1 is a proof-net system for linear propositional calculus based on the idea of putting all contextual rules (i.e., rules which in sequent calculus depend on the context in sequent calculus) into proof-boxes without trying to limit the number of boxes. Besides boxes and the multiplicative links, there will be a certain number of links corresponding to unary rules

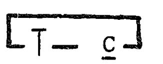
$$\frac{A}{B};$$

such links are so harmless (trip: from $A_{\vee}$ go to $B_{\vee}$, from $B^{\wedge}$ go to $A^{\wedge}$) that they practically do not alter the multiplicative paradise.

Axioms:

$\overline{A \quad A^{\perp}}$ logical axiom,

$\mathbf{1}$ axiom for $\mathbf{1}$,



axiom for $\top$.

Technically speaking, these three axioms are boxes, in which we distinguish *front doors* (A and $A^{\perp}$ in the logical axiom, $\mathbf{1}$ in the axiom for $\mathbf{1}$, $\top$ in the last case), and *auxiliary doors* (C in the last case).

Unary links:

$$\frac{A}{A\oplus B}1\oplus,\qquad \frac{B}{A\oplus B}2\oplus,\qquad \frac{A}{?A}D?.$$

Binary links:

$$\frac{A\quad B}{A\otimes B}\otimes,\qquad \frac{A\quad B}{A ⅋ B}⅋,$$

$$\frac{?A\quad ?A}{?A}C?,\qquad \frac{A\quad A^{\perp}}{\text{CUT}}.$$

Box formation: From proof-nets β' and β'' whose respective conclusions (CUT is not counted) are A, C and B, C, we form a box whose conclusions are $A \& B, C$ (see Fig. 13: front door: $A \& B$; auxiliary door: C).

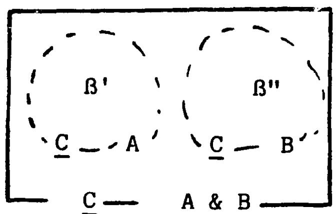


Fig. 13.

From a proof-net β with conclusions C (CUT is not counted), we form a box with conclusion $C, \perp$ (respectively $C, ?A$) (see Fig. 14: front door: $\perp$ (respectively $?A$); auxiliary doors: C).

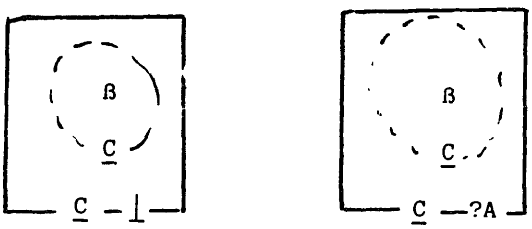


Fig. 14.

From a proof-net β whose conclusions (CUT is not counted) are $A, ?B$, we form a box whose conclusions are $!A, ?B$ (see Fig. 15: front door: $!A$, auxiliary doors: $?B$).

We recall that, geometrically speaking, the rule (CUT) behaves like ($\otimes$)[4] and that rule ($C?$) behaves like (⅋). It is not difficult to translate propositional linear

[4] It is convenient to consider the CUT link symmetric, i.e., there is no 'left' or 'right' in the rule as in the logical axiom.

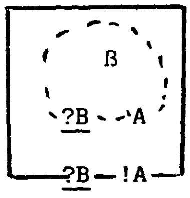


Fig. 15.

logic within this system, the only remark to make being that the rule ($W?$) is now done with the logical axioms. The fact that this translation is faithful offers no difficulty.

2.5. *The system PN2*

We finally direct our attention to the predicate case. Here, however, we make a different syntactic choice, moving from first-order to second-order. It was natural to deal with first-order linear logic to get a completeness theorem as obtained in Section 1. Already at that moment we did not spend too much time with the quantifiers! As the paper continues it focuses more and more on computational aspects, so it is more and more interesting to consider second-order logic in the spirit of the system F of [1]. So we shall develop here the system PN2 of second-order propositional logic. First-order quantifiers will from now on disappear from our world. The reader who would like to consider second-order predicate calculus would not encounter the slightest problem, just as the results on F were immediately transferable to Takeuti's system. Second-order propositional logic (linear version) is obtained by considering propositional variables $a, b, c, \ldots$ (instead of arbitrary constants) together with the quantifiers $\bigwedge a.$ and $\bigvee a.$. The crucial concept is that of substitution of a proposition B for a propositional variable a in A, denoted $A[B/a]$: Replace all occurrences of a by B, all occurrences of $a^{\perp}$ by $B^{\perp}$. In terms of sequent calculus, the rules for second-order quantification can be written as

$$\frac{\vdash A, B}{\vdash \bigwedge a.A, b}\bigwedge, \qquad \frac{\vdash A[B/a], C}{\vdash \bigvee a.A, C}\bigvee.$$

In ($\bigwedge$), there is the familiar restriction: a not free in B.
The rules for quantifiers are handled as follows:
- on the one hand, the unary link

$$\frac{A[B/a]}{\bigvee a.A}\bigvee;$$

- on the other hand, the box-formation scheme: from a proof-net β whose conclusions B, A are such that a does not occur free in B, we form a box whose outputs are B and $\bigwedge a.A$ (see Fig. 16).

2.16. Definition (*terminology*). PN1 and PN2 have already been introduced; PN0 is the subsystem of PN1 concerned with the multiplicative fragment only including

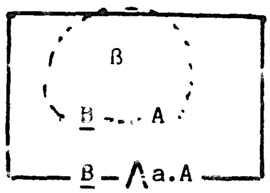


Fig. 16.

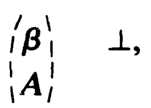
cut. PN0$^-$ is the cut-free part of PN0; similarly, PN1$^-$ and PN2$^-$ are the cut-free parts of PN1 and PN2.

2.6. *Discussion*

Here we shall shortly discuss the possibility of improving the syntax by removing or tranforming some boxes.

(i) The box for ! is an absolute one; later on, we shall write no contraction rules making this scheme commute with others. Hence, the box, viewed as an interruption of the linear features, should not be touched. Even if, by some sort of miracle, it were possible to erase such boxes, this would mean that one can reconstruct them in a unique way, so why not directly note them?

(ii) The boxes for weakening seem of very limited interest. The problem is that if we admit configurations like



then we are forced to accept certain shorttrips from which endless complications start. However, the only kind of shorttrip that is definitely bad is the one including only one gate for a formula, e.g. $A^\wedge$ but not $A_\vee$, and the inclusion of unboxed weakening would not introduce these bad boxes. A nice criterion for soundness for this variant is however not yet known.

(iii) The box for $\bigwedge$ seems removable too; the same comments hold as in the case of the weakening boxes.

(iv) Finally, the case of &-boxes is the most delicate; when we contract such boxes, there is a phenomenon of duplication, and removing the boxes would mean that we have a way to write the two boxes in Fig. 17 as the same proof-net. This goal seems out of reach. However, see Section 6 for what could be a solution by means of families of proof-structures.

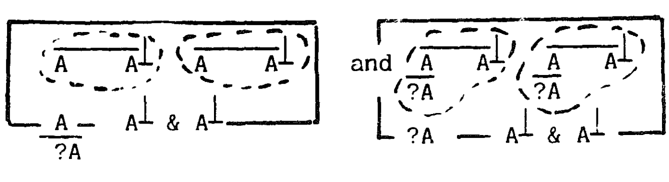


Fig. 17.

3. The coherent semantics

In Section 1, we developed the semantics of linear logic in the Tarskian style, i.e., by explaining the formulas. But there is another semantic tradition, amounting by Heyting's semantics of proofs, which consists in modelling the proofs themselves. In this tradition, proofs are considered as functions, relations, etc. on some kind of constructive space, which has often been viewed as Scott domains. In previous work, we have simplified Scott domains in order to get what we called *qualitative domains*. The *binary* ones, rebaptized *coherent spaces*, will form the core of our semantics of proofs.

3.1. Definition. A *coherent space* X is a set satisfying the following conditions:

(i) $a \in X \wedge b \subset a \Rightarrow b \in X$.

(ii) Say that $a, b \in X$ are *compatible* w.r.t. X when $a \cup b \in X$; then, if A is any subset of X formed of pairwise compatible elements, we have $\bigcup A \in X$. In particular, $\emptyset \in X$.

3.2. Definition. Assume that X is a coherent space. The *web* of X, $W(X)$ is a reflexive, unoriented graph, defined by

(i) the domain $|X|$ of the graph is $\bigcup X = \{z; \{z\} \in X\}$;

(ii) the linkage relation (*coherence* modulo X) is defined as follows:

$$x \frown\!\!\!\!\smile y\ [\text{mod}\ X] \ \text{ iff } \ \{x, y\} \in X.$$

3.3. Proposition. *The map associating to any coherent space X its web $W(X)$ is a bijection between coherent spaces and reflexive unoriented graphs. The graph may be recovered from its web by means of the formula*

$$a \in X \ \leftrightarrow \ a \subset |X| \wedge \forall x, y \in a\colon x \frown\!\!\!\!\smile y\ [\text{mod}\ X].$$

Proof. More or less immediate. □

3.4. Remark. In general, the webs stay denumerable and effective, whereas the spaces have the power of the continuum. So, most of the time we deal with webs. Instead of coherence, it is sometimes more convenient to consider *strict coherence*:

$$x \frown y\ [\text{mod}\ X] \ \text{ iff } \ x \frown\!\!\!\!\smile y\ [\text{mod}\ X] \text{ and } x \neq y.$$

The following relations are also of interest:

- *strict incoherence*: $x \smile y\ [\text{mod}\ X]$ iff $\neg(x \frown\!\!\!\!\smile y\ [\text{mod}\ X])$;
- *incoherence*: $x \asymp y\ [\text{mod}\ X]$ iff $\neg(x \frown y\ [\text{mod}\ X])$.

We shall now proceed with the definition of the interpretation of the operations of linear logic in terms of coherent spaces. A large attention will be paid to *canonical isomorphisms* (indicated by $\sim$). It is however necessary to attract the reader's attention to the two following points:

(i) $\sim$ is not a connective of linear logic; in particular, it is strictly stronger than $\multimapboth$. In fact, it would be easy to define a coherent semantics for something like $\sim$, but we would get problems in other stages, i.e., the phase semantics and the proof-nets.

(ii) Some canonical isomorphisms are omitted: this is because coherent spaces do not catch all of linear logic: typically, $\mathbf{1}$ and $\bot$ are the same coherent space for stupid reasons, but their equivalence (translated as $\mathbf{1} \multimapboth \bot$) is not provable. So, writing the isomorphisms would have been misleading.

3.5. Definition (*linear negation*). If X is a coherent space, its linear negation $X^\perp$ is defined by

$$|X^\perp| = |X|,$$

$$x \mathrel{\bigcirc} y\ [\text{mod } X^\perp] \text{ iff } x \asymp y\ [\text{mod } X].$$

In other terms, the operation *nil* exchanges coherence and incoherence. We clearly have $X^{\perp\perp} = X$, i.e., *nil* is involutive. The existence of a constructive involution is a tremendous improvement on intuitionistic logic. As we shall proceed with the other connectives, it will be our first task to check the De Morgan equalities, which will justify, at the level of the coherent semantics, the restriction of *nil* to atoms.

3.6. Definition (*multiplicatives*). The name itself comes from the coherent semantics: the three multiplicatives are variations on the theme of the cartesian product:

$$|X \otimes Y| = |X \parr Y| = |X \multimap Y| = |X| \times |Y|,$$

$$(x, y) \mathrel{\bigcirc} (x', y')\ [\text{mod } X \otimes Y] \text{ iff } x \mathrel{\bigcirc} x'\ [\text{mod } X] \text{ and } y \mathrel{\bigcirc} y'\ [\text{mod } Y],$$

$$(x, y) \frown (x', y')\ [\text{mod } X \parr Y] \text{ iff } x \frown x'\ [\text{mod } X] \text{ or } y \frown y'\ (\text{mod } Y],$$

$$(x, y) \mathrel{\bigcirc} (x', y')\ [\text{mod } X \multimap Y] \text{ iff } (x \frown x'\ [\text{mod } X] \Rightarrow y \frown y'\ [\text{mod } Y]) \text{ and }$$
$$(x \mathrel{\bigcirc} x'\ [\text{mod } X] \Rightarrow y \mathrel{\bigcirc} y'\ [\text{mod } Y]).$$

Observe that $\otimes$ is defined in terms of $\bigcirc$, whereas $\parr$ is defined in terms of $\frown$, and $\multimap$ is defined in terms of preservation properties, which is in the spirit of an implication, linear or not.

The De Morgan equalities are easily verified:

$$(X \otimes Y)^\perp = X^\perp \parr Y^\perp, \qquad (X \parr Y)^\perp = X^\perp \otimes Y^\perp, \qquad X \multimap Y = X^\perp \parr Y \quad \text{etc.}$$

The *commutativity* of the multiplicatives is expressed by

$$X \otimes Y \sim Y \otimes X, \qquad X \parr Y \sim Y \parr X, \qquad X \multimap Y \sim Y^\perp \multimap X^\perp.$$

The *associativity* of the multiplicatives is expressed by

$$X \otimes (Y \otimes Z) \sim (X \otimes Y) \otimes Z, \qquad X \parr (Y \parr Z) \sim (X \parr Y) \parr Z,$$

$$X \multimap (Y \multimap Z) \sim (X \otimes Y) \multimap Z, \qquad X \multimap (Y \parr Z) \sim (X \multimap Y) \parr Z.$$

3.7. Definition (*the unit coherent space*). Up to isomorphism, there is exactly one coherent space whose web consists of one point, say 1. In particular, this space is equal to its linear negation. The space will be indifferently denoted **1** or $\bot$ since it will stand as the interpretation of these constants. We write the isomorphisms expressing the *neutrality* of the unit space, choosing between the two notations **1** and $\bot$ the one which has also a logical meaning:

$$X \otimes \mathbf{1} \sim X, \qquad X \parr \bot \sim X, \qquad \mathbf{1} \multimap X \sim X, \qquad X \multimap \bot \sim X^{\bot}.$$

3.8. Definition (*the additives*). Here too the name comes from the coherent semantics: the two additives are variations on the theme of the direct sum:

$$|X \,\&\, Y| = |X \oplus Y| = |X| + |Y| = \{0\} \times |X| \cup \{1\} \times |Y|,$$

$$(0, x) \coh (0, x') \,[\text{mod } X \,\&\, Y] \text{ or } [\text{mod } X \oplus Y] \text{ iff } x \coh x' \,[\text{mod } X],$$

$$(1, y) \coh (1, y') \,[\text{mod } X \,\&\, Y] \text{ or } [\text{mod } X \oplus Y] \text{ iff } y \coh y' \,[\text{mod } Y],$$

$$(0, x) \frown (1, y) \,[\text{mod } X \,\&\, Y] \quad \text{for all } x \in |X| \text{ and } y \in |Y|,$$

$$(0, x) \smile (1, y) \,[\text{mod } X \oplus Y] \quad \text{for all } x \in |X| \text{ and } y \in |Y|.$$

X and Y have been recopied as if they were in the direct sum; now there are only two good-taste solutions as to coherence between X and Y: either always "yes" (&) or always "no" ($\oplus$).

The De Morgan equalities are immediate:

$$(X \,\&\, Y)^{\bot} = X^{\bot} \oplus Y^{\bot}, \qquad (X \oplus Y)^{\bot} = X^{\bot} \,\&\, Y^{\bot}.$$

The *commutativity* of the additives is expressed by

$$X \,\&\, Y \sim Y \,\&\, X, \qquad X \oplus Y \sim Y \oplus X.$$

The *associativity* of the additives is expressed by

$$X \,\&\, (Y \,\&\, Z) \sim (X \,\&\, Y) \,\&\, Z, \qquad X \oplus (Y \oplus Z) \sim (X \oplus Y) \oplus Z.$$

The *distributivity* of the multiplicatives w.r.t. the additives is expressed by

$$X \otimes (Y \oplus Z) \sim (X \otimes Y) \oplus (X \otimes Z),$$

$$X \parr (Y \,\&\, Z) \sim (X \,\&\, Y) \parr (X \,\&\, Z),$$

$$X \multimap (Y \,\&\, Z) \sim (X \multimap Y) \,\&\, (X \multimap Z),$$

$$(X \oplus Y) \multimap Z \sim (X \multimap Y) \,\&\, (X \multimap Z).$$

The *semi-distributivity* of the multiplicatives w.r.t. the additives is the fact that

between $X \otimes (Y \,\&\, Z)$ and $(X \otimes Y) \,\&\, (X \otimes Z)$,

between $(X \parr Y) \oplus (X \parr Z)$ and $X \parr (Y \oplus Z)$,

between $(X \multimap Y) \oplus (X \multimap Z)$ and $X \multimap (Y \oplus Z)$,

between $(X \multimap Z) \oplus (Y \multimap Z)$ and $(X \,\&\, Y) \multimap Z$

it is possible to construct a bijective mapping from the web of the left-hand space to the web of the right-hand space, preserving coherence. In general, these maps are not isomorphisms.

3.9. Definition (*the null coherent space*). This space has a void web. We shall denote it by **0** or $\top$, with the same hidden intentions as in the case of the unit coherent space: the *neutrality* of the void space w.r.t. additives is expressed by

$$X \oplus \mathbf{0} \sim X, \qquad X \,\&\, \top \sim X.$$

The fact that they *absorb* multiplicatives is expressed by

$$X \otimes \mathbf{0} \sim \mathbf{0}, \qquad X \,⅋\, \top \sim \top, \qquad \mathbf{0} \multimap X \sim \top, \qquad X \multimap \top \sim \top.$$

3.10. Definition (*the exponentials*).

$$|!X| = \{a\,; a \in X \text{ and } a \text{ finite}\}, \qquad a \frown\!\!\!\!\smile b \,[\bmod\ !X] \text{ iff } a \cup b \in X,$$

$$|?X| = \{a\,; a \in X^{\perp} \text{ and } a \text{ finite}\}, \qquad a \frown b \,[\bmod\ ?X] \text{ iff } a \cup b \notin X^{\perp}.$$

The De Morgan formulas are immediate:

$$(!X)^{\perp} = ?(X^{\perp}), \qquad (?X)^{\perp} = !(X^{\perp}).$$

The *distributivity* isomorphisms are

$$!(X \,\&\, Y) \sim (!X) \otimes (!Y), \qquad ?(X \oplus Y) \sim (?X) \,⅋\, (?Y).$$

The following can be taken as definitions of **1** and $\perp$:

$$!\top \sim \mathbf{1}, \qquad ?\mathbf{0} \sim \perp.$$

We now have to deal with quantifiers. First-order quantifiers have a boring coherent semantics; hence, we concentrate on second-order propositional quantifiers, which will be handled by means of the category-theoretic methods introduced in [5] which we quickly recall in the following items (i)-(v):

(i) Coherent spaces form a category COH by taking as morphisms from X to Y the set COH(X, Y) of all injective functions from $|X|$ to $|Y|$ such that

$$\forall x, y \in |X| \quad x \frown\!\!\!\!\smile y \,[\bmod X] \leftrightarrow f(x) \frown\!\!\!\!\smile f(y) \,[\bmod Y].$$

Associated with such a morphism are two (linear) maps:

- f^{+} from X to Y: $f^{+}(a) = \{f(z)\,; z \in a\}$,
- f^{-} from Y to X: $f^{-}(b) = \{z\,; f(z) \in b\}$.

(ii) Now, if $A[\alpha_1, \ldots, \alpha_n]$ is a formula of second-order propositional calculus, we can define an associated functor (denoted $A[X_1, \ldots, X_n]$, $A[f_1, \ldots, f_n]$) from COH^n to COH:

(1) when A is one of the constants **0**, **1**, $\top$, or $\perp$, then $A[X]$ is the coherent space already described in Definitions 3.7 or 3.9, whereas $A[f]$ is the identity morphism of this space;

(2) when A is the propositional variable α_i, then $A[X]=X_i$ and $A[f]=f_i$;
(3) when A is $\alpha_i^\perp$, then $A[X]=X_i^\perp$ and $A[f]=f_i$;
(4) when A is $B\otimes C$, then $A[X]=B[X]\otimes C[X]$ and $A[f](x,y)=(B[f](x), C[f](y))$;
(5) when A is $B ⅋ C$, then $A[X]=B[X] ⅋ C[X]$ and $A[f](x,y)=(B[f](x), C[f](y))$;
(6) when A is $B \& C$, then $A[X]=B[X] \& C[X]$ and

$$A[f](0,x)=(0, B[f](x)), \qquad A[f](1,y)=(1, C[f](y));$$

(7) when A is $B\oplus C$, then $A[X]=B[X]\oplus C[X]$ and

$$A[f](0,x)=(0, B[f](x)), \qquad A[f](1,y)=(1, C[f](y));$$

(8) when A is $!B$, then $A[X]=!B[X]$ and $A[f](a)=B[f]^+(a)$;
(9) when A is $?B$, then $A[X]=?B[X]$ and $A[f](a)=B[f]^+(a)$.

The definition of $A[X]$ when A is of the form $\bigwedge\beta.B$ or $\bigvee\beta.B$ will be given soon. For the definition of $A[f]$ in both cases, we have to define $A[f](X_0,z)=(X_0',z')$, where X_0' and z' are obtained as follows: Let $y=B[g,f](z)$ where g is the identity of X_0; then, w.r.t. the functor $A[.,Y]$ ($f\in \mathrm{COH}(X,Y)$), y has a normal form $y=A[h,\mathrm{id}_Y](z')$ for a suitable X_0', a suitable $h\in\mathrm{COH}(X_0,X_0')$ and a suitable z' such that (X_0',z') is in the trace of $A[.,Y]$.

(iii) The functors defined in that way preserve
- direct limits,
- pull-backs.

Unfortunately, preservation of kernels is lost when dealing with exponentials and this is one of the reasons for looking for variants of the exponentials.

(iv) For such functors, we have a *Normal Form Theorem* [5, Theorem 1.3]: let F be a functor from COH to COH preserving direct limits and pull-backs; if $z\in|F(X)|$, then it is possible to find a finite coherent space X_0 together with an $f\in\mathrm{COH}(X_0,X)$ and a $z_0\in|F(X_0)|$ such that

(1) $z\in F(f)(z_0)$;

(2) if Y, $g\in\mathrm{COH}(Y,X)$ and $z_1\in|F(Y)|$ are such that $z=F(g)(z_1)$, then there is a unique $h\in\mathrm{COH}(X_0,Y)$ such that $z=F(g)(z_1)$ and $f=gh$.

(v) A *trace* of F is a set T such that

(1) the elements of T are pairs (X,z), with X finite coherent space and $z\in|F(X)|$;

(2) given any coherent space Y and any $y\in|F(Y)|$, there is a *unique* $(X,z)\in\mathrm{Tr}(F)$ and a morphism $f\in\mathrm{COH}(X,Y)$ such that $y=F(f)(z)$. One uses the notation $\mathrm{Tr}(F)$ to denote an arbitrary trace of F, chosen once and for all; the actual choice of the trace does not matter.

3.11. Definition (*the quantifiers*). Assume that F is a functor from COH to COH preserving direct limits and pull-backs; then $\bigwedge F$ and $\bigvee F$ are defined as follows:

(1) $|\bigwedge F|$ consists of those (X,z) in $\mathrm{Tr}(F)$ such that, for all f_1,f_2 in $\mathrm{COH}(X,Y)$: $F(f_1)(z)\supset F(f_2)(z)$ [mod $F(Y)$]. Further, $(X,z)\supset(X',z')$ [mod $\bigwedge F$]

iff, for all Y, for all $f \in \mathrm{COH}(X, Y)$ and $f' \in \mathrm{COH}(X', Y)$: $F(f)(z) \mathrel{\overset{\frown}{\smile}} F(f')(z')$ [mod $F(Y)$].

(2) $|\bigvee F|$ consists of those (X, z) in $\mathrm{Tr}(F)$ such that, for all Y and all f_1, f_2 in $\mathrm{COH}(X, Y)$: $F(f_1)(z) \asymp F(f_2)(z)$ [mod $F(Y)$]. Further, $(X, z) \mathrel{\overset{\frown}{\smile}} (X', z')$ [mod $\bigvee F$] iff for some Y, for some $f \in \mathrm{COH}(X, Y)$ and $f' \in \mathrm{COH}(X', Y)$: $F(f)(z) \mathrel{\overset{\frown}{\smile}} F(f')(z)$ [mod $F(Y)$].

3.12. Remarks. (i) $\bigwedge \beta.A[\beta, X]$ and $\bigvee \beta.A[\beta, X]$ are defined as respectively, $\bigwedge F$ and $\bigvee F$, where F is the functor $A[., X]$.

(ii) The quantifiers involved in Definition 3.11 can be bounded. This was indeed shown in [5], and *binary* qualitative domains (now called coherent spaces) were introduced just to ensure that.

(iii) Computations made in the same paper show that

$$\bigwedge \alpha.\alpha = \mathbf{0}, \qquad \bigwedge \alpha.\alpha \multimap \alpha \sim \mathbf{1}.$$

(In fact, it is shown that $\bigwedge \alpha.\alpha \Rightarrow \alpha \sim \mathbf{1}$, from which we easily get $\bigwedge \alpha.\alpha \multimap \alpha \sim \mathbf{1}$.)

(iv) The interpretation of existence is a pure De-Morganization; in particular, $|\bigvee F|$ has an unexpected definition which has deep consequences, and is presumably far from being well understood.

3.13. Example (*some isomorphisms*). Among the many isomorphisms involving quantifiers, let us mention:

(1) the De Morgan equalities:

$$(\bigwedge \alpha.A)^{\perp} = \bigvee \alpha.A^{\perp}, \qquad (\bigvee \alpha.A)^{\perp} = \bigwedge \alpha.A^{\perp};$$

(2) the *distributivity* formulas:

$$\bigwedge \alpha.(A[\alpha] \parr B) \sim (\bigwedge \alpha.A[\alpha]) \parr B, \qquad \bigvee \alpha.(A[\alpha] \otimes B) \sim (\bigvee \alpha.A[\alpha]) \otimes B,$$

$$\bigwedge \alpha.(A[\alpha] \multimap B) \sim (\bigvee \alpha.A[\alpha]) \multimap B, \qquad \bigwedge \alpha.(A \multimap B[\alpha]) \sim A \multimap \bigwedge \alpha.B[\alpha];$$

(3) the *associativity* formulas:

$$\bigwedge \alpha.(A[\alpha] \mathbin{\&} B) \sim (\bigwedge \alpha.A) \mathbin{\&} B, \qquad \bigvee \alpha.(A[\alpha] \oplus B) \sim (\bigvee \alpha.A[\alpha]) \oplus B;$$

etc. To remember them, recall that $\bigwedge$ and $\bigvee$ are to some extent generalized additives, and that $\bigwedge$ looks like &, $\bigvee$ looks like $\oplus$.

Our next goal is to define the semantics of the proofs of second-order linear propositional logic. For this, it is simpler to stay with sequents in order to avoid combining the novelty of the semantics with the novelty of the syntax!

3.14. Remark (*general pattern*). Our goal is to interpret a proof π of a sequent $\vdash A$ in linear sequent calculus. For this, we shall make a certain number of conventions:

(1) We consider A to be made out of *closed* propositions; if not, substitute for the free propositions of A arbitrary coherent spaces. This means that we are in fact

working with additional constants, namely X for each coherent space X; but this is enough for these trifles.

(2) We shall identify a formula with its associated coherent space; a convention that saves boring extra symbols like *, etc.

(3) However, we use * for the interpretation of π; hence, π^* is an object of the coherent space that would interpret the *par* of the sequence A. In other terms, π^* will be made of sequences $z \in A$. By this we mean, in case A is $A_1, \ldots, A_n$ that z is of the form $z_1, \ldots, z_n$, and $z_1 \in A_1, \ldots, z_n \in A_n$. Strict coherence modulo A is defined by

$$z \frown z' \, [\mathrm{mod}\, A] \text{ iff } z_i \frown z'_i \, [\mathrm{mod}\, A_i] \quad \text{for some } i.$$

Of course, one has to check that the subsets π^* are made of pairwise coherent elements. (For readability, we add a symbol $\vdash$ in front of the sequences z.)

3.15. Definition. π^* is defined by induction on the proof π of $\vdash A$. We simultaneously check the coherence of π^* modulo A.

(i) If π is the axiom $\vdash B, B^\perp$, then $\pi^* = \{\vdash z, z; z \in |B|\}$. Observe that when $z \neq z'$, either $z \frown z' \, [\mathrm{mod}\, B]$ or $z \frown z' \, [\mathrm{mod}\, B^\perp]$.

(ii) If π is obtained from π' and π'' by the cut-rule

$$\frac{\vdash B, C \qquad \vdash B^\perp, D}{\vdash C, D},$$

then $\pi^* = \{\vdash z', z''; \exists y \in |B| \, (\vdash y, z' \in \pi'^* \text{ and } \vdash y, z'' \in \pi''^*)\}$. To show the compatibility of two different points $\vdash z', z''$ and $\vdash t', t''$ of π^*, assume that y, x are such that $\vdash y, z'$ and $\vdash x, t' \in \pi'^*$, and $\vdash y, z''$ and $\vdash x, t'' \in \pi''^*$. Then, either $y \asymp x \, [\mathrm{mod}\, B]$ or $y \asymp x \, [\mathrm{mod}\, B^\perp]$. In the first case, $\vdash z' \frown \vdash z'' \, [\mathrm{mod}\, C]$; in the other one, $\vdash t' \frown \vdash t'' \, [\mathrm{mod}\, D]$; and in both cases, $\vdash z', t' \frown \vdash z'', t'' \, [\mathrm{mod}\, C, D]$.

The most important semantic feature of the cut-rule is the existential quantifier in front of the interpretation. Observe that this quantifier has a very unexpected feature: the *unicity* of y. (*Proof*: Assume that $\vdash y, z'$ and $\vdash x, z' \in \pi'^*$, and that $\vdash y, z''$ and $\vdash x, z'' \in \pi''^*$; necessarily, $y \supset x \, [\mathrm{mod}\, B]$ and $y \supset x \, [\mathrm{mod}\, B^\perp]$ and necessarily, $y = x$.)

(iii) If π is obtained from π' by the exchange rule replacing A by a permutation $\sigma(A)$, then $\pi^* = \{\vdash \sigma(z); \vdash z \in \pi'^*\}$.

(iv) If π is the axiom $\vdash \top, A$, then $\pi^* = \emptyset$.

(v) If π is obtained from π' and π'' by (&):

$$\frac{\vdash A, C \qquad \vdash B, C}{\vdash A \,\&\, B, C},$$

then $\pi^* = \{\vdash (0, a), z'; \vdash a, z' \in \pi'^*\} \cup \{\vdash (1, b), z''; \vdash b, z'' \in \pi''^*\}$.

(vi) If π is obtained from π' by (1$\oplus$), then $\pi^* = \{\vdash (0, a), z'; \vdash a, z' \in \pi'^*\}$.

(vii) If π is obtained from π' by (2$\oplus$), then $\pi^* = \{\vdash (1, a), z'; \vdash a, z' \in \pi'^*\}$.

(viii) If π is the axiom $\vdash \mathbf{1}$, then π^* consists of one point, namely $\vdash 1$.

(ix) If π comes from π' by the weakening rule ($\perp$):

$$\frac{\vdash A}{\vdash \perp, A},$$

then $\pi^* = \{\vdash 1, z; \vdash z \in \pi'^*\}$.

(x) If π comes from π' and π'' by ($\otimes$):

$$\frac{\vdash A, C \qquad \vdash B, D}{\vdash A \otimes B, C, D},$$

then $\pi^* = \{\vdash (z', z''), t', t''; \vdash z', t' \in \pi'^* \text{ and } \vdash z'', t'' \in \pi''^*\}$. Assume that $\vdash (z', z''), t', t''$ and $\vdash (x', x''), w', w''$ are two distinct points (u and v) or π^* so that, for instance, $\vdash z', t' \neq \vdash x', w'$; then,

- either $t' \frown w'$ [mod C] in which case u and v are coherent,
- or $t' \asymp w'$ [mod C] in which case $z' \frown x'$ [mod A].

Now, if $z'' = x''$, we get the coherence of (z', z'') with (x', x''), from which u and v are coherent; hence, we can assume that $\vdash z'', t'' \neq \vdash x'', w''$, in which case we have

- either $t'' \frown w''$ [mod D] and u and v are coherent,
- or $t'' \asymp w''$ [mod D] and so, $z'' \frown x''$ [mod B]; but then, $(z', z'') \frown (x', x'')$ [mod $A \otimes B$] and once more u and v are coherent.

(xi) If π comes from π' by ($⅋$):

$$\frac{\vdash A, B, C}{\vdash A ⅋ B, C},$$

then $\pi^* = \{\vdash (x, y), z; \vdash x, y, z \in \pi'^*\}$. In this case, there is practically nothing to verify.

(xii) If π is obtained from π' by weakening rule ($W?$):

$$\frac{\vdash B}{\vdash ?A, B},$$

then $\pi^* = \{\vdash \emptyset, z; \vdash z \in \pi'^*\}$.

(xiii) If π is obtained from π' by the dereliction rule ($D?$):

$$\frac{\vdash A, B}{\vdash ?A, B},$$

then $\pi^* = \{\vdash \{a\}, z; \vdash a, z \in \pi'^*\}$.

(xiv) If π is obtained from π' by the contraction rule ($C?$):

$$\frac{\vdash ?A, ?A, B}{\vdash ?A, B},$$

then $\pi^* = \{\vdash a \cup b, z; \vdash a, b, z \in \pi'^* \text{ and } a \cup b \in A^\perp\}$. The coherence of π^* is practically immediate.

(xv) If π is obtained from π' by the rule (!):

$$\frac{\vdash A, ?B}{\vdash !A, ?B},$$

then π^* is the set of all sequences of the form $\vdash\{x_1, \dots, x_n\}, z_1 \cup \cdots \cup z_n$ such that

(1) $\{x_1, \dots, x_n\} \in A$,

(2) for $i = 1, \dots, n$, $\vdash x_i, z_i \in \pi'^*$,

(3) $z_1 \cup \cdots \cup z_n \in B^\perp$

(By the way, $x_1, \dots, x_n$ are necessarily distinct: if $x_1 = x_2$, then $z_1 \cup z_2 \in B^\perp$ and the fact that the $\vdash x_i, z_i$'s are coherent shows that $z_1 = z_2$, in which case we have a useless repetition.)

The coherence of two elements $u = \{x_1, \dots, x_n\}, z_1 \cup \cdots \cup z_n$ and $v = \{y_1, \dots, y_m\}, t_1 \cup \cdots \cup t_m$ is shown as follows: assume $u \neq v$;

- if $\{x_1, \dots, x_n, y_1, \dots, y_m\} \in A$, then u and v are coherent;
- otherwise, one x_i (say x_1) is incoherent with one y_j (say y_1). But $\vdash x_1, z_1$ and $\vdash y_1, t_1$ are coherent; hence, $z_1 \frown t_1$ [mod $?B$]; this shows that $z_1 \cup \cdots \cup z_n \frown t_1 \cup \cdots \cup t_m$ [mod $?B$] and u and v are coherent.

(xvi) If π is obtained from π' by the rule ($\bigwedge$):

$$\frac{\vdash A, B}{\vdash \bigwedge\alpha.A, B},$$

then $\pi^* = \{\vdash (X, a), z; (X, a) \in |\bigwedge\alpha.A| \text{ and } \vdash a, z \in \pi'[X/\alpha]^*$. Here we use the substitution of the new constant X for α.

The compatibility of distinct sequences $u = (X, a), z$ and $v = (Y, b), t$ is shown as follows: take Z together with $f \in \mathrm{COH}(X, Z)$, $g \in \mathrm{COH}(Y, Z)$; then, consider $a' = A[f/\alpha](a)$ and $b' = A[g/\alpha](b)$. Then, the general properties of the interpretation, as studied in [5], show that $\vdash a', z$ and $\vdash b', t$ both belong to $\pi[Z/\alpha]^*$, hence, are coherent. Moreover, they are still distinct by the general properties of a trace. If z is coherent with t, then u is coherent with v; otherwise, $a' \frown b'$ mod $A[Z/\alpha]$ and then, $(X, a) \frown (Y, b)$ [mod $\bigwedge\alpha.A$] from which we conclude again.

(xvii) If π is obtained from π' by the rule ($\bigvee$):

$$\frac{\vdash A[B/\alpha], C}{\vdash \bigvee\alpha.A, C},$$

then $\pi^* = \{\vdash (X, a), z; (X, a) \in |\bigvee\alpha.A| \text{ and } \exists f \in \mathrm{COH}(X, B): \vdash A[f/\alpha](a), z \in \pi'^*\}$. Assume that $(X, a), z$ $(= u)$ and $(Y, b), t$ $(= v)$ are two distinct elements of π^* with witnesses f, g in $\mathrm{COH}(X, B)$ and $\mathrm{COH}(Y, B)$. If $a' = A[f/\alpha](a)$, if $b' = A[f/\alpha](b)$, then $a', z \supset b', t$ mod $A[B/\alpha]$, C, and these two sequences are distinct. Then, either $z \frown t$ [mod C] or $a' \frown b'$ mod $A[B/\alpha]$, in which case $(X, a) \frown (Y, b)$ [mod $\bigvee\alpha.A$]; in both cases, u is coherent with v. This rule is the other case (after the cut-rule) of an existential quantifier. Here too, we have *unicity*: if the element $(X, a), z$ of π^* is witnessed by both $f, g \in \mathrm{COH}(X, A[B/\alpha])$, then, with $a' = A[f/\alpha](a)$, $a'' = A[g/\alpha](a)$, we get the coherence of a', z and a'', z. But, looking at the definition of $\bigvee\alpha.A$ we see that $a' \asymp a''$ mod $A[B/\alpha]$, so $a' = a''$. Hence, the element a' 'above' a is uniquely determined (unfortunately, f is not uniquely determined because of the nonpreservation of kernels; however, the range of f is uniquely determined).

3.16. Remarks. (i) The fact that the cut-rule formally behaves like the existential rule can be explained as follows: Define a formula CUT as $\bigvee \alpha.\alpha \otimes \alpha^{\perp}$; then, semantically speaking, CUT consists of a trivial unit coherent space with only one point, namely $(1,(1,1))$ $(= a_0)$. Now it is easy to show that the cut-rule can be mimicked by the formula CUT:

$$\cfrac{\cfrac{\vdash A, B \quad \vdash A^{\perp}, C}{\vdash A \otimes A^{\perp}, B, C}\otimes}{\vdash \text{CUT}, B, C}\bigvee .$$

(This is not far from some techniques used in [6].) Now, given premises π' and π'' of a cut-rule, let us compare the semantic interpretation of the proofs π and μ obtained by applying in one case the cut-rule, in the other case the procedure given above: $\mu^* = \{\vdash a_0, z; z \in \pi^*\}$, which means that there is no significant difference between both interpretations, and the unicity features of the cut-rule are therefore explainable from the similar features of the existential rule.

(ii) Assume, just for a minute, that we try to compute the *semantics* of a proof π, i.e., we try to decide, given $z \in |A|$, whether or not $z \in \pi^*$. In all cases but (ii) and (xvii), the problem is immediately reduced to several simpler problems of the same type, with an effective bound on these problems. But in Definition 3.15(ii) and (xvii), the problem is reduced to infinitely many simpler ones, due to the presence of an unbounded existential quantifier. But this existential part is unique, i.e., in some sense there is a well-defined z' 'above' z, summarizing the existential requirements on z. We can also think of z' as *implicitly* defined by z. In particular, cut-elimination looks as the elimination of implicit features, an elimination which can be achieved when the formula proved has no existential type.

This basic remark is the key to a semantic approach to computation, which will be undertaken somewhere else. Also observe that we are tempted to remove not only cuts, but also ($\bigvee$)-rules!

The just given semantics of sequential proofs induces a semantics of the corresponding proof-nets: one simply has to make the boring verification that, whenever $\pi^- = \mu^-$, we have $\pi^* = \mu^*$. Now, one could dream to work with proof-nets directly, without sequentializing. We shall illustrate this by a semantic interpretation of PN0, directly defined on the proof-nets. PN0 has been chosen because of its syntactic simplicity, and because in PN0 all the difficulties of the task are concentrated; from the sketch given here, there is no problem to move to PN2.

3.17. Definition (*experiments*). Let β be a proof-net in PN0. For each terminal formula A of β, we select an element a from the web of (the interpretation of) A. Now, by moving upwards, we construct, for each formula of the net, say B, an element $b \in |B|$. For the symmetry of the argument, it is convenient to consider the terminal symbol CUT as interpreted as a unit coherent space $\{a_0\}$.

(i) Assume that we have a binary ⊗- or ⅋-link $\frac{A \quad B}{C}$; we have already chosen $c \in |C|$; now, $|C| = |A| \otimes |B|$, i.e., $c = (a, b)$; then we choose a in $|A|$, b in $|B|$.

(ii) Assume that we have a cut-link $\frac{A \quad A^{\perp}}{\text{CUT}}$; in $|\text{CUT}|$, a_0 has been chosen by convention. We select an arbitrary element $a \in |A|$ (if $|A| \neq \emptyset$) and we put this element a both in $|A|$ and $|A^{\perp}|$; in case $|A| = \emptyset$, the process fails.

By moving upwards, we eventually obtain (provided we never fail) an assignment of points for each formula of the net and this assignment is called an *experiment*. In the experiment, some choices have been made which were not mechanical. Now, for each axiom link $\overline{A \quad A^{\perp}}$, we have selected points $a' \in |A|$, $a'' \in |A|$. The experiment *succeeds* when, for any such link, the two choices are the same, i.e., $a' = a''$.

A sequence of points in the interpretations of the conclusion of β belongs to β^* exactly when there is a successful experiment starting with those points.

3.18. Compatibility Theorem. *Assume that we have made two experiments in β (corresponding to two terminal sequences $\mathbf{a}'$, $\mathbf{a}''$) and that these experiments succeed; then $\mathbf{a}'$ and $\mathbf{a}''$ are coherent (in the sense of the* par *of the conclusions).*

Proof. Immediate consequence of the following lemma.

3.18.1. Compatibility Lemma. *Assume that $\mathbf{b}'$ and $\mathbf{b}''$ are two experiments in β, and that, for any conclusion A of β, the corresponding points a', a'' are incoherent: $a' \asymp a''$* [mod A]; *then the two experiments coincide. (In particular, there is exactly one successful experiment corresponding to a given sequence of β^*.)*

Proof. For each formula A of B, we shall write $A : \supset\!\!\subset$ (respectively, $\frown$, $\asymp$, $\smile$, $=$, $\neq$) to indicate the corresponding relation between the two inhabitants of A that have been chosen in both experiments. Now, the proof proceeds as follows.

(i) In the case of a ⊗-link $\frac{A \quad B}{A \otimes B}$, we shall distinguish between two cases:

Case 1: $A \otimes B : \supset\!\!\subset$ or $A : \smile$;

Case 2: $A \otimes B : \supset\!\!\subset$ or $B : \smile$.

Clearly, one of these cases holds.

(ii) In the case of a ⅋-link $\frac{A \quad B}{A ⅋ B}$, we shall distinguish between two cases:

Case 1: $A ⅋ B : =$ or $A : \neq$;

Case 2: $A ⅋ B : =$ or $B : \neq$.

Here too, one of these cases holds.

(iii) We shall now prove the impossibility of being always in Case 1, except if $\mathbf{b}' = \mathbf{b}''$. In fact, if n is the number of multiplicative switches, there are 2^n simultaneous cases that may occur, and, by symmetry, the argument made in this particular situation immediately extends. From this Lemma 3.18.1 will follow.

(iv) We shall make a trip starting with a given terminal formula A_1. The hypothesis is $A_1 : \asymp$. At the end of the trip, we want to arrive with the conclusion $A_1 : \supset\!\!\subset$. We shall visit all formulas twice: upwards, we shall conclude $A : \asymp$; downwards, we shall conclude $A:\supset$. Hence, a formula A visited twice is such that $A:=$, and we shall be done. When we start, the switches are not yet set; this will be done as follows:

(1) Assume that we arrive for the *first* time in a link $\frac{A \quad B}{A\otimes B}$, and that we arrive through the conclusion $A\otimes B$; so we are upwards, i.e., $A\otimes B:\asymp$. Since we are in Case 1, $A:\asymp$, so set the *switch on* "R" to go to $A^{\wedge}$.

(2) Assume that we arrive for the *first* time in a link $\frac{A \quad B}{A\otimes B}$, and that we arrive through the premise A; so we are downwards, i.e., $A:\supset$. Since we are in Case 1, $A\otimes B:\supset$, so set the *switch on* "L" to move to $A\otimes B_{\vee}$.

(3) As in (2), but arriving through B: *switch on* "R".

(4) Assume that we arrive for the *first* time in a link $\frac{A \quad B}{A ⅋ B}$, and that we arrive through the conclusion $A ⅋ B$; so we are upwards, i.e., $A ⅋ B:\asymp$. But then, observe that $B:\asymp$, so set the *switch on* "R" to move to $B^{\wedge}$.

(5) Assume that we arrive for the *first* time in a link $\frac{A \quad B}{A ⅋ B}$, and that we arrive through the premise A; so we are downwards, i.e., $A:\supset$. Since we are in Case 1, $A ⅋ B:\supset$, so set the *switch on* "L" to move to $A ⅋ B_{\vee}$.

(6) As in (5), but arriving through B: we are downwards, i.e., $B:\supset$; we have two subcases:

(6)(a): if $B:=$, then *switch on* "L" to go to $B^{\wedge}$;

(6)(b): if $B:\neq$, then $A ⅋ B:\frown$; hence, $A ⅋ B:\supset$; *switch on* "R" to go to $A ⅋ B_{\vee}$.

(7) Assume that we arrive for the *second* time in a link $\frac{A \quad B}{A\otimes B}$ and that the first passage was as in (1); then we come back through $A_{\vee}$, i.e., $A:\supset$. Since we are in Case 1, $A\otimes B:\supset$ and since we had $A\otimes B:\asymp$, we get $A\otimes B:=$. But then, $B:\supset$ and the next step to $B^{\wedge}$ will be sound.

(8) Assume that we arrive for the *second* time in a link $\frac{A \quad B}{A\otimes B}$, and that the first time was as in (2) or (3); now, the order of passage is such that we arrive through $A\otimes B^{\wedge}$, i.e., $A\otimes B:\asymp$; but we already had $A\otimes B:\supset$, so $A\otimes B:=$ and so, $A:\asymp$, $B:\asymp$ and the next step to $A^{\wedge}$ or $B^{\wedge}$ will be sound.

(9) Assume that we arrive for the *second* time in a link $\frac{A \quad B}{A ⅋ B}$, and that the first time was as in (4); so we arrive through $A_{\vee}$, i.e., $A:\supset$; but we had $A ⅋ B:\asymp$ from which $A:\asymp$, so $A:=$, but we are in Case 1, hence, $A ⅋ B:=$. The next step of the trip to $A^{\wedge}$ is clearly sound since $A:\supset$.

(10) Assume that we arrive for the *second* time in a link $\frac{A \quad B}{A ⅋ B}$, and that the first passage was as in (5) or (6); we have two subcases:

(10)(a): first passage as in (6)(a): we arrive through $A_{\vee}$, i.e., $A:\supset$, but we had $B:=$, hence, $A ⅋ B:\supset$ and the next step to $A ⅋ B$ will be sound;

(10)(b): first passage as in (5) or (6)(b): we arrive through $A ⅋ B^{\wedge}$ and we had already passed through $A ⅋ B_{\vee}$, so $A ⅋ B:=$, hence, the next step to $A^{\wedge}$ or $B^{\wedge}$ will be sound, since $A:=$, $B:=$.

(11) Assume that we arrive for the *third* time in a multiplicative link $\frac{A \quad B}{C}$; we have already seen in cases (7), (8), (9) and (10)(b) that $C:=$. In case (10)(a) for the second passage, we are in $A ⅋ B^{\wedge}$ after being in $A ⅋ B_{\vee}$, so here too, $C:=$; but then, in every case, $A:=$, $B:=$, and the next step is sound.

(12) Assume that we arrive in a *conclusion* A_i, i.e., $A_i : \bigcirc$; then we can move to $A_i^{\wedge}$ because of the hypothesis $A_i : \asymp$.

(13) Assume that we arrive on one side of a cut-rule $\frac{A \quad A^{\perp}}{\text{CUT}}$, say through $A_{\vee}$; then, $A : \bigcirc$ from which it is immediate that $A^{\perp} : \asymp$ (the points are the same) and so the move to $A^{\perp\wedge}$ is logically sound.

(14) Assume that we arrive on one side of an axiom $\overline{A \quad A^{\perp}}$, say through $A^{\wedge}$; then, $A : \asymp$ from which it is immediate that $A^{\perp} : \bigcirc$ (the points are the same) and the move to $A^{\perp}_{\vee}$ is logically sound.

(v) Summing up now, we have obtained $A : =$ for any formula A. □

3.19. Remark. The same argument works for PN2, with a heavier apparatus. In experiments, we have to choose existential witnesses, but the same unicity can be obtained; i.e., there is only one successful experiment. (In a rule

$$\frac{A[B/a]}{\bigvee a.A}$$

observe that $\bigvee a.A : \asymp$ implies $A[B/a] : \asymp$, and that $A[B/a] : \bigcirc$ implies $\bigvee a.A : \bigcirc$.)

4. Normalization in PN2

We shall define a normalizing algorithm for proof-nets. The algorithm is a rewriting procedure (*contraction*) whose properties are as follows:

(i) If β **cntr** β', then β and β' have the same terminal formulas (conclusions).

(ii) If β **cntr** β', then $\beta'^{*} = \beta^{*}$ (semantic soundness).

(iii) There is a β' such that β **cntr** β' iff β contains CUT-links; a proof-net β with no CUT-link is said to be *normal*.

Reduction is defined as the transitive closure of **cntr**, and is denoted $B =/ \beta'$. It is sometimes useful to speak of a *reduction sequence* from β to β', i.e.,

$$\beta = \beta_0 \,\mathbf{cntr}\, \beta_1 \,\mathbf{cntr} \cdots \beta_{n-1} \,\mathbf{cntr}\, \beta_n = \beta'.$$

We can also use the notation $\beta =/_n \beta'$ to indicate the existence of a reduction sequence of length $n+1$ from β to β'. The main property of reduction is the following:

(iv) *Strong normalization*: A proof-net β is said to be SN when there is no infinite reduction sequence starting from β; in this case we define the integer $N(\beta)$ as the greatest n such that $\beta =/_n \beta'$ for some β'. (By König's Lemma, using the fact that, for a given β, there are finitely many β' such that $\beta =/ \beta'$, $N(\beta)$ is definable.) It may be useful to define $N(\beta)$ in general by

$$N(\beta) = \{\sup n ; \exists \beta' (\beta =/_n \beta')\}$$

so that β is SN exactly when $N(\beta) < \omega$. The Strong Normalization Theorem, which is the main result of this section, says that all proof-nets of PN2 are SN. One of the main problems with our procedure is that it is not Church–Rosser, i.e., β may reduce to *normal forms* β' and β'' which are distinct. However, $\beta'^* = \beta''^*$ so that they do not differ too much. In particular we shall see in Section V how to represent booleans, integers, etc. in PN2.

A given integer (say 10) has several representations in PN2, all very close one to another. What we want is that if one reduction sequence yields the result 10, another does not yield the result 28; semantically speaking however, any representation of 10 in PN2 is interpreted in the same way as some set 10*, similarly for 28 and, clearly, $10^* \neq 28^*$; hence there is no ambiguity.

One of the most interesting features of strong normalization is the possibility to ignore certain normalization rules. If we decide to use all contractions except, say (R), we eventually reach a normal form relative to (R), i.e., a proof-net where the only contractions that can be performed are instances of (R). In some cases, this is enough to ensure that we reach a normal form without using (R).

Let us give an example demonstrating this possibility: Imagine that we ignore the commutation rule for &. If we start with a proof-net not involving & or $\bigvee$ in its conclusions, and take a relative normal form β for it, then assume β is not normal. We start with a CUT $\frac{C \quad C^\perp}{\text{CUT}}$, where $C^\perp$ is the auxiliary door of a &-box. So let us go to the front-door C' of this box: the downmost formula below C', say D, bears one of the symbols & or $\bigvee$, so it cannot be a conclusion, and is therefore the premise of a CUT-rule. The other premise $D^\perp$ is again the auxiliary door of a &-box. Iterating the process we eventually find a cycle since there are finitely many cuts in β from which a shorttrip is easily made. Thus, we have contradicted the assumption that β was not cut-free.

By the way, observe that it was also possible in that case to forbid contractions inside &-boxes and still arrive at a normal form: strong normalization justifies *lazy* strategies for computation.

We shall now describe the normalization procedure by giving the contraction rules. They are divided into three groups, *axiom* contraction (Section 4.1) *symmetric* contractions (Section 4.2), and *commutative* contractions (Section 4.3). For each of these groups we establish, with the definition, the semantic soundness of the contraction rule. In order to save space, it is very convenient to consider the CUT-link as symmetric, i.e., $\frac{C \quad C^\perp}{\text{CUT}}$ and $\frac{C^\perp \quad C}{\text{CUT}}$ are exactly the same link. In particular, in our graphical representations, it will be possible to put what is most convenient on the left, in order to avoid duplicating all cases!

4.1. Axiom contraction

The axiom contraction (AC) can be performed every time one of the premises of a cut-link is the conclusion of an axiom link: it consists in replacing a configuration:

$$\begin{array}{ccc} \overline{A \qquad A^\perp} & \downarrow & \\ & A & \\ \uparrow \qquad \text{CUT} & & \end{array} \qquad \text{by:} \qquad \begin{array}{c} \downarrow \\ A, \\ \uparrow \end{array}$$

i.e., by identifying the two occurrences of A and removing $A^\perp$ and the CUT- and axiom-links. We have already seen that the two occurrences of A are necessarily distinct.

We show the semantic soundness of (AC): assume that β **cntr** β' by (AC), then $\beta^* = \beta'^*$ is shown by induction on the number n of boxes containing the CUT we contract.

Basis step: If $n = 0$, then consider an experiment in β' corresponding to a certain choice of values z for the conclusions. If the experiment succeeds, let x be the point chosen in the formula A. Now, we can form an experiment in β corresponding to the same initial choice z: for the cut-rule, choose x, both in A and $A^\perp$, and report the values from the given experiment everywhere else. This new experiment succeeds, so $\beta'^* \subset \beta^*$. Conversely, from a successful experiment in β, it is not difficult to deduce a successful experiment in β': at the level of the cut-rule, we have to guess a value $y \in |A| = |A^\perp|$, but, in order for the experiment to succeed, this value has to be the same as the value x already chosen in the 'left' A, and so it is perfectly harmless to identify the two A's: $\beta^* \subset \beta'^*$.

Induction step: If $n \neq 0$, then the contraction takes place within a box $\mathbb{B}$, replaced by $\mathbb{B}'$. Now, boxes are built in such a way that (using the induction hypothesis) $\mathbb{B}^* = \mathbb{B}'^*$. Then the semantics of β and β' will coincide since they are built in the same way from these boxes. (In later contractions, as this argument is perfectly general, we shall always assume that the contraction is not done within a box.)

4.2. Symmetric contractions

This concerns all CUT-links where one side comes by a rule, whereas the other side comes by the dual rule.

4.1. Definition (*T-contraction*). Since there is no rule for **0**, this case never occurs!

4.2. Definition (*addition contraction*: (&/1⊕-SC)). A configuration as in Fig. 18(a) is replaced by the one in Fig. 18(b).

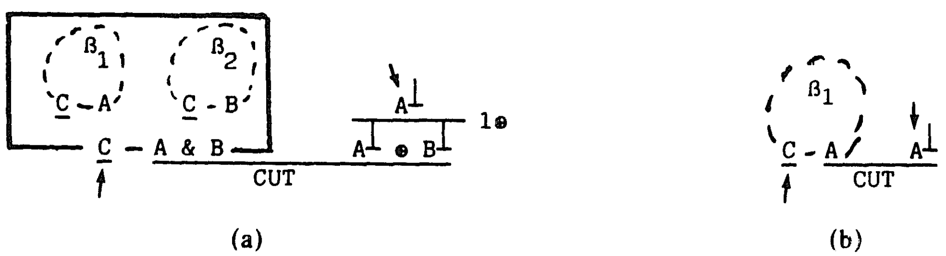


Fig. 18. (&/1 ⊕-SC).

The semantic soundness is shown as follows: In β, at some moment, we have $c \in |C|$ and we have to guess an element $(0, a)$ or $(1, b)$ of $|A \,\&\, B|$. Inside the box, one goes on with c, a in β_1 or c, b in β_2, according to the case. However, due to the (1⊕)-rule, the choice of $(1, b)$ would be a failure, so the only successful choice is $(0, a)$, in which case we proceed above $A^\perp$ by choosing a. Now, in β', we have to choose some a in $|A|$, and then go up in β_1 and in $A^\perp$. This establishes clearly that $\beta^* = \beta'^*$.

4.3. Definition (*additive contraction*: (&/2⊕-SC)). This case is perfectly symmetric to (&/1⊕-SC).

4.4. Definition (*unit contraction*: (1/⊥-SC)). A configuration as in Fig. 19(a) is replaced by the one in Fig. 19(b).

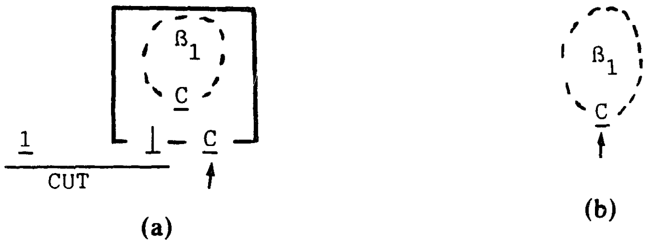


Fig. 19. (1/⊥-SC).

The semantic soundness is immediate: In β, we have to guess an element of $|\mathbf{1}|$, and there is only one choice: 1. This value is reported in the box, but not used. What matters is the choice of values $c \in C$, just as in β', so $\beta^* = \beta'^*$.

4.5. Definition (*multiplicative contraction*: (⊗/⅋-SC)). A configuration as in Fig. 20(a) is replaced by the one in Fig. 20(b).

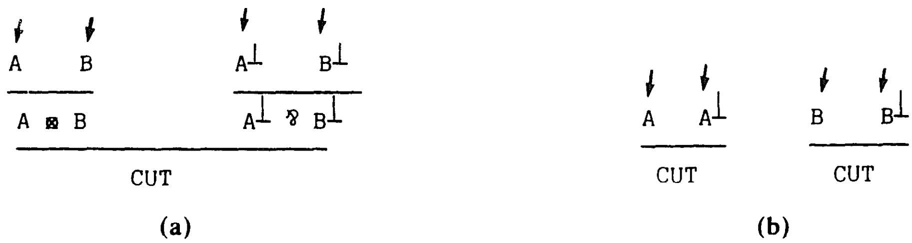


Fig. 20. (⊗/⅋-SC).

The semantic soundness is proved as follows: In β we have to guess a pair (x, y), then report it in the two premises, and then dispatch the components, x in A and $A^\perp$, y in B and $B^\perp$. But this amounts exactly to guessing (in β') x in $|A|$ and y in $|B|$, etc.

4.6. Definition (*exponential contraction* (!/ W?-SC)). A configuration as in Fig. 21(a) is replaced by the one in Fig. 21(b), where several weakenings are performed. This is a typical example where Church–Rosser is violated. Now, if we were allowed to put several weakenings in the same box (what any reasonable implementation should do), the problem would at least disappear from this case.

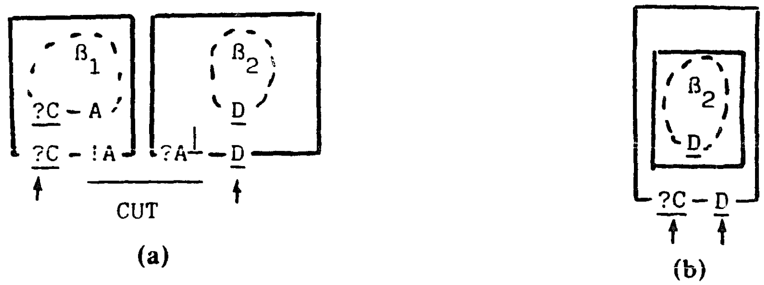


Fig. 21. (!/*W*?-SC).

The semantic soundness is proved as follows: In β, we are given sequences $\boldsymbol{c}$ (in ?C) and $\boldsymbol{d}$ (in D). We have to guess an element $\boldsymbol{a}$ of $|!A|$. Now, the element has to be transferred into the right box, and the general definition says that $a = \emptyset$. Also, if $a = \emptyset$, the general definition for the left box imposes the requirement that all elements of the sequence $\boldsymbol{c}$ are void, nothing else. Hence, in β, we only require $c = \emptyset$ and then transfer $\boldsymbol{d}$ to β_2, which is exactly what we do in β'.

4.7. Definition (*exponential contraction* (!/*D*?-SC)). A configuration as in Fig. 22(a) is replaced by the one in Fig. 22(b).

Fig. 22. (!/*D*?-SC).

The semantic soundness is established as follows: In β, given the sequence $\boldsymbol{c}$ in ?C, we have to guess a point $a \in |!A|$. Now, the dereliction rule imposes that a is a singleton $\{z\}$, and we proceed with z in $A^{\perp}$. But, if we enter the box with $\boldsymbol{c}$ and $\{z\}$, this means that c, z are reported inside, as can be seen from the 'unicity remark' in Definition 3.15(xv). This means that, in fact, we guess a point z in $|A|$, and proceed as in β'.

4.8. Definition (*exponential contraction* (!/*C*?-SC)). A configuration as in Fig. 23(a) is replaced by the one in Fig. 23(b). In this contraction, there is a duplication of a

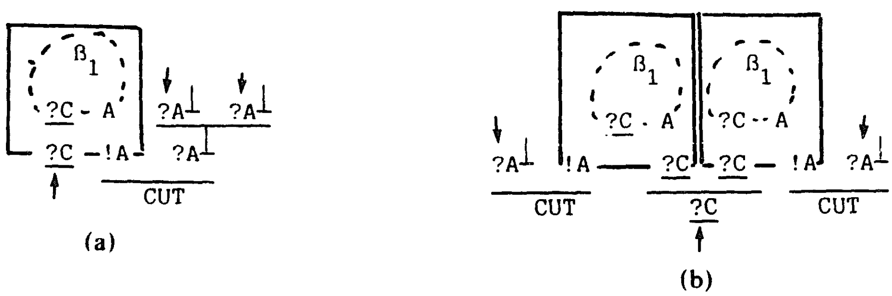


Fig. 23. (!/*C*?-SC).

box, and several rules ($C?$) are used; this rule is responsible for the loss of control over the normalization time.

The semantic soundness of the rule is established as follows: In β, we start with c in $?C$, and we guess some a in $|!A|$. For the rule ($C?$), we have to guess again a decomposition $a = a' \cup a''$. We enter the box with c, a. In β', we have to guess elements a' (left) and a'' (right) of $|A|$. But we also have to guess a decomposition of c as $c' \cup c''$. Then we have two boxes to enter: one with c', a' and one with c'', a''. Now, the box $\mathbb{B}$, which is used in β and recopied twice in β', is such that if $c', a'/c'', a''$ belong to $\mathbb{B}^*$, so do $c' \cup c''$ and $a' \cup a''$. Conversely, if $c, a' \cup a'' \in \mathbb{B}^*$, then c can be uniquely decomposed as $c' \cup c''$ so that $c', a'/c'', a'' \in \mathbb{B}^*$. This means that the successful experiments in both proof-nets are in 1-1 correspondence and we are done.

4.9. Definition (*quantification contraction* ($\bigwedge/\bigvee$-SC)). Replace a configuration as in Fig. 24(a) by the one in Fig. 24(b).

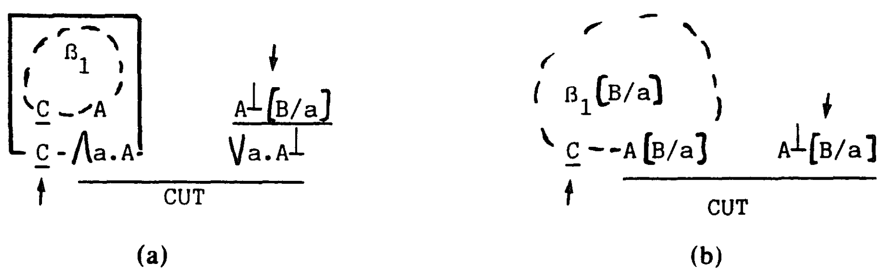


Fig. 24. ($\bigwedge/\bigvee$-SC).

The semantic soundness is checked as follows: We start with c in C and we have to guess an element (X, z) in $|\bigwedge a.A|$. Above the rule ($\bigvee$), we have to guess an element $f \in \text{COH}(X, B)$ in order to get $z' = A[f/a]^+(z)$, a point in $|A[B/a]|$. We also have to check that c and (X, z) are in $\mathbb{B}^*$, where $\mathbb{B}$ is the left box. Now, the general functoriality properties (see [5]) make it the same as to verify that c, z' is in $\beta_1[B/a]^*$. From this, the soundness of this case follows.

4.3. Commutative contractions

4.10. Definition. In order to state the commutative contractions, we need the concept of a *ghost box*. The ghost boxes can be formed for CUT- or $\otimes$-links (this is just a geometrical feature), and we discuss them for a CUT-link only. So, if we have such a link $\frac{B \quad B^\perp}{\text{CUT}}$, we can consider the empire eB of B. In eB, there is a *frontier*, made of B, of the conclusions of β belonging to eB (in this description, we assume that the CUT is not in a box, just to simplify everything), and of premises of ($⅋$)- and ($C?$)-links such that the other premise is not in the box. Let us list the frontier of eB as A, B; we can form a box (the ghost box) by putting eB inside, and the doors of the box will be A, B. If we replace eB in β by this ghost box, we are executing a *materialization.*

4.11. Remarks. (i) In terms of computation, materialization is a very feasible process since it suffices to start a trip with $B^\wedge$ and to set the switches arbitrarily. However, in the case we enter a rule (⅋) or (C?) for the first time and from above, the switch is set such that we move backwards: when we eventually arrive in $B_\vee$, we have passed exactly through *eB*, so there is no problem to describe *eB* very quickly.

(ii) When, in turn, our CUT is in a box, then the box is made out of one of several proof-nets arranged together, and the CUT occurs in one of these nets; in this net, the concept of a ghost box makes sense, and it is in this way that we define the ghost box of a nested cut.

(iii) The main point is to prove that *materialization* produces a sound proof-net; in fact, we shall establish more, namely that we can simultaneously materialize *eB* and $eB^\perp$ as is stated in the following theorem.

4.12. Theorem. *The proof-structure obtained by materializing eB and* $eB^\perp$ *is sound.*

Proof. We argue by induction on the number of links outside $eB \cup eB^\perp$. In the case of PN0 all the difficulties are concentrated, hence, we restrict ourselves to this case.

(i) If the only link outside $eB \cup eB^\perp$ is the CUT-link, then the proof splits and the materialization looks as shown in Fig. 25, which is a proof-net.

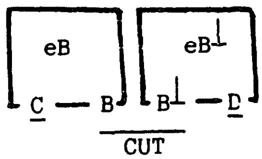


Fig. 25.

(ii) If there is a terminal ⅋-link outside $eB \cup eB^\perp$, then remove it in order to get β'. Now, the materialization of β is the materialization of β' to which we have added the terminal ⅋-link.

(iii) *Otherwise*, observe that there is a terminal ⊗- or CUT-link with premises E and F such that $eE \cup eF$ is maximal and $eB \cup eB^\perp \subset eE$ (or eF). The reason for this is that, when we have a terminal ⊗- or CUT-link such that the empires eC and eD above are not such that $eC \cup eD$ is maximal, then we can conclude the existence of a conflicting ⅋-link, as in the Splitting Theorem; now, what is below this conflict is in neither empires; hence, if all terminal ⅋-links are in $eC \cup eD$, it follows that there is a terminal ⊗- or CUT-link below the conflict. We can conclude as in the Splitting Theorem that $eC \cup eD \subset eE$ (or eF). Now, there is a splitting of β as shown in Fig. 26 (case of a tensor-link, $eC \cup eD \subset eE$). From the materialization in β_1, it is easy to obtain the materialization in β. □

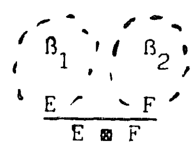


Fig. 26.

It remains to examine the CUT-links which are not covered by the two cases already considered in the proof of Theorem 4.12. The only possibility is that at least one premise is the auxiliary door of a box.

4.13. Definition (*zero commutation* (⊤-CC)). The configuration in Fig. 27(a) is replaced by the one in Fig. 27(b). To do this, we have proceeded as follows: In a

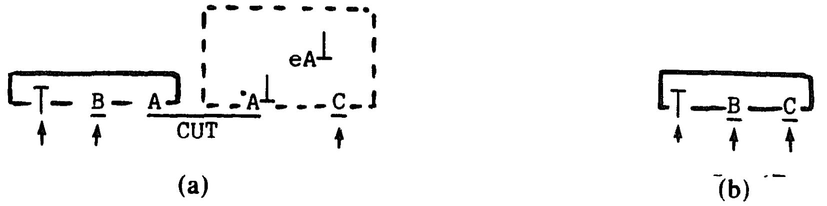


Fig. 27. (⊤-CC).

first time, we have a materialization of the ghost box $eA^{\perp}$, in a second time, we have the replacement of the configuration involving both boxes by another one.

The semantic soundness is vacuously satisfied since, due to the presence of ⊤ with its void web, in both cases we never have anything to verify at this stage.

4.14. Definition (*additive commutation* (&-CC)), The configuration in Fig. 28(a) is replaced by the one in Fig. 28(b).

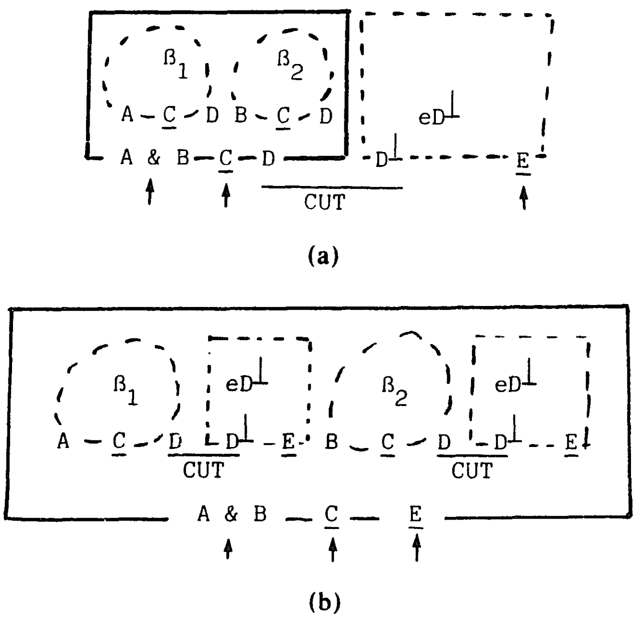


Fig. 28. (&-CC).

The semantic soundness is established as follows: We are given c, e in C, E and a point in $|A \& B|$, say $(0, a)$. In β we have to guess $d \in |D|$; from this we proceed by putting $(0, a)$, c, d in the left box and after a, c, d in β_1, we also put d, e in the ghost box. In β', we proceed in a different order, namely that $(0, a)$, c, e is transferred to the left box and then looks as *a*, *c*, *e*, inside the box. Then we have to guess $d \in |D|$ and then dispatch *a*, *c*, *d* in β_1, *d*, *e* in $eD^{\perp}$.

Remark: Of course, the ghost box is used only temporarily: after writing the contractum, one erases it; the fact that the erasing of a ghost box causes no problem is practically immediate.

4.15. Definition (*unit commutation* ($\perp$-CC)). The configuration in Fig. 29(a) is replaced by the one in Fig. 29(b).

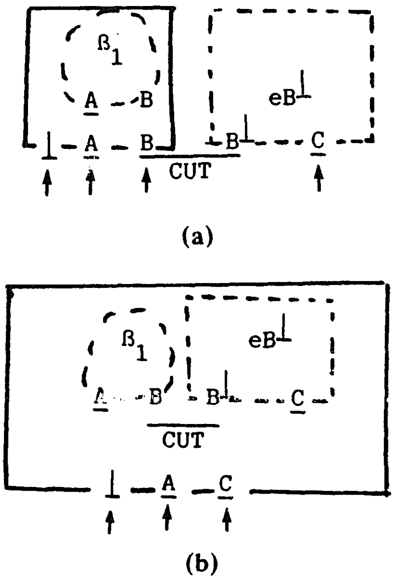


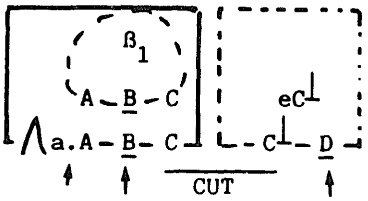
Fig. 29. ($\perp$-CC).

The semantic soundness of this contraction is established as follows: In β, we start with $1 \in |\perp|$, *a* in *A*, and $c \in |C|$. We then have to guess some $b \in |B|$. Once *b* has been chosen, these elements are dispatched between the two boxes: *b*, *c* in the ghost box, 1, *a*, *b* in the left box, i.e., *a*, *b* in β_1. What happens in β is almost the same: 1, *a*, *b* is transferred to the box so that it becomes *a*, *C*; then we guess *d*, and *a*, *b* go to β_1 while *b*, *c* go to *eB*.

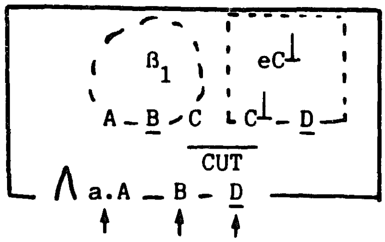
4.16. Definition (*exponential commutation* ($W?$-CC)). This case is exactly the same as the one in Definition 4.15 except that the formula $\perp$ is replaced by a formula $?D$.

4.17. Definition (*quantificative commutation* ($\bigwedge$-CC)). The configuration in Fig. 30(a) is replaced by the one in Fig. 30(b).

The soundness is proved as follows: In β, we start with $(X, z) \in |\bigwedge a.A|$, *b* in *B*, and *d* in *D*; we guess $c \in |C|$. From this, we dispatch *c*, *d* to $eC^{\perp}$ and (X, z), *b*, *c* to β_1. This means that *z*, *b*, *c* is transferred to $\beta_1[X/a]$. In β', starting with the same data, we transfer them to the interior of the box and they become *z*, *b*, *d*, but β_1 has been replaced by $\beta_1[X/a]$. Then we guess $c \in |C|$, and *z*, *b*, *c* is transferred to $\beta_1[X/a]$ while *c*, *d* goes to $eC^{\perp}$.



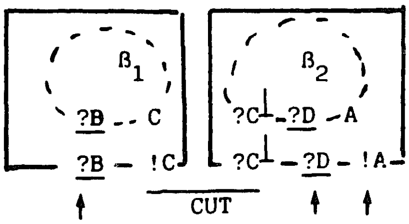
(a)



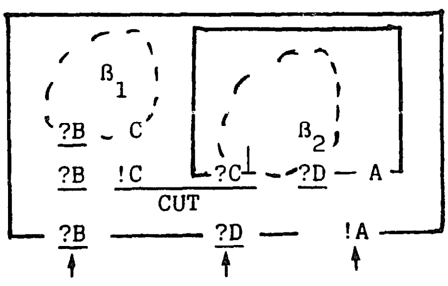
(b)

Fig. 30. (⋀-CC).

4.18. Definition (*exponential commutation* (!-CC)). The last case is that of a !-box. Here, the ghost boxes would be of no use, so there is no general pattern of commutation. However, we must be able to do something in all possible CUT-situations. Let us look, however, at the other premise of the CUT: it is of the form !*C*. Now, if we cannot do something on that side, this means that !*C* is neither the conclusion of an axiom-link nor the auxiliary door of a box; the only possibility is that !*C* is the main door of a !-box: the configuration in Fig. 31(a) is replaced by the one in Fig. 31(b).



(a)



(b)

Fig. 31. (!-CC).

In order to prove the soundness, consider b, d, a in $?B$, $?D$, $?A$; we first have to guess (in β) an element $c \in |C|$, and then to dispatch it between the two boxes: on the left, it becomes b, c on the right c, d, a. Assume that $a = \{x_1, \ldots, x_n\}$, $c = \{z_1, \ldots, z_m\}$; within the boxes, we have to guess decompositions $b = b_1 \cup \cdots \cup b_m$, and then consider simultaneously $b_1, z_1 / \cdots / b_m, z_m$. Further, $c = c_1 \cup \cdots \cup c_n$, $d = d_1 \cup \cdots \cup d_n$ and we have to consider simultaneously $c_1, d_1, x_1 / \cdots / c_n, d_n, x_n$. Now, consider $b'_1, \cdots, b'_n$ defined by $b'_i = \bigcup\{b_j ; z_j \in c_i\}$. Then the left part of the experiment amounts to the same as verifying that $b'_1, c_1 / \cdots / b'_n, c_n$ belong to the left box. Now, if we start with the same initial data in β', we are first faced with guessing a decomposition $b = b'_1 \cup \cdots \cup b'_n$ and $d = d_1 \cup \cdots \cup d_n$ followed by separately studying $b_1, d_1, x_1 / \cdots / b_n, d_n, x_n$. In the case of b_i, d_i, x_i, we have to guess some c_i in $|C|$ and then dispatch between the two sides b_i, c_i on the left, c_i, d_i, x_i on the right, and the problem is therefore strictly equivalent to the problem of β.

4.19. Remark. Any binary link with the same geometry as CUT, typically $\otimes$, can give rise to commutative contractions; in particular, the analogues of Definitions 4.13–4.18 for $\otimes$ are easily written.

Our goal is, as stated, *strong* normalization. It is however quite interesting to give the *Small Normalization Theorem* first, which gives a very quick normalization proof for PN2, provided ($!/C?$-SC) is not used.

4.20. Definition. The *size* $s(\beta)$ of a proof-net β is defined by induction on β as follows:

(i) If β is built from boxes $\mathbb{B}_1, \ldots, \mathbb{B}_n$ and formulas $A_1, \ldots, A_k$ by means of usual unary or binary links, then $s(\beta) = s(\mathbb{B}_1) \cdot \ldots \cdot s(\mathbb{B}_n) \cdot 2^k$ (in other terms, formulas are counted to be of size 2).

(ii) If β is a box, then the size of β is defined as follows:

- if β is a $\top$-box, $\top, C_1, \ldots, C_n$, then $s(\beta) = 2 + n$;
- if β comes from β' by the schemes for $\perp$, $W?$, $\bigwedge$, $!$, then $s(\beta) = s(\beta') + 1$;
- if B comes from β', β'' by the scheme for &, then $s(\beta) = s(\beta') + s(\beta'') + 1$.

4.21. Lemma. *In all contraction rules but* ($!/C?$-SC), *the size strictly diminishes.*

Proof. We have to go through 14 cases. First, observe that $s(\beta) > n$ if β has n conclusions. In order to prove the lemma, one can make the simplifying hypothesis that the cut contracted is not inside a box.

- (AC) divides the size by 4;
- in (&/1$\oplus$-SC), a size $(x+y+1) \cdot 2z$ is replaced by a size $x \cdot z$;
- in ($1/\perp$-SC), a size $2 \cdot (x+1)$ is replaced by a size x;
- in ($\otimes/⅋$-SC), a size $x \cdot 2^6$ is replaced by a size $x \cdot 2^4$;
- in ($!/W?$-SC), a size $(x+1) \cdot (y+1) \cdot z$ is replaced by a size $(y+n) \cdot z$ for some $n \leq x$;

- in ($!/D?$-SC), a size $(x+1)\cdot 4z$ is replaced by a size $x\cdot 2z$;
- in ($\bigwedge/\bigvee$-SC), a size $(x+1)\cdot 4z$ is replaced by a size $x\cdot 2z$;
- in ($\top$-CC), a size $(3+n)\cdot x\cdot y$ is replaced by a size $(2+n+m)\cdot y$, with $m<x$;
- in (&-CC), a size $(x+y+1)\cdot z\cdot t$ is replaced by a size $(xz+yz+1)\cdot t$;
- in ($\perp$-CC), a size $(x+1)\cdot y\cdot z$ is replaced by a size $(xy+1)\cdot z$;
- in ($\bigwedge$-CC), a size $(x+1)\cdot y\cdot z$ is replaced by a size $(xy+1)\cdot z$;
- in (!-CC), a size $(x+1)\cdot(y+1)\cdot z$ is replaced by a size $(x\cdot(y+1)+1)\cdot z$.

All cases have been treated except trivial variants. □

4.22. Corollary (*Small Normalization Theorem*). PN2 *without* ($!/C?$-SC) *satisfies strong normalization.*

Proof. Clearly, $N(\beta)\leqslant s(\beta)$. □

4.23. Remarks. (i) In ($!/C?$-SC), a size $(x+1)\cdot 8z$ becomes $(x+1)^2\cdot 4z\cdot 2^n$, where $n<x$. Suddenly, the size explodes. Is it possible to fix this defect? Obviously not, because PN2 has, roughly spoken, the same expressive power as the system *F*, i.e. the normalization theorem for *F* is not provable in PA_2, so it would simply be childish to try to improve the Small Normalization Theorem by toying with the definition. There is however an open possibility: one can define generalized sizes (no longer integers), typically in the case of the !-box, so that we still have a phenomenon of decrease; these generalized numbers could be something like ordinals, dilators, ptykes. Such an achievement seems highly probable (but difficult); it would immediately yield something like an ordinal analysis of PA_2.

(ii) To some extent, the Small Normalization Theorem is more important than its more refined and delicate elder brother! This is because normalization is now clearly divided into two aspects.

(1) The quick normalization devices which reduce size: These devices must be seen as *communication devices* and our small theorem says that this part works wonderfully well and quickly in PN2 (for those who may be afraid by the exponentials involved in sizes, remember that there is no necessity, no intention to work sequentially...). In particular, we could have put nonterminating devices inside those awful !-boxes; this would not affect the quality of the parallelism, the programs would simply run forever. But the structure Q/A would remain sound.

(2) The device ($!/?C$-SC), which can be seen as the *execution device* of the system: The Strong Normalization Theorem will prove that this device, together with the others, leads to terminating algorithms, and this is essential for the expressive power of PN2 which should be essentially the same as that of *F*. General parallel systems, for instance those that run forever, could be obtained by taking different execution devices, to be put in carefully sealed boxes!

In spite of the defect w.r.t. the Church–Rosser property, the interconvertibility w.r.t. the communication devices could be seen as a serious candidate for expressing equivalence of programs.

We now turn our attention towards the general case; for this we need to establish some facts about strong normalization relating it to simple normalization.

4.24. Definition. A contraction is *standard* when it does not erase any symbol CUT besides the one explicitly considered. Concretely, this means that some parts of the configuration we replace have to be cut-free, namely:
- in (&/1⊕-SC), β_2 must be cut-free,
- in (&/2⊕-SC), β_1 must be cut-free,
- in (!/ W?-SC), β_1 must be cut-free,
- in (⊤-CC), $eA^{\perp}$ must be cut-free.

A reduction sequence is *standard* when made of standard contractions.

4.25. Theorem (Standardization Lemma).[5] *Let β be a proof-net and assume that there is a standard reduction from β to a cut-free β'. Then β is SN.*

The delicate but boring proof is postponed. We immediately jump to the proof of the Strong Normalization Theorem.

4.26. Theorem (Strong Normalization Theorem). *In* PN2 *all proof-nets are strongly normalizable.*

Proof. The idea is to adapt the technique of 'candidats de réductibilité' of [1] to the case of PN2; in fact, the symmetry of linear negation makes it easier to some extent. During the proof, the Standardization Lemma (Theorem 4.25) is constantly used.

4.26.1. Definition. A *term of type A* is a proof-net β together with a distinguished conclusion which is an occurrence of A. In notations, we shall always make the harmless abuse that consists in considering the distinguished conclusion to be obvious from the context.

4.26.2. Definition. Let X be a set of terms of type A; we define $X^{\perp}$ as the set of those terms u of type $A^{\perp}$ such that, for all t in X, the proof-net CUT(t, u) (see Fig. 32) is SN

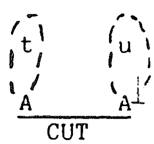


Fig. 32.

4.26.3. Proposition. (i) *If $X \neq \emptyset$, then all elements of $X^{\perp}$ are SN.*

(ii) *If all elements of X are SN, then $X^{\perp}$ is nonvoid; it contains in fact the term $A \quad A^{\perp}$ of type $A^{\perp}$.*

[5] The proof will be sketched in Section 6.

Proof. (i): If CUT(t, u) is SN, then u is SN.

(ii): This proof already uses the Standardization Lemma: CUT(t, u) **contr** t by a standard contraction when u is the axiom link. Hence, if u is SN, there is a standard reduction from u to a normal form, and therefore, from t: but then t is SN, too. □

4.26.4. Definition. A CR (*candidat de réductibilité*) of type A is a set X of terms of type A such that

(i) $X \neq \emptyset$;

(ii) all terms of X are SN;

(iii) $X = X^{\perp\perp}$.

4.26.5. Examples. A typical way to generate a CR of type A is to take the orthogonal $Y^\perp$ of any nonvoid set Y of SN-terms of type $A^\perp$. In particular, taking Y to consist of an axiom link, we obtain the CR SN(A) formed of all SN terms of type A.

4.26.6. Definition. Assume that $A[\boldsymbol{a}]$ (where $\boldsymbol{a} = a_1, \ldots, a_n$ is a sequence of propositional variables) is a proposition, and let $\boldsymbol{B}$ $(= B_1, \ldots, B_n)$ be a sequence of propositions; let $\boldsymbol{X}$ $(= X_1, \ldots, X_n)$ be a sequence of CR's of respective types $\boldsymbol{B}$. Then we define a CR RED($A[\boldsymbol{X}/\boldsymbol{a}]$) of type $A[\boldsymbol{B}/\boldsymbol{a}]$, as follows:

(i) $\text{RED}(\mathbf{1}[\boldsymbol{X}/\boldsymbol{a}]) = \{\mathbf{1}\}^\perp$; $\text{RED}(\top[\boldsymbol{X}/\boldsymbol{a}]) = \text{SN}(\top)$;

(ii) if A is a_i, then RED($A[\boldsymbol{X}/\boldsymbol{a}]$) is X_i;

(iii) if A is $A' ⅋ A''$, then RED($A[\boldsymbol{X}/\boldsymbol{a}]$) is defined as the orthogonal of the set $\text{RED}(A'[\boldsymbol{X}/\boldsymbol{a}])^\perp.\text{RED}(A''[\boldsymbol{X}/\boldsymbol{a}])^\perp$ formed of all proof-nets as shown in Fig. 33 with $t \in \text{RED}(A'[\boldsymbol{X}/\boldsymbol{a}])^\perp$, $u \in \text{RED}(A''[\boldsymbol{X}/\boldsymbol{a}])^\perp$.

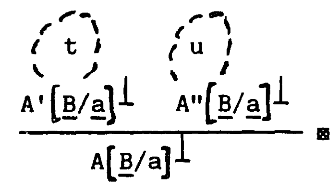


Fig. 33.

(iv) If S is $A' \& A''$, then RED($A[\boldsymbol{X}/\boldsymbol{a}]$) is defined as the orthogonal of the union of the two sets:

$$⫫^1(\text{RED}(A'[\boldsymbol{X}/\boldsymbol{a}]^\perp)) \quad \text{and} \quad ⫫^2(\text{RED}(A''[\boldsymbol{X}/\boldsymbol{a}]^\perp))$$

formed of all proof-nets obtained from a term t of $A'[\boldsymbol{X}/\boldsymbol{a}]^\perp$ (respectively $A''[\boldsymbol{X}/\boldsymbol{a}]^\perp$) by applying (1⊕) (respectively (2⊕)) (see Fig. 34).

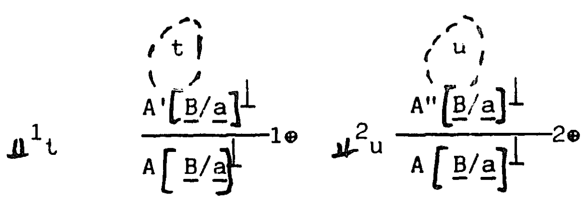


Fig. 34.

(v) If A is $?A'$, then $\mathrm{RED}(A[\boldsymbol{X}/\boldsymbol{a}])$ is defined as the orthogonal of the set $\S\mathrm{RED}(A'[\boldsymbol{X}/\boldsymbol{a}]^{\perp})$ formed of all terms of type $!A'[\boldsymbol{B}/\boldsymbol{a}]^{\perp}$ of the form shown in Fig. 35.

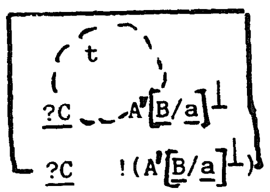


Fig. 35.

(vi) If A is $\bigwedge b.A'$, then $\mathrm{RED}(A[\boldsymbol{X}/\boldsymbol{a}])$ is defined as the orthogonal of the set Z formed of all terms of type $\bigwedge b.A'^{\perp}[\boldsymbol{B}/\boldsymbol{a}]$ of the form shown in Fig. 36 such that, for some CR Y of type C, $t \in \mathrm{RED}(A'^{\perp}[\boldsymbol{X}, Y/\boldsymbol{a}, b])$.

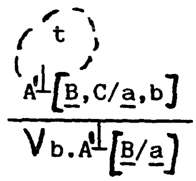


Fig. 36.

(vii) All other cases are treated by orthogonality, e.g., $\mathrm{RED}(!A[\boldsymbol{X}/\boldsymbol{a}]) = \mathrm{RED}(?A^{\perp}[\boldsymbol{X}/\boldsymbol{a}])^{\perp}$.

4.26.7. Lemma (Substitution Lemma)

$$\mathrm{RED}(A[C/b][\boldsymbol{X}/\boldsymbol{a}]) = \mathrm{RED}(A[\boldsymbol{X}, \mathrm{RED}(C[\boldsymbol{X}/\boldsymbol{a}])/\boldsymbol{a}, b]).$$

Proof. This lemma simply states that the definition of reducibility is compatible with substitution. This is established without difficulty once the formalism has been understood. Remark that there is here a hidden use of second-order comprehension (to define a *set* by $\mathrm{RED}(C[\boldsymbol{X}/\boldsymbol{a}])$. The proof of the Strong Normalization Theorem is not formalizable in PA_2 because of the use of the Substitution Lemma with unbounded C's. This must be familiar to people who know the original proof for F. □

4.26.8. Definition. A proof-net β with conclusions $\boldsymbol{C}$ is said to be *reducible* when the following holds: Let $\boldsymbol{a}$ be the list of all free variables of $\boldsymbol{C}$ and let us choose a sequence of formulas $\boldsymbol{B}$ and a sequence $\boldsymbol{X}$ of CR of types $\boldsymbol{B}$ (same length as $\boldsymbol{a}$); select terms $\boldsymbol{t}$ in the CR $\boldsymbol{C}^{\perp}[\boldsymbol{X}/\boldsymbol{a}]$ and make several cuts with all formulas of $\boldsymbol{C}[\boldsymbol{B}/\boldsymbol{a}]$ as shown in Fig. 37. The result of these operations, called $\mathrm{CUT}(\beta[\boldsymbol{B}/\boldsymbol{a}]; \boldsymbol{t})$, is SN.

4.26.9. Theorem. *All proof-nets of* PN2 *are reducible.*

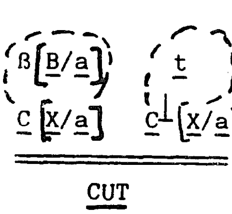


Fig. 37.

Proof. We argue by induction on a proof-net β. Since the substitutions $\boldsymbol{B}/\boldsymbol{a}$ and $\boldsymbol{X}/\boldsymbol{a}$ play no active role but make everything hard to read (and to write!), we shall not indicate them, working therefore with RED($C^{\perp}$), etc. The reader will be able to make the straightforward reconstruction himself.

Case 1: β is $\overline{A \qquad A^{\perp}}$; we have to show that, for any t, u in RED$(A)^{\perp}$, RED(A), CUT(β; t, u) is SN. But there is a standard reduction from CUT(β; t, u) to CUT(t, u) which is SN by the definition of orthogonality; by standardization, CUT(β; t, u) is SN.

Case 2: β is **1**; we have to show that CUT(β; t) $\in$ SN for all $t \in$ RED($\perp$). However RED($\perp$) $= \{\mathbf{1}\}^{\perp\perp}$; hence, CUT($\beta$; t) $=$ CUT(β, t) is SN.

Case 3: β is $\llcorner \top - C \lrcorner$; choose $t \in$ RED(**0**), $u \in$ RED($C^{\perp}$); here a *general remark* must be made: Since RED$(\mathbf{0})^{\perp} = Z^{\perp}$, where Z consists of the axiom-link $\overline{\top \qquad \mathbf{0}}$, it is possible to replace RED(**0**) by Z! Hence, t is the axiom-link. Now it is very easy, using the fact that u are SN, to produce a standard normalization for CUT(β; t, u).

Case 4: β comes from β' by the $\perp$-box in Fig. 38. Choose $t \in$ RED(**1**), choose $u \in$ RED($A^{\perp}$) and consider CUT(β; t, u). As we did in Case 3, one can restrict to $t = \mathbf{1}$, and there is a standard reduction from CUT(β; t, u) to CUT(β'; u) which is SN by hypothesis.

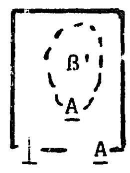


Fig. 38.

Case 5: β comes from β' by ($⅋$) (see Fig. 39). After simplification (this refers to the remark made in Case 3), we are led to form CUT(β; $t \otimes u$, v) with $t \in$ RED($A^{\perp}$), $u \in$ RED($B^{\perp}$), $v \in$ RED($C^{\perp}$). $t \otimes u$ indicates the term obtained from t and u by ($\otimes$).

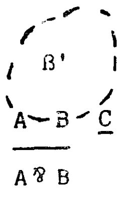


Fig. 39.

Now observe that there is a standard reduction sequence from CUT(β; $t\otimes u$, v) to CUT(β'; t, u, v), which is SN by the induction hypothesis.

Case 6: β is obtained from β' and β'' by ($\otimes$) (see Fig. 40). The induction hypothesis yields that CUT(β'; t', u') is SN for all $t'\in \mathrm{RED}(A')^{\perp}$ and $u'\in \mathrm{RED}(C')^{\perp}$, which means that CUT($\beta'$; u') $\in$ RED(A'). For symmetric reasons, CUT(β''; u'') $\in$

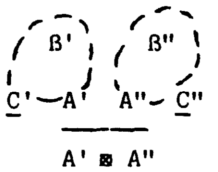


Fig. 40.

RED(A''). By tensorization,

$$\mathrm{CUT}(\beta;\, u',\, u'') \in \mathrm{RED}(A').\mathrm{RED}(A'') \subset \mathrm{RED}(A'\otimes A'')$$

by biorthogonality. But then CUT(β; u', u'', v) is SN for any $v\in \mathrm{RED}(A'\otimes A'')^{\perp}$.

Case 7: β is obtained from β' by ($1\oplus$) (see Fig. 41). The induction hypothesis yields CUT(β'; c) $\in$ RED(A) for any $c\in \mathrm{RED}(C)^{\perp}$; then CUT($\beta'$; c) $\in$ $\mathbb{L}^1\,\mathrm{RED}(A)\subset \mathrm{RED}(A\oplus B)$ and we are done.

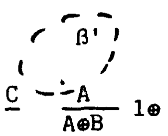


Fig. 41.

Case 8: β is obtained from β' by ($2\oplus$): As Case 7.

Case 9: β is obtained from β' and β'' by the &-box in Fig. 42. After simplification, we see that we have to check if CUT(β; c, $\mathbb{L}^1 t$) and CUT(β; c, $\mathbb{L}^2 u$) is SN for any $c\in \mathrm{RED}(C)^{\perp}$, $t\in \mathrm{RED}(A)^{\perp}$, and $u\in \mathrm{RED}(B)^{\perp}$. Now, the induction hypothesis yields that β'' is SN, and it is therefore possible to write a standard normalization from CUT(β; c, $\mathbb{L}^1 t$) to CUT(β'; c, t), which is SN by induction hypothesis. For symmetric reasons, CUT(β''; c, $\mathbb{L}^2 t$) is SN.

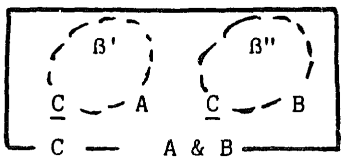


Fig. 42.

Case 10: β is obtained from β' by the !-box in Fig. 43. The induction hypothesis is that CUT(β'; c, t) is SN for all $c\in \mathrm{RED}(?C)^{\perp}$ and t in $\mathrm{RED}(A)^{\perp}$ and we want to conclude that CUT(β; c, u) is SN for all u in $\mathrm{RED}(!A)^{\perp}$. Now, it is easy to show that we can make a simplification on the whole sequence c, namely that c is of the

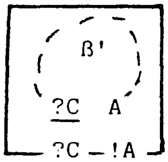


Fig. 43.

form $!d$, for $d \in \mathrm{RED}(C)^{\perp}$. But there is a standard reduction from $\mathrm{CUT}(\beta; c, u)$ to $\mathrm{CUT}(!\mathrm{CUT}(\beta'; c); u)$ (several (!-CC)). Now, the induction hypothesis says that $\mathrm{CUT}(\beta'; c) \in \mathrm{RED}(A)$; hence, $!\mathrm{CUT}(\beta'; c) \in !\mathrm{RED}(A) \subset \mathrm{RED}(!A)$.

Case 11: β is obtained from β' by the weakening box in Fig. 44. We must show that $\mathrm{CUT}(\beta; c, !t)$ is SN for all $c \in \mathrm{RED}(C)^{\perp}$ and $t \in \mathrm{RED}(A)^{\perp}$. Now t is SN, and there is a standard reduction from $\mathrm{CUT}(\beta; c, !t)$ to $\mathrm{CUT}(\beta'; c)$ plus weakenings, which is SN by induction hypothesis.

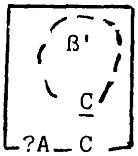


Fig. 44.

Case 12: β is obtained from β' by the dereliction rule shown in Fig. 45. We must show that $\mathrm{CUT}(\beta; c, !t)$ is SN; but it reduces in a standard way to $\mathrm{CUT}(\beta'; c, t)$, which is SN by induction hypothesis.

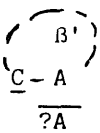


Fig. 45.

Case 13: β is obtained from β' by the contraction rule in Fig. 46. We must show that $\mathrm{CUT}(\beta; c, !t)$ is SN; but it reduces in a standard way to $\mathrm{CUT}(\beta; c, !t, !t)$ plus contractions, which is SN by induction hypothesis.

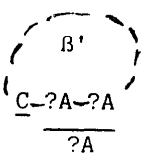


Fig. 46.

Case 14: β is obtained from β' by the $\bigwedge$-rule shown in Fig. 47. For the first time, we actually use the substitutions: we have to show that $\mathrm{CUT}(\beta; c, t)$ is SN for any $c \in \mathrm{RED}(C)^{\perp}$ and t coming from some u of some type $A^{\perp}[B/a]$ by the $\bigvee$-rule, s.t.

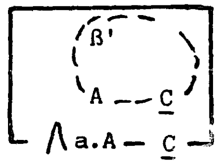


Fig. 47.

$u \in \text{RED}(A[X/a])^{\perp}$ for some CR X of type B. Now the induction hypothesis yields that $\text{CUT}(\beta'[B/a]; c, u)$ is SN, and observe the existence of a standard reduction from $\text{CUT}(\beta; c, t)$ to this proof-net.

Case 15: β is obtained from β' by the $\bigvee$-rule in Fig. 48. The induction hypothesis says that $\text{CUT}(\beta'; c) \in \text{RED}(A[B/a])$ for any c in $\text{RED}(C)^{\perp}$. Now, the Substitution Lemma says that $\text{RED}(A[B/a]) = \text{RED}(A[\text{RED}(B)/a])$; this shows that $\text{CUT}(\beta'; c) \in Z$ where Z is the set of existential terms such that $\text{RED}(\bigvee a.A) = Z^{\perp\perp}$; from $Z \subset Z^{\perp\perp}$, we conclude. □

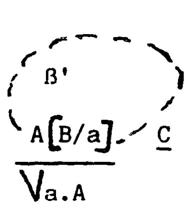


Fig. 48.

Proof of Theorem 4.26 (*conclusion*). Apply Theorem 4.26.9 to $B = X$ and t = axiom links. □

5. Some useful translations

5.1. The translation of intuitionistic logic

The intuitionistic symbols $\bigwedge, \bigvee, \Rightarrow, \neg, \forall, \exists$ can be considered as defined in linear logic. The definitions are as follows:

$$A \wedge B = A \,\&\, B, \qquad A \vee B = (!A) \oplus (!B),$$

$$A \Rightarrow B = (!A) \multimap B, \qquad \neg A = (!A) \multimap \mathbf{0},$$

$$\forall x A = \bigwedge x.A, \qquad \exists x A = \bigvee x.!A.$$

In particular, we see that we could have kept the symbols $\wedge$ and $\forall$ in linear logic, provided we have in mind *intuitionistic* conjunction and 'for all'. The translation$(.)^0$ from intuitionistic logic to linear logic is defined by means of the above dictionary, together with $A^0 = A$ when A is atomic.

The translation just given shows the unexpected heterogeneity of intuitionistic connectives (in sharp contrast with the translation into modal logic which is very regular, if nonconstructive). The most shocking point is the translation of negation:

one would have preferred $\bot$ instead of $\mathbf{0}$, i.e., $(!A)^{\bot}$ for $\neg A$. However, intuitionistic negation has always been widely criticized. On the other hand Andrej Ščedrov pointed out to me the formal analogy between these translations and Kleene's 'slash', in particular where the places are determined where a "!" is needed. Even with this analogy in mind, it is hard to find a leading principle in this translation; for instance, the translation does not follow the principle of restricting to open facts, as a coarse analogy with the modal translation would suggest.

The translation $(.)^0$ is extended to a translation of proofs. The most efficient one is a translation of intuitionistic natural deduction (with $\neg A$ defined as $A \Rightarrow \mathbf{0}$) into linear sequent calculus. Here we use a version of linear sequent calculus with formulas on both sides of "$\vdash$": $A \vdash B$ is short for $\vdash A^{\bot}, B$.[6] The translation is defined by induction on the length of a deduction (d) of $(A)B$, i.e., of B under the hypotheses (A): to such a (d) we associate $(d)^0$, a linear proof of $!A^0 \vdash B^0$.

(i) (d) is the deduction of $(B)B$ consisting of a hypothesis B; then $(d)^0$ is

$$\frac{B^0 \vdash B^0}{!B^0 \vdash B^0}\ell D!.$$

(ii) (d) is a deduction of $(A', A'')B' \wedge B''$ obtained from a deduction (d') of $(A')B'$ and a deduction (d'') of $(A'')B''$ by $\wedge$-introd; then $(d)^0$ is

$$\dfrac{\dfrac{\overset{(d')^0}{!A'^0 \vdash B'^0}}{!A'^0, !A''^0 \vdash B'^0}\text{several } \ell W! \qquad \dfrac{\overset{(d'')^0}{!A''^0 \vdash B''^0}}{!A'^0, !A''^0 \vdash B''^0}\text{several } \ell W!}{!A'^0, !A''^0 \vdash B'^0 \,\&\, B''^0}\,r\&.$$

(iii) (d) is a deduction of $(A)B'$ obtained from a deduction (d') of $(A)B' \wedge B''$ by the first $\wedge$-elim; then $(d)^0$ is

$$\dfrac{\overset{(d')^0}{!A^0 \vdash B'^0 \,\&\, B''^0} \qquad \dfrac{B'^0 \vdash B'^0}{B'^0 \,\&\, B''^0 \vdash B'^0}\ell 1\&}{!A^0 \vdash B'^0}\text{CUT}.$$

(iv) (d) ends with the second $\wedge$-elim: symmetric to (iii).

(v) (d) is a deduction of $(A)B' \vee B''$, obtained from (d') of $(A)B'$ by the first $\wedge$-introd; then $(d)^0$ is

$$\dfrac{\dfrac{\overset{(d')^0}{!A^0 \vdash B'^0}}{!A^0 \vdash !B'^0}r!}{!A^0 \vdash !B'^0 \oplus !B''^0}r1\oplus.$$

[6] The rules are now written as left and right rules; e.g., the rule (&) now becomes (r&) (on the right) and ($\ell\oplus$) (on the left) etc.

(vi) (d) ends with the second $\vee$-introd: symmetric to (v).

(vii) (d) is a deduction of $(A, \boldsymbol{B}', \boldsymbol{B}'')D$ obtained from (e) of $(A)C' \vee C''$ and (d') of $(\boldsymbol{B}', \boldsymbol{C}')D$ and (d'') of $(\boldsymbol{B}'', \boldsymbol{C}'')D$ by $\vee$-elim, where $\boldsymbol{C}'$ and $\boldsymbol{C}''$ are made of repetitions of C' and C'' respectively. Then $(d)^0$ is

$$\dfrac{\overset{(e)^0}{!A^0 \vdash !C'^0 \oplus !C''^0} \qquad \dfrac{\dfrac{\dfrac{\overset{(d')^0}{!\boldsymbol{B}'^0, !\boldsymbol{C}'^0 \vdash D^0}}{!\boldsymbol{B}'^0, !C'^0 \vdash D^0}\,\text{several } \ell C!, \text{ or one } \ell W!}{!\boldsymbol{B}'^0, !\boldsymbol{B}''^0, !C'^0 \vdash D^0}\,\text{several } \ell W! \qquad \dfrac{\dfrac{\overset{(d'')^0}{!\boldsymbol{B}''^0, !\boldsymbol{C}''^0 \vdash D^0}}{!\boldsymbol{B}''^0, !C''^0 \vdash D^0}\,\text{several } \ell C!, \text{ or one } \ell W!}{!\boldsymbol{B}'^0, !\boldsymbol{B}''^0, !C''^0 \vdash D^0}\,\text{several } \ell W!}{!\boldsymbol{B}'^0, !\boldsymbol{B}''^0, !C'^0 \oplus !C''^0 \vdash D^0}\,\ell\oplus}{!A^0, !\boldsymbol{B}'^0, !\boldsymbol{B}'' \vdash D^0}\,\text{CUT.}$$

(viii) (d) is a deduction of $(A)B \Rightarrow C$, obtained from a deduction (d') of $(A, \boldsymbol{B})C$ by $\Rightarrow$-introd, where $\boldsymbol{B}$ is made of repetitions of B; then $(d)^0$ is

$$\dfrac{\dfrac{\overset{(d')^0}{!A^0, !\boldsymbol{B}^0 \vdash C^0}}{!A^0, !B^0 \vdash C^0}\,\text{several } \ell C! \text{ or one } \ell W!}{!A^0 \vdash !B^0 \multimap C^0}\,r\multimap.$$

(ix) (d) is a deduction of $(\mathscr{A}', A'')B''$ obtained from (d') of $(A')B'$ and (d'') of $(A'')B' \Rightarrow B''$ by $\Rightarrow$-elim; then $(d))$ is

$$\dfrac{\overset{(d'')^0}{!A''^0 \vdash !B'^0 \multimap B''^0} \qquad \dfrac{\dfrac{\overset{(d')^0}{!A'^0 \vdash B'^0}}{!A'^0 \vdash !B'^0}\,r! \qquad B''^0 \vdash B''^0}{!A'^0, !B'^0 \multimap B''^0 \vdash B''^0}\,\ell\multimap}{!A'^0, !A''^0 \vdash B''^0}\,\text{CUT.}$$

(x) (d) is a deduction of $(A)B$ coming from a deduction of $(A)\mathbf{0}$ by the rule $\mathbf{0}$-elim (we have used the notation $\mathbf{0}$ for the absurdity of intuitionistic logic); then $(d)^0$ is

$$\dfrac{\overset{(d')_0}{!A^0 \vdash \mathbf{0}} \qquad \mathbf{0} \vdash B^0 \quad \text{(axiom)}}{!A^0 \vdash B^0}\,\text{CUT.}$$

(xi) (d) is a deduction of $(A)\forall x\, B$ coming from a deduction (d') of $(A)B$ by the rule $\forall$-introd; then $(d)^0$ is

$$\dfrac{\overset{(d')_0}{!A^0 \vdash B^0}}{!A^0 \vdash \bigwedge x. B^0}\,r\bigwedge.$$

(xii) (d) is a deduction of $(A)B[t/x]$ coming from a deduction (d') of $(A)\forall x.B$ by the rule $\forall$-elim; then $(d)^0$ is

$$\dfrac{\overset{(d')^0}{!A^0 \vdash \bigwedge x.B^0} \qquad \dfrac{B^0[t/x] \vdash B^0[t/x]}{\bigwedge x.B^0 \vdash B^0[t/x]}\,\ell\bigwedge}{!A^0 \vdash B^0[t/x]}\,\text{CUT}.$$

(xiii) (d) is a deduction of $(A)\exists x.B$ obtained from a deduction (d') of $(A)B[t/x]$ by the rule $\exists$-introd; then $(d)^0$ is

$$\dfrac{\overset{(d')^0}{!A^0 \vdash B^0[t/x]}}{!A^0 \vdash \bigvee x.B^0}\,r\bigvee.$$

(xiv) (d) is a deduction of $(A, B)D$ obtained from (e') of $(A)\exists x.C$ and (e'') of $(B, C)D$ by $\exists$-elim; then (d^0) is

$$\dfrac{!A \vdash \bigvee x!C^0 \qquad \dfrac{\dfrac{\overset{(e'')^0}{!B^0, !C^0 \vdash D^0}}{!B^0, !C^0 \vdash D^0}\,\text{several } \ell C! \text{ or one } \ell W!}{!B^0, \bigvee x!C^0 \vdash D^0}\,\ell\bigvee}{!A^0, !B^0 \vdash D^0}\,\text{CUT}.$$

Remark. If we start with a deduction which is normal (in the strongest sense, involving commutative conversions), then the construction of $(d)^0$ can be slightly modified, yielding a normal proof.

Observe that our translation is faithful in the sense that if A^0 is provable in linear logic, A is provable in intuitionistic logic. This can be easily justified as follows: Linear logic satisfies cut-elimination, hence a subformula property. A^0 is written in the fragment $\mathbf{0}, \multimap, \oplus, \&, !, \bigwedge, \bigvee$ of linear logic. In particular, writing the rules for this fragment with two-sided sequents we see that the proof of A^0 uses only intuitionistic linear sequents. Now, if we erase all symbols !, and replace $\oplus, \&, \multimap, \bigwedge, \bigvee$ by $\vee, \wedge, \Rightarrow, \forall, \exists$, then we get a proof of A in intuitionistic logic.

The translation just chosen has been historically spoken the first work on linear logic, dating back to the end of 1984 for disjunction, and October 1985 for the full language. The translation is sound, not only for provability, but also w.r.t. the coherent semantics. Once more, this is the coherent semantics of intuitionistic logic which suggested to put the hidden linear features on the front stage. Other translations are possible, for instance one following the idea of sticking to open facts:

$A^* = {!A}$ for A atomic,

$(A \wedge B)^* = A^* \otimes B^*$, $\qquad (A \vee B)^* = A^* \oplus B^*$,

$(A \Rightarrow B)^* = {!(A^* \multimap B^*)}$, $\qquad \mathbf{0}^* = \mathbf{0}$,

$(\forall x A)^* = {!\bigwedge A^*}$, $\qquad (\exists x A)^* = \bigvee x A^*$.

This boring translation is reminiscent of the modal translation of intuitionistic logic. This translation is not sound w.r.t. the coherent semantics and this is enough to show that its interest is limited.

5.2. *The translation of the system F*

F is nothing but a functional notation for second-order propositional natural deduction, so we can essentially apply the translation of Section 5.1. We concentrate here on the reduced version of F (variables, $\Rightarrow$, $\forall$), and we make a slightly more precise definition so that normal terms of F are directly translated as normal proof-nets.

5.2.1. *Types*

The types of F are translated as follows:

$$a^0 = a \quad \text{when } a \text{ is a variable,}$$

$$(S\Rightarrow T)^0 = !S^0 \multimap T^0, \qquad (\forall a.T)^0 = \bigwedge a.T^0.$$

The translation has the *substitution property*: $(S[T/a])^0 = S^0[T^0/a]$.

5.2.2. *Translation of terms*

We distinguish two kinds of terms:

(i) General terms of $t[x]$: x is a sequence of variables of types S, which are exactly the free variables of t which is of type T. Such a term will be translated into a proof-net t^0 whose conclusions will be exactly $!S^{0\perp}$ and T^0.

(ii) Terms with a distinguished headvariable y: $t[x, y]$ with x of type S, y of type H, t of type T. y is distinct from all variables in t. When we say that y is distinguished, this only means that we have decided to remark that we are in case (ii), instead of using the more general procedure case of (i). In that case, t^0 has the conclusions $?S^{0\perp}$, $H^{0\perp}$, T^0.

Step 1: $t = y$ (so that $T = H$); then t^0 is

$$\overline{T^{0\perp} \qquad T^0}.$$

Step 2: t has a headvariable and may be written as $u[y(v)/y']$, where $u[y']$ is already a term with a headvariable. The other variables of u and v are x and x', some of them being common. Then we form the configuration shown in Fig. 49 (y

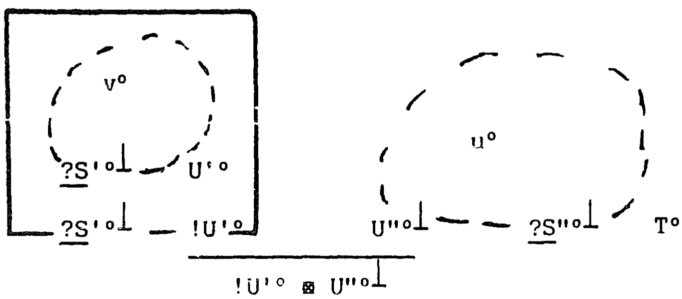


Fig. 49.

is of type $U' \Rightarrow U''$ so that we must get a conclusion $!U'^0 \otimes U''^{0\perp}$). From this, by using the rule

$$\frac{?S_i'^{0\perp} \qquad ?S_j''^{0\perp}}{?S_i'^{0\perp}}\,?C$$

every time $x_i' = x_j''$, we obtain the interpretation t^0.

Step 3: t has a headvariable and may be written as $u[u\{U'\}/y']$ for a certain term $u[y']$ (which already has a headvariable y' of type $U''[U'/a]$, y being of type $\bigwedge a.U''$). We form the configuration in Fig. 50.

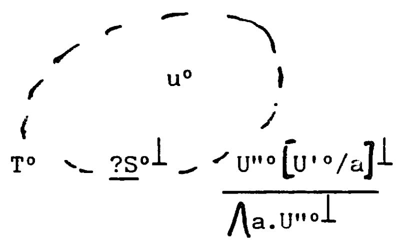


Fig. 50.

Step 4: t has a headvariable, but we do not want to distinguish it any longer. Consider the term $u[y, x]$ with a new distinguished headvariable such that t is $u[x_i, x]$; then form the configuration in Fig. 51. This will be t^0, except if x_i is equal to some variable of x, in which case one must end with a contraction.

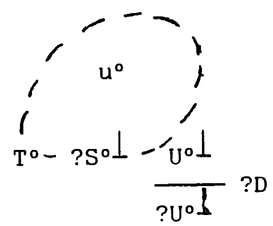


Fig. 51.

Note: All other steps are steps involving no terms with headvariables.

Step 5: t is $u(v)$; u is of type $T' \Rightarrow T''$, v is of type T': we now form the configuration in Fig. 52 to which we apply, if necessary, a certain number of contractions between $?S_i'^{0\perp}$ and $?S_j''^{0\perp}$ when $x_i' = x_j''$.

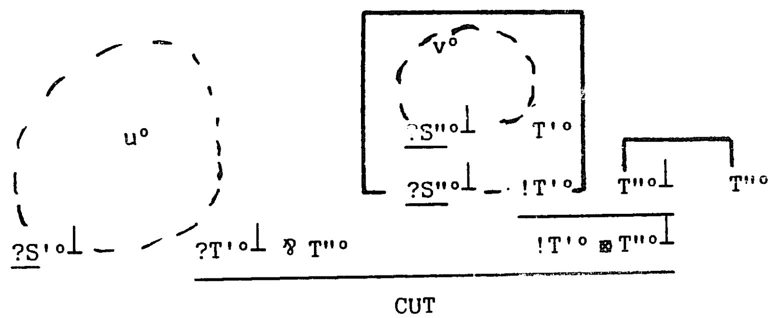


Fig. 52.

Step 6: t is $u\{U\}$ with u of type $\forall a.T'$; the corresponding configuration is shown in Fig. 53.

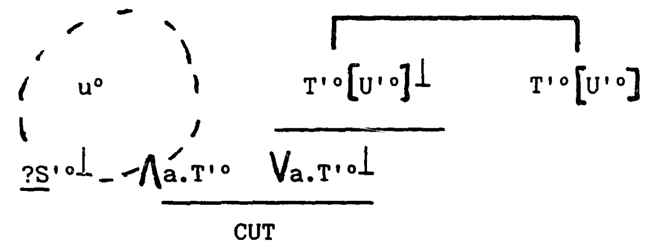


Fig. 53.

Step 7: t is $\lambda x.u$, with u, x of types T'', T'. Let u' be u^0 if x occurs in u and otherwise, let u' be the result of a $W?$ on $?T'^{0\perp}$ applied to u^0 (see Fig. 54).

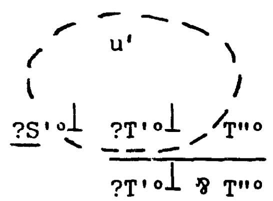


Fig. 54.

Step 8: t is $\forall a.u$, with u of type T', the configuration is given in Fig. 55.

The steps must now be put together to form a correct definition by induction of the translation $(\ )^0$. The details are left to the reader, but, by correctly handling the case with headvariables, one translates a normal term of F into a cut-free proof-net of PN2.

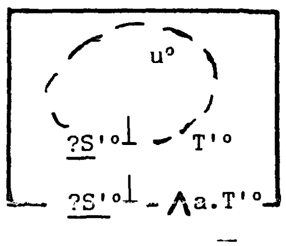


Fig. 55.

5.2.3. Semantic soundness of the translation

The coherent semantics of F has been given in [5]; we also have a coherent semantics for PN2 (Section 3); the semantic soundness of the translation is that $t^{0*} = t^*$ if the symbol * denotes both semantic interpretations.

(1) The first thing to check is that there is a semantic soundness of the translation of types, namely that $T^{0*} = T^*$. This easily follows from two remarks:

(i) *the type constructor* $\forall$ *is exactly the same thing as* $\bigwedge$.

(ii) *semantically speaking,* $X \Rightarrow Y = !X \multimap Y$.

Proof: $X \Rightarrow Y$ is the set of traces of stable functions from X to Y. This coherent space is defined by

$$|X \Rightarrow Y| = X_{\text{fin}} \times |Y|$$

where X_{fin} is the set of all finite objects of X.

$$(a, z) \bigcirc (b, t) \,[\text{mod } X \Rightarrow Y] \text{ iff } (1)\ a \cup b \in X \rightarrow z \bigcirc t \,[\text{mod } Y],$$
$$(2)\ a \cup b \in X \text{ and } a \neq b \rightarrow z \neq t.$$

Now, observe $!X$; we have $|!X| = X_{\text{fin}}$; moreover, $a \bigcirc b \,[\text{mod } !X]$ iff $a \cup b \in X$; in other terms, $|X \Rightarrow Y| = |!X| \times |Y|$ and

$$(a, z) \bigcirc (b, t) \,[\text{mod } X \Rightarrow Y] \text{ iff } (1)\ a \bigcirc b \,[\text{mod } !x] \rightarrow z \bigcirc t \,[\text{mod } Y],$$
$$(2)\ a \frown b \,[\text{mod } !X] \rightarrow z \frown t \,[\text{mod } Y].$$

This clearly established that $X \Rightarrow Y = !X \multimap Y$ (*end of proof*).

(2) The semantic soundness of the interpretation of term is based on the ideas already introduced in the *pons asinorum* (Section IV); checking all the cases here would be a pure loss of time.

5.2.4. Syntactic soundness of the translation

This is another issue, albeit of slightly less importance than the semantic soundness. The semantic soundness is enough to ensure that there is no loss of expressive power between F and PN2. (There is no essential gain either: the Normalization Theorem for PN2 can be carried out in PA_2). However, a syntactic relation between normalization in F and PN2 would be welcome. In fact, when $t =/ u$ in F, one can find a u' such that $t^0 =/ u'$ in PN^2 and u' differs from u^0 by a different order of use of the rules (C?). This problem is due to the fact that, in F (and systems based on λ-abstraction), the identification of variables is indeed made just before the abstraction, while PN2 is more refined and gives a specific order for the identification. This is a quality of PN2: a proof-net of PN2 contains additional information which the theory usually ignores (the order of the contractions), but which is very relevant in practice (e.g., for retaining the substitution as long as possible) and so traditionally belongs to the sphere of '*bricolage*' (i.e., handicraft).

The verification of the syntactic soundness of the translation is boring and without surprise.

5.3. Translation of current data in PN2

Since we know that current data types can be translated in F and that F translates in PN2, we can, by transitivity, translate integers, booleans, trees, lists, etc. into PN2. However, PN2 has subtler handling of the types and we often find a better translation, not transiting through F as will be shown in the following subsections.

5.3.1. Booleans

The best is to choose the definition **bool** $= \mathbf{1} \oplus \mathbf{1}$, with

$$\text{TRUE} = \frac{\mathbf{1}}{\mathbf{bool}} 1\oplus, \qquad \text{FALSE} = \frac{\mathbf{1}}{\mathbf{bool}} 2\oplus.$$

The instruction IF β THEN β' ELSE β'' is defined by the configuration in Fig. 56, where β and β' have the same conclusions C. It is immediate that IF TRUE THEN β' ELSE $\beta'' =/ \beta'$ and IF FALSE THEN β' ELSE $\beta'' =/ \beta''$. There are however two strategies possible for normalizing such instructions.

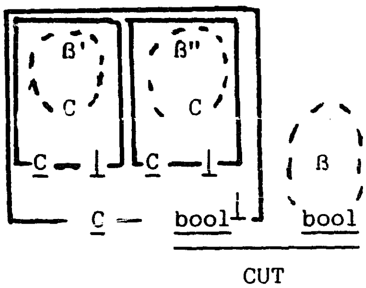


Fig. 56.

(i) The *lazy strategy* consists in not using the commutative conversions (&-CC). The IF-THEN-ELSE instruction is viewed as a sealed box from the viewpoint of β' and β''. The only way to get the box opened is to wait until the main door is opened (cut on TRUE or FALSE). This strategy is particularly interesting in the case we are normalizing a procf-net whose conclusions involve neither $\oplus$ nor $\bigvee$ since we are sure that all such boxes will eventually be opened by the main door. In terms of parallelism, such a box can be viewed as a moment when one has to wait (the communication through C is temporarily interrupted) (and maybe this waiting time can be used to do other tasks).

(ii) The *general strategy* consists in entering the boxes even by the auxiliary doors; this causes a duplication of certain tasks, with the unpleasant feature that if the main door is eventually opened, half of what has been done may be erased. But if we have indications that the main door may never open, this strategy may be of some use.

5.3.2. Integers

In F integers are translated as $\forall a.a \Rightarrow (a \Rightarrow a) \Rightarrow a$. The straightforward translation of F into PN2 would yield three "!"s, including a nested one! It turns out that, among the three implications used in this type, the first two can be taken as linear. We therefore define **int** as $\bigwedge a.a \multimap ((a \multimap a) \Rightarrow a)$. Equivalently, **int** $= \bigwedge a.a \multimap (!(a \multimap a) \multimap a)$, and this is isomorphic to $\bigwedge a.!(a \multimap a) \multimap (a \multimap a)$. For instance, the number "3" can be represented in F (natural deduction) as shown in Fig. 57(a). In PN2, the representation of "3" looks as shown in Fig. 57(b).

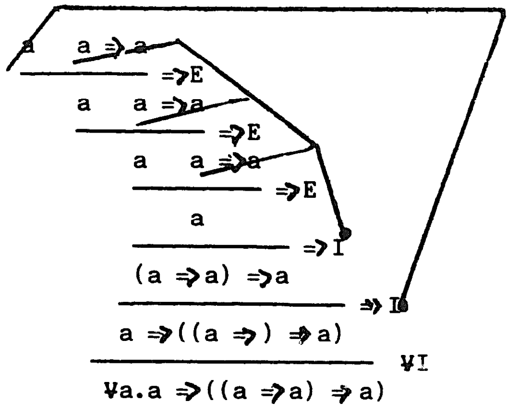


Fig. 57(a). The number “3” represented in the system F (natural deduction).

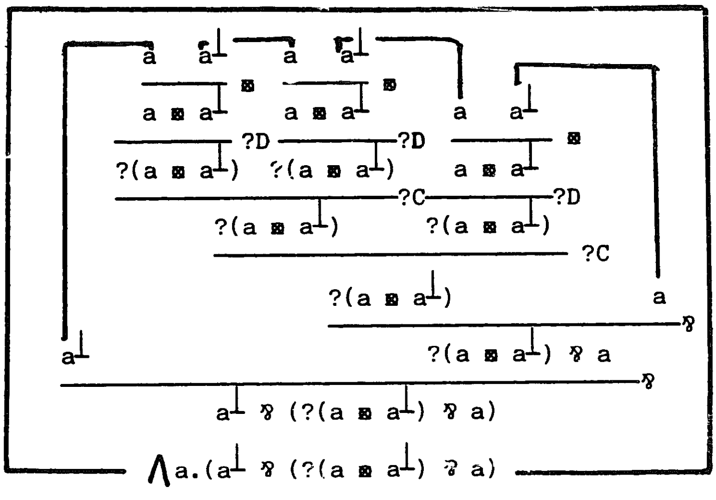


Fig. 57(b). The number “3” represented in PN2.

The functional notation for “3” in F is $\forall a.\lambda x^{a}.\lambda y^{a\Rightarrow a}.y(y(a)))$. There are other possible representations of “3” in PN2: before the first ⅋-rule, we can use a different combination of the rules $(C?)$ and maybe some rules $(W?)$.

Consider all possible proof-nets with **int** as conclusion; the end is necessarily as in the described example a $\bigwedge$-box, then a ⅋-link whose premises are $a^{\perp}$ and $?(a\otimes a^{\perp})$ ⅋ a, which in turn follows from a ⅋-link applied to a and $?(a\otimes a^{\perp})$. Now, there are several possibilities: this $?(a\otimes a^{\perp})$ may arise by several combinations of $(W?)$, $(C?)$, and $(D?)$. Above the combination of these rules there are several premises $a\otimes a^{\perp}$, and above them premises a, $a^{\perp}$. These premises are linked together by axiom-links, and are also linked to the formulas a and $a^{\perp}$ mentioned much earlier. The only possible configuration, in terms of proof-nets is the one shown in Fig. 58 and the number of axiom-links determines which integer we are speaking of: $n+1$ axiom-links are needed to represent n. For instance, Fig. 57(b) was using four axiom-links, and therefore represents “3”. But any other variation on this picture (with extra $(W?)$, different order for the rules $(C?)$) would yield a term with the same semantics.

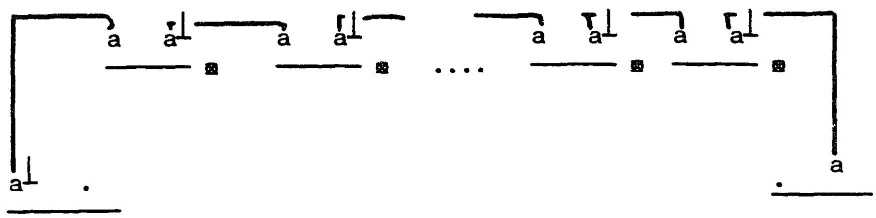


Fig. 58.

Let us explain the interpretation of the proof-net "3"; for this, the best is to return to the *pons asinorum* (Section IV). Specify a coherent space X in order to interpret a in the inner part of the box. Now, $X^{\perp} ⅋ (?(X \otimes X^{\perp}) ⅋ X)$ means $X \multimap (!X \multimap X) \multimap X)$, and should be viewed as the set of all linear maps from X to $!(X \multimap X) \multimap X$. Then, $!(X \multimap X) \multimap X$ is $(X \multimap X) \Rightarrow X$ and should be seen as the set of all stable maps from $X \multimap X$ to X. Summing up, what is inside the box describes, for $a = X$, a binary stable function mapping X, $X \multimap X$ into X, and linear in the first argument. This function is defined by $F(x, f) = f(f(f(x)))$, as the semantic computation easily shows. More generally, all proof-nets which represent n are semantically identical: on X, they correspond to the function $F(x, f) = f^n(x)$.

This is very close to the more familiar representation of integers in F; the main difference is that the argument f is linear. It is possible to stick to 'f linear' because f^n is still linear. But the function associating f^n with f is never linear, but for $n = 1$. Now, look at Fig. 59; the diagram represents the successor function, which linearly

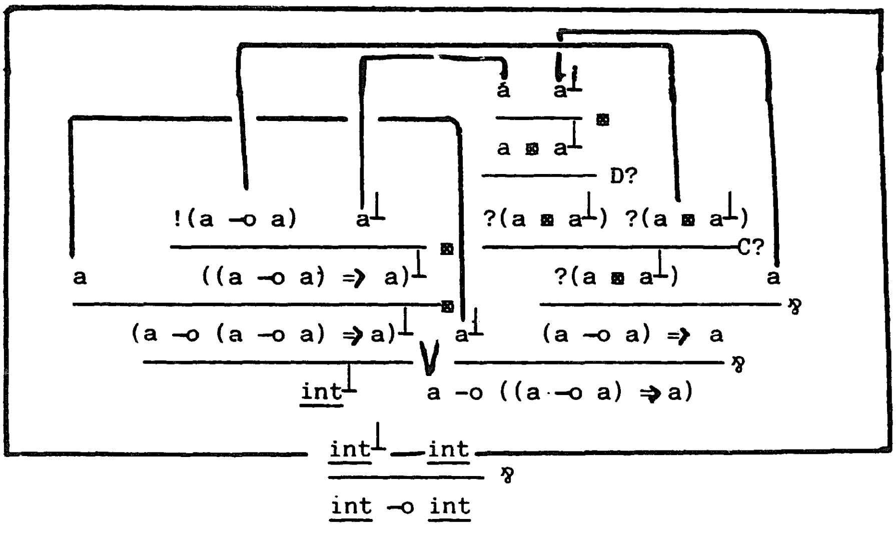


Fig. 59.

maps **int** into **int**; if one applies SUC to a repressentation $\bar{n}$ of the integer n, by means of the configuration in Fig. 60, then, after normalization, one gets a representation $\overline{n+1}$ of the next integer.

As usual, **int** allows definitions by primitive recursion: typically, from a proof-net β with A as conclusion and from β' with $A \multimap A$ as conclusion, one can form

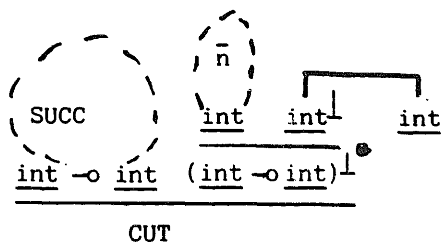


Fig. 60.

REC(β, β') with the conclusion **int** ⊸ A (see Fig. 61). (In fact, there is no need that A and $A \multimap A$ should be the unique conclusions of β or β'; arbitrary other conclusions are welcome in β and additional ?-conclusions can be accepted in β'.)

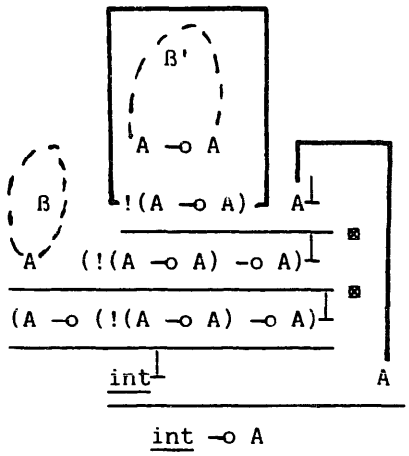


Fig. 61.

When we apply REC(β, β') to $\bar{n}$, by the configuration in Fig. 62(a), this reduces to the configuration in Fig. 62(b) (n occurrences of β'; this proof-net is "β' applied n times to β").

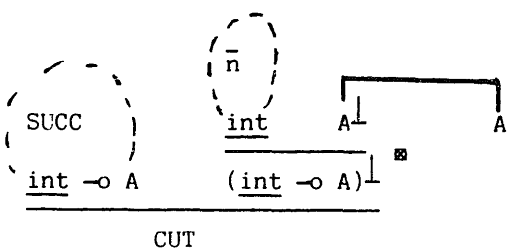


Fig. 62(a).

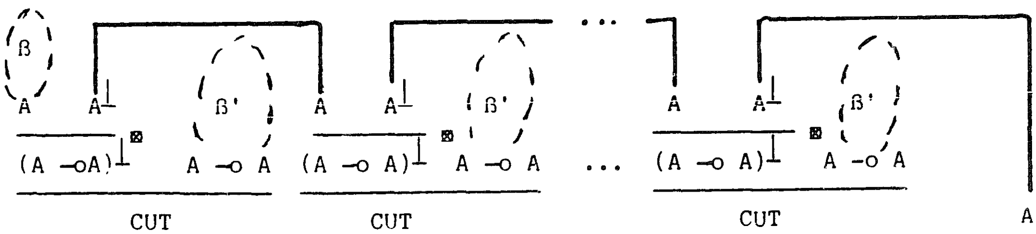


Fig. 62(b). "β' applied n times to β".

The problem is that, in practice, we do not necessarily have linear functions to start with when making recursion. However, what about a recursion with β as a proof of A, β' as a proof of $A \Rightarrow A$, i.e., $!A \multimap A$? This can be reduced to the previous case: from the configuration in Fig. 63 (here, auxiliary conclusions can be accepted

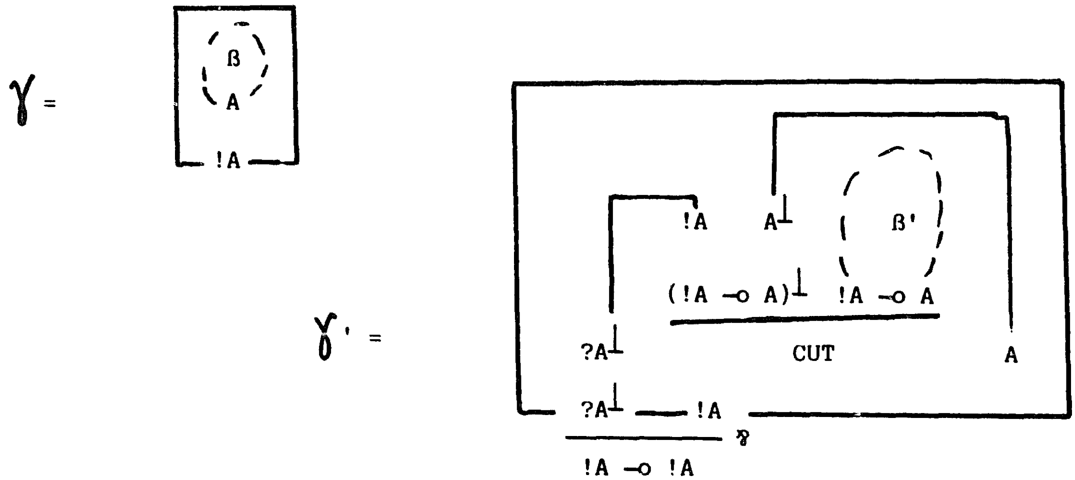


Fig. 63.

too, but only of the ?-form). Then, what we want can be expressed by derelicting REC(γ, γ') as shown in Fig. 64.

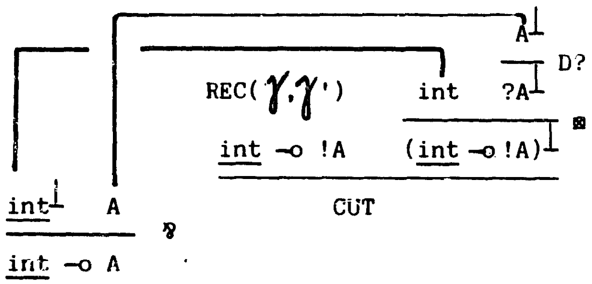


Fig. 64.

Exercise. (i) Show the existence of a proof-net whose conclusion is **int** ⊸ **!int** and such that when applied to an integer $\bar{n}$, it yields $!\bar{n}$; i.e., it is put into a !-box.

(ii) Conclude that the same functions from ℕ to ℕ are representable in PN2 as of type **int** ⊸ **int** and of type **int**⇒**int**.

5.3.3. Lists, trees

We can take more or less the usual translations while being careful to use linear implications everytime ⇒ is not actually needed. The details would be too much here.

5.4. Interpretation of classical logic

Of course, there is the ¬¬-translation of Gödel, which reduces the problem to the one already solved in Section 5.1. However, the ¬¬-translation has a terrible defect, namely using intuitionistic negation which is the only citicizable intuitionistic connective. But Gödel's translation is still working if, instead of $\neg\neg A$, one uses $((A\Rightarrow F)\Rightarrow F)$ where F is any formula which is not provable. Here we shall use $\perp$ for F. Formulas of the form $((A\Rightarrow\perp)\Rightarrow\perp)$ are equivalent to $?!A$; i.e., in topolinear terms, they are closed facts, equal to the closure of their interior. Let us call a fact ***regular*** when it is equal to the closure of its interior, i.e., to the closure of some closed fact. We have the following properties.

(i) If A is regular, then $?(A^{\perp})$ is regular.

(ii) If A and B are regular, then $A ⅋ B$ is regular: $A ⅋ B = ?!A ⅋ ?!B = ?(!A \oplus !B)$ where $!A \oplus !B$ is open.

From this, the principles of the translation are immediate:

$$A^+ = ?!A \quad \text{when } A \text{ is atomic}$$

$$(A \vee B)^+ = A^+ ⅋ B^+, \qquad (\neg A)^+ = ?(A^{+\perp}), \qquad (\forall x A)^{\perp} = ?!\bigwedge x.A^+.$$

The interpretation of classical disjunction is particularly simple. This justifies our claim that par *is the constructive contents of classical disjunction.*

Now, the regular subsets of a topolinear space form a complete boolean algebra and from this, it will be possible to justify all classical laws translated by $(\,)^+$. In particular, to some extent, we get a semantics of proofs for classical logic. But the translation used lacks the purity obtained in the intuitionistic case. This is due to the fact that the equivalence $?!A^+ \multimap\!\!\circ\, A^+$ is used too often.

5.5. *Translation of cut-free classical logic*

If we turn our attention towards cut-free classical logic, the situation changes radically, in the sense that a convincing interpretation inside linear logic is possible. The interpretation is based on the familiar three-valued idea of giving independent meanings to positive and negative occurrences of formulas (see [3]).

By induction on the formula A, we define pA and nA:

$$pA = nA = A \quad \text{when } A \text{ is atomic}$$

$$p\neg A = (nA)^{\perp}, \qquad n\neg A = (pA)^{\perp},$$

$$pA \vee B = (pA) \oplus (pB), \qquad nA \vee B = !(nA) ⅋ !(nB),$$

$$pA \wedge B = ?(pA) \otimes ?(pB), \qquad nA \wedge B = (nA) \& (nB),$$

$$pA \Rightarrow B = (nA)^{\perp} \oplus (pB), \qquad nA \Rightarrow B = ?(pA) \multimap !(nB),$$

$$p\forall x A = \bigwedge x.?(pA), \qquad n\forall x A = \bigwedge x.nA,$$

$$p\exists x A = \bigvee x.pA, \qquad n\exists x A = \bigvee x.!nA.$$

The reader will toy a little with the definition to see that this definition is built on strong symmetries, unlike the less satisfactory translation in Section 5.4. In fact, the situation is the same between the very good translation $(\ )^0$ and the less satisfactory translation $(\ )^*$, both considered in Section 5.1. But here we have to give up the cut-rule.

Now, with each proof in cut-free classical logic of a sequent $A \vdash B$, one associates a proof of the sequent $!nA \vdash ?pB$ in linear logic in a more or less straightforward way. This paper has already been too long, and a reader not already dead at this point will have no trouble in finding out the details. In particular, due the cut-elimination theorem, *there is a very nice and natural semantics of proofs of full classical logic.*

The cut-rule has a very ambiguous status: it can be seen as the general scheme: $?pA \multimap\, !nA$, i.e., $n(A \Rightarrow A)$. This scheme, although wrong, is conservative over sequents of the form $!nB \vdash ?pC$, i.e., over formulas of the form $?pA$, whose main feature seems to be the absence of "!".

Conjecture and question. It should be possible to find a formulation of cut-elimination as a general conservativity result over a fragment of linear logic, e.g., the fragment free from !. The cut-elimination procedure would appear as the effective way of eliminating the conservative principles (which should be the formulas $n(A \Rightarrow A)$ or something more general). *Such a program, if properly carried out, would give a full constructive content to classical logic.*

5.6. The Approximation Theorem

In the introduction we have already mentioned that linear logic was able to control the length of disjunctions in Herbrand's theorem. Let us now use what we know, in particular, the translation of classical logic just given. To simplify the matter, we shall work with a prenex formula $\exists x \forall y \exists z \forall t\ Rxyzt$ with R quantifier-free. The p-translation of such a formula is $\bigvee x \bigwedge y ?\bigvee z \bigwedge t\ ?Sxyzt$ with S quantifier-free and !-free.

Now, if our formula is provable in classical logic, the ? of its p-translation, i.e., $?\bigvee x \bigwedge y ?\bigvee z \bigwedge t\ ?Sxyzt$ will be provable in linear logic.

Now, consider the connectives $?_n$ defined by

$$?_n A = (\bot \oplus A) \parr (\bot \oplus A) \parr \cdots \parr (\bot \oplus A) \quad (n \text{ times}).$$

The Approximation Theorem (see below) says that we can replace each ? by approximants $?_n$; in particular, $?_n \bigvee x \bigwedge y ?_m \bigvee z \bigwedge t\ ?Sxyzt$ (in the quantifier-free part, we have kept ?). This formula clearly expresses that we had a midsequent with at most $n \cdot m$ formulas or, equivalently, a Herbrand disjunction of length at most $n \cdot m$. It also says something about the intimate structure of this Herbrand disjunction (the n and the m).

This example clearly shows how linear logic can be used to control the length (and also) the structure of Herbrand disjunctions. Now, the approximation theorem we were speaking about is just the mathematical contents of our slogan: *usual logic is obtained from linear logic* (without modalities) *by a passage to limit.*

5.1. Definition (*approximants*). The connectives ! and ? are approximated by the connectives $!_n$ and $?_n$ $(n \neq 0)$:

$$!_n A = (1 \mathbin{\&} A) \otimes \cdots \otimes (1 \mathbin{\&} A) \quad (n \text{ times}),$$

$$?_n A = (\bot \oplus A) \mathbin{⅋} \cdots \mathbin{⅋} (\bot \oplus A) \quad (n \text{ times}).$$

5.2. Theorem (Approximation Theorem). *Let A be a theorem of linear logic; with each occurrence of ! in A, assign an integer $\neq 0$; then it is possible to assign integers $\neq 0$ to all occurrences of ? in such a way that if B denotes the result of replacing each occurrence of ! (respectively ?) by $!_n$ (respectively $?_n$) where n is the integer assigned to it, then B is still a theorem of linear logic.*

Proof. One can formulate the exact analogue for a sequent $\vdash A$ and we prove the result by induction on a cut-free proof of $\vdash A$ in linear sequent calculus. We content ourselves with a few interesting cases:

(i) One can assume that the axioms $\vdash A, A^\perp$ are restricted to the atomic case and so, no problem.

(ii) If the last rule is (!): from $\vdash A, ?\boldsymbol{B}$ infer $\vdash !A, ?\boldsymbol{B}$, then we already obtained $\vdash A', ?_{\boldsymbol{n}}\boldsymbol{B}'$ (approximation A' of A, $\boldsymbol{B}'$ of $\boldsymbol{B}$, and a sequence $\boldsymbol{n}$). Now, one easily gets $\vdash 1, ?_{\boldsymbol{n}}\boldsymbol{B}'$ and so, $\vdash 1 \mathbin{\&} A', ?_{\boldsymbol{n}}\boldsymbol{B}'$. Let k be the integer associated with the first ! in $!A$; we get $\vdash !_k A', ?_{\boldsymbol{n}k}\boldsymbol{B}'$.

(iii) If the last rule is $(D?)$: from $\vdash ?A, ?A, \boldsymbol{B}$ infer $\vdash ?A, \boldsymbol{B}$ and if we have already approximations $\vdash ?_n A', ?_m A'', \boldsymbol{B}$, we can first ensure $A' = A''$ by increasing the respective ?-assignments. Then let A''' be the result; we then form $\vdash ?_{n+m} A''', \boldsymbol{B}'$.

(iv) If the last rule is (&): from $\vdash A, \boldsymbol{C}$ and $\vdash B, \boldsymbol{C}$ infer $\vdash A \mathbin{\&} B, \boldsymbol{C}$, then we have approximants $\vdash A', \boldsymbol{C}'$ and $\vdash B', \boldsymbol{C}''$; by increasing the ?-assignments in $\boldsymbol{C}'$ and $\boldsymbol{C}''$, we get $\boldsymbol{C}'''$ and $\vdash A' \mathbin{\&} B', \boldsymbol{C}'''$ is still provable.

(v) We have gone through the main steps: The rule $(D?)$ would involve a $?_1$ and the rule $(W?)$ a $?_0$ (which we have excluded, so a $?_1$, too).

The crucial fact that one can always replace $?_n$ by $?_m$ when $n < m$ is more or less immediate. In a similar way, $!_m$ can be replaced by $!_n$ when $n < m$. □

6. Work in progress: slices

There are several directions of work in progress, e.g., the theory of totality (solved by saying that an element of a coherent space X is total with the phase p), which have not reached maturity and which have therefore been omitted from this paper, which is already a bit long. However, the presentation of *slices* has been finally decided, for two reasons:

(i) this is the only immediate approach we know to the boring standardization theorem;

(ii) it presents the absolute limit for a parallelization of the syntax, i.e., the removal of all boxes but !-ones.

The present stage of the theory is a list of definitions with no theorem....

6.1. Definition. A *slice* is a proof-structure making use of (1) all links and axioms already introduced, (2) !-boxes (the contents of such boxes are sets of slices all ending with $?B, A$), and (3) unary rules:

$$\frac{A}{\bigwedge a.A}\bigwedge, \quad \frac{A}{A \,\&\, B}1\&, \quad \frac{B}{A \,\&\, B}2\&.$$

6.2. Definition. The *slicing* of a proof-net β is a nonempty set of slices with the same conclusions as β.

(1) *Slicing of a box*: the !-box built from β slices into the set consisting of just one !-box; the contents of this box is the slicing of β. Boxes as shown in Fig. 65(a) slice into the disconnected structures of Fig. 65(b), where the β_i's are the slices of

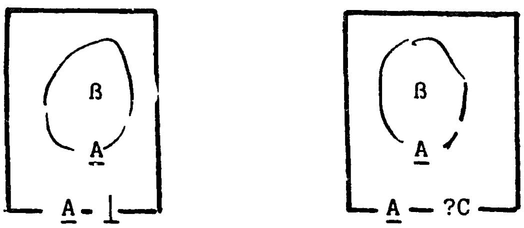


Fig. 65(a).

Fig. 65(b).

β. A box of the type shown in Fig. 66(a) slices into the structures of Fig. 66(b), where the β_i's and β_j''s are the respective slices of β and β'. Finally, a box as in

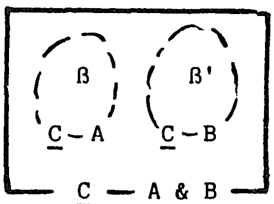


Fig. 66(a).

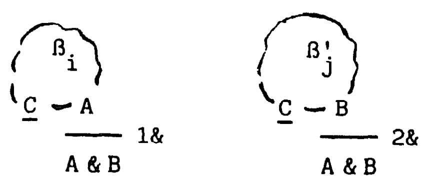


Fig. 66(b).

Fig. 67(a) slices into the structures of Fig. 67(b) where the β_i's are the slices of β.

An axiom └T—C┘ (seen as a box) slices into the disconnected structure T C.

(2) If β is a proof-net built from boxes $\mathbb{B}_1, \ldots, \mathbb{B}_n$ by means of links, say $\beta(\mathbb{B}_1, \ldots, \mathbb{B}_n)$, then the slices of β are all $\beta(\beta_1^{i_1}, \ldots, \beta_n^{i_n})$ for all slices $\beta_1^{i_1}, \ldots, \beta_n^{i_n}$ of $\mathbb{B}_1, \ldots, \mathbb{B}_n$ respectively.

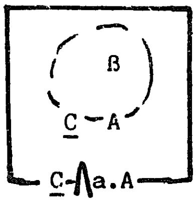


Fig. 67(a).

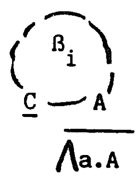


Fig. 67(b).

6.3. Remarks. (i) The slicing is the 'development' of a proof-net by 'distributivity'. Each slice is in itself logically incorrect, but it is expected that the total family of slices has a logical meaning. However, there is no characterization of sets of slices which are the slices of a proof-net. Moreover, we do not even know whether or not identifying proof-nets with the same slices could lead to logical atrocities; for instance, is it possible to define the coherent semantics of a set of slices?

(ii) The slicing does not work for !-boxes; this is because the modalities ! and ? are the only nonlinear operations of linear logic. However, if we were working with ! as an infinite tensorization $\bigotimes \mathbf{1} \mathbin{\&} A$, then it would be possible to slice, but we would get a nondenumerable family of slices! A more reasonable approach would be to make finite developments based on $!A = (\mathbf{1} \mathbin{\&} A) \otimes !A$ so that we never go to the ultimate, nondenumerable slicing, but generate it continuously. This idea is of interest because it could serve, without changing anything essential to PN2, for expressing nonterminating processes.

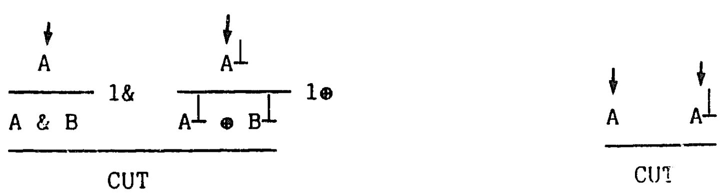
(iii) We shall now try to define the normalization procedure on the slices directly; however, the rule (T-CC), which involves the erasing of a ghost-box, is difficult to handle in those terms and this rule is therefore not considered in what we are doing. Strictly speaking, we shall get a proof of standardization without this rule, but there is so little to add to take care of this rule.... The fact that we are forced to make '*bricolage*' on such details is an illustration of the difficulties that arise from the absence of a real understanding of slices.

6.4. Definition (*contraction of a slice*). A slice β contracts into a set β' of slices (very often consisting of one slice) as follows:

(1) (A): Axiom-contraction, like (AC).

(2) (1&/1⊕): Replace a configuration as in Fig. 68(a) by the one in Fig. 68(b).

(3) (2&/2⊕): Symmetric to (2).



(a) (b)

Fig. 68.

(4) ($1\&/2\oplus$): If in β a configuration occurs as in Fig. 69, then β **contr** $\emptyset$ (the void set of slices).

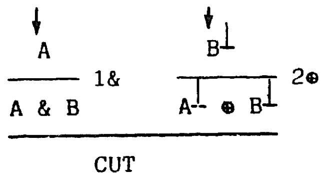


Fig. 69.

(5) ($2\&/1\oplus$): Symmetric to (4).
(6) ($\otimes/$): As ($\otimes/$-SC).
(7) ($1/\bot$) a configuration

$$\frac{\mathbf{1} \quad \bot}{\text{CUT}} \quad \text{(no premises above } \mathbf{1} \text{ or } \bot\text{)}$$

in β is simply cancelled.

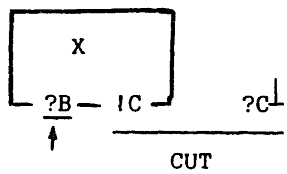
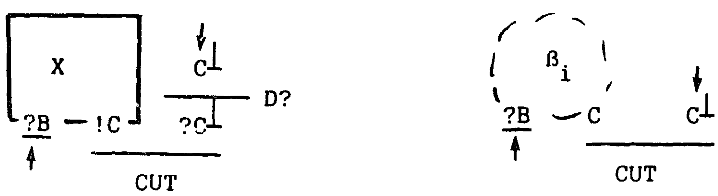


Fig. 70.

(8) ($!/W?$) a configuration as in Fig. 70 (no premise above $?C$) is replaced by

$$\underset{\uparrow}{?B}.$$

(9) ($!/D?$): If β contains the configuration as in Fig. 71(a) and X is made of slices β_i (with conclusions $?B, A$), then β contracts to the set made of the slices in Fig. 71(b).



(a) (b)

Fig. 71.

(10) ($!/C?$): As ($!/C?$-SC).
(11) ($\bigwedge/\bigvee$): If in β occurs

$$\frac{\dfrac{\overset{\downarrow}{A[a]}}{\bigwedge a.A} \qquad \dfrac{\overset{\downarrow}{A^{\bot}[B]}}{\bigvee a.A^{\bot}}}{\text{CUT}},$$

then one contracts it by replacing a by B everywhere, and then by replacing the pattern by the cut

$$\frac{\overset{\downarrow}{A[B]} \qquad \overset{\downarrow}{A^{\perp}[B]}}{\text{CUT}}.$$

(12) (!): Adaptation of (!-CC) to slices, details omitted.

6.5. Remark (*Church-Rosser property*). Using Definition 6.4, one gets a notion of reduction for sets of slices. This concept is Church-Rosser, as one can easily check.

6.6. Remark (*standardization property*). If, in the definition of contraction, we restrict cases (3) and (4) to the case where β cannot be normalized further, and case (8) to the case where the elements of X cannot be normalized further, we obtain the notion of standard contraction from which we derive the notion of standard reduction. The standardization property says that if a set of slices has a standard normalization into a set of slices which cannot be normalized further, then all normalization sequences starting from it are finite. This is easy to prove from the Church-Rosser property and the fact that the erasure is well-controlled in the standard case.

6.7. Remark (*standardization of proof-nets*). If $\beta =/ \beta'$, then $\mathrm{sl}(\beta) =/ \mathrm{sl}(\beta')$, where sl(.) stands for the slicing (provided ($\top$-CC) has not been used). This lies in the fact that the commutation rules (&-CC), (-CC), ($\perp$-CC), (W?-CC) do nothing on the slicings. Now, this fact can easily be used to transfer standardization to proof-nets. One must work a little more to consider ($\top$-CC) but there are no real problems.

V. Two years of linear logic: selection from the garbage collector

This short historical note is here to explain the successive states of linear logic and, in particular, to mention possibilities that have been discarded for reasons that may seem excessive to further researchers.

V.1. The first glimpses

Linear logic first appeared after the author had been challenged by Berry and Curien to extend the coherent semantics (at that time: qualitative semantics) to the sum of types, their claim being that it was necessary to reintroduce complications that are typical for Scott domains. The answer is reproduced in an appendix of [4], and (except for the notations which are absent), one can recognize (semantically) the decomposition of the type $A \Rightarrow B$ as $!A \multimap B$ and $A \vee B$ as $!A \oplus !B$. Incidentally, the answer was found because of the insistence on interpreting all known rules for type sum, including so-called 'commutative rules'; commutative rules were interpreted by remarking that eliminations were linear in their main premise and so, it was sufficient to 'linearize' the treatment of the sum. By the way, observe that the claim that 'eliminations are linear' sound deliciously obsolete now since, by the existence of the involutive *nil*, there is no real distinction between introduction and elimination!

V.2. The quantitative attempt

After this isolated remark, nothing happened before October 1985; the subject came back to life during a working session in Torino. At that moment, the quantitative semantics was considered by the author as more promising because more weird. The quantitative semantics was showing the decomposition in a very conspicuous way, e.g., $A \Rightarrow B = \text{Int}(A).B$, suggesting to consider separately Int and ., i.e., 'of course' and 'entails'. The decompositions in the case of qualitative domains (coherent spaces) were less obvious.

The first formalism for linear logic was therefore a functional language essentially based on $\multimap$, !, $\otimes$, $\oplus$, and second-order quantification, in which it was possible to make arbitrary sums (superpositions) of terms of the same type and in particular the void sum 0 of any type. Commutation rules of the sum with formation schemes expressed the linearity of everything but !-introduction and terms could be normalized as sums of primitive ones. This decomposition in sum has been given up when moving to the coherent semantics, though it was a very nice feature of the calculus. In particular, it was possible to decompose a term t as $t_0 + t_1$, with t_0 the normal part of t, t_1 the part still to be executed. Also, at that time, it was possible to write an equation of the style $!A = 1 + A.!A$ and to treat normalization as a process of Taylor expansion (this formula being reminiscent of $f(\mathrm{d}x) = f(0) + \mathrm{d}x.f'(0)$, something like that...). This approach had the clear advantage of making the execution of a program never end, except in trivial cases, and this is an aspect of parallelism that has been (perhaps temporarily) lost in the formalism we have chosen in this reference version. This approach was given up because of the lack of any *logical* justification, i.e., because it was not a proof-system. For instance, the system was not making any difference between $\oplus$ and &, or accepted void terms. The coherent semantics was then introduced more and more seriously, and the logical approach became more prominent.

V.3. The intuitionistic attempt

As the logical framework was clarifying itself, the formulation remained strictly intuitionistic; at that moment I made attempts (with Mascari) to work out an implementation of linear sequent calculus; although the formalism was intuitionistic, the communication between sequents stayed furiously symmetric. From this moment on (January 1986), the following things appeared:
- the well-hidden connectives nil, par, why-not;

- the coherent semantics (restriction to binary qualitative domains to get an involutive negation) in its definite version;
- the first attempt for proof-nets.

If there are features of the quantitative attempt that we mourn, then the giving up of the intuitionistic framework is apparently completely positive. The only inconvenience is the abandonment of the functional notation which is so easy to understand; but, '*il faut souffrir pour être belle*'.

V.4. Recent developments

One of the most irritating questions was the question of *totality*: in two words, not every element of a coherent space is *noble*; e.g., most of the time, we like to exclude the empty set from the possible semantics of our terms. For this we have to speak of *total* elements, which look like the potential proofs of their type; in [5], the situation has been clarified, but in an intuitionistic framework. The newly discovered 'classical' features of linear logic made it very complicated because one would have necessarily arrived at something like: 'given two types which are linear negations of one another, one of the types is empty (nothing total) while the other is full (everything is total)'. The solution found in April '86 was 'totality with phases' and led to the Tarskian semantics of phases of August '86. Before the phase semantics, several others were tried, without complete success, and, for instance, a functional interpretation of the classical case within the intuitionistic one (with the advantage of getting rid of problems of totality) was worked out and then dumped.

Another difficult question was the way to formulate proofs: sequent calculus was appropriate, but since there is no nice normalization in sequent calculus, we got into trouble. This is the reason why, before July '86, when the fundamental results on proof-nets were obtained, the following interesting solution was considered: compute directly with the semantics. The only thing one has to be sure of is that the objects we look for are finite enough and this offers no essential difficulty. Even now, this approach retains part of its attraction.

A rule was under discussion for a while, namely the rule of MIX, which can be formulated as

$$\frac{\vdash A \qquad \vdash B}{\vdash A, B}.$$

The rule says that $C \otimes D \multimap C \parr D$, which seems natural, and can be derived from $\perp \multimap \perp \otimes \perp$. Its adjunction would not change linear logic too much, however, there was a negative feeling about the rule, essentially the idea that $\vdash A, B$ means that A and B *do* communicate. The situation was clarified after the phase semantics appeared: semantically speaking, MIX is the requirement that $\perp$ is closed under product. There is no clear reason to require this (without requiring $1 \in \perp$, which would be a bit too much) and MIX is now no longer under serious discussion. One of the arguments for MIX is that, without it, the type of communication considered in proof-nets is very totalitarian: everything communicates with everything, while MIX could accept more liberal solutions, typically two non-interconnected proof-nets etc. However, what we said is true as long as $\perp$ is not used in proof-nets since it is only for formal reasons that $\perp$ can communicate upwards. In particular, two separate proof-nets β and β' can be put together by using $\perp$, the only price to pay being that another conclusion $(\perp \otimes \perp)$ has been added.

But what has been under discussion most and has still not been clarified is the exact formalism for ! (and ?). The question is of great interest because behind it, the implementation of beta-conversion lies; we devote the rest of this note to give a panorama of the main lines that have been considered.

V.5. *The exponentials*

Already in the well-known case of lambda-calculus, there are two traditions:

- the tradition of identifying the variables, which comes from beta-conversion, when we substitute u for all occurrences of x;
- the tradition coming from the implementation, which tries to repress or control substitution.

The first tradition rests on safe logical grounds, whereas the second one is a kind of *bricolage* with hazardous justifications.

In the first tradition, a term $t[x, y]$ will by identification yield the term $t[x, x]$ which would have also resulted from the consideration of $t[y, x]$. This is why the modelling of beta-conversion uses spaces of finite coherent *sets* (which can be seen in the definition of $!X$): we use a space of repetitions, but we make identifications between (x, y) and (y, x). After many hesitations, we have eventually chosen this definition, but it is far from being perfect:

(1) Nowhere has it been said that we should model the implicative types by plain graphs (which is done with this solution). On the other hand, modelling more subtle distinctions (for instance, distinguishing between the function coming from $t[x, y]$ and the one coming from $t[y, x]$) would give a more serious basis to the second tradition.

(2) The consideration of the space of *sets* has a bad consequence on the formal behaviour of our semantics: functorially speaking, we lose preservation of *kernels*, which seems to indicate that a theoretical mistake has been made.

Besides the variant retained in this reference paper, several others have been considered to remedy criticisms (1) and (2):

(i) As long as linear logic was resting on quantitative ideas, the principle (inspired on the way Krivine handled beta-conversion by restricting substitution to headvariables, and on the fact that a term is linear in its headvariable) suggested to use a linear 'first-order development', based on the identification between $!A$ and $1 + A.!A$. The operations of identification could be seen as formal derivation or formal primitive. The interest of this approach was to propose, at the theoretical level, to replace brutal beta-conversion by iterated linear conversions.

(ii) When the quantitative approach turned out to be insufficient, the author tried to preserve this idea by means of $!A \sim 1 \,\&\, A \otimes !A$. Unfortunately, this principle is not enough to justify contraction, even if written in a form inspired on inductive definitions (projective definitions). From this attempt a variant remained, due to Lafont, where $!A$ was defined as a projective definition, and satisfies $!A \sim 1 \,\&\, A \,\&\, (!A \otimes !A)$. This idea can be modelled in terms of coherent spaces and leads to considering certain trees instead of sets.

(iii) Another possibility seriously considered is the infinitary approach: $!A = \bigotimes(1 \,\&\, A)$, the tensorization being made on ω copies. Here we get quite a nice theory, but we have to take care of the heavy apparatus of infinite proof-nets. For implementation we also have to restore a finitary calculus, i.e., to do some *bricolage* again. Here the semantics involves finite sequences with holes, e.g., (x_1, x_3, x_7).[7]

The last two variants model what we have called the second tradition; i.e., they model strategies of progressive substitutions. Also, they are free from the defect we have mentioned w.r.t. kernels. However,

(3) the isomorphism $!(A \,\&\, B) \sim !A \otimes !B$ is no longer valid (it would be valid if we were considering multisets instead of sets);

(4) both variants do not succeed in catching the idea of first-order development.

Acknowledgment

The author is indebted to Jean-Louis Krivine, whose simultaneous work on lambda-calculus and its execution had some crucial influence on earlier versions of linear logic. Also, the author wishes to express his indebtedness to Gianfranco Mascari who introduced him to the subject of parallelism and whose influence was important for the shift to 'classical' framework and parallel syntax.

References

[1] J.-Y. Girard, Une extension de l'interprétation de Gödel à l'analyse et son application à l'elimination des coupures dans l'analyse et dans la théorie des types, in: J.E. Fenstadt, ed., *Proc. 2nd Scand. Logic Symp.* (North-Holland, Amsterdam, 1971) 63-92.

[2] J.-Y. Girard, Interprétation fonctionnelle et élimination des cupures de l'arithmétique d'ordre supérieur, Thèse de Doctorat d'Etat, UER de Mathématiques, Univ. de Paris VII (1972).

[3] J.-Y. Girard, Three-valued logic and cut-elimination: the actual meaning of Takeuti's conjecture, *Dissertationes Math.* CXXXVI (1976).

[4] J.-Y. Girard, Normal functors, power series and lambda-calculus, *Ann. Pure Appl. Logic*, to appear.

[5] J.-Y. Girard, The system *F* of variable types, fifteen years later, *Theoret. Comput. Sci.* 45(2) (1986) 159-192.

[6] G. Kreisel and G. Takeuti, Formally self-referential propositions for cut-free classical analysis and related systems, *Dissertationes Math.* CXVIII (1974).

[7] What remains of this idea is the Approximation Theorem of Section 5.