TRANSFORMATION METHODS IN LDS

K. BRODA, M. D'AGOSTINO AND A. RUSSO
Department of Computing
Imperial College of Science, Technology and Medicine
180, Queen's Gate, London SW7 2BZ
kb,mda,ar3@doc.ic.ac.uk

1. Introduction

The methodology of *Labelled Deductive Systems*— or simply LDS[1]— is a unifying framework for the study of logics and their interactions. It was proposed by Dov Gabbay a few years ago in response to conceptual pressure arising from application areas, and has now become a large and influential research programme providing logicians, both pure and applied, with a common language and a common set of basic principles in which to express and to solve their problems.

For the theoretician LDS provides the foundations for a *taxonomy* of logics and brings out the common structure underlying different logical systems proposed for a variety of different purposes— whether philosophical, or mathematical, or practical. For the applied logician (e.g. the computer scientist or the engineer) it provides a powerful technique to develop logical systems tailored to the needs of a specific application (see (Gabbay, 1992)), so as to maximize the role of logic in applied research.

In the LDS approach the basic declarative units of a deductive process are not just formulae but *labelled formulae*, i.e. expressions of the form $x : A$ where A is a formula of a standard logical language and x is a term of a given labelling language. The labels refer to elements of a suitable algebraic structure that we call *the algebra of the labels* (or *labelling algebra*). In general, a labelled formula $x : A$ expresses a relation between a formula A and an element x of the labelling algebra. It is not necessary to commit to

[1]Ian Hodkinson reports that Gabbay's original choice was LSD, for "Labelled Systems of Deduction"; the acronym LDS was adopted, to avoid ambiguities, when someone pointed out that LSD was already in use with a slightly different meaning.

any particular assumption on the nature of this relation. On some occasions we can read $x{:}A$ "semantically" as "A holds *relative to* x", where x is an element of an appropriate space to which it makes sense to relativize the truth of the proposition A[2]. On others, the label x may identify a region of a *structured database* – whose structure is modelled by the labelling algebra –, and the labelled formula records metalevel information about the dependence of A on x. According to some interpretations (originating in (Howard, 1980)) A may also represent a *type* and the label x a *term* of type A. What counts here is that in a labelled deductive system, the deduction rules act on the labels as well as on the formulae, according to certain fixed rules of propagation which depend on the intepretation of the labelled formulae.

For example, given a language whose only connective is $\rightarrow$, the *modus ponens* (or $\rightarrow$-elimination) rule may be given a general formulation as

$$\frac{\begin{array}{c} x : A \rightarrow B \\ y : A \end{array}}{f(x,y) : B}$$

where x and y are terms of a given labelling algebra and f is a function (associated with modus ponens) giving the new label after the rule has been applied.

We can obtain an appropriate $\rightarrow$-introduction rule by inverting the $\rightarrow$-elimination, as follows:

$$\frac{\begin{array}{c} t : A \\ f(x,t) : B \end{array}}{x : A \rightarrow B} \text{ with } t \text{ atomic.}$$

To show $x : A \rightarrow B$ we assume $t : A$ (with a new atomic label t) and we must prove $z : B$, for a z such that $z = f(x,t)$.

Different f's or different labelling algebras yield different variants of modus ponens and, possibly, different logics. For example, if we take an arbitrary semigroup as the algebra of the labels, with $\circ$ as multiplication and $f(x,y) = x \circ y$, we have the rules:

$$\frac{\begin{array}{c} x : A \rightarrow B \\ y : A \end{array}}{x \circ y : B} \qquad \frac{\begin{array}{c} t : A \\ x \circ t : B \end{array}}{x : A \rightarrow B}$$

The interpretation of these rules depends on the interpretation of the labelling algebra. For instance, the terms of the semigroup may represent

[2]For instance a Kripke frame for modal logics; for an LDS approach to normal modal logics see (Russo, 1996).

tasks and a labelled formula $x : A$ may mean that the task x *verifies* the proposition expressed by A. A "composite" task $x \circ y$ can then be interpreted as meaning the result of performing task x followed by task y. To be even more concrete, one can think of atomic formulae as describing different states of a system M, and of labels as describing actions that can make M change its state. A labelled formula $x : A$, with A atomic, means that the task x makes M change state from its initial state to the state described by A. A formula like A is then interpreted as *a type* and one can say that the task x is *of type* A. A conditional labelled formula $x : A \rightarrow B$ means that, whenever y is a task of type A, the composition of x and y, written as $x \circ y$, is a task of type B. From this point of view, x can be seen as a function and $\circ$ as functional composition. Under this interpretation the conditional rules given above become sound rules for reasoning about the system M and its state transitions.

Clearly, these rules are a (labelled) generalization of the traditional rules of Natural Deduction (Jaśkowski, 1934; Gentzen, 1935; Prawitz, 1965). The latter— together with the standard notion of a natural deduction proof of A from a set of assumptions Γ— characterize a well-known logical system, i.e. *intuitionistic* implication. But what about the labelled rules? Once a suitable labelling algebra and the interpretation of the function f have been fixed, do they still characterize a *logic*?

Let's consider the toy-example outlined above. Here the language contains only a binary logical operator that resembles the standard (classical or intuitionistic) conditional. We can consider an "empty task" 1, such that $1 \circ x = x \circ 1$ and agree that the empty task "verifies" the initial state of M (that is to say, we have to do nothing to bring the system to its initial state). Let $\mathbf{T}$ be the class of all formulae which are "verified" by the empty task (i.e. are always true at the initial state). Under this interpretation, $A \rightarrow A$ belongs to $\mathbf{T}$ and so does $(A \rightarrow B) \rightarrow ((C \rightarrow A) \rightarrow (C \rightarrow B))$. However, the formula $A \rightarrow (B \rightarrow A)$ does not. For a counterexample, think of tasks a and b verifying A and B respectively, but such that $a \circ b$ does *not* verify A. The *monotonicity* principle of classical and intuitionistic logic does not hold in this context. A little reflection shows that the same applies to the formulae $(A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B)$, embodying the *contraction* principle, and $(A \rightarrow (B \rightarrow C)) \rightarrow (B \rightarrow (A \rightarrow C))$, embodying the *permutation* principle. So, we have a "semantics" and a labelled deductive system for it. But do we also have a "logic"?

An answer to this question depends, of course, on our definition of "logic". Very seldom definitions are completely stable, and this is no exception. A "logic" is usually identified with a *consequence relation*, i.e. a binary relation formalizing the intuitive notion of logical consequence. The traditional definition of consequence relation, which was first formulated explicitly by Tarski (Tarski, 1930a; Tarski, 1930b), can be described as a relation $\vdash$ between sets of formulae and formulae satisfying the following conditions:

(Identity) $\quad A \vdash A$

(Monotonicity) $\quad \Gamma \vdash B$ implies $\Gamma, A \vdash B$

(Transitivity) $\quad \Gamma, A \vdash B$ and $\Delta \vdash A$ implies $\Gamma, \Delta \vdash B$

where Γ and Δ are *sets* of formulae.

For instance, the system of intuitionistic implication can be shown to correspond to the smallest consequence relation closed under the following additional condition concerning the $\rightarrow$ operator:

(Cond) $\quad \Gamma, A \vdash B$ iff $\Gamma \vdash A \rightarrow B$

The closure conditions in the definition of a consequence relation are all *structural* conditions, i.e. they do not involve any specific logical operator, but express basic properties of the notion of logical consequence[3]. The emergence of Relevance Logic (for an overview see (Dunn, 1986)) showed the inadequacy of the traditional definition of consequence relation. If the system R proposed by Andersonn and Belnap had to be called "logic", this definition had to be amended. Let's see why.

The whole idea of Relevance Logic is that in a "proper" deduction all the premisses *have to be used at least once* to establish the conclusion, so as to stop the validity of the notorious "fallacies of relevance" such as the so-called "positive paradox" $A \rightarrow (B \rightarrow A)$, or the *ex-falso quodlibet* principle $\neg A \wedge A \rightarrow B$, or its dual $B \rightarrow A \vee \neg A$. This criterion of use is sufficient to prevent the fallacies from being provable. For instance, in the typical Natural Deduction proof of the positive paradox the assumption B is discharged "vacuously" by the application of the $\rightarrow$-introduction rule, i.e. it is not used in obtaining the conclusion of the subproof constituting the premiss of the rule application. If we restrict our notion of proof in such a way that all the assumptions have to be *used* in order to obtain the conclusion of the proof, such "vacuous" applications of the $\rightarrow$-introduction rule are not allowed, since the subproof which occurs as premiss would not be a "proper proof". So, the standard proof is no longer an acceptable proof of the positive paradox, and it can be shown that no alternative proofs can be found. Similar considerations apply to the other fallacies.

[3]This terminology goes back to Gentzen (Gentzen, 1935) and his distinction between structural and operational rules in the sequent calculi.

The criterion of use is clearly of a "metalevel" nature. It takes the form of a side-condition on the application of the natural deduction rules. Let us consider the restricted ND-deducibility relation $\vdash'_{ND}$. Is it a consequence relation in the traditional sense? The answer is obviously "no", because it does not satisfy (Monotonicity), in that this condition would allow the addition of "irrelevant" assumptions which are not used in deducing the conclusion. So, if we want to consider the system of relevant implication as a logic, we have to drop (Monotonicity) from our definition of a consequence relation.

But this is not the whole story. In the system R, the definition of a "relevant" deduction requires that every single *occurrence* of an assumption is used to obtain the conclusion. Now, let us write $\Gamma \vdash_R A$, where Γ is a finite sequence of formulae, to mean that there is an R-deduction of A using all the elements of Γ (which are *occurrences* of formulae). Consider the statement $A, A \vdash_R A$. This is not provable because there is no way of using both occurrences of A in the antecedent in order to obtain the conclusion, i.e. one of these two occurrences is "irrelevant". Therefore, while $A \vdash_R A$ is trivially provable, $A, A \vdash_R A$ is not. It becomes provable if we "dilute" the criterion of use to the effect that *at least one* occurrence of each assumption needs to be used, as in the "Mingle" system (see (Dunn, 1986)). The trouble is that the distinction between these two different approaches to the notion of relevance cannot be expressed by the traditional notion of consequence relation. Indeed, according to this notion, a consequence relation is taken to be a relation between *sets* of formulae and formulae. But there is no way to distinguish the set $\{A, A\}$ from the set $\{A\}$ and, therefore, between $A, A \vdash_R A$ and $A \vdash_R A$. To make this sort of distinction we need a finer-grained notion of consequence relation.

In fact, such a finer-grained notion was developed by Gentzen back in the mid 1930's, with his calculus of sequents (Gentzen, 1935). In Gentzen's approach, a (single-conclusion[4]) consequence relation is axiomatized as a relation between a finite *sequence* of formulae and a formula. That such a relation holds between a sequence Γ and a formula A is expressed by the *sequent* $\Gamma \Longrightarrow A$. The axioms of Gentzen's system are given by all the sequents of the form $A \Longrightarrow A$. Gentzen specified also two sets of rules to derive new sequents from given ones, that he called *operational rules* and *structural rules*. While the first type of rules embodied, in his view, the meaning of the logical operators, the second type embodied the meaning of $\Longrightarrow$. These structural rules are the following:

[4] Gentzen also considered *multi-conclusion* consequence relations and showed how this more general notion was more convenient for the formalization of classical logic. See (Gentzen, 1935).

(Thinning)
$$\frac{\Gamma, \Delta \Longrightarrow B}{\Gamma, A, \Delta \Longrightarrow B}$$

(Permutation)
$$\frac{\Gamma, A, B, \Delta \Longrightarrow C}{\Gamma, B, A, \Delta \Longrightarrow C}$$

(Contraction)
$$\frac{\Gamma, A, A, \Delta \Longrightarrow B}{\Gamma, A, \Delta \Longrightarrow B}$$

(Cut)
$$\frac{\Gamma \Longrightarrow A \quad \Delta, A, \Lambda \Longrightarrow B}{\Delta, \Gamma, \Lambda \Longrightarrow B}$$

Here the rule (Thinning) corresponds to the (Monotonicity) condition of a consequence relation, while (Cut) corresponds to (Transitivity), except that, in the sequent formulation, the structure on the left of $\Longrightarrow$ is a *sequence*, rather than a *set*, of formulae. Moreover, in Gentzen's system the role of the (Identity) condition is played by the assumption that every axiom $A \Longrightarrow A$ can be used at any step of a sequent proof. The remaining rules of (Contraction) and (Permutation) have the effect of making Gentzen's relation $\Longrightarrow$ deductively equivalent to the consequence relation $\vdash$ with *sets* as the first argument.

Gentzen's richer axiomatization provides the means of characterizing "occurrence-sensitive" systems like R as "substructural" consequence relations, i.e. logics for which the standard structural rules of Gentzen's axiomatization may not hold. In the case of R, the rule which is dropped is the (Thinning) rule, responsible for the arbitrary introduction of "irrelevant" items in the antecedent of a sequent. After removing (Thinning), one can consider a rule symmetric to the contraction rule:

(Expansion)
$$\frac{\Gamma, A, \Delta \Longrightarrow B}{\Gamma, A, A, \Delta \Longrightarrow B}$$

The more "liberal" Mingle system is then distinguished from R by the fact that it allows for this weaker version of the (Thinning) rule.

The discussion of Relevance Logic clearly brings out the general idea that variations in the notion of logical consequence correspond to variations in the allowed *structural rules* of a suitable sequent-based system, leaving the basic *operational* rules unchanged. With Girard's Linear Logic (Girard, 1987; Avron, 1988) the "substructural movement" reached its climax. Linear Logic completely rejects the "vagueness" of traditional proof-theory concerning the use and manipulation of assumptions in a deductive process. A "proper" proof is one in which every assumption is used *exactly* *once*. If a particular assumption A can be used *ad libitum*, this has to be made *explicit* by prefixing it with the "storage" operator !. This means that the (Contraction) rule is not sound in Linear Logic, since it says, informally, that a proof of B from two occurrences of A can be turned into a proof of B from one occurrence only of A. But this is impossible, unless A is one of those assumptions which can be used *ad libitum*, in which case we should prefix it with the storage operator. In the "non-commutative" variant of Linear Logic – which was put forward in 1958 by Lambek (Lambek, 1958) as a system intended for applications to mathematical linguistics –, also the *order* in which assumptions are used becomes crucial. This means that the (Permutation) rule is also put in question.

We shall not dive further into the sea of substructural logics. For an overview of the subject the reader is referred to the excellent historical reconstruction in (Dôsen, 1993). In this paper we shall, instead, use a fragment of this family of logics as a case-study to illustrate a set of methods originating in the LDS program. In particular, we aim to illuminate the following aspects:

(I) By virtue of the extra power of labels and labelling algebras, traditional proof systems can be transformed so as to become applicable over a much wider territory whilst retaining a uniform structure. Different logics can be obtained by defining different labelling algebras, which therefore act as "parameters", and the transition from one logic to another can be captured as a parameter-changing process which leaves the structure of deductions unchanged[5]. In LDS logical systems are not studied "statically", in isolation, but "dynamically", observing the process of their generation and their interaction (via modifications of the labelling algebras) on the basis of a fixed proof-theoretical hard-core.

(II) The labelling algebras can be "discovered" by means of a systematic process which leads from a sequent-based definition of a consequence relation to a Kripke-style semantics for it which is then used to provide a suitable algebra of the labels. A crucial intermediate step of this process consists in applying the Lindenbaum-Tarski method to formulate the "algebraic semantics" of a given class of consequence relations. This time-honoured approach to the study of logics has been recently discredeted because of its lack of "usefulness". Here we show how algebraic semantics can be taken as an important step in the systematic development of a

[5]Although the idea of annotating formulae with labels is not new (in particular Anderson and Belnap used labels to keep track of "relevance of assumptions"), LDS has developed this idea in a very general and systematic way, turning it into a research program which is gradually changing our basic "logical perception". As with the well-known pictures of Gestalte psychology, Gabbay is pushing us to see a "duck" (an LDS) where we used to see only a "rabbit" (a traditional system of deduction).

suitable LDS[6]

(III) The theory of LDS is not confined to the natural deduction presentation given in (Gabbay, 1994). In the sequel we shall illustrate, in a concrete example, how LDS can be reformulated as a general theory of *Labelled Analytic Deduction* based on the system **KE**. This system is somewhat similar to the method of analytic tableaux, but drastically departs from the latter in the fact that it is *not* cut-free. On the contrary, it hinges on an *analytic cut* rule which is the only branching rule, all the elimination rules being of a *linear* format. For an exposition of the **KE** system and of its advantages over the tableaux method see (D'Agostino and Mondadori, 1994; D'Agostino, 1992) and (Broda *et al.*, 1996).

(IV) The **KE**-style analytic formulation can be used as an algorithm for finding a (labelled) natural deduction proof. While the natural deduction rules may be very suitable to present a proof *once it has been found*, the **KE** rules are more suitable when the problem is that of *finding* it[7]. We shall therefore describe an algorithm to turn a labelled **KE**-refutation into a labelled natural deduction proof. This algorithm consists in inverting the derivation process and showing that the refutation rules can be read as backward reasoning in a natural deduction system.

2. Consequence relations

2.1. PRELIMINARIES

We assume a propositional language $\mathcal{L}$ with two binary operators $\rightarrow$ and $\otimes$ and a constant $\top$. We shall use the upper case letters A, B, C, etc. as variables ranging over formulae of $\mathcal{L}$. The set of all well-formed formulae

[6]The material devoted to the semantics of substructural logics contained in this paper overlaps with the content of many other papers, in particular (Dôsen, 1988; Dôsen, 1989), but also (Avron, 1988). The Kripke-style "semantics" of implication which arises in section 4.2, and which can be found in a number of papers, is nothing but a generalization of Urquhart's semantics for relevant implication described in (Urquhart, 1972). For related semantic investigations into substructural logics see also (Ono, 1993), (Sambin, 1993), and (Abrusci, 1991). As is partially shown in the present paper, under certain circumstances, semantics for arbitrary consequence relations can be generated in a systematic way. On this problem see also (Gabbay, 1993a; Gabbay and Ohlbach, 1993a; Gabbay and Ohlbach, 1993b; Gabbay, 1993b). In our view, however, the main use of these ideas is in the development of a suitable LDS. "Semantics", here, is only a by-product of this process and its distinction from "syntax" is blurred by the LDS perspective.

[7]This conflict between *discovery* and *justification* is not a recent one. The Greek mathematician Pappus (3rd century A.D.) called the first process "analysis" and the second process "synthesis". He agreed with the nowadays ATP community that the "analysis" involves reasoning backwards, from the theorem that one wants to prove to the axioms (or the data), while the "synthesis" requires us to retrace our steps, somehow inverting them, from the axioms to the theorem. (See (Broda *et al.*, 1994) for a description of a goal directed approach to Natural Deduction.).

of $\mathcal{L}$ will be denoted by $\mathcal{F}$, and $\mathcal{F}^*$ will denote the set of all *sequences* of elements of $\mathcal{F}$. Sequences of formulae will be denoted by enclosing formulae, separated by commas, within square brackets, as in $[A, B, C]$. We shall use $[\,]$ to denote the empty sequence. We shall also use the capital Greek letters Γ, Δ, etc. as variables ranging over sequences of formulae. We shall often write $A, B, C, \ldots$ instead of $[A, B, C, \ldots]$ and Γ, Δ for the concatenation of sequences Γ and Δ.

Definition 1 We take a *consequence* relation[8] $\vdash$ as a relation $\mathcal{F}^* \times \mathcal{F}^*$ satisfying:

(Identity) $$A \vdash A$$

(Surgical Cut) $$\frac{\Delta, A, \Lambda \vdash B \quad \Gamma \vdash A}{\Delta, \Gamma, \Lambda \vdash B}$$

As we mentioned in the introduction, the notion of consequence relation is traditionally defined in terms of sets rather than sequences. The two approaches are equivalent whenever $\vdash$ is closed under the following conditions:

$$S_1 \quad \frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C} \qquad\qquad S_2 \quad \frac{\Gamma, A, A, \Delta \vdash C}{\Gamma, A, \Delta \vdash C}$$

$$S_3 \quad \frac{\Gamma, A, \Delta \vdash C}{\Gamma, A, A, \Delta \vdash C}$$

So, in a consequence relation satisfying all the structural rules S_1–S_3, the sequences occurring on the left of the turnstile can be replaced by *sets*. In the traditional definition, consequence relations are also required to satisfy the following *monotonicity* condition:

$$S_4 \quad \frac{\Gamma \vdash C}{\Gamma, A \vdash C}$$

We shall refer to conditions S_1–S_4 as to the *structural rules*. A logical system is said to be *substructural* if one or more of these structural rules are not assumed as valid.

The first question related to "substructural" logics concerns the logical interpretation of sequences of formulae. When all the structural rules are

[8] For a general approach to *structured* consequence relations see (Gabbay, 1993c).

allowed, the antecedent of a sequent is equivalent to the conjunction of its elements. In substructural logics, this is no longer the case, and a sequence of formulae represents different "conjunctions" depending on which structural rules are allowed. We shall use the symbol "$\otimes$" to denote any binary operator which satisfies the *minimal* condition for a conjunction operator, namely

$$(\mathrm{C}_\otimes) \qquad \Gamma, A, B, \Delta \vdash C \;\text{ iff }\; \Gamma, A \otimes B, \Delta \vdash C$$

So we can always consider a sequent $\Gamma \vdash A$ as equivalent to $\otimes\Gamma \vdash A$, where $\otimes\Gamma$ denotes the $\otimes$-concatenation of the formulae in Γ.

We shall also denote by "$\rightarrow$" any binary operator satisfying the minimal condition for an implication operator, namely

$$(\mathrm{C}_\rightarrow) \qquad \Gamma, A \vdash B \;\text{ iff }\; \Gamma \vdash A \rightarrow B$$

Finally, we shall use the constant $\top$ to represent the empty sequence in the antecedent of a sequent. So, the following holds for every consequence relation $\vdash$:

$$\mathrm{C}_\top \qquad \Gamma, \Delta \vdash B \;\text{ iff }\; \Gamma, \top, \Delta \vdash B.$$

2.2. SUBSTRUCTURAL CONSEQUENCE RELATIONS

The smallest consequence relation containing the operators $\rightarrow, \otimes, \top$ and closed under the structural rules S_1–S_4 corresponds to the $\{\rightarrow, \wedge, \top\}$ fragment of intuitionistic logic, where $\wedge$ corresponds to $\otimes$.

In the sequel we shall consider the class of subsystems of this fragment of intuitionistic logic obtained by restricting the allowed structural rules to any subset of $\mathbf{S} = \{S_1, S_2, S_3, S_4\}$. Such logics can be classified according to the set of operational conditions (defining the meaning of the logical operators) and the subset of structural rules that they satisfy. We shall use the notation $\mathrm{LJ}^\gamma_\alpha$, where γ is a set of logical operators (defined by conditions like $\mathrm{C}_\otimes$, $\mathrm{C}_\rightarrow$ and $\mathrm{C}_\top$ above) and α is a subset of $\{1, 2, 3, 4\}$, to indicate the smallest consequence relation closed under the appropriate conditions for the operators in γ (whatever they may be), and the structural rules in $\{S_i | i \in \alpha\}$. For every fixed γ, the set of logics $\{\mathrm{LJ}^\gamma_\alpha | \alpha \in 2^{\mathbf{S}}\}$ represents a family that we call "the LJ^γ family". In this paper we shall restrict ourselvers to the logics of the $\mathrm{LJ}^{\{\otimes,\rightarrow,\top\}}$ family and, therefore, we shall systematically omit the superscript. For instance $\mathrm{LJ}_\emptyset$ will denote the smallest consequence relation of the $\mathrm{LJ}^{\{\otimes,\rightarrow,\top\}}$ family which is not closed under any of the structural rules S_1–S_4, while $\mathrm{LJ}_{\{1,3\}}$ will be the smallest consequence relation of this family closed under S_1 and S_3.

3. The Algebraic Approach

3.1. THE ALGEBRAIC INTERPRETATIONS OF SEQUENTS

The well-known method of Lindenbaum-Tarski provides a way of turning the operators of a logical system L into the operators of an abstract algebra $\mathcal{A}$, which can then be used to define a valuation system characteristic of L. Provided L contains as theorems:

L1 $$A \rightarrow A$$
L2 $$(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$$

one defines $A \cong B$ iff $A \rightarrow B$ and $B \rightarrow A$ are both theorems of L, and observes that $\mathcal{F}/\cong$, where $\mathcal{F}$ is the set of all well-formed formulae of the language of L, is partially ordered by the relation $\|A\| \leq \|B\|$ iff $A \rightarrow B$ is a theorem of L.

One can then consider the equivalence classes $\|A\|$, $\|B\|$ etc. as the set of truth-values of a valuation system, and define operations on this set corresponding to the logical operators of L as defined by the proof-theory.

This method was meant to provide a characterization of the set of theorems of a logical system which was more convenient than the Hilbert-style axiomatic presentation for the purpose of mathematical investigation. It was originally applied to classical and intuitionistic logic, but can be in principle applied to any logic (satisfying L1 and L2 above) in which the relation $\cong$ is a congruence. In an axiom system, the notion of *theoremhood* comes first and that of *derivability* second. Today we tend to characterize a logical system in terms of its derivability relation, rather than in terms of theoremhood, and fix the axiomatic properties of this relation in terms of sequents, namely objects of the form $A_1, \ldots, A_n \vdash B$.

To apply the Lindenbaum-Tarski method to abstract consequence relations constructed as sets of sequents closed under certain conditions, we have to clarify the logical role played by the extra-logical symbols, like the comma and the turnstile. We have seen in the previous section that the comma is naturally associated with the operator $\otimes$. We can then interpret the comma by means of a binary operation, say $\circ$, satisfying different properties depending on the allowed structural rules. In any case, since the comma is a list-constructor, this operation is required to satisfy at least the laws of a monoid, with identity element 1 representing the empty list.

As we have seen in the previous section, a sequent $\Gamma \vdash A$ is equivalent to $\otimes\Gamma \vdash A$. Once we have shifted to this interpretation of a sequent as a relation between two formulae, it is easy to see that in every sequent calculus closed under Identity and Surgical Cut, the turnstile denotes a quasi-ordering[9] relation on the set of formulae. We can, therefore, take the equivalence classes of formulae under the equivalence relation $\cong$, defined as $A \cong B$ iff $A \vdash B$ and $B \vdash A$. These equivalence classes are partially ordered by: $\|A\| \leq \|B\|$ iff $A \vdash B$.

Let us turn our attention to the logical operators. It is not difficult to see that C$\otimes$, together with Surgical Cut, implies:

$$\frac{A \vdash B \quad C \vdash D}{A \otimes C \vdash B \otimes D} \tag{1}$$

This means that the condition

$$x \leq y \text{ and } w \leq z \text{ implies } x \circ w \leq y \circ z \tag{2}$$

has to be satisfied, for all $x, y, w, z \in \mathcal{F}/\cong$.

As to implication, in all the consequence relations considered here, its behaviour is characterized by C$\rightarrow$:

(C$\rightarrow$) $\qquad \Gamma, A \vdash B$ iff $\Gamma \vdash A \rightarrow B$

By using Surgical Cut and C$\rightarrow$ it is easy to derive:

$$\Gamma \vdash A \rightarrow B \text{ iff } \forall\Delta(\Delta \vdash A \text{ implies } \Gamma, \Delta \vdash B) \tag{3}$$

First assume $\Gamma \vdash A \rightarrow B$. By C$\rightarrow$, $\Gamma, A \vdash B$. Suppose $\Delta \vdash A$, then by (Surgical Cut), $\Gamma, \Delta \vdash B$. For the converse, assume $\forall\Delta(\Delta \vdash A$ implies $\Gamma, \Delta \vdash B)$. Hence, if Δ is the sequence $[A]$, we have, by (Identity), $\Gamma, A \vdash B$ and, by $C_{\rightarrow}$, we can conclude $\Gamma \vdash A \rightarrow B$. $\square$

In terms of the partial ordering $\leq$ defined above, this means that we can define an operator $\triangleright$, corresponding to $\rightarrow$, such that for all $x, y, z, v \in \mathcal{F}/\cong$:

$$x \leq v \triangleright z \text{ iff } \forall y(y \leq v \text{ implies } x \circ y \leq z) \tag{4}$$

Notice that this implies:

$$x \circ y \leq z \text{ iff } x \leq y \triangleright z \tag{5}$$

The study of the algebra outlined above allows for a sharp analysis of a wide variety of sub-systems of intuitionistic logic and provides a means of separating, within a given logic, what pertains to the minimal "meaning" of the logical operators, expressed by invariant inference rules, and what

[9] We recall that a binary relation is a quasi-ordering if it is reflexive and transitive.

pertains to our *procedural uses* of such operators, expressed by the changing properties of the relation $\leq$.

3.2. THE ALGEBRA OF LJ-STRUCTURES

In this section we describe the algebra which arises from the previous discussion. It should be obvious that this algebra is in no way more informative than the definition of a consequence relation in terms of sets of sequents. However, we shall see in the next sections that it is *heuristically useful*, in that it suggests developments which do not arise naturally from the consideration of sequents.

3.2.1. *LJ-structures*

Definition 2 An LJ-structure is a structure $(M, \circ, \rhd, 1, \leq)$ where

1. $(M, \circ, 1)$ is a monoid with identity 1, i.e. it satisfies:
   (a) $x \circ (y \circ z) = (x \circ y) \circ z$
   (b) $1 \circ x = x \circ 1 = x$
2. $\leq$ is a partial ordering of M, i.e. is a reflexive, antisymmetric and transitive binary relation between its elements;
3. the operation $\circ$ is *order-preserving*, that is:
$$x_1 \leq y_1 \text{ and } x_2 \leq y_2 \text{ iff } x_1 \circ x_2 \leq y_1 \circ y_2$$
4. For every x, y the set $\{z | z \circ x \leq y\}$ has a maximum element denoted by $x \rhd y$.

The binary operation $\circ$ may or may not satisfy (any combination of) the following additional axioms:

$\mathbf{B}_1$ $x \circ y = y \circ x$
$\mathbf{B}_2$ $x \leq x \circ x$
$\mathbf{B}_3$ $x \circ x \leq x$
$\mathbf{B}_4$ $x \circ y \leq x$.

We shall speak of *LJ_α-structures*, where α is a subset of $\{1, 2, 3, 4\}$, to denote LJ-structures such that the operation $\circ$ satisfies the additional conditions in $\{\mathrm{B}_i | i \in \alpha\}$.

4. Semantic Consequence Relations

In this section we shall define two kinds of semantic consequence relations, which are associated with LJ-structures: *many-valued* consequence relations and *possibile-world* consequence relations. We show that the two characterizations coincide.

4.1. MANY-VALUED CONSEQUENCE RELATIONS

Definition 3 Let S be an LJ-structure. A *many-valued valuation*, or *MV-valuation* for short, over S is a function h from the well-formed formulae of $\mathcal{F}$ to the terms of S, satisfying:

1. $h(A \to B) = h(A) \triangleright h(B)$
2. $h(A \otimes B) = h(A) \circ h(B)$
3. $h(\top) = 1$

Definition 4 The *MV-consequence relation* associated with a class of LJ-structures $\mathbf{C}$, is the relation $\models_{\mathbf{C}}^{MV}$ between finite sequences of formulae and formulae defined as follows:

$A_1, \ldots, A_n \models_{\mathbf{C}}^{MV} B$ iff $h(A_1) \circ \cdots \circ h(A_n) \leq h(B)$ for every LJ-structure $S \in \mathbf{C}$ and every MV-valuation h over S.

A formula A is *valid* in $\mathbf{C}$ whenever $\top \models_{\mathbf{C}}^{MV} A$.

It should be clear that every LJ_α-structure corresponds to the consequence relation LJ_α defined in the previous section, as stated in the following theorem:

Theorem 1 $\Gamma \vdash_{\mathrm{LJ}_\alpha} A$ *iff* $\Gamma \models_{\mathrm{LJ}_\alpha}^{MV} A$.

Proof: It is easy to verify that $\models_{\mathrm{LJ}_\alpha}^{MV}$ satisfies (Identity), (Surgical Cut) and $\mathrm{C}_\to$. Moreover, it satisfies also the structural rule S_i whenever $\leq$ satisfies B_i, for $i = 1, \ldots, 4$. This is sufficient to establish the soundness of $\vdash_{\mathrm{LJ}_\alpha}$ with respect to $\models_{\mathrm{LJ}_\alpha}^{MV}$. For the completeness, consider the equivalence relation $\cong$, defined as:

$$A \cong B \text{ iff } A \vdash_{\mathrm{LJ}_\alpha} B \text{ and } B \vdash_{\mathrm{LJ}_\alpha} A.$$

The set $\mathcal{F}/\!\cong$ is partially ordered by:

$$\|A\| \leq \|B\| \text{ iff } A \vdash_{\mathrm{LJ}_\alpha} B. \tag{6}$$

Consider the operation $\bullet$ on $\mathcal{F}/\!\cong$ defined by $\|A\| \bullet \|B\| = \|A \otimes B\|$. It follows from the properties of $\otimes$ that $\mathcal{F}/\!\cong$ is a monoid with identity $\|\top\|$. Moreover, it is easy to show that in every consequence relation:

$$A \vdash B \text{ and } C \vdash D \text{ implies } A \otimes C \vdash B \otimes D.$$

Hence, for every $x_1, x_2, y_1, y_2 \in \mathcal{F}/\!\cong$,

$$x_1 \leq y_1 \text{ and } x_2 \leq y_2 \text{ implies } x_1 \bullet x_2 \leq y_1 \bullet y_2$$

so that $(\mathcal{F}/\cong, \bullet)$ is coordinatewise ordered by $\leq$. Finally, for every $\|A\|$ and $\|B\|$ in $(\mathcal{F}/\cong, \bullet, \leq)$,

$$\|A \to B\| = \max\{z | z \bullet \|A\| \leq \|B\|\}.$$

For, suppose $\|C\| \bullet \|A\| \leq \|B\|$. Then

$$\begin{array}{lll} \|C\| \bullet \|A\| \leq \|B\| & \text{iff} & \|C \otimes A\| \leq \|B\| \text{ iff } C \otimes A \vdash B \\ & \text{iff} & C, A \vdash B \text{ iff } C \vdash A \to B \text{ iff } \|C\| \leq \|A \to B\| \end{array}$$

So $(\mathcal{F}/\cong, \bullet, \leq)$ is an LJ-structure where $\|A\| \rhd \|B\|$ is defined as $\|A \to B\|$. Moreover, it is easy to see that it is also an LJ_α-structure. Now, consider the function h defined by $h(A) = \|A\|$. By definition h is an MV-valuation. Now, suppose, $\Gamma \not\vdash_{\mathrm{LJ}_\alpha} A$. Then $\otimes\Gamma \not\vdash_{\mathrm{LJ}_\alpha} A$ and $\| \otimes \Gamma\| \not\leq \|A\|$ by (6). So, $\|B_1\| \bullet \cdots \bullet \|B_n\| \not\leq \|A\|$, where $B_1, \ldots, B_n$ are all the elements of Γ. Therefore, $h(B_1) \bullet \cdots \bullet h(B_n) \not\leq h(A)$ for some MV-valuation over some LJ_α-structure. $\Box$

4.2. POSSIBLE-WORLD CONSEQUENCE RELATIONS

In this section we show how to translate a many-valued consequence relation, such as the one outlined in the previous section, into an equivalent "possible-world" consequence relation. For this purpose we have to replace the plurality of truth-values with a plurality of "worlds" or "pieces of information" considered as the valuation space of a *two-valued* relative valuation (or "forcing" relation).

Definition 5 We define:

1. A *quasi-ordered monoid* is a structure $(M, \circ, 1, \preceq)$ such that $(M, \circ, 1)$ is a monoid with identity 1, and $\preceq$ is a quasi-ordering satisfying

$$x \preceq y \text{ and } v \preceq z \text{ implies } x \circ v \preceq y \circ z$$

2. A *partially ordered monoid* is a quasi-ordered monoid where the relation $\preceq$ is a partial ordering.

Definition 6 Let Q be a quasi-ordered monoid. A *PW-valuation over* Q is a two-argument function $\mathcal{F} \times Q \mapsto \{T, F\}$, where $\mathcal{F}$ is the set of formulae of the language, satisfying the following conditions:

1. $v(A, x) = T$ and $x \preceq y$ implies $v(A, y) = T$.
2. $v(A \to B, x) = T$ iff $\forall y, v(A, y) = T$ implies $v(B, x \circ y) = T$.
3. $v(A \otimes B, x) = T$ iff $\exists y, z,\ y \circ z \preceq x$ and $v(A, y) = T$ and $v(B, z) = T$.
4. $v(\top, x) = T$ iff $1 \preceq x$

So, the quasi-ordering of the monoid behaves like an "accessibility relation". The monoid Q is also called the *valuation space* or the *frame* of the PW-valuation. Sometimes the pair (Q, v), where Q is a frame and v is a PW-valuation over Q, is called a *model.*

Definition 7 A point z in the valuation space Q of a PW-valuation is A-*minimal* if $v(A, z) = T$ and $(\forall x \in Q)(v(A, x) = T$ implies $x \geq z)$.

We say that a PW-valuation over Q is *regular* if

$$(\forall A \in \mathcal{F})((\exists w \in Q)v(A, w) = T \text{ implies } (\exists z \in Q)z \text{ is } A\text{-minimal}).$$

Definition 8 The *regular PW-consequence relation* $\models^{PW}_{\mathbf{C}}$, associated with a class $\mathbf{C}$ of quasi-ordered monoids, is the relation between sequences of formulae and formulae defined as follows:

$$A_1, \ldots, A_n \models^{PW}_{\mathbf{C}} B \text{ iff } \begin{array}[t]{l} v(B, x_1 \circ \cdots \circ x_n) = T \text{ whenever } v(A_i, x_i) = T \\ \text{for all } i = 1, \ldots, n, \text{ for every } Q \in \mathbf{C} \text{ and every} \\ \text{regular PW-valuation } v \text{ over } Q. \end{array}$$

A formula A is valid in $\mathbf{C}$ if $\top \models^{PW}_{\mathbf{C}} A$.

Consider a quasi-ordered monoid Q. Let $\simeq$ be the equivalence relation defined as

$$x \simeq y =_{\text{def}} x \preceq y \text{ and } y \preceq x.$$

Let the operation $\bullet$ on $Q/\simeq$ be defined as follows: $\|x\| \bullet \|y\| =_{\text{def}} \|x \circ y\|$. Obviously, $(Q/\simeq, \bullet)$ is a monoid with identity $\|1\|$. Moreover, we can define a partial ordering $\sqsubseteq$ on $(Q/\simeq, \bullet)$ as usual: $\|x\| \sqsubseteq \|y\|$ iff $x \preceq y$. It is easily checked that $(Q/\simeq, \bullet, \sqsubseteq)$ is a partially ordered monoid.

Lemma 1 *Let v be a regular PW-valuation over a quasi-ordered monoid Q. Let $(Q/\simeq, \bullet, \sqsubseteq)$ be the associated partially ordered monoid. Moreover, let v' be the valuation function over $(Q/\simeq)$ defined as follows:*

$$v'(A, \|x\|) = T \quad \textit{iff} \quad v(A, x) = T.$$

Then v' is a regular PW-valuation, namely it satisfies all the conditions in Definition 6 and the regularity condition in Definition 7.

We now consider classes of quasi-ordered monoids satisfying any combination of the following axioms

$\mathbf{B}'_1$ $x \circ y \preceq y \circ x$ (commutativity)
$\mathbf{B}'_2$ $x \circ x \preceq x$ (contraction)
$\mathbf{B}'_3$ $x \preceq x \circ x$ (expansion)
$\mathbf{B}'_4$ $x \preceq x \circ y$ (monotonicity)

Notice that these conditions are obtained by inverting the conditions B_1-B_4 of page 13.

We shall also denote each class by $\mathbf{Q}_\alpha$, where α is, as usual, a subset of $\{1,2,3,4\}$. We shall denote by $\models^{PW}_{\mathbf{Q}_\alpha}$ the regular consequence relation associated with the class $\mathbf{Q}_\alpha$ of quasi-ordered monoids.

Corollary 1 *Let $\models^{PW}_{\mathbf{Q}_\alpha}$ be the regular PW-consequence relation associated with the class $\mathbf{Q}_\alpha$ of quasi-ordered monoids, and let $\models^{PW}_{\mathbf{P}_\alpha}$ be the similar consequence relation associated with the subclass $\mathbf{P}_\alpha$ of $\mathbf{Q}_\alpha$ such that $\mathbf{P}_\alpha = \{Q \in \mathbf{Q}_\alpha | Q$ is partially ordered$\}$. Then*

$$\Gamma \models^{PW}_{\mathbf{Q}_\alpha} A \quad \textit{iff} \quad \Gamma \models^{PW}_{\mathbf{P}_\alpha} A$$

Proof: The *only-if* direction is trivial, because if there is a PW-valuation v over a partially ordered monoid which falsifies $\Gamma \vdash A$ then, a fortiori, there is such a valuation over a quasi-ordered monoid. For the *if* direction, if v is a PW-valuation over a quasi-ordered monoid which falsifies $\Gamma \vdash A$, consider the valuation v' over $Q/\simeq$ defined in Lemma 1. Then v' is a PW-valuation over the partially ordered monoid $Q/\simeq$ and it is easy to see that v' falsifies $\Gamma \vdash A$. □

4.3. CORRESPONDENCE WITH MANY-VALUED SEMANTICS

Lemma 2 *Let v be a regular PW-valuation over a partially ordered monoid $(Q, \circ, \sqsubseteq)$ and let Q' be the set of all $x \in Q$ such that x is A-minimal for some $A \in \mathcal{F}$. Then*

$$(\forall x, y \in Q')(\exists w)(w = \min\{z | y \sqsubseteq z \circ x\}.$$

Moreover, Q' is closed under $\circ$.

Proof: If x, y are in Q', then x is A-minimal for some A and y is B-minimal for some B. Since v is regular, there is $w \in Q$ such that w is $A \to B$-minimal. By definition of PW-valuation, $v(B, w \circ x) = T$ and, since y is B-minimal, $y \sqsubseteq w \circ x$. Now, suppose $y \sqsubseteq z \circ x$. Since x is A-minimal, it follows that for all u such that $v(A, u) = T$, $y \sqsubseteq z \circ u$ and, since y is A-minimal, $v(A, z \circ u) = T$. Hence, by definition of a PW-valuation, $v(A \to B, z) = T$ and, since w is $A \to B$-minimal, $w \sqsubseteq z$. Moreover, it is easy to verify that Q' is closed under $\circ$. □

Corollary 2 *Let $(Q, \circ, \sqsubseteq)$ be a partially ordered monoid and v a regular valuation over it. The structure $(Q', \circ, \leq)$, where Q' is defined as in Lemma 2, and $x \leq y$ iff $y \sqsubseteq x$ is an LJ-structure.*

Theorem 2 *For all finite sequences Γ and all formulae A:*

$$\Gamma \models^{MV}_{\mathrm{LJ}_\alpha} A \quad \textit{iff} \quad \Gamma \models^{PW}_{\mathbf{Q}_\alpha} A$$

Proof: Consider an LJ_α-structure $S = (M, \circ, \leq)$. We define an associated structure $S' = (M', \bullet, \subseteq)$ as follows:

- M' is the set of all *increasing sets* or *order filters* generated by the points x of S, namely the sets $\uparrow x = \{y | x \leq y\}$, according to the partial ordering $\leq$.
- $\bullet$ is defined as follows: $\uparrow x \bullet \uparrow y = \uparrow(x \circ y)$;
- $\subseteq$ is ordinary set inclusion.

It is easy to verify that S' is partially ordered monoid belonging to $\mathbf{Q}_\alpha$.

Now, given a many-valued valuation h over S, we define an associated PW-valuation v_h over S' as follows:

$$v_h(A, x) = T \quad \text{iff} \quad h(A) \geq x (\text{i.e. } h(A) \in \uparrow x).$$

The reader can verify that v_h satisfies all the conditions for a *regular* PW-valuation.

Similarly, given a quasi-ordered monoid $S = (Q, \circ, \preceq)$ such that $S \in \mathbf{Q}_\alpha$, and a *regular* PW-valuation v over it, we can define an associated LJ_α-structure $S' = (Q', \circ, \leq)$. First we take the valuation v' over the partially ordered monoid $(Q/\simeq)$, defined as in Lemma 1. Then, we consider the function $h_v : \mathcal{F} \mapsto Q/\simeq$ defined as follows:

$$h_v(A) = \min\{z | v'(A, z) = T\}.$$

Let $h_v(\mathcal{F}) = \{h_v(A) | A \in \mathcal{F}\}$. It follows from Lemma 2 that $S' = (h_v(\mathcal{F}), \circ, \leq)$, where $\leq$ is the reverse of $\sqsubseteq$, is an LJ_α-structure. It is not difficult to verify that h_v satisfies all the conditions for an MV-valuation over the LJ_α-structure S', namely:

1. $h_v(A \otimes B) = h_v(A) \circ h_v(B)$
2. $h_v(A \to B) = h_v(A) \rhd h_v(B)$
3. $h_v(\top) = \|1\|$

The correspondence between MV and PW-valuations that we have just outlined is expressed by the following identities:

$$h_{v_h} = h \qquad\qquad v_{h_v} = v.$$

Now, for the if-direction of the theorem, suppose $h(A_1) \circ \cdots \circ h(A_n) \not\leq h(B)$ for some MV-valuation h over S. Then, it is easy to verify that $v_h(B, x_1 \circ \cdots \circ x_n) = F$ and $v_h(A_i, x_i) = T$ for all i. For the only-if direction, assume $v(B, x_1 \circ \cdots \circ x_n) = F$ and $v(A_i, x_i) = T$, for some regular PW-valuation v over a quasi-ordered monoid. Then it is easy to verify that $h_v(A_1) \circ \cdots \circ h_v(A_n) \not\leq h_v(B)$. $\square$

It follows from Theorem 1 and Theorem 2 that:

Corollary 3 *For all finite sequences of formulae* Γ *and all formulae* A

$$\Gamma \vdash_{\mathrm{LJ}_\alpha} A \quad \textit{iff} \quad \Gamma \models^{MV}_{\mathrm{LJ}_\alpha} A \quad \textit{iff} \quad \Gamma \models^{PW}_{\mathbf{Q}_\alpha} A$$

Forcing Notation. A PW-valuation v is the characteristic function of a relation between points of the valuation space and formulae of the language. This kind of relation is usually called a *forcing* relation and is denoted by $\Vdash$, the translation between the two formulations being the following: $v(A, x) = T$ iff $x \Vdash A$

5. From PW semantics to LKE-refutations

We now show how the PW-semantics described in the previous section can be reformulated in terms of a *labelled refutation system.* This will take the form of a generalization of the classical tableau-like system **KE** (see (D'Agostino and Gabbay, 1994)). The generalization involves shifting from signed formulae to *labelled* signed formulae, or LS-formulae, as basic units of a refutation, and so allows us to incorportate the semantics into the syntax, in the spirit of Gabbay's LDS approach.

5.1. THE IMPLICATION FRAGMENT

Definition 9 Let $\mathcal{A}$ be an alphabet containing (i) denumerably many symbols "a_1",..., "a_n" called *atomic labels*, (ii) the symbol "$\circ$". The set of labels is defined as the least set satisfying

1. Every atomic label is a label.
2. If x and y are labels, $x \circ y$ is also a label.

It is called *the labelling language* and is denoted by $\mathcal{L}_L$. A *label* is any element of $\mathcal{L}_L$. A *labelled signed formula*, or *LS-formula* for short, is an expression of the form $TA : x$ or $FA : x$ where A is a well-formed formula and x is a label.

The interpretation of LS-formulae is as follows: $TA : x$ stands for $v(A, x) = T$ (or, equivalently, $h(A) \geq x$ in terms of MV-valuations) and $FA : x$ for $v(A, x) = F$ (or, equivalently, $h(A) \not\geq x$).

Implication Rules. It follows from Definition 6 that for every PW-valuation:

$$v(A \rightarrow B, x) = T \quad \text{and} \quad v(A, y) = T \quad \text{imply} \quad v(B, x \circ y) = T \tag{7}$$

and that

$$v(A \rightarrow B, x) = F \quad \text{implies} \quad \exists y, v(A, y) = T \quad \text{and} \quad v(B, x \circ y) = F \tag{8}$$

Therefore, using the notation defined above, the following expansion rules are sound:

$$\frac{\begin{array}{l} TA \rightarrow B : x \\ TA : y \end{array}}{TB : x \circ y} \qquad\qquad \frac{FA \rightarrow B : x}{\begin{array}{l} TA : a \\ FB : x \circ a \end{array}} \tag{9}$$

where a is a *new atomic* label.

We have seen that (Corollary 1) we can restrict our attention, without loss of generality, to regular PW-valuations of partially ordered monoids. So in such valuations if there is a point that verifies a formula A, than there is also the *least* point that verifies A. (This property corresponds to the regularity property of PW-valuations.) So, we can always identify the new atomic label a in the rule for $FA \rightarrow B : x$ with such a minimum x at which A is true. Therefore, in every subsequent application of the EF$\rightarrow$ rule to a conditional with the same antecedent, we can *reuse* the same atomic label a, instead of introducing a new one. This amounts to allowing for the rule:

$$\frac{\begin{array}{l} TA : a \\ FA \rightarrow B : x \end{array}}{FB : x \circ a} \tag{10}$$

when a is atomic.

These rules are the *universal* rules for the conditional operator: they hold for every consequence relation which contains an operator $\rightarrow$ satisfying the deduction theorem.

We observe that our relative valuations are *bivalent*, so that for all formulae A and all points x of the valuation space:

$$v(A, x) = F \quad \text{or} \quad v(A, x) = T. \tag{11}$$

This means that the following branching rule is allowed:

$$\frac{}{FA : x \mid TA : x}$$

where A is an arbitrary formula of the logical language and x is an *arbitrary* label of the labelling language.

The rules just described are tree-expansion rules similar to the classical tableau rules. A branch is *closed* when it contains both $TA : x$ and $FA : x$ for some formula A and some label x. The extra-conditions on the accessibility relation, which characterize each particular implication logic, can be

TABLE 1. Types of Rules

Operational	Fundamental	Structural	
		Permutation	Contraction
ET→ $\dfrac{TA \to B:x \qquad TA:y}{TB:x\circ y}$	Clos $\dfrac{TA:x \qquad FA:x}{\times}$	$\dfrac{TA:z\circ x\circ y\circ v}{TA:z\circ y\circ x\circ v}$	$\dfrac{TA:z\circ x\circ x\circ y}{TA:z\circ x\circ y}$
		Expansion	Monotonicity
EF→ $\dfrac{FA \to B:x \qquad TA:a}{FB:x\circ a}$	PB $\dfrac{}{FA:x \mid TA:x}$	$\dfrac{TA:z\circ x\circ y}{TA:z\circ x\circ x\circ y}$	$\dfrac{TA:z\circ x\circ v}{TA:z\circ x\circ y\circ v}$

expressed as *structural* rules by means of the same notation. All the rules are summarized in Table 1

The operational rules are the *universal rules* for implication, derived from the general definition of a PW-valuation. The fundamental rules express the basic properties of our semantic notion of truth and falsity. The structural rules correspond to the properties of the accessibility relation expressed in axioms B_1'–B_4' of Section 4.

Derivations in these systems are *trees* of labelled signed formulae constructed according to the rules which characterize each system. A branch is *closed* if it ends with the symbol $\times$, otherwise it is *open*. A tree is *closed* when all its branches are closed. An *LKE-tree for* $\mathbf{F}$, where $\mathbf{F}$ is a set of LS-formulae, is a tree whose nodes are either LS-formulae in $\mathbf{F}$ or are obtained by an application of one of the LKE-rules. A formula A is *provable* from the set of assumptions Γ if and only if there is a closed tree for the set $\{TB_1 : b_1, \ldots, TB_n : b_n, FA : b_1\circ\cdots\circ b_n\}$, where $B_i \in \Gamma$, for $i = 1, \ldots, n$ and $b_i \neq b_j$ whenever $B_i \neq B_j$. Notice that the system of classical implication is obtained by simply ignoring the labels.

In our set-up the difference between the various implication systems is reduced to the difference between the corresponding structural rules. Such structural rules are not very handy from a practical point of view and may be troublesome in the formulation of a decision procedure. However, it is not difficult to show (see (D'Agostino and Gabbay, 1994)) that the application of the structural rules can be pushed down to end of each branch.

Theorem 3 *Every closed tree* $\mathcal{T}$ *for* $\mathbf{F}$ *can be transformed into a closed tree* $\mathcal{T}'$ *such that, in each branch, no application of an operational rule follows an application of a structural rule.*

So, for all practical purposes, we can dispense with the structural rules altogether, provided that we replace the closure rule with the following one:

(Clos)
$$\frac{\begin{array}{c} TA : x \\ FA : y \end{array}}{\times} \text{ provided } x \sqsubseteq y$$

where $\sqsubseteq$ is the partial ordering of the class $\mathbf{P}_\alpha$ of partially ordered monoids under consideration. In this formalization the difference between the various implication logics is reduced to a difference in the side-condition associated with the "closure" rule. This side-condition can be easily checked in each case. Without any side-condition, the resulting system is, of course, classical implication. In Tables 2 and 3 we show some examples of refutations.

TABLE 2.

$\vdash_{\mathrm{LJ}_\emptyset} (A \to B) \to ((C \to A) \to (C \to B))$	$\vdash_{\mathrm{LJ}_{\{1\}}} (A \to B) \to ((B \to C) \to (A \to C))$
$F(A \to B) \to ((C \to A) \to (C \to B)) : 1$	$F(A \to B) \to ((B \to C) \to (A \to C)) : 1$
$TA \to B : a$	$TA \to B : a$
$F(C \to A) \to (C \to B) : a$	$F(B \to C) \to (A \to C) : a$
$TC \to A : b$	$TB \to C : b$
$FC \to B : a \circ b$	$FA \to C : a \circ b$
$TC : c$	$TA : c$
$FB : a \circ b \circ c$	$FC : a \circ b \circ c$
$TA : b \circ c$	$TB : a \circ c$
$TB : a \circ b \circ c$	$TC : b \circ a \circ c$
$\times$	$\times$
	for commutative frames

TABLE 3.

$\vdash_{\mathrm{LJ}_{\{2\}}} A \to (A \to B)) \to (A \to B$	$\vdash_{\mathrm{LJ}_{\{3\}}} A \to (A \to A)$	$\vdash_{\mathrm{LJ}_{\{4\}}} A \to (B \to A)$
$F(A \to (A \to B)) \to (A \to B) : 1$	$FA \to (A \to A) : 1$	$FA \to (B \to A) : 1$
$TA \to (A \to B) : a$	$TA : a$	$TA : a$
$FA \to B : a$	$FA \to A : a$	$FB \to A : a$
$TA : b$	$FA : a \circ a$	$TB : b$
$FB : a \circ b$	$\times$	$FA : a \circ b$
$TA \to B : a \circ b$	for expansive frames	$\times$
$TB : a \circ b \circ b$		for monotonic frames
$\times$		
for contractive frames		

Notice how the right derivation of Table 2 fails for the logic $\mathrm{LJ}_\emptyset$ (i.e. with no structural rules), since $b \circ a \circ c \not\sqsubseteq a \circ b \circ c$ for some monoid in $\mathbf{P}_\emptyset$. Notice also how in Table 3 the leftmost derivation fails in $\mathrm{LJ}_{\{1\}}$ (and, *a fortiori*, in $\mathrm{LJ}_\emptyset$), since $a \circ b \circ b \not\sqsubseteq a \circ b$ for some monoid in $\mathbf{P}_{\{1\}}$, the derivation in the center one fails in $\mathrm{LJ}_{\{1,2\}}$ and its subsystems, since $a \not\sqsubseteq a \circ a$ for some monoid in $\mathbf{P}_{\{1,2\}}$ and the rightmost one fails in $\mathrm{LJ}_{\{1,2,3\}}$ and its subsystems, since $a \not\sqsubseteq a \circ b$ for some monoid in $\mathbf{P}_{\{1,2,3\}}$.

The reader can easily verify that any attempt to derive Peirce's law fails in all the implication systems except, of course, the one in which the labels are ignored, corresponding to classical logic.

Are the rules we have been described so far *complete* for every logic of the $\mathrm{LJ}^{\{\to\}}$ family? The answer is not utterly positive because of a disturbing exception, namely the family of logics which satisfy the structural rule S_3 (Expansion), but not its stronger version S_4 (Monotonicity). These logics, which include the well-known system of "mingle" implication, are provably closed under the following structural rule:

$$\frac{\Gamma \vdash A \qquad \Delta \vdash A}{\Gamma, \Delta \vdash A}$$

This rule can be justified "semantically" as follows. Suppose there are two points x_1 and x_2 at which a formula A is true. By definition of valuation there is the *minimum* point at which A is true. Let a be such a minimum point. If the frame is expansive we have that $a \sqsubseteq a \circ a$ Therefore, since $a \sqsubseteq x_i$ for $i = 1, 2$, A is verified also by the point $x_1 \circ x_2$. The argument can, of course, be generalized to any finite numbers of points which verify A.

This means that in any logic satisfying (Expansion) there cannot be a valuation v such that, for some finite set $\{x_1, \ldots, x_n\}$ of points $v(A, x_i) = T$, for all i, but $v(A, x_1 \circ \ldots \circ x_n) = F$. Hence, a branch containing all $TA : x_i$ and $FA : x_1 \circ \cdots \circ x_n$ should be considered closed.

This problem can be overcome by introducing a function h which picks up the *least* point (if any) that verifies A. This allows for a *more general* closure rule of which the previous one is just a special case:

$$\frac{\begin{array}{c} TA : x_1 \\ \vdots \\ TA : x_n \\ FA : y \end{array}}{\times} \tag{12}$$

provided that $h(A) \sqsubseteq y$.

We call $\mathrm{LKE}_{\to}$ (for "Labelled KE") the refutation system characterized by the operational rules for $\to$ plus the general closure rule (12). In fact, this is not a single refutation system, but a *family* of refutation systems which differ from each other only for the *algebra of the labels* used in checking the side-condition on the closure rule. This takes the form of a set of axioms $\mathcal{A}$, characterizing the class $\mathbf{Q}_\alpha$ of partially-ordered monoids under consideration (i.e. it will comprise the usual axioms for the partial ordering $\sqsubseteq$ plus a set of axioms characterizing the additional constraints identified by the subscript α). We shall use the notation $\mathcal{A}_\alpha$ to indicate the algebra of the labels corresponding to the class $\mathbf{Q}_\alpha$.

Accordingly, given any specific algebra of the labels $\mathcal{A}_\alpha$, we shall say that a branch of an LKE$_\rightarrow$-tree is *closed for* $\mathcal{A}_\alpha$ if (i) it contains a suitable set of premisses for the closure rule and (ii) the set of axioms of $\mathcal{A}_\alpha$ implies that the side-condition associated with this putative application of the closure rule is satisfied. Obviously, an LKE$_\rightarrow$-*tree* will be *closed for* $\mathcal{A}_\alpha$, if all its branches are closed for $\mathcal{A}_\alpha$.

It is not difficult to show (see (D'Agostino and Gabbay, 1994) for the details) that

Theorem 4 *For every finite* Γ, $\Gamma \vdash^{\{\rightarrow\}}_{\mathrm{LJ}_\alpha} A$ *if and only if there is a closed LKE*$_\rightarrow$*-tree for* $\mathcal{A}_\alpha$.

The implication fragments of some of the logics LJ_α are well-known logical systems. The correspondence is summarized in the following table:

$\mathrm{LJ}_\emptyset$	Lambek's right implication
$\mathrm{LJ}_{\{1\}}$	Girard's linear implication
$\mathrm{LJ}_{\{1,2\}}$	Andersonn's and Belnap's relevant implication
$\mathrm{LJ}_{\{1,2,3\}}$	Mingle implication
$\mathrm{LJ}_{\{1,2,4\}}$	Direct implication
$\mathrm{LJ}_{\{1,2,3,4\}}$	Intuitionistic implication

5.2. VARIABLES IN THE LABELS

Each label occurring in a tree is built up from atomic labels and the relevant operations of the labelling algebra. New atomic labels are introduced by applications of the rule EF$\rightarrow$, and the propagation of the labels is uniquely determined by the tree rules. By contrast, the rule PB is sound for *every* choice of the label x and we only know that, for every valid sequent $\Gamma \vdash A$, *there exists* a set of choices for the labels generated by the application of PB which leads to a closed tree. It is therefore convenient in practice to apply the rule PB with a *variable* label x and postpone the evaluation of this variable until enough information is available. For this purpose we need some new notions.

Definition 10 We enrich our labelling language with a denumerable set of *variables* denoted by $\gamma_1, \gamma_2, \gamma_3$ etc. A *label-scheme* α is a label containing variables. A *potential closure set* is a set of LS-formulae of the form

$\{TA : \alpha_1, \ldots, TA : \alpha_n, FA : \beta\}$, where $\alpha_1, \ldots, \alpha_n, \beta$ are label-schemes. A *potentially closed* branch is a branch containing a potential closure set. A *tree* $\mathcal{T}$ is *potentially closed* if all its branches are potentially closed.

Notice that a potentially closed branch may contain more than one potential closure set. So, every potentially closed branch ϕ determines a finite set I_ϕ of inequations, one for each potential closure set occurring in it. (Recall that, with the exception of the logics characterized by frames which are both non-monotonic and expansive, a closure set is always a pair $\{TA : \alpha, FA : \beta\}$, so that the inequations I_ϕ have the simple form $\alpha \sqsubseteq \beta\}$. Otherwise we have to resort to the more general inequations involving the funcion h.) Therefore, the "closure problem" for a potentially closed tree takes the following form:

> *Is there a set* $\mathbf{S}$ *of inequations such that* $\mathbf{S} \cap I_\phi \neq \emptyset$ *for all branches* ϕ*, and all the inequations in* $\mathbf{S}$ *have a simultaneous solution in a given algebra of the labels* $\mathcal{A}_\alpha$*?*

Example 1 As a simple example of the use of variables in the labels Figure 1 shows a tree for $\vdash ((A \rightarrow A) \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow C)$. This is a

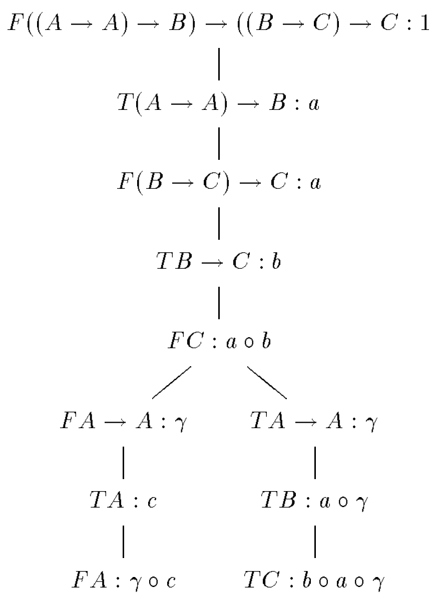

Figure 1.

potentially closed tree which is turned into a closed tree for all commutative labelling algebras under the substitution $\gamma = 1$.

5.3. RULES FOR $\otimes$

The appropriate tree-expansion rules for $\otimes$ can be read off the valuation condition for $\otimes$ in Definition 6, namely

$$v(A \otimes B, x) = T \quad \text{iff} \quad \exists y, z,\ y \circ z \sqsubseteq x \quad \text{and} \quad v(A, y) = T \quad \text{and} \quad v(B, z) = T.$$

Accordingly, the rules for $\otimes$ are as follows:

ET$\otimes$	EF$\otimes$	EF$\otimes$
$\dfrac{TA \otimes B : x}{\begin{array}{l} TA : a \\ TB : x/a \end{array}}$ with a *new*	$\dfrac{\begin{array}{l} FA \otimes B : x \\ TA : y \end{array}}{FB : x/y}$	$\dfrac{\begin{array}{l} FA \otimes B : x \\ TB : y \end{array}}{FA : x/y}$

where x/y denotes the *set* of points z such that $y \circ z \sqsubseteq x$. The consideration of a *set* of points as a label of a signed formula involves a reinterpretation of its intuitive meaning. If X is a set of points, an LS-formula $TA : X$ will mean that $v(A, x) = T$ for *some* $x \in X$, while the LS-formula $FA : X$ will mean that $v(A, x) = F$ for *all* $x \in X$. Indeed, we can re-interpret *all* the labels as sets of points. For this purpose it is sufficient to regard an atomic label a as shorthand for the singleton $\{e\}$. Moreover, if X and Y are sets of points, we can define $X \circ Y = \{x \circ y \mid x \in X \text{ and } y \in Y\}$. Under this intepretation the rules given above (as well as the implication rules) are easily seen to be sound. We can also formulate a "liberalized" version of the ET$\otimes$ rule, as we did for the EF$\rightarrow$ rule, by assuming that the atomic label a denotes the *least* point at which A is true. Therefore, in every subsequent application of the ET$\otimes$ rule to an LS-formula of the form $TA \otimes C : x$, we can *reuse* the same atomic label a, instead of introducing a new one.

We call $\text{LKE}_{\{\rightarrow,\otimes\}}$ the system which results by adding the three rules for $\otimes$ to the system $\text{LKE}_{\rightarrow}$ considered in the previous section. An example of an $\text{LKE}_{\{\rightarrow,\otimes\}}$-refutation is given in the next example.

Example 2 The formula $(A \otimes B \to C) \to (A \to (B \to C))$ is valid in all frames.

$$
\begin{array}{c}
F(A \otimes B \to C) \to (A \to (B \to C)) : 1 \\
| \\
T A \otimes B \to C : a \\
| \\
F A \to (B \to C) : a \\
| \\
T A : b \\
| \\
F B \to C : a \circ b \\
| \\
T B : c \\
| \\
F C : a \circ b \circ c \\
\swarrow \qquad \searrow \\
\begin{array}{cc}
F A \otimes B : x & T A \otimes B : x \\
| & | \\
F B : x/b & T C : a \circ x
\end{array}
\end{array}
$$

The right-hand branch of the above tree is closed under the substitution $x = b \circ c$. Since c is certainly an element of the set $(b \circ c)/b$, the left-hand branch is closed under the same substitution. □

Again it can be shown (see (D'Agostino and Gabbay, 1994) for the details) that $\text{LKE}_{\{\to,\otimes\}}$ is complete for all the logics of the $\text{LJ}^{\{\to,\otimes\}}$ family.

5.4. ANALYTIC CUT

All the elimination rules of the system $\text{LKE}_{\{\to,\otimes\}}$ are *analytic* in the sense that they "analyse" complex formulae by specifying the *consequences* of their truth and falsity in terms of the truth and falsity of their subformulae. On the other hand the branching rule PB (which is clearly related to the cut rule of the sequent calculus) can introduce *arbitrary formulae* and *arbitrary labels*. So the problem of proof-search in our system depends crucially on *two* components: the choice of the formulae introduced via PB and the choice of the associated labels.

In the classical **KE** system – with no labels – the use of the cut rule PB can be restricted to *analytic* applications, i.e. applications preserving the subformula property, without loss of completeness. Indeed, the choice of the "cut formulae", namely the formulae introduced by an application of PB, can be even further restricted so that the resulting refutations follow a regular pattern or *canonical form* and can be found by means of a simple systematic procedure, similar to the usual tableau-completion procedure (see (D'Agostino and Mondadori, 1994) for a detailed discussion of this point). Moreover, such canonical refutations are often essentially shorter than the corresponding LKE-tree refutations and *never* significantly longer (see again (D'Agostino and Mondadori, 1994) for the related technical results in terms of polynomial simulations).

This property extends also to the labelled version of **KE** that we are considering here: we can restrict the applications of PB to subformulae without loss of completeness, and define a refutation procedure which is a labelled generalization of the classical one. However, the use of *labelled* formulae introduces a further degree of freedom in the application of the PB rule: the choice of the label. Investigating possible ways in which this freedom can be restricted for each given algebra of the labels is the key to the decision problem for the corresponding substructural systems. Of course, fully mechanized proof-search can be achieved only when the complexity of the labels can be bound in one way or the other. This investigation goes beyond the limits of the present exposition, but the reader can find some clues in (D'Agostino and Gabbay, 1994) and a more detailed discussion in the (forthcoming) Part II of the same article.

6. A procedural interpretation of the LKE-rules

In this section we describe an algorithm to find (labelled) natural deduction proofs for logics of the $\mathrm{LJ}^{\{\rightarrow,\otimes\}}$ family, based on the $\mathrm{LKE}_{\{\rightarrow,\otimes\}}$ system. It consists of inverting $\mathrm{LKE}_{\{\rightarrow,\otimes\}}$-refutations, showing that the EF$\otimes$ and EF$\rightarrow$ rules can be read as backward reasoning in a natural deduction style. This algorithm could be useful in applications of substructural logics to all the areas where a human–oriented interface, based on natural deduction, is required.

6.1. SUBSTRUCTURAL NATURAL DEDUCTION

Following the framework developed in (Gabbay, 1994; Broda *et al.*, 1994) we define in Table 4 six Natural Deduction (ND) inference rules which correspond to the tree-expansion rules for $\{\rightarrow,\otimes\}$[10], and to the closure and PB rules. The ND rules can be proved to be complete and sound with respect to the LKE-rules (see Section 6.3). We will restrict our attention

[10] $\otimes$ has higher precedence than $\rightarrow$

$\boxed{\begin{array}{l} A : a \\ \vdots \\ B : x \circ a \end{array}}$ $A \to B : x \quad \to\mathcal{I}$ (i)	$\to\mathcal{E} \quad \begin{array}{l} A \to B : x \\ A : y \\ \hline B : x \circ y \end{array}$
$\boxed{\begin{array}{l} \vdots \\ A : \gamma_1 \end{array}} \boxed{\begin{array}{l} \vdots \\ B : \gamma_2 \end{array}}$ $A \otimes B : x \quad \otimes\mathcal{I}$ (ii)	$\otimes\mathcal{E} \quad \begin{array}{ll} \underline{A \otimes B : x} & \\ A : a & \text{(i)} \\ B : b & \text{(iii)} \end{array}$
$\vdots$ $A : x$ $\vdots$ $A : y \quad \surd$ if $x \sqsubseteq y$	$\vdots$ $\boxed{\text{proof of lemma } A : \gamma}$ $A : \gamma$ Lemma $\vdots$ goal

(i) a is a new atomic label
(ii) $\gamma_1 \circ \gamma_2 \sqsubseteq x$
(iii) $a \circ b \sqsubseteq x$

TABLE 4. Natural Deduction Rules

to regular PW-valuations, namely for every formula A, if there is a point x that verifies A, then there is also the least point, called a, that verifies A. We will refer to this least point as the *A-characteristic* label.

In the rule $\to\mathcal{I}$, the box can be seen as introducing a new context composed of the assumption $A : a$ together with the other sentences already assumed or derived. To deduce $A \to B : x$, it is needed to show that $B : x \circ a$ can be derived using the extra assumption $A : a$. The rule $\otimes\mathcal{E}$ corresponds to the ET$\otimes$ rule of LKE$_{\{\to,\otimes\}}$ system. However, to maintain a uniformity in the constraints on the labels the expression x/a used in Section 5.3 is replaced by the B-characteristic-label b together with the imposed constraint $a \circ b \sqsubseteq x$. The atomic label a introduced in $\to\mathcal{I}$ or $\otimes\mathcal{E}$ (see note (i)) is the A-characteristic label corresponding to the assumption A. The rule $\to\mathcal{E}$ is the usual Modus Ponens adapted for labelled formulas. The inference rule $\otimes\mathcal{I}$ replaces the two EF$\otimes$ rules shown in Section 5.3. It is a symmetric rule which can be read as: "to prove a formula $A \otimes B : x$, show $A : \gamma_1$ and $B : \gamma_2$", where the free variables γ_1 and γ_2 must satisfy the constraint $\gamma_1 \circ \gamma_2 \sqsubseteq x$. If $A : \gamma_1$ is already derived for some γ_1, then the rule reduces to the left EF$\otimes$ rule shown in Section 5.3. Analogously, if $B : \gamma_2$ is already derived for some γ_2, then the rule reduces to the right EF$\otimes$ rule. The variables γ_1 and γ_2 eventually have to be instantiated by particular ground labels[11] which satisfy the required constraint $\gamma_1 \circ \gamma_2 \sqsubseteq x$ (see note (ii)). Depending on the logic in use, there may be different instantiation values. If it is obvious that $A : \gamma_1$ and $B : \gamma_2$ are already provable, with $\gamma_1 \circ \gamma_2 \sqsubseteq x$, then $A \otimes B : x$ can be derived immediately. The $\surd$-rule recognises when a particular goal has been achieved and corresponds to the closure rule, with the required constraint $x \sqsubseteq y$. As free variables may be introduced in the application of the $\otimes\mathcal{I}$ rule, the labels x and y may be label-schemes. Therefore, suitable ground values have to be found so that the inequality $x \sqsubseteq y$ is satisfied.

Since LKE-trees make use of the semantic equivalent of the cut rule, namely the PB rule, we also need some way of including intermediate lemmas. This is achieved by the *lemma*-rule. Once a proof of $A : \gamma$ has been found, possibly with imposed or required constraints, then $A : \gamma$ can be used as an assumption in proving the main goal.

A proof of a theorem is constructed by working backwards from the formula (seen as a goal) using introduction rules and, as further labelled assumptions are introduced by $\rightarrow\mathcal{I}$, by working forwards using elimination rules. As in the classical case, declarative units assumed or derived in a particular context can be imported into a larger context. In general, a proof will impose some constraints and will require others to be satisfied. We call the imposed constraints *ICs* and the constraints to be satisfied *RCs*. *ICs* arise from the use of $\otimes\mathcal{E}$ and *RCs* occur with the application of $\otimes\mathcal{I}$ and $\surd$ rules. Unlike assumptions, *ICs* are global and may be used anywhere in a proof to satisfy *RCs*. Thus there is an overall first order constraint on a proof of the form

$$\bigwedge_{i=0}^{n} IC_i \rightarrow \bigwedge_{j=0}^{m} RC_j \tag{13}$$

where $\{IC_0, \ldots, IC_n\}$ are the imposed constraints and $\{RC_0, \ldots, RC_m\}$ are the required constraints. In order to satisfy (13) – find values for the variables (if any occur in it) that make it true – the properties of the labelling algebra are used.

An example of a proof of a theorem in Linear Logic, which satisfies the commutative property of $\circ$, is given in Figure 2. In this example, the general constraint corresponding to the implication (13) is given by

$$(a \circ c \sqsubseteq b) \rightarrow (a \circ a \sqsubseteq \gamma_3) \wedge (c \circ \gamma_3 \sqsubseteq a \circ b)$$

[11] We call a label-scheme *ground* when all its variables have been instantiated either to an atomic label or to a concatenation of atomic labels.

1	$A : a$		
2	$A \otimes (A \otimes A \to B) : b$		
3	$A : a$		can be omitted
4	$A \otimes A \to B : c$		$\otimes\mathcal{E}$ $(a \circ c \sqsubseteq b)$
5	$\surd$ $(\gamma_1 = a)$	$\surd$	$(\gamma_2 = a)$
6	$A : \gamma_1$	$A : \gamma_2$	
7	$A \otimes A : \gamma_3$		$\otimes\mathcal{I}$ if $\gamma_1 \circ \gamma_2 \sqsubseteq \gamma_3$
8	$A \otimes A : \gamma_3$		Lemma
9	$B : c \circ \gamma_3$		$\to\mathcal{E}$
10	$B : a \circ b$		$\surd$ if $c \circ \gamma_3 \sqsubseteq a \circ b$
11	$(A \otimes (A \otimes A \to B)) \to B : a$		$\to\mathcal{I}$
12	$A \to ((A \otimes (A \otimes A \to B)) \to B) : 1$		$\to\mathcal{I}$

Figure 2. An Example of Natural DeductionProof

This is satisfied by choosing $\gamma_3 = a \circ a$ and using commutativity and the order-preserving property of $\circ$ (i.e $a \circ c \sqsubseteq b \Rightarrow c \circ a \sqsubseteq b \Rightarrow c \circ a \circ a \sqsubseteq b \circ a \Rightarrow c \circ a \circ a \sqsubseteq a \circ b$.

6.2. LKE-RULES AS NATURAL DEDUCTION

In this section we describe a procedural interpretation of the LKE-rules and show how an LKE-tree can be reformulated as a Natural Deduction proof. Let us interpret the labelled signed formulae as follows:

- $TA : x$ means that the labelled formula $A : x$ belongs to the data or is a provisional assumption.
- $FA : x$ means that the labelled formula $A : x$ is to be proven.

Under this interpretation, the rule (EF$\to$) can be read as a *goal-reduction rule*:

$$\begin{array}{ll} FA \to B : x & \text{prove } [A \to B : x] \\ \hline TA : a & \text{assume } [A : a] \text{ and} \\ FB : x \circ a & \text{prove } [B : x \circ a] \end{array}$$

The rule (ET $\to$), on the other hand, is a *data-expansion rule* which processes the data via applications of the labelled version of Modus Ponens. (It should be clear to the reader that these correspond exactly to the Natural Deduction rules $\rightarrow\mathcal{I}$ and $\rightarrow\mathcal{E}$ of Section 6.1).

Let us see a simple example of how this interpretation works: consider the contraction axiom $(A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B)$. We show below its proof in LKE side by side with its procedural interpretation (recall that 1 is the empty data structure). Whether the LKE-tree closes depends on the

1.	$F(A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B) : 1$	prove $[(A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B) : 1]$
2.	$T(A \rightarrow (A \rightarrow B)) : a$	assume $[(A \rightarrow (A \rightarrow B)) : a]$ and
3.	$FA \rightarrow B : a$	prove $[A \rightarrow B : a]$
4.	$TA : b$	assume $[A : b]$ and
5.	$FB : a \circ b$	prove $[B : a \circ b]$
6.	$TA \rightarrow B : a \circ b$	$A \rightarrow B : a \circ b$ follows from 2 and 4
7.	$TB : a \circ b \circ b$	$B : a \circ b \circ b$ follows from 6 and 4
8.	$\times$ (if $a \circ b \circ b \sqsubseteq a \circ b$)	$\surd$ (if $a \circ b \circ b \sqsubseteq a \circ b$)

constraint $a \circ b \circ b \sqsubseteq a \circ b$ being satisfied by the labelling algebra. Similarly, in the procedural version (right-hand side), the deduction of $B : a \circ b \circ b$ from $B : a \circ b$ succeeds under the same condition. Closure is therefore the same as the $\surd$-rule of Section 6.1. We can therefore invert the above labelled analytic proof and turn it into a direct proof in which each application of a goal-reduction rule EF$\rightarrow$ becomes an $\rightarrow\mathcal{I}$ rule. A ND proof of the above theorem is given below.

1	$A \rightarrow (A \rightarrow B) : a$	
2	$A : b$	
3	$A \rightarrow B : a \circ b$	$\rightarrow\mathcal{E}$
4	$B : a \circ b \circ b$	$\rightarrow\mathcal{E}$
5	$B : a \circ b$	$\surd$ if $a \circ b \circ b \sqsubseteq a \circ b$
6	$A \rightarrow B : a$	$\rightarrow\mathcal{I}$
7	$(A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B) : 1$	$\rightarrow\mathcal{I}$

This is a correct proof in relevance logic but not in linear logic because the assumption A is used twice: first with $A \rightarrow (A \rightarrow B)$ to infer $A \rightarrow B$ and then with $A \rightarrow B$ itself to infer B. Indeed, to satisfy the RC $a \circ b \circ b \sqsubseteq a \circ b$ the contraction property is required.

This procedural interpretation can be extended to the PB, ET$\otimes$ and EF$\otimes$ rules. As for the PB rule, it is not difficult to see that its role is that of generating appropriate *lemmas* to be used in a proof. Consider an arbitrary application of the PB rule in a LKE-tree as shown in Figure 3. If the sub-tree T_1 is closed, then the "goal" $FA : x$ succeeds. Hence the labelled formula $TA : x$ is provable and can be used as a lemma in the

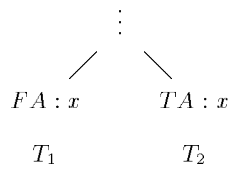

Figure 3.

sub-tree T_2. So, this rule can be interpreted as follows: first generate as a sub-proof the proof of the lemma, which corresponds to the left sub-tree, and then add the statement of the lemma as an assumption below the subproof (in the same way as $TA : x$ is added by the PB rule in the right sub-tree). The proof of a lemma may lead to additional global RCs and ICs.These are propagated throughout the remaining ND proof.

We now consider an example in Linear Logic, in which use is made of a lemma. The LKE-tree is given in Figure 1 of Example 1. The left sub-tree yields the following trivial natural deduction sub-proof with the single RC $c \sqsubseteq \gamma \circ c$. This is satisfied for $\gamma = 1$ as shown also in Example 1.

1	$A : c$	
2	$A : \gamma \circ c$	$\surd$ if $(c \sqsubseteq \gamma \circ c)$
3	$A \rightarrow A : \gamma$	

The instantiation of the free variable γ is propagated in the proof by replacing any of its occurrences with 1. The complete corresponding Natural Deduction proof is shown in Figure 4. Here, the remaining RC $b \circ a \sqsubseteq a \circ b$ is satisfied, as the operator $\circ$ is commutative in the labelling algebra $\mathcal{A}_{\{1\}}$ corresponding to the class $\mathbf{Q}_{\{1\}}$ of partially-ordered monoids of Linear Logic.

As for the EF$\otimes$ rule, the correspondence with the ND $\otimes\mathcal{I}$ rule is not quite immediate. In the LKE$_{\{\rightarrow,\otimes\}}$ system there are two sub-cases, depending on whether or not $TA : \gamma$ is already in the branch for some γ. In the first case (i.e. $TA : \gamma$ is already in the branch for some γ) the ND left subproof of the $\otimes\mathcal{I}$ rule succeeds immediately by the $\surd$ rule and the variable γ_1 is bound to γ. This leaves only the labelled formula $B : \gamma_2$ to be shown, with the RC $\gamma_1 \circ \gamma_2$ $(= \gamma \circ \gamma_2) \sqsubseteq x$. In the second case (i.e. $TA : \gamma$ is not in the branch) the EF$\otimes$ rule is preceded by the application of the PB rule on the LS-formula $TA : \gamma$. The left branch of this rule will provide a refutation proof of the formula $TA : \gamma$, whereas in the right branch the added assumption $TA : \gamma$ will allow the application of the EF$\otimes$. This case and its corresponding ND interpretation is shown in Figure 5. In the ND proof, the application of the $\otimes\mathcal{I}$ rule is equally preceded by the application

1	$(A \rightarrow A) \rightarrow B : a$	
2	$B \rightarrow C : b$	
3	proof of $A \rightarrow A : \gamma$	
4	$A \rightarrow A : 1$	Lemma$\gamma = 1$
5	$B : a$	$\rightarrow\mathcal{E}$
6	$C : b \circ a$	$\rightarrow\mathcal{E}$
7	$C : a \circ b$	$\surd$ if $(b \circ a \sqsubseteq a \circ b)$
8	$(B \rightarrow C) \rightarrow C : a$	$\rightarrow\mathcal{I}$
9	$((A \rightarrow A) \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow C) : 1$	$\rightarrow\mathcal{I}$

Figure 4.

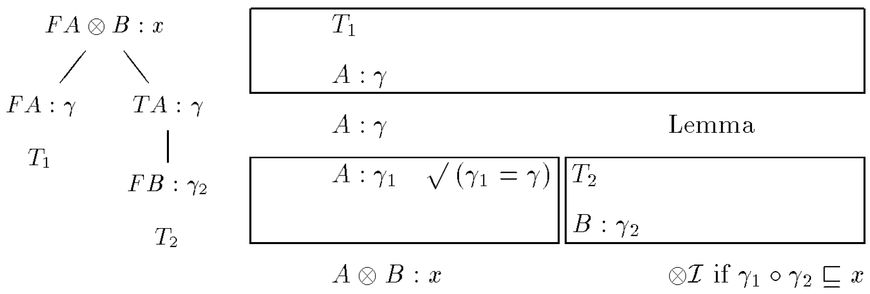

Figure 5. Translation of $\otimes$ rule

of the lemma rule (interpretation of the PB rule) as shown in Figure 5. Here the generation of the lemma $A : \gamma$ allows the left-hand subgoal of the $\otimes\mathcal{I}$ rule to be immediately satisfied by the application of the $\surd$ rule. The right-hand box of the $\otimes\mathcal{I}$ rule will include the right sub-tree of the PB rule application, and its context will be composed also of the lemma $A : \gamma$.

A more natural proof of $A \otimes B : x$ using the natural deduction rules contains the proof of $A : \gamma_1$ directly within the left-hand box of the rule $\otimes\mathcal{I}$. If $A : \gamma_1$ already exists, then the result is similar to that obtained using trees. If not, instead of using the Lemma rule, notice that the proof of $A : \gamma_1$ is not in any different context than the subsequent proof of $B : \gamma_2$, so its proof can be repeated if necessary and the two schemes are the same again.

6.3. THE TRANSLATION PROCEDURE

6.3.1. *Making Non-redundant LKE-proofs*

In order to translate a LKE-tree into a natural deduction proof the LKE-tree refutation should be non-redundant. Many steps may be made before it is known whether they are needed or not, so the resulting LKE-tree should be pruned, removing the steps which turn out to be unnecessary.

Definition 11 A LKE-tree is *non-redundant* if each node[12] is used in some step – that is, it either contributes to closure or one of its descendants does. The set of *used nodes* is the least set satisfying the following conditions: We inductively define the notion of a *used* node in an LKE-tree:

- A node is *used* if at least one of its formulas is a premise of a closure step
- If n is a used node and m is a premise of the rule that generated n then m is used.

To generate a non-redundant LKE-tree we use the following procedure:

PNR - *Procedure to generate a non-redundant LKE-tree*

1. Using the definition of a used node we can form the set of *used* nodes in a LKE-tree.
2. The remaining nodes are not used and may be deleted.
3. If one of the nodes of an application of the PB-rule is deleted then the whole sub-tree below it closes using the remaining nodes in that branch. So the PB application turns out to be unnecessary and the sub-tree beneath the other node of the PB application may be deleted. The procedure PNR is called recursively on the remaining tree.

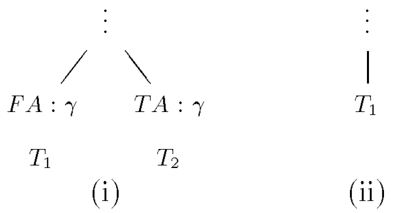

Figure 6.

[12] A node is either an assumption or the result of an LKE-rule application. So a node may contain more than one formula – e.g. the conclusion of the EF→ rule.

For example, in Figure 6(i) suppose F$A : \gamma$ is unused. Then the tree can be reduced to Figure 6(ii) by removing the right branch altogether, since T_1 can still be correctly closed.

Since PNR only deletes nodes it must eventually terminate giving a non-redundant tree. Note that by definition of used node the tree will still close.

6.3.2. *Deriving a Natural Deduction proof*

In this section we describe a procedure called PND which generates from a given non-redundant proper LKE-tree a Natural Deduction proof.

Definition 12 A *proper closed* LKE-tree is a closed tree containing only ground labels and in which no atomic label introduced for the first time in a branch by an application of $\rightarrow\mathcal{I}$ or $\otimes\mathcal{E}$ occurs in the label of any PB-formula in the tree above that step.

Lemma 3 *If a closed LKE-tree exists then a proper closed tree exists.*

Proof: This follows from the completeness of the "unliberalised" rules[13] for EF$\rightarrow$ and ET$\otimes$, as shown in (D'Agostino and Gabbay, 1994). □

We also show that the procedure PND is correct, i.e. given a LKE-refutation there exists a corresponding ND proof generated by the PND interpretation. This will guarantee, together with the completeness of the LKE system, that the ND rules are also complete.

PND - *Procedure to form a Natural Deduction proof from a proper non-redundant LKE-tree*

The proof procedure is applied to an LKE-refutation step by step. Depending on the type of LKE rules used, a different action is taken:

closure Use the $\surd$-rule. The closure constraint becomes a RC.

PB Complete the translation of the left-hand (F branch) within a box and write the conclusion below the box (corresponds to the T-formula of the right-hand branch. The contents of the box are not needed any more.) Then complete the translation of the right-hand branch. (Propagate any substitution that is found for the free variables.)

ET$\rightarrow$ The two premises will already be in the proof and available to the current box, so apply $\rightarrow\mathcal{E}$.

EF$\rightarrow$ Apply $\rightarrow\mathcal{I}$ and put the T-formula at the top and the F-formula at the bottom of the new box.

EF$\otimes$ Suppose the major premise of the step is $FA \otimes B : x$. Introduce two boxes. The left one will contain the proof of $TA : \gamma_1$. This can be

[13] As we have seen in Section 5.1 and Section 5.3 there are two versions of the rules EF$\rightarrow$ and EF$\otimes$. The "unliberalised" version requires the atomic label a introduced by the rule application to be a new label (i.e. not occurring anywhere else in the tree).

obtained immediately by the $\surd$-rule using the T-premise of the step which is of the form $TA : y$, and instantiating γ_1 to y. The right box will contain the proof of the conclusion of the LKE-rule, (i.e. $FB : x/y$), but with the label γ_2 and the RC $\gamma_1 \circ \gamma_2 \sqsubseteq x$.

ET$\otimes$ Suppose the major premise of the step is $TA \otimes B : x$. Add the two conclusions $TA : a$ and $TB : b$ and the IC $a \circ b \sqsubseteq x$. This IC implies $b \sqsubseteq x/a$, and $TB : b$ implies $TB : x/a$ as used in the LKE-tree.

The initial F-formula of the LKE-tree is at the bottom of the proof.

If PND is applied to the tree in Figure 7 then the Natural Deduction proof in Figure 2 is obtained. The left branch of the tree closes under the

$$
\begin{array}{c}
FA \rightarrow ((A \otimes (A \otimes A \rightarrow B)) \rightarrow B) : 1 \\
| \\
TA : a \\
| \\
F(A \otimes (A \otimes A \rightarrow B)) \rightarrow B) : a \\
| \\
TA \otimes (A \otimes A \rightarrow B) : b \\
| \\
TA : a \\
| \\
FB : a \circ b \\
| \\
TA \otimes A \rightarrow B : b/a \\
\diagup \qquad \diagdown \\
\begin{array}{cc}
FA \otimes A : \gamma & TA \otimes A : \gamma \\
| & | \\
FA : \gamma/a & TB : (b/a) \circ \gamma \\
\times & \times
\end{array}
\end{array}
$$

Figure 7. LKE-tree that gives rise to proof in Figure 2

condition $a \sqsubseteq \gamma/a$, which is satisfied for $\gamma = a \circ a$. The right branch closes with the condition $(b/a) \circ a \circ a \sqsubseteq a \circ b$, which also holds by definition of / and commutativity of $\circ$. (In Figure 2 these conditions became $a \circ a \sqsubseteq \gamma_3$ and $c \circ a \circ a \sqsubseteq a \circ b$).

It seems likely that a reverse translation method could be developed: given a natural deduction proof derive a LKE-tree so showing the soundness of the Natural Deduction proofs.

6.3.3. *Correctness of* PND

We show here that the PND procedure is correct – i.e given a LKE-tree there exists a corresponding ND proof generated by PND. To do so we will restrict ourselves, without loss of generality, to "atomic closures" only and show first, as an intermediate result, that in such LKE-trees all the F-conclusions of EF$\rightarrow$ rule applications are used. This will allow us to prove that there exists a resulting Natural Deduction proof.

Definition 13 Let X be a formula. We define the *complexity* of X, written $C(X)$, as follows. $C(X) = 0$ if X is an atomic formula. $C(X) = 1 + \max\{C(Y),\ C(Z)\}$ if $X = Y op Z$, where Y and Z are formulae and *op* is one of the two binary operators $\rightarrow$, $\otimes$.

Let us say that an LKE-tree is *atomically closed* if the closure rule is restricted to atomic formulae.

Lemma 4 *If there is a closed LKE-tree $\mathcal{T}$ for a set* $\mathbf{F}$ *of LS-formulae, then there is an atomically closed LKE-tree $\mathcal{T}'$ for* $\mathbf{F}$.

Proof: The proof is by induction on the complexity of the formulae used in the closure rule in a branch of a given LKE-tree.

Base Case: Trivial.

Inductive Hypothesis: Suppose that a given LKE-tree in which all branch closures occur between formulas with complexity $\leq k$ can be reduced to a tree in which the branch closures occur between atomic formulas only.

Inductive Step: Consider a tree in which the maximum complexity of formulas involved in branch closures is $k+1$. There are two cases: (i) the closure formulae are of the form $TX \rightarrow Y : x$ and $FX \rightarrow Y : y$ with $x \sqsubseteq y$, or (ii) the closure formulae are of the form $TX \otimes Y : x$ and $FX \otimes Y : y$, with $x \sqsubseteq y$. In both cases the closure can be reduced to be between formulas of complexity $\leq k$, namely Y, as shown below. Then by the inductive hypothesis the resulting tree can be reduced to one in which branch closure occurs between atomic formulae.

(i)			(ii)
	$FX \rightarrow Y : x$	$TX \otimes Y : x$	
	$FX \rightarrow Y : y$	$FX \otimes Y : y$	
	$TX : a$	$TX : a$	
	$FY : y \circ a$	$TY : x/a$	
	$TY : x \circ a$	$FY : y/a$	

(Note that $TX : a$ may already have occurred.) Tableau (i) closes because by hypothesis $x \sqsubseteq y$ and then $x \circ a \sqsubseteq y \circ a$. Analogously, in LKE-tree (ii) the hypothesis $x \sqsubseteq y$ implies that $a \circ (x/a) \sqsubseteq y$, which is equivalent to $x/a \sqsubseteq y/a$. □

Definition 14 The T-formula resulting from the application of a EF$\rightarrow$ rule to a labelled formula X is called the *T-conclusion* of X and the resulting F-formula is called the *F-conclusion* of X.

Theorem 5 *Let $\mathcal{T}$ be a non-redundant proper LKE-tree. Every F-conclusion resulting from the application of the EF$\rightarrow$ rule to some used formula is also used in $\mathcal{T}$.*

Proof outline: The proof is by contradiction. Let B be a branch of $\mathcal{T}$ in which the F-conclusion of a used labelled formula X is not used. Since X is not used in B the T-conclusion must be used. Let $TA : a$ be the T-conclusion of X. Note that the formula $TA : a$ will not have previously been introduced in B by a EF$\rightarrow$ or ET$\otimes$ application, else again X would not have been used. Therefore, given also the fact that $\mathcal{T}$ is proper, the label a occurs in B only in formulae below the introduction of a. Applications of rules using $TA : a$ are either ET$\rightarrow$ or EF$\otimes$ steps. The first yields a T-formula with label of the form $x \circ a$, and the second an F-formula with label of the form x/a. These forms of labels may subsequently appear in other terms as follows (y and z are labels):

1. $y \circ a$ (after ET$\rightarrow$)
2. $(y \circ a)/b$ (after ET$\rightarrow$, ET$\otimes$)
3. $x/(y \circ a)$ (after ET$\rightarrow$, EF$\otimes$)
4. $(x/(y \circ a))/(z \circ a)$ $(= x/(y \circ a \circ z \circ a))$
   (after ET$\rightarrow$, EF$\otimes$, ET$\rightarrow$, EF$\otimes$)
5. $x/(y \circ (z \circ a)/b)$ (after ET$\rightarrow$, ET$\otimes$, ET$\rightarrow$, EF$\otimes$)

Forms 1 and 2 label T-formulae and forms 3-5 label F-formulae. In 3-5 a does not occur in x. For T-formulae a occurs in the numerator and for F-formulae a occurs in the denominator. Subsequent closure in a branch using such steps would involve labels in the following combinations:

- 1 with either 3,4, 5 or a term labelling an F-formula not including a
- 2 with either 3,4, 5 or a term labelling an F-formula not including a
- a term labelling a T-formula with no occurrence of a with either 3,4, 5 or another term labelling an F-formula with no a

In the case of 1 with 3, for example, it is required to satisfy $w_1 \sqsubseteq w_2$, where $w_1 = U_1 a$ and $w_2 = x/(U_2 a)$ (where U_1 and U_2 are concatenations of labels). But since a does not occur in x, this cannot hold. We can reason analogously and reach a contradiction in all cases. Hence the F-conclusion of X is used in the branch B. $\Box$

Definition 15 Let $\mathcal{T}$ be a non-redundant LKE-tree. We define a *closure formula* in a branch B of $\mathcal{T}$ to be the atomic formula $FX : x$ used within the closure of B. F_1 *directly supports* F_2 ($F_1 \ll F_2$) if it is the F-formula resulting from EF$\rightarrow$ or EF$\otimes$ rules applied to F_2. The *root* R of F_1 is the terminal formula of the (finite) chain $F_1 \ll \ldots \ll R$.

Lemma 5 *Let $\mathcal{T}$ be a non-redundant proper LKE-tree. The root of the closure formula in a branch B is either the F-formula of the PB-step at greatest depth in B or, if there is no such formula, the initial F-formula.*

Proof outline: The proof is by induction on the number of PB-applications.
Base Case: Suppose there are no PB-steps, then by Theorem 5 every F formula in the single branch is used. Hence, in a closed branch B, the chain beginning with the single closure formula must terminate at the initial F formula.
Inductive Step: Suppose the lemma holds for $\leq k$ PB-steps. Let $\mathcal{T}$ be a non-redundant proper LKE-tree with $k+1$ PB-applications in a closed branch $B1$. Consider the PB-step at greatest depth in $B1$ (i.e. the PB-step in which the closure in its two branches uses no other PB-steps). T has the form shown in Figure 8(i), in which B_1 is the rest of the branch above the considered application of PB. If $FX : x$ is not the root of the closure

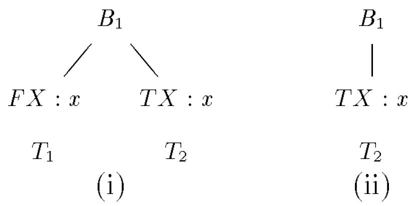

Figure 8.

formula in T_1, then we can use Theorem 5 to show $FX : x$ is not used, which contradicts the non-redundancy of $\mathcal{T}$. The restricted LKE-tree of Figure 8(ii) uses k applications of PB and so, by the inductive hypothesis, the lemma holds. $\square$

Theorem 6 *Let $\mathcal{T}$ be a non-redundant proper LKE-tree. There exists an equivalent ND proof obtained by applying the* PND *procedure to $\mathcal{T}$.*

The above theorem can be proved by showing that the PND procedure allows each ND rule to be successfully applied. By Lemma 5 the closure below the root F-formula of an application of a PB rule in $\mathcal{T}$ uses only F-formulae derived from the root. Moreover, these used F-formulae form a chain starting from the closure formula. This guarantees that the PND procedure will generate a successful sub-proof within the box of the lemma rule, where the innermost goal corresponds to the closure formula and the last conclusion corresponds to the root formula (the chain of F-formulae appears in the ND sub-proof in the reverse order). Moreover, T-formulae will appear in the ND proof in the same order as in the tree, so the T-formulae necessary for the application of $\to\mathcal{E}$, $\otimes\mathcal{I}$ and $\surd$ rules will be available in the ND proof. Finally, our earlier discussion shows that the constraint implication (13) consisting of ICs introduced by $\otimes\mathcal{E}$ and RCs introduced by $\otimes\mathcal{I}$ and $\surd$ will be satisfied.

References

Abrusci, V.M. (1991). Phase semantics and sequent calculus for pure noncommutative classical linear propositional logic. *Journal of Symbolic Logic* 56:1403–1451.

Avron, A. (1988). The semantics and proof theory of linear logic. *Theoretical Computer Science* 57:161–184.

Broda, K.; Eisenbach, S.; Khoshnevisan, H.; and Vickers, S. (1994). *Reasoned Programming.* Prentice Hall.

Broda, Krysia; D'Agostino, Marcello; and Mondadori, Marco (1996). A solution to a problem of popper. To appear in *The Epistemology of Karl Popper*, Kluwer.

D'Agostino, Marcello and Gabbay, Dov M. (1994). A generalization of analytic deduction via labelled deductive systems. part I: Basic substructural logics. *Journal of Automated Reasoning* 13:243–281.

D'Agostino, Marcello and Mondadori, Marco (1994). The taming of the cut. *Journal of Logic and Computation* 4:285–319.

D'Agostino, Marcello (1992). Are tableaux an improvement on truth-tables? *Journal of Logic, Language and Information* 1:235–252.

Dôsen, Kosta (1988). Sequent systems and groupoid models I. *Studia Logica* 47:353–385.

Dôsen, Kosta (1989). Sequent systems and groupoid models II. *Studia Logica* 48:41–65.

Dôsen, Kosta (1993). A historical introduction to substructural logics. In Heister, P. Schroeder and Dôsen, Kosta, editors, *Substructural Logics.* Oxford University Press. 1–31.

Dunn, Michael (1986). Relevance logic and entailment. In Gabbay, Dov M. and Guenthner, F., editors, *Handbook of Philosophical Logic*, volume III. Kluwer Academic Publishers. chapter 3, 117–224.

Gabbay, Dov M. and Ohlbach, Hans J. (1993). An algebraic fine structure for logical systems. In Gabbay, Dov and Guenthner, Franz, editors, *What is a Logical System?* Oxford University Press. To appear.

Gabbay, Dov M. and Ohlbach, Hans J. (1993). From hilbert calclus to possible-world semantics. In Broda, Krysia, editor, *Proceedings of ALPUK Logic Programming Conference 1992.* Springer. 218–252. Lecture Notes in Computer Science.

Gabbay, Dov M. (1992). How to construct a logic for your application. In Ohlbach, Hans Juergen, editor, *GWAI-92: Advances in Artificial Intelligence (LNAI 671).* Springer. 1–30.

Gabbay, Dov (1993). Fibred semantics and the weaving of logics, I. Technical Report, University of Stutgart.

Gabbay, Dov M. (1993). Classical versus non-classical logics. In Gabbay, Dov; Hogger, Chris; and Robinson, J.H., editors, *Handbook of Logic in AI and Logic Programming.* Oxford University Press. To appear.

Gabbay, Dov M. (1993). General theory of structured consequence relations. In Heister, P. Schroeder and Dôsen, Kosta, editors, *Substructural Logics.* Oxford University Press. 109–151.

Gabbay, D.M. (1994). Labelled deductive systems, volume 1 - foundations. Technical Report MPI-I-94-223, Max-Planck-Institut Für Informatik. Preliminary partial draft of a book intended for Oxford University Press.

Gentzen, Gerhard (1935). Unstersuchungen über das logische Schliessen. *Math. Zeitschrift* 39:176–210. English translation in (Szabo, 1969).

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science* 50:1–102.

Howard, William A. (1980). The formuale-as-types notion of construction. In Seldin, J.P. and Hindley, J.R., editors, *To H.B. Curry: Essays on Combinatory Logics, Lambda Calculus and Formalism.* Academic Press, London.

Jaśkowski, (1934). On the Rules of Suppositions in Formal Logics. *Studia Logica* 1.

Lambek, J. (1958). The mathematics of sentence structure. *Amer. Math. Monthly* 65:154–169.

Ono, H. (1993). Semantics for substructural logics. In Schroeder-Heister, Peter, editor, *Substructural Logics.* Oxford University Press. 259–291.

Prawitz, D. (1965). *Natural Deduction. A Proof-Theoretical Study.* Almqvist & Wilksell, Uppsala.

Russo, A. (1996). Generalising propositional modal logics using labelled deductive systems. In *Proceedings of Frontiers of Combining Systems '96.*

Sambin, G. (1993). The semantics of pretopologies. In Schroeder-Heister, Peter, editor, *Substructural Logics.* Oxford University Press. 293–307.

Szabo, M., editor (1969). *The Collected Papers of Gerhard Gentzen.* North-Holland, Amsterdam.

Tarski, Alfred (1930). Fundamentale begriffe der methodologie der deduktiven wissenschaften, I. *Monatshefte für Mathematik und Physik* 37:361–404.

Tarski, Alfred (1930). Über einige fundamentale begriffe der metmathematik. *Comptes Rendus des Séances de la Société des Sciences et des Lettres de Varsovie* 23:22–29.

Urquhart, Alasdair (1972). Semantics for relevant logic. *The Journal of Symbolic Logic* 37:159–170.