Chapter 1

Labelled Natural Deduction

1.1 The rôle of the labels

Unlike axiomatic systems, natural deduction proofs need not start from axioms. More interested in the *structure* of proofs, Gentzen conceived his natural deduction as a logical system which would make explicit the *structural properties of connectives*. Inference rules would be made of not simply premises and conclusions (as inference rules in Hilbert-style axiomatic systems happened to be, e.g. *modus ponens*), but also of *assumptions*. They would also be framed into a pattern of *introduction* and *elimination* rules, with a logical *principle of inversion* underlying the harmony between those two kinds of rules. Connectives which behaved like conditionals (most notably, implication, but also the universal quantifier) would have *introduction* rules doing the job of *withdrawing assumptions* in favour of the corresponding conditional statement.

As defined in the logic literature, a *valid* statement would be one which would rely on no assumptions. Now, in order to test for the validity of statements using Gentzen's method of natural deduction, one would need to make sure that, by the time a deduction of the statement was achieved, *all* assumptions had been withdrawn. Thus, whenever we were to construct proofs in natural deduction, we would need to look at the rules which could withdraw assumptions. This would only be too natural, because when starting from hypotheses and arriving at a certain thesis we need to say that the hypotheses imply the thesis. So, we need to look at the rules of inference which allow us to 'discharge' assumptions (hypotheses) without introducing further assumptions. As we mentioned above, it so happens that the *introduction* rules for the conditionals (namely, implication, universal quantifier, necessity) do indeed perform the task for us. They allow us to 'get rid of the hypotheses' by making a step to 'hypotheses imply thesis'. Let us look at the introduction rules for implication in the plain natural deduction style:

$\rightarrow$ *-introduction*

$$\frac{\begin{array}{c}[A]\\ B\end{array}}{A \rightarrow B}$$

Note that the hypothesis 'A' is discharged, and by the introduction of the implication, the conclusion '$A \rightarrow B$' (i.e. hypothesis 'A' implies thesis 'B') is reached.

Now, if we introduce labels alongside formulas this 'discharge' of hypotheses will be reflected on the label of the intended conclusion by a device which makes the arbitrary name introduced as the label of the corresponding assumption 'lose its identity', so to speak. It is the device of 'abstracting' a variable from a term containing one or more 'free' occurrences of that variable. So, let us look at how the rule given above looks like when augmented by putting labels alongside formulas:

$$\frac{\begin{array}{c}[x : A]\\ b(x) : B\end{array}}{\lambda x.b(x) : A \rightarrow B}$$

Notice that now by the time we reach the conclusion the arbitrary name 'x' loses its identity because the abstractor 'λ' binds its free occurrences in the term 'b', which in its turn may have one or more free occurrences of 'x' (the notation '$b(x)$' indicates that 'b' is a functional term which depends on 'x').

The moral of the story here is that the last inference rule of any complete proof must be the introduction rule of a conditional, simply because those are the rules which do the job we want: discharging assumptions already made, without introducing any further assumptions. We are now speaking in more general terms ('conditional', rather than 'implication') because the introduction rules for the universal quantifier and the necessitation connectives, namely:

$\forall$*-introduction* $\quad$ $\Box$*-introduction*

$$\frac{\begin{array}{c}[x : D]\\ f(x) : P(x)\end{array}}{\Lambda x.f(x) : \forall x^D.P(x)} \qquad \frac{\begin{array}{c}[\mathbb{W} : \mathcal{U}]\\ F(\mathbb{W}) : A(\mathbb{W})\end{array}}{\Lambda \mathbb{W}.F(\mathbb{W}) : \Box A}$$

also discharge old assumptions without introducing new ones. (The treatment of 'necessity' — '$\Box$' — by means of the functional interpretation will be sketched in Chapter 7, and is given in more detail in [de Queiroz and Gabbay (1997)].)

Our motto here is that all labelled assumptions must be discharged by the end of a deduction, and we should be able to check this very easily just by looking at the label of the intended conclusion and check if all 'arbitrary' names (labels) of hypotheses are bound by any of the available abstractors. (As we will see later on, abstractors other than 'λ' and 'Λ' will be used. We have already mentioned '$\acute{}$', and 'ε'.) So, in a sense our proofs will be *categorical* proofs, to use a terminology from [Anderson and Belnap Jr. (1975)].[1] The connection with the realisability interpretation will be made in the following sense: e realises P iff e is a *complete* object (no free-variables). For stronger logics (e.g. classical positive logic) there will be additional ways of binding free variables of the label expression which may establish an extended harmony between the functional calculus on the labels and the logical calculus on the formulas. (Here we have in mind the *generalised reductio ad absurdum* defined in [Gabbay and de Queiroz (1992)], where a λ-abstraction binds a variable occurring as function and as argument in the label expression, and there is no introduction of an implication.)

The device of variable-binding, and the idea of having terms representing incomplete 'objects' whenever they contain free variables, were both introduced in a systematic way by Frege in his *Grundgesetze*. As early as 1893 Frege developed in his *Grundgesetze I* what can be seen as the early origins of the notions of *abstraction* and *application*, when showing techniques for transforming functions (expressions with free variables) into value-range terms (expressions with no free variables) by means of an 'introductory' operator of abstraction producing the *Werthverlauf* expression,[2] e.g. '$\acute{\varepsilon} f(\varepsilon)$', and the effect of its corresponding 'eliminatory' operator '$\cap$' on a *value-range* expression.[3]

The idea of forming *value-range* function-terms by abstracting from the corresponding free variable is in fact very useful in representing the handling of

[1]"A proof is *categorical* if all hypotheses in the proof have been discharged by use of $\rightarrow$-I, otherwise *hypothetical*; and A is a *theorem* if A is the last step of a categorical proof." [Anderson and Belnap Jr. (1975)] (p. 9).

[2][Frege (1893)] (§3, p. 7), translated as *course-of-values* in [Furth (1964)] (p. 36), and *value-range* in most other translations of Frege's writings, including the translation of [Frege (1891)] (published in [McGuinness (1984)]) where the term first appeared.

[3]Cf. [Frege (1893)] (§34, p. 52ff), (translated in [Furth (1964)] (p. 92)):

"(...) it is a matter only of designating the value of the function $\Phi(\xi)$ for the argument Δ, i.e. $\Phi(\Delta)$, by means of 'Δ' and '$\acute{\varepsilon}\Phi(\varepsilon)$'. I do so in this way:

$$`\Delta \cap \acute{\varepsilon}\Phi(\varepsilon)'$$

which is to mean the same as '$\Phi(\Delta)$'."
(Note the similarity to the rule of functional *application*, where 'Δ' is the argument, '$\acute{\varepsilon}\Phi(\varepsilon)$' is the function, and '$\cap$' is the application operator 'APP'.)

Expressing how important he considered the introduction of a variable-binding device for the functional calculus (recall that the variable-binding device for the logical calculus had been introduced earlier in *Begriffsschrift*), Frege says:

"The introduction of a notation for courses-of-values [value-ranges] seems to me to be one of the most important supplementations that I have made of my *Begriffsschrift* since my first publication on this subject." [*Grundgesetze I*, §9, p. 15f.]

assumptions within a natural deduction style calculus. In particular, when the natural deduction presentation system is based on a 'labelling' mechanism the binding of free variables in the labels corresponds to the discharge of respective assumptions. In the sequel we shall be using 'abstractors' (such as 'λ' in '$\lambda x.f(x)$') to bind free-variables and discharge the assumption labelled by the corresponding variable.

We consider a formula to be a theorem of the logical system if it can be derived in a way that its corresponding 'label' contains no free variable, i.e. the deduction rests on no assumption. In other words, a formula is a theorem if it admits a categorical proof to be constructed.

1.1.1 *Dividing the tasks: A functional calculus on the labels, a logical calculus on the formula*

We have seen that the origins of variable binding mechanisms, both on the formulas of logic (the propositions) and on the expressions of the functional calculus (the terms), go back at least as far as Frege's early investigations on a 'language of concept writing'. Although the investigations concerned essentially the establishment of the basic laws of logic, for Frege the functional calculus would have the important rôle of demonstrating that arithmetic could be formalised simply by defining its most basic laws in terms of rules of the 'calculus of concept writing'. Obviously, the calculus defined in *Begriffsschrift* [Frege (1879)], in spite of its functional style, was primarily concerned with the 'logical' side, so to speak. The novel device of binding free variables, namely the universal quantifier, was applicable to propositional functions. Thus, *Grundgesetze* [Frege (1893, 1903)] was written with the intention of fulfilling the ambitious project of designing a language of concept-writing which could be useful to formalise mathematics. Additional mechanisms to handle the functional aspects of arithmetic (e.g. equality between number-expressions, functions over number-expressions, etc.) had to be incorporated. The outcome of Frege's second stage of investigations also brought pioneering techniques of formal logic, this time with respect to the handling of functions, singular terms, definite descriptions, etc. An additional mechanism of variable binding was introduced, this time to bind variables of functional expressions, i.e. expressions denoting individuals, not truth-values.

Summarising, we can see the pioneering work of Frege in its full significance if we look at the two sides of formal logic he managed to formulate a calculus for

(1) the 'logical' calculus on formulas (*Begriffsschrift*)
(2) the 'functional' calculus on terms (*Grundgesetze*)

As a pioneer in any scientific activity one is prone to leave gaps and loopholes to be later filled by others. It happened with Frege that a big loophole was discovered earlier than he would himself have expected: Russell's discovery of the antinomies of his logical notion of set was a serious challenge. There may be several ways of explaining why the resulting calculus was so much susceptible to that sort of challenge. We feel particularly inclined to think that the use of devices which were designed to handle the so-called 'objects', i.e. expressions of the functional calculus, ought to have been kept apart from, and yet harmonised with, the logical calculus on the formulas. Thus, here we may start wondering what might have been the outcome had Frege kept the two sides separate and yet harmonious.

Let us for a moment think of a connection to another system of language analysis which would seem to have some similarity in the underlying ontological assumption, with respect to the idea of dividing the logical calculus into two dimensions, i.e. functional versus logical. The semantical framework defined in Montague's [Montague (1970)] intensional logic makes use of a distinction among the semantic *types* of the objects handled by the framework, namely e, t and s, in words: *entities*, *truth-values*, and *senses*. The idea was that logic (language) was supposed to deal with objects of three kinds: names of entities, formulas denoting truth-values, and possible-worlds/contexts of use. Now, here when we say that we wish to have the bi-dimensional calculus, we are saying that the entities which are *namable* (i.e. individuals, possible-worlds, etc.) will be dealt with separately from (yet harmoniously with) the logical calculus on the formulas, by a calculus of functional expressions. Whereas the variables for individuals are handled 'naturally' in the interpretation of first-order logic with our labelled natural deduction, the introduction of variables to denote *contexts*, or *possible-worlds* (structured collection of labelled formulas), as in [de Queiroz and Gabbay (1997)], is meant to account for Montague's *senses*.

1.1.2 *Reassessing Frege's two-dimensional calculus*

In our attempt to reassess the benefits of having those two sides working together, we would like to insist on the two sides being treated separately. Thus, instead of binding variables on the formulas with the device of forming 'value-range' expressions as Frege does,[4] we shall have a clear separation of functional versus

[4]Cf. the following opening lines of *Grundgesetze I*, §10:

"Although we laid it down that the combination of signs '$\acute{\varepsilon}\Phi(\varepsilon) = \acute{\alpha}\Psi(\alpha)$' has the same denotation as '$\overset{\boldsymbol{a}}{\smile}\ \Phi(\boldsymbol{a}) = \Psi(\boldsymbol{a})$', this by no means fixes completely the denotation of a name like '$\acute{\varepsilon}\Phi(\varepsilon)$'."

Note that both the abstractor ' ' ' and the universal quantifier '$\smile$' are used for binding free variables of formulas of the logical calculus such as 'Φ' and 'Ψ'. In our labelled natural deduction we shall take the separation 'functional versus logical' more strictly than Frege himself did. While the logical devices. We still want to have the device of forming propositional functions, so we still need to have the names of variables of the functional calculus being carried over to take part in the formulas of the logical side. That will be dealt with accordingly when we describe what it means to have predicate formulas in a labelled system. Nevertheless, abstractors shall only bind variables occurring in expressions of the functional calculus, and quantifiers shall bind variables occurring in formulas of the logical calculus. For example, in

$$\begin{array}{cc} \forall\textit{-introduction} & \exists\textit{-introduction} \\ \dfrac{\begin{array}{c}[x : D] \\ f(x) : P(x)\end{array}}{\Lambda x.f(x) : \forall x^D.P(x)} & \dfrac{a : D \qquad f(a) : P(a)}{\varepsilon x.(f(x), a) : \exists x^D.P(x)} \end{array}$$

whilst the abstractors 'Λ' and 'ε' bind variables of the functional calculus, the quantifiers '$\forall$' and '$\exists$' bind variables of the logical calculus, even if the same variable name happens to be occurring in the functional expression as well as in the logical formula.

Notice that although we are dealing with the two sides independently, the harmony seems to be maintained: to each discharge of assumption in the logical calculus there will correspond an abstraction in the functional calculus. In the case of our quantifier rules, we observe that the *introduction* of the universal quantifier is made with the arbitrary name x being bound in both sides (functional and logical) at the same time. In the existential case the 'witness' a is kept unbound in the functional calculus, whilst in the formula the binding is performed.

This is not really the place to discuss the paradoxes of Frege's formalised set theory, but it might be helpful to single out one particularly relevant facet of his 'mistake'. First, let us recall that the development of mechanisms for handling both sides of a calculus of concept writing, namely the logical and the functional, would perhaps recommend special care in the harmonising of these two sides. We all know today (thanks to the intervention of the likes of M. Furth [Furth (1964)], P. Aczel [Aczel (1980)], M. Dummett [Dummett (1973, 1991a)], H. Sluga [Sluga (1980)], and others) that one of the fundamental flaws of Frege's attempt to put the two sides together was the so-called 'Law V' of *Grundgesetze*, which did exactly what we shall avoid here in our 'functional interpretation' of logics: using functions where one should be using propositions, and vice versa. The 'Law V' was stated as follows:

$$\vdash (\acute{\varepsilon} f(\varepsilon) = \acute{\alpha} g(\alpha)) = (\overset{\boldsymbol{a}}{\smile} f(\boldsymbol{a}) = g(\boldsymbol{a}))$$

abstractors will be used to bind variables in the functional calculus, the quantifiers will be used to bind variables in the logical calculus. Obviously, variables may be occurring in both 'sides', but in each side the appropriate mechanism will be used accordingly.

Here we have equality between terms — i.e. $\acute{\varepsilon}f(\varepsilon) = \acute{\alpha}g(\alpha)$ and $f(\boldsymbol{a}) = g(\boldsymbol{a})$ — on par with equality between truth-values — i.e. the middle equality sign.

In his thorough analysis of Frege's system, Aczel makes the necessary distinction by introducing the sign for propositional equality [Aczel (1980)]:

$$(\lambda x.f(x) \doteq \lambda x.g(x)) \leftrightarrow \forall x.(f(x) \doteq g(x)) \text{ is true}$$

where '$\doteq$' stands for propositional equality, and '$\leftrightarrow$' is to mean logical equivalence (i.e. 'if and only if').[5]

Despite the challenges to his theories of formal logic, Frege's tradition has remained very strong in mathematical logic. Indeed, there is a tendency among the formalisms of mathematical logic to take the same step of 'blurring' the distinction between the functional and the logical side of formal logic. As we have already mentioned, Frege introduced in the *Grundgesetze* the device of binding variables in the functional calculus,[6] in addition to the variable-binding device presented in *Begriffsschrift*, but allowed variables occurring in the formulas to be bound not only by the quantifier(s), but also by a device of the functional calculus, namely the 'abstractors'. One testimony to the strength of Frege's legacy which is particularly relevant to our purposes here is the formalism described in Hilbert and Bernays' [Hilbert and Bernays (1934, 1939)] book where various calculi of singular terms are established. One of these calculi was called the ε-calculus, and consisted of an extension of first-order logic by adding the following axiom schema:

$$(\varepsilon_1) \qquad A(a) \rightarrow A(\varepsilon_x A(x)) \qquad \text{(for any term '}a\text{')}$$
$$(\varepsilon_2) \qquad \forall x.(A(x) \leftrightarrow B(x)) \rightarrow (\varepsilon_x A(x) = \varepsilon_x B(x))$$

where any term of the form '$\varepsilon_x A(x)$' is supposed to denote a term 't' with the property that '$A(t)$' is true, if there is one such term.

Now, observe that the addition of these new axioms has to be proven 'harmless' to the previous calculus, namely the first-order calculus with bound variables, in the sense that no formulas involving only the symbols of the language of the old calculus which was not previously a theorem, is a theorem of the new calculus.

[5] Later in this book we shall be dealing with the problem of handling equality on the 'logical side', so to speak: we demonstrate how to provide an analysis of deduction (Gentzen style, i.e. via rules of *introduction* and *elimination* with appropriate labelling discipline) for a proposition saying that two expressions of the functional calculus denote the same object. In order to explain the properties of this new propositional connective we will be discussing the issues of 'extensional versus intensional' approaches to equality. An analysis of propositional equality via our labelled natural deduction may serve as the basis for a proof theory for descriptions.

[6] In fact, Frege had already introduced the device which he called *Werthverlauf* in his article on 'Function and Concept' [Frege (1891)], which, in its turn, may have been inspired by Peano's functional notation [Peano (1889)].

For that one has to prove the fundamental theorems stating that the new calculus is only a 'conservative extension' the old calculus (First and Second ε-Theorems).

The picture becomes slightly different when we follow somewhat more strictly the idea of dividing, as sharply as we can, the two tasks: let all that has to do with entities to be handled by the functional calculus on the labels, and leave only what is 'strictly logical' to the logical calculus on the formulas. So, in the case of ε-terms, we shall not simply replace an existentially quantified variable in a formula (e.g. 'x' in '$\exists x.A(x)$') by an ε-term involving a formula (e.g. '$A(\varepsilon_x A(x))$'). Instead, we shall use 'ε' as an abstractor binding variables of the functional calculus, as we have seen from our rule of $\exists$-*introduction* shown previously. In other words, we do not have the axioms (or rules of inference) for the existential quantifier plus other axiom(s) for the ε-symbol. We shall be presenting the existential quantifier with its usual logical calculus on the formulas, alongside our ε-terms taking care of the 'functional' side. Therefore

$\exists$-*introduction* $\qquad\qquad$ $\exists$-*elimination*

$$\frac{a : D \qquad f(a) : P(a)}{\varepsilon x.(f(x), a) : \exists x^D.P(x)} \qquad \frac{e : \exists x^D.P(x) \qquad \begin{matrix}[t : D, g(t) : P(t)] \\ d(g,t) : C\end{matrix}}{INST(e, \acute{g}\acute{t}d(g,t)) : C}$$

(side condition: the variables t and g must occur free in the term $d(g,t)$ labelling C).

Notice that here our concern with the 'conservative extension' shall be significantly different from the one Hilbert and Bernays had [Hilbert and Bernays (1934, 1939)]. We have the ε-symbol appearing on the label (the functional side, so to speak), and it is only introduced alongside the corresponding existential formula. (More details of our treatment of the peculiarities of the existential quantifier are given in Chapter 3, as well as in [de Queiroz and Gabbay (1995)].)

1.2 Canonical proofs and normalisation

Since Heyting's definition of each (intuitionistic) logical connective in terms of proof conditions (as opposed to the then usual truth-valuation technique), there emerged a whole tradition within mathematical logic of replacing the declarative concept of truth-functions by its procedural counterpart proof-conditions. By providing a 'language-based' (as opposed to Brouwer's language*less*) explanation of intuitionistic mathematics, Heyting put forward a serious alternative approach to the usual truth-tables-based definitions of logical connectives, which was adequate for a certain tradition in the philosophy of language and philosophy of mathematics, namely the so-called anti-realist tradition.

With the advent of Gentzen's 'mathematical theory of proofs', its corresponding classification of 'natural deduction' inference rules into *introduction* and *elimination*, and the principle (advocated by Gentzen himself) saying that the conditions under which one can assert a logical proposition (formalised by the *introduction* rules) define the meaning of its major connective,[7] an intuitionistic proof-theoretic approach to semantics was given a (meta-)mathematical status. Later, the philosophical basis of this particular approach to intuitionism via proof theory is found in Dummett and Prawitz, its main advocates. In his book on the foundations of (language-based) intuitionistic mathematics, Dummett advocates that "the meaning of each [logical] constant is to be given by specifying, for any sentence in which that constant is the main operator, what is to count as a proof of that sentence, it being assumed that we already know what is to count as a proof of any of the constituents." [Dummett (1977)] (p. 12)

A further refinement of the notion of meaning as being determined by the proof-conditions is given by P. Martin-Lof when he elegantly points out the crucial and often neglected distinction between *propositions* and *judgements*, and introduces the notion of *canonical* (or *direct*) proof. By making an attempt to formalise the basic principles of a particular strand of intuitionism called *constructive* mathematics, as practiced by, e.g. E. Bishop in *Foundations of Constructive Analysis* [Bishop (1967)], his explanations of meaning in terms of *canonical* proofs further advocates the replacement of truth-valuation-based accounts of meaning by a canonical-proof-based one.[8]

---

[7]When commenting on the rôle of the natural deduction rules of *introduction* and *elimination*, Gentzen says:

"The introductions represent, as it were, the 'definitions' of the symbols concerned, and the eliminations are no more, in the final analysis, than the consequences of these definitions." ([Gentzen (1935)], p. 80 of the English translation.)

[8]In a series of lectures entitled 'On the meanings of the logical constants and the justifications of the logical laws', Martin-Lof presents philosophical explanations concerning the distinction between propositions and judgements, as well as the connections between, on the one hand, Heyting's explanation of propositions in terms of proofs rather than truth-values, and, on the other hand, the principle that "the meaning of a proposition is determined by what it is to verify it, or what counts as a verification of it" [Martin-Lof (1985)] (p. 43). Although in those lectures the emphasis appears to be on a sort of phenomenological interpretation of the notions of proposition and judgement, in what concern the formalisation of the concepts, the explanations suggest the important rôle of the definition of direct/canonical proofs for the proof-based account of meaning. In a subsequent paper the interpretation is carried a step further, and the connections with Gentzen's claim is spelled out in a more explicit fashion:

"The intuitionists explain the notion of proposition, not by saying that a proposition is the expression of its truth conditions, but rather by saying, in Heyting's words, that a proposition expresses an expectation or an intention, and you may ask, An expectation or an intention of what? The answer is that it is an expectation or an intention of a proof of that proposition. And Kolmogorov phrased essentially the same explanation by saying that a proposition expresses a problem or task (Ger. *Aufgabe*). Soon

It does not seem unreasonable, however, to say that the use of natural deduction and the semantics of assertability conditions advocated by the intuitionists is not the only way to provide a language-based account of the meaning of logical symbols. If for nothing else, the guiding principle that assertability conditions constitute the main semantical device is clearly not in tune with the proclaimed source of inspiration for the critique of the truth-valuation approach and its replacement by explanations of proof conditions.[9] So, there appears to be room for

---

afterwards, there appeared yet another explanation, namely, the one given by Gentzen, who suggested that the introduction rules for the logical constants ought to be considered as so to say the definitions of the constants in question, that is, as what gives the constants in question their meaning. What I would like to make clear is that these four seemingly different explanations actually all amount to the same, that is, they are not only compatible with each other but they are just different ways of phrasing one and the same explanation." (...) "If you interpret truth conditions in this way, you see that they are identical with the introduction rules for the logical constants as formulated by Gentzen. So I have now explained why, suitably interpreted, the explanation of a proposition as the expression of its truth conditions is no different from Gentzen's explanation to the effect that the meaning of a proposition is determined by its introduction rules."

[Martin-Lof (1987)] (pp. 410, 411)

Cf. also: "The introduction rules say what are the canonical elements (and equal canonical elements) of the set, thus giving its meaning." [Martin-Lof (1984)] (p. 24)

Observe that the principle of 'meaning is determined by the assertability (proof) conditions' is unequivocally advocated. The introduction rules of Gentzen-type natural deduction are said to constitute definitions (as Gentzen himself had advocated earlier). Here one might start to wonder what class of definition (abbreviation, presentation, etc.) these introduction rules would fall into. And indeed, as it stands, the principle of 'introduction rules as definitions' does not seem to find a place in virtually any classification of definitions used in mathematical logic, even less so in, e.g. Frege's [Frege (1914)] classification of definitions into *constructive* and *analytic* [de Queiroz (1987)].

Perhaps it should be remarked here that in spite of the 'unequivocal' position expressed by P. Martin-Lof, there still seems to be room for interpretation. Cf., e.g. the following observation in M. Dummett's *The Logical Basis of Metaphysics*:

"Intuitively, Gentzen's suggestion that the introduction rules be viewed as fixing the meanings of the logical constants has no more force than the converse suggestion, that they are fixed by the elimination rules; intuitive plausibility oscillates between these opposing suggestions as we move from one logical constant to another. Per Martin-Lof has, indeed, constructed an entire meaning-theory for the language of mathematics on the basis of the assumption that it is the elimination rules that determine meaning." [Dummett (1991b)] (p. 280)

[9]Cf., e.g.:

"*We no longer explain the sense of a statement by stipulating its truth-value in terms of the truth-values of its constituents, but by stipulating when it may be asserted in terms of the conditions under which its constituents may be asserted.* The justification for this change is *how we in fact learn to use these statements*: furthermore, the notions of truth and falsity cannot be satisfactorily explained so as to form a basis for an account of meaning once we leave the realm of effectively decidable statements."

[Dummett (1959)] (p. 161)

(The italics is ours.)

Cf. also:

"As pointed out by Dummett, this whole way of arguing with its stress on communication and the role of the language of mathematics is inspired by ideas of Wittgenstein and is very different from Brouwer's rather solipsistic view of mathematics as a languageless activity. Nevertheless, as it seems, the adoption of a different one, such as, for example, instead of advocating the replacement of the truth-valuation systems by explanations of proof-conditions, one can propose to have the explanation of how the *elimination* inferences act on the result of *introduction* steps, i.e. the β-normalisation procedure, as the main semantical device. The normalisation procedure would be looked at, not merely as a meta-mathematical device introduced to prove consistency (as it is usually seen by proof-theorists), but it would be seen as the formal explanation of the 'functionality' of the corresponding logical sign. Although some other principle might be found to be more appropriate, this is, in fact, what we adopt here.

1.2.1 *Canonical proofs*

Instead of Heyting's explanation of the logical constants solely in terms of proofs (or, 'canonical' proofs as in [Martin-Lof (1984)]), the explanations given by the approach to the functional interpretation taken here involve *both* the notion of *canonical proofs* and that of *normalisation of noncanonical proofs*.

The *canonical* proofs are explained as:

a proof of the proposition:	has the canonical form of:
$A_1 \wedge A_2$	$\langle a_1, a_2 \rangle$ where a_1 is a proof of A_1 and a_2 is a proof of A_2
$A_1 \vee A_2$	$inl(a_1)$ where a_1 is a proof of A_1 or $inr(a_2)$ where a_2 is a proof of A_2 ('inl' and 'inr' abbreviate 'into the left disjunct' and 'into the right disjunct', respectively)
$A \rightarrow B$	$\lambda x.b(x)$ where $b(a)$ is a proof of B provided a is a proof of A

it constitutes the best possible argument for some of Brouwer's conclusions. (...)

I have furthermore argued that the rejection of the platonistic theory of meaning depends, in order to be conclusive, on the development of an adequate theory of meaning along the lines suggested in the above discussion of the principles concerning meaning and use. Even if such a wittgensteinian theory did not lead to the rejection of classical logic, it would be of great interest in itself." [Prawitz (1977)] (p. 18)

We have previously endeavoured to demonstrate that the so-called semantics of use advocated by Wittgenstein did not involve simply assertability conditions, but it also accounted for the explanation of (immediate) consequences [de Queiroz (1989)]. So, as it seems, it would be unreasonable to call a theory of meaning based on *assertability conditions* a 'Wittgensteinian theory'.

$\forall x^D.P(x)$	$\Lambda x.f(x)$ where $f(a)$ is a proof of $P(a)$ provided a is an arbitrary individual chosen from the domain D
$\exists x^D.P(x)$	$\varepsilon x.(f(x), a)$ where a is an individual (witness) from the domain D, and $f(a)$ is a proof of $P(a)$

As the reader can easily notice, the explanation of the logical connectives in terms of canonical proofs only cover the rules of *introduction*.[10] They constitute an explanation of the conditions under which one can form a canonical (direct) proof of the corresponding proposition. Its counterpart in Gentzen's natural deduction is the *introduction* rule, now enriched with the 'witnessing' construction which shall be handled by the functional calculus on the labels which we have mentioned above.

Thus, the corresponding formal presentations *a la* natural deduction are, e.g.:

$\wedge$-*introduction*

$$\frac{a_1 : A_1 \qquad a_2 : A_2}{\langle a_1, a_2 \rangle : A_1 \wedge A_2}$$

$\vee$-*introduction*

$$\frac{a_1 : A_1}{inl(a_1) : A_1 \vee A_2} \qquad \frac{a_2 : A_2}{inr(a_2) : A_1 \vee A_2}$$

$\rightarrow$-*introduction*

$$\frac{\begin{array}{c}[x : A] \\ b(x) : B\end{array}}{\lambda x.b(x) : A \rightarrow B}$$

$\forall$-*introduction*[11]

$$\frac{\begin{array}{c}[x : D] \\ f(x) : P(x)\end{array}}{\Lambda x.f(x) : \forall x^D.P(x)}$$

[10] When looking at some of those informal explanations of intuitionistic connectives, such as the one via canonical proofs (or, indeed, the one via a realisability predicate given in [Kleene (1945)]), one is often tempted to see the explanations given for '$\wedge$', '$\rightarrow$' and '$\forall$' as covering the procedures corresponding to *elimination* rules. For the sake of the argument, however, let us stick to the usual intuitionistic account of Heyting's semantics.

[11] Note that in our formulation, where the domain over which one is quantifying is explicitly stated, the *introduction* of the universal quantifier *does* require the discharge of an assumption, namely the assumption which indicates a choice of an arbitrary individual from the domain. This account of the Universal Generalisation does not run into the difficulties related to the classification of $\forall$-*introduction* either as proper or as improper inference rule, because, similarly to the case of $\rightarrow$-*introduction*, this

$\exists$-*introduction*

$$\frac{a : D \qquad f(a) : P(a)}{\varepsilon x.(f(x), a) : \exists x^D.P(x)}$$

When constructing an 'ε'-term, we make an 'inverse' substitution: we replace all the occurrences of 'a' in '$f(a)$' by a new variable 'x' which is bound by the 'εx'-constructor.

1.2.2 *Normalisation*

Within the functional interpretation the operators forming the canonical proofs are usually referred to as the 'constructors' (e.g. 'λ', '$\langle,\rangle$', 'inl/inr', 'Λ', 'ε'), whereas the eliminatory operators which form the noncanonical proofs are referred to as the 'DESTRUCTORS' [de Queiroz and Maibaum (1990)].

To recall our previous discussion, we know that, according to the intuitionistic semantics of canonical proofs, a proposition is characterised by the explanation of the conditions under which one can prove it. The procedure of exhibiting the canonical elements of a type (the canonical proofs of a proposition), which gets formalised by the *introduction* rules, is the key semantical device for the intuitionistic account of meaning. The functional interpretation, however, which accounts for the match between the functional calculus on the labels and the logical calculus on the formulas, does not have to abide by the Heyting-like account of meaning. One can, for example, take the explanation of the convertibility (normalisation) relation as the key semantical device, such as it is done in Tait's intensional interpretation of Godel's *T*, and the result is an account of the meaning of logical signs which does not rely on intuitionistic principles.[12]

---

rule requires the discharge of an assumption and would therefore have to be classified as an improper inference rule if one were to use Prawitz's [Prawitz (1965)] terminology. Cf. Fine's remark on such difficulties concerning the classification of $\forall$-*introduction*:

"Some of the rules require the discharge of suppositions and so have to be classified as improper. Others are so obviously proper that it seems absurd to classify them in any other way. The only real choice concerns universal generalisation ($\forall$); this requires no discharge of suppositions and might, intuitively, be classified as either proper or improper. In considering any proposed account of validity therefore, it must be decided what status this rule is to have." [Fine (1985)] (p. 72).

[12]When explaining the doctrine that a proposition is the type of its proofs, and suggesting that this might have been implicit in Brouwer's writings, Tait says:

"Now, although I shall not have time to explain my views, I believe that, with certain modifications, this idea [propositions are types of their proofs] provides an account of the meaning of mathematical propositions which is adequate, not only for constructive mathematics, but for classical mathematics as well." [Tait (1983)] (p. 182)

This is clearly a departure from Heyting's strictly intuitionistic principles, and, as it seems, it comes as no surprise, given that Tait's semantical instrument is *convertibility* (*normalisation*), rather than canonical proofs.

An important step in the characterisation of the logical connectives which is not covered by the (language-based) intuitionistic account of meaning based on proofs is the explanation of the functionality of the logical sign in the calculus. In the formal apparatus, this means that the explanation as to how the DESTRUCTORS operate on terms built up by the constructors, i.e. the explanation of what (immediate) consequences one can draw from the corresponding proposition, does not play a major semantical rôle for the intuitionists.[13] Within the general (not just intuitionistic) functional interpretation, which we believe might have been implicit in Tait's semantics of convertibility [Tait (1965, 1967)], this aspect is given by the so-called β-normalisation rules. They have the rôle of spelling out the effect of an *elimination* inference on the result of *introduction* steps.

1.2.2.1 β*-type reductions*

The explanation of the normalisation of noncanonical proofs, i.e. those which contain 'redundant' steps identified by an *introduction* inference immediately followed by an *elimination* inference, are framed in the following way (where '$\rhd_\beta$' represents 'β-converts/normalises to'):

$\wedge$*-reduction*

$$\dfrac{\dfrac{a_1 : A_1 \qquad a_2 : A_2}{\langle a_1, a_2\rangle : A_1 \wedge A_2}\wedge\textit{-intr}}{FST(\langle a_1, a_2\rangle) : A_1}\wedge\textit{-elim} \qquad \rhd_\beta \qquad a_1 : A_1$$

$$\dfrac{\dfrac{a_1 : A_1 \qquad a_2 : A_2}{\langle a_1, a_2\rangle : A_1 \wedge A_2}\wedge\textit{-intr}}{SND(\langle a_1, a_2\rangle) : A_2}\wedge\textit{-elim} \qquad \rhd_\beta \qquad a_2 : A_2$$

$\vee$*-reduction*

$$\dfrac{\dfrac{a_1 : A_1}{inl(a_1) : A_1 \vee A_2}\vee\textit{-intr} \quad \begin{array}{cc}[x : A_1] & [y : A_2]\\ d(x) : C & e(y) : C\end{array}}{CASE(inl(a_1), \acute{x}d(x), \acute{y}e(y)) : C}\vee\textit{-elim} \quad \rhd_\beta \quad \begin{array}{c}a_1 : A_1\\ d(a_1/x) : C\end{array}$$

$$\dfrac{\dfrac{a_2 : A_2}{inr(a_2) : A_1 \vee A_2}\vee\textit{-intr} \quad \begin{array}{cc}[x : A_1] & [y : A_2]\\ d(x) : C & e(y) : C\end{array}}{CASE(inr(a_2), \acute{x}d(x), \acute{y}e(y)) : C}\vee\textit{-elim} \quad \rhd_\beta \quad \begin{array}{c}a_2 : A_2\\ e(a_2/y) : C,\end{array}$$

where '´' is an abstractor which forms value-range terms such as '$\acute{x}d(x)$' where 'x' is bound, discharging the corresponding assumption labelled by x.

[13] In an analysis of the relevant aspects of proof-theoretic semantics [de Queiroz (1991)], we have suggested that the *introduction* rules only cover one aspect, namely the grammatical (formational) aspect: they only say how to construct a proof, leaving untouched the aspect as to how to de-construct (challenge) this same proof.

→-reduction

$$\dfrac{a : A \qquad \dfrac{\dfrac{[x : A]}{b(x) : B}}{\lambda x.b(x) : A \to B} \to\textit{-intr}}{APP(\lambda x.b(x), a) : B} \to\textit{-elim} \quad \rhd_\beta \quad \dfrac{a : A}{b(a/x) : B}$$

∀-reduction

$$\dfrac{a : D \qquad \dfrac{\dfrac{[x : D]}{f(x) : P(x)}}{\Lambda x.f(x) : \forall x^D.P(x)} \forall\textit{-intr}}{EXTR(\Lambda x.f(x), a) : P(a)} \forall\textit{-elim} \quad \rhd_\beta \quad \dfrac{a : D}{f(a/x) : P(a)}$$

∃-reduction

$$\dfrac{\dfrac{a : D \quad f(a) : P(a)}{\varepsilon x.(f(x), a) : \exists x^D.P(x)} \exists\textit{-intr} \qquad \dfrac{[t : D, g(t) : P(t)]}{d(g,t) : C}}{INST(\varepsilon x.(f(x), a), \acute{g}\acute{t}d(g,t)) : C} \exists\textit{-elim} \quad \rhd_\beta$$

$$\dfrac{a : D, f(a) : P(a)}{d(f/g, a/t) : C},$$

where '´' is an abstractor which binds the free variables of the label, discharging the corresponding assumptions made in eliminating the existential quantifier, namely the 'Skolem'-type assumptions '$[t : D]$' and '$[g(t) : P(t)]$', forming the value-range term '$\acute{g}\acute{t}d(g,t)$' where both the Skolem-constant 't', and the Skolem-function 'g', are bound. In the ∃-*elimination* the variables 't' and 'g' must occur free at least once in the term alongside the formula 'C' in the premise, and will be bound alongside the same formula in the conclusion of the rule.

It is useful to compare our definition of the β-normalisation rules with the original definitions given by Prawitz for plain natural deduction systems [Prawitz (1965, 1971)]. Whereas in the latter there was the need to refer to whole branches of deductions (which in the Prawitz terminology were referred to as $\Pi_1(a)$, $\Sigma(t)$, $\Sigma_2(a)$, etc.), here we only need to refer to assumptions, premises and conclusions. The relevant information on the dependency of premises from names (variables, constants, etc.) occurring in the assumptions are to be recorded in the label alongside the formula in the respective premise by whatever proof step(s) eventually made from assumptions to premises. It would seem fair to say that this constitutes an improvement on the formal presentation of proof reductions, reflecting the (re-)gain of local control by the use of labels.[14]

[14] Obviously, the first steps towards such improvement was already made by Martin-Lof in the definition of an intuitionistic theory of types [Martin-Lof (1971, 1975b)], but here we want to see it applicable to a wide range of logics.

1.2.2.2 β-equality

By using equality to represent the β-convertibility ('$\rhd_\beta$') relation between terms we can present the *reductions* in the following way[15]:

$\wedge$-β-*equality*

$$FST(\langle a_1, a_2\rangle) =_\beta a_1 \qquad SND(\langle a_1, a_2\rangle) =_\beta a_2$$

$\vee$-β-*equality*

$$CASE(inl(a_1), \acute{x}d(x), \acute{y}e(s_2)) =_\beta d(a_1/x)$$
$$CASE(inr(a_2), \acute{x}d(x), \acute{y}e(s_2)) =_\beta e(a_2/y)$$

$\rightarrow$-β-*equality*

$$APP(\lambda x.b(x), a) =_\beta b(a/x)$$

$\forall$-β-*equality*

$$EXTR(\Lambda x.f(x), a) =_\beta f(a/x)$$

$\exists$-β-*equality*

$$INST(\varepsilon x.(f(x), a), \acute{g}\acute{t}d(g,t)) =_\beta d(f/g, a/t).$$

Remark. Here it is useful to think in terms of 'DESTRUCTORS acting on constructors', especially in connection with the fact that a proof containing an *introduction* inference followed by an *elimination* step is only β-normalisable at that point if the *elimination* has as *major* premise the formula produced by the previous *introduction* step. For example, as remarked by Girard *et al.* [Girard *et al.*

[15]The reader may find it unusual that we are here indexing the (definitional) equality with its kind (β, η, ξ, ζ, etc.). But we shall demonstrate that it makes sense in the context of the functional interpretation to classify (and name) the equalities: one has distinct equalities according to the distinct logical equivalences on the deductions. For example, in the presentation of a set of proof rules for a certain logical connective, the second *introduction* rule is meant to show when two canonical proofs are to be taken as equal, so it is concerned with ξ-equality. The *reduction* rule shows how non-canonical expressions can be brought to normal form, so it is concerned with β-equality. Finally, the *induction* rule shows that by performing an introduction step right after an elimination inference, one gets back to the original proof (and corresponding term), thus it concerns η-equality.

As it will be pointed out later on, it is important to identify the kind of definitional equality, as well as to have a logical connective of 'propositional equality' in order to be able to reason about the functional objects (those to the left hand side of the ':' sign). The connective will have an 'existential' flavour: two referents are verified to be equal if there exists a reason (composition of rewrites) for asserting it. For example, one might wish to prove that for any two functional objects of →-type, if they are equal then their application to all objects of the domain type must result in equal objects of the codomain type.

(1989)],[16] despite involving an $\rightarrow$-*introduction* immediately followed by an $\rightarrow$-*elimination*, the following proof fragment is not β-normalisable:

$$\dfrac{\dfrac{\begin{array}{c}[x : A]\\ b(x) : B\end{array}}{\lambda x.b(x) : A \rightarrow B}\rightarrow\textit{-intr} \qquad c : (A \rightarrow B) \rightarrow D}{APP(c, \lambda x.b(x)) : D}\rightarrow\textit{-elim.}$$

Here the major premise of the *elimination* step is not the same formula as the one produced by the *introduction* inference. Moreover, it is clear that the DESTRUCTOR 'APP' is not acting on the term built up with the constructor '$\lambda x.b(x)$' in the previous step, but it is operating on an unanalysed term 'c'.

1.2.2.3 *η-type reductions*

In a natural deduction proof system there is another way of making 'redundant' steps that one can make, apart from the above '*introduction* followed by *elimination*'.[17] It is the exact inverse of this previous way of introducing redundancies: an *elimination* step is followed by an *introduction* step. As it turns out, the convertibility relation will be revealing another aspect of the 'propositions-are-types' paradigm, namely that there are redundant steps which from the point of view of the definition/presentation of propositions/types are saying that given any arbitrary proof/element from the proposition/type, it must be of the form given by the *introduction* rules. In other words, it must satisfy the '*introduction* followed by an *elimination*' convertibility relation. In the typed λ-calculus literature, this 'inductive' convertibility relation has been referred to as 'η'-convertibility.[18]

---

[16] Chapter Sums in Natural Deduction, Section Standard Conversions.

[17] Some standard texts in proof theory, such as Prawitz' classic survey [Prawitz (1971)], have referred to this proof transformation as 'expansions'. Here we are referring to those proof transformations as *reductions*, given that our main measuring instrument is the label, and indeed the label is reduced.

[18] The classification of those η-conversion rules as inductive rules was introduced by the methodology of defining types used in our reformulated Type Theory described in [de Queiroz and Maibaum (1990)], and first presented publicly in [de Queiroz and Smyth (1989)]. It seems to have helped to give a 'logical' status which they were given previously in the literature.

In a discussion about the Curry–Howard isomorphism and its denotational significance, Girard *et al.* say:

"Denotationally, we have the following (*primary*) equations

$$\pi^1\langle u, v\rangle = u \qquad \pi^2\langle u, v\rangle = v \qquad (\lambda x^U.v)u = v[u/x]$$

together with the *secondary* equations

$$\langle \pi^1 t, \pi^2 t\rangle = t \qquad \lambda x^U.tx = t \quad (x \text{ not free in } t)$$

which have never been given adequate status." [Girard *et al.* (1989)] (p. 16)

Cf. also:

The '$\rhd_\eta$'-convertibility relation then defines the *induction* rules:

$\wedge$-*induction*

$$\dfrac{\dfrac{c : A_1 \wedge A_2}{FST(c) : A_1}\wedge\textit{-elim} \qquad \dfrac{c : A_1 \wedge A_2}{SND(c) : A_2}\wedge\textit{-elim}}{\langle FST(c), SND(c)\rangle : A_1 \wedge A_2}\wedge\textit{-intr} \qquad \rhd_\eta \qquad c : A_1 \wedge A_2$$

$\vee$-*induction*

$$\dfrac{c : A_1 \vee A_2 \qquad \dfrac{[x : A_1]}{inl(x) : A_1 \vee A_2}\vee\textit{-intr} \qquad \dfrac{[y : A_2]}{inr(y) : A_1 \vee A_2}\vee\textit{-intr}}{CASE(c, \acute{x}inl(x), \acute{y}inr(y)) : A_1 \vee A_2}\vee\textit{-elim} \rhd_\eta$$

$$c : A_1 \vee A_2$$

$\rightarrow$-*induction*

$$\dfrac{\dfrac{[x : A] \qquad c : A \rightarrow B}{APP(c, x) : B}\rightarrow\textit{-elim}}{\lambda x.APP(c, x) : A \rightarrow B}\rightarrow\textit{-intr} \qquad \rhd_\eta \qquad c : A \rightarrow B$$

where c does not depend on x.

$\forall$-*induction*

$$\dfrac{\dfrac{[t : D] \qquad c : \forall x^D.P(x)}{EXTR(c, t) : P(t)}\forall\textit{-elim}}{\Lambda t.EXTR(c, t) : \forall t^D.P(t)}\forall\textit{-intr} \qquad \rhd_\eta \qquad c : \forall x^D.P(x)$$

where x does not occur free in c.

$\exists$-*induction*

$$\dfrac{c : \exists x^D.P(x) \qquad \dfrac{[t : D] \qquad [g(t) : P(t)]}{\varepsilon y.(g(y), t) : \exists y^D.P(y)}\exists\textit{-intr}}{INST(c, \acute{g}\acute{t}\varepsilon y.(g(y), t)) : \exists y^D.P(y)}\exists\textit{-elim} \qquad \rhd_\eta \qquad c : \exists x^D.P(x)$$

"Let us note for the record the analogues of $\langle \pi^1 t, \pi^2 t\rangle \rhd t$ and $\lambda x.tx \rhd t$:

$$\varepsilon_{\mathsf{Emp}} t \rhd t \qquad\qquad \delta x.(\iota^1 x)y.(\iota^2 y)t \rhd t$$

Clearly the terms on both sides of the '$\rhd$' are denotationally equal." (Ibid., p. 81.)

Here 'δ' is used instead of '$CASE$', and 'ι^1'/'ι^2' are used instead of 'inl'/'inr' respectively.

Later on, when discussing the coherence semantics of the lifted sum, a reference is made to a rule which we here interpret as the induction rule for $\vee$-types, no mention being made of the rôle such 'equation' is to play in the proof calculus:

"Even if we are unsure how to use it, the equation

$$\delta x.(\iota^1 x)y.(\iota^2 y)t = t$$

plays a part in the implicit symmetries of the disjunction." (Ibid., p. 97.)

By demonstrating that these kind of conversion rules have the rôle of guaranteeing minimality for the non-inductive types such as the logical connectives (not just $\rightarrow$, $\wedge$, $\vee$, but also $\forall$, $\exists$) characterised by types, we believe we have given them adequate status. (That is to say: the rules of η-reduction state that any proof of $A \rightarrow B$, $A \wedge B$, $A \vee B$, will have in each case a unique form, namely $\lambda x.y$, $\langle a, b\rangle$, $inl(a)/inr(b)$, resp.)

In the terminology of [Prawitz (1971)], these rules (with the conversion going from right to left) are called *immediate expansions*. Notice, however, that whilst in the latter the purpose was to bring a derivation in full normal form to expanded normal form where all the minima formulas are atomic, here we are still speaking in terms of *reductions*: the large terms alongside the formulas resulting from the derivation on the left are reduced to the smaller terms alongside the formula on the right. Moreover, the benefit of this change of emphasis is worth pointing out here: whereas in the Prawitz plain natural deduction the principal measure is the degree of formulas (i.e. minimal formulas, etc.) here the labels (or proof constructions) take over the main rôle of measuring instrument. The immediate consequence of this shift of emphasis is the replacement of the notion of *subformula* by that of *subdeduction*, which not only avoids the complications of proving the subformula property for logics with 'Skolem-type' connectives (i.e. those connectives whose *elimination* rules may violate the subformula property of a deduction, such as ∨, ∃, $\doteq$), but it also seems to retake Gentzen's analysis of deduction in its more general sense. That is to say, the emphasis is put back into the deductive properties of the logical connectives, rather than on the truth of the constituent formulas.

Remark. Notice that the mere condition of '*elimination* followed by *introduction*' is not sufficient to allow us to perform an η-conversion. We still need to take into consideration what subdeductions we are dealing with. For example, in:

$$\dfrac{c : A \vee A \quad \dfrac{[x : A]}{inl(x) : A \vee B}\vee\textit{-intr} \quad \dfrac{[y : A]}{inl(y) : A \vee B}\vee\textit{-intr}}{CASE(c, \acute{x}inl(x), \acute{y}inl(y)) : A \vee B}\vee\textit{-elim} \quad \not\rhd_\eta$$

$$c : A \vee A$$

we have a case where an ∨-*elimination* is immediately followed by an ∨-*introduction*, and yet we are not prepared to accept the proof transformation under η-conversion. Now, if we analyse the subdeductions (via the labels), we observe that

$$CASE(c, \acute{x}inl(x), \acute{y}inl(y)) \neq_\eta c$$

therefore, if the harmony between the functional calculus on the labels and the logical calculus on the formulas is to be maintained, we have good enough reasons to reject the unwanted proof transformation.

1.2.2.4 η-*equality*

In terms of rewriting systems where '=' is used to represent the 'reduces to' relation, indexed by its kind, i.e. β-, η-, ξ-, ζ-, etc., conversion, the above *induction* rules become:

$\wedge$-η-*equality*

$$\langle FST(c), SND(c)\rangle =_\eta c$$

$\vee$-η-*equality*

$$CASE(c, \acute{x}inl(x), \acute{y}inr(y)) =_\eta c$$

$\rightarrow$-η-*equality*

$$\lambda x.APP(c, x) =_\eta c$$

provided x does not occur free in c.
$\forall$-η-*equality*

$$\Lambda t.EXTR(c, t) =_\eta c$$

provided c has no free occurrences of x.
$\exists$-η-*equality*

$$INST(c, \acute{g}\acute{t}\varepsilon y.(g(y), t)) =_\eta c.$$

The presentation taken by each of the rules above does indeed reveal an 'inductive' character: they all seem to be saying that if any arbitrary element 'c' is in the type then it must be reducible to itself via an *elimination* step with the DESTRUCTOR(s) followed by an *introduction* step with the constructor(s).

1.2.2.5 *ζ-type reductions: The permutative reductions turned unidirectional*

For the connectives that make use of 'Skolem'-type procedures of opening local branches with new assumptions, locally introducing new names and making them 'disappear' (or lose their identity via an abstraction) just before coming out of the local context or scope, there is another way of transforming proofs, which goes hand-in-hand with the properties of 'value-range' terms resulting from abstractions.

In the literature these proof transformations are called 'permutative' reductions because there is no *preferred* deduction: either one is considered to be as near to the normal form as the other. Nevertheless, if we impose an order between the two, and say that the more the 'extraneous' formulas of the rule of $\vee$-*elimination* are pushed upwards the better is the deduction with respect to approaching the normal form. Thus:

$\vee$-(*permutative*) *reduction*

$$\dfrac{\dfrac{p : A_1 \vee A_2 \quad \begin{matrix}[s_1 : A_1] \\ d(s_1) : C\end{matrix} \quad \begin{matrix}[s_2 : A_2] \\ e(s_2) : C\end{matrix}}{CASE(p, \acute{s}_1 d(s_1), \acute{s}_2 e(s_2)) : C}}{w(CASE(p, \acute{s}_1 d(s_1), \acute{s}_2 e(s_2))) : W} \;\rhd_\zeta$$

$$\dfrac{p : A_1 \vee A_2 \quad \dfrac{\begin{matrix}[s_1 : A_1] \\ d(s_1) : C\end{matrix}}{w(d(s_1)) : W} \quad \dfrac{\begin{matrix}[s_2 : A_2] \\ e(s_2) : C\end{matrix}}{w(e(s_2)) : W}}{CASE(p, \acute{s}_1 w(d(s_1)), \acute{s}_2 w(e(s_2))) : W}$$

$\exists$-(*permutative*) *reduction*

$$\dfrac{\dfrac{e : \exists x^D.P(x) \quad \begin{matrix}[t : D, g(t) : P(t)] \\ d(g,t) : C\end{matrix}}{INST(e, \acute{g}\acute{t}d(g,t)) : C}}{w(INST(e, \acute{g}\acute{t}d(g,t))) : W} \;\rhd_\zeta$$

$$\dfrac{e : \exists x^D.P(x) \quad \dfrac{\begin{matrix}[t : D, g(t) : P(t)] \\ d(g,t) : C\end{matrix}}{w(d(g,t)) : W}}{INST(e, \acute{g}\acute{t}w(d(g,t))) : W}$$

All this means that the deduction on the left is further away from its normal form than the one on the right of the '$\rhd$' sign.

Missing conversions. Of course this creates a problem for the uniqueness of normal forms, as A. de Oliveira has shown [de Oliveira (1995)].

As we have demonstrated [de Oliveira and de Queiroz (1995)], by proving the *termination* and *confluence* properties for the *term rewriting system* associated to the *LND* system (*TRS-LND*) [de Oliveira and de Queiroz (2005)], we have in fact proved the normalization and strong normalization theorems for the *LND* system, respectively. The *termination* property guarantees the existence of a normal form of the *LND*-terms, while the *confluence* property is its uniqueness. Thus, because of the Curry–Howard isomorphism, we have that every *LND* derivation converts to a normal form and it is unique.

The significance of applying this technique in the proof of the normalization theorems lies in the presentation of a simple and computational method, which allowed the discovery of a new basic set of transformations between proofs, which we baptized as "ι (iota)-reductions" [de Oliveira (1995)]. With this result, we obtained a confluent system which contains the η-reductions. Traditionally, the η-reductions have not been given an adequate status, as rightly pointed out by Girard in [Girard *et al.* (1989)] (p. 16), when he defines the *primary* equations, which correspond to the β-equations and the *secondary* equations, which are the η-equations. Girard says that the system given by these equations is consistent and decidable, however he notes the following:

> "Although this result holds for the whole set of equations, one only ever considers the first three. It is a consequence of the *Church–Rosser property* and the *normalization theorem* (...)" [Girard *et al.* (1989)]

The first three equations, referred to by Girard, are the *primary* ones, i.e. β-equations.

Applying the so-called *completion procedure*, proposed by Knuth and Bendix in [Knuth and Bendix (1970)], to *TRS-LND*, the following term, which causes a non-confluence in the system, is produced (i.e. a divergent critical pair is generated):

$$w(CASE(c, \acute{x}inl(x), \acute{y}inr(y))).$$

This term can be rewritten in two different ways[19]:

1. $\rhd_\eta \; w(c)$ 2. $\rhd_\zeta \; CASE(c, \acute{x}w(inl(x)), \acute{y}w(inr(y)))$

The method of Knuth and Bendix says that when a terminating system is not confluent it is possible to add rules in such a way that the resulting system becomes confluent. Thus applying this procedure to *TRS-LND*, a new rule is added to the system:

$$CASE(c, \acute{x}w(inl(x)), \acute{y}w(inr(y))) \rhd_\iota w(c).$$

Since terms represent proof-constructions in the *LND* system, this rule defines a new transformation between proofs:

ι-reduction-$\vee$

$$\dfrac{c : A_1 \vee A_2 \quad \dfrac{\dfrac{[x : A_1]}{inl(x) : A_1 \vee A_2}\vee\textit{-intr}}{w(inl(x)) : W}\mathsf{r} \quad \dfrac{\dfrac{[y : A_2]}{inr(y) : A_1 \vee A_2}\vee\textit{-intr}}{w(inr(y)) : W}\mathsf{r}}{CASE(c, \acute{x}w(inl(x)), \acute{y}w(inr(y))) : W}\vee\textit{-elim} \;\; \rhd_\iota \;\; \dfrac{c : A_1 \vee A_2}{w(c) : W}\mathsf{r}.$$

---

[19]The η-reduction for the $\vee$ connective is framed as follows:

$$\dfrac{c : A_1 \vee A_2 \quad \dfrac{[x : A_1]}{inl(x) : A_1 \vee A_2}\vee\textit{-intr} \quad \dfrac{[y : A_2]}{inr(y) : A_1 \vee A_2}\vee\textit{-intr}}{CASE(c, \acute{x}inl(x), \acute{y}inr(y)) : A_1 \vee A_2}\vee\textit{-elim} \;\; \rhd_\eta \;\; c : A_1 \vee A_2$$

('´' is an abstractor, similarly to 'λ')

and the ζ-reduction is defined as follows:

$$\dfrac{\dfrac{p : A_1 \vee A_2 \quad \begin{matrix}[s_1 : A_1]\\ d(s_1) : C\end{matrix} \quad \begin{matrix}[s_2 : A_2]\\ e(s_2) : C\end{matrix}}{CASE(p, \acute{s_1}d(s_1), \acute{s_2}e(s_2)) : C}}{w(CASE(p, \acute{s_1}d(s_1), \acute{s_2}e(s_2))) : W}\mathsf{r} \;\; \rhd_\zeta \;\; \dfrac{p : A_1 \vee A_2 \quad \dfrac{\begin{matrix}[s_1 : A_1]\\ d(s_1) : C\end{matrix}}{w(d(s_1)) : W}\mathsf{r} \quad \dfrac{\begin{matrix}[s_2 : A_2]\\ e(s_2) : C\end{matrix}}{w(e(s_2)) : W}\mathsf{r}}{CASE(p, \acute{s_1}w(d(s_1)), \acute{s_2}w(e(s_2))) : W}.$$

Whilst 'r' is usually restricted to an *elimination* rule, we have relaxed this condition: it is only required that 'r' does not discharge any assumptions from the other (independent) branch, i.e. that the auxiliary branches do not interfere with the main branch.

Similarly, the $iota$ reduction for the existential quantifier is defined [de Oliveira (1995)], since, similarly to $\vee$, the quantifier $\exists$ is a "Skolem type" connective (i.e. in the *elimination* inference for this type of connective is necessary to open local assumptions):

ι-reduction-$\exists$

$$\frac{c : \exists x^D.P(x) \qquad \dfrac{\dfrac{[t : D] \quad [g(t) : P(t)]}{\varepsilon y.(g(y), t) : \exists y^D P(y)}\exists\text{-}intr}{w(\varepsilon y.(g(y), t)) : W}\mathsf{r}}{INST(c, \acute{g}\acute{t}w(\varepsilon y.(g(y), t))) : W}\exists\text{-}elim \quad \rhd_\iota \quad \frac{c : \exists x^D P(x)}{w(c) : W}\mathsf{r}$$

(where 'ε' is an abstractor).

With this result, we believe that we have answered the question as to why the η-reductions are not considered in the proofs of the normalization theorems (*confluence* requires ι-reductions). However, by applying a computational and well-defined method, the completion procedure, it seems that this problem of the non-confluence caused by η-reductions are solved.

Theorem 1.1 ([de Oliveira (1995)]). *Every proof in the* LND *system has a unique normal form.*[20]

1.2.2.6 ζ-*type equality*

Now, if the functional calculus on the labels is to match the logical calculus on the formulas, we must have the following ζ-equality (read 'zeta'-equality) between terms:

$$w(CASE(p, \acute{s_1}d(s_1), \acute{s_2}e(s_2)), u) =_\zeta CASE(p, \acute{s_1}w(d(s_1), u), \acute{s_2}w(e(s_2), u))^{21}$$

[20] In fact, when we have disjunction and η-rules, uniqueness is not guaranteed. It is necessary to add new transformation rules baptised in [de Oliveira and de Queiroz (2005); de Oliveira (1995)] as 'ι' ('iota') rules.

[21] When defining 'Linearised sum', Girard *et al.* [Girard *et al.* (1989)] give the following equation as the term-equality counterpart to the permutative reduction:

"Finally, the commuting conversions are of the form

$$\mathsf{E}(\delta\ x.u\ y.v\ t) \rhd \delta\ x.(\mathsf{E}u)\ y.(\mathsf{E}v)\ t$$

where E is an elimination." [Girard *et al.* (1989)] (p. 103)

Note the restriction on the step corresponding to the operator 'E' (which corresponds to our 'w'): it has to be an elimination.

In our ζ-equality the operator 'w' does not have to be an eliminatory operator, but it only needs to be such that it preserves the dependencies of the term coming from the main branch, namely the step must preserve the free variables on which our 'p' depends. In other words, w cannot be an abstraction over free variables of p.

Our generalised ζ-equality also finds parallels in the literature on equational counterparts to commutative diagrams of category theory. For example, in the definition of *binary sums* given by A. Poigne for disjunction, and

$$w(INST(e, \acute{g}\acute{t}d(g,t)), u) =_{\zeta} INST(e, \acute{g}\acute{t}w(d(g,t), u))$$

for the existential quantifier.

Note that both in the case of ‘$\vee$’ and ‘$\exists$’ the operator ‘w’ could be ‘pushed inside’ the value-range abstraction terms. In the case of disjunction, the operator could be pushed inside the ‘´’-abstraction terms, and in the $\exists$-case, the ‘w’ could be pushed inside the ‘´’-abstraction term.

In terms of the proof theory, these reductions imply that the newly opened branches must be independent from the main branch. And, indeed, notice that in the proof-trees above, the step coming after the *elimination* of the connective concerned ($\vee$-, $\exists$-*elimination*) is taken to be as general as possible, provided that it does not affect the dependencies on the main branch (i.e. ‘$p : A_1 \vee A_2$’, ‘$e : \exists x^D P(x)$’, respectively). (E.g. any deduction step involving discharge of assumptions may disturb the dependencies.) Those reductions will then uncover β-type redundancies which may be hidden by an $\vee$-, $\exists$- *elimination* rule. Perhaps for this reason, in the literature it is common to restrict that particular step to a deduction to an *elimination* rule where the formula ‘C’ is to be its major premise.[22]

The restriction to the case when the step is an *elimination* rule seems to be connected with the idea that the *permutative* conversions are brought in to help recover the so-called *subformula property*.[23] We would prefer to see the rôle of those

[Poigne (1992)], the counterpart of our ζ-equality for disjunction appears as:

$$h \circ case(f, g) = case(h \circ f, h \circ g)$$

where ‘$\circ$’ is the basic operation of composition. Note that the function ‘h’ can be pushed inside the ‘$case$’-term, similarly to our ζ-equality where the ‘w’ can be pushed inside the ´-abstraction terms of our $CASE$-expression.

[22]When commenting on the requirements of permutative reductions, Prawitz remarks:

“It has been remarked by Martin-Lof that it is only necessary to require in the $\vee$E- and $\exists$E-reductions that the lowest occurrence of C is the major premiss of an elimination. A reduction of this kind can then always be carried out and we can sharpen the requirements as to the normal form accordingly.” [Prawitz (1971)] (p. 253ff)

And, indeed, for his proof of the strong validity lemma (p. 295) Prawitz needs the condition on the permutative reductions that the step after the $\vee$-*elimination* (resp. $\exists$-*elimination*) be also an *elimination* inference.

No restriction to an *elimination* step is mentioned in [Martin-Lof (1975a)]. Rather, it is required that the dependencies be preserved:

“(...) the *permutative* rules for $\vee$ and $\exists$, (...) provided the inference from C to D neither binds any free variable nor discharges any assumption in the derivation of $A \vee B$ and $(\exists x)B[x]$, respectively.” [Martin-Lof (1975a)] (p. 100f)

Cf. also other standard texts in the literature where the restriction is unnecessarily imposed: Troelstra and van Dalen’s [Troelstra and van Dalen (1988)] (p. 534ff) and Girard *et al.*’s [Girard *et al.* (1989)] definitions of *permutative conversions* have the requirement that the step following the $\vee$- ($\exists$-)*elimination* be an ‘E-rule’ (*Elimination* rule).

[23]Cf. [Girard *et al.* (1989); Troelstra and van Dalen (1988)].

rules of proof transformation as that of guaranteeing a 'pact of non-interference' between the main branch and those new branches created by the elimination rules of 'Skolem-type' connectives ($\vee$, $\exists$, $\doteq$).

Thus, in the more general case, it seems as though the restriction (to the case where the formula 'C' is a major premise of an *elimination* inference) is unnecessary. And this is because we can have the following conversion using an *introduction* inference instead:

$$\dfrac{\dfrac{p : A_1 \vee A_2 \qquad \begin{matrix}[x : A_1]\\ d(x) : C\end{matrix} \qquad \begin{matrix}[y : A_2]\\ e(y) : C\end{matrix}}{CASE(p, \acute{x}d(x), \acute{y}e(y)) : C}}{inl(CASE(p, \acute{x}d(x), \acute{y}e(y))) : C \vee U}(*) \qquad \rhd_\zeta$$

$$\dfrac{p : A_1 \vee A_2 \qquad \dfrac{\begin{matrix}[x : A_1]\\ d(x) : C\end{matrix}}{inl(d(x)) : C \vee U} \qquad \dfrac{\begin{matrix}[y : A_2]\\ e(y) : C\end{matrix}}{inl(e(y)) : C \vee U}}{CASE(p, \acute{x}inl(d(x)), \acute{y}inl(e(y))) : C \vee U}.$$

One can readily notice that the $\vee$-*introduction* step marked '$(*)$' does not affect the dependencies (i.e. does not involve any assumption discharge), so the constructor 'inl' can be pushed inside the ´-abstraction terms. The same holds if, instead of $\vee$-*introduction*, one performs an $\wedge$-*introduction* as in:

$$\dfrac{\dfrac{p : A_1 \vee A_2 \qquad \begin{matrix}[x : A_1]\\ d(x) : C\end{matrix} \qquad \begin{matrix}[y : A_2]\\ e(y) : C\end{matrix}}{CASE(p, \acute{x}d(x), \acute{y}e(y)) : C} \qquad u : U}{\langle CASE(p, \acute{x}d(x), \acute{y}e(y)), u\rangle : C \wedge U} \qquad \rhd_\zeta$$

$$\dfrac{p : A_1 \vee A_2 \qquad \dfrac{\begin{matrix}[x : A_1]\\ d(x) : C\end{matrix} \quad u : U}{\langle d(x), u\rangle : C \wedge U} \qquad \dfrac{\begin{matrix}[y : A_2]\\ e(y) : C\end{matrix} \quad u : U}{\langle e(y), u\rangle : C \wedge U}}{CASE(p, \acute{x}\langle d(x), u\rangle, \acute{y}\langle e(y), u\rangle) : C \wedge U}$$

and, clearly:

$$\langle CASE(p, \acute{x}d(x), \acute{y}e(y)), u\rangle =_\zeta CASE(p, \acute{x}\langle d(x), u\rangle, \acute{y}\langle e(y), u\rangle),$$

given that the pairing operation can be pushed inside the ´-abstraction terms without disturbing the dependencies. One can readily see that the $\wedge$-*introduction* is harmless with respect to the dependencies. (Note that the same observation applies to ζ-reduction of $\exists$, and, as we shall see later on, to the permutative reduction of $\doteq$.)