VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematical and Computing Sciences
Computer Science

Meaning as consequence and the justification of inference rules

Neil Leslie
School of Mathematics, Statistics and Computer Science &
Centre for Logic, Language and Computation
Victoria University
P. O. Box 600
Wellington
New Zealand

Technical Report CS-TR-05-6
December 14, 2005

School of Mathematical and Computing Sciences
Victoria University
PO Box 600, Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematical and Computing Sciences
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341, Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

Meaning as consequence and the justification of inference rules

Neil Leslie
School of Mathematics, Statistics and Computer Science &
Centre for Logic, Language and Computation
Victoria University
P. O. Box 600
Wellington
New Zealand

Technical Report CS-TR-05-6
December 14, 2005

Abstract

We often think of a proposition as being defined in terms of what (immediate) *evidence* for it is, and that the meanings of the logical constants are determined by their introduction rules. Instead, let's think of a proposition as being defined in terms of what an immediate *consequence* of it is, and take the elimination rules as determining the meanings of the constants.

Publishing Information

This is a preliminary version of a paper submitted for publication elsewhere.

Author Information

Neil's main interest is in exploiting the observation that propositions and types are the same thing. He also often works on his favourite hobby, the rejection of the law of the excluded middle.

1 Introduction

Much work has been done by, amongst others, Michael Dummett (e.g. [3, 4, 6]), Per Martin-Löf (e.g. [19]) and Dag Prawitz (e.g. [23]) to provide proof-theoretical explanations of the meanings of the logical constants, essentially following from Gentzen's assertion [10]:

> 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.

The aim of this paper is to show that we can just as easily take the *eliminations* to represent the definitions of the symbols, and treat the introductions as no more than consequences of these. Informally, we are identifying the meaning of a proposition with its consequences, rather than with the evidence for it. If we take this approach then [7] Dummett points out that 'all we need to do is explain the *immediate* consequences' of a sentence. We do this using a revised version of the Brouwer-Heyting-Kolmogorov (BHK) interpretation [13, 15].

Throughout we will work implicitly with the Curry-Howard 'propositions as types' analogy [14, 2]. We can make this explicit to see some connections with computer programming. The approach of defining the logical constants in terms of the introduction rules translates to the setting of programming as defining algebraic data types. The approach of defining the logical constants in terms of their elimination rules corresponds to defining co-data types. More work remains to be done to formalise this fully, and we will not explore it here.

The structure of this paper is as follows. In Section 2 we present some background on judgements and rules, and in Section 3 we briefly discuss the Dummett-Martin-Löf-Prawitz explanation of the meanings of the logical constants, concentrating on the propositional fragment. We take some care over the elimination rules. In Section 4 we identify the novelty of our explanation. In Section 5 we develop an alternative explanation of the meanings of the logical constants, taking the elimination rules as primitive. In Section 7 we discuss the quantifiers. We present our conclusions in Section 8.

2 Judgements, rules, derivations

It is well-known that we must take some care when asserting that the rules for the use of a connective define its meaning. Prior's `tonk` [24] has what appear to be perfectly good rules, but a perfectly bad meaning. One solution to the challenge of `tonk` is given by Belnap [1]:

> we are not defining our connectives *ab initio*, but rather in terms of an *antecedently given context of deducibility*, concerning which we have some definite notions.

The emphasis is in the original. For us the *context of deducibility* is given by a theory of judgements along the lines of that given by Martin-Löf [17, 18, 19] and also presented in [22]. We will now outline this theory. We draw particular attention to the use of hypothetical judgements as assumptions.

We make a careful distinction between *proposition* and *judgement*. A proposition is something like: 'I live on Mars'; 'Paris is sunny'. A judgement is something like 'There is evidence that I live on Mars'; 'It is true that Paris is sunny'. Although this distinction has been recognised by logicians since at least Frege [9], often, when writing logical rules down we ignore the difference between propositions and judgements. To redress this we might prefer to write $\vdash A$ when we give rules but this notation can also be rather cluttered, and is easily confused with the sequent calculus, so we will not use it.

Next we explain what categorical and hypothetical judgements are.

DEFINITION 1 A categorical judgement is one which can be made on its own, without dependence on any other judgement.

DEFINITION 2 A hypothetical judgement is one which which relies on one or more other judgements, called assumptions.

An assumption is itself a *judgement*, and hence may be a hypothetical judgement. We are not used to seeing hypothetical judgements used as assumptions in a logical setting, but they are used explicitly in type theory (see, for example, [22]) in the Π, Σ, and W types. We will exploit them later.

2.1 Rules not involving logical constants

Rules not concerning any logical constant can be called 'structural rules', by analogy with the sequent calculus, and are often left implicit in presentations of natural deduction. Although have not yet said anything about the logical constants we already have a theory judgements. In particular, we can introduce the notion of a valid rule. For example, if we are capable of judging A, and we are capable of judging B then we are capable of judging A. Hence, we consider that:

$$\frac{\begin{matrix}\vdots\\ A\end{matrix} \qquad \begin{matrix}\vdots\\ B\end{matrix}}{A}$$

is a valid rule. Of course, there are logics where this is rule is not valid, and for those logics we can think of ourselves as having a different notion of judgement.

Furthermore, based only on our theory of judgements, this rule:

$$\frac{\begin{matrix}\vdots\\ A\end{matrix} \qquad \begin{matrix}[A]\\ \vdots\\ B\end{matrix}}{B}$$

is valid (as usual, the square brackets indicate the discharge of an assumption). This rule is justified by explaining that if we can judge A, and if we could judge B on the assumption that we could judge A, then we can judge B, and hence is equivalent to:

$$\begin{matrix}\vdots\\ B\end{matrix}$$

B will, of course, rely on whatever assumptions A and B relied on before.

Not every structural figure is a valid rule. For example,

$$\frac{\begin{matrix}\vdots\\ A\end{matrix} \qquad \begin{matrix}\vdots\\ B\end{matrix}}{C}$$

is not. [16] shows that structural rules can be used in non-trivial ways partly to define the meanings of the connectives.

3 Dummett-Martin-Löf-Prawitz explanation

We briefly give some definitions concerning rules dealing with logical constants, taken from [6].

DEFINITION 3 A rule is an introduction rule for a logical constant # if # is the principal connective of the conclusion of the rule.

DEFINITION 4 A rule is an elimination rule for a logical constant # if # is the principal connective of one of the premisses of the rule.

DEFINITION 5 A rule is single-ended if it is either an introduction rule or an elimination rule, but not both.

DEFINITION 6 A rule is pure if only one logical constant figures in it.

DEFINITION 7 An introduction rule satisfies the complexity condition if its conclusion is of higher logical complexity than any premiss or discharged assumption.

The claim (of Dummett explicitly in [6]) is that single-ended introduction rules satisfying the complexity condition are enough to define the meanings of the logical constants. In Martin-Löf's language we would say that the meaning of a sentence is determined by what counts as evidence for it. Introduction rules of the form Dummett describes explain what counts as immediate evidence for compound sentences.

3.1 The Brouwer-Heyting-Kolmogorov interpretation and the introduction rules

The BHK interpretation [13, 15] explains the meaning of a compound sentence in terms of what counts as *immediate evidence* for it, given evidence for its sub-sentences. The table is:

Proposition	Immediate evidence
$A \vee B$	Either evidence for A, or evidence for B
$A \mathbin{\&} B$	Evidence for A together with evidence for B
$A \supset B$	A method to turn evidence for A into evidence for B
$\perp$	—

Recall that $\neg A =_{\text{def}} A \supset \perp$.

The introduction rules are then constructed by inspection of this table:

$$\dfrac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{A \supset B}\supset \mathrm{I} \qquad \dfrac{\begin{matrix}\vdots \\ A\end{matrix} \qquad \begin{matrix}\vdots \\ B\end{matrix}}{A \mathbin{\&} B}\ \& \mathrm{I}$$

$$\dfrac{\begin{matrix}\vdots \\ A\end{matrix}}{A \vee B}\vee\mathrm{I} \qquad \dfrac{\begin{matrix}\vdots \\ B\end{matrix}}{A \vee B}\vee\mathrm{I}$$

There is no immediate evidence for $\perp$, and hence no introduction rule of the appropriate form.

The $\neg$ introduction rule is produced from the $\supset$ introduction rule by identifying B with $\perp$:

$$\frac{\begin{array}{c}[A]\\ \vdots \\ \bot\end{array}}{A \supset \bot}\neg\mathrm{I}$$

3.2 Harmony, and the elimination rules

If we believe that these are the correct introduction rules, and we believe that the introduction rules define the meanings of the logical constants then we should be able to produce the elimination rules mechanically, by appeal to our general notions of what proofs and derivations are.

Let # be an n-ary constant. The elimination rule for # will have $\#A_1 \ldots A_n$ as its major premiss, and some arbitrary[1] conclusion C. The minor premisses are generated by considering each of the introduction rules for # in turn. In each case C must be derivable from the premisses to the introduction rule.

This process ensures that using # elimination immediately after # introduction in a proof is redundant. We have a single step of reduction:

$$\frac{\dfrac{\vdots}{\#A_1 \ldots A_n}\#\mathrm{I} \qquad \vdots}{C}\#\mathrm{E} \quad \rhd_1 \quad \begin{array}{c}\vdots\\ C\end{array}$$

Showing that $\rhd$, the reflexive, transitive closure of $\rhd_1$ is confluent shows that proofs normalise. In λ-calculus terms $\rhd$ is β-reduction.

Because we use a general method to produce elimination rules all the elimination rules are uniform. Each connective has only one elimination rule, and they all follow the pattern familiar from $\vee$ elimination, $\exists$ elimination and the elimination rules used in type theory [22]. We might argue that C is symbiotic, rather than parasitic.

3.3 Disjunction

There are two $\vee$ introduction rules, so there are two minor premisses in the $\vee$ elimination rule: one is a derivation of C from A, the other a derivation of C from B.

$$\frac{\begin{array}{c}\vdots\\ A \vee B\end{array} \qquad \begin{array}{c}[A]\\ \vdots\\ C\end{array} \qquad \begin{array}{c}[B]\\ \vdots\\ C\end{array}}{C}\vee\mathrm{E}$$

3.4 The absurd

The is no introduction rule for $\bot$, so there is no minor premiss in the $\bot$ elimination rule.

$$\frac{\begin{array}{c}\vdots\\ \bot\end{array}}{C}\bot\mathrm{E}$$

[1] Or even 'parasitic', see p. 74 of [11].

3.5 Conjunction

There is one & introduction rule, so there is one minor premiss in the & elimination rule: a derivation of C which may use A and which may use B

$$\dfrac{\begin{array}{c}\vdots\\ A \mathbin{\&} B\end{array} \qquad \begin{array}{c}[A \qquad B]\\ \vdots\\ C\end{array}}{C}\ \&\,\mathrm{E}$$

3.5.1 Other elimination rules for &

There are other pure elimination rules for & . We are used to seeing these two:

$$\dfrac{\begin{array}{c}\vdots\\ A \mathbin{\&} B\end{array}}{A}\ \&\,\mathrm{E}_l \qquad \dfrac{\begin{array}{c}\vdots\\ A \mathbin{\&} B\end{array}}{B}\ \&\,\mathrm{E}_r$$

We can recover $\&\,\mathrm{E}_l$ (resp. $\&\,\mathrm{E}_r$) from $\&\,\mathrm{E}$ by identifying C with A (resp. B):

$$\dfrac{\begin{array}{c}\vdots\\ A \mathbin{\&} B\end{array} \qquad [A]}{A}\ \&\,\mathrm{E}$$

$$\dfrac{\begin{array}{c}\vdots\\ A \mathbin{\&} B\end{array} \qquad [B]}{B}\ \&\,\mathrm{E}$$

We can recover $\&\,\mathrm{E}$ from $\&\,\mathrm{E}_l$ and $\&\,\mathrm{E}_r$ by making a detour (we choose to use one of the structural rules presented earlier):

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ A \mathbin{\&} B\end{array}}{A}\ \&\,\mathrm{E}_l \qquad \dfrac{\dfrac{\begin{array}{c}\vdots\\ A \mathbin{\&} B\end{array}}{B}\ \&\,\mathrm{E}_r \qquad \begin{array}{c}[A \qquad B]\\ \vdots\\ C\end{array}}{C}}{C}$$

So, $\&\,\mathrm{E}_l$ and $\&\,\mathrm{E}_r$ can be removed from proofs in favour of $\&\,\mathrm{E}$, and *vice versa*, and hence these rules are equivalent.

3.6 Implication

There is one $\supset$ introduction rule, so there is one minor premisses in the $\supset$ elimination rule: a derivation of C which may use a derivation of B, which may itself use a derivation of A:

$$\dfrac{\begin{array}{c}\vdots\\ A \supset B\end{array} \qquad \begin{array}{c}[[A]\cdots B]\\ \vdots\\ C\end{array}}{C}\ \supset \mathrm{E}$$

This rule makes essential use of a hypothetical hypothesis, but is, as we have already emphasised, just the rule we get in type theory.

3.6.1 Other elimination rules for $\supset$

There are other pure elimination rules for $\supset$. We are more used to seeing *modus ponens*[2]:

$$\frac{\begin{matrix}\vdots\\ A \supset B\end{matrix} \qquad \begin{matrix}\vdots\\ A\end{matrix}}{B}\,MP$$

Another pure, and more generalised, $\supset$ elimination rule is given in [8, 20, 21]:

$$\frac{\begin{matrix}\vdots\\ A \supset B\end{matrix} \qquad \begin{matrix}\vdots\\ A\end{matrix} \qquad \begin{matrix}[B]\\ \vdots\\ C\end{matrix}}{C}\supset \mathrm{E}'$$

We can recover *modus ponens* from $\supset \mathrm{E}'$ by identifying C with B, which gives:

$$\frac{\begin{matrix}\vdots\\ A \supset B\end{matrix} \qquad \begin{matrix}\vdots\\ A\end{matrix} \qquad [B]}{B}\supset \mathrm{E}'$$

We can recover $\supset \mathrm{E}'$ from *modus ponens* by making a detour (we choose to use $\supset$ I and *modus ponens* itself) which gives:

$$\frac{\dfrac{\begin{matrix}\vdots\\ A \supset B\end{matrix} \qquad \begin{matrix}\vdots\\ A\end{matrix}}{B}\,MP \qquad \dfrac{\begin{matrix}[B]\\ \vdots\\ C\end{matrix}}{B \supset C}\supset \mathrm{I}}{C}\,MP$$

So, *modus ponens* can be removed from proofs in favour of $\supset \mathrm{E}'$, and *vice versa*, and hence these two rules are equivalent.

Now we show that $\supset$ E subsumes $\supset \mathrm{E}'$. Suppose we have a derivation of A. We can use structural rules from before and $\supset$ E to obtain:

$$\frac{\dfrac{\begin{matrix}\vdots\\ A \supset B\end{matrix} \qquad \begin{matrix}\vdots\\ A\end{matrix}}{A \supset B} \qquad \begin{matrix}\left[\dfrac{\begin{matrix}\vdots\\ A\end{matrix} \qquad \begin{matrix}[A]\\ \vdots\\ B\end{matrix}}{B}\right]\\ \vdots\\ C\end{matrix}}{C}\supset \mathrm{E}$$

This figure simplifies to:

$$\frac{\begin{matrix}\vdots\\ A \supset B\end{matrix} \qquad \begin{matrix}\vdots\\ A\end{matrix} \qquad \begin{matrix}[B]\\ \vdots\\ C\end{matrix}}{C}\supset \mathrm{E}$$

We can also recover *modus ponens* from $\supset$ E by using the definition of apply in terms of $\mathsf{funsplit}$ familiar from type theory [22]:

$$\mathsf{apply}(f, x) =_{\mathrm{def}} \mathsf{funsplit}(f, (y)(y(x)))$$

[2] More accurately: *modus ponendo ponens*.

3.7 Dummett-Martin-Löf-Prawitz summary

What we have presented so far is the outline of the Dummett-Martin-Löf-Prawitz explanation of the meanings of the logical constants. There are several key points:

- meaning is determined by use;
- there is a theory of rules and judgements;
- introduction rules of a special form are taken as definitional;
- the introduction rules are justified with reference to the BHK explanation of what counts as immediate evidence for a compound sentence;
- the elimination rules are constructed in a uniform way;
- each constant has only one elimination rule;
- the use of an introduction rule followed by an elimination rule is redundant.

We have, of course, justified the rules of constructive logic. Had we a different theory of judgements, or accepted different forms of primitive rule we could have developed other logics.

4 Alterations to take the elimination rules as definitional

Now we will alter the approach and take the elimination rules as definitional. We follow the pattern used before, but:

- elimination rules of a special form are taken as definitional;
- the elimination rules are justified with reference to an explanation of what counts as an immediate consequence for a compound sentence.

The introduction rules must now be justified in a uniform way, and the key is to demand that

- the use of an elimination rule followed by some introduction rules is redundant.

5 BHK for consequences

Now we must give an explanation of the meanings of the logical constants in terms of what counts as an *immediate consequence* of a compound sentence, given that we know what the consequences of its sub-sentences are. We follow the pattern of the BHK explanation and produce this table:

Proposition	Immediate consequence
$A \vee B$	Anything which is a consequence of A, and is a consequence of B
$A \,\&\, B$	Anything which is a consequence of A or of B (or possibly both)
$A \supset B$	Anything which is a consequence of the fact that B is a consequence of A
$\perp$	Anything at all

From this table we generate an elimination rule for each constant. The elimination rules will all be single-ended, pure elimination rules and will meet a complexity condition. The rules will turn out to be exactly the uniform elimination rules that we presented before, although the method of justification is different.

The complexity condition is that the discharged assumptions in the minor premisses are less complex than the major premiss, and the conclusions of the minor premisses are no more complex than the conclusion of the rule.

5.1 The absurd

The easiest constant to begin with is $\perp$. Anything at all is an immediate consequence of $\perp$, so the elimination rule is simply:

$$\dfrac{\begin{matrix}\vdots\\ \perp\end{matrix}}{C}\ \perp\mathrm{E}$$

This certainly meets the complexity condition, and seems at least as good a candidate for stipulating the meaning of $\perp$ as the non-existence of an introduction rule.

5.2 Disjunction

Anything which is a consequence of A and which is a consequence of B is an immediate consequence of $A \vee B$. Notice here that we have an definition of $\vee$ which involves our informal notion of 'and', thus giving a duality between 'or' and 'and'. The rule is just as before:

$$\dfrac{\begin{matrix}\\ \vdots\\ A \vee B\end{matrix}\qquad\begin{matrix}[A]\\ \vdots\\ C\end{matrix}\qquad\begin{matrix}[B]\\ \vdots\\ C\end{matrix}}{C}\ \vee\mathrm{E}$$

5.3 Conjunction

Anything which is a consequence of A or which is a consequence of B is an immediate consequence of $A \,\&\, B$. Again we have a duality between 'and' and 'or', and once again, the rule we get is just the uniform rule that we had before:

$$\dfrac{\begin{matrix}\\ \vdots\\ A \,\&\, B\end{matrix}\qquad\begin{matrix}[A \quad B]\\ \vdots\\ C\end{matrix}}{C}\ \&\,\mathrm{E}$$

5.4 Implication

Implication is the most interesting of the propositional connectives in this setting. The rule that we get by reading the revised BHK table, and which is uniform with the other elimination rules is:

$$\dfrac{\begin{matrix}\\ \vdots\\ A \supset B\end{matrix}\qquad\begin{matrix}[[A] \cdots B]\\ \vdots\\ C\end{matrix}}{C}\ \supset\mathrm{E}$$

5.5 Comments on these rules

By thinking of defining the logical constants in terms of their consequences we have arrived at just the same set of uniform elimination rules that we had before. *Modus ponens*, the rule $\supset \mathrm{E}'$, $\& \mathrm{E}_l$ and $\& \mathrm{E}_r$ are still admissible rules. We still have the task of explaining how to justify the introduction rules, and then of generating them.

6 Establishing harmonious introduction rules

We observe that every proposition is a consequence of itself, and that harmony between elimination and introduction rules must respect this.

In terms of inference figures what we are now demanding is:

$$\dfrac{\begin{array}{c}\vdots\\ \#A_1 \ldots A_n\end{array} \quad \dfrac{\vdots}{\#A_1 \ldots A_n}\#\mathrm{I} \quad \cdots \quad \dfrac{\vdots}{\#A_1 \ldots A_n}\#\mathrm{I}}{\#A_1 \ldots A_n}\#\mathrm{E} \quad \rhd_1 \quad \begin{array}{c}\vdots\\ \#A_1 \ldots A_n\end{array}$$

In other words an elimination followed by an introduction is redundant.

6.1 Disjunction

In the case of $\vee$, we demand that $A \vee B \vdash A \vee B$. We simply try to construct a proof of this, starting with a use of $\vee$ elimination. We get this figure:

$$\dfrac{\begin{array}{c}\vdots\\ A \vee B\end{array} \quad \begin{array}{c}[A]\\ \vdots\\ A \vee B\end{array} \quad \begin{array}{c}[B]\\ \vdots\\ A \vee B\end{array}}{A \vee B}\vee\mathrm{E}$$

The two minor premisses must be immediate judgements, that is $A \vee B$ must be a consequence of A, and $A \vee B$ must be a consequence of B. Hence, we obtain two new rules:

$$\dfrac{\begin{array}{c}\vdots\\ A\end{array}}{A \vee B}\vee\mathrm{I}_1$$

and

$$\dfrac{\begin{array}{c}\vdots\\ B\end{array}}{A \vee B}\vee\mathrm{I}_2$$

These are just the introduction rules for $\vee$ that we are familiar with, but we have justified them by taking meaning to be determined by consequence.

6.2 Conjunction

For $\&$ we demand that $A \,\&\, B \vdash A \,\&\, B$, and try to prove this using $\&\,\mathrm{E}$:

$$\dfrac{A \,\&\, B \qquad \begin{matrix}[A \quad B] \\ \vdots \\ A \,\&\, B\end{matrix}}{A \,\&\, B}\ \&\,\mathrm{E}$$

Again the minor premiss must be immediate, i.e. $A \,\&\, B$ is a consequence of A and B so we get:

$$\dfrac{\begin{matrix}\vdots \\ A\end{matrix} \qquad \begin{matrix}\vdots \\ B\end{matrix}}{A \,\&\, B}\ \&\,\mathrm{I}$$

This is just the introduction rule for $\&$ that we had before.

6.3 The absurd

In the case of $\bot$ we demand that $\bot \vdash \bot$:

$$\dfrac{\begin{matrix}\vdots \\ \bot\end{matrix}}{\bot}\ \bot\mathrm{E}$$

We see that there is no $\bot$ introduction rule.

6.4 Implication

Implication is a bit trickier, and it is here that hypothetical hypotheses pay dividends. Again we just demand that $A \supset B \vdash A \supset B$:

$$\dfrac{\begin{matrix}\vdots \\ A \supset B\end{matrix} \qquad \begin{matrix}[[A] \cdots B] \\ \vdots \\ A \supset B\end{matrix}}{A \supset B}\ \supset \mathrm{E}$$

We get this as the $\supset$ introduction rule:

$$\dfrac{\begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{A \supset B}\ \supset \mathrm{I}$$

6.5 Comments

By taking meaning to be determined by consequence, and hence the elimination rules to be primitive we have produced just the introduction rules that we are familiar with. The explanation of why these are the correct rules is, of course different. Whether one prefers to define, say, $\vee$, in terms of evidence or consequence seems to be just a matter of taste. The explanation of $\perp$ in terms of its consequences seems neater. On the other hand, if we have a trivial truth $\top$ then the explanation in terms of immediate consequence is not so clear. We have this table:

Proposition	Immediate consequence
$\top$	-

Just as there is no introduction rule of a suitable form for $\perp$ there is no elimination rule of suitable form for $\top$. In the cases of $\perp$ and $\top$ there does seems to be a good reason to prefer one explanation over the other.

If we are thinking in terms of computer programming, then there are some data types for which the algebraic approach is more natural, and some for which the coalgebraic approach is more natural.

7 First-order logic

We have focussed exclusively on propositional logic so far. Now we tackle first-order logic. The table that we get for the quantifiers is:

Proposition	Immediate consequence
$(\forall x)F(x)$	Anything which is a consequence of $F(t)$, for some t
$(\exists x)F(x)$	Anything which is a consequence of $F(t)$, for every t

If we let Q be either $\forall$ or $\exists$ then the Q elimination rule looks like:

$$\dfrac{\begin{array}{c}\vdots\\ \mathrm{(Qx)F(x)}\end{array}\qquad\qquad \begin{array}{c}[F(t)]\\ \vdots\\ C\end{array}}{C}\,Q\mathrm{E}$$

Identifying C with $(Qx)F(x)$ gives:

$$\dfrac{\begin{array}{c}\vdots\\ \mathrm{(Qx)F(x)}\end{array}\qquad\qquad \begin{array}{c}[F(t)]\\ \vdots\\ (Qx)F(x)\end{array}}{(Qx)F(x)}$$

Hence the Q introduction rule looks like:

$$\dfrac{\begin{array}{c}\vdots\\ F(t)\end{array}}{(Qx)F(x)}\,Q\mathrm{I}$$

As we might expect these rules look just like the Σ and Π rules of type theory.

The difference between $\forall$ and $\exists$ lies in the side-conditions. In $\forall$ elimination there are no conditions on t, in $\exists$ elimination there is a condition that there is nothing special about t, that is that t not occur free in C or any of the undischarged assumptions in the minor premiss. The side-conditions on the introduction rules are a consequence of the requirement that elimination-introduction pairs be redundant.

8 Summary

We have presented an outline of the Dummett-Martin-Löf-Prawitz proof-theoretic explanation of the meanings of the logical constants, based on a theory of judgements and an informal description of what counts as immediate evidence for a compound sentence.

We then presented an informal description of what counts as an immediate consequence of a sentence, and showed that this provides just as good a basis for the explanation of the meanings of the constants.

References

[1] Nuel D Belnap. Tonk, plonk and plink. *Analysis*, 22:130–134, 1961-1962.

[2] Phillipe de Groote. *The Curry-Howard Isomorphism*, volume 8 of *Cahiers du Centre de Logique*. Academia, Louvain-la-Neuve, Belgium, 1995.

[3] Michael Dummett. The justification of deduction. In *Truth and Other Enigmas* [5], pages 290–318. First published in 1973.

[4] Michael Dummett. The philosophical basis of intuitionistic logic. In *Truth and Other Enigmas* [5], pages 215–247. First published in [25].

[5] Michael Dummett. *Truth and Other Enigmas*. Duckworth, London, England, 1978.

[6] Michael Dummett. *The Logical Basis of Metaphysics*. Duckworth, London, England, 1991.

[7] Michael Dummett. Language and truth. In *The Seas of Language*, pages 117–146. Oxford University Press, Oxford, England, 1993. First published in [12].

[8] Roy Dyckhoff. Implementing a simple proof assistant. In J. Derrick and H. Lewis, editors, *Workshop on Programming for Logic Teaching*. Centre for Theoretical Computer Science, University of Leeds, 1987.

[9] Peter Geach and Max Black, editors. *Translations from the Philosophical Writings of Gottlob Frege*. Barnes and Noble, Totawa, New Jersey, USA, third edition, 1952.

[10] Gerhard Gentzen. Investigations into logical deduction. In M E Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, The Netherlands, 1969.

[11] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, England, 1989.

[12] Roy Harris, editor. *Approaches to Language*, volume iv of *Language and Communication Library*. Pergamon, Oxford, 1983.

[13] A Heyting. *Intuitionism An Introduction*. Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam, The Netherlands, 3rd edition, 1971.

[14] William Howard. The formulae-as-types notion of construction. In John Roger Hindley and Jonathan Seldin, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*, pages 479–490. Academic Press, 1969.

[15] Andrei Kolmogorov. On the interpretation of intuitionistic logic. In Paulo Mancosu, editor, *From Brouwer to Hilbert The Debate on the Foundations of Mathematics in the 1920s*, pages 324–334. Oxford University Press, Oxford, England, 1998. Originally published in German as Zur Deutung der intuitionistichen Logik, *Mathematische Zeitschrift* 35:58–65, 1932.

[16] Neil Leslie. Characterising the difference between **NJ** and **NK** in a connective-free manner. In *Natural Deduction Rio*. `http://www.inf.puc-rio.br/nd/program.html`, 2001. A version of this paper was also presented at the Australasian Association for Logic, Wellington, 2002.

[17] Per Martin-Löf. *Intuitionistic Type Theory*, volume 1 of *Studies in Proof Theory Lecture Notes*. Bibliopolis, Napoli, Italy, 1984. Notes taken by Giovanni Sambin from a series of lectures given in Padua, June 1980.

[18] Per Martin-Löf. Truth of a proposition, evidence of a judgement, validity of a proof. *Synthese*, 73:407–420, 1987.

[19] Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. *Nordic Journal of Philosophical Logic*, 1(1):11–60, 1996. Originally presented at the meeting *Teoria della Dimostrazione e Filosofia della Logica*, Siena 6–9 April, 1983.

[20] Sara Negri and Jan von Plato. Sequent calculus in natural deduction style. *The Journal of Symbolic Logic*, 66(4):1803–1816, 2001.

[21] Sara Negri and Jan von Plato. *Structural Proof Theory*. Cambridge University Press, 2001.

[22] Bengt Nordström, Kent Petersson, and Jan M Smith. *Programming in Martin-Löf's Type Theory An Introduction*. Clarendon Press, Oxford, England, 1990.

[23] Dag Prawitz. Meaning and proofs: on the conflict between classical and intuitionistic logic. *Theoria*, 77(1):1–40, 1977.

[24] Arthur N Prior. The runabout inference-ticket. *Analysis*, 21:38–39, 1961.

[25] H. E. Rose and J. C. Shepherdson, editors. *Logic Colloquium '73*, Amsterdam, The Netherlands, 1975. North-Holland.