PETER SCHROEDER-HEISTER

VALIDITY CONCEPTS IN PROOF-THEORETIC SEMANTICS

ABSTRACT. The standard approach to what I call "proof-theoretic semantics", which is mainly due to Dummett and Prawitz, attempts to give a semantics of proofs by defining what counts as a valid proof. After a discussion of the general aims of proof-theoretic semantics, this paper investigates in detail various notions of proof-theoretic validity and offers certain improvements of the definitions given by Prawitz. Particular emphasis is placed on the relationship between semantic validity concepts and validity concepts used in normalization theory. It is argued that these two sorts of concepts must be kept strictly apart.

1. INTRODUCTION: PROOF-THEORETIC SEMANTICS

Proof-theoretic semantics is an alternative to truth-condition semantics. It is based on the fundamental assumption that the central notion in terms of which meanings can be assigned to expressions of our language, in particular to logical constants, is that of *proof* rather than *truth*. In this sense proof-theoretic semantics is inherently inferential in spirit, as it is the inferential activity of human beings which manifests itself in proofs.

Proof-theoretic semantics has several roots, the most specific one being Gentzen's (1934) remarks that the introduction rules in his calculus of natural deduction define the meanings of logical constants, while the elimination rules can be obtained as a consequence of this definition. More broadly, it is part of the tradition according to which the meaning of a term should be explained by reference to the way it is *used* in our language.

Although the "*meaning as use*" approach has been quite prominent for half a century now and has provided one of the cornerstones of the philosophy of language, in particular of ordinary language philosophy, it has never prevailed in the *formal* semantics of artificial and natural languages. In formal semantics, the *denotational* approach, which starts with interpretations of singular terms and predicates, then fixes the meaning of sentences in terms of truth conditions, and finally defines logical consequence as truth preservation under all interpretations, has

always been dominant. The main reason for this, as I see it, is the fact that from the very beginning, denotational semantics received an authoritative rendering in Tarski's (1933) theory of truth, which combined philosophical claims with a sophisticated technical exposition and, at the same time, laid the ground for model theory as a mathematical discipline. Compared to this development, the "meaning as use" idea was a slogan supported by strong philosophical arguments, but without much formal underpinning.

There has been a lot of criticism of classical model-theoretic semantics from the denotational side itself. Examples are partial logics such as situation semantics, and dynamic approaches such as discourse representation theory and dynamic semantics. These logics reject the idea that *total* information about the world is always available and evaluate formulas with respect to certain information states.[1] Another example is Etchemendy's (1990) critique of classical consequence, which attracted much attention. However, in mainstream semantics, there has never been a fundamental reorientation, which could have turned the "meaning as use" idea into something that resembles a formalized theory.

Proof-theoretic semantics, as a sidestream development, attempts to achieve exactly this. As one would expect, it uses ideas from *proof theory* as a mathematical discipline, similar to the way truth-condition semantics relies on model theory. However, just this is the basis of a fundamental misunderstanding of proof-theoretic semantics. To a great extent, the development of mathematical proof theory has been dominated by the formalist reading of Hilbert's program as dealing with *formal* proofs exclusively, in contradistinction to model theory as concerned with the (denotational) *meaning* of expressions. This dichotomy has entered many textbooks of logic in which "semantics" means model-theoretic semantics and "proof theory" denotes the proof theory of formal systems. The result is that "proof-theoretic semantics" sounds like a contradiction in terms even today.

When I first used this term in the 1980s,[2] it was not very common, although the idea behind it was there in the Swedish school of proof theory established by Prawitz and Martin-Löf (see Kahle and Schroeder-Heister 2006). In the meantime, it has gained some ground and there have been some occasional references to it. Perhaps it will become more popular within general philosophy in the backwater of inferentialist approaches such as Brandom's[3], which more explicitly than ordinary language philosophy attempt to derive

denotational meaning from inferential meaning, i.e., use the idea that meaning is rooted in proofs as their starting point.

Strictly speaking, the formalist reading of proof theory is not any more foreign to the understanding of 'real' argumentation than model theory is to the interpretation of natural language. In order to apply proof-theoretic results, one has to consider formal proofs to be *representations* of proper arguments, just as, in order to apply model-theoretic methods, one has to consider formulas to be *representations* of proper sentences of a natural language like English. English is not *per se* a formal language, and arguments are not *per se* formal derivations. In this sense, the term "proof-theoretic semantics" is not any more provocative than Montague's (1970) conception of "English as a formal language". Both proof-theoretic semantics and model-theoretic semantics are *indirect* in that they can only be applied via a *formal reading* of aspects of natural language. The basic difference lies in what these aspects are: proof-theoretic semantics starts with arguments and represents them by derivations, whereas model-theoretic semantics starts with names and sentences and represents them by individual terms and formulas.

As indicated above, it was the Swedish school of proof theory, which paved the way for a non-formalist philosophical understanding of proofs. Although originally dealing with problems of the proof-theory of formal systems, Prawitz and Martin-Löf soon realized that many of the concepts and methods developed there had a non-technical counterpart when looking at formal proofs as formal representations of "genuine" proofs. In taking Gentzen's remarks on the definitional significance of introduction and elimination rules seriously, they developed the cornerstones of proof-theoretic semantics.

An immediate predecessor of proof-theoretic semantics was Tait (1967), who, in his work on the *convertibility* of terms, developed concepts which are closely related to those later employed in proof-theoretic semantics. Another predecessor was Lorenzen (1955), who, in his *operative logic*, used arbitrary production rules as definitional rules from which, by means of an inversion principle[4], corresponding elimination rules can be obtained.

In this paper I shall deal with *proof-theoretic validity* as one of the basic technical tools developed within proof-theoretic semantics. As this notion was essentially developed by Prawitz, my exposition is to a great extent a re-interpretation and, I hope, an improvement

to his approach. I shall not deal with the broader philosophical background of "anti-realism" and "verificationism" into which the concept of validity may be embedded, but mainly with the technical constructs and their (narrower) philosophical motivation. The reason for this is, besides lack of space, the fact indicated above that the desideratum of proof-theoretic semantics is not so much a *general* philosophical understanding of its position, but the formal development and philosophical clarification of its fundamental concepts. One result of this restriction is that I cannot give Dummett's work the attention it deserves, since his technical notions do not differ considerably from Prawitz's. I am well aware that he has made enormous contributions to the philosophical understanding of proof-theoretic semantics in general. To a considerable extent it is due to his work that the general climate is now more in favour of proof-theoretic semantics than it used to be.[5]

Validity is a property of derivations, or more general "derivation structures", which are considered to be representations of arguments. The format of these derivations is Gentzen-style natural deduction. In defining validity, attempts are made to justify arguments by turning certain proof-theoretic methods and results into semantic conditions, most prominently the following two: (1) Derivations can be simplified (or made more "direct") by certain reduction methods (terminating in *normal* derivations). (2) Assumption-free derivations in normal form are canonical (or "direct") in the sense that they apply an introduction rule in the last step. Valid arguments are then defined as derivation structures which exhibit properties like (1) and (2). However, I shall strictly distinguish between genuine semantic features and technical properties used in normalization proofs. This is extremely important, as Prawitz originally developed his semantic notion of validity along with adapting certain proof-theoretic concepts proposed by Tait and Martin-Löf to proofs of strong normalization. My main criticism of Prawitz will be that in his earlier writings on validity (Prawitz 1971, 1973, 1974) he does not sufficiently distinguish between semantic concepts and concepts used in proofs of (strong) normalization. I shall argue that they differ in fundamental respects.[6]

In spite of much philosophical discussion about meaning and theories of meaning, no thorough investigation of Prawitz's validity concept has been undertaken so far, although this concept is based on very elementary principles which are very close to Gentzen's original programme of justifying natural deduction. This is

why I chose this notion as my topic here. I want to leave open the question of whether validity should be taken as the *ultimate basis* of proof-theoretic semantics. I myself tend to favour a different approach which chooses rules as the unity of semantic investigation. Whereas in proof-theoretic validity in Prawitz's sense, derivations or arguments come first, and rules or consequences are regarded as steps which preserve the validity of arguments, a rule-based approach would first distinguish certain individual proof steps and then compose derivations or arguments from them. Whereas the first approach is *global*, dealing with proofs as a whole and imposing requirements on them, the second approach is *local*, as it interprets individual proof steps without demanding from the very onset that a proof composed of such single steps have special features. The rule-based approach has the advantage that the dependency of global features of arguments on local features of rules can be investigated separately, which makes this approach more flexible and capable of dealing with phenomena such as circular reasoning. Ideas in this direction have been developed in the context of logic programming jointly with Hallnäs[7], and will be dealt with in subsequent work.

This paper starts with recalling Gentzen's characterization of natural deduction and the way this characterization is turned into an inversion principle by Prawitz. The semantic validity concepts proposed are contrasted with concepts used in proofs of (strong) normalization, which were originally introduced by Tait and Martin-Löf. Special emphasis is placed on the difference between these concepts and semantic concepts, by calling those used for normalization "computability" and only the semantic ones "validity". Various forms of validity are defined and compared, among them notions of strict and strong validity which go beyond Prawitz's definitions. These notions are then extended to general derivation structures with arbitrary reductions serving as justifications, where the definition of a justification differs slightly from that of Prawitz. Finally, it is argued that *proof-theoretic* validity and the resulting notion of consequence is different from, and in a sense more specific than, *constructive* validity and consequence based on the notion of a constructive function.

For lack of space, Martin-Löf's meaning theory, which may be correctly viewed as carrying out a whole programme of proof-theoretic semantics, cannot be dealt with here (see e.g. Martin-Löf 1995, 1998). For the particular purpose of elucidating proof-theoretic

validity, this seems to me to be justified, since Martin-Löf's semantics is not explicitly concerned with formal notions of proof-theoretic validity. I cannot discuss Lorenzen's "operative logic" (1955) either, although it is very close to Gentzen's programme (at least "in spirit"). Furthermore, I do not consider categorical approaches to proof-theoretic semantics. The discussion about classical vs. intuitionistic logic is left out as well.[8] Even a rudimentary account of these items would turn this paper into a substantial monograph.

As a general framework, I use the implicational fragment of intuitionistic propositional logic, i.e. positive implicational logic, which suffices to demonstrate and exemplify all basic ideas. An adequate account of implication provides strong guidelines for the handling of other logical operators. Implication is the most complicated propositional operator, sharing crucial properties with universal quantification. The distinction between open and closed derivations, which will turn out to be semantically fundamental, is to a great extent due to its presence. It is intertwined with the notion of "assumption", which Gentzen gave a prominent role in logical calculi, and whose proper treatment is the cornerstone of proof-theoretic semantics.

Logical Preliminaries and Notational Conventions

In this paper, I stick to Prawitz's tree-based proof notation and do not use a term calculus via the Curry–Howard correspondence (although the typed λ-calculus would be a natural candidate). The tree-based proof notation is philosophically more natural, as proof terms obtain their philosophical significance through their reading as codes for "real" proofs.

Following Prawitz, I shall use the following conventions: If a derivation $\mathcal{D}$ ends with A, I shall also write $\begin{array}{c}\mathcal{D}\\A\end{array}$. If it depends on an assumption B, I shall write $\begin{array}{c}B\\\mathcal{D}\end{array}$ or $\begin{array}{c}B\\\mathcal{D}\\A\end{array}$. This means that the notations $\mathcal{D}$, $\begin{array}{c}\mathcal{D}\\A\end{array}$, $\begin{array}{c}B\\\mathcal{D}\end{array}$ and $\begin{array}{c}B\\\mathcal{D}\\A\end{array}$ do not denote different derivations, but just differ in what they make explicit. The *open assumptions* of a derivation are the assumptions on which the end-formula depends. A derivation is called *closed* if it contains no open assumptions, otherwise it is called *open*.

The system of natural deduction I shall use is that described by Gentzen (1934) and Prawitz (1965). Its positive implicational fragment contains only the schemata →-introduction and →-elimination (*modus ponens*):

$$
\begin{array}{cc}
\begin{array}{c}
[A] \\
B \\
\hline
A \to B
\end{array} \to \mathrm{I} &
\begin{array}{c}
A \to B \quad A \\
\hline
B
\end{array} \to \mathrm{E}
\end{array}
$$

The reduction of a *maximum formula*, which is a conclusion of an application of an introduction inference and at the same time the major premiss of an elimination inference, is in our restricted framework represented as the schema of →-reduction:

$$
\begin{array}{c}
A \\
\mathcal{D} \\
B \qquad \mathcal{D}' \\
\hline
A \to B \quad A \\
B
\end{array}
\quad \text{reduces to} \quad
\begin{array}{c}
\mathcal{D}' \\
A \\
\mathcal{D} \\
B
\end{array}
$$

Occasionally I shall also refer to the reductions for other connectives as described in Prawitz (1965). These reductions will be called the "standard reductions" (in contradistinction to arbitrary reductions for generalized derivation structures).

A derivation is in normal form if it cannot be further reduced, which means that it contains no maximum formula. Prawitz (1965) showed that by iterated application of reduction steps, every derivation in intuitionistic logic can be normalized, i.e., can be rewritten as a derivation in normal form.[9] One corollary of this result is that every closed derivation in intuitionistic logic can be reduced to one using an introduction rule in the last step, as a closed normal derivation is of exactly that form. I call this the *fundamental corollary* of normalization theory. As seen below, the fundamental corollary is philosophically interpreted by *requiring* that a valid closed derivation be reducible to one using an introduction inference in the last step. In this sense, introduction rules describe the basic meaning-giving inferences.

The normalization result mentioned is also called *weak normalization*. The *strong* normalization result says that any reduction sequence terminates in a normal derivation, regardless of the order in which reductions are performed. Methods used to prove strong normalization have provided the basis for semantic validity concepts.

2. GENTZEN'S PROGRAMME AND PRAWITZ'S INVERSION PRINCIPLE

Proof-theoretic semantics in the sense discussed in this paper goes back to certain programmatic remarks in Gentzen's *Investigations into Natural Deduction*, where he gives a semantic interpretation of his inference rules.

Gentzen's remarks deal with the relationship between introduction and elimination inferences in natural deduction.

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. This fact may be expressed as follows: In eliminating a symbol, we may use the formula with whose terminal symbol we are dealing only 'in the sense afforded it by the introduction of that symbol'. (Gentzen 1934, p. 80)

This cannot mean, of course, that the elimination rules are *deducible* from the introduction rules in the literal sense of the word; in fact, they are not. It can only mean that they can be *justified* by them in some way.

By making these ideas more precise it should be possible to display the E-inferences as unique functions of their corresponding I-inferences, on the basis of certain requirements. (Gentzen 1934, p. 81)

So the idea underlying Gentzen's programme is that we have "definitions" in the form of introduction rules and some sort of semantic reasoning which, by using "certain requirements", validate the elimination rules.

As indicated in the introduction, I shall not discuss in detail the philosophical reasons which might support Gentzen's programme. For that I would have to refer to Dummett's work and in particular to his claim that there are two different aspects of language use: one connected with 'directly' or 'canonically' asserting a sentence, and another one with drawing consequences from such an assertion.[10] The first is the primary or 'self-justifying' way corresponding to reasoning by introduction rules, whereas the second one, which corresponds to reasoning by elimination rules, is in need of justification. This justification relies on the *harmony* which is required to hold between both aspects: The possible *consequences* to be drawn from an assertion are determined by the *premisses* from which the assertion can possibly be inferred by direct means.

Prawitz, in an "inversion principle"[11] formulated in his classic monograph on *Natural Deduction* of 1965, tried to make Gentzen's remarks more precise in the following way.

Let α be an application of an elimination rule that has B as consequence. Then, deductions that satisfy the sufficient condition [...] for deriving the major premiss of α, when combined with deductions of the minor premisses of α (if any), already 'contain' a deduction of B; the deduction of B is thus obtainable directly from the given deductions without the addition of α. (Prawitz 1965, p. 33)

Here the sufficient conditions are given by the premisses of the corresponding introduction rules. Thus the inversion principle says that a derivation of the conclusion of an elimination rule can be obtained without an application of the elimination rule if its major premiss has been derived using an introduction rule in the last step, which means that a combination

$$\frac{\dfrac{\mathcal{D}}{A}\text{I-inference} \qquad \{\mathcal{D}_i\}}{B}\text{E-inference}$$

of steps, where $\{\mathcal{D}_i\}$ stands for a (possibly empty) list of deductions of minor premisses, can be avoided.

At first glance, this simply states the fact that maximum formulas, i.e. formulas being conclusions of an I-inference and at the same time major premiss of an E-inference (in the example: A), can be removed by means of certain reductions, which leads to the idea of a normal derivation. However, it also represents a *semantical* interpretation of elimination inferences by saying that nothing is gained by an application of an elimination rule if its major premiss has been derived *according to its meaning* (i.e., by means of an introduction rule). So the reductions proposed by Prawitz for the purpose of normalization are at the same time semantic justifications of elimination rules with respect to introduction rules. His inversion principle elaborates Gentzen's idea of "special requirements" needed for this justification, by demanding that elimination rules invert introduction rules in a precise sense.

That it corresponds indeed to what Gentzen had in mind can be seen by taking a closer look at the example Gentzen gives:

We were able to introduce the formula $A \rightarrow B$ when there existed a derivation of B from the assumption formula A. If we then wished to use that formula by eliminating the $\rightarrow$-symbol (we could, of course, also use it to form longer formulae, e.g., $(A \rightarrow B) \vee C$, $\vee$-I), we could do this precisely by inferring B directly, once A has been proved, for

what $A \rightarrow B$ attests is just the existence of a derivation of B from A. (Gentzen 1934, pp. 80–81)

This may be read as follows: Given the situation

$$
\begin{array}{cc}
A & \\
\mathcal{D} & \\
B & \mathcal{D}' \\
\hline
A \rightarrow B & A \\
& B
\end{array}
$$

where $\mathcal{D}$ is "a derivation of B from the assumption formula A", and $\mathcal{D}'$ is the derivation showing that "A has been proved", so that we can use $A \rightarrow B$ to obtain B "by eliminating the $\rightarrow$-symbol". Then by means of

$$
\begin{array}{c}
\mathcal{D}' \\
A \\
\mathcal{D} \\
B
\end{array}
$$

we can infer "B directly, once A has been proved [by means of $\mathcal{D}'$]", as "$A \rightarrow B$ attests [...] the existence of a derivation [viz. $\mathcal{D}$] of B from A". According to this reading, Gentzen describes the standard reduction for implication later made explicit by Prawitz (1965) and used in his normalization proof.

However, although Gentzen's remarks are correctly read as outlining a semantic programme, he himself takes a more formalistic stance, which is clear from his writings in general and from the continuation of the passage quoted above:

Note that in saying this we need not go into the 'informal sense' ['inhaltlicher Sinn']^{12} of the $\rightarrow$-symbol. (Gentzen 1934, p. 81)

Prawitz (1965) deserves credit to have drawn our attention to the genuine semantic content of Gentzen's remarks, though this is not spelled out in detail in his monograph. Only later in Prawitz (1971) and in particular in Prawitz (1973, 1974) is it turned into a full-fledged semantic theory.

3. NORMALIZATION, COMPUTABILITY AND VALIDITY

3.1. *Normalization and Computability*

Normalization plays a prominent role in the formal background of proof-theoretic semantics, in particular the result that normal closed

proofs are in introduction form, i.e., use an introduction inference in the last step.

Of equal importance is a technical method within normalization theory, which is especially used in proofs of *strong* normalization. By means of this method, a certain predicate P of proofs is defined which has the property that it entails (strong) normalizability. The predicate P has some flavour of a semantic predicate, and in a kind of correctness proof it can be shown that every derivation satisfies P, yielding as a corollary that every derivation is (strongly) normalizable. Such a predicate was first defined by Tait (1967) under the name "convertibility" and used to demonstrate (weak) normalizability of terms. Martin-Löf (1971) carried Tait's idea over from terms to derivations and defined a corresponding predicate which he called "computability", proving (weak) normalization for an extension of first-order logic, called the theory of iterated inductive definitions. At the same time, Girard (1971) used this method to prove (weak) normalization for second-order logic. Again at the same time, it was Prawitz (1971) who emphasized its particular usefulness for proving *strong* normalization, calling it "strong validity". Since then, it has served as the basis of proofs of strong normalization for a variety of systems.[13]

In the following I shall speak of *computability* predicates or *the computability* predicate when dealing with this notion as it is used in normalization proofs, thus adopting Martin-Löf's terminology. The term "valid" will be reserved for genuinely semantic notions. I consider the terminology of Prawitz, who speaks of "validity based on the introduction rules" (1971, p. 284) in contradistinction to "validity used in proofs of normalizability" (1971, p. 290), somewhat unfortunate. It is one of the basic claims of this paper that there are fundamental differences between these two concepts.

I restrict Prawitz's notion of computability ("validity used in proofs of normalizability") to positive implicational logic $\mathcal{L}$, i.e., to the system with only introduction and elimination rules for implications as primitive rules of inference. Under this restriction, Prawitz's computability notion is basically the same as Martin-Löf's.

A derivation is in *I-form* if it uses an introduction rule in the last step, i.e., if it is of the form

$$\begin{array}{c} [A] \\ \mathcal{D} \\ B \\ \hline A \to B \end{array}.$$

Using a term employed by Prawitz (1974) and Dummett (1975) and now common in proof-theoretic semantics, such a derivation is also called *canonical*. Let $\mathcal{D} \succ_1 \mathcal{D}'$ mean that the derivation $\mathcal{D}$ reduces to the derivation $\mathcal{D}'$ by applying a single reduction step to a subderivation of $\mathcal{D}$.

DEFINITION OF COMPUTABILITY

(i) A derivation of the form
$$\begin{array}{c} [A] \\ \mathcal{D} \\ B \\ \hline A \to B \end{array}$$
is *computable*, if for every computable
$$\begin{array}{c} \mathcal{D}' \\ A \end{array},$$
$$\begin{array}{c} \mathcal{D}' \\ A \\ \mathcal{D} \\ B \end{array}$$
is *computable*.

(ii) If a derivation $\mathcal{D}$ is not in I-form and is normal, then it is *computable*.

(iii) If a derivation $\mathcal{D}$ is not in I-form and is not normal, then $\mathcal{D}$ is *computable*, if every $\mathcal{D}'$, such that $\mathcal{D} \succ_1 \mathcal{D}'$, is *computable*.

This is a generalized inductive definition. It uses induction on the degree of the end formula of the derivation (clause i), and, within each degree, induction on the reducibility relation[14] (clauses ii and iii).

The proof of strong normalization then proceeds by establishing the following two propositions:

PROPOSITION 1. Every computable derivation is strongly normalizable.

PROPOSITION 2. Every derivation is computable.

Proposition 1 is a (nearly) immediate consequence of the definition of computability. Proposition 2 is based on a kind of correctness proof, verifying step by step that computability is carried over from the premises to the conclusion of an inference step. Other formulations of "computability" differ slightly from the one given here. However, the basic features remain the same. The resulting normalization proofs all proceed via Propositions 1 and 2.

Computable derivations are closed under substitution with computable derivations, i.e., the following lemma holds:

SUBSTITUTION LEMMA FOR COMPUTABILITY

If $\begin{array}{c} A_1 \dots A_n \\ \mathcal{D} \\ B \end{array}$ is computable, where all open assumptions of $\mathcal{D}$ are among $A_1, \dots, A_n$, then for any list of computable derivations $\begin{array}{c} \mathcal{D}_i \\ A_i \end{array}$ $(1 \leq i \leq n),$ $\begin{array}{c} \mathcal{D}_1 \qquad \mathcal{D}_n \\ A_1, \dots, A_n \\ \mathcal{D} \\ B \end{array}$ is computable.

Note that the converse direction of the lemma is trivial, as every assumption A_i is itself a normal, and therefore computable, derivation of A_i from A_i.

If closure under substitution with computable derivations is called *computability under substitution*, the lemma says that computability implies computability under substitution.

Weaker versions of computability entail (weak) normalization. Instead of requiring in clause (iii) that *every* $\mathcal{D}'$, such that $\mathcal{D} \succ_1 \mathcal{D}'$, be computable, we might demand that a certain $\mathcal{D}'$, which is obtained from $\mathcal{D}$ *in a particular way* (i.e., by performing a particular reduction step) be computable. This yields the notion defined by Martin-Löf (1971). We might even weaken this by not referring to a particular procedure and just postulate in (iii) that $\mathcal{D}$ reduces to a computable $\mathcal{D}'$, without specifying the procedure in the definition (it must then be specified in the normalization proof, of course).

3.2. *From Computability to Validity*

Validity is a core notion of proof-theoretic semantics. Prawitz introduced it as a semantic predicate for derivations, in analogy to truth as a semantic predicate of propositions in model-theoretic semantics. He developed it in connection with computability predicates, to which it bears a strong resemblance. As his terminology ("strong validity" for "computability" in our sense) suggests, Prawitz actually considers computability and validity to be concepts on one scale, computability being the stronger one. There are several remarks in his 1971, 1973, 1974 papers, where he deals with both notions, which indicate that computability is obtained by augmenting validity, some of them even stating that these extensions make the concept of validity more plausible or convenient.[15] However, Prawitz never explains the exact relationship between these two concepts. In

particular, he never attempts to formally prove that computability (strong validity) implies validity – a result one should expect to hold if the relationship is as simple as the terminology suggests. In his publications after 1974, Prawitz never returns to computability and its relation to validity.

In the following I shall argue that, in spite of many similarities, and contrary to Prawitz's opinion, semantically useful validity notions must differ considerably from computability. Crucial modifications are necessary to turn computability into validity. I shall make the following points:

(1) The notion of computability is not suitable as a foundational semantic notion, because it stipulates normal derivations as computable without further justification.
(2) In order to adjust the notion of computability to serve foundational purposes, *closed* derivations must be given a distinguished role in the justification of irreducible (= normal) derivations.
(3) This distinguished role of closed derivations includes, as a semantic condition, their reducibility to canonical form.

Ad (1): Computability is not a semantic notion

According to clause (ii) in the definition of computability, every normal derivation which is not in I-form, is computable.[16] This could be counted as a semantic clause only if in proof-theoretic semantics we are prepared to consider non-canonical normal derivations as valid *by definition*. However, as we have seen, it is one of the ideas of proof-theoretic semantics in the sense of Gentzen's programme to consider introduction inferences as basic and to *justify* all other inferences by them. In other words, only derivations based on introduction rules should be taken for granted. In any other case the definition of validity should rely on some *justification* procedure rather than on the syntactic form of derivations. This is obviously violated by clause (ii), which simply stipulates irreducible non-canonical derivations as valid. There is no semantic reason whatsoever to consider non-canonical irreducibility as a definition case of validity. According to such a definition, the derivation

$$\frac{A \to B \quad A}{B}$$

would be valid *by definition* and not *by justification*, which is not what is intended. *Modus ponens*, as an elimination rule, definitely needs semantic justification. Of course, for the purpose of proving normalizability, clause (ii) is absolutely natural, as normal derivations are trivially (strongly) normalizable. For semantic purposes, however, we would have to argue that non-canonical irreducible derivations have some special status, which exempts them from justification. Since there is no argument at hand to support this, using normal derivations as a starting point in defining validity is an ill-guided approach.

In contradistinction to clause (ii), clauses (i) and (iii) make good semantic sense. In terms of validity, clause (i) says that a canonical derivation of $A \rightarrow B$ is valid if its immediate predecessor, a derivation of B from A, provides a way of transferring every valid derivation of A into a valid derivation of B, which corresponds to the meaning one wants to associate with $A \rightarrow B$. Furthermore, clause (iii) says that a non-canonical derivation may be considered as valid if it reduces to a valid derivation. This reflects the idea that non-canonical derivations are valid if they reduce to derivations which are already justified as valid (such as canonical ones).

Therefore the *basic flaw* in computability, understood as a semantic notion, is the following implicit assumption:

If $\mathcal{D}$ is non-canonical and irreducible (=normal), then $\mathcal{D}$ is valid.

Ad (2): Semantically modified computability: open assumptions and closed derivations

One could try to modify the definition of computability to make it suitable for a definition of semantic validity. This would mean that clause (ii) of the definition is dropped and replaced with something which justifies non-canonical irreducible derivations as valid. An obvious possibility would be to consider such a derivation as valid if the replacement of open assumptions with valid derivations yields a valid derivation of the end formula. This idea would follow the substitution lemma for computability, according to which computability is the same as computability under substitution. More formally, clause (ii) would then read as follows (where we now use the term "valid", as we are dealing with turning the computability notion into a semantic concept):

(ii)* A non-canonical irreducible derivation

$$\begin{array}{c} A_1 \ldots A_n \\ \mathcal{D} \\ B \end{array}$$

where all open assumptions of $\mathcal{D}$ are among $A_1, \ldots, A_n$, is *valid*, if for every list of *valid* derivations $\begin{array}{c} \mathcal{D}_i \\ A_i \end{array}$ $(1 \le i \le n)$,

$$\begin{array}{cc} \mathcal{D}_1 & \mathcal{D}_n \\ A_1 & \ldots A_n \\ & \mathcal{D} \\ & B \end{array}$$

is *valid*.

For example, the one step non-canonical irreducible derivation

$$\frac{A \to B \quad A}{B}$$

would be considered as valid, if for each pair of valid derivations $\begin{array}{c} \mathcal{D}_1 \\ A \to B \end{array}$ and $\begin{array}{c} \mathcal{D}_2 \\ A \end{array}$, the derivation $\dfrac{A \to B \quad A}{B}$ with $\begin{array}{cc} \mathcal{D}_1 & \mathcal{D}_2 \end{array}$ is valid. However, a clause like (ii)* would then no longer proceed by induction on the complexity of the end formula but on the complexities of the open assumptions plus that of the end formula, in the example: on the complexities of $A \to B$, A and B. But then the quantification over all valid derivations of the open assumptions is no longer feasible, since these derivations may depend on assumptions of arbitrary complexity. Therefore this is no viable solution.[17]

The way out of this problem used in semantic definitions of validity is to use *closed* valid proofs rather than arbitrary valid proofs as a basis. Instead of (ii)*, one would then propose the following clause.

(ii)** A non-canonical non-reducible derivation

$$\begin{array}{c} A_1 \ldots A_n \\ \mathcal{D} \\ B \end{array}$$

where all open assumptions of $\mathcal{D}$ are among $A_1, \ldots, A_n$, is *valid*, if for every list of *closed valid* derivations $\begin{matrix} \mathcal{D}_i \\ A_i \end{matrix}$ $(1 \leq i \leq n)$,

$$
\begin{matrix}
\mathcal{D}_1 & \mathcal{D}_n \\
A_1 \ldots A_n \\
\mathcal{D} \\
B
\end{matrix}
$$

is *valid*.

However, even now we are proceeding by induction on the joint complexity of $A_1, \ldots, A_n, B$ rather than only the complexity of B, even if we only quantify over *closed* valid derivations. This is not compatible with clause (i), where we proceed by induction on the end formula only. In order to cope with that, we would also have to change clause (i) to

(i)** A derivation of the form $\begin{matrix} [A] \\ \mathcal{D} \\ B \\ \hline A \to B \end{matrix}$ is *valid*, if for every *closed*

valid $\begin{matrix} \mathcal{D}' \\ A \end{matrix}$, $\begin{matrix} \mathcal{D}' \\ A \\ \mathcal{D} \\ B \end{matrix}$ is *valid*

where this is understood as proceeding by induction on the joint complexity of open assumptions plus end-formula of a derivation.

The definition based on (i)**, (ii)** and (iii) may be called *validity**. In passing from computability to validity* we have interpreted *open assumptions as placeholders for closed derivations*.

Ad (3): The reducibility of closed derivations

Unfortunately, validity* does not yet eliminate the possibility that irreducible (= normal) derivations are considered valid without any further justification. In the case of open derivations, this possibility has been removed, but not so in the case of closed derivations. Suppose $\mathcal{D}$ is a closed non-canonical derivation which is irreducible. Then clause (ii)** applies, and, as there are no open assumptions, $\mathcal{D}$ is (vacuously) valid*.

One might argue that there are no closed non-canonical irreducible derivations. However, this is an accidental property of first-order logic with the standard reductions. Since the notion of validity should in principle be applicable to more general notions of derivations and reductions, the formal possibility of closed non-canonical irreducible derivations must be taken into account. Such a derivation should simply turn out to be invalid by definition. This is accomplished by transforming a corollary of the normalization of proofs into a semantic condition:

A closed non-canonical derivation is valid, if it is reducible to a valid closed canonical derivation.

It was Dummett in particular who repeatedly stressed as a fundamental epistemological principle[18] that, if something is known in an indirect (non-canonical) way, it must be possible to turn this indirect knowledge into direct (canonical) knowledge. This is part of the reason why this sort of semantics is also called verificationist, and it is part of the interpretation of Gentzen's programme of the primacy of introduction rules: In the closed case an I-rule derivation can always be found. With this motivation we arrive at Prawitz's definition of the validity of derivations.

3.3. *Validity of Derivations*

We follow Prawitz (1971) in defining validity with respect to atomic systems S, which are given by production rules for atomic formulas. Let then $\mathcal{L}(S)$ be implicational logic over S, i.e., the system given by introduction and elimination rules for implication plus the production rules of S. We may identify $\mathcal{L}(\mathcal{S})$ with the set of all derivations in this system. A system S' is an *extension* of S ($S' \geq S$) if S' is S itself or results from S by adding further production rules. As a limiting case, we consider the empty atomic system S_0 without any inference rules and with propositional variables as formulas, and correspondingly $\mathcal{L}(S_0)$ as standard implicational logic over propositional variables. Obviously, as a formal system, $\mathcal{L}(\mathcal{S}_0)$ is the same as $\mathcal{L}$. It will turn out that validity with respect to S_0 is the same as *universal* validity when defined in an appropriate way. We say that $\mathcal{D}$ *reduces to* $\mathcal{D}'$ ($\mathcal{D} \succeq \mathcal{D}'$), if $\mathcal{D}'$ can be obtained from $\mathcal{D}$ by applying a (finite) number of reduction steps. As a limiting case, $\mathcal{D}$ reduces to itself. In the context of atomic systems, we also extend the notion of a canonical derivation. A canonical derivation of an

atom of S is a derivation in S, whereas, as before, a canonical der-
ivation of a complex formula is a derivation in I-form, i.e., a deri-
vation using an introduction rule in the last step.

Then our first definition of validity corresponding to the one
given in Prawitz (1971) runs as follows:

DEFINITION OF S-VALIDITY (1)

(i) For atomic A, a closed derivation of A is *S-valid*, if it reduces
to a derivation in S.

(ii) A closed derivation $\begin{array}{c}\mathcal{D}\\ A \to B\end{array}$ is *S-valid*, if $\mathcal{D}$ reduces to a

derivation of the form $\begin{array}{c}[A]\\ \mathcal{D}'\\ B\\ \hline A \to B\end{array}$ such that for every $S' \geq S$ and

every closed S'-valid $\begin{array}{c}\mathcal{D}''\\ A\end{array}$, $\begin{array}{c}A\\ \mathcal{D}'\\ B\end{array}$ is S'-valid.

(iii) An open derivation $\begin{array}{c}A_1 \ldots A_n\\ \mathcal{D}\\ B\end{array}$, where all open assumptions of $\mathcal{D}$
are among $A_1, \ldots, A_n$, is *S-valid*, if for every $S' \geq S$ and every

list of closed S'-valid $\begin{array}{c}\mathcal{D}_i\\ A_i\end{array}$ $(1 \leq i \leq n)$, $\begin{array}{c}\mathcal{D}_1 \quad \mathcal{D}_n\\ A_1 \ldots A_n\\ \mathcal{D}\\ B\end{array}$ is S'-valid.

This inductive definition proceeds on the joint complexities of the
open assumptions and the end formula of the given derivation.

In view of clause (iii), clause (ii) can be changed to

(ii) A closed derivation of $A \to B$ is *S-valid* if it reduces to a canonical
derivation of $A \to B$ whose immediate subderivation is *S-valid*.

By putting reduction into a clause of its own, the whole defini-
tion can then be equivalently stated as follows:

DEFINITION OF S-VALIDITY (2)

(I) Every closed derivation in S is *S-valid*.

(II) A closed canonical derivation of $A \to B$ is *S-valid*, if its imme-
diate subderivation is *S-valid*.

(III) A closed non-canonical derivation is *S-valid*, if it reduces to an *S-valid* canonical derivation.

(IV) An open derivation $\begin{array}{c} A_1 \ldots A_n \\ \mathcal{D} \\ B \end{array}$, where all open assumptions of $\mathcal{D}$ are among $A_1, \ldots, A_n$, is *S-valid*, if for every $S' \geq S$ and for every list of closed S'-valid $\begin{array}{c} \mathcal{D}_i \\ A_i \end{array} (1 \leq i \leq n)$, $\begin{array}{c} \mathcal{D}_1 \quad \mathcal{D}_n \\ A_1 \ldots A_n \\ \mathcal{D} \\ B \end{array}$ is *S'-valid*.

The equivalence of these two definitions of *S*-validity is easy to prove. Obviously, every (not necessarily closed) derivation in *S* is *S*-valid, since every closed *S*-valid derivation of an atom reduces to a derivation in *S*. The second definition corresponds to the one proposed by Prawitz (1974, 2006). As explained in the last subsection, the philosophical motivation behind this definition is that, in the closed case, derivations in *S* as well as introduction steps are self-justifying (clauses I and II), whereas all other steps are justified on the basis that they *reduce* to something which is already justified (clause III), or, in the open case, produce justified closed derivations when combined with such derivations (clause IV).

The reason for considering arbitrary extensions S' of S, is to block arguments for *S*-validity based on the underivability of certain formulas in *S*. Otherwise, for example, every derivation in $\mathcal{L}$ starting with a propositional variable as an open assumption, should be counted as S_0-valid, because there is no closed derivation of a propositional variable in S_0. In this sense, the consideration of extensions $S' \geq S$ is a *monotonicity* condition for *S*-validity. *S*-valid derivations should remain *S-valid* if one's knowledge incorporated in the atomic system *S* is increased.[19] In fact, it is easy to show that we have a

MONOTONICITY THEOREM FOR *S*-VALIDITY
A derivation $\mathcal{D}$ in $\mathcal{L}(S)$ is *S*-valid iff for every $S' \geq S$, $\mathcal{D}$ is *S'*-valid.

Investigating the consequences of permitting non-monotonicity of *S*-validity is beyond the scope of this paper.

As compared to computability, this definition relies on two crucial insights:

(1) The distinction between closed and open derivations is primary as compared to that between canonical and non-canonical derivations. The latter plays the role of a subdistinction within

closed derivations. In the definition of *S*-validity, we proceed according to the concept tree

$$\left.\begin{array}{l} closed \left|\begin{array}{l} canonical \\ non\text{-}canonical \end{array}\right. \\ \\ open \end{array}\right.$$
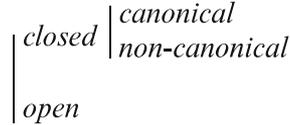
whereas the definition of computability rests on

$$\left.\begin{array}{l} canonical \\ non\text{-}canonical \end{array}\right. \quad \left|\begin{array}{l} reducible \\ irreducible \end{array}\right.$$

In *S*-validity, closed canonical derivations are self-justifying, carrying the burden of semantic justification. In computability, this holds of non-canonical irreducible (= normal) derivations.[20]

(2) The reduction clause for closed derivations (clause III) uses an existence condition corresponding to *weak* normalization, which is again due to the self-justifying character of closed canonical derivations. Whereas in computability, self-justifying derivations are by definition tied to the reducibility concept, viz. as derivations which are *irreducible*, in *S*-validity self-justifying derivations are defined independently of reducibility and are not trivially available when a derivation is not reducible, which means that we have to postulate their existence as a result of reduction.

For our case of implicational logic we can easily show the following:

SOUNDNESS THEOREM FOR *S*-VALIDITY
For any *S*, every derivation in $\mathcal{L}(S)$ is *S*-valid.

3.4. *Validity and Universal Validity*

Universal validity will be defined for derivations in $\mathcal{L}$. Intuitively, a derivation in $\mathcal{L}$ should be universally valid if it is *S*-valid for every *S*. For that, we must interpret derivations of $\mathcal{L}$ in $\mathcal{L}(S)$. Let an *S*-assignment v be a mapping of propositional variables to *S*-formulas. Then for an $\mathcal{L}$-derivation $\mathcal{D}$, $\mathcal{D}^v$ is the $\mathcal{L}(S)$-derivation resulting from $\mathcal{D}$ by replacing every propositional variable with the corresponding *S*-formula assigned to it via v. We can then say that $\mathcal{D}$ is *valid in S under* v, if $\mathcal{D}^v$ is *S*-valid in the sense defined in the previous section. $\mathcal{D}$ is then called *valid in S* if it is valid in *S* under every

v, and it is called *universally valid*, if it is valid in S for every S. Now the following can be shown to hold:

PROPOSITION   Let $\mathcal{D}$ be a derivation in $\mathcal{L}$. Then $\mathcal{D}$ is universally valid iff $\mathcal{D}$ is S_0-valid.

Proof. We use the fact that when $\mathcal{L}$ is interpreted in $\mathcal{L}(S)$, every extension $S' \geq S$ can be viewed as an interpretation of an extension of S_0 via an assignment.   □

Therefore, from now on we shall use the term "valid" terminologically as meaning universal or S_0-validity.

Then as a corollary of the soundness theorem for S-validity we have the following:

SOUNDNESS THEOREM FOR VALIDITY
Every derivation in $\mathcal{L}$ is valid.

As we have a corresponding theorem for computability (Proposition 2), and as we are so far only considering derivations in implicational logic, computability and validity coincide in the sense that any computable derivation (i.e., any derivation in implicational logic) is a valid derivation (i.e., a derivation in implicational logic) and vice versa. So extensionally, computability and validity coincide. We can differentiate between them when we consider more general notions of derivation structures. Then we can give actual counterexamples which show that computability and validity differ not only with respect to their contents, but are in fact *extensionally* different concepts (see Section 6). This further substantiates our claim that, contrary to Prawitz, computability is at best a forerunner to validity but not a semantic concept in itself.

3.5. *Validity Concepts which Imply Normalizability: Strict and Strong Validity*

Our basic semantic argument against computability and for validity was that irreducible derivations should never be counted as valid without further justification, i.e., the implication

       irreducible implies valid

should *not* hold by definition. One might, however, expect that the implication

valid implies normalizable

holds.[21] According to the present definition of validity, normalizability is not implied by validity. If we consider intuitionistic logic with no introduction rule for absurdity $\perp$, then according to our definition of validity, $\dfrac{\perp}{\mathcal{D}}$ is vacuously valid for any $\mathcal{D}$ with $\perp$ as the only open assumption, even if $\mathcal{D}$ is not normalizable. Now one might argue that a semantic justification of open derivations in terms of substitution with closed valid derivations should only be applied if the derivation is reduced as far as possible, and not already in a situation, where $\mathcal{D}$ can still be reduced. This means that the substitution justification in clause (IV) of the definition of *S*-validity should be put into action only if all possibilities of obtaining a justification by means of reduction are exhausted, i.e., when the derivation in question is irreducible. Calling this notion "strict *S*-validity" (or "strict validity" [simpliciter] for the universal concept), we reach the following definition:

DEFINITION OF STRICT *S*-VALIDITY

(I) Every closed derivation in *S* is *strictly S-valid*.

(II) A closed canonical derivation of $A \rightarrow B$ is *strictly S-valid*, if its immediate subderivation is *strictly S-valid*.

(III) A closed non-canonical derivation is *strictly S-valid*, if it reduces to a *strictly S-valid* canonical derivation.

(IV) An open reducible derivation is *strictly S-valid*, if it reduces to a *strictly S-valid* derivation.

(V) An open irreducible derivation $\begin{array}{c} A_1 \ldots A_n \\ \mathcal{D} \\ B \end{array}$, where all open assumptions of $\mathcal{D}$ are among $A_1, \ldots, A_n$, is *strictly S-valid*, if for every $S' \geq S$ and for every list of closed and *strictly S'-valid* $\begin{array}{c} \mathcal{D}_i \\ A_i \end{array} (1 \leq i \leq n)$, $\begin{array}{c} \mathcal{D}_1 \quad \mathcal{D}_n \\ A_1 \ldots A_n \\ \mathcal{D} \\ B \end{array}$ is *strictly S'-valid*.

The difference to the definition of *S*-validity is that clause (IV) is split up into clauses (IV) and (V), where the new clause (IV) demands the reduction of reducible open derivations, while the new clause (V) is the old clause (IV), but applied only to the irreducible case. So the conceptual tree of this definition is the following one

$$
\left|
\begin{array}{l}
closed \left|
\begin{array}{l}
canonical \\
non\text{-}canonical
\end{array}
\right. \\[2ex]
open \left|
\begin{array}{l}
reducible \\
irreducible
\end{array}
\right.
\end{array}
\right.
$$

which contrasts sharply with computability, where the reducible/ irreducible distinction is a subdistinction of non-canonical derivations.

I speak of "strict" rather than "strong" *S*-validity to distinguish it from Prawitz's notion of strong validity, which corresponds to computability, and from associations with strong normalization. Furthermore, I should like to reserve "strong *S*-validity" for a notion defined below for which this association is justified. Strict *S*-validity as considered here is indeed a notion on the same scale as *S*-validity. It is obvious that strict *S*-validity implies *S*-validity, but not necessarily vice versa.[22] The corresponding universal notion of strict validity (simpliciter) is defined as in Subsection 3.4.

Let us define (weak) normalizability inductively as follows:

DEFINITION OF NORMALIZABILITY

 (i) Every canonical derivation is *normalizable* if its immediate sub-derivation is *normalizable*.
 (ii) Every non-canonical normal derivation is *normalizable*.
(iii) Every non-canonical reducible derivation is *normalizable*, if it reduces to a *normalizable* derivation.

We can then formulate as a theorem that strict validity implies (weak) normalizability.

THEOREM. Every strictly valid derivation is normalizable.

By *strong S-validity* we denote a further strengthened concept, which implies strong normalization.

DEFINITION OF STRONG *S*-VALIDITY

 (I) Every closed derivation in *S* is *strongly S-valid*.
 (II) A closed canonical derivation of $A \to B$ is *strongly S-valid*, if its immediate subderivation is *strongly S-valid*.

(III) A closed non-canonical derivation $\mathcal{D}$ is *strongly S-valid*, if $\mathcal{D}$ is reducible, and if every $\mathcal{D}'$, such that $\mathcal{D} \succ_1 \mathcal{D}'$, is *strongly S-valid*.

(IV) An open reducible derivation $\mathcal{D}$ is *strongly S-valid*, if every $\mathcal{D}'$, such that $\mathcal{D} \succ_1 \mathcal{D}'$, is *strongly S-valid*.

(V) An open irreducible derivation $\begin{array}{c} A_1 \ldots A_n \\ \mathcal{D} \\ B \end{array}$, where all open assumptions of $\mathcal{D}$ are among $A_1, \ldots, A_n$, is *strongly S-valid*, if for every $S' \geq S$ and for every list of closed and *strongly S'-valid* $\begin{array}{c} \mathcal{D}_i \\ A_i \end{array}$ $(1 \leq i \leq n)$, $\begin{array}{c} \mathcal{D}_1 \quad \mathcal{D}_n \\ A_1 \ldots A_n \\ \mathcal{D} \\ B \end{array}$ is *strongly S'-valid*.

Obviously, strong *S*-validity implies strict *S*-validity.

A corresponding universal notion of strong validity (simpliciter) is defined as in Subsection 3.4. We extend the definition of normalizability to a definition of strong normalizability by replacing "if it reduces to" with "if every derivation it reduces to in a single step is" in clause (iii) of this definition. In analogy with the case of strict validity, we can show that strong validity implies strong normalizability.

THEOREM. Every strongly valid derivation is strongly normalizable.

There are also soundness theorems for strict and strong [*S*-]validity.

SOUNDNESS THEOREMS FOR STRICT AND STRONG *S*-VALIDITY

All [*S*-]derivations are both strictly and strongly [*S*-]valid.

With strict and strong validity we have obtained concepts which are semantically satisfying and at the same time imply weak and strong normalization, respectively.

4. VALIDITY AND COMPUTABILITY BASED ON ELIMINATION RULES

A central idea of proof-theoretic semantics is to consider one set of rules as basic and justify derivations based on other rules with respect to this first set of rules as valid. The standard approach is to consider the introduction rules as primitive or "self-justifying" (Dummett).

However, as envisaged by Prawitz[23], one might try an approach from the opposite direction, starting with elimination inferences. Prawitz's presentation is very sketchy. I reconstruct it as follows:

According to the I-rule conception, if in $\begin{smallmatrix}\mathcal{D}\\A\end{smallmatrix}$ the formula A is the conclusion of an introduction rule whose premiss derivation is S-valid, then $\mathcal{D}$ is S-valid by definition. If A is not derived by an introduction rule, $\mathcal{D}$ is S-valid if it can be *reduced* to an S-valid derivation. Analogously, one might postulate within an E-rule conception that, if all applications of elimination rules to the end-formula A of $\mathcal{D}$ yield S-valid derivations, then $\mathcal{D}$ is itself S-valid by definition. If no elimination rule can be applied to A, then $\mathcal{D}$ is S-valid if it can be *reduced* to an S-valid derivation. (Obviously, the latter case only arises when A is atomic.)

This suggests the following definition.

DEFINITION OF *S*-VALIDITY BASED ON ELIMINATION RULES

(I) Every closed derivation in S is *S-valid$_E$*.

(II) A closed derivation $\begin{smallmatrix}\mathcal{D}\\A \to B\end{smallmatrix}$ of $A \to B$ is *S-valid$_E$*, if for every $S' \geq S$ and every closed *S'-valid$_E$* $\begin{smallmatrix}\mathcal{D}'\\A\end{smallmatrix}$, the (closed) derivation

$$\frac{\begin{matrix}\mathcal{D} & \mathcal{D}'\\ A \to B & A\end{matrix}}{B}$$ is *S'-valid$_E$*.

(III) A closed derivation $\begin{smallmatrix}\mathcal{D}\\A\end{smallmatrix}$ of an atomic formula A, which is not a derivation in S, is *S-valid$_E$*, if it reduces to a derivation in S.

(IV) An open derivation $\begin{smallmatrix}A_1 \ldots A_n\\ \mathcal{D}\\B\end{smallmatrix}$, where all open assumptions of $\mathcal{D}$ are among $A_1, \ldots, A_n$, is *S-valid$_E$*, if for every $S' \geq S$ and for every list of closed *S'-valid$_E$* $\begin{smallmatrix}\mathcal{D}_i\\A_i\end{smallmatrix}$ $(1 \leq i \leq n)$, $\begin{smallmatrix}\mathcal{D}_1 & & \mathcal{D}_n\\ A_1 \ldots A_n\\ \mathcal{D}\\B\end{smallmatrix}$ is

S'-valid$_E$.

Clause (IV) is identical with clause (IV) in the definitions of *S*-validity in Section 3.3, i.e., open assumptions in derivations are

interpreted in the same way as they were previously. Clauses (I) and (III) can be conjoined to form the single clause

(I/III) A closed derivation $\begin{smallmatrix} \mathcal{D} \\ A \end{smallmatrix}$ of an atomic formula A is *S-valid$_E$*, if it reduces to a derivation in S.

Using the main reductions, it can again be shown that all derivations in $\mathcal{L}(S)$ are *S*-valid$_E$.

As Prawitz remarks, this approach only works for logical constants with "direct" elimination rules such as $\rightarrow$, $\wedge$ and $\forall$. There is no way to extend this to constants like $\vee$ and $\exists$ with "indirect" elimination rules.

Corresponding to the procedure in Section 3.5, notions of strict *S*-validity$_E$ and strong *S*-validity$_E$ can be defined such that strict *S*-validity$_E$ implies weak normalizability and strong *S*-validity$_E$ implies strong normalizability.[24]

There is also a corresponding notion of computability based on elimination rules for the purpose of strong normalization proofs. Actually, this notion is more common in today's presentations than computability based on introduction rules, as long as one does not have to deal with $\exists$ or $\vee$. For example, Troelstra and Schwichtenberg (1996) define computability as follows:

DEFINITION OF COMPUTABILITY BASED ON ELIMINATION RULES

(1) For atomic A, $\begin{smallmatrix} \mathcal{D} \\ A \end{smallmatrix}$ is *computable$_E$*, if $\begin{smallmatrix} \mathcal{D} \\ A \end{smallmatrix}$ is strongly normalizable.

(2) $\begin{smallmatrix} \mathcal{D} \\ A \rightarrow B \end{smallmatrix}$ is *computable$_E$*, if for every *computable$_E$* $\begin{smallmatrix} \mathcal{D}' \\ A \end{smallmatrix}$, $\dfrac{\begin{smallmatrix} \mathcal{D} \\ A \rightarrow B \end{smallmatrix} \quad \begin{smallmatrix} \mathcal{D}' \\ A \end{smallmatrix}}{B}$ is *computable$_E$*.

Similar to computability based on introduction rules, this notion again has the feature that normal derivations – here even normalizable ones – are considered computable$_E$ without further justification, which is natural for proving normalization, but cannot be used for a semantic foundation.

As a characteristic feature of the defintions of validity$_E$ and computability$_E$, it might be noted that the notion of reduction does not come in until the atomic stage is reached (in the definition of computability$_E$ in the form of a derivation being strongly normalizable).

In the terminology of terms, one might say that everything is played down to the atomic level by means of term application, whereas the I-rule conceptions were based on what corresponds to term substitution.

The approach sketched here is not the only possible and perhaps not even the most genuine way of putting elimination rules first. If one really tried to dualize the I-rule approach by putting "deriving from" rather than "deriving of" in front, one should develop ideas such as the following: A *closed derivation from A* should be a derivation of absurdity from A, and a derivation $\begin{matrix} A \\ \mathcal{D} \\ B \end{matrix}$ should be justified, if, for every closed valid derivation $\begin{matrix} B \\ \mathcal{D}' \end{matrix}$ from B,

$\begin{matrix} A \\ \mathcal{D} \\ B \\ \mathcal{D}' \end{matrix}$ is a closed valid derivation from A, etc. This, however, would

be in conflict with the asymmetry of derivations, which usually have exactly one end formula, but possibly more than one open assumption. So full dualization would perhaps lead to some variant of a single-premiss/multiple-conclusion logic. A genuine E-rule approach might be desirable if one wanted to logically elaborate ideas like Popper's falsificationism by establishing refutation as the basis of reasoning.[25]

5. DERIVATION STRUCTURES, JUSTIFICATIONS AND ARGUMENTS

The soundness theorems for derivations in $\mathcal{L}$ are interesting metalogical facts. However, of a semantic notion of validity we expect more than that. Validity should be a *distinguishing* feature, telling that some derivations are valid while others are not. This is quite analogous to the notion of truth which states that some propositions are true, whereas others are not true. A result showing that every proposition is true, making truth a general feature of propositions, would be considered inadequate. Similarly, there should be a more general notion of derivation within which the notion of validity determines a *subclass*. It is easy to construct such derivations by simply combining arbitrary rules, not only the rules which belong to $\mathcal{L}$. For example, a single-step derivation in $\mathcal{L}$ of the form

$$\frac{A \to B}{B}$$

should turn out not to be valid, because for certain $S \geq S_0$, not every closed S-valid derivation of $A \to B$ becomes a closed S-valid derivation of B when B is appended at the end of this derivation.[26] This means that we must be able to talk about arbitrary derivations which are not built according to a previously given set of rules. This is important particularly if one would like to pose the question of *completeness*, i.e., the question of whether every valid derivation can be represented in $\mathcal{L}$. As long as [S-]validity is only defined for $\mathcal{L}$ or $\mathcal{L}(S)$, completeness is absolutely trivial. It simply says that every [S-]valid derivation is a derivation, as there are no candidates for derivations which are not in $\mathcal{L}$ or $\mathcal{L}(S)$. This problem does not arise when we are dealing with computability and normalizability only. Computability, as an auxiliary concept to prove normalization, is not necessarily a concept which aims at classifying derivations as computable and non-computable, at least not primarily. In the context of computability, we would simply like to show that *all* derivations exhibit the property of being normalizable.

Since by "derivations" one normally understands derivations in a given system, one should choose a different term for candidates of derivations. I propose talking of *derivation structures*.[27] Hence, the purpose of this section is to define a notion of a derivation structure and of the [S-]validity of derivation structures in such a way that derivations in $\mathcal{L}$ or $\mathcal{L}(S)$ become special derivation structures generated by particular rules of inference. Such a definition will also require generalizing the notion of *reducing* a derivation, which in the standard case is only defined for elimination inferences (in the implicational fragment only *modus ponens*), provided its major premiss results from applying an introduction rule. In principle, reductions should be definable for derivation structures ending with any non-introduction inference.

In order to develop a notion of derivation in a generalized sense, we make use of concepts from the theory of natural deduction and extend them to arbitrary formula trees. A derivation structure over the language of implicational logic (and possibly over atomic systems S as well) can be defined as follows: A *derivation structure* is a *formula tree* together with a *discharge function*. A discharge function for a formula tree is a function which associates with every top formula[28] a formula occurring below (on the same branch in the

tree).[29] The intended reading is the following. Suppose $A_1, \ldots, A_n$ and B occur in the tree as follows:

$$
\begin{array}{ccc}
\diagdown \; \diagup & & \diagdown \; \diagup \\
A_1 & & A_n \\
\diagdown & & \diagup \\
 & B & \\
 & \vdots &
\end{array}
$$
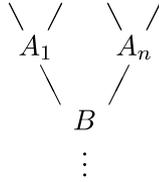
where each A_i is the value of the discharge function f for top-formulas $C_{i1}, \ldots, C_{im_i}$, i.e. $f(C_{i1}) = \ldots = f(C_{im_i}) = A_i$. Then B is inferred as a conclusion from the premisses $A_1, \ldots, A_n$, where at this application, for each i ($1 \leq i \leq n$), the assumptions $C_{i1}, \ldots, C_{im_i}$ in the derivation of each A_i are discharged. This means that the step depicted can be viewed as governed by the following inference rule:

$$
\cfrac{[C_{11}, \ldots, C_{1m_1}] \qquad [C_{n1}, \ldots, C_{nm_n}]}{\cfrac{A_1 \qquad \ldots \qquad A_n}{B}} R.
$$

Conversely, an inference rule of the form R can be used to create a step in a derivation structure, where the C_{ij} above the A_i describe the appropriate discharge function $f(C_{i1}) = \ldots = f(C_{im_i}) = A_i$, where if all C_{ij} are missing there is no discharge function with value A_i, and where in the absence of all A_i we are left with B as an axiom. In this way, inference rules can be *extracted* from a derivation structure, and we can check if a given set of inference rules allows us to *generate* this derivation structure. This means that every occurrence of a formula in a derivation structure uniquely *determines* a rule leading to it; in particular, it uniquely determines the rule applied in the last step. This rule is the most specific rule which just describes the derivation step in question. Borrowing a term from the philosophy of science, it may be called the *minimal covering rule* of this derivation step. We may then define a generality order on rules, according to which rules which are more general than the minimal covering rule allow one to gener-ate this derivation step as well. For example, the rule $\cfrac{\overset{[C]}{A}}{B}$ may be considered more general than the rule $\cfrac{A}{B}$, as it not only allows

one to pass from A to B, but also to discharge the assumption C at the same time. So if $\dfrac{A}{B}$ is the minimal covering rule of a step in a derivation structure, this step may also be viewed as resulting from the application of $\dfrac{\overset{[C]}{A}}{B}$. The possible generality orders depend on various parameters. In the given example, it is essential that "vacuous discharging" of assumptions be permitted. This implies in particular that issues of substructural logics may come into play. I cannot discuss these points here. In what follows I shall apply the usual structural conventions common in natural deduction, such as viewing sequences of assumptions $[C_{i1}, \ldots, C_{im_i}]$ as sets, permitting vacuous discharging of assumptions etc.[30]

It should be emphasized that rules are understood as "concrete rules" rather than rule schemata. Others (such as Prawitz 1973, p. 231) would speak of *instances* of rules instead. Thus, when talking of *modus ponens*

$$\frac{A \to B \quad A}{B}$$

in a general fashion, I would refer to this as a *rule schema*, whereas, if *modus ponens* for particular formulas A and B is meant, I speak of a *rule*. There are various options for capturing the notion of a rule in relation to that of a derivation structure. In the terminology used here, a rule of the form R might be applied in the last step of different derivation structures. Such a derivation structure may be written as

$$\frac{\begin{matrix} [C_{11}, \ldots, C_{1m_1}] & & [C_{n1}, \ldots, C_{nm_n}] \\ \mathcal{D}_1 & & \mathcal{D}_n \\ A_1 & \ldots & A_n \end{matrix}}{B}$$

(for concrete $\mathcal{D}_1, \ldots, \mathcal{D}_n$) and viewed as an *application* of R. This means that even a (concrete, non-schematic) rule is uniform in the sense that all applications follow the same pattern. A different view would be to define a rule simply as a set of such patterns, meaning that the applications of a rule are in no way structurally related.[31]

According to our definition,

$$\frac{A \rightarrow B}{B}$$

is a (very simple) derivation structure. Therefore, once we have defined *S*-validity and validity for derivation structures, we are in the position to state that this derivation structure is not valid. Now why should

$$\frac{A \rightarrow B}{B}$$

be invalid, whereas a corresponding instance

$$\frac{A \rightarrow B \quad A}{B}$$

of *modus ponens* is obviously valid? Both rules share the feature that they are not (self-justifying) introduction rules. However, for *modus ponens* a reduction procedure is available which helps generate a valid derivation when valid derivations of the premisses are given, whereas for

$$\frac{A \rightarrow B}{B}$$

no such procedure is at hand. So non-validity is due to the lack of appropriate reductions. On the other hand, the one-step derivation structure

$$\frac{A \rightarrow (B \rightarrow C)}{B \rightarrow (A \rightarrow C)} \tag{$\star$}$$

should be counted as valid, even if it is not a derivation in $\mathcal{L}$ and no standard reduction applies. Here, to ensure validity, we must add a new reduction, which is different from the standard reduction. For example, if we use the reduction step

$$\frac{\begin{array}{c}\mathcal{D}\\ A \rightarrow (B \rightarrow C)\end{array}}{B \rightarrow (A \rightarrow C)} \quad \text{reduces to} \quad \cfrac{\cfrac{\cfrac{\cfrac{\overset{\mathcal{D}}{A \rightarrow (B \rightarrow C)} \quad [A]^{(2)}}{B \rightarrow C} \quad [B]^{(1)}}{C}}{A \rightarrow C}\,^{(1)}}{B \rightarrow (A \rightarrow C)}\,^{(2)}, \tag{$\star\star$}$$

then the derivation ($\star$) would indeed turn out as valid according to our definition of validity.[32]

So what needs to be changed is not so much the notion of validity itself, but the notion of reduction the definition of validity refers to. This fits in very well with the general idea of reduction. Reductions serve as justifying procedures for non-canonical steps, i.e., for steps, which are not self-justifying. When we consider validity for arbitrary derivation structures, we should not only consider the topological structure of derivations, but also generalize their reductions. This means that a more appropriate concept would be that of a derivation structure *combined* with a set of permitted reductions, which need not coincide with the set of standard reductions used in the normalization of derivations in $\mathcal{L}$.

This is exactly the step taken by Prawitz (1973) and in his later publications (see Prawitz 2006). I present it in modified form, where the modifications do not only affect terminology. By an *argument* I understand a pair $\langle \mathcal{D}, \mathcal{J} \rangle$, where $\mathcal{D}$ is a derivation structure and $\mathcal{J}$ is a *justification* consisting of a set of reductions. This conforms with our previous talking of derivations in a particular system like $\mathcal{L}$, as in such a system certain standard reductions are available.

A *reduction* is a pair

$$\mathcal{D}_1 \rhd \mathcal{D}_2,$$

also read as

$$\mathcal{D}_1 \text{ reduces to } \mathcal{D}_2,$$

which associates a derivation structure $\mathcal{D}_2$ with the derivation structure $\mathcal{D}_1$, such that $\mathcal{D}_2$ has the same end formula as $\mathcal{D}_1$ and no open assumptions beyond those in $\mathcal{D}_1$ (but possibly less assumptions). We say that this reduction is *assigned* to $\mathcal{D}_1$, or that it is a reduction *for* $\mathcal{D}_1$. If it belongs to a justification $\mathcal{J}$, we say that it is *assigned* to $\mathcal{D}_1$ *via* $\mathcal{J}$. If $\mathcal{D}_1$ is an application of a rule R, we call $\mathcal{D}_1 \rhd \mathcal{D}_2$ a reduction *for* R (remember that an application of a rule R is a derivation applying this rule in its last step). By $\mathcal{J}(R)$ or $\mathcal{J}_R$ we denote the subset of $\mathcal{J}$ containing all reductions for R via $\mathcal{J}$. Conversely, if j_R is a set of reductions for R, then we may compose some $\mathcal{J}$ as the union of all j_R for a given set of rules R. It is not expected that j_R comprise reductions for *all* potential applications of R; as a limiting case, j_R may even be empty. Furthermore, it is not excluded that there might be more than one reduction for the same derivation structure. This corresponds to the idea that there might be "alternative justifications" for the same derivation structure. Reductions for

introduction rules are also not excluded in principle, although they are of no real use as introduction rules are self-justifying without any need for reduction. The only constraint we have to impose on a justification $\mathcal{J}$ is that it be closed under substitution in the following sense.

Closure under substitution:

If the reduction

$$
\begin{array}{ccc}
A_1 \dots A_n & & A_1 \dots A_n \\
\mathcal{D} & \rhd & \mathcal{D}'
\end{array}
$$

is in $\mathcal{J}$, then for any $\dfrac{\mathcal{D}_1}{A_1}, \dots, \dfrac{\mathcal{D}_n}{A_n}$,

$$
\begin{array}{ccc}
\mathcal{D}_1 \quad \mathcal{D}_n & & \mathcal{D}_1 \quad \mathcal{D}_n \\
A_1 \dots A_n & \rhd & A_1 \dots A_n \\
\mathcal{D} & & \mathcal{D}'
\end{array}
$$

is in $\mathcal{J}$ as well.

So a justification $\mathcal{J}$ in our sense is nothing but a proof reduction system, for which closure under substitution holds.[33] If $\mathcal{J}$ is a justification, then $\mathcal{J}'$ is called an extension of $\mathcal{J}$ ($\mathcal{J}' \geq \mathcal{J}$) if $\mathcal{J}'$ results from $\mathcal{J}$ by adding reductions such that closure under substitution continues to hold. In other words, an extension of $\mathcal{J}$ is any superset of $\mathcal{J}$ which is itself a justification.

This definition differs considerably from Prawitz's, as he uses a so-called "consistency" requirement for justifications which restricts the formation of extensions, and perhaps even disallows alternative reductions for the same derivation structure. As I see it, this consistency requirement plays a role only if strong normalization is the aim, which is not in the centre of interest when the semantic concept of validity is defined. Adding reductions for the same derivation structure may only be detrimental to *strong* validity, as this introduces new reduction sequences not previously considered.

Normally, the set j_R of reductions for R will be given schematically, which means that it does not depend on the particular application of R. When R is an instance of a uniform rule schema, the reductions for R are often given schematically in the more general sense that they are independent of the particular formula which

occurs in R, as in the case of the standard reduction for implication

$$
mp: \quad \begin{array}{c} A \\ \mathcal{D} \\ \dfrac{B}{\underline{A \to B} \quad A} \\ B \end{array} \quad \mathcal{D}' \quad \triangleright \quad \begin{array}{c} \mathcal{D}' \\ A \\ \mathcal{D} \\ B \end{array} \;.
$$

(If we wanted to specialize this to particular formulas A and B, we would write $mp_{A \to B}$.) However, this is not mandatory, and it is not excluded that different applications of the same rule or applications of different rules which are instances of the same rule schema receive entirely different reductions. Only with respect to the substitution of derivation structures for open assumptions are reductions schematic, as required by closure under substitution.

A derivation structure $\mathcal{D}$ *reduces in one step* to a derivation structure $\mathcal{D}'$ ($\mathcal{D} \succ_1 \mathcal{D}'$), if $\mathcal{D}'$ results from $\mathcal{D}$ by finitely often applying a reduction to a substructure of $\mathcal{D}$. Here, a substructure of $\mathcal{D}$ is a subtree of $\mathcal{D}$ with the discharge function f restricted to assumptions whose values under f occur above the end formula of $\mathcal{D}$. A derivation structure $\mathcal{D}$ *reduces to* a derivation structure $\mathcal{D}'$ ($\mathcal{D} \succeq \mathcal{D}'$), if $\mathcal{D}'$ is identical with $\mathcal{D}$ or results from $\mathcal{D}$ by a finite number of one-step reductions. It is important to note that reductions apply to derivation structures, given a justification $\mathcal{J}$. So we could more explicitly write $\mathcal{D} \succeq_{\mathcal{J}} \mathcal{D}'$ rather than $\mathcal{D} \succeq \mathcal{D}'$. Reductions cannot change or generate justifications, which means that a notion such as $\langle \mathcal{D}, \mathcal{J} \rangle \succeq \langle \mathcal{D}', \mathcal{J}' \rangle$ is not defined.

Let $\mathcal{L}^*(S)$ be the *logic of arguments* over S, which may be identified with the set of arguments $\langle \mathcal{D}, \mathcal{J} \rangle$, where the derivation structure $\mathcal{D}$ is built up from implicational formulas over formulas of S as atoms, and $\mathcal{J}$ is a justification whose reductions are defined for such derivation structures. As a limiting case, we again have $\mathcal{L}^*(S_0)$, in short $\mathcal{L}^*$, which uses only propositional variables as atoms. Standard implicational logic $\mathcal{L}$ would then be obtained by considering the set of all $\langle \mathcal{D}, \mathcal{J} \rangle$ such that $\mathcal{D}$ is a derivation in standard implicational logic, whereas $\mathcal{J}$ is fixed for all derivations and comprises exactly the standard reductions.

Then the *S-validity* of arguments $\langle \mathcal{D}, \mathcal{J} \rangle$, which is the same as the *S-validity* of derivation structures $\mathcal{D}$ with respect to justifications $\mathcal{J}$, is defined as follows:

DEFINITION OF S-VALIDITY FOR ARGUMENTS

(I) Every closed derivation in S is *S-valid* with respect to $\mathcal{J}$ (for every $\mathcal{J}$).

(II) A closed derivation structure
$$\begin{array}{c} A \\ \mathcal{D} \\ B \\ \hline A \to B \end{array}$$
is *S-valid* with respect to $\mathcal{J}$, if its immediate substructure
$$\begin{array}{c} A \\ \mathcal{D} \\ B \end{array}$$
is *S-valid* with respect to $\mathcal{J}$.

(III) A closed non-canonical derivation structure is *S-valid* with respect to $\mathcal{J}$, if it reduces, with respect to $\mathcal{J}$, to a canonical derivation structure, which is *S-valid* with respect to $\mathcal{J}$.

(IV) An open derivation structure
$$\begin{array}{c} A_1 \dots A_n \\ \mathcal{D} \\ B \end{array}$$
, where all open assumptions of $\mathcal{D}$ are among $A_1, \dots, A_n$, is *S-valid* with respect to $\mathcal{J}$, if for every $S' \geq S$ and $\mathcal{J}' \geq \mathcal{J}$, and for every list of closed derivation structures
$$\begin{array}{c} \mathcal{D}_i \\ A_i \end{array}$$
$(1 \leq i \leq n)$, which are *S'-valid* with respect to $\mathcal{J}'$,
$$\begin{array}{c} \mathcal{D}_1 \quad \mathcal{D}_n \\ A_1 \dots A_n \\ \mathcal{D} \\ B \end{array}$$
is *S'-valid* with respect to $\mathcal{J}'$.[34]

In clause (IV), the reason for considering extensions $\mathcal{J}' \geq \mathcal{J}$ of justifications, in addition to extensions $S' \geq S$ of atomic systems, is again, in my view, a monotonicity constraint. It is obvious that the following holds:

MONOTONICITY OF S-VALIDITY (FOR ARGUMENTS)

An argument $\langle \mathcal{D}, \mathcal{J} \rangle$ in $\mathcal{L}^*(S)$ is S-valid iff for every $S' \geq S$ and $\mathcal{J}' \geq \mathcal{J}$, $\langle \mathcal{D}, \mathcal{J}' \rangle$ is S'-valid.

The corresponding universal concept is then defined as follows: If v is an assignment of S-formulas to propositional variables, then for a $\mathcal{J}$ comprising reductions for arguments in $\mathcal{L}^*$, $\mathcal{J}^v$ is defined as the set of reductions which acts on derivations $\mathcal{D}^v$ in the same way as $\mathcal{J}$ acts on $\mathcal{D}$ (i.e., $\mathcal{J}^v$ is the homomorphic image of $\mathcal{J}$ under v). Then an argument $\langle \mathcal{D}, \mathcal{J} \rangle$ in $\mathcal{L}^*$ is defined *universally valid* iff for every S and every v, $\langle \mathcal{D}^v, \mathcal{J}^v \rangle$ is S-valid. Again we can prove:

PROPOSITION   Let $\langle \mathcal{D}, \mathcal{J} \rangle$ be in $\mathcal{L}^*$. Then $\langle \mathcal{D}, \mathcal{J} \rangle$ is universally valid iff $\langle \mathcal{D}, \mathcal{J} \rangle$ is S_0-valid.

This means that we can continue to use the term "valid" (now with respect to some $\mathcal{J}$) interchangeably for both universal and S_0-validity.

It is obvious how notions of strict S-validity for $\mathcal{L}^*(\mathcal{S})$ and of strict validity for $\mathcal{L}^*$ can be defined. We can also prove (weak) normalization from strict validity. However, as mentioned above, for strong validity, problems arise with our unrestricted notions of justifications $\mathcal{J}$ and extensions $\mathcal{J}' \geq \mathcal{J}$.

Suppose $\mathcal{J}$ is the set of standard reductions. Then it is obvious that all derivations in $\mathcal{L}(S)$ are valid arguments with respect to $\mathcal{J}$, and all derivations in $\mathcal{L}$ are universally valid with respect to $\mathcal{J}$. So the "new" concept of validity is a generalization of the "old" concept, which yields the same results for derivations in standard implicational logic.

The basic difference between derivations in the old sense and arguments is, of course, that soundness no longer holds in every case; it simply depends on the justifications provided, as was intended by the introduction of the general notion of an argument.

Returning to our previous example, we can now specify what is meant by the validity of the one-step derivation

$$\frac{A \rightarrow (B \rightarrow C)}{B \rightarrow (A \rightarrow C)}. \qquad\qquad (\star)$$

This derivation is obviously valid with respect to the standard reductions of implicational logic extended with the reduction given by $(\star\star)$.[35] We may ask whether completeness of intuitionistic logic, or at least of minimal or positive implicational logic holds in the sense that for any derivation structure, which *can be justified* as valid (i.e., which is valid with respect to *some* justification), a derivation in $\mathcal{L}$ of its end formula from its open assumption formulas can be found. That this is indeed the case was posed by Prawitz as a conjecture (1973, p. 246), without his being able to indicate so far how it might be proved.

In addition to *validity* in the sense sketched here, Prawitz also defines a notion of *computability* for arguments, which he (unfortunately) calls *strong validity*. It is not surprising that he is able to establish strong normalization of minimal logic with respect to the general context of arbitrary justifications, given a notion of "consistent" extensions of justifications. I cannot present this here. From

my point of view, his general computability concept suffers from the same defect as did the less general computability concept dealt with in Sections 3 and 4. Again, Prawitz has to consider irreducible non-canonical arguments as strongly valid, the only difference being that irreducibility is now taken with respect to a justification $\mathcal{J}$, which is not confined to the standard reductions (1973, p. 239; 1974, p. 74).

6. THE RELATIONSHIP BETWEEN COMPUTABILITY, VALIDITY AND NORMALIZABILITY: COUNTEREXAMPLES

In Section 3, we claimed that computability and validity are crucially different, particularly by arguing that normal derivations need to be justified semantically. However, at that stage we were not able to give counterexamples establishing this difference, as extensionally the concepts were identical, comprising all derivations in $\mathcal{L}$. Now with respect to the generalized concept of an argument, we can provide counterexamples.

We understand the computability of an argument $\langle \mathcal{D}, \mathcal{J} \rangle$, i.e. the computability of $\mathcal{D}$ with respect to a justification $\mathcal{J}$, in the following sense, which leads to weak normalization, and compare it with the validity of $\mathcal{D}$ with respect to $\mathcal{J}$.

DEFINITION OF (WEAK) COMPUTABILITY OF ARGUMENTS

(i) A derivation structure of the form $\begin{array}{c} [A] \\ \mathcal{D} \\ B \\ \hline A \to B \end{array}$ is *computable* with

respect to $\mathcal{J}$, if for every $\mathcal{J}' \geq \mathcal{J}$ and every $\begin{array}{c} \mathcal{D}' \\ A \end{array}$ *computable* with

respect to $\mathcal{J}'$, $\begin{array}{c} \mathcal{D}' \\ A \\ \mathcal{D} \\ B \end{array}$ is *computable* with respect to $\mathcal{J}'$.

(ii) If a derivation structure $\mathcal{D}$ is not in I-form and is normal (= irreducible) with respect to $\mathcal{J}$, then it is *computable* with respect to $\mathcal{J}$.

(iii) If a derivation is not in I-form and is not normal with respect to $\mathcal{J}$, then $\mathcal{D}$ is *computable* with respect to $\mathcal{J}$, if $\mathcal{D}$ reduces with respect to $\mathcal{J}$ to a $\mathcal{D}'$ which is *computable*.

Counterexample 1: Computability of $\langle \mathcal{D}, \mathcal{J} \rangle$ does not imply validity of $\langle \mathcal{D}, \mathcal{J} \rangle$

We construct an argument $\langle \mathcal{D}, \mathcal{J} \rangle$ in such a way that $\langle \mathcal{D}, \mathcal{J} \rangle$ is closed, non-canonical and normal, and therefore computable, but not valid. Choose a closed non-canonical derivation structure $\mathcal{D}$ ending with a non-atomic formula, e.g.,

$$\frac{\dfrac{\overset{(1)}{[B \to C]}}{(B \to C) \to (B \to C)}}{B \to C} \, (\to \mathrm{I})(1).$$

Choose $\mathcal{J}$ in such a way that with $\mathcal{D}$ no reduction is associated (e.g., take $\mathcal{J}$ to be empty). Then $\langle \mathcal{D}, \mathcal{J} \rangle$ is computable, because it is irreducible. However, $\langle \mathcal{D}, \mathcal{J} \rangle$ is not valid, because it cannot, as required for validity, be reduced to a canonical derivation structure, since no reduction for $\mathcal{D}$ is available in $\mathcal{J}$.

Counterexample 2: Validity of $\langle \mathcal{D}, \mathcal{J} \rangle$ does not imply computability of $\langle \mathcal{D}, \mathcal{J} \rangle$

We consider $\langle \bot, \to \rangle$-logic, i.e., a system with a logical constant $\bot$, for which there is no introduction rule. In such a system, the derivation $\dfrac{\bot}{A}$, and therefore $\dfrac{\dfrac{\overset{(1)}{[\bot]}}{A}}{\bot \to A}(1)$ is valid with respect to any $\mathcal{J}$. Now for some B, let $\mathcal{J}$ be chosen in such a way that $\dfrac{B}{\bot}$ is irreducible. Let $\mathcal{J}$ furthermore be chosen such that $\dfrac{\dfrac{B}{\bot}}{A}$ reduces to itself with respect to $\mathcal{J}$, i.e., the reduction for $\dfrac{\dfrac{B}{\bot}}{A}$ is non-terminating.

Then $\dfrac{\dfrac{\overset{(1)}{[\bot]}}{A}}{\bot \to A}(1)$ is not computable with respect to $\mathcal{J}$, because for computable $\dfrac{B}{\bot}$, $\dfrac{\dfrac{B}{\bot}}{A}$ is not computable (with respect to $\mathcal{J}$).[36]

It can easily be seen that these counterexamples also hold for strict validity instead of validity. Furthermore, they hold for strong validity, when computability is defined in the strong sense (demanding in clause (iii) that all one-step reductions lead to computable derivations).

It might be added that Counterexample 1 is at the same time a counterexample showing that normalizability does not imply validity.[37] Similarly, Counterexample 2 shows that normalizability does not imply computability. The latter is not surprising, as computability is a stronger concept than normalizability, using infinite branching when quantifying over substitution instances of open derivation structures.

7. LOGICAL CONSEQUENCE AND THE VALIDITY OF INFERENCE RULES

It is natural that the *S*-validity of an inference rule

$$\frac{A_1 \quad \ldots \quad A_n}{A}$$

with respect to a justification $\mathcal{J}$, should mean that the one-step derivation structure of the same form is *S*-valid with respect to $\mathcal{J}$. We can even define the *S*-validity of an inference rule which allows the discharging of assumptions, such as the generalized rule

$$\frac{[C_{11}, \ldots, C_{1m_1}] \qquad [C_{n1}, \ldots, C_{nm_n}]}{B}.$$

This rule is called *S-valid* with respect to $\mathcal{J}$, if for all $S \geq S_0$, all $\mathcal{J}' \geq \mathcal{J}$, and every list of derivation structures $\begin{matrix} C_{i1}, \ldots, C_{im_i} \\ \mathcal{D}_i \\ A_i \end{matrix}$ $(1 \leq i \leq n)$, which are *S-valid* with respect to $\mathcal{J}'$, the derivation structure $\frac{\mathcal{D}_1 \quad \ldots \quad \mathcal{D}_n}{B}$ is *S-valid* with respect to $\mathcal{J}'$.

This gives rise to a corresponding notion of consequence.[38] Instead of saying that the rule

$$\frac{A_1 \quad \ldots \quad A_n}{A}$$

is *S*-valid with respect to $\mathcal{J}$, we may say that *A* is a *consequence of $A_1, \ldots, A_n$ with respect to S and $\mathcal{J}$ ($A_1, \ldots, A_n \models_{S,\mathcal{J}} A$)*; if we

consider universal validity with respect to $\mathcal{J}$, we say speak of *consequence with respect to $\mathcal{J}$* ($A_1, \ldots, A_n \models_{\mathcal{J}} A$); and finally, if there is some $\mathcal{J}$ such that universal validity holds for $\mathcal{J}$, then we may speak of *logical consequence* ($A_1, \ldots, A_n \models A$). Corresponding to the case of rules discharging assumptions, we obtain a notion of consequence

$$\Gamma_1 \Rightarrow A_1, \ldots, \Gamma_n \Rightarrow A_n \models_{S, \mathcal{J}} A$$

for sets of formulas Γ_i. This is to express that the rule

$$\frac{\begin{matrix} [\Gamma_1] & & [\Gamma_n] \\ A_1 & \ldots & A_n \end{matrix}}{A}$$

is *S*-valid with respect to $\mathcal{J}$, i.e., we have some notion of implication in the antecedent of $\models$, which is independent of whether the logical constant of implication is available in our language.

This goes crucially beyond any classical notion of consequence. In proof-theoretic semantics, we use (mostly implicitly) some *structural* notion of implication throughout, which is due to the fact that rules can discharge assumptions. As a structural concept it is comparable to the comma as a structural conjunction. This structural notion of implication ("$\Rightarrow$" in my terminology) has been used in generalized concepts of inference rules. It is also important for the formulation of a basic sequent calculus in theories of definitional reflection (see Hallnäs 1991, 2006, Schroeder-Heister 1991b, 1993).

It should be emphasized that it is extremely misleading to write a valid rule or consequence as

$$\frac{A_1 \quad \ldots \quad A_n}{A} j$$

with j being understood as the justification of the step from $A_1, \ldots, A_n$ to A. In simple (or "direct") cases like *modus ponens*

$$\frac{A \rightarrow B \quad A}{B} mp$$

the reduction *mp* (which is actually a reduction schema) is indeed a justification of this single step. However, in a case like

$$\frac{A \rightarrow (B \rightarrow C)}{B \rightarrow (A \rightarrow C)} j$$

with j being the reduction given by $(\star\star)$, j alone does not suffice to justify this step, as the result of using j, given a valid derivation of the premiss $A \to (B \to C)$, uses *modus ponens*. So the result of applying j is valid only if the *modus ponens* reduction mp is available. This again means that the step

$$\frac{A \to (B \to C)}{B \to (A \to C)}$$

is justified only with respect to some $\mathcal{J}$, where $\mathcal{J}$ comprises both j and the *modus ponens* reduction mp. Thus

$$\frac{A \to (B \to C)}{B \to (A \to C)} \{j, mp\}$$

or

$$A \to (B \to C) \models_{\{j, mp\}} B \to (A \to C)$$

would be an appropriate notation. What is involved in the justification of single inference steps is often a whole reduction system, not a single justifying reduction.

This makes *proof-theoretic* consequence differ from *constructive* consequence according to which

$$\frac{A_1 \quad \ldots \quad A_n}{A}$$

might be defined as valid with respect to a *constructive function* f, if f transforms valid arguments of the premisses $A_1, \ldots, A_n$ into a valid argument of the conclusion A. Actually, it is not always possible to extract such a constructive function from our proof reduction system, as a reduction system $\mathcal{J}$ serving as a justification need not be deterministic, which means that it merely generates a constructive relation on arguments. In any case, the notion of a justification as a proof reduction system presents an intensional analysis of the transformation of arguments which is more fine-grained and more specific than approaches based on the abstract notion of a constructive function.

NOTES

[1] See Muskens et al. (1997), Blamey (2002) and the references therein.

[2] First in print in Schroeder-Heister (1991c).

[3] See especially Brandom (2000), where the relationship to Dummett's and Gentzen's approaches is expressed very clearly.

[4] Actually, the term "inversion principle" was coined by Lorenzen.

[5] For similar reasons I do not deal with projects like that of Tennant, who combines an anti-realist meaning theory with an alternative approach to relevant logic (see Tennant 1987, 1997).

[6] In his later writings, in which he focuses on semantic aspects, Prawitz does not explicitly return to the relationship with normalization.

[7] See Hallnäs (1991, 2006), Hallnäs and Schroeder-Heister (1990, 1991), Schroeder-Heister (1991a, 1992, 1993, 1994b).

[8] Tait (2006) presents some ideas on how to deal with classical logic in proof-theoretic semantics.

[9] In addition to Prawitz (1965), the monographs by Tennant (1978), Troelstra and Schwichtenberg (1996) and Negri and von Plato (2001b) can be recommended as introductory references.

[10] See especially Dummett (1991).

[11] Named following Lorenzen (1955).

[12] Quotes by Gentzen.

[13] See the recent paper by Joachimski and Matthes (2003), which contains many references to the literature.

[14] More precisely: induction given by the operator associating with a set of derivations X of a formula the set of those derivations which reduce in one step to a derivation in X.

[15] See Prawitz (1971, p. 289); Prawitz (1973, p. 238).

[16] In other renderings of computability, all normal derivations are computable by definition, not only those which are not in I-form. For the definition of computability chosen here, this follows as a lemma.

[17] It is bound to fail due to the impredicative character of the substitution lemma, when it is turned into a definition. "Impredicative" here means that computability is defined by quantifying over all substitution instances obtained by substituting arbitrary computable derivations.

[18] In the context of natural deduction derivations it is called the "fundamental assumption", see Dummett (1991, p. 254).

[19] I suppose that Prawitz had something similar in mind (see Prawitz 1971, p. 276). In later papers he ceases to consider extensions $S' \geq S$, considering only extensions of justifying procedures (see Section 5).

[20] This does not mean that the S-validity of closed and of open derivations is defined separately. These two cases occur intertwined in the same derivation. This is due to the fact that the immediate subderivation of a closed canonical derivation of $A \rightarrow B$ is a derivation of B *from* the assumption A.

[21] This is not exactly the converse "*valid implies normal*", which is, of course, wrong.

[22] Again some emphasis has to be placed on "necessarily", as in the case of intuitionistic logic, all derivations are strictly S-*valid*, i.e., strict S-validity and S-validity coincide in this case.

[23] Prawitz (1971, p. 289f. [= appendix A.2]).

[24] However, in the case of strict S-validity$_E$ (but not in the case of strong S-validity$_E$), we would have to distinguish between reducible and irreducible derivations not only in the open case, but also in the closed case, i.e., clause (II) should only be applicable if $\begin{array}{c} \mathcal{D} \\ A \to B \end{array}$ has been reduced as far as possible, meaning that it is irreducible. Otherwise, we cannot prove that $\begin{array}{c} \mathcal{D} \\ A \to B \end{array}$ is weakly normalizable given that $\dfrac{\begin{array}{cc} \mathcal{D} & \mathcal{D}' \\ A \to B & A \end{array}}{B}$ is weakly normalizable. (In the case of strong normalizability this is trivial.)

[25] Dummett (1991, Ch. 13, pp. 283–286) attempts to develop some kind of a "genuine" E-rule approach (within the standard setting of derivations with multiple premises and single conclusions).

[26] For example, if A and B are propositional variables, we may choose S as having no axiom and $A \Rightarrow B$ as the only inference rule. Then there is an S-valid derivation of $A \to B$, but no S-valid derivation of B.

[27] Prawitz (1973) speaks of "argument schemata" (with arguments being closed argument schemata), Prawitz (2006) of "argument skeletons" (with arguments being argument skeletons together with justifications).

[28] More precisely, we should talk of top formula *occurrences*. I do not always terminologically distinguish between formulas and their occurrences. It will always be clear from the context what is meant.

[29] The use of discharge functions was introduced by Prawitz (1965, pp. 20–31). Here it is used in the generalized form as proposed in Schroeder-Heister (1984a).

[30] A corresponding notion of *rule* and *derivation structure* is spelled out in detail in Schroeder-Heister (1984a,b).

[31] Prawitz (1973, p. 31) follows such a general approach, calling $(\langle \mathcal{D}_1, \ldots, \mathcal{D}_n \rangle, A)$ an inference (= rule instance), whenever $\dfrac{\mathcal{D}_1 \quad \ldots \quad \mathcal{D}_n}{A}$ is a derivation structure.

[32] Step $(\star)$ is reduced not directly, but indirectly by invoking *modus ponens* in the reduction result. See the final remarks in Section 7.

[33] Technically, this proof reduction system can be viewed as a higher-order term rewriting system (it is of higher order due to the assumption structure corresponding to λ-abstraction).

[34] See Prawitz (1973, p. 236; 1974, p. 73; 2006). Prawitz does not consider extensions of atomic systems S.

[35] It is not valid with respect to $(\star\star)$ alone – see the final remarks in Section 7.

[36] The intuitive reason for this behaviour is the following: $\dfrac{\bot}{A}$ is always valid as there is no *closed* valid derivation of $\bot$. However, for open normal derivations $\dfrac{B}{\bot}$, the reduction of $\dfrac{\begin{array}{c} B \\ \bot \end{array}}{A}$ can be made non-terminating by means of an appropriate $\mathcal{J}$. (Note that we do not choose $\dfrac{B}{\bot}$ to be simply $\bot$, because then the example would not work for *strict* validity, as the reduction for $\dfrac{\bot}{A}$ would not terminate.)

[37] Actually, normalizability is implied by computability, but this fact is not used in the counterexample.

[38] See also Prawitz (1985).

REFERENCES

Blamey, S.: 2002, 'Partial Logic', in D. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic*, *2nd edn.*, Volume 5, Kluwer, Dordrecht, pp. 261–353.

Brandom, R.B.: 2000, *Articulating Reasons: An Introduction to Inferentialism*, Harvard University Press, Cambridge Mass.

Dummett, M.: 1975, 'The Philosophical Basis of Intuitionistic Logic', in H. E. Rose and J. C. Shepherdson (eds.), *Logic Colloquium '73*, North Holland, Amsterdam, pp. 5–40, repr. in M. Dummett, *Truth and Other Enigmas*, Duckworth, London 1978, pp. 215–247.

Dummett, M.: 1991, *The Logical Basis of Metaphysics*, Duckworth, London.

Etchemendy, J.: 1990, *The Concept of Logical Consequence*, Harvard University Press, Cambridge Mass.

Gentzen, G.: 1934, 'Untersuchungen über das logische Schließen', *Mathematische Zeitschrift* **39** (1934/35), 176–210, 405–431, English translation ('Investigations into Logical Deduction') in M. E. Szabo (ed), *The Collected Papers of Gerhard Gentzen*, North Holland, Amsterdam 1969, pp. 68–131. Quotations are according to Szabo's translation.

Girard, J.-Y.: 1971, 'Une extension de l'interprétation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types', in J. E. Fenstad (ed.), *Proceedings of the 2nd Scandinavian Logic Symposium (Oslo 1970)*, North Holland, Amsterdam, pp. 63–92.

Hallnäs, L.: 1991, 'Partial Inductive Definitions', *Theoretical Computer Science* **87**, 115–142.

Hallnäs, L.: 2006, 'On the Proof-Theoretic Foundation of General Definition Theory', *Synthese* (this issue).

Hallnäs, L. and P. Schroeder-Heister: 1990, 'A Proof-Theoretic Approach to Logic Programming. I. Clauses as Rules', *Journal of Logic and Computation* **1** (1990/91), 261–283.

Hallnäs, L. and P. Schroeder-Heister: 1991, 'A Proof-Theoretic Approach to Logic Programming. II. Programs as Definitions', *Journal of Logic and Computation* **1** (1990/91), 635–660.

Joachimski, F. and R. Matthes: 2003, 'Short Proofs of Normalization for the Simply-Typed λ-calculus, Permutative Conversions and Gödels T', *Archive for Mathematical Logic* **42**, 59–87.

Kahle, R. and P. Schroeder-Heister: 2006, 'Introduction: Proof-Theoretic Semantics', *Synthese* (this issue).

Lorenzen, P.: 1955, *Einführung in die operative Logik und Mathematik*, Springer, Berlin, 2nd edn., 1969.

Martin-Löf, P.: 1971, 'Hauptsatz for the Intuitionistic Theory of Iterated Inductive Definitions', in J. E. Fenstad (ed.), *Proceedings of the 2nd Scandinavian Logic Symposium (Oslo 1970)*, North Holland, Amsterdam, pp. 179–216.

Martin-Löf, P.: 1995, 'Verificationism Then and Now', in W. DePauli-Schimanovich et al. (eds.), *The Foundational Debate: Complexity and Constructivity in Mathematics and Physics*, Kluwer, Dordrecht, pp. 187–196.

Martin-Löf, P.: 1998, 'Truth and Knowability: On the Principles C and K of Michael Dummett', in H. G. Dales and G. Oliveri (eds.), *Truth in Mathematics*, Clarendon Press, Oxford, pp. 105–114.

Montague, R.: 1970, 'English as a Formal Language', in B. Visentini et al. (eds.), *Linguaggi nella società e nella tecnica*, Milano. Repr. in R. H. Thomason (ed.), *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press, New Haven 1974, pp. 188–221.

Muskens, R. A., J. van Benthem and A. Visser: 1997, 'Dynamics', in J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language*, Elsevier, Amsterdam, pp. 587–648.

Negri, S. and J. von Plato: 2001, *Structural Proof Theory*, Cambridge University Press.

Prawitz, D.: 1965, *Natural Deduction: A Proof-Theoretical Study*, Almqvist & Wiksell, Stockholm.

Prawitz, D.: 1971, 'Ideas and Results in Proof Theory', in J. E. Fenstad (ed.), *Proceedings of the 2nd Scandinavian Logic Symposium (Oslo 1970)*, North Holland, Amsterdam, pp. 235–308.

Prawitz, D.: 1973, 'Towards a Foundation of a General Proof Theory', in P. Suppes et al. (eds.), *Logic, Methodology, and Philosophy of Science IV*, North Holland, Amsterdam, pp. 225–250.

Prawitz, D.: 1974, 'On the Idea of a General Proof Theory', *Synthese* **27**, 63–77.

Prawitz, D.: 1985, 'Remarks on Some Approaches to the Concept of Logical Consequence', *Synthese* **62**, 152–171.

Prawitz, D.: 2006, 'Meaning Approached Via Proofs', *Synthese* (this issue).

Schroeder-Heister, P.: 1984a, 'A Natural Extension of Natural Deduction', *Journal of Symbolic Logic* **49**, 1284–1300.

Schroeder-Heister, P.: 1984b, 'Generalized Rules for Quantifiers and the Completeness of the Intuitionistic Operators &, ∨, ⊃, ⋏, ∀, ∃', in M.M. Richter et al., *Computation and Proof Theory. Proceedings of the Logic Colloquium held in Aachen, July 1983, Part II.* Springer, Berlin, LNM, Vol. 1104, pp. 399–426.

Schroeder-Heister, P.: 1991a, 'Hypothetical Reasoning and Definitional Reflection in Logic Programming', in P. Schroeder-Heister (ed.), *Extensions of Logic Programming. International Workshop, Tübingen, December 1989, Proceedings.* Springer, Berlin, LNCS, Vol. 475, pp. 327–340.

Schroeder-Heister, P.: 1991b, 'Structural Frameworks, Substructural Logics, and the Role of Elimination Inferences', in G. Huet and G. Plotkin (eds.), *Logical Frameworks.* Cambridge University Press, pp. 385–403.

Schroeder-Heister, P.: 1991c, 'Uniform Proof-Theoretic Semantics for Logical Constants. Abstract.' *Journal of Symbolic Logic* **56**, 1142.

Schroeder-Heister, P.: 1992, 'Cut-Elimination in Logics with Definitional Reflection', in D. Pearce and H. Wansing (eds.), *Nonclassical Logics and Information Processing. International Workshop, Berlin 1990, Proceedings.* Springer, Berlin, LNCS, Vol. 619, pp. 146–171.

Schroeder-Heister, P.: 1993, 'Rules of Definitional Reflection', in *8th Annual IEEE Symposium on Logic in Computer Science (Montreal 1993).* IEEE Computer Society Press, Los Alamitos, pp. 222–232.

Schroeder-Heister, P.: 1994b, 'Definitional Reflection and the Completion', in R. Dyckhoff (ed.), *Extensions of Logic Programming. Proceedings of the 4th International Workshop, ELP '93, St. Andrews, March/April 1993*, Springer, Berlin, LNCS, Vol. 798, pp. 333–347.

Tait, W. W.: 1967, 'Intensional Interpretations of Functionals of Finite Type I', *Journal of Symbolic Logic* **32**, 198–212.

Tait, W. W.: 2006, 'Proof-Theoretic Semantics for Classical Mathematics', *Synthese* (this issue).

Tarski, A.: 1933, 'Der Wahrheitsbegriff in den formalisierten Sprachen', *Studia Philosophica* **1** (1935), 261–405 (translated from the Polish original of 1933, with a postscript). Reprinted in K. Berka and L. Kreiser (eds.), *Logik-Texte*, Berlin 1971. English translation of the German version in A. Tarski, *Logic, Semantics, Metamathematics*, Clarendon Press, Oxford, 1956.

Tennant, N. W.: 1978, *Natural Logic,* Edinburgh University Press.

Tennant, N. W.: 1987, *Anti-Realism and Logic,* Clarendon Press, Oxford.

Tennant, N. W.: 1997, *The Taming of the True,* Clarendon Press, Oxford.

Troelstra, A. S. and H. Schwichtenberg: 1996, *Basic Proof Theory.* Cambridge University Press, 2nd edn. 2000.

Wilhelm-Schickard-Institut, Universität Tübingen
Sand 13
72076 Tübingen
Germany
E-mail: psh@informatik.uni-tuebingen.de