HAUPTSATZ FOR THE INTUITIONISTIC THEORY OF ITERATED INDUCTIVE DEFINITIONS

Per MARTIN-LÖF
University of Stockholm

1. Introduction.

1.1. The principle of definition by generalized induction, perhaps best exemplified by the definition of the constructive second number class given by Church and Kleene, and the corresponding principle of proof by generalized induction were first formalized by Kreisel 1963. Also, the idea of iterating generalized inductive definitions, as done by Church and Kleene in their definition of the higher constructive number classes, gives rise to a corresponding principle of proof which was first stated as a formal schema by Kreisel 1964 in his proof of the wellordering of Takeuti's 1957 ordinal diagrams of finite order. A complete formulation of a classical theory of generalized inductive definitions iterated along a primitive recursive wellordering was given by Feferman 1969 whose main object was to establish the relation between his theory and certain subsystems of classical analysis.

1.2. In the present paper I shall give a proof theoretical analysis of the intuitionistic theory of generalized inductive definitions iterated an arbitrary finite number of times. Like the Hilbert type systems of first order predicate logic which were used before Gentzen 1934, the theories of single and iterated generalized inductive definitions formulated by Kreisel and Feferman do not lend themselves immediately to a proof theoretical analysis. My first aim is therefore to reformulate the axioms expressing the principles of definition and proof by generalized induction as rules of inference similar to those introduced by Gentzen 1934 in his system of natural deduction for first order predicate logic. As in Gentzen's case, this reformulation leads to a notable systematization which is interesting already in the case of ordinary inductive definitions, the rules corresponding to the axioms which express the principle of definition by induction appearing as introduction rules for the inductively defined predicates, whereas the axioms which express the principle of proof by induction give rise to the corresponding elimination rules. Moreover, the generalized inductive definitions appear as inductive definitions iterated once and the iterated generalized inductive definitions as inductive definitions iterated twice or more. This explains why I shall omit the attribute generalized in the sequel and talk simply about iterated inductive definitions.

1.3. As soon as the rules for the inductively defined predicates have been separated into introduction and elimination rules, it becomes clear that, in addition to the logical cuts discovered by Gentzen 1934, there arise certain new cuts corresponding to the inductively defined predicates. Also, just as with the logical cuts, there is associated in a natural way with each new form of cut a rule of contraction which shows how to transform the deduction so that the cut becomes eliminated. My main object is to show that, by successive applications of the rules of contraction, every deduction can be reduced to a cut free deduction. This constitutes an extension of Gentzen's 1934 Hauptsatz to the intuitionistic theory of iterated inductive definitions.

1.4. The opinion seems to have been generally accepted that there be no real cut elimination theorem for first order arithmetic and that such a theorem could only be obtained by eliminating the induction schema in favour of the ω-rule. However, when arithmetic is formulated as a theory of ordinary inductive definitions, it becomes possible to formulate and prove a cut elimination theorem which is just as natural and basic as the one for pure first order logic, although, like in second order logic, the subformula principle is necessarily lost. This cut elimination theorem for first order arithmetic is just a special case of the Hauptsatz for the theory of iterated inductive definitions and is obtained by allowing no other predicates in that theory than those defined by ordinary induction.

1.5. The method I shall use in order to prove Hauptsatz for the intuitionistic theory of iterated inductive definitions is an extension of the method that Tait 1967 used in his proof of the normal form theorem for the terms of Gödel's 1958 theory of primitive recursive functionals of finite type. That Tait's method can be carried over from terms denoting functionals of finite type to formal intuitionistic proofs is not astonishing, because Gödel 1958 noted that there is a close connection between the notion of computable functional of finite type and the intuitionistic notion of proof, and Curry and Feys 1958 established an isomorphism between two theories that formalize the very simplest properties of these notions, namely, their basic theory of functionality and the positive implicational calculus, respectively.

1.6. Since Hauptsatz implies consistency, it cannot, according to Gödel's second theorem, be proved by exclusive use of principles which are formalizable in the theory itself. Nevertheless, for every specific deduction in the theory of iterated inductive definitions, the proof that it reduces to a cut free deduction may be formalized in the theory itself. Thus, Hauptsatz becomes provable if the theory is slightly strengthened, for example, by adding the reflection principle

if $F(t)$ is provable for all closed terms t, then $\Lambda x F(x)$.

1.7. A comparison between the method used by Gentzen 1936 in his consistency proof for first order arithmetic and the method I shall use in the present paper may be illuminating. Gentzen's proof can be divided into the following six parts.

1.7.1. Definition of the reduction procedure to be applied to the proof figures.
1.7.2. Definition of an appropriate system of ordinal notations.
1.7.3. Definition of a recursive total ordering between the ordinal notations.
1.7.4. Proof of the wellfoundedness of the order relation. In the case of Takeuti's ordinal diagrams of finite order this proof uses the notion of accessibility which is defined by iterated generalized induction.
1.7.5. Recursive assignment of an ordinal notation to every proof figure.
1.7.6. Proof that a reduction step diminishes the ordinal assigned to a proof figure.

1.8. These six parts of Gentzen's proof have the following counterparts in my proof.

1.8.1. Definition of the rules of contraction.
1.8.2. Disappears, because instead of the ordinal notations I shall use the proof figures themselves.
1.8.3. Definition of a recursive predecessor relation between the proof figures.
1.8.4. Proof of the wellfoundedness of the predecessor relation. This proof uses the notion of computability which is defined by iterated generalized induction.
1.8.5. Disappears.
1.8.6. Disappears, because it is immediately clear that, when the reduction procedure is applied to a proof figure, one obtains a proof figure which precedes the given one.

1.9. I am very grateful to Dag Prawitz who checked in detail an early version of this paper.

2. A canonical form for the iterated inductive definitions.

2.1. The language I shall use is the standard one for first order logic. There may be an arbitrary finite number of function symbols, but, typically, these are just 0 and s, denoting the natural number zero and the successor function, respectively. With each predicate symbol there is associated not only a place index, indicating the number of argument places, but also a nonnegative integer called its *level*. The level of a formula is defined to be the maximum of the levels of the predicate symbols which occur in it.

2.1.1. For the sake of notational simplicity, finite sequences of variables and terms will be denoted by single letters. For example, an atomic formula will be written Pt where P is an $n$ary predicate symbol and t a sequence of n terms.

2.2. An *ordinary production* is a figure of the form

$$\frac{Qq(x) \quad \dots \quad Rr(x)}{Pp(x)}$$

with zero or more atomic formulae $Qq(x)$, ..., $Rr(x)$ as premises of the conclusion $Pp(x)$. I use x to denote the totality of all variables that occur in the production. The level of P must be greater than or equal to the levels of Q, ..., R.

2.3. A *generalized production* is either of the form

$$\frac{H(x) \to Qq(x)}{Pp(x)}$$

called $\to$*production* or of the form

$$\frac{\bigwedge y Qq(x,y)}{Pp(x)}$$

called $\bigwedge$*production*. In the first case the level of P must be greater than or equal to the level of Q and greater than the level of the possibly composite formula $H(x)$ and, in the second case, the level of P must be positive and greater than or equal to the level of Q. In both cases x denotes the totality of all variables that occur free in the production.

2.4. The level of a production is defined to be the level of the predicate symbol which occurs in its conclusion.

2.5. The productions are schemata for defining predicates and are to be understood as they stand once it has been stipulated, first, that the logical constants are to have their constructive meaning, second, that the variables range over the closed terms and, third, that the statement below a horizontal line follows from the statements above the line. For example, the first and second Peano axiom may be written

$$N0 \qquad \frac{Nx}{Nsx}$$

provided the unary predicate symbol N is used to express the property of being a natural number.

2.6. The above interpretation of the productions may be elaborated a bit more so as to conform with the usual intuitionistic interpretation of the logical constants. The productions are then understood as instructions telling us how we are allowed to construct proofs of atomic statements of the form Pt where t is a sequence of closed terms. Consider first a production of level 0. Such a production is necessarily ordinary and tells us that if we have proofs of $Qq(t)$, ..., $Rr(t)$ where t is a sequence of closed terms, then we have a proof of $Pp(t)$. Thus a proof of Pt where P is a predicate symbol of level 0 may be viewed as a finite tree made up by closed substitution instances of the productions of level 0 and may, if so desired, be identified with its symbolic representation. Having defined what constitutes a proof of a closed atomic formula of level 0, we know automatically from the intuitionistic interpretation of the logical constants what constitutes a proof of a closed composite formula of level 0. Consider now a production of level 1. If it is ordinary, it tells us just as before that if we have proofs of $Qq(t)$, ..., $Rr(t)$ then we have a proof of $Pp(t)$. If it is a →production it tells us that if we have a method of transforming an arbitrary proof of $H(t)$ into a proof of $Qq(t)$ where t is a sequence of closed terms, then we have a proof of $Pp(t)$. Note that, since the level of $H(t)$ equals 0, we are supposed to have understood already what constitutes a proof of $H(t)$. Finally, a Λproduction tells us that if we have a method which allows us for every closed term u to construct a proof of $Qq(t,u)$ where t is a sequence of closed terms, then we have a proof of $Pp(t)$. Now we know we are allowed to prove closed atomic formulae of level 1 and can proceed inductively to define what constitutes a proof of a closed composite formula of level 1, a closed atomic formula of level 2 and so on.

2.6.1. Note that the definition of what constitutes a proof of an atomic statement Pt where t is a sequence of closed terms is itself an iterated inductive definition which after arithmetization can be expressed by means of a finite number of ordinary and generalized productions.

2.7. The productions can also be interpreted by means of the impredicative notion of species. Indeed, the level restrictions ensure that there are minimal species corresponding to the predicate symbols of level 0 that satisfy the productions of level 0 and that, given these, there are minimal species corresponding to the predicate symbols of level 1 that satisfy the productions of level 1 and that, given the species determined by the predicate symbols of level 0 and 1, there are minimal species corresponding to the predicate symbols of level 2 that satisfy the productions of level 2 and so on. The species determined by the predicate symbols of level n are precisely the species which are ω_n recursively enumerable in the sense of alfa recursion theory where $\omega = \omega_0, \omega_1, \ldots, \omega_n, \ldots$ denote the recursively regular ordinals enumerated in increasing order. In particular, the species determined by the predicate symbols of level 0 are precisely the species which are recursively enumerable in the ordinary sense.

3. **The intuitionistic theory of itereated inductive definitions.** This theory formalizes the principles of proof that are implicit in the concepts just introduced together with the usual concepts of first order intuitionistic logic. I shall formulate it as an extension of the system of natural deduction introduced by Gentzen 1934 and studied by Prawitz 1965.

3.1. A *deduction* is started by making some *assumptions* from which conclusions are drawn by repeatedly applying the following *rules of inference.*

3.2. Rules of inference associated with the logical constants.

3.2.1. $\rightarrow$introduction.

$$\frac{\begin{array}{c}\not F \\ \vdots \\ G\end{array}}{F \to G}$$

The formula F has been crossed out in order to indicate that some occurrences of F as assumption of the deduction of G may have been *cancelled.* This means that the assumptions of the deduction of $F \to G$ are the assumptions of the deduction of G minus the occurrences of F which are cancelled at the inference from G to $F \to G$. When an assumption is cancelled, it must be indicated in some unambiguous way at what inference this happens. For example, Gentzen 1934 marks an assumption that is cancelled by a number and writes the same number at the inference by which it is cancelled.

3.2.2. $\to$elimination or modus ponens.

$$\frac{F \to G \quad F}{G}$$

3.2.3. $\wedge$introduction.

$$\frac{F \quad G}{F \wedge G}$$

3.2.4. $\wedge$elimination.

$$\frac{F \wedge G}{F} \qquad \frac{F \wedge G}{G}$$

3.2.5. $\vee$introduction.

$$\frac{F}{F \vee G} \qquad \frac{G}{F \vee G}$$

3.2.6. $\vee$elimination.

$$\frac{F \vee G \qquad \begin{matrix} \not{F} \\ \vdots \\ H \end{matrix} \qquad \begin{matrix} \not{G} \\ \vdots \\ H \end{matrix}}{H}$$

3.2.7. Λ introduction.

$$\frac{F(x)}{\Lambda x F(x)}$$

This rule is subjected to the restriction that the variable x, whose free occurrences in the deduction of $F(x)$ become bound by the Λ introduction, must not occur free in any assumption of the deduction of $F(x)$.

3.2.8. Λ elimination.

$$\frac{\Lambda x F(x)}{F(t)}$$

3.2.9. V introduction.

$$\frac{F(t)}{\mathrm{V} x F(x)}$$

3.2.10. V elimination.

$$\frac{\mathrm{V} x F(x) \qquad \begin{matrix} \not{F}(x) \\ \vdots \\ G \end{matrix}}{G}$$

This rule is subjected to the restriction that the variable x, whose free occurrences in the deduction of G from $F(x)$ become bound by the V elimination, must not occur free in G or in any assumption of the deduction of G other than $F(x)$.

3.3. The *major premise* of an elimination inference is the premise whose outermost logical sign is eliminated by the inference. The other premises of the inference are called *minor premises*. The deduction of the major premise is called the *major deduction* and the deductions of the minor premises are called the *minor deductions* of the elimination inference.

3.4. Rules of inference for the inductively defined predicates.

3.4.1. Ordinary production.

$$\frac{Qq(t) \quad \dots \quad Rr(t)}{Pp(t)}$$

3.4.2. →production.

$$\begin{array}{c} \not{H}(t) \\ \vdots \\ \dfrac{Qq(t)}{Pp(t)} \end{array}$$

The formula $H(t)$ has been crossed out in order to indicate that some occurrences of $H(t)$ as assumption of the deduction of $Qq(t)$ may be cancelled at the inference from $Qq(t)$ to $Pp(t)$.

3.4.3. ⋀production.

$$\frac{Qq(t,y)}{Pp(t)}$$

This rule is subjected to the restriction that the variable y must occur neither in t nor free in any assumption of the deduction of $Qq(t,y)$.

3.4.4. Elimination of an inductively defined predicate.

$$\frac{Pt \quad \begin{array}{c}\text{minor}\\ \text{deductions}\end{array}}{F(t)}$$

A production should be considered as an introduction rule for the predicate which occurs in its conclusion. The rule which has been schematically represented above is the corresponding elimination rule.

3.5. In an application of the elimination rule for an inductively defined predicate the formula Pt is called the major premise. Definition of minor premise, major deduction and minor deduction as for the logical elimination rules.

3.6. Before explaining the elimination rule for an inductively defined predicate which has been schematically represented above, I need to define what it means for a predicate symbol to be *linked* with another predicate symbol. First, every predicate symbol is linked with itself. Second, if P occurs in the conclusion of an ordinary production

$$\frac{Qq(x) \quad \ldots \quad Rr(x)}{Pp(x)}$$

then P is linked with every predicate symbol which is linked with one of $Q, \ldots, R$. Third, if P occurs in the conclusion of a generalized production

$$\frac{H(x) \to Qq(x)}{Pp(x)} \qquad \frac{\Lambda y Qq(x,y)}{Pp(x)}$$

then P is linked with every predicate symbol which is linked with Q.

3.7. An instance of the elimination rule for an inductively defined predicate P is obtained as follows. Associate with every predicate symbol which is linked with P an abstraction term, that is, a formula and as many variables as indicated by the place index of the predicate symbol in question

$$\begin{array}{cccc} P & Q & R & \ldots \\ \\ \lambda x F(x) & \lambda y G(y) & \lambda z H(z) & \ldots \end{array}$$

For every ordinary production which in its conclusion has a predicate symbol which is linked with P, say

$$\frac{Qq(x) \quad \ldots \quad Rr(x)}{Pp(x)}$$

there should among the minor deductions of the elimination inference be one of the form

$$\begin{array}{ccc} G(q(x)) & \ldots & H(r(x)) \\ \vdots & & \vdots \\ & F(p(x)) & \end{array}$$

satisfying the restriction that a variable in the sequence x, whose free occurrences in the minor deduction become bound by the elimination inference, must not occur free in an assumption other than the indicated $G(q(x))$, ..., $H(r(x))$ which are all cancelled at the inference we are considering. Similarly, for every generalized production which in its conclusion has a predicate symbol which is linked with P, say

$$\frac{H(x) \to Qq(x)}{Pp(x)} \qquad \frac{\bigwedge y Qq(x,y)}{Pp(x)}$$

there should among the minor deductions of the elimination inference be one of the form

$$\begin{array}{c} H(x) \to G(q(x)) \\ \vdots \\ F(p(x)) \end{array} \qquad \begin{array}{c} \bigwedge y G(q(x,y)) \\ \vdots \\ F(p(x)) \end{array}$$

satisfying the restriction that a variable in the sequence x, whose free occurrences in the minor deduction become bound by the elimination inference, must not occur free in an assumption other than the indicated $H(x) \to G(q(x))$ and $\bigwedge y G(q(x,y))$, respectively, which are cancelled at the inference we are considering.

3.8. Examples.

3.8.1. Let $\perp$ be a 0ary predicate symbol of level 0 which does not occur in the conclusion of any of the productions. Thus there is no introduction rule for $\perp$. The elimination rule described above takes the form

$$\frac{\perp}{F}$$

which is nothing but the intuitionistic rule of absurdity.

3.8.2. Suppose we introduce a binary predicate symbol E of level 0 for equality by means of the ordinary production

$$Exx$$

with zero premises. E is not to occur in the conclusion of any other production. The introduction rule for E makes

$$Ett$$

an axiom for every term t, and the corresponding elimination rule takes the form

$$\frac{Etu \quad F(x,x)}{F(t,u)}$$

which is one way of formulating the standard rules for equality as seen by choosing $F(x,y)$ of the form $F(x) \to F(y)$.

3.8.3. Introduce N of level 0 for the property of being a natural number by means of the productions

$$N0 \qquad \frac{Nx}{Nsx}$$

and the stipulation that N must not occur in the conclusion of any other production. The elimination rule for N then takes the form

$$\frac{Nt \quad F(0) \quad \begin{matrix} \cancel{F(x)} \\ \vdots \\ F(sx) \end{matrix}}{F(t)}$$

which is nothing but the induction schema.

3.9. A deduction all of whose assumptions have been cancelled is said to be a *proof* of its end formula. A formula is *provable* if there exists a proof of it.

4. Rules of contraction.

4.1. If a logical constant or an inductively defined predicate is introduced only to be immediately eliminated we shall say that a *cut* occurs, and a formula which is at the same time the conclusion of an introduction inference and the major premise of an elimination inference will be called a *cut formula*.

4.2. To each possible form of cut there corresponds a *rule of contraction* which tells us how we are allowed to simplify a deduction which ends with an elimination inference whose major premise is the conclusion of an introduction inference by eliminating the cut.

4.2.1. →contraction.

$$\dfrac{\dfrac{\begin{array}{c}\not F\\ \vdots\\ G\end{array}}{F \to G} \quad \begin{array}{c}\vdots\\ F\end{array}}{G} \quad \text{contr} \quad \begin{array}{c}\vdots\\ F\\ \vdots\\ G\end{array}$$

Before the contraction can be carried out some bound variables in the deduction of G from F may have to be renamed so that no free variable in the deduction of F becomes bound after the contraction.

4.2.2. ∧ contraction.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ F\end{array} \quad \begin{array}{c}\vdots\\ G\end{array}}{F \wedge G}}{F} \quad \text{contr} \quad \begin{array}{c}\vdots\\ F\end{array}$$

The case when G instead of F is inferred from $F \wedge G$ is quite similar.

4.2.3. ∨ contraction.

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ F\end{matrix}}{F\vee G}\quad\begin{matrix}F\\ \vdots\\ H\end{matrix}\quad\begin{matrix}G\\ \vdots\\ H\end{matrix}}{H}\qquad\text{contr}\qquad\begin{matrix}\vdots\\ F\\ \vdots\\ H\end{matrix}$$

Before the contraction can be carried out, some bound variables in the deduction of H from F may have to be renamed so that no free variable in the deduction of F becomes bound after the contraction. The case when $F\vee G$ is inferred from G instead of F is quite similar.

4.2.4. Λ contraction.

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ F(x)\end{matrix}}{\Lambda x F(x)}}{F(t)}\qquad\text{contr}\qquad\begin{matrix}\vdots\\ F(t)\end{matrix}$$

The simplified deduction of $F(t)$ is obtained by substituting the term t for all free occurrences of x in the deduction of $F(x)$. Before doing this, however, some of the bound variables of this deduction may have to be renamed so that no variable in t becomes bound after the substitution. In the sequel it will be tacitly assumed that bound variables are renamed whenever necessary in order to avoid undesired ties.

4.2.5. V contraction.

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ F(t)\end{matrix}}{V x F(x)}\quad\begin{matrix}F(x)\\ \vdots\\ G\end{matrix}}{G}\qquad\text{contr}\qquad\begin{matrix}\vdots\\ F(t)\\ \vdots\\ G\end{matrix}$$

The lower part of the simplified deduction is obtained by substituting the term t for all free occurrences of x in the deduction of G from $F(x)$.

4.2.6. Contraction of an ordinary production.

$$\dfrac{\dfrac{\begin{matrix}\vdots & & \vdots\\ Qq(t) & \dots & Rr(t)\end{matrix}}{Pp(t)} \qquad \begin{matrix}\text{minor}\\ \text{deductions}\end{matrix}}{F(p(t))} \quad \text{contr}$$

$$\begin{matrix} \dfrac{\begin{matrix}\vdots\\ Qq(t)\end{matrix} \quad \begin{matrix}\text{minor}\\ \text{deductions}\end{matrix}}{G(q(t))} & \dots & \dfrac{\begin{matrix}\vdots\\ R(t)\end{matrix} \quad \begin{matrix}\text{minor}\\ \text{deductions}\end{matrix}}{H(r(t))} \\ & \ddots \quad \therefore & \\ & F(p(t)) & \end{matrix}$$

The lower part of the simplified deduction is obtained from that one of the minor deductions of the given deduction which is of the form

$$\begin{matrix} G(q(x)) & \dots & H(r(x)) \\ & \ddots \quad \therefore & \\ & F(p(x)) & \end{matrix}$$

by substituting each term in the sequence t for all free occurrences of the respective variable in the sequence x.

4.2.7. Contraction of a →production.

$$\dfrac{\dfrac{\begin{matrix}[H(t)]\\ \vdots\\ Qq(t)\end{matrix}}{Pp(t)} \quad \begin{matrix}\text{minor}\\ \text{deductions}\end{matrix}}{F(p(t))} \quad \text{contr} \quad \begin{matrix} \dfrac{\dfrac{\begin{matrix}[H(t)]\\ \vdots\\ Qq(t)\end{matrix} \quad \begin{matrix}\text{minor}\\ \text{deductions}\end{matrix}}{G(q(t))}}{H(t) \to G(q(t))} \\ \vdots \\ F(p(t)) \end{matrix}$$

The lower part of the simplified deduction is obtained from that one of the minor deductions of the given deduction which is of the form

$$H(x) \to G(q(x))$$
$$\vdots$$
$$F(p(x))$$

by substituting each term in the sequence t for all free occurrences of the respective variable in the sequence x.

4.2.8. Contraction of a Λproduction.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ Qq(t,y)\end{array}}{Pp(t)} \quad \text{minor deductions}}{F(p(t))} \quad \text{contr} \quad \begin{array}{c}\dfrac{\dfrac{\begin{array}{c}\vdots\\ Qq(t,y)\end{array} \quad \text{minor deductions}}{G(q(t,y))}}{\Lambda y G(q(t,y))}\\ \vdots\\ F(p(t))\end{array}$$

The lower part of the simplified deduction is obtained from that one of the minor deductions of the given deduction which is of the form

$$\Lambda y G(q(x,y))$$
$$\vdots$$
$$F(p(x))$$

by substituting each term in the sequence t for all free occurrences of the respective variable in the sequence x.

4.3. A deduction *reduces* to another deduction if the latter can be obtained from the former by repeated contractions of subdeductions, where by a subdeduction I mean an initial part of a deduction.

4.4. A deduction which cannot be further reduced is said to be *cut free* or *normal*. We are now prepared to formulate the *Hauptsatz* or *normal form theorem* for the intuitionistic theory of iterated inductive definitions.

5. **Hauptsatz.** *Every deduction reduces to a normal deduction.*

5.1. The structure of my proof of Hauptsatz can be described as follows. First, I shall define by ordinary induction what it means for a deduction to be *normalizable.* Roughly speaking, a deduction is normalizable if it can be brought on normal form by carrying out successive contractions of subdeductions in a specific order which I believe is the most natural one. Second, I shall define what it means for a deduction to be *computable.* The definition of computability, which utilizes iterated inductive definitions of precisely the kind that my theory formalizes, is such that it is immediately clear that a computable deduction is also normalizable. The proof is then completed by showing that every deduction is computable. Once the notion of computability has been defined, this final part of the proof, although involving many cases, is in principle a mere verification.

5.2. Suppose a deduction ends with an elimination inference. Note that an elimination inference always has one and only one major premise. Thus, by always choosing the major premise of an elimination inference, we can in a unique way proceed upwards in the deduction from the end formula until we reach either a cut or a top formula. In the first case, the cut we hit upon will be called the *main cut* and, in the second case, the branch we have proceeded along will be called the *main branch.* Since the main branch never passes through an introduction inference and always through the major premise of an elimination inference, the top formula in the beginning of the main branch cannot have been cancelled. This fact will be crucial when we come to the determination of the form of cut free deductions.

6. Definition of what it means for a deduction to be normalizable.

6.1. The deduction consists solely of an assumption. Then it is normalizable outright.

6.2. The last inference of the deduction is an introduction. Then it is normalizable provided the deductions of the premises of this inference are all normalizable.

6.3. The last inference of the deduction is an elimination.

6.3.1. The deduction has a main cut. Then it is normalizable provided the deduction which is obtained from it by eliminating the main cut is normalizable.

6.3.2. The deduction has no main cut. Then it is normalizable provided the minor deductions of the eliminations on the main branch are all normalizable.

6.4. Each clause in the definition of normalizability asserts that a given deduction is normalizable provided certain (finitely many) other deductions, which we may call the predecessors of the given deduction, are all normalizable. Thus, a deduction is normalizable if and only if the tree of its successive predecessors is wellfounded. This finite tree is then called the *normalization* of the deduction.

6.5. If a deduction is normalizable, then it reduces to a normal deduction. This is seen immediately by induction on the normalization of the deduction.

7. Definition of what it means for a deduction to be computable. I proceed by induction on $\omega n + m$ where m is the number of logical signs and n the level of the end formula of the deduction. Thus, in the definition of computability for deductions whose end formula contains m logical signs and is of level n, I assume that computability has already been defined for all deductions whose end formula has a lower value of $\omega n + m$.

7.1. The deduction consists solely of an assumption. Then it is computable outright.

7.2. The last inference of the deduction is an introduction.

7.2.1. $\rightarrow$introduction.

$$\dfrac{\begin{matrix} \not F \\ \vdots \\ G \end{matrix}}{F \rightarrow G}$$

is computable provided

$$\begin{array}{c} \vdots \\ F \\ \vdots \\ G \end{array}$$

is computable for every computable deduction

$$\begin{array}{c} \vdots \\ F \end{array}$$

Note that the formula F has a lower value of $\omega n + m$ than $F \to G$.

7.2.2. $\wedge$ introduction.

$$\frac{\begin{array}{cc} \vdots & \vdots \\ F & G \end{array}}{F \wedge G}$$

is computable provided

$$\begin{array}{ccc} \vdots & \qquad & \vdots \\ F & & G \end{array}$$

are both computable.

7.2.3. $\vee$ introduction.

$$\frac{\begin{array}{c} \vdots \\ F \end{array}}{F \vee G}$$

is computable provided

$$\begin{array}{c}\vdots\\ F\end{array}$$

is computable. Similarly when $F \vee G$ is inferred from G instead of F.

7.2.4. Λ introduction.

$$\begin{array}{c}\vdots\\ \dfrac{F(x)}{\Lambda x F(x)}\end{array}$$

is computable provided

$$\begin{array}{c}\vdots\\ F(t)\end{array}$$

is computable for every term t.

7.2.5. V introduction.

$$\begin{array}{c}\vdots\\ \dfrac{F(t)}{\mathsf{V} x F(x)}\end{array}$$

is computable provided

$$\begin{array}{c}\vdots\\ F(t)\end{array}$$

is computable.

7.2.6. Ordinary production.

$$\frac{\begin{array}{ccc} \vdots & & \vdots \\ Qq(t) & \ldots & Rr(t) \end{array}}{Pp(t)}$$

is computable provided

$$\begin{array}{ccc} \vdots & & \vdots \\ Qq(t) & \ldots & Rr(t) \end{array}$$

are all computable.

7.2.7. →production.

$$\frac{\begin{array}{c} \not{H}(t) \\ \vdots \\ Qq(t) \end{array}}{Pp(t)}$$

is computable provided

$$\begin{array}{c} \vdots \\ H(t) \\ \vdots \\ Qq(t) \end{array}$$

is computable for every computable deduction

$$\begin{array}{c} \vdots \\ H(t) \end{array}$$

Note that, because of the level restrictions on a →production, the formula $H(t)$ has a lower value of $\omega n + m$ than $Qq(t)$.

7.2.8. $\wedge$ production.

$$\begin{array}{c} \vdots \\ \underline{Qq(t,y)} \\ Pp(t) \end{array}$$

is computable provided

$$\begin{array}{c} \vdots \\ Qq(t,u) \end{array}$$

is computable for every term u.

7.3. The last inference of the deduction is an elimination.

7.3.1. The deduction has a main cut. Then it is computable provided the deduction which is obtained from it by eliminating the main cut is computable.

7.3.2. The deduction has no main cut. Then it is computable provided the minor deductions of the eliminations on the main branch are all normalizable.

7.4. Each clause in the definition of computability asserts that a deduction is computable provided certain (infinitely many, in general) other deductions, which we may call the predecessors of the given deduction, are all computable. Thus, a deduction is computable if and only if the tree of its successive predecessors is wellfounded. This infinite wellfounded tree is then called the *computation* of the deduction.

7.5. A deduction which is computable is also normalizable. This is seen by comparing clause for clause the definition of normalizability with the definition of computability, thereby remembering that a deduction which consists solely of an assumption is computable. Expressed differently, what we have achieved is an imbedding of the finite tree which we have called the normalization of a deduction into the huge infinite tree which we have called its computation. Therefore, the wellfoundedness of the former follows from the wellfoundedness of the latter.

8. **Theorem**. *Every deduction is computable.* This is proved by induction on the length of the deduction, but we have to make a stronger induction hypothesis, namely, that if we, first, substitute arbitrary terms for its free variables and, second, to the deduction obtained after the substitution attach arbitrary computable deductions of its assumptions, then the resulting deduction is computable. Several cases have to be distinguished depending on how the end formula of the deduction has been inferred.

8.1. The deduction consists solely of an assumption. Trivial.

8.2. →introduction. We have to show that a deduction of the form

$$\begin{array}{c} \not{F} \\ \vdots \\ G \\ \hline F \to G \end{array}$$

is computable. By 7.2.1 this is so if

$$\begin{array}{c} \vdots \\ F \\ \vdots \\ G \end{array}$$

is computable for every computable deduction

$$\begin{array}{c} \vdots \\ F \end{array}$$

which follows immediately from the induction hypothesis.

8.2.2. ∧introduction. We have to show that a deduction of the form

$$\frac{\begin{matrix}\vdots & \vdots \\ F & G\end{matrix}}{F \wedge G}$$

is computable. By 7.2.2 this is so if

$$\begin{matrix}\vdots & \qquad & \vdots \\ F & \qquad & G\end{matrix}$$

are both computable which follows immediately from the induction hypothesis.

8.2.3. $\vee$introduction. We have to show that a deduction of the form

$$\frac{\begin{matrix}\vdots \\ F\end{matrix}}{F \vee G}$$

is computable. By 7.2.3 this is so if

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

is computable which follows immediately from the induction hypothesis.

8.2.4. $\bigwedge$introduction. We have to show that a deduction of the form

$$\frac{\begin{matrix}\vdots \\ F(x)\end{matrix}}{\bigwedge x F(x)}$$

is computable. By 7.2.4 this is so if

$$\begin{matrix}\vdots \\ F(t)\end{matrix}$$

is computable for every term t which follows immediately from the induction hypothesis.

8.2.5. $\vee$ introduction. We have to show that a deduction of the form

$$\frac{\begin{matrix}\vdots\\ F(t)\end{matrix}}{\vee x F(x)}$$

is computable. By 7.2.5 this is so if

$$\begin{matrix}\vdots\\ F(t)\end{matrix}$$

is computable which follows immediately from the induction hypothesis.

8.2.6. Ordinary production. We have to show that a deduction of the form

$$\frac{\begin{matrix}\vdots\\ Qq(t)\end{matrix} \quad \dots \quad \begin{matrix}\vdots\\ Rr(t)\end{matrix}}{Pp(t)}$$

is computable. By 7.2.6 this is so if

$$\begin{matrix}\vdots\\ Qq(t)\end{matrix} \quad \dots \quad \begin{matrix}\vdots\\ Rr(t)\end{matrix}$$

are all computable which follows immediately from the induction hypothesis.

8.2.7. $\rightarrow$production. We have to show that a deduction of the form

$$\frac{\begin{matrix}\not{H}(t)\\ \vdots\\ Qq(t)\end{matrix}}{Pp(t)}$$

is computable. By 7.2.7 this is so if

$$\begin{array}{c} \vdots \\ H(t) \\ \vdots \\ Qq(t) \end{array}$$

is computable for every computable deduction

$$\begin{array}{c} \vdots \\ H(t) \end{array}$$

which follows immediately from the induction hypothesis.

8.2.8. Λproduction. We have to show that a deduction of the form

$$\begin{array}{c} \vdots \\ \underline{Qq(t,y)} \\ Pp(t) \end{array}$$

is computable. By 7.2.8 this is so if

$$\begin{array}{c} \vdots \\ Qq(t,u) \end{array}$$

is computable for every term u which follows immediately from the induction hypothesis.

8.3. The last inference of the deduction is an elimination. We use induction on the computation of the major deduction of this elimination inference. This induction is, of course, subordinate to the basic induction on the length of the given deduction. Basis. The major deduction of the final elimination is computable according to 7.1 or 7.3.2. Then the computability of the deduction follows from 7.3.2 by using the fact that, according to the basic induction hypothesis, the minor deductions of the final elimination are computable and a fortiori normalizable. Induction step. If the major deduction of the final elimination is computable according to 7.3.1, then the computability of the deduction follows immediately from the subordinate induction hypothesis and 7.3.1. The crucial case arizes when the last inference of the major deduction of the final elimination is an introduction, that is, when the major premise of the final elimination is a cut formula. Eight subcases have to be distinguished depending on the form of this cut.

8.3.1.

$$\dfrac{\dfrac{\begin{matrix}\not{F}\\ \vdots\\ G\end{matrix}}{F\to G}\quad \begin{matrix}\vdots\\ F\end{matrix}}{G}$$

According to the basic induction hypothesis, the deductions

$$\dfrac{\begin{matrix}\not{F}\\ \vdots\\ G\end{matrix}}{F\to G}\qquad\qquad \begin{matrix}\vdots\\ F\end{matrix}$$

are both computable. By 7.2.1 so is

$$\begin{matrix}\vdots\\ F\\ \vdots\\ G\end{matrix}$$

The computability of the given deduction now follows from 7.3.1.

8.3.2.

$$\dfrac{\dfrac{\begin{matrix}\vdots & \vdots\\ F & G\end{matrix}}{F\wedge G}}{F}$$

According to the basic induction hypothesis

$$\dfrac{\begin{matrix}\vdots & \vdots\\ F & G\end{matrix}}{F\wedge G}$$

is computable. By 7.2.2 so is

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

The computability of the given deduction now follows from 7.3.1.

8.3.3.

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ F\end{matrix}}{F\vee G} \quad \begin{matrix}\not{F}\\ \vdots\\ H\end{matrix} \quad \begin{matrix}\not{G}\\ \vdots\\ H\end{matrix}}{H}$$

According to the basic induction hypothesis

$$\dfrac{\begin{matrix}\vdots\\ F\end{matrix}}{F\vee G}$$

is computable. By 7.2.3 so is

$$\begin{array}{c} \vdots \\ F \end{array}$$

Applying the basic induction hypothesis again, we can conclude that

$$\begin{array}{c} \vdots \\ F \\ \vdots \\ H \end{array}$$

is computable. The computability of the given deduction now follows from 7.3.1.

8.3.4.

$$\begin{array}{c} \vdots \\ \dfrac{\dfrac{F(x)}{\Lambda x F(x)}}{F(t)} \end{array}$$

According to the basic induction hypothesis

$$\begin{array}{c} \vdots \\ \dfrac{F(x)}{\Lambda x F(x)} \end{array}$$

is computable. By 7.2.4 so is

$$\begin{array}{c} \vdots \\ F(t) \end{array}$$

The computability of the given deduction now follows from 7.3.1.

8.3.5.

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ F(t)\end{matrix}}{\mathrm{V}xF(x)} \qquad \begin{matrix}\not{F}(x)\\ \vdots\\ G\end{matrix}}{G}$$

According to the basic induction hypothesis

$$\dfrac{\begin{matrix}\vdots\\ F(t)\end{matrix}}{\mathrm{V}xF(x)}$$

is computable. By 7.2.5 so is

$$\begin{matrix}\vdots\\ F(t)\end{matrix}$$

Applying the basic induction hypothesis again, we can conclude that

$$\begin{matrix}\vdots\\ F(t)\\ \vdots\\ G\end{matrix}$$

is computable. The computability of the given deduction now follows from 7.3.1.

8.3.6.

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ Qq(t)\end{matrix} \quad \dots \quad \begin{matrix}\vdots\\ Rr(t)\end{matrix}}{Pp(t)} \qquad \begin{matrix}\text{minor}\\ \text{deductions}\end{matrix}}{F(p(t))}$$

By 7.3.1 it suffices to show that

$$\begin{array}{ccc} \vdots & & \vdots \\ \dfrac{Qq(t) \quad \text{minor deductions}}{G(q(t))} & \cdots & \dfrac{Rr(t) \quad \text{minor deductions}}{H(r(t))} \\ & \ddots \qquad \cdot^{\cdot^{\cdot}} & \\ & F(p(t)) & \end{array}$$

is computable. Now, the computability of the deductions of $G(q(t))$, ..., $H(r(t))$ follows from the subordinate induction hypothesis stated in 8.3, and the computability of the whole deduction then follows from the basic induction hypothesis. Remember that the deduction of $F(p(t))$ from $G(q(t))$, ..., $H(r(t))$ is obtained from that one of the minor deductions which is of the form

$$\begin{array}{ccc} G(q(x)) & \cdots & H(r(x)) \\ & \ddots \qquad \cdot^{\cdot^{\cdot}} & \\ & F(p(x)) & \end{array}$$

by substituting each term in the sequence t for all free occurrences of the respective variable in the sequence x.

8.3.7.

$$\begin{array}{l} \cancel{H(t)} \\ \quad \vdots \\ \dfrac{\dfrac{Qq(t)}{Pp(t)} \quad \begin{array}{c}\text{minor} \\ \text{deductions}\end{array}}{F(p(t))} \end{array}$$

By 7.3.1 it suffices to show that

$$
\begin{array}{l}
\not{H}(\not{t})\\
\vdots \qquad \text{minor}\\
\underline{Qq(t) \quad \text{deductions}}\\
\underline{\qquad G(q(t)) \qquad}\\
H(t) \to G(q(t))\\
\qquad \vdots\\
\qquad F(p(t))
\end{array}
$$

is computable. This follows from the basic induction hypothesis if we can prove the computability of

$$
\begin{array}{l}
\not{H}(\not{t})\\
\vdots \qquad \text{minor}\\
\underline{Qq(t) \quad \text{deductions}}\\
\underline{\qquad G(q(t)) \qquad}\\
H(t) \to G(q(t))
\end{array}
$$

By 7.2.1 the latter deduction is computable provided

$$
\begin{array}{l}
\vdots\\
H(t)\\
\vdots \qquad \text{minor}\\
\underline{Qq(t) \quad \text{deductions}}\\
\qquad G(q(t))
\end{array}
$$

is computable for every computable deduction

$$
\begin{array}{l}
\vdots\\
H(t)
\end{array}
$$

This, in turn, follows from the subordinate induction hypothesis stated in 8.3.

8.3.8.

$$\cfrac{\cfrac{\begin{array}{c}\vdots\\ Qq(t,y)\end{array}}{Pp(t)}\quad \begin{array}{c}\text{minor}\\ \text{deductions}\end{array}}{F(p(t))}$$

By 7.3.1 it suffices to show that

$$\begin{array}{c}\cfrac{\cfrac{\begin{array}{c}\vdots\\ Qq(t,y)\end{array}\quad \begin{array}{c}\text{minor}\\ \text{deductions}\end{array}}{G(q(t,y))}}{\bigwedge y G(q(t,y))}\\ \vdots \\ F(p(t))\end{array}$$

is computable. This follows from the basic induction hypothesis if we can prove the computability of

$$\cfrac{\cfrac{\begin{array}{c}\vdots\\ Qq(t,y)\end{array}\quad \begin{array}{c}\text{minor}\\ \text{deductions}\end{array}}{G(q(t,y))}}{\bigwedge y G(q(t,y))}$$

By 7.2.4 the latter deduction is computable provided

$$\cfrac{\begin{array}{c}\vdots\\ Qq(t,u)\end{array}\quad \begin{array}{c}\text{minor}\\ \text{deductions}\end{array}}{G(q(t,u))}$$

is computable for every term u. This, in turn, follows from the subordinate induction hypothesis stated in 8.3.

9. Corollaries which follow from Hauptsatz by combinatorial reasoning.

9.1. *If an atomic formula of level* 0 *is provable, then it has a proof which consists entirely of applications of the productions of level* 0. *Note that these are all ordinary.*

9.1.1. Suppose we are given a proof of an atomic formula of level 0. According to Hauptsatz, it reduces to a normal proof. This normal proof must consist entirely of applications of the productions of level 0, because, otherwise, there would be a lowest formula in the proof which is not the conclusion of a production of level 0. The proof of this formula must end with an elimination inference and, consequently, the assumption in the beginning of its main branch cannot have been cancelled. However, all asumptions of a proof are cancelled. We have reached a contradiction.

9.1.2. We might, using the terminology of Hilbert, say that atomic formulae of lowest level express *real* statements and that composite formulae as well as atomic formulae of higher level express *ideal* statements. Corollary 9.1 may then be interpreted as saying that we can always eliminate the use of ideal statements from a proof of a real statement. However, our proof of this fact uses the full force of the ideal statements which means that, in agreement with Gödel's second theorem, no reduction of the kind Hilbert aimed at is achieved. Nevertheless, something else and important follows from our analysis, namely, that, once we have formalized a proof of a real statement, a proof in which ideal statements may occur as a vehicle, we can find the direct proof, which does not make the excursion via ideal statements, *mechanically*, that is, by symbol manipulation.

9.2. *If* $F \vee G$ *is provable, then either* F *or* G *is provable.*

9.2.1. Suppose we are given a proof of $F \vee G$. According to Hauptsatz, it reduces to a normal proof. This normal proof cannot end with an elimination inference, because in that case the assumption in the beginning of its main branch could never have been cancelled. Thus, it ends with an introduction inference which necessarily must be an application of the $\vee$introduction rule. The proof of the premise of this final $\vee$introduction is either a proof of F or a proof of G.

9.3. *If* $\vee xF(x)$ *is provable, then so is* $F(t)$ *for some term* t.

9.3.1. The normal proof of $\bigvee x F(x)$ which we get by applying Hauptsatz must end with an introduction inference, because, otherwise, the assumption in the beginning of its main branch could not have been cancelled. Consequently, it is of the form

$$\frac{\begin{array}{c}\vdots\\ F(t)\end{array}}{\bigvee x F(x)}$$

and the desired proof of $F(t)$ is obtained by deleting the last inference.

10. Probable wellorderings. *The precise bound on the provable wellorderings of the intuitionistic theory of iterated inductive definitions equals*

$$\operatorname*{Lim}_n F^1_{F^2_{\cdot^{\cdot^{\cdot}}_{F^n_2(1)}}(1)}(1)$$

in Isles's 1968 *generalized Bachmann notation.*

10.1. Let $O(n)$ denote the least upper bound of Takeuti's 1957 ordinal diagrams of order n. According to Levitz's lecture at the conference in Buffalo 1968

$$O(n) = F^1_{F^2_{\cdot^{\cdot^{\cdot}}_{F^n_\omega(1)}}(1)}(1)$$

and

$$O(n-1) < F^1_{F^2_{\cdot^{\cdot^{\cdot}}_{F^n_2(1)}}(1)}(1) < O(n)$$

for every n. Also, Kreisel 1964 has formalized the proof of the wellfoundedness of $O(n)$ in the intuitionistic theory of iterated inductive definitions with predicates of level n at most. This shows that the least upper bound of the provable wellorderings of the intuitionistic theory of iterated inductive definitions is at least as big as

$$\operatorname{Lim}_n F^1_{F^2_{\cdot^{\cdot^{\cdot}}_{F^n_2(1)}}(1)}(1)$$

In order to prove the converse inequality we shall consider the intuitionistic theory of iterated inductive definitions based on the productions

$$N0 \qquad \frac{Nx}{Nsx} \qquad \frac{Esxsy}{Exy} \qquad \frac{E0sx}{\perp} \qquad Exx \qquad \frac{Exy}{Esxsy}$$

the productions

$$\frac{Px_1 \ldots x_n \quad Ex_1y_1 \quad \ldots \quad Ex_ny_n}{Py_1 \ldots y_n} \qquad \frac{\perp}{Px_1 \ldots x_n}$$

for every predicate symbol P and defining productions of arbitrarily many further predicates with which neither $\perp$ nor E is to be linked. Replacing every predicate except $\perp$ and E by its least species interpretation as described in 2.7, in particular, N by

$$\lambda x \Lambda X(X0 \wedge \Lambda x(Xx \rightarrow Xsx) \rightarrow Xx)$$

we interpret this theory into intuitionistic second order logic with the axioms for equality, the third and fourth Peano axiom

$$\Lambda x \Lambda y(Esxsy \rightarrow Exy) \qquad \Lambda x(E0sx \rightarrow \perp)$$

and the comprehension axiom restricted to formulae which are semi isolated in the sense of Takeuti 1967. Now, Takeuti 1967 showed the consistency of his system SJNN, which is equivalent to classical second order logic with the semi isolated comprehension axiom, the axioms for equality and the third and fourth Peano axiom, by using the principle of transfinite induction on

$$\operatorname*{Lim}_n O(n)$$

as the only non finite method of proof. Consequently, the wellfoundedness of this ordinal cannot be proved in the intuitionistic theory of iterated inductive definitions since, in that case, we could formalize a consistency proof for the theory specified above in the theory itself, contradicting Gödel's second theorem.

10.2. At a seminar in Stanford summer 1969, I conjectured that if we only allow predicates of level less than n in the intuitionistic theory of iterated inductive definitions the precise bound on the provable wellorderings equals

$$F^1_{F^2_{\cdot_{\cdot_{\cdot_{F^n_2(1)}}}}(1)}(1)$$

For $n = 1$ and $n = 2$ this has been proved by Gentzen 1943 and Howard and Gerber 1968, respectively, because

$$F^1_2(1) = \epsilon_0 \qquad F^1_{F^2_2(1)}(1) = \varphi_{\epsilon_{\Omega+1}}(1)$$

Also, Zucker 1969 has, without knowledge of my conjecture, demonstrated its validity for $n = 3$.

10.3. In the above determination of the ordinal associated with the intuitionistic theory of iterated inductive definitions, all the hard part of the analysis was taken from Takeuti 1967. I believe, however, that it will be possible to carry out the ordinal analysis in a much more perspicuous way directly for the theory of iterated inductive definitions. This belief is based on the following observations. Looking at the definition of computability, one sees that the computation of a deduction with end formula of level n is an ω_n arithmetical tree of deductions and, hence, that its length is dominated by ω_{n+1}. However, once it has been proved that every deduction is computable, it appears that the computations are actually recursive trees of deductions and hence that their lengths are dominated by ω_1 already. (This does not mean, of course, that we have eliminated the use of the higher constructive number classes, because they enter effectively into the proof of the recursiveness of the computations.). Also, since there is a recursive procedure which associates with every deduction its computation, the lengths of the computations will be uniformly bounded by a certain recursive ordinal. I expect that it will be possible to estimate the lengths of the computations by means of the ordinal diagrams of finite order or, equivalently, the generalized Bachmann notations considered above. Conversely, let R be a binary predicate of level 0 and let the unary predicate A of level 1 express accessibility with respect to the relation R. Then it is easy to see that the computation of a proof of At where t is a closed term cannot be shorter than the rank of t with respect to the relation R. Consequently, it is not possible to measure the lengths of the computations by means of a system of ordinal notations which is smaller than

$$\operatorname*{Lim}_n F^1_{F^2_{\cdot^{\cdot^{\cdot^{F^n_2(1)}}}}(1)}(1)$$

References

H.B.Curry and R.Feys, Combinatory logic (North-Holland, Amsterdam, 1958).

S.Feferman, Formal theories for transfinite iterations of generalized inductive definitions and some subsystems of analysis (1969). To appear.

G.Gentzen, Untersuchungen über das logische Schliessen, Math. Z. 39 (1934) 176–210, 405–431, Die Widerspruchsfreiheit der reinen Zahlentheorie, Math. Ann. 112 (1936) 493–565, Beweisbarkeit und Unbeweisbarkeit von Anfangsfällen der transfiniten Induktion in der reinen Zahlentheorie, Math. Ann. 119 (1943) 140–161.

H.Gerber, Brouwer's bar theorem and a system of ordinal notations, in: Intuitionism and Proof Theory, eds. A.Kino, J.Myhill and R.E.Vesley (North-Holland, Amsterdam, 1968).

K.Gödel, Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes, Dialectica 21 (1958) 280–287.

D.Isles, Regular ordinals and normal forms, in: Intuitionism and Proof Theory, eds. A.Kino, J.Myhill and R.E.Vesley (North-Holland, Amsterdam, 1970).

G.Kreisel, Generalized inductive definitions, Reports of the seminar on foundations of analysis, Sect. III, Stanford, 1963, Review, Zentralblatt für Mathematik 106 (1964) 237–238.

D.Prawitz, Natural deduction (Almqvist and Wiksell, Stockholm, 1965).

W.W.Tait, Intentional interpretations of functionals of finite type, J. Symbolic Logic 32 (1967) 198–212.

G.Takeuti, Ordinal diagrams, J. Math. Soc. Japan 9 (1957) 386–394, Consistency proofs of subsystems of classical analysis, Ann. Math. 86 (1967) 299–348.

J.Zucker, Characterizations of the provably recursive ordinals of ID_ν for $\nu \geqslant 2$ (1969). Preliminary report.

HAUPTSATZ FOR THE THEORY OF SPECIES

Per MARTIN-LÖF
University of Stockholm

1. Introduction.

1.1. The completeness of the cut free rules for the (impredicative) theory of species was proved by Prawitz 1968. However, using his method, it has not been possible to prove that every deduction can be normalized by successive eliminations of cuts. This seems to be due to the fact that, although one is primarily interested in properties of the proof figures, the semantical notions used apply not to the proof figures but to the formulae of the system.

1.2. A semantical notion, specially invented for the purpose of proving normal form theorems, is Tait's 1967 notion of computability (or convertibility as he says). Martin-Löf 1970 showed that the method of computability applies not only to terms but also to formal proofs and extended it to the intuitionistic theory of iterated inductive definitions. Simultaneously, Girard 1970 has extended the method to a system of terms which is so strong that it can be used to interpret full classical analysis.

1.3. The purpose of the present paper is to show that by making use of Girard's idea it is now possible to analyse the theory of species by means of the method of computability(*). It follows from this analysis that every deduction of the theory of species actually reduces to a cut free deduction.

2. Syntax.

2.1. The language we shall consider contains individual variables, possibly function constants, species variables, possibly species constants, and finally,

(*) This possibility has also been realized by Prawitz. See appendix B of his contribution to this volume.

the logical constants $\to$ and Λ. A universal quantifier binds either an individual variable or a species variable. Absurdity, conjunction, disjunction and existential quantification are all defined as in Prawitz 1965, that is, by putting

$$\begin{aligned}
\bot &= \Lambda Y Y \\
F \wedge G &= \Lambda Y((F \to (G \to Y)) \to Y) \\
F \vee G &= \Lambda Y((F \to Y) \to ((G \to Y) \to Y)) \\
\mathsf{V} x F(x) &= \Lambda Y(\Lambda x(F(x) \to Y) \to Y) \\
\mathsf{V} X F(x) &= \Lambda Y(\Lambda X(F(X) \to Y) \to Y)
\end{aligned}$$

where Y is a 0ary species variable.

2.2. Finite sequences of variables and terms will be denoted by bold face letters. If $\mathbf{x}$ is a sequence of n individual variables and $F(\mathbf{x})$ a formula, then $T = \lambda\mathbf{x}F(\mathbf{x})$ is an $n$ary species term. If $\mathbf{t}$ is a sequence of n individual terms, then $T\mathbf{t}$ denotes the formula $F(\mathbf{t})$.

2.3. Free and bound occurrences of a variable in a formula are defined as usual. If x is one of the variables in the sequence $\mathbf{x}$, then every occurrence of x in $T = \lambda\mathbf{x}F(\mathbf{x})$ is bound. Formulae and species terms which only differ in the naming of their bound variables are identified.

2.4. If T is an $n$ary species term, X an $n$ary species variable and $F(X)$ a formula, then $F(T)$ denotes the formula which is obtained by replacing every part of $F(X)$ of the form $X\mathbf{t}$ for which X is free by $T\mathbf{t}$. Before doing this, however, one may have to rename some bound individual variables in T so that no variable occurrence in $\mathbf{t}$ becomes bound in $T\mathbf{t}$. Likewise, one may have to remane some bound variables in $F(X)$ so that no variable occurrence which is free in T becomes bound in $F(T)$.

3. **Rules of inference**.

3.1. We shall use Prawitz's 1965 system of natural deduction for second order logic in its first version. Thus, deductions are built up from assumptions by means of the following rules of inference.

3.1.1. →introduction.

$$\frac{\begin{matrix} F' \\ \vdots \\ G \end{matrix}}{F \to G}$$

3.1.2. →elimination or modus ponens.

$$\frac{F \to G \quad F}{G}$$

3.1.3. ⋀introduction of first order.

$$\frac{F(x)}{\bigwedge x F(x)}$$

3.1.4. ⋀elimination of first order.

$$\frac{\bigwedge x F(x)}{F(t)}$$

3.1.5. ⋀introduction of second order.

$$\frac{F(X)}{\bigwedge X F(X)}$$

3.1.6. ⋀elimination of second order.

$$\frac{\bigwedge X F(X)}{F(T)}$$

3.2. Free and bound occurrences of a variable in a deduction are defined as in Martin-Löf 1970. Deductions which only differ in the naming of their bound variables are identified. It will be tacitly assumed that bound variables are renamed whenever necessary in order to avoid undesired ties.

3.3. The notions of major premise, minor premise, major deduction, minor deduction, cut, main cut and main branch are defined as in Martin-Löf 1970.

4. Rules of contraction.

4.1. →contraction.

$$\cfrac{\cfrac{\begin{array}{c}\not F\\ \vdots\\ G\end{array}}{F\to G}\qquad \begin{array}{c}\vdots\\ F\end{array}}{G}\quad \text{contr}\quad \begin{array}{c}\vdots\\ F\\ \vdots\\ G\end{array}$$

4.2. Λcontraction of first order.

$$\cfrac{\cfrac{\begin{array}{c}\vdots\\ F(x)\end{array}}{\Lambda x F(x)}}{F(t)}\quad \text{contr}\quad \begin{array}{c}\vdots\\ F(t)\end{array}$$

4.3. Λcontraction of second order

$$\cfrac{\cfrac{\begin{array}{c}\vdots\\ F(X)\end{array}}{\Lambda X F(X)}}{F(T)}\quad \text{contr}\quad \begin{array}{c}\vdots\\ F(T)\end{array}$$

5. The definition of what it means for a deduction to be normalizable can be carried over word for word from Martin-Löf 1970.

6. Computability predicates.

6.1. Let T be a species term. A predicate α_T which is defined for deductions of the form

$$\begin{array}{c}\vdots\\ Tt\end{array}$$

will be called a computability predicate of type T (candidat de réductibilité in Girard's terminology) if it satisfies the following conditions.

6.1.1. A deduction which consists solely of an assumption satisfies α_T.

6.1.2. A deduction which ends with an elimination inference and has a main cut satisfies α_T if and only if the deduction which is obtained from it by eliminating the main cut satisfies α_T.

6.1.3. A deduction which ends with an elimination inference and has a cut free main branch satisfies α_T if and only if the minor deductions of the applications of modus ponens on the main branch are all normalizable.

6.1.4. If a deduction satisfies α_T, then it is normalizable.

6.2. There are plenty of computability predicates. For example, the predicate which holds precisely for the normalizable deductions with end formula of the form $T\mathbf{t}$ is a computability predicate of type T.

7. Let $\mathbf{X} = X_1, \ldots, X_n$ be a sequence of species variables and let $\mathbf{T} = T_1, \ldots, T_n$ and $\alpha_{\mathbf{T}} = \alpha_{T_1}, \ldots, \alpha_{T_n}$ be corresponding sequences of terms and computability predicates, respectively. Then, if $T(\mathbf{X}) = \lambda\mathbf{x}F(\mathbf{x},\mathbf{X})$ is a species term all of whose free species variables occur in the sequence $\mathbf{X}$, we shall introduce a new predicate $\varphi_{T(\mathbf{X})}(\alpha_{\mathbf{T}})$ which is to be defined for deductions of the form

$$\begin{array}{c} \vdots \\ F(\mathbf{t},\mathbf{T}) \end{array}$$

Such a deduction is to satisfy $\varphi_{T(\mathbf{X})}(\alpha_{\mathbf{T}})$ if and only if it satisfies $\varphi_{F(\mathbf{t},\mathbf{T})}(\alpha_{\mathbf{T}})$ and so it suffices to define $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$ for an arbitrary formula $F(\mathbf{X})$. This we do by induction on the number of logical signs in $F(\mathbf{X})$. Basis. $F(\mathbf{X})$ is atomic. If C is a species constant, then a deduction with end formula $C\mathbf{t}$ satisfies $\varphi_{C\mathbf{t}}(\alpha_{\mathbf{T}})$ if and only if it is normalizable. If X is a species variable, then a deduction with end formula $T\mathbf{t}$ satisfies $\varphi_{X\mathbf{t}}(\alpha_T,\alpha_{\mathbf{T}})$ if and only if it satisfies α_T. Induction step. $F(\mathbf{X})$ is composite. Several cases have to be distinguished depending on the last inference of the deduction for which $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$ is to be defined.

7.1. The deduction which consists solely of the assumption $F(\mathbf{T})$ satisfies $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$.

7.2. The last inference of the deduction is an introduction.

7.2.1. →introduction.

$$\frac{\begin{array}{c}\not{F}(\mathbf{T})\\ \vdots \\ G(\mathbf{T})\end{array}}{F(\mathbf{T})\to G(\mathbf{T})}$$

satisfies $\varphi_{F(\mathbf{X})\to G(\mathbf{X})}(\alpha_{\mathbf{T}})$ provided

$$\begin{array}{c}\vdots\\ F(\mathbf{T})\\ \vdots\\ G(\mathbf{T})\end{array}$$

satisfies $\varphi_{G(\mathbf{X})}(\alpha_{\mathbf{T}})$ for all

$$\begin{array}{c}\vdots\\ F(\mathbf{T})\end{array}$$

that satisfy $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$.

7.2.2. Λintroduction of first order.

$$\frac{\begin{array}{c}\vdots\\ F(x,\mathbf{T})\end{array}}{\Lambda x F(x,\mathbf{T})}$$

satisfies $\varphi_{\Lambda x F(x,\mathbf{X})}(\alpha_{\mathbf{T}})$ provided

$$\begin{array}{c} \vdots \\ F(t,\mathbf{T}) \end{array}$$

satisfies $\varphi_{F(t,\mathbf{X})}(\alpha_{\mathbf{T}})$ for all individual terms t.

7.2.3. Λintroduction of second order.

$$\dfrac{\begin{array}{c} \vdots \\ F(X,\mathbf{T}) \end{array}}{\Lambda XF(X,\mathbf{T})}$$

satisfies $\varphi_{\Lambda XF(X,\mathbf{X})}(\alpha_{\mathbf{T}})$ provided

$$\begin{array}{c} \vdots \\ F(T,\mathbf{T}) \end{array}$$

satisfies $\varphi_{F(X,\mathbf{X})}(\alpha_T,\alpha_{\mathbf{T}})$ for all species terms T and computability predicates α_T.

7.3. The last inference of the deduction is an elimination.

7.3.1. The deduction has a main cut. Then it satisfies $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$ provided the deduction which is obtained from it by eliminating the main cut satisfies $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$.

7.3.2. The deduction has a cut free main branch. Then it satisfies $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$ provided the minor deductions of the applications of modus ponens on the main branch are all normalizable.

8. $\varphi_{\mathcal{T}(\mathbf{X})}(\alpha_{\mathbf{T}})$ is a computability predicate of type $\mathcal{T}(\mathbf{T})$.

8.1. That $\varphi_{T(\mathbf{X})}(\alpha_{\mathbf{T}})$ satisfies 6.11, 6.12 and 6.13 follows immediately from the definition. To verify 6.14 it clearly suffices to show for an arbitrary formula $F(\mathbf{X})$ that if a deduction satisfies $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$ then it is normalizable. This we do by induction on the number of logical signs in $F(\mathbf{X})$. Basis. Immediate from the definition. Induction step. If $F(\mathbf{X})$ is composite, then a deduction satisfies $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$ if and only if, by a finite number of eliminations of main cuts, it reduces to a deduction which consists solely of an assumption, has a cut free main branch with normalizable minor deductions or else ends with an introduction inference. In the first two cases we are done immediately and in the third case we have to consider separately each possible form of the introduction inference.

8.1.1. $\rightarrow$introduction. If

$$\begin{array}{c} \not{F}(\not{\mathbf{T}}) \\ \vdots \\ \dfrac{G(\mathbf{T})}{F(\mathbf{T}) \rightarrow G(\mathbf{T})} \end{array}$$

satisfies $\varphi_{F(\mathbf{X})\rightarrow G(\mathbf{X})}(\alpha_{\mathbf{T}})$, then

$$\begin{array}{c} F(\mathbf{T}) \\ \vdots \\ G(\mathbf{T}) \end{array}$$

satisfies $\varphi_{G(\mathbf{X})}(\alpha_{\mathbf{T}})$. By induction hypothesis, the latter deduction is normalizable, and, consequently, so is the former.

8.1.2. Λ introduction of first order. If

$$\begin{array}{c} \vdots \\ \dfrac{F(x,\mathbf{T})}{\Lambda x F(x,\mathbf{T})} \end{array}$$

satisfies $\varphi_{\Lambda x F(x,\mathbf{X})}(\alpha_{\mathbf{T}})$, then

$$\begin{array}{c} \vdots \\ F(x,\mathbf{T}) \end{array}$$

satisfies $\varphi_{F(x,\mathbf{X})}(\alpha_{\mathbf{T}})$. By induction hypothesis, the latter deduction is normalizable and, consequently, so is the former.

8.1.3. Λ introduction of second order. If

$$\begin{array}{c} \vdots \\ \dfrac{F(X,\mathbf{T})}{\Lambda X F(X,\mathbf{T})} \end{array}$$

satisfies $\varphi_{\Lambda XF(X,\mathbf{X})}(\alpha_{\mathbf{T}})$, then

$$\begin{array}{c} \vdots \\ F(X,\mathbf{T}) \end{array}$$

satisfies $\varphi_{F(X,\mathbf{X})}(\alpha_X,\alpha_{\mathbf{T}})$ for all α_X. By induction hypothesis, the latter deduction is normalizable and, consequently, so is the former.

9. Consider a deduction

$$\begin{array}{ccc} F_1(\mathbf{X}) & \dots & F_n(\mathbf{X}) \\ & \ddots \quad \therefore & \\ & F(\mathbf{X}) & \end{array}$$

whose free species variables form the sequence $\mathbf{X}$ and whose assumptions are $F_1(\mathbf{X}), \dots, F_n(\mathbf{X})$. Then, for all individual terms that we substitute for its free individual variables and for all sequences of species terms $\mathbf{T}$ that we substitute for $\mathbf{X}$ and for all $\alpha_{\mathbf{T}}$, if the deductions

$$\begin{array}{ccc} \vdots & & \vdots \\ F_1(\mathbf{T}) & \dots & F_n(\mathbf{T}) \end{array}$$

satisfy $\varphi_{F_1(\mathbf{X})}(\alpha_{\mathbf{T}}), \dots, \varphi_{F_n(\mathbf{X})}(\alpha_{\mathbf{T}})$, respectively, then

$$\begin{array}{ccc} \vdots & & \vdots \\ F_1(\mathbf{T}) & \dots & F_n(\mathbf{T}) \\ & \ddots \quad \therefore & \\ & F(\mathbf{T}) & \end{array}$$

satisfies $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$. The proof is by induction on the length of the deduction. Several cases have to be distinguished depending on how the end formula of the deduction has been inferred.

9.1. The deduction consists solely of an assumption. Trivial.

9.2. The last inference is an introduction.

9.2.1. $\rightarrow$introduction. We have to show that a deduction of the form

$$\frac{\begin{array}{c}\not{F}(\not{\mathbf{T}})\\ \vdots \\ G(\mathbf{T})\end{array}}{F(\mathbf{T}) \rightarrow G(\mathbf{T})}$$

satisfies $\varphi_{F(\mathbf{X})\rightarrow G(\mathbf{X})}(\alpha_{\mathbf{T}})$. By 7.2.1 this is so if

$$\begin{array}{c}\vdots \\ F(\mathbf{T}) \\ \vdots \\ G(\mathbf{T})\end{array}$$

satisfies $\varphi_{G(\mathbf{X})}(\alpha_{\mathbf{T}})$ for all

$$\begin{array}{c}\vdots \\ F(\mathbf{T})\end{array}$$

that satisfy $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$ which follows immediately from the induction hypothesis.

9.2.2. Λintroduction of first order. We have to show that a deduction of the form

$$\frac{\begin{array}{c}\vdots \\ F(x,\mathbf{T})\end{array}}{\Lambda x F(x,\mathbf{T})}$$

satisfies $\varphi_{\wedge xF(x,\mathbf{X})}(\alpha_{\mathbf{T}})$. By 7.2.2 this is so if

$$\begin{array}{c} \vdots \\ F(t,\mathbf{T}) \end{array}$$

satisfies $\varphi_{F(t,\mathbf{X})}(\alpha_{\mathbf{T}})$ for all individual terms t which follows immediately from the induction hypothesis.

9.2.3. $\wedge$introduction of second order. We have to show that a deduction of the form

$$\cfrac{\begin{array}{c} \vdots \\ F(X,\mathbf{T}) \end{array}}{\wedge XF(X,\mathbf{T})}$$

satisfies $\varphi_{\wedge XF(X,\mathbf{X})}(\alpha_{\mathbf{T}})$. By 7.2.3 this is so if

$$\begin{array}{c} \vdots \\ F(T,\mathbf{T}) \end{array}$$

satisfies $\varphi_{F(X,\mathbf{X})}(\alpha_T,\alpha_{\mathbf{T}})$ for all T and α_T which follows immediately from the induction hypothesis.

9.3. The last inference of the deduction is an elimination.

9.3.1. $\rightarrow$elimination. We have to show that a deduction of the form

$$\cfrac{\begin{array}{c} \vdots \\ F(\mathbf{T}) \rightarrow G(\mathbf{T}) \end{array} \quad \begin{array}{c} \vdots \\ F(\mathbf{T}) \end{array}}{G(\mathbf{T})}$$

satisfies $\varphi_{G(\mathbf{X})}(\alpha_{\mathbf{T}})$. By induction hypothesis

$$\begin{array}{c} \vdots \\ F(\mathbf{T}) \rightarrow G(\mathbf{T}) \end{array}$$

satisfies $\varphi_{F(\mathbf{X})\to G(\mathbf{X})}(\alpha_{\mathbf{T}})$. Consequently, by a finite number of eliminations of main cuts, it reduces to a deduction which consists solely of an assumption, has a cut free main branch with normalizable minor deductions or else ends with an introduction inference. In the first two cases we are done since by induction hypothesis.

$$\begin{array}{c}\vdots\\ F(\mathbf{T})\end{array}$$

satisfies $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$ and is a fortiori normalizable. In the third case we know that

$$\dfrac{\begin{array}{c}\not{F}(\mathbf{T})\\ \vdots\\ G(\mathbf{T})\end{array}}{F(\mathbf{T})\to G(\mathbf{T})} \qquad \begin{array}{c}\vdots\\ F(\mathbf{T})\end{array}$$

satisfy $\varphi_{F(\mathbf{X})\to G(\mathbf{X})}(\alpha_{\mathbf{T}})$ and $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{T}})$, respectively. By 7.2.1

$$\begin{array}{c}\vdots\\ F(\mathbf{T})\\ \vdots\\ G(\mathbf{T})\end{array}$$

satisfies $\varphi_{G(\mathbf{X})}(\alpha_{\mathbf{T}})$ and, consequently, so does

$$\dfrac{\dfrac{\begin{array}{c}\not{F}(\mathbf{T})\\ \vdots\\ G(\mathbf{T})\end{array}}{F(\mathbf{T})\to G(\mathbf{T})} \quad \begin{array}{c}\vdots\\ F(\mathbf{T})\end{array}}{G(\mathbf{T})}$$

which was to be proved.

9.3.2. Λ elimination of first order. We have to show that a deduction of the form

$$\begin{array}{c}\vdots\\ \dfrac{\Lambda x F(x,\mathbf{T})}{F(t,\mathbf{T})}\end{array}$$

satisfies $\varphi_{F(t,\mathbf{X})}(\alpha_{\mathbf{T}})$. By induction hypothesis

$$\begin{array}{c}\vdots\\ \Lambda x F(x,\mathbf{T})\end{array}$$

satisfies $\varphi_{\Lambda x F(x,\mathbf{X})}(\alpha_{\mathbf{T}})$. Consequently, by a finite number of eliminations of main cuts, it reduces to a deduction which consists solely of an assumption, has a cut free main branch with normalizable minor deductions or else ends with an introduction inference. In the first two cases we are done and in the third case we know that

$$\begin{array}{c}\vdots\\ \dfrac{F(x,\mathbf{T})}{\Lambda x F(x,\mathbf{T})}\end{array}$$

satisfies $\varphi_{\Lambda x F(x,\mathbf{X})}(\alpha_{\mathbf{T}})$. By 7.2.2

$$\begin{array}{c}\vdots\\ F(t,\mathbf{T})\end{array}$$

satisfies $\varphi_{F(t,\mathbf{X})}(\alpha_{\mathbf{T}})$ and, consequently, so does

$$\begin{array}{c}\vdots\\ \dfrac{\dfrac{F(x,\mathbf{T})}{\Lambda x F(x,\mathbf{T})}}{F(t,\mathbf{T})}\end{array}$$

which was to be proved.

9.3.3. Λelimination of second order. We have to show that a deduction of the form

$$\begin{array}{c} \vdots \\ \dfrac{\Lambda X F(X,\mathbf{T})}{F(T(\mathbf{T}),\mathbf{T})} \end{array}$$

satisfies $\varphi_{F(T(\mathbf{X}),\mathbf{X})}(\alpha_{\mathbf{T}})$. By induction hypothesis

$$\begin{array}{c} \vdots \\ \Lambda X F(X,\mathbf{T}) \end{array}$$

satisfies $\varphi_{\Lambda X F(X,\mathbf{X})}(\alpha_{\mathbf{T}})$. Consequently, by a finite number of eliminations of main cuts, it reduces to a deduction which consists solely of an assumption, has a cut free main branch with normalizable minor deductions or else ends with an introduction inference. In the first two cases we are done and in the third case we know that

$$\begin{array}{c} \vdots \\ \dfrac{F(X,\mathbf{T})}{\Lambda X F(X,\mathbf{T})} \end{array}$$

satisfies $\varphi_{\Lambda X F(X,\mathbf{X})}(\alpha_{\mathbf{T}})$. By 7.2.3

$$\begin{array}{c} \vdots \\ F(T(\mathbf{T}),\mathbf{T}) \end{array}$$

satisfies $\varphi_{F(X,\mathbf{X})}(\alpha_{T(\mathbf{T})},\alpha_{\mathbf{T}})$ for all computability predicates $\alpha_{T(\mathbf{T})}$ and, consequently, so does

$$\begin{array}{c} \vdots \\ \dfrac{\dfrac{F(X,\mathbf{T})}{\Lambda X F(X,\mathbf{T})}}{F(T(\mathbf{T}),\mathbf{T})} \end{array}$$

It now only remains to put $\alpha_{T(\mathbf{T})} = \varphi_{T(\mathbf{X})}(\alpha_{\mathbf{T}})$ and use the subsitution property

$$\varphi_{F(X,\mathbf{X})}(\varphi_{T(\mathbf{X})}(\alpha_{\mathbf{T}}),\alpha_{\mathbf{T}}) = \varphi_{F(T(\mathbf{X}),\mathbf{X})}(\alpha_{\mathbf{T}})$$

which is immediately verified by induction on the number of logical signs in the formula $F(X,\mathbf{X})$.

10. **Hauptsatz**. *Every deduction is normalizable.*

10.1. Given an arbitrary deduction, let $\mathbf{X}$ be the totality of its free species variables and $F(\mathbf{X})$ its end formula. From what we have just proved it follows that the deduction satisfies $\varphi_{F(\mathbf{X})}(\alpha_{\mathbf{X}})$ whatever be the choice of the sequence of computability predicates $\alpha_{\mathbf{X}}$. In particular, it is normalizable.

10.2. Inspection of the proof of Hauptsatz shows that for every specific deduction the proof that it is normalizable can be formalized in the theory of species itself.

11. The following three corollaries which follow from Hauptsatz by combinatorial reasoning are due to Prawitz 1968 who derived them from his completeness theorem for the cut free sequent calculus which corresponds to the system of natural deduction we are considering. We provide the proofs since they are somewhat different for the system of natural deduction.

11.1. *From a proof of $F \vee G$ we can find a proof of either F or G.*

11.1.1. The normalized proof of $F \vee G = \Lambda Y((F\to Y) \to ((G\to Y)\to Y))$ must have one of the forms

$$\begin{array}{c} [F\to Y] \quad [G\to Y] \\ \vdots \\ \dfrac{\dfrac{\dfrac{\dfrac{[F\to Y] \quad F}{Y}}{(G\to Y)\to Y}}{(F\to Y)\to((G\to Y)\to Y)}}{F\vee G} \end{array} \qquad \begin{array}{c} [F\to Y] \quad [G\to Y] \\ \vdots \\ \dfrac{\dfrac{\dfrac{\dfrac{[G\to Y] \quad G}{Y}}{(G\to Y)\to Y}}{(F\to Y)\to((G\to Y)\to Y)}}{F\vee G} \end{array}$$

The left proof contains a deduction of F from the assumptions $F \to Y$ and $G \to Y$. By substituting $F \vee G$ for Y we obtain a deduction of F from the assumptions $F \to (F \vee G)$ and $G \to (F \vee G)$. Attaching proofs of $F \to (F \vee G)$ and $G \to (F \vee G)$ to this deduction we obtain the desired proof of F. If the proof of $F \vee G$ has the form pictured to the right, we obtain a proof of G by the same argument.

11.2. *From a proof of* $\bigvee xF(x)$ *we can find an individual term* t *and a proof of* $F(t)$.

11.2.1. The normalized proof of $\bigvee xF(x) = \bigwedge Y(\bigwedge x(F(x) \to Y) \to Y)$ must have the form

$$\dfrac{\dfrac{\dfrac{\dfrac{\cancel{\bigwedge x(F(x)\to Y)}}{F(t)\to Y} \qquad \begin{matrix}\cancel{\bigwedge x(F(x)\to Y)}\\ \vdots \\ F(t)\end{matrix}}{Y}}{\bigwedge x(F(x)\to Y)\to Y}}{\bigvee xF(x)}$$

Take the subdeduction of $F(t)$ from the assumption $\bigwedge x(F(x) \to Y)$, substitute $\bigvee xF(x)$ for Y and attach a proof of $\bigwedge x(F(x) \to \bigvee xF(x))$. We then get the desired proof of $F(t)$.

11.3. *From a proof of* $\bigvee XF(X)$ *we can find a species term* T *and a proof of* $F(T)$.

11.3.1. The normalized proof of $\bigvee XF(X) = \bigwedge Y(\bigwedge X(F(X) \to Y) \to Y)$ must have the form

$$\dfrac{\dfrac{\dfrac{\dfrac{\cancel{\bigwedge X(F(X)\to Y)}}{F(U(Y))\to Y} \qquad \begin{matrix}\cancel{\bigwedge X(F(X)\to Y)}\\ \vdots \\ F(U(Y))\end{matrix}}{Y}}{\bigwedge X(F(X)\to Y)\to Y}}{\bigvee XF(X)}$$

Take the subdeduction of $F(U(Y))$ from $\bigwedge X(F(X) \to Y)$, substitute $\bigvee XF(X)$ for Y and attach a proof of $\bigwedge X(F(X) \to \bigvee XF(X))$. We then get a proof of $F(T)$ where $T = U(\bigvee XF(X))$.

12. The extension of the treatment of second order logic given in the present paper to the full theory of types is rather straightforward and will be published elsewhere.

References

J.Y.Girard, Une extension de l'interpretation de Gödel a l'analyse, et son application a l'elimination des coupures dans l'analyse et la theorie des types (1970). This volume.

P.Martin-Löf, Hauptsatz for the intuitionistic theory of iterated inductive definitions (1970). This volume.

D.Prawitz, Natural deduction (Almqvist and Wiksell, Stockholm, 1965), Some results for intuitionistic logic with second order quantification rules, in: Intuitionism and Proof Theory, eds. A.Kino, J.Myhill and R.E.Vesley (North-Holland, Amsterdam, 1970).

W.W.Tait, Intentional interpretations of functionals of finite type, J. Symbolic Logic 32 (1967) 198–212.