A UNIFYING PRINCIPLE FOR EXTENSIONAL HIGHER–ORDER LOGIC

by

Michael Kohlhase
Fachbereich Informatik, Unisversität Saarbrücken
Im Stadtwald, D–6600 Saarbrücken 11, Germany

currently at

Department of Mathematics
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.

Research Report No. 93–153
January, 1993

A Unifying Principle for Extensional Higher-Order Logic

Michael Kohlhase*
Fachbereich Informatik, Universität Saarbrücken
Im Stadtwald, D-6600 Saarbrücken 11, Germany
kohlhase@cs.uni-sb.de
currently at
Dept. of Mathematics
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
kohlhase@cs.cmu.edu

Research Report No. 93-153
Department of Mathematics
Carnegie Mellon University

February 1993

abstract

In higher-order automated theorem proving the extensionality of equality has not yet been satisfactorily dealt with. This situation is especially unsatisfying since higher-order logic without extensionality does not admit a natural set-theoretic semantics.

In this paper we present a version of Smullyan's Unifying Principle for simple type theory with extensionality. This result can serve as a tool for the development of extensionally complete calculi that are well-suited for mechanization on computers, and furthermore enables us to give an elegant completeness proof for Henkin's original calculus.

*This work was supported by the "Sonderforschungsbereich 314, Künstliche Intelligenz" of the Deutsche Forschungsgemeinschaft (DFG) and the "Studienstiftung des deutschen Volkes".

Contents

1 Introduction **2**

2 Simple Type Theory **4**
2.1 Syntax of $\mathcal{Q}$ . . . 4
2.2 Classical Formulations of Type Theory . . . 5
2.3 Substitutions, Lambda-Conversion and Normal Forms . . . 6

3 Semantics of $\mathcal{Q}$ **8**
3.1 Σ-Algebras . . . 8
3.2 Σ-Congruences . . . 10
3.3 General Models . . . 11

4 Calculi for $\mathcal{Q}$ **13**
4.1 Calculi, Derivations and Consistency . . . 13
4.2 The Hilbert Calculi $\mathfrak{T}$ and $\mathfrak{TE}$ . . . 15

5 Unifying Principle for General Models **17**
5.1 Abstract Consistency Classes . . . 17
5.2 Higher-Order Hintikka Sets . . . 19
5.3 Unifying Principle for General Models . . . 21
5.4 Completeness . . . 23

6 Conclusion and Further Work **25**

7 The Index of defined symbols **29**

A A Resolution Calculus for $\mathcal{Q}$ **30**
A.1 Resolution in Extensional Type Theory . . . 31

1 Introduction

One key objective of the field of higher-order automated theorem proving is finding calculi for higher-order logic that are well-suited for mechanization on computers. But higher-order logic is known to be incomplete with respect to the standard model semantics, so in particular it does not admit calculi that are complete with respect to this class of models. Fortunately there is another notion of semantics for higher-order logic based on a class of simple set-theoretic models, called general models or Henkin-models, that allows complete calculi (cf. [Hen50, And72]). Therefore the notion of completeness with respect to general models is natural for measuring the deductive power of calculi. So far all efforts to find machine-oriented calculi for higher-order logic that are complete with respect to the general model semantics have failed. In particular all such calculi known to the author (cf. Huet's constrained resolution calculus [Hue72, Hue73, Koh93], Andrews' resolution in type theory [And71], Miller's expansion proofs [Mil83] and the TPS procedure [ALCMP84, And89]) are not complete in this sense. But there exists even another notion of completeness in which all of the above calculi are complete: the notion of completeness relative to certain Hilbert style calculus $\mathfrak{T}$. In general a calculus $\mathcal{C}$ is complete relative to a calculus $\mathcal{T}$, iff $\mathcal{C}$ proves all theorems of $\mathcal{T}$. It is rooted in the tradition of proof theory, where questions of relative deductive power of calculi are analyzed without referring to model-theoretic arguments. However it turns out that $\mathfrak{T}$ is not complete with respect to general models, so completeness relative to $\mathfrak{T}$ is a strictly weaker notion than general model completeness. One Approach to the task of finding machine-oriented calculi in higher-order logic consists in analyzing and overcoming the shortcomings of the existing, incomplete calculi.

Unifying Principles have become a standard tool for proving the completeness of calculi for automated theorem proving (see for instance the introductory textbooks [And86, Fit90]). A unifying principle for a logical system $\mathcal{L}$ is a theorem of the form: If a set of sentences Φ in $\mathcal{L}$ is a member of an abstract consistency class Γ, then there exists an $\mathcal{L}$-model for Φ. Thus if we want to show the completeness of a particular calculus $\mathcal{C}$, we first prove that the class Γ of sets of sentences Φ that are $\mathcal{C}$-consistent (cannot be refuted in $\mathcal{C}$) is an abstract consistency class, then the unifying principle tells us that $\mathcal{C}$-consistent sets of sentences are satisfiable in $\mathcal{L}$. Now we assume that a sentence $\mathbf{A}$ is valid in $\mathcal{L}$, so $\neg\mathbf{A}$ does not have an $\mathcal{L}$-model and is therefore $\mathcal{C}$-inconsistent. From this it is for most calculi easy to verify that $\mathbf{A}$ is a theorem of $\mathcal{C}$. Note that with this argumentation the completeness proof for $\mathcal{C}$ condenses to verifying that Γ is an abstract consistency class, a task that does not refer to $\mathcal{L}$-models. Thus the usefulness of unifying principles comes from the fact that a unifying principle abstracts away all the model-theoretic analysis in completeness proofs by providing a sufficient set of proof-theoretic conditions (membership in Γ) for a calculus to be complete. In this respect a unifying principle is similar to a Herbrand Theorem, but it is easier to generalize to other logic systems like higher-order logic. The technique was developed for first-order logic by J. Hintikka and R. Smullyan [Hin55, Smu63, Smu68].

In [And71] P. Andrews presented a unifying principle for simple type theory, on which the relative completeness results of the above calculi are based. It has the form of a relative consistency theorem: If $\Phi \in \Gamma$, then Φ is consistent with respect to $\mathfrak{T}$. Therefore this unifying principle can only lead to the notion of relative completeness introduced above. But since relative completeness is a weaker notion than general model completeness, it would be desirable to have a unifying principle for general models to serve as a basis for future developments of machine-oriented calculi that are complete with respect to the general model semantics.

In [Hen50] L. Henkin presented a Hilbert-style calculus $\mathfrak{TE}$ for simple type theory, which is complete with respect to general models. The difference between the Hilbert-style calculus $\mathfrak{T}$ used by P. Andrews and $\mathfrak{TE}$ are the extensionality axioms in the latter. So the basic idea behind the extension of the results in [And71] is to add the proper treatment of extensionality, which we have undertaken in this paper.

The main theorem of this report is a unifying principle for general models:

Theorem 5.3.6 (Unifying Principle for General Models) *If* Γ *is an abstract consistency class and* $H \in \Gamma$ *is sufficiently pure, then* H *has a countable general model.*

Here the definition of abstract consistency class differs from that in [And71] in that we require three additional clauses that formalize the extensionality of equality and the number of truth values. The proof of the main theorem proceeds by showing that any member Φ of an abstract consistency class Γ can be extended to a higher-order Hintikka set $\mathcal{H}$ that contains $\mathcal{D}$. From $\mathcal{H}$ we can read off an congruence relation $\sim_{\mathcal{H}}$ on the term algebra $cwff(\Sigma)$ that is extensional, since Γ has provisions for extensionality. It is then simple to verify that the quotient algebra $cwff(\Sigma)/_{\sim_{\mathcal{H}}}$ is a general model for $\mathcal{H}$ and therefore for H.

The rest of this paper is structured as follows, we will first review simple type theory and give a careful exhibition of its semantics and calculi (Sections 2 to 4). In Section 5 we will prove the unifying principle for extensional type theory and relate it to that of type theory without extensionality. Finally we will use the unifying principle to give a short completeness proof for a variant of Henkin's calculus 5.4. In Section 6 we conclude by sketching a couple of possible applications of the unifying principle. Finally we present some ideas for a resolution calculus that is complete with respect to general models in appendix A.

The author would like to thank Peter Andrews and Frank Pfenning for stimulating discussions.

2 Simple Type Theory

In this chapter we will give a formulation of simple type theory, which we will call $\mathcal{Q}$. This system only differs from Church's formulation [Chu40] in the use of Henkin's general model semantics [Hen50] and the use of equality as the primitive notion (introduced by P. Andrews, cf. [And86]). Furthermore $\mathcal{Q}$ differs from all these systems by not assuming the existence of description functions.

2.1 Syntax of $\mathcal{Q}$

Definition 2.1.1 (Type Symbols) The set $\mathcal{T}$ of **type symbols** is inductively defined by

1. ι is a type symbol (denoting the type of individuals).

2. o is a type symbol (denoting the type of truth values).

3. If α and β are type symbols, then $(\alpha\beta)$ is a type symbol (denoting the type of functions with domain α and codomain β).

The type symbols in $\mathcal{T}_0 := \{o, \iota\} \subseteq \mathcal{T}$ are called **base type symbols**. We will often use the word "type" as an abbreviation for "type symbol".

Notation 2.1.2 As syntactic variables for type symbols we use lower case Greek letters. We use the convention of association to the left for omitting parentheses in type symbols, thus $\alpha\beta\gamma$ is an abbreviation for $((\alpha\beta)\gamma)$. This way the type symbol $(\alpha\beta_1 \ldots \beta_n)$ denotes the type of n-ary functions, which take n arguments of the types $\beta_1, \ldots, \beta_n$ and have the range in the type α.

Remark 2.1.3 In $\mathcal{Q}$ it will be possible to view unary functions as elementary objects, because n-ary functions can be reduced to unary ones by currying (cf. 3.1.2). Thus for simplicity of the presentation of the logic, we will restrict ourselves to the unary case. Note that with the use of conventions for eliminating parentheses in type symbols discussed above, we will regain the appearance of the general case.

General Assumption 2.1.4 For every type symbol $\alpha \in \mathcal{T}$ we assume the existence of a nonempty set Σ_α of **constant symbols of type** α and a countably infinite set $\mathcal{V}_\alpha$ of **variable symbols of type** α.

Furthermore we assume for each $\alpha \in \mathcal{T}$ the existence of **logical constants** $q_{o\alpha\alpha} \in \Sigma_{o\alpha\alpha}$. All other constant symbols are called **parameters**. The sets $\mathcal{V} := \mathcal{V}_\mathcal{T} := \bigcup_{\alpha\in\mathcal{T}} \mathcal{V}_\alpha$ and $\Sigma := \Sigma_\mathcal{T} := \bigcup_{\alpha\in\mathcal{T}} \Sigma_\alpha$ are called the **set of variables** and the **signature**, respectively.

Definition 2.1.5 The **primitive symbols** of $\mathcal{Q}$ consist of the improper symbols λ, [,], the variable symbols and the constant symbols.

For each $\alpha \in \mathcal{T}$ we define the set $\mathit{wff}_\alpha(\Sigma)$ of **well-formed formulae of type** α inductively by

1. If $\mathbf{A}_\alpha$ is a variable or constant of type α, then $\mathbf{A}_\alpha \in \mathit{wff}_\alpha(\Sigma)$.

2. If $\mathbf{A}_{\alpha\beta} \in \mathit{wff}_{\alpha\beta}(\Sigma)$ and $\mathbf{B}_\beta \in \mathit{wff}_\beta(\Sigma)$, then $[\mathbf{A}_{\alpha\beta}\mathbf{B}_\beta] \in \mathit{wff}_\alpha(\Sigma)$.

3. If $\mathbf{A}_\alpha \in \mathit{wff}_\alpha(\Sigma)$ and $X_\beta \in \mathcal{V}_\beta$, then $[\lambda X_\beta.\mathbf{A}_\alpha] \in \mathit{wff}_{\alpha\beta}(\Sigma)$.

Obviously each choice of parameters determines a particular set of well-formed formulae, therefore we speak of a formulation $\mathcal{Q}(\Sigma)$ of $\mathcal{Q}$. Note that the formulation of $\mathcal{Q}$ does not depend on the choice of the set of variables, because all $\mathcal{V}_\alpha$ are assumed to be countably infinite, thus each choice will give rise to an isomorphic set of well-formed formulae.

Notation 2.1.6 We will denote the constants by lower case letters and the variables by upper case letters. We will use bold upper case letters $\mathbf{A}_\alpha$, $\mathbf{B}_{\alpha\beta}$, $\mathbf{C}_\gamma \ldots$ as syntactical variables for well-formed formulae. The type of an object will be denoted as a subscript, if it is not irrelevant or clear from the context.

Definition 2.1.7 A variable X_α is called **free in** a well-formed formula $\mathbf{A}_\beta$, iff it is not in a well-formed part of the form $[\lambda X.\mathbf{B}]$ in $\mathbf{A}$, otherwise it is **bound in A**. The respective sets of variables are denoted by $\mathbf{Var}(\mathbf{A})$, $\mathbf{Free}(\mathbf{A})$ and $\mathbf{Bound}(\mathbf{A})$ for a well-formed formula $\mathbf{A}$. A well-formed formula is called **closed**, if it does not contain free variables. We denote the set of closed well-formed formulae of type α by $\mathit{cwff}_\alpha(\Sigma)$. A well-formed formula of type o is called a **proposition** and a closed proposition a **sentence**.

Notation 2.1.8 In order to make the notation of well-formed formulae more legible, we use the convention that the group brackets $[\cdot]$ associate to the left and that the square dot . denotes a left bracket, whose mate is as far right as consistent with the brackets already present. Additionally we combine successive λ-abstractions, so that the well-formed formula $[\lambda X^1.\lambda X^2.\ldots\lambda X^n.\mathbf{A}\mathbf{E}^1\ldots\mathbf{E}^m]$ becomes $[\lambda X^1,\ldots,X^n.\mathbf{A}\mathbf{E}^1\ldots\mathbf{E}^m]$. Furthermore we shorten the expression to $[\lambda\overline{X^n}.\mathbf{A}\overline{\mathbf{E}^m}]$ by a kind of vector notation, where $\overline{X^k}$ means $X^1,\ldots,X^k$.

To avoid confusion with equality in the logic we will denote the meta-logical relation of syntactic equality of well-formed formulae by $\doteq$ and in definitions $:\doteq$.

2.2 Classical Formulations of Type Theory

In the later parts of the paper we will investigate formulations of type theories based on the more conventional concept of logical connectives and quantifiers. The formulation $\mathcal{Q}$ of simple type theory subsumes these, since we can define the classical logical constants (like $\neg, \vee, \forall$ as abbreviations in $\mathcal{Q}$. On the other hand, if we take a formulation of type theory, where the connectives are primitive, the equality constant is definable by the property of indiscernability.

Therefore the "style" of type theory only depends on the primitive constants present in the signature. Thus in the assumption 2.1.4 we could also have assumed the existence of a sufficient subset of the connectives, e.g. $\{\vee_{ooo}, \neg_{oo}\}$ and the quantor $\Pi_{o(o\alpha)}$ and have defined all other connectives and equality from that. The critical part in this choice is that for the semantics we have to require that there exists the identity relation on each type in each model (see [And72]). So it seems more natural to treat equality as primitive in this and the following chapter, whereas in the calculi in chapters 4 and 5 equality only plays a minor role and we assume it to be an abbreviation only.

Definition 2.2.1 (Connectives and Quantors)

The formula	stands for
$[\mathbf{A}_\alpha = \mathbf{B}_\alpha]$	$[q_{o\alpha\alpha}\mathbf{A}_\alpha\mathbf{B}_\alpha]$
$\top_o$	$[q_{ooo} = q_{ooo}]$
$\bot_o$	$[\lambda X_o.\top_o] = [\lambda X_o.X_o]$
$\Pi_{o(o\alpha)}$	$[q_{o(o\alpha)(o\alpha)}[\lambda X_\alpha.\top_o]]$
$\forall X_\alpha \mathbf{A}_o$	$[\Pi_{o(o\alpha)}[\lambda X_\alpha.\mathbf{A}_o]]$
$\wedge_{ooo}$	$[\lambda X_o Y_o.[\lambda G_{ooo}.G_{ooo}\top_o\top_o] = [\lambda G_{ooo}.G_{ooo}X_oY_o]]$
$[\mathbf{A} \wedge \mathbf{B}]$	$[\wedge \mathbf{A}\mathbf{B}]$
$\Rightarrow_{ooo}$	$[\lambda X_o Y_o.X_o = .X_o \wedge Y_o]$
$[\mathbf{A} \Rightarrow \mathbf{B}]$	$[\Rightarrow_{ooo} \mathbf{A}_o\mathbf{B}_o]$
$[\mathbf{A} \Leftrightarrow \mathbf{B}]$	$[\mathbf{A} \Rightarrow \mathbf{B}] \wedge .\mathbf{B} \Rightarrow \mathbf{A}$
$\neg_{oo}$	$[q_{ooo}\bot_o]$
$\vee_{ooo}$	$[\lambda X_o Y_o.\neg.[\neg X_o] \wedge [\neg Y_o]]$
$[\mathbf{A} \vee \mathbf{B}]$	$[\vee \mathbf{A}\mathbf{B}]$
$[\exists X_\alpha \mathbf{A}_o]$	$[\neg\forall X_\alpha.\neg\mathbf{A}_o]$
$[\mathbf{A}_\alpha \neq \mathbf{B}_\alpha]$	$[\neg.\mathbf{A}_\alpha = \mathbf{B}_\alpha]$

We also call the newly defined constants $\top_o, \bot_o, \neg_{oo}, \Pi_{o(o\alpha)}, \wedge_{ooo}, \vee_{ooo}, \Rightarrow_{ooo}, \Leftrightarrow_{ooo}$ **logical constants.** We call them **logical connectives** and **quantors** in order to distinguish them from the equality constant $q_{o\alpha\alpha}$.

In order to see that these definitions indeed yield the well-known connectives and quantors we refer to lemma 3.3.4.

Definition 2.2.2 (Leibniz' Formulation for Equality (Indiscernability))

$$\mathbf{Q}_{o\alpha\alpha} :\doteq [\lambda X_\alpha Y_\alpha.\forall P_{o\alpha}.PX \Rightarrow PY]$$

This formula instantiates to $[\mathbf{A}_\alpha = \mathbf{B}_\alpha] :\doteq [\forall P_{o\alpha}.P\mathbf{A} \Rightarrow P\mathbf{B}]$, which can be read as: formulae **A** and **B** are not equal, iff there exists a discerning property P. We will justify this definition in the semantics by 3.3.5.

2.3 Substitutions, Lambda-Conversion and Normal Forms

In this section we will introduce the closely related notions of λ-conversion and substitutions. The λ-conversion relations establish certain well-formed formulae as functions, by giving interpretations to function application and function equality.

Definition 2.3.1 (Substitutable) A well-formed formula $\mathbf{A}_\alpha$ is called **substitutable for** X_α **in** $\mathbf{B}_\beta$, iff the following condition holds: If $Y \in \mathbf{Free}(\mathbf{A})$, then X is not free in any well-formed part $[\lambda Y.\mathbf{C}]$ of **B**.

It is easy to see, that if **A** is not substitutable for X in **B** and **B**$'$ is obtained from **B** by replacing all free occurrences of X with **A**, then **B**$'$ has bound occurrences of variables Y, that were free occurrences in **A**. We call this situation **variable capture.** Thus the condition defined above is sufficient for avoiding variable capture in substitutions.

Definition 2.3.2 (Substitution) A **substitution** is a map $\sigma: \mathcal{V} \longrightarrow \mathit{wff}(\Sigma)$ with finite **support** $\mathrm{supp}(\sigma) := \{X \in \mathcal{V} \mid \sigma(X) \not\equiv X\}$, such that $\sigma(X_\alpha) \in \mathit{wff}_\alpha(\Sigma)$. We will write a substitution σ with $\mathrm{supp}(\sigma) = \{X^1, \ldots, X^n\}$ as $\{X^1 \mapsto \sigma(X^1), \ldots, X^n \mapsto \sigma(X^n)\}$.

Remark 2.3.3 Since every substitution can be extended to a homomorphism $\sigma: \mathit{wff}(\Sigma) \longrightarrow \mathit{wff}(\Sigma)$ (cf. 3.1.11), we will always think of a substitution as a homomorphism. The set of substitutions is denoted by $\mathbf{SUB}(\Sigma)$.

Definition 2.3.4 (α-Conversion) If Y does not occur free in $\mathbf{C}$, then $[\lambda X.\mathbf{C}] =_\alpha [\lambda Y.\sigma(\mathbf{C})]$, where $\sigma = \{X \mapsto Y\}$. The relation $=_\alpha$ is called **α-conversion** and $\mathbf{A}$ is called an **alphabethical variant** of $\mathbf{B}$, if $\mathbf{A} =_\alpha \mathbf{B}$.

General Assumption 2.3.5 We assume α-equality to be built into the system. That is, we regard well-formed formulae as syntactically equal, iff they are alphabetical variants. With this assumption we have to assume the process of applying substitutions to change the names of bound variables variables, so that no variables are captured in the process. For a more rigorous approach we could have used DeBruijns indices [dB72], and all results would still hold.

Definition 2.3.6 (β-Reduction) We define the β-reduction relation, that we consider fundamental to type theory. We say that $\mathbf{B} \in \mathit{wff}(\Sigma)$ is obtained from $\mathbf{A} \in \mathit{wff}(\Sigma)$ by an **one-step β-reduction** ($\mathbf{A} \longrightarrow_\beta \mathbf{B}$), if it is obtained by applying the following rule to a well-formed part of $\mathbf{A}$:

$$[\lambda X.\mathbf{C}]\mathbf{D} \longrightarrow_\beta \sigma(\mathbf{C})$$

where $\sigma = \{X \mapsto \mathbf{D}\}$. As usual we will denote the transitive closure of the β-reduction relation with $\stackrel{*}{\longrightarrow}_\beta$. Thus $\mathbf{A} \stackrel{*}{\longrightarrow}_\beta \mathbf{B}$, iff there is a sequence of one-step β-reductions

$$\mathbf{A} \longrightarrow \mathbf{A}^1 \longrightarrow_\beta \ldots \longrightarrow_\beta \mathbf{A}^n \longrightarrow_\beta \mathbf{B}$$

This induces the equivalence relation $=_\beta$ of **β-equality** on $\mathit{wff}(\Sigma)$.

Lemma 2.3.7 *We state the following well-known results. For a detailed discussion we refer to* [HS86].

1. *β-reduction is terminating and confluent.*
2. *The β-reduced form of a well-formed formula* $\mathbf{B}$ *is of the form* $[\lambda \overline{X^n}.\mathbf{A}\overline{\mathbf{E}^m}]$, *where* $\mathbf{A}$ *is a constant or a variable, and the subterms* $\mathbf{E}^i$ *are in β-reduced form.*
3. $\mathbf{A}_\alpha =_\beta \mathbf{B}_\alpha$, *iff the β-reduced forms of* $\mathbf{A}$ *and* $\mathbf{B}$ *are alphabetical variants.*

3 Semantics of $\mathcal{Q}$

In this paper we find it useful to study the general model semantics for $\mathcal{Q}$ in a more algebraic setting than for example in [And86]. In particular the notions of homomorphism and congruence will be useful later on.

3.1 Σ-Algebras

Definition 3.1.1 (Partial Functions) Let $A_1, \ldots, A_n$ and B be sets. The **cartesian product** $A_1 \times \ldots \times A_n$ is the set of ordered tuples $\{(a_1, \ldots, a_n) \mid a_i \in A_i\}$. An ($n$-ary) **partial function** $\Phi: A_1 \times \ldots \times A_n \longrightarrow B$ is a subset of $A_1 \times \ldots \times A_n \times B$, that does not contain two different tuples having the same first n components. The **domain** $\mathbf{Dom}(\Phi)$ of Φ is the set $\{(a_1, \ldots, a_n) \mid (a_1, \ldots, a_n, a_{n+1}) \in \Phi\}$, the **image** $\mathbf{Im}(\Phi)$ of Φ is the set $\{a_{n+1} \mid (a_1, \ldots, a_n, a_{n+1}) \in \Phi\}$. A partial function $\Phi \subseteq A_1 \times \ldots \times A_n \times B$ is called **total**, iff $\mathbf{Dom}(\Phi) = A_1 \times \ldots \times A_n$; we will assume all functions in this paper to be total and therefore sometime use the word function instead. We denote the family of all total functions by $\mathcal{F}(A_1, \ldots, A_n; B)$ and use the clause

$$\Phi: A_1 \times \ldots \times A_n \longrightarrow B; (a_1, \ldots, a_n) \mapsto b$$

synonymously for $\Phi \in \mathcal{F}(A_1, \ldots, A_n; B)$ and $(a_1, \ldots, n, b) \in \Phi$. The **application of a function Φ to an n-tuple** $(a_1, \ldots, a_n)$, denoted by $\Phi(a_1, \ldots, a_n)$, is the unique value $b \in B$, such that $(a_1, \ldots, a_n, b) \in \Phi$.

Remark 3.1.2 The process of applying a function Φ to an n-tuple $(a_1, \ldots, a_n)$ can be considered as applying Φ to the sequence of values $a_1, \ldots, a_n$ one after the other. The application of Φ to part of the sequence yields a function that will give a_{n+1}, if applied to the rest of the sequence. This process is called **currying**. Thus $\mathcal{F}(A_1, \ldots, A_n; B)$ becomes $\mathcal{F}(A_1; \mathcal{F}(A_2; \ldots; \mathcal{F}(A_n; B) \ldots)$ and therefore we can and will restrict ourselves to unary functions.

Definition 3.1.3 (Extensionality) Two partial functions $f, g: A \longrightarrow B$ are equal, iff they are equal as binary relations, that is, if $\mathbf{Dom}(f) = \mathbf{Dom}(g)$ and for all $x \in \mathbf{Dom}(f)$ we have $f(x) = g(x)$. This property is called the **extensionality** of equality.

Definition 3.1.4 (Typed Collection) A collection of sets $\mathcal{D}_\mathcal{T} = \{\mathcal{D}_\alpha \mid \alpha \in \mathcal{T}\}$ indexed by the set $\mathcal{T}$ of type symbols is called a **typed collection (of sets)**. Let $\mathcal{D}_\mathcal{T}$ and $\mathcal{E}_\mathcal{T}$ be typed collections, then a collection $\mathcal{I} := \{\mathcal{I}_\alpha: \mathcal{D}_\alpha \longrightarrow \mathcal{E}_\alpha \mid \alpha \in \mathcal{T}\}$ of maps is called a **typed map** $\mathcal{I}: \mathcal{D}_\mathcal{T} \longrightarrow \mathcal{E}_\mathcal{T}$.

We will often view a typed collection $\mathcal{D}_\mathcal{T}$ as the union $\bigcup_{\alpha \in \mathcal{T}} \mathcal{D}_\alpha$ and typed maps $\mathcal{I}: \mathcal{D}_\mathcal{T} \longrightarrow \mathcal{E}_\mathcal{T}$ as maps

$$\mathcal{I}: \bigcup_{\alpha \in \mathcal{T}} \mathcal{D}_\alpha \longrightarrow \bigcup_{\alpha \in \mathcal{T}} \mathcal{E}_\alpha$$

with $\mathbf{Im}\left(\mathcal{I}|_{\mathcal{D}_\alpha}\right) \subseteq \mathcal{E}_\alpha$. We will switch the point of view whenever convenient. A collection $\{R^\alpha \subseteq \mathcal{D}_\alpha \times \mathcal{D}_\alpha \mid \alpha \in \mathcal{T}\}$ is called a **typed binary relation**.

Definition 3.1.5 (Σ-Quasi-Algebra) A **Σ-quasi-algebra** $\mathcal{A} = (\mathcal{D}, \mathcal{I})$ consists of a typed collection $\mathcal{D} = \mathcal{D}_{\mathcal{T}}$ of nonempty sets, such that $\mathcal{D}_{\alpha\beta} \subseteq \mathcal{F}(\mathcal{D}_\beta; \mathcal{D}_\alpha)$, and a typed map $\mathcal{I}: \Sigma \longrightarrow \mathcal{D}$. The collection $\mathcal{D}$ is called the **carrier set** or the **frame of $\mathcal{A}$** and the map $\mathcal{I}$ the **interpretation of constants.**

Example 3.1.6 We can think of the formula $\mathbf{A}_{\alpha\beta} \in \mathit{wff}_{\alpha\beta}(\Sigma)$ as a function

$$\mathbf{A}_{\alpha\beta}: \mathit{wff}_\beta(\Sigma) \longrightarrow \mathit{wff}_\alpha(\Sigma) \, ; \, \mathbf{B}_\beta \mapsto [\mathbf{AB}] \, .$$

Thus $(\mathit{wff}(\Sigma), \mathrm{Id}_\Sigma)$ is a total Σ-quasi-algebra. Therefore the set $\mathit{wff}(\Sigma)$ is often called the **term algebra for the signature Σ.**

Definition 3.1.7 (Assignment) Let $\mathcal{A} = (\mathcal{D}, \mathcal{I})$ be a Σ-quasi-algebra. A typed map $\varphi: \mathcal{V} \longrightarrow \mathcal{D}$ is called an **assignment into $\mathcal{A}$.** We denote the assignment ψ with $\psi(X) = g$, $\psi(Y) = \varphi(Y)$ and $Y \neq X$ by $(\varphi : X \mapsto g)$.

Definition 3.1.8 (Homomorphic Extension) Let $\mathcal{A} = (\mathcal{D}, \mathcal{I})$ be a Σ-quasi-algebra and φ an assignment into $\mathcal{A}$. The **homomorphic extension $\mathcal{I}_\varphi$ of φ to** $\mathit{wff}(\Sigma)$ is inductively defined to be a map $\mathcal{I}_\varphi: \mathit{wff}(\Sigma) \longrightarrow \mathcal{D}$, such that

1. $\mathcal{I}_\varphi(X) = \varphi(X)$, if X is a variable,
2. $\mathcal{I}_\varphi(c) = \mathcal{I}(c)$, if c is a constant,
3. $\mathcal{I}_\varphi([\mathbf{AB}]) = \mathcal{I}_\varphi(\mathbf{A})(\mathcal{I}_\varphi(\mathbf{B}))$,
4. $\mathcal{I}_\varphi([\lambda X_\alpha.\mathbf{B}_\beta]) \in \mathcal{F}(\mathcal{D}_\alpha; \mathcal{D}_\beta)$ is defined by $\mathcal{I}_\varphi([\lambda X.\mathbf{B}])(z) := \mathcal{I}_{(\varphi : X \mapsto z)}(\mathbf{B})$.

For any well-formed formula $\mathbf{A}_\alpha$ we call $\mathcal{I}_\varphi(\mathbf{A}) \in \mathcal{D}_\alpha$ the **value** or **denotation of A in $\mathcal{A}$ for the assignment** φ. This definition does not imply that $\mathcal{I}_\varphi(\mathbf{A})$ is defined for each well-formed formula $\mathbf{A}$. This condition is enforced by the following definition.

Definition 3.1.9 (Σ-Algebra) A Σ-quasi-algebra $\mathcal{A} = (\mathcal{D}, \mathcal{I})$ is called **comprehension-closed,** iff for each assignment φ into $\mathcal{A}$ the homomorphic extension $\mathcal{I}_\varphi$ of φ to $\mathit{wff}(\Sigma)$ is everywhere defined, i.e. $\mathbf{Im}(\mathcal{I}_\varphi) \subseteq \mathcal{D}$. A Σ-quasi-algebra is called **Σ-algebra,** iff it is comprehension-closed.

These closure conditions for the carrier set $\mathcal{D}$ of $\mathcal{A}$ assure that the universes of functions $\mathcal{D}_{\alpha\beta}$ are rich enough to contain a value for all $\mathbf{A}_{\alpha\beta} \in \mathit{wff}_{\alpha\beta}(\Sigma)$. For a detailed discussion we refer the reader to [And72].

A Σ-quasi-algebra is called **full,** iff $\mathcal{D}_{\alpha\beta}$ contains all functions $\mathcal{D}_\beta \longrightarrow \mathcal{D}_\alpha$.

Example 3.1.10 The typed sets of well-formed formulae $\mathit{wff}(\Sigma)$ and $\mathit{cwff}(\Sigma)$ are Σ-algebras. Now we can understand why they are traditionally called **term algebras** for the signature Σ.

Definition 3.1.11 (Homomorphism of Σ-Algebras) Let $\mathcal{M} = (\mathcal{D}, \mathcal{I})$ and $\mathcal{N} = (\mathcal{E}, \mathcal{J})$ be Σ-algebras. A **homomorphism of Σ-algebras** is a typed map $\tau: \mathcal{D} \longrightarrow \mathcal{E}$, such that

1. $\tau \circ \mathcal{I} = \mathcal{J}$.

2. For all $f \in \mathcal{D}_{\alpha\beta}$ and $g \in \mathcal{D}_\beta$, we have: if $g \in \mathbf{Dom}(f)$, then $\tau(g) \in \mathbf{Dom}(\tau(f))$ and $\tau(f)[\tau(g)] = \tau(f[g])$.

As usual we define an **endomorphism τ on $\mathcal{M}$** to be a homomorphism $\tau: \mathcal{M} \longrightarrow \mathcal{M}$, an Σ-**epimorphism** and a Σ-**monomorphism** to be surjective and injective Σ-homomorphisms respectively. Note that for any assignment $\varphi: \mathcal{V} \longrightarrow \mathcal{M}$, the homomorphic extension $\mathcal{I}_\varphi: \mathit{wff}(\Sigma) \longrightarrow \mathcal{M}$ is a homomorphism of Σ-algebras.

3.2 Σ-Congruences

Definition 3.2.1 (Σ-Congruence) Let $\mathcal{A} = (\mathcal{D}, \mathcal{I})$ be a Σ-quasi-algebra, then a typed binary relation $\equiv$ is called a **Σ-congruence on $\mathcal{A}$**, iff the following conditions hold:

1. The relation $\equiv$ is an equivalence relation on $\mathcal{D}$.
2. If $f \in \mathcal{D}_{\alpha\beta}$ and $g \equiv g' \in \mathcal{D}_\beta$, then $f(g) \equiv f(g')$.
3. If $f \equiv f' \in \mathcal{D}_{\alpha\beta}$ and $g \in \mathcal{D}_\beta$, then $f(g) \equiv f'(g)$.

For an equivalence relation $\equiv$ we will denote the equivalence class of $f \in \mathcal{D}$ by $[f] := [f]_\equiv := \{g \in \mathcal{D} \mid g \equiv f\}$.

A Σ-congruence $\equiv$ is called **extensional**, iff for all types α and all $f, g \in \mathcal{D}_{\alpha\beta}$ the fact that $f(a) \equiv g(a)$ for all $a \in \mathcal{D}_\beta$ implies that $f \equiv g$.

Lemma 3.2.2 *Let $\mathcal{A}$ be a term algebra. Then an equivalence relation $\equiv$ that contains the β-equality relation $=_\beta$ is a Σ-congruence on $\mathcal{A}$ if condition 3.2.1(2) holds.*

Proof: We will discuss the assertion for $\mathcal{A} = \mathit{wff}(\Sigma)$ as an example for $\mathit{cwff}(\Sigma)$.

Let $\mathbf{A}_{\alpha\beta} \equiv \mathbf{B}_{\alpha\beta}$ and $\mathbf{D} := [\lambda X_{\alpha\beta} . X\mathbf{C}]$, where $\mathbf{C} \in \mathit{wff}_\beta(\Sigma)$ is an arbitrary formula, then $\mathbf{AC} =_\beta \mathbf{DA} \equiv \mathbf{DB} =_\beta \mathbf{BC}$ by 3.2.1(2). Since $\mathbf{C}$ was chosen arbitrarily we have 3.2.1(3). □

Lemma 3.2.3 *Let $\mathcal{A}$ be a Σ-quasi-algebra and $\sim$ be an extensional Σ-congruence on $\mathcal{A}$, then $\mathcal{A}/_\sim$ is also a Σ-algebra. Furthermore the* **canonical projection** *$\pi_\sim: \mathcal{A} \longrightarrow \mathcal{A}/_\sim; f \mapsto [f]_\sim$ is a Σ-epimorphism.*

Proof: In order to prove that $\mathcal{A}/_\sim = (\mathcal{D}^\sim, \mathcal{I}^\sim)$ is a Σ-algebra, we have to show that $\mathcal{D}^\sim_{\alpha\beta} \subseteq \mathcal{F}_p(\mathcal{D}^\sim_\beta; \mathcal{D}^\sim_\alpha)$ and that $\mathbf{Im}(\mathcal{I}^\sim_\varphi) \subseteq \mathcal{D}^\sim$ for all assignments φ into $\mathcal{A}/_\sim$.

Let $[f] \in \mathcal{D}^\sim_{\alpha\beta}$, then we can consider $[f]$ as a map

$$f^* := [f]: \mathcal{D}^\sim_\beta \longrightarrow \mathcal{D}^\sim_\alpha; [g] \mapsto [f(g)]$$

This map is well-defined: suppose, that $f' \in [f]$ and $g' \in [g]$, then $[f(g)] = [f'(g)] = [f'(g')] = [f(g')]$. Therefore the above definition depends only on equivalence classes. The extensionality of $\sim$ ensures, that each function is represented by at most one congruence class and therefore $\mathcal{D}^\sim_{\alpha\beta}$ is a subset of $\mathcal{F}(\mathcal{D}^\sim_\beta; \mathcal{D}^\sim_\alpha)$.

To convince ourselves that $\pi_\sim$ is indeed a homomorphism of Σ-quasi-algebras, we note that by definition $\pi_\sim$ is surjective and $\mathcal{I}^\sim = \pi_\sim \circ \mathcal{I}$. Now let $f \in \mathcal{D}_{\alpha\beta}$, and $g \in \mathbf{Dom}(f) \subseteq \mathcal{D}_\beta$, then $g' \in [g]$ for all $g' \in \mathbf{Dom}(f)$ and therefore $[g] = \pi_\sim(g) \in \mathbf{Dom}([f]) = \mathbf{Dom}(\pi_\sim(f))$ and $\pi(f)[\pi(g)] = [f]([g]) = [f(g)] = \pi(f(g))$. □

Definition 3.2.4 (Quotient Algebra) Let $\sim$ be a Σ-congruence, $\mathcal{D}_\alpha^\sim := \{[f] \mid f \in \mathcal{D}_\alpha\}$ and $\mathcal{I}^\sim(c_\alpha) := [\mathcal{I}(c_\alpha)]$ for all constants c_α. Then $\mathcal{A}/_\sim = (\mathcal{D}^\sim, \mathcal{I}^\sim)$ is called the **quotient algebra of $\mathcal{A}$ for the relation $\sim$.**

Lemma 3.2.5 *If $\mathcal{A}$ is a Σ-algebra and $\sim$ is an extensional Σ-congruence on $\mathcal{A}$, then $\mathcal{A}/_\sim$ is a Σ-algebra and $\pi_\sim$ is a homomorphism of Σ-algebras.*

Proof: Let ψ be an assignment into $\mathcal{A}/_\sim$, then there exists an assignment φ into $\mathcal{A}$, such that $\Psi = \pi_\sim \circ \varphi$. We will prove that the denotation $\mathcal{I}_\psi^\sim = \pi_\sim \circ \mathcal{I}_\varphi$ which induces the assertion, by induction over the structure of well-formed formulae. In order to simplify the notation we will abbreviate $\pi_\sim$ by π.

1. $\mathcal{I}_\psi^\sim(X) = \psi(X) = \pi \circ \varphi(X) = \pi(\mathcal{I}_\varphi(X))$
2. $\mathcal{I}_\psi^\sim(c) = \mathcal{I}^\sim(c) = \pi \circ \mathcal{I}(c) = \pi(\mathcal{I}_\varphi(c))$
3. $\mathcal{I}_\psi^\sim(\mathbf{AB}) = \mathcal{I}_\psi^\sim(\mathbf{A})(\mathcal{I}_\psi^\sim(\mathbf{B})) = \pi \circ \mathcal{I}_\varphi(\mathbf{A})(\pi \circ \mathcal{I}_\varphi(\mathbf{B})) = \pi(\mathcal{I}_\varphi(\mathbf{AB})$
4. $\mathcal{I}_\psi^\sim([\lambda X\mathbf{A}])(\pi(g)) = \mathcal{I}_{(\psi:X\mapsto\pi(g))}^\sim(\mathbf{A}) = \pi(\mathcal{I}_{(\varphi:X\mapsto g)}(\mathbf{A}) = \pi \circ \mathcal{I}_\varphi([\lambda X\mathbf{A}])(\pi(g))$

□

3.3 General Models

Definition 3.3.1 (General Model for $\mathcal{Q}(\Sigma)$) A Σ-algebra $\mathcal{M} = (\mathcal{D}, \mathcal{I})$ is called **a general model for** $\mathcal{Q}(\Sigma)$, iff $\mathcal{D}_o$ is the set $\{\mathtt{T}, \mathtt{F}\}$ of truth values and $\mathcal{I}(q_{o\alpha\alpha})$ is the identity relation on $\mathcal{D}_\alpha$.

We are striving for a general notion of algebraic model, so we only require $\mathcal{M}$ to be comprehension-closed. In particular we do not require $\mathcal{M}$ to be full. A full general model is called a **standard model.**

Remark 3.3.2 Note that the class of general models defined above is rich in nonstandard models, since we do not require it to contain a description function. In this detail we differ from most systems of simple type theory (cf. [Rus08, Chu40, Hen50, And71, And86]), which do require the existence of a constant $\iota_{\alpha(o\alpha)}$ for each type α. Correspondingly these approaches require that this constant denotes the function that maps singleton sets to their unique member in (general) models. Even though our's may not be the most interesting notion of general model, we choose not to deal with descriptions in the current paper, since we want to treat one problem at a time, in this case extensionality.

Remark 3.3.3 Let $\mathcal{M} = (\mathcal{D}, \mathcal{I})$ and $\mathcal{N} = (\mathcal{E}, \mathcal{J})$ be general models for $\mathcal{Q}(\Sigma)$, and let $\Phi: \mathcal{M} \longrightarrow \mathcal{N}$ be a homomorphism of Σ-algebras, then by 3.3.4(3) and 3.1.11(2) we have $\Phi(\mathtt{T}) = \Phi(\mathcal{I}_\varphi(\mathsf{T}_o)) = \mathcal{J}_{\Phi\circ\varphi}(\mathsf{T}_o) = \mathtt{T}$ and $\Phi(\mathtt{F}) = \mathtt{F}$.

Lemma 3.3.4 *Let $\mathcal{M} = (\mathcal{D}, \mathcal{I})$ be a general model of $\mathcal{Q}(\Sigma)$ and φ an assignment into $\mathcal{M}$, then we have*

1. $\mathcal{I}_\varphi([[\lambda X.\mathbf{B}]\mathbf{A}]) = \mathcal{I}_{(\varphi:X\mapsto\mathcal{I}_\varphi(\mathbf{A}))}(\mathbf{B})$,

2. $\mathcal{I}_\varphi([\mathbf{A} = \mathbf{B}]) = \mathtt{T}$, *iff* $\mathcal{I}_\varphi(\mathbf{A}) = \mathcal{I}_\varphi(\mathbf{B})$,

3. $\mathcal{I}_\varphi(\top_o) = \mathtt{T}$ *and* $\mathcal{I}_\varphi(\bot_o) = \mathtt{F}$,

4. $\mathcal{I}_\varphi(\vee_{ooo})(g,h) = \mathtt{T}$, *iff* $g = \mathtt{T}$ *or* $h = \mathtt{T}$,

5. $\mathcal{I}_\varphi(\Rightarrow_{ooo})(g,h) = \mathtt{T}$, *iff* $g = \mathtt{F}$ *or* $h = \mathtt{T}$,

6. $\mathcal{I}_\varphi(\wedge_{ooo})(g,h) = \mathtt{T}$, *iff* $g = \mathtt{T}$ *and* $h = \mathtt{T}$,

7. $\mathcal{M} \models [\forall X.\mathbf{A}_o]$ *iff* $\mathcal{I}_\psi(\mathbf{A}) = \mathtt{T}$ *for all assignments* ψ*, that agree with* φ *off* X.

Proof: See Lemma 5401 in [And86]. □

Lemma 3.3.5 *Let* $\mathcal{M} = (\mathcal{D},\mathcal{I})$ *be a general model for* $\mathcal{Q}(\Sigma)$ *and let* $\mathbf{Q}_{o\alpha\alpha}$ *be defined as in* 2.2.2*, then* $\mathcal{I}_\varphi(\mathbf{Q})$ *is the identity relation on* $\mathcal{D}_\alpha$.

Proof: Let $a,b \in \mathcal{D}_\alpha$, then we have $\mathcal{I}_\varphi(\mathbf{Q}) = \mathcal{I}_\varphi(\lambda X.\lambda Y.\forall P.PX \Rightarrow PY)$ by definition 2.2.2, so obviously $\mathcal{I}_\varphi(\mathbf{Q})(a,a) = \mathtt{T}$. Now let $a \neq b \in \mathcal{D}_\alpha$, then $\mathcal{I}_\varphi(\mathbf{Q})(a,b) = \mathtt{T}$, iff $\mathcal{I}_\psi(\mathbf{Q}) = \mathtt{T}$ with $\psi := (\varphi{:}X \mapsto a, Y \mapsto b, P \mapsto r)$ for all $r \in \mathcal{D}_{o\alpha}$. However there is a relation $r := \mathcal{I}_{(\varphi:Y\mapsto a)}(\lambda X.Y = X) = \{a\} \in \mathcal{D}_\alpha$, since $\mathcal{D}_{o\alpha\alpha}$ contains the identity relation $\mathcal{I}(=_{o\alpha\alpha})$ by 3.3.1 that makes $\mathcal{I}_\psi(PX \Rightarrow PY) = ra \Rightarrow rb$ false. Thus we have $\mathcal{I}_\varphi(\mathbf{Q})(a,b) = \mathtt{F}$. □

Definition 3.3.6 Let $\mathcal{K}$ be a class of models for $\mathcal{Q}(\Sigma)$, $\mathcal{M} = (\mathcal{D},\mathcal{I}) \in \mathcal{K}$, φ an assignment into $\mathcal{M}$ and $\mathbf{A}_o \in \mathit{wff}_o(\Sigma)$. We say that

1. φ **satisfies A in** $\mathcal{M}$ ($\mathcal{M} \models_\varphi \mathbf{A}$), iff $\mathcal{I}_\varphi(\mathbf{A}) = \mathtt{T}$.

2. **A is satisfiable in** $\mathcal{M}$, iff there is an assignment φ that satisfies **A** in $\mathcal{M}$.

3. **A** is **satisfiable in** $\mathcal{K}$, iff there is a model $\mathcal{M}$, such that **A** is satisfiable in $\mathcal{M}$.

4. **A** is **valid in** $\mathcal{M}$ ($\mathcal{M} \models \mathbf{A}$), iff all assignments into $\mathcal{M}$ satisfy **A** in $\mathcal{M}$.

5. **A** is **valid in** $\mathcal{K}$ ($\models_\mathcal{K} \mathbf{A}$), iff **A** is valid in all $\mathcal{M} \in \mathcal{K}$.

Definition 3.3.7 Let $\mathcal{K}$ be a class of models for $\mathcal{Q}(\Sigma)$, then we say a proposition $\mathbf{A}_o$ **entails** a proposition $\mathbf{B}_o$ **in** $\mathcal{K}$ ($\mathbf{A} \models_\mathcal{K} \mathbf{B}$), iff for all $\mathcal{M} \in \mathcal{K}$ we have that $\mathcal{M} \models \mathbf{A}$ implies $\mathcal{M} \models \mathbf{B}$.

4 Calculi for $\mathcal{Q}$

In this section we will introduce the syntactic counterparts of the entailment relation.

4.1 Calculi, Derivations and Consistency

Definition 4.1.1 (Calculus) Let $\mathcal{Q}(\Sigma)$ be a formulation of type theory, then an **inference rule** is an effectively computable relation on $wff_o(\Sigma)$. Inference rules are traditionally represented by a schema

$$\frac{\textsc{Ante}}{\textsc{Succ}}\; R$$

where the **antecedent** ANTE is a set $\{\mathbf{A}_o^1, \ldots, \mathbf{A}_o^n\}$ of propositions, the **succedent** SUCC is a proposition $\mathbf{B}_o$ and R is the set of tuples $(\mathbf{C}^1, \ldots, \mathbf{C}^n, \mathbf{D})$, such that $\mathbf{C}^i$ and $\mathbf{D}$ are substitution instances of $\mathbf{A}^i$ and $\mathbf{B}$. In order to give a finite presentation of a calculus the schemata may be schematic in types and terms. Inference rules with empty antecedent are called **axioms** and otherwise **proper inference rules**. A **calculus** $\mathcal{C}$ **for** $\mathcal{Q}(\Sigma)$ is a finite set of **inference rules**.

Definition 4.1.2 (Higher-Order Theory) Let $\mathcal{Q}(\Sigma)$ be a formulation of simple type theory and $\mathcal{C}$ a calculus for $\mathcal{Q}(\Sigma)$, then we call the pair $\mathcal{T} := (\mathcal{Q}(\Sigma), \mathcal{C})$ a **higher-order theory**. The set $wff(\Sigma)$ of well-formed formulae of $\mathcal{Q}(\Sigma)$ is called the **language** $\mathcal{L}(\mathcal{T})$ **of** $\mathcal{T}$. If $\mathcal{K}$ is a class of models for $\mathcal{Q}(\Sigma)$, then we will also call $\mathcal{K}$ a class of models for $\mathcal{T}$.

Definition 4.1.3 ($\mathcal{T}$-Derivation) Let $\mathcal{T} := (\mathcal{Q}(\Sigma), \mathcal{C})$ be a higher-order theory and $\mathcal{D}$ be a finite tree, where each node $\mathcal{N}$ in $\mathcal{D}$ is labeled with a triple $(\mathbf{B}_o, R, \{\mathbf{A}_o^1, \ldots, \mathbf{A}_o^n\})$, such that $R \in \mathcal{C}$ and $(\mathbf{A}^1, \ldots, \mathbf{A}^n, \mathbf{B}) \in R$. $\mathbf{B}$ is called the **assertion**, R the **justification**, and the set of $\mathbf{A}^i$ the **support** of $\mathcal{N}$. $\mathcal{D}$ is called a **$\mathcal{T}$-derivation**, iff each node $\mathcal{N}$ with label $(\mathbf{B}, R, \{\mathbf{A}^1, \ldots, \mathbf{A}^n\})$ has n children $\mathcal{N}^i$ with assertions $\mathbf{A}^i$. Since the theory can often be identified by the calculus alone, we will often simply speak of $\mathcal{C}$-derivations. Because of the tree nature, we will often call $\mathcal{T}$-derivations **proof-trees**.

Let $\mathbf{A}_o$ be a proposition and Φ be a set of sentences. We call a $\mathcal{T}$-derivation $\mathcal{D}$ a **$\mathcal{T}$-derivation of A from the set Φ of hypotheses**, if $\mathbf{A}$ is the assertion of the root of $\mathcal{D}$ and the supports of the leaves of $\mathcal{D}$ are subsets of Φ. If there exists a $\mathcal{T}$-derivation of $\mathbf{A}$ from Φ, then we write $\Phi \vdash_{\mathcal{T}} \mathbf{A}$. Let $\mathcal{C}$ be a calculus, then a proposition $\mathbf{A}_o$ is called a **theorem of** $\mathcal{T}$, iff there exists a $\mathcal{T}$-derivation of $\mathbf{A}$ from the empty set of hypotheses.

Definition 4.1.4 Let $\mathcal{T}$ and $\mathcal{T}'$ be higher-order theories, then $\mathcal{T}'$ is called an

expansion of $\mathcal{T}$, iff $\mathcal{L}(\mathcal{T}) \subseteq \mathcal{L}(\mathcal{T}')$,

extension of $\mathcal{T}$, iff $\mathcal{T}'$ is an expansion of $\mathcal{T}$ and every theorem of $\mathcal{T}$ is also a theorem of $\mathcal{T}'$,

conservative extension of $\mathcal{T}$, iff for every proposition $\mathbf{A}_o \in \mathcal{L}(\mathcal{T})$ we have $\vdash_{\mathcal{T}'} \mathbf{A}$, iff $\vdash_{\mathcal{T}} \mathbf{A}$.

Remark 4.1.5 Since $wff(\Sigma)$ is inductively defined from Σ and the set of variables is fixed, $\mathcal{T}'$ is an expansion of $\mathcal{T}$, iff each constant in $\mathcal{L}(\mathcal{T})$ is also a constant in $\mathcal{L}(\mathcal{T}')$. When expanding a theory $\mathcal{T}$ by new constants, one naturally has to add new instances of the schemata of the rules of inference to the calculus of $\mathcal{T}$.

Lemma 4.1.6 *Let $\mathcal{T} = (\mathcal{Q}(\Sigma), \mathcal{C})$ and $\mathcal{T}' = (\mathcal{Q}(\Sigma'), \mathcal{C})$ be higher-order theories, such that $\mathcal{T}'$ is an expansion of $\mathcal{T}$ obtained by adding new constants to Σ (and not changing the calculus). Then $\mathcal{T}'$ is a conservative extension of $\mathcal{T}$.*

Proof: Let $\mathbf{A}_o \in \mathit{wff}(\Sigma)$ be a theorem of $\mathcal{T}'$ and $\mathcal{D}'$ be a $\mathcal{T}'$-proof of $\mathbf{A}$, then we can replace all the constants in $\Sigma' \setminus \Sigma$ by new variables, and the result will still be a proof for $\mathbf{A}$. Thus $\mathbf{A}$ is a theorem of $\mathcal{T}$. □

Definition 4.1.7 Let $\mathcal{T} := (\mathcal{Q}(\Sigma), \mathcal{C})$ be a higher-order theory, then an inference rule $\mathcal{R}$ is called **admissible in** $\mathcal{T}$, iff adding $\mathcal{R}$ to the calculus of $\mathcal{T}$ does not change the set of theorems. $\mathcal{R}$ is called **derivable in** $\mathcal{T}$, iff for each $\mathbf{A}^1, \ldots, \mathbf{A}^n \vdash_{\mathcal{R}} \mathbf{B} \in \mathcal{R}$ there already exists a $\mathcal{T}$-derivation $\mathbf{A}^1, \ldots, \mathbf{A}^n \vdash_{\mathcal{T}} \mathbf{B}$.

General Assumption 4.1.8 We assume that in all higher-order theories in this paper, the inference rules of implication introduction (deduction theorem) and introduction elimination (Modus Ponens) are admissible rules of inference. This is a sufficient condition to get nice correspondences between the implication constant $\Rightarrow$ in the logic and the consequence relation in the model theory. We will justify this assumption for every concrete higher-order theory later on.

Definition 4.1.9 ($\mathcal{T}$-Consistent) Let $\mathcal{T}$ be a higher-order theory, then a set Φ of $\mathcal{L}(\mathcal{T})$-sentences is called **$\mathcal{T}$-inconsistent** or **inconsistent with respect to** $\mathcal{T}$, iff $\Phi \vdash_{\mathcal{T}} \bot_o$. Φ is called **$\mathcal{T}$-consistent**, iff it is not $\mathcal{T}$-inconsistent. If we are in a formulation of type theory, where we do not have the constant $\bot_o$ in the signature, we will define $\mathcal{T}$-inconsistency by the derivability of a basic contradiction of the form $\mathbf{A}_o \wedge \neg\mathbf{A}$. We call a set Ψ **$\mathcal{T}$-consistent with** a set Φ, iff $\Phi \cup \Psi$ is $\mathcal{T}$-consistent.

Lemma 4.1.10 *Let $\mathcal{T}'$ be a conservative extension of $\mathcal{T}$. Then a set Φ of propositions is $\mathcal{T}$-consistent, iff it is $\mathcal{T}'$-consistent.*

Proof: Immediate from the definition. □

Lemma 4.1.11 *If $\mathcal{T}$ is a higher-order theory and Φ is a $\mathcal{T}$-inconsistent set of propositions, then there exists a finite $\mathcal{T}$-inconsistent subset of Φ.*

Proof: Let $\mathcal{D}$ be a $\mathcal{T}$-derivation of $\bot_o$ from Φ. As $\mathcal{D}$ is a finite tree, the set $\Psi \subseteq \Phi$ of labels of the leaves of $\mathcal{D}$ is finite. Thus Ψ is a $\mathcal{T}$-inconsistent and finite subset of Ψ. □

Definition 4.1.12 (Soundness) Let $\mathcal{T}$ be a higher-order theory and $\mathcal{K}$ a class of models for $\mathcal{T}$, then $\mathcal{T}$ is called **sound with respect to** $\mathcal{K}$, iff each theorem of $\mathcal{T}$ is valid in $\mathcal{K}$.

Theorem 4.1.13 *Let $\mathcal{K}$ be a class of models for $\mathcal{Q}(\Sigma)$ and $\mathcal{T} = (\mathcal{Q}(\Sigma), \mathcal{C})$ be a higher-order theory that is sound with respect to $\mathcal{K}$. Furthermore let $\Phi \subseteq \mathit{cwff}_o(\Sigma)$.*

1. *If $\mathbf{A}_o^1, \ldots, \mathbf{A}_o^n \vdash_{\mathcal{T}} \mathbf{B}_o$, such that $\mathbf{B}$ is false in some $\mathcal{M} \in \mathcal{K}$, then $\mathcal{M} \models \bigvee_{i=1}^{n} [\neg\mathbf{A}^i]$.*
2. *If $\mathcal{M} \in \mathcal{K}$, then $\Psi := \{\mathbf{A}_o \in \mathit{cwff}_o(\Sigma) \mid \mathcal{M} \models \mathbf{A}\}$ is $\mathcal{T}$-consistent.*

3. Φ *is* $\mathcal{T}$*-inconsistent with* $\mathbf{A}^1,\ldots,\mathbf{A}^n$*, iff* $\Phi \vdash_{\mathcal{T}} \bigvee_{i=1}^n[\neg\mathbf{A}^i]$.

Proof: We will show the first assertion by induction on the size of the $\mathcal{T}$-derivation $\mathcal{D}$ of $\mathbf{B}$ from Φ. Since $\mathcal{T}$ is sound with respect to $\mathcal{K}$, the proposition $\bot_o$ cannot be an axiom of $\mathcal{T}$, thus the base case is vacuously true. For the inductive case let $\mathcal{M} = (\mathcal{A},\mathcal{I}) \in \mathcal{K}$ be a model such that $\mathcal{I}_\varphi(\mathbf{B}) = \mathtt{F}$, and let $\mathcal{D}$ end in an application of the rule $\mathcal{R} := \mathbf{A}^1,\ldots,\mathbf{A}^n \vdash \mathbf{B} \in \mathcal{C}$. Since $\mathcal{T}$ is sound with respect to $\mathcal{K}$ and implication introduction is admissible in $\mathcal{T}$, we have that $\mathcal{I}_\varphi(\mathbf{A}^1,\ldots,\mathbf{A}^n \Rightarrow \mathbf{B}) = \mathtt{T}$ and therefore $\mathcal{I}_\varphi(\bigvee_{i=1}^n[\neg\mathbf{A}^i]) = \mathtt{T}$. Therefore one of the premises of $\mathcal{R}$ is false in $\mathcal{M}$ and by induction we get the assertion.

To prove the second assertion consider the contrapositive statement: Let Ψ be $\mathcal{T}$-inconsistent and $\Phi := \{\mathbf{A}^1,\ldots,\mathbf{A}^n\}$ be an $\mathcal{T}$-inconsistent subset of Ψ, then $\mathbf{A}^1,\ldots,\mathbf{A}^n \vdash_{\mathcal{T}} \bot_o$ and therefore one of the $\mathbf{A}^i$ is false in $\mathcal{M}$ by the first assertion, which contradicts the assumption.

For the third assertion we note that if Φ is $\mathcal{T}$-inconsistent with $\mathbf{A}^1,\ldots,\mathbf{A}^n$, then $\Phi, A^1,\ldots,\mathbf{A}^n \vdash_{\mathcal{T}} \neg\mathbf{A}^1$, so $\Phi \vdash_{\mathcal{T}} \mathbf{A}^1 \Rightarrow \ldots.. \Rightarrow \mathbf{A}^n \Rightarrow \neg\mathbf{A}^1$ and therefore $\Phi \vdash_{\mathcal{T}} \bigvee_{i=1}^n[\neg\mathbf{A}^i]$. The other direction is immediate. □

4.2 The Hilbert Calculi $\mathfrak{T}$ and $\mathfrak{TE}$

In this section we review the calculi $\mathfrak{T}$ from P. Andrews' "Resolution in Type Theory" paper [And71] and $\mathfrak{TE}$ from L. Henkin's "Completeness in the Theory of Types" paper [Hen50]. Note that these differ only in the treatment of extensionality.

Definition 4.2.1 (Andrews' Calculus $\mathfrak{T}$) The calculus $\mathfrak{T}$ consists of the following axioms and axiom schemata:

1. $[p_o \vee p] \Rightarrow p$
2. $p_o \Rightarrow [p \Rightarrow p]$
3. $[p_o \vee q_o] \Rightarrow [q \vee p]$
4. $[p_o \Rightarrow q_o] \Rightarrow .[r_o \vee p] \Rightarrow [r \vee q]$

5^α. $\Pi_{o(o\alpha)}F_{o\alpha} \Rightarrow FX_\alpha$

6^α. $\forall X_\alpha[Y_\alpha \vee F_{o\alpha}X] \Rightarrow .Y \vee \Pi_{o(o\alpha)}F$

and the β-conversion, Modus Ponens, substitution, and universal generalization inference rules

$$\frac{\mathbf{A}=_\beta\mathbf{B} \qquad \mathbf{A}}{\mathbf{B}}\,\beta \qquad \frac{\mathbf{A} \Rightarrow \mathbf{B} \qquad \mathbf{A}}{\mathbf{B}}\,MP$$

$$\frac{\mathbf{A}_{o\alpha}X_\alpha}{\mathbf{A}\mathbf{B}_\alpha}\,Subst \qquad \frac{\mathbf{A}_{o\alpha}X_\alpha}{\Pi_{o(o\alpha)}\mathbf{A}}\,UG$$

where the lower rules have the proviso that the variable X is not free in $\mathbf{A}$.

Definition 4.2.2 (Henkin's Calculus $\mathfrak{TE}$) The extensional variant $\mathfrak{TE}$ of $\mathfrak{T}$ is obtained from $\mathfrak{T}$ by adding the following axioms

10^o. $[X_o \Leftrightarrow Y_o] \Rightarrow .X = Y$

10^α. $\forall X_\beta [F_{\alpha\beta} X = FY] \Rightarrow .X = Y$

These axioms are traditionally called the axioms of extensionality, even though at only 10^α corresponds to the extensionality of equality as defined in 3.1.3. Originally the word "extensionality" was used for predicates, that are extensionally equal (co-extensive), if they denote the same sets. However in our formulation of type theory predicates are proper functions and therefore this notion of extensionality can be broken down to the above axioms. Note that axiom 10^o formalizes the fact that all intended models can only have exactly two truth values (cf. 5.2.6).

Remark 4.2.3 We have adopted the numbering from [Chu40]. To be precise in [Hen50] Henkin introduces the additional axiom of choice

11^α. $F_{o\alpha} X_\alpha \Rightarrow F[\iota_{\alpha(o\alpha)} F]$

for dealing with the description function ι and proves that system complete. This simplifies that completeness proof, since in a logic with the axiom of choice we do not have to postulate the countable infinit set fo existential witnesse c^n, which we have to our system. However the reduced calculus $\mathfrak{TE}$ as defined above is also complete (cf. 5.4.5) with respect to the definition of general models without descriptions (cf. 3.3.2).

Theorem 4.2.4 (Soundness) *$\mathfrak{T}$ and $\mathfrak{TE}$ are sound with respect to the class of general models.*

Proof: In § 52 of [And86] all axioms of $\mathfrak{TE}$ are proven to be theorems of $\mathcal{Q}$ and all rules of inference are proven to be derivable. □

Theorem 4.2.5 (Deduction Theorem) *If $\mathcal{H}, \mathbf{A}_o \vdash_{\mathcal{C}} \mathbf{B}_o$, then $\mathcal{H} \vdash_{\mathcal{C}} \mathbf{A} \Rightarrow \mathbf{B}$, where $\mathcal{C} = \mathfrak{T}$ or $\mathcal{C} = \mathfrak{TE}$.*

Proof: We refer to Lemma 5240 in [And86]. □

In section 5.4 we will prove a completeness theorem for $\mathfrak{TE}$ and show that $\mathfrak{T}$ is not complete with respect to general models by exhibiting a counterexample.

5 Unifying Principle for General Models

In this chapter we will introduce the unifying principle for extensional type theory. The definitions and constructions are similar to those in [And71], therefore we will extend these to the case of general models and discuss the non-extensional case as we go along. As in the intensional case the proof of the unifying principle formalizes the process of extending a given consistent set Φ of sentences and constructing from it a term model for Φ. The most significant difference is that in the extensional setting we work with general models instead of υ-complexes and obtain a completeness theorem with respect to general models.

5.1 Abstract Consistency Classes

Definition 5.1.1 A set $\Phi \subseteq \mathit{wff}(\Sigma)$ is called **sufficiently pure**, iff for each $\alpha \in \mathcal{T}$ there is a countably infinite set $C_\alpha \subseteq \Sigma_\alpha$ of constants that do not occur anywhere in Φ.

Definition 5.1.2 Let Γ be a class of sets.

1. Γ is called **closed under subsets**, iff for all sets S and T the following condition holds: if $S \subseteq T$ and $T \in \Gamma$, then $S \in \Gamma$.
2. Γ is called **of finite character**, iff for every set S the following condition holds: $S \in \Gamma$ iff every finite subset of S is a member of Γ.

Lemma 5.1.3 *If Γ is of finite character, then Γ is closed under subsets.*

Proof: Suppose $S \subseteq T$ and $T \in \Gamma$. Every finite subset A of S is a finite subset of T, and since Γ is of finite character we know that $A \in \Gamma$. Thus $S \in \Gamma$. □

Remark 5.1.4 For reasons of legibility we will write $S * a$ for $S \cup \{a\}$, where S is a set. We will use this notation with the convention that $*$ associates to the left.

Definition 5.1.5 (Abstract Consistency Class) A class Γ of sets of sentences is called an **abstract consistency class**, iff Γ is closed under subsets and for all sets $\Phi \in \Gamma$ the following conditions hold:

1. If $\mathbf{A}_o$ is atomic, then $\mathbf{A} \notin \Phi$ or $[\neg\mathbf{A}] \notin \Phi$.
2. If $\mathbf{A}_o \in \Phi$, then $\mathbf{A}{\downarrow} * \Phi \in \Gamma$, where $\mathbf{A}{\downarrow}$ is the the β-normal form of $\mathbf{A}$.
3. If $\neg\neg\mathbf{A}_o \in \Phi$, then $\mathbf{A} * \Phi \in \Gamma$.
4. If $\mathbf{A} \vee \mathbf{B} \in \Phi$, then $\Phi * \mathbf{A} \in \Gamma$ or $\Phi * \mathbf{B} \in \Gamma$.
5. If $\neg[\mathbf{A} \vee \mathbf{B}] \in \Phi$, then $\Phi * \neg\mathbf{A} * \neg\mathbf{B} \in \Gamma$.
6. If $\Pi_{o(o\alpha)}\mathbf{A}_{o\alpha} \in \Phi$, then $\Phi * [\mathbf{AB}] \in \Gamma$ for each $\mathbf{B}_\alpha \in \mathit{wff}_\alpha(\Sigma)$.
7. If $\neg\Pi_{o(o\alpha)}\mathbf{A}_{o\alpha} \in \Phi$, then $\Phi * \neg[\mathbf{A}c] \in \Gamma$ for each new constant c_α.

Theorem 5.1.6 *The class $\{S \subseteq \mathit{cwff}_o(\Sigma) \mid S$ is $\mathfrak{T}$-consistent$\}$ is an abstract consistency class.*

Proof: We can obviously prove the conditions 5.1.5(1) to 5.1.5(5) in by elementary use of the propositional axioms and the substitution rule. For 5.1.5(6) we note that an appropriate substitution instance of 4.2.1(4) together with Modus Ponens gives the result. □

Theorem 5.1.7 *For each abstract consistency class Γ there exists an abstract consistency class Δ, such that $\Gamma \subseteq \Delta$ and Δ is of finite character.*

Proof: We will follow [And86]. Let

$$\Delta := \{S \subseteq cwff_o(\Sigma) \mid \text{every finite subset of } S \text{ is in } \Gamma\} .$$

To see that $\Gamma \subseteq \Delta$, suppose that $\Phi \in \Gamma$. Γ is closed under subsets, so every finite subset of Φ is in Γ and thus $\Phi \in \Delta$.

Next let us show that Δ is of finite character. Suppose $\Phi \in \Delta$ and Ψ is an arbitrary finite subset of Φ.

By definition of Δ all finite subsets of Ψ are in Γ, and therefore $\Psi \in \Delta$. Thus all finite subsets of Φ are in Δ whenever Ψ is in Δ. On the other hand suppose all finite subsets of Ψ are in Δ. Then by the definition of Δ the finite subsets of Ψ are also in Γ, so $\Phi \in \Delta$. Thus Δ is of finite character.

Now we show that Δ is an abstract consistency class and $\Phi \in \Delta$. By lemma 5.1.3 it is closed under subsets.

1. Suppose there is an atom $\mathbf{A}_o \in \Phi$, such that $\neg\mathbf{A} \in \Phi$. Then $\{\mathbf{A}, \neg\mathbf{A}\} \in \Gamma$ contradicting 5.1.5(1).

2. Let $[\neg\neg\mathbf{A}_o] \in \Phi$, and Ψ be any finite subset of $\Phi * \mathbf{A}$ and $\Theta := (\Psi \setminus \{\mathbf{A}\}) * [\neg\neg\mathbf{A}]$. Θ is a finite subset of Φ, so $\Theta \in \Gamma$. Since Γ is an abstract consistency class and $[\neg\neg\mathbf{A}] \in \Theta$, we get $\Theta * \mathbf{A} \in \Gamma$. We know that $\Psi \subseteq \Theta * \mathbf{A}$ and Γ is closed under subsets, so $\Psi \in \Gamma$. Thus every finite subset Ψ of $\Phi * \mathbf{A}$ is in Γ, therefore by definition $\Phi * \mathbf{A} \in \Delta$.

3. Suppose that $[\mathbf{A}_o \vee \mathbf{B}_o] \in \Phi$, but neither $\Phi * \mathbf{A}$ nor $\Phi * \mathbf{B}$ are in Δ. Then there are finite subsets $\Phi_\mathbf{A}$ and $\Phi_\mathbf{B}$ of Φ, such that $\Phi_\mathbf{A} * \mathbf{A} \notin \Gamma$ and $\Phi_\mathbf{B} * \mathbf{B} \notin \Gamma$ (since all finite subsets of Φ are in Γ). As $\Psi := \Phi_\mathbf{A} \cup \Phi_\mathbf{B} * [\mathbf{A} \vee \mathbf{B}]$ is a finite subset of Φ, we have $\Psi \in \Gamma$. Furthermore $\Psi * \mathbf{A} \in \Gamma$ or $\Psi * \mathbf{A} \in \Gamma$, because Γ is an abstract consistency class and $[\mathbf{A} \vee \mathbf{B}] \in \Psi$. Γ is closed under subsets, so $\Phi_\mathbf{A} * \mathbf{A} \in \Gamma$ or $\Phi_\mathbf{B} * \mathbf{B} \in \Gamma$. This is a contradiction, so we can conclude that if $[\mathbf{A} \vee \mathbf{B}] \in \Phi$, then $\Phi * \mathbf{A} \in \Delta$ or $\Phi * \mathbf{B} \in \Delta$.

4.-7. are treated analogously to 2. (see [And86]).

We have verified all conditions of 5.1.5 and thus proven the assertion. □

Definition 5.1.8 (Extensional Abstract Consistency Class) An abstract consistency class Γ is called an **extensional abstract consistency class**, iff the following additional conditions hold for all sets $\Phi \in \Gamma$:

8. If $\neg[\mathbf{A}_{\alpha\beta} = \mathbf{B}_{\alpha\beta}] \in \Phi$, then for each variable or parameter c_β that does not occur in Φ, $\Phi * [\neg\mathbf{A}c = \mathbf{B}c] \in \Gamma$.

9. If $\{\mathbf{A}_o, \mathbf{B}_o\} \subseteq \Phi$, then $\Phi * [\mathbf{A} = \mathbf{B}] \in \Gamma$.

10. If $\{[\neg \mathbf{A}_o], [\neg \mathbf{B}_o]\} \subseteq \Phi$, then $\Phi * [\mathbf{A} = \mathbf{B}] \in \Gamma$.

Theorem 5.1.9 *For each extensional abstract consistency class Γ there exists an extensional abstract consistency class Δ, such that $\Gamma \subseteq \Delta$ and Δ is of finite character.*

Proof: To convince ourselves that the additional conditions for extensional case hold we will redo the proof for 5.1.8(8) as a model for the rest.

Let $\mathbf{A}_o, \mathbf{B}_o \in \Phi$ and Ψ be any finite subset of $\Phi * [\mathbf{A} = \mathbf{B}]$. Then $\Theta := (\Psi \setminus \{\mathbf{A}\}) * [\neg\neg \mathbf{A}]$ is a finite subset of Φ, and therefore $\Theta \in \Gamma$. Since Γ is an abstract consistency class and $\mathbf{A}, \mathbf{B} \in \Theta$, we have $\Theta * [\mathbf{A} = \mathbf{B}] \in \Gamma$. Furthermore $\Psi \subseteq \Theta * \mathbf{A}$ and Γ is closed under subsets, so $\Psi \in \Gamma$. Thus every finite subset Ψ of $\Phi * [\mathbf{A} = \mathbf{B}]$ is in Γ, therefore by definition we have $\Phi * [\mathbf{A} = \mathbf{B}] \in \Delta$. □

5.2 Higher-Order Hintikka Sets

In this section we turn our attention to the so-called higher-order Hintikka sets, which play a significant role in the proof of the unifying principle. Since higher-order Hintikka sets are maximal closures of Γ-consistent sets of propositions they allow computations that resemble that in a model.

Definition 5.2.1 (Higher-Order Hintikka Set) Let Γ be an abstract consistency class and $H \in \Gamma$. A set $\mathcal{H}$ is called **maximal in** Γ, iff for each sentence $\mathbf{D}$, such that $\mathcal{H} * \mathbf{D} \in \Gamma$, we already have $\mathbf{D} \in \Gamma$. A set $\mathcal{H} \in \Gamma$ is called a **higher-order Hintikka set for Γ and H**, iff $\mathcal{H}$ is maximal in Γ and $H \subseteq \mathcal{H}$.

We will need some technical properties of higher-order Hintikka sets in abstract consistency classes for manipulating formulae.

Theorem 5.2.2 *If Γ is an abstract consistency class, and $\mathcal{H}$ is a higher-order Hintikka set for Γ, then the following statements hold:*

1. *For any sentence $\mathbf{A}_o$ we have: $A \in \mathcal{H}$, iff $[\neg A] \notin \mathcal{H}$.*
2. *$[\neg\neg \mathbf{A}] \in \mathcal{H}$, iff $\mathbf{A} \in \mathcal{H}$.*
3. *$\mathbf{A}_o \in \mathcal{H}$, iff $\mathbf{A}{\downarrow} \in \mathcal{H}$.*
4. *If $\mathbf{A}_o =_{\alpha\beta\eta} \mathbf{B}_o$, then we have $\mathbf{A} \in \mathcal{H}$, iff $\mathbf{B} \in \mathcal{H}$.*
5. *$[\mathbf{A} \lor \mathbf{B}] \in \mathcal{H}$, iff $\mathbf{A} \in \mathcal{H}$ or $\mathbf{B} \in \mathcal{H}$.*
6. *$\neg[\mathbf{A} \lor \mathbf{B}] \in \mathcal{H}$, iff $\neg \mathbf{A} \in \mathcal{H}$ and $\neg \mathbf{B} \in \mathcal{H}$.*
7. *$\Pi_{o(o\alpha)} \mathbf{A}_{o\alpha} \in \mathcal{H}$, iff for each $\mathbf{B}_\alpha \in wff_\alpha(\Sigma)$ we have $\mathbf{AB} \in \mathcal{H}$.*
8. *$\neg\Pi_{o(o\alpha)} \mathbf{A}_{o\alpha} \in \mathcal{H}$, iff for each new constant c_α we have $\neg \mathbf{Ac} \in \mathcal{H}$.*

Proof: We prove the first assertion by induction over the number of occurrences of the connectives $\neg$, $\lor$ and Π in $\mathbf{A}$. For $\mathbf{A}_o \in \mathcal{H}$ we will inductively contradict the assumption $[\neg \mathbf{A}] \in \mathcal{H}$ in case analysis, where the cases correspond to the cases in the definition of the abstract consistency class. We will only show the first three cases, since the analysis of the rest is analogous.

A is atomic by 5.1.5(1),

$\mathbf{A} \doteq \neg\mathbf{B}$ then $[\neg\mathbf{A}] \doteq [\neg\neg\mathbf{B}] \in \mathcal{H}$ and therefore $\mathbf{B} \in \mathcal{H}$ by 5.1.5(3), contradicting the induction hypothesis,

$\mathbf{A} \doteq \mathbf{B} \vee \mathbf{C}$, then $\mathbf{B} \in \mathcal{H}$ or $\mathbf{C} \in \mathcal{H}$ by 5.1.5(4). On the other hand $\neg\mathbf{A} \doteq \neg[\mathbf{B} \vee \mathbf{C}]$ and by 5.1.5(5) we have $\{\neg\mathbf{B}, \neg\mathbf{C}\} \subseteq \mathcal{H}$, contradicting the inductive hypothesis.

The remaining assertions are all of the form: $\Phi \in \mathcal{H}$, iff $\Phi \cup \Psi \in \mathcal{H}$. Thus we can prove all of them by the following schema: If $\Phi \in \mathcal{H}$, then $\mathcal{H} * \Psi \in \Gamma$ (Γ is abstract consistency class). The maximality of $\mathcal{H}$ now gives the assertions. □

Theorem 5.2.3 *If Γ is an extensional abstract consistency class, and $\mathcal{H}$ is a higher-order Hintikka set for Γ, then the following statements hold in addition to those stated in Theorem* **5.2.2**.

9. $\Phi * \neg[\mathbf{A}_{\alpha\beta} = \mathbf{B}_{\alpha\beta}] \in \Gamma$, *iff there is a* $\mathbf{C} \in wff_\beta(\Sigma)$, *such that* $[\neg\mathbf{AC} = \mathbf{BC}] \in \Phi$.

10. $\Phi * [\mathbf{A}_o = \mathbf{B}_o] \in \Gamma$, *iff* $\{\mathbf{A}, \mathbf{B}\} \subseteq \Phi$.

11. $\Phi * [\mathbf{A}_o = \mathbf{B}_o] \in \Gamma$, *iff* $\{[\neg\mathbf{A}], [\neg\mathbf{B}]\} \subseteq \Phi$.

Proof: The proofs are analogous to those of 5.2.2. □

Lemma 5.2.4 *If Γ is an extensional abstract consistency class, and $\mathcal{H}$ is a higher-order Hintikka set for Γ and $H \subseteq cwff_o$ then $\forall X_\beta[\mathbf{A}_{\alpha\beta}X = \mathbf{B}_{\alpha\beta}X] \in \mathcal{H}$, iff $[\mathbf{A} = \mathbf{B}] \in \mathcal{H}$.*

Proof: If $[\mathbf{A} = \mathbf{B}] \notin \mathcal{H}$, then by 5.2.2(1) (and possibly 5.2.2(2)) we have $\neg[\mathbf{A} = \mathbf{B}] \in \mathcal{H}$. So by 5.2.3(9) there is a $\mathbf{C}_\beta \in wff_\beta(\Sigma)$, such that $\neg[\mathbf{AC} = \mathbf{BC}] \in \mathcal{H}$. On the other hand from $\forall X_\beta[\mathbf{A}X = \mathbf{B}X] \in \mathcal{H}$ we obtain $[\mathbf{AC} = \mathbf{BC}] \in \mathcal{H}$ by 5.2.2(7). □

Lemma 5.2.5 *Let Γ be an extensional abstract consistency class and $H \in \Gamma$. If $\mathcal{H}$ is a higher-order Hintikka set for Γ and H, then $[\mathbf{A}_o \Leftrightarrow \mathbf{B}_o] \in \mathcal{H}$, iff $[\mathbf{A} = \mathbf{B}] \in \mathcal{H}$.*

Proof: Let $[\mathbf{A}_o \Leftrightarrow \mathbf{B}_o] \in \mathcal{H}$, then by 5.2.2(6) we also have $\neg\mathbf{A} \vee \mathbf{B} \in \mathcal{H}$ and furthermore $\mathbf{A} \vee \neg\mathbf{B} \in \mathcal{H}$. Because of 5.2.2(5) we have to consider two cases: If $\mathbf{B} \in \mathcal{H}$, then $\neg\mathbf{B} \notin \mathcal{H}$ and therefore $\mathbf{A} \in \mathcal{H}$. If $\neg\mathbf{A} \in \mathcal{H}$, then $\mathbf{A} \notin \mathcal{H}$ and therefore $\neg\mathbf{B} \in \mathcal{H}$. In both cases we get the assertion $[\mathbf{A} = \mathbf{B}] \in \mathcal{H}$ by 5.2.3(10) or 5.2.3(11). □

Lemma 5.2.6 *If $\mathcal{H}$ is a higher-order Hintikka set and* **A**, **B** *are propositions, then either* $\mathbf{A} = \mathbf{B} \in \mathcal{H}$ *or* $\mathbf{A} = \neg\mathbf{B} \in \mathcal{H}$.

Proof: A tedious, but straightforward computation using the results from Lemma 5.2.2 shows that $\neg[\mathbf{A} \Leftrightarrow \mathbf{B}] \in \mathcal{H}$, iff $\mathbf{A} \Leftrightarrow \neg\mathbf{B} \in \mathcal{H}$. Now we conclude with 5.2.2(1), that either $\mathbf{A} \Leftrightarrow \mathbf{B} \in \mathcal{H}$ or $\mathbf{A} \Leftrightarrow \neg\mathbf{B} \in \mathcal{H}$, from which we get the assertion by 5.2.5. □

Theorem 5.2.7 (Abstract Extention Lemma) *Let Γ be an abstract consistency class of finite character and let $H \in \Gamma$ be a sufficiently pure set of sentences. Then there exists a higher-order Hintikka set $\mathcal{H}$ for Γ and H.*

Proof: We can arrange all sentences of $\mathcal{Q}(\Sigma)$ as an infinite sequence $S^1, S^2, \ldots$. For each $n \in \mathbb{N}$ we inductively define a set H^n of sentences by

1. $H^0 := H$.
2. If $H^n * S^n \notin \Gamma$, then $H^{n+1} := H^n$.
3. If $H^n * S^n \in \Gamma$ and S^n is not of the form $[\neg\Pi_{o(o\alpha)}\mathbf{A}_{o\alpha}]$, then $H^{n+1} := H^n * S^n$.
4. If $H^n * S^n \in \Gamma$ and S^n is of the form $[\neg\Pi_{o(o\alpha)}\mathbf{A}_{o\alpha}]$, but not of the form $[\neg\mathbf{A}_{\alpha\beta} = \mathbf{B}_{\alpha\beta}]$, then $H^{n+1} := H^n * \neg\Pi\mathbf{A} * \neg[\mathbf{A}c^n_\alpha]$, where c^n is a new constant.
5. If $H^n * S^n \in \Gamma$ and S^n is of the form $[\neg\mathbf{A}_{\alpha\beta} = \mathbf{B}_{\alpha\beta}]$, then $H^{n+1} := H^n * \neg[\mathbf{A} = \mathbf{B}] * \neg[\mathbf{A}c^n_\alpha = \mathbf{B}c^n]$, where c^n_α is a new constant.

Note that there always exists a new constant c^n, since H was assumed to be sufficiently pure. We also have to separate the cases 4. and 5., since without 5. we would only obtain the witness $c^n\mathbf{A} \neq c^n\mathbf{B}$ for $\mathbf{A} \neq \mathbf{B}$ instead of $\mathbf{A}c^n \neq \mathbf{B}c^n$.

Next we show by induction that $H^n \in \Gamma$ for all $n \in \mathbb{N}$. The base case holds by definition. The only interesting case for the induction step are the last two, which are analogously done. So let $H^n * S^n \in \Gamma$, where S^n is of the form $\neg\Pi\mathbf{A}$. By construction c^n does not occur anywhere in H^n, so by 5.1.5(7) we have $H^{n+1} \in \Gamma$. Now we define $\mathcal{H} := \bigcup_{n\in\mathbb{N}} H^n$. Since Γ is of finite character, we also have $\mathcal{H} \in \Gamma$.

In order to prove the maximality of $\mathcal{H}$ let $\mathbf{D} \in cwff_o(\Sigma)$ be an arbitrary sentence, such that $\mathcal{H} * \mathbf{D} \in \Gamma$. By definition we know that $\mathbf{D}$ is the n-th term in the above enumeration of sentences for some $n \in \mathbb{N}$. Thus $H^n * \mathbf{D} \subseteq \mathcal{H} * \mathbf{D} \in \Gamma$ and $H^n * \mathbf{D} \in \Gamma$, since Γ is closed under subsets. Hence by definition we know that $\mathbf{D} \in H^{n+1} \subseteq \mathcal{H}$ and therefore $\mathbf{D} \in \mathcal{H}$. □

5.3 Unifying Principle for General Models

Definition 5.3.1 Let Γ be an extensional abstract consistency class and $\mathcal{H}$ be maximal in Γ. Then formulae $\mathbf{A}$ and $\mathbf{B}$ are called **$\mathcal{H}$-congruent** ($\mathbf{A} \sim_{\mathcal{H}} \mathbf{B}$), iff the universal closure of $\mathbf{A} = \mathbf{B}$ is a member of $\mathcal{H}$.

Lemma 5.3.2 (Congruence Lemma) *Let Γ be an abstract consistency class, and let $\mathcal{H} \neq \emptyset$ be maximal in Γ, then $\sim_{\mathcal{H}}$ is a Σ-congruence on $cwff(\Sigma)$.*

Proof: To obtain the assertion we first have to make sure that $\sim_{\mathcal{H}}$ is an equivalence relation. We will only give the tedious details of the proof of symmetry as an example for proofs in abstract consistency classes, since the syntactic manipulations for transitivity and reflexivity are analogous.

Let $[\mathbf{A}_\alpha = \mathbf{B}_\alpha] \doteq [\forall P_{o\alpha}.P\mathbf{A} \Rightarrow P\mathbf{B}] \in \mathcal{H}$ and $\mathbf{Q}_{o\alpha}$ be an arbitrary formula, then by 5.2.2(7) we have $[\neg\mathbf{QA} \Rightarrow \neg\mathbf{QB}] = [[\neg\neg\mathbf{QA}] \vee .\neg\mathbf{QB}] \in \mathcal{H}$. Now by 5.2.2(5) we have to consider two cases. If $\neg\neg\mathbf{QA} \in \mathcal{H}$, then $\mathbf{QA} \in \mathcal{H}$ and therefore $\mathbf{QA} \vee .\neg\mathbf{QB} \in \mathcal{H}$ by 5.2.2(2) and 5.2.2(5). If on the other hand $\neg\mathbf{QB} \in \mathcal{H}$, then $\neg\mathbf{QB} \vee \mathbf{QA} \in \mathcal{H}$. In both cases we have $[\neg\mathbf{QB} \Rightarrow \mathbf{QA}] \in \mathcal{H}$ for all $\mathbf{Q} \in cwff_{o\alpha}(\Sigma)$ and therefore $\forall P_{o\alpha}.P\mathbf{B} \Rightarrow P\mathbf{A} \in \mathcal{H}$ by 5.2.2(7).

Now we will verify the congruence property. By Lemma 3.2.2 we only have to prove that $\mathbf{CA}_\beta \sim_{\mathcal{H}} \mathbf{CB}_\beta$ for all $\mathbf{C} \in cwff_{\alpha\beta}(\Sigma)$, whenever $\mathbf{A} \sim_{\mathcal{H}} \mathbf{B}$. So let $\mathbf{Q}_{o\alpha} \in cwff_{o\alpha}(\Sigma)$ be an arbitrary formula, then $\mathbf{QCA} \Rightarrow \mathbf{QCB} \in \mathcal{H}$ and therefore $\mathbf{CA} = \mathbf{CB} \in \mathcal{H}$, which gives the assertion. □

Remark 5.3.3 Note that in the proof of the congruence lemma we have implicitly used lemma 5.2.5, since we have only considered the congruence properties of $\sim_{\mathcal{H}}$ as given by the presence of some equalities in $\mathcal{H}$. Since we treat equality as an abbreviation of Leibniz' Indecernability formula, the congruence properties follow almost immediately from the use of logical constants and the definition of the abstract consistency class. Thus, with the help of 5.2.5, we do not have to consider the congruence properties of equivalence and the interaction of equivalence and equality.

Definition 5.3.4 (Ground Term Model) If Γ is an extensional abstract consistency class and $\mathcal{H}$ is maximal in Γ, then the quotient algebra $\mathcal{M}^{\mathcal{H}} := (\mathcal{A}^{\mathcal{H}}, \mathcal{I}^{\mathcal{H}})$ with respect to $\mathcal{H}$-congruence is called the **ground term model for $\mathcal{H}$**. Note that $\mathcal{A}^{\mathcal{H}} := cwff(\Sigma)/_{\sim_{\mathcal{H}}}$ and $\mathcal{I}^{\mathcal{H}} := \mathrm{Id}^{\mathcal{H}}$.

Theorem 5.3.5 *Let Γ be an abstract consistency class and let $\mathcal{H}$ be a higher-order Hintikka set for Γ and some sufficiently pure $H \subseteq cwff_o(\Sigma)$, then the ground term model $\mathcal{M}^{\mathcal{H}}$ for $\mathcal{H}$ is a general model for $\mathcal{Q}(\Sigma)$ with $\mathcal{M}^{\mathcal{H}} \models H$.*

Proof: By Lemma 3.2.3 $\mathcal{M}^{\mathcal{H}}$ is a Σ-algebra, and from Lemma 5.2.6 we know that $\sim_{\mathcal{H}}$ has exactly two equivalence classes. Thus we have $\mathcal{D}_o = \{\mathtt{T}, \mathtt{F}\}$, if we define $\mathtt{T} := \pi_{\mathcal{H}}([\mathbf{A} = \mathbf{A}])$ and $\mathtt{F} := \pi_{\mathcal{H}}([\mathbf{A} \neq \mathbf{A}])$.

Obviously $\mathcal{I}$ maps each constant $c_\alpha \in \Sigma_\alpha$ into $\mathcal{D}_\alpha$. We must show, that $\mathcal{I}(\mathbf{Q}_{o\alpha\alpha})$ is the identity relation on $\mathcal{D}_\alpha$. Let $\mathcal{I}(\mathbf{A}) = [\mathbf{A}]$ and $\mathcal{I}(\mathbf{B}) = [\mathbf{B}]$ be two arbitrary members of $\mathcal{D}_\alpha$. By construction $\mathcal{I}(\mathbf{A}) = \mathcal{I}(\mathbf{B})$, iff $[\mathbf{A} = \mathbf{B}] \in \mathcal{H}$, iff $\mathtt{T} = \mathcal{I}([\mathbf{Q}_{o\alpha\alpha}\mathbf{AB}]) = \mathcal{I}(\mathbf{Q}_{o\alpha\alpha})\mathcal{I}(\mathbf{A})\mathcal{I}(\mathbf{B})$, i.e. $\mathcal{I}(\mathbf{Q}_{o\alpha\alpha})$ is indeed the identity relation on $\mathcal{D}_\alpha$.

From $\mathcal{I}_\varphi(\mathcal{H}) = \{\mathtt{T}\}$ for each assignment φ into $\mathcal{D}$ and $H \subseteq \mathcal{H}$ we get $\mathcal{I}_\varphi(H) = \{\mathtt{T}\}$, and therefore $\mathcal{M} \models H$. □

Theorem 5.3.6 (Unifying Principle for General Models) *If Γ is an extensional abstract consistency class and $H \in \Gamma$ is sufficiently pure, then H has a countable general model.*

Proof: We can assume without loss of generality (5.1.9) that Γ is of finite character, so the preconditions of 5.2.7 are met and therefore there exists a higher-order Hintikka set $\mathcal{H}$ for Γ and H. Now we can use the previous theorem to construct a ground term model $\mathcal{M}^{\mathcal{H}}$ for H. If we pay attention to the constructions in the proof of 5.2.7 it is easy to see that $\mathcal{M}^{\mathcal{H}}$ is indeed countable, since the sets of constants and variables both were assumed to be countable. □

Theorem 5.3.7 (Unifying Principle for Intensional Type Theory) *If Γ is an abstract consistency class and $\Phi \in \Gamma$ is a finite set of sentences, then Φ is $\mathfrak{T}$-consistent.*

Proof sketch: (following [And86]) The proof of this theorem is similar to the one presented in this section. We can define higher-order Hintikka sets for abstract consistency classes without extensionality. If we just drop the clause 5. in 5.2.7. we get a proof that higher-order Hintikka sets always exist. From a higher-order Hintikka set $\mathcal{H}$ we can derive a semi-valuation υ that gives rise to a υ-complex, which serves as a kind of non-extensional model for $\mathcal{H}$. Now we convince ourselves that the calculus $\mathfrak{T}$ is sound with respect to the class $\mathcal{K}$ of υ-complexes and obtain the assertion by 4.1.13(2). □

Definition 5.3.8 Let $c_{oo} \in \Sigma_{oo}$ and $b_o \in \Sigma_o$ be constants and $\mathbf{A}_o :\doteq [cb]$ and $\mathbf{B}_o :\doteq [c\neg\neg b]$. Since we will use it as a principle example let us fix the notation $\clubsuit_o :\doteq \mathbf{A} \Rightarrow \mathbf{B}$.

Lemma 5.3.9 *$\mathfrak{T}$ is not complete with respect to general models.*

Proof: We show that the formula $\clubsuit_o$ is not derivable in $\mathfrak{T}$, by convincing ourselves that $\neg\clubsuit_o$ is $\mathfrak{T}$-consistent. Let $c_{oo} \in \Sigma_{oo}$ and $b_o \in \Sigma_o$ be constants and $\mathbf{A}_o := [cb]$ and $\mathbf{B}_o := [c\neg\neg b]$ as in Definition 5.3.8. Furthermore let Γ be the powerset of $\Phi := \{\mathbf{A} \wedge \neg\mathbf{B}, \mathbf{A}, \neg\mathbf{B}, [\neg\neg b_o], b\}$. It is easily checked that Γ is an abstract consistency class by the above definition. However the formula $\neg\clubsuit_o = \mathbf{A} \wedge \neg\mathbf{B}$ is obviously unsatisfiable in the class of general models for $\mathcal{Q}(\Sigma)$. The assertion now follows from the soundness of $\mathfrak{T}$. □

5.4 Completeness

In this section we will use the unifying principles above to give a short and elegant proof for the completeness theorem for $\mathfrak{TE}$. Note that this is not precisely the result in [Hen50], since it holds for the system without description functions.

Definition 5.4.1 We will call a proposition $\mathbf{A}_o$ a **tautology**, iff it is a substitution instance of a proposition $\mathbf{P}_o$ that only contains logical connectives and variables of type o and is valid in all general models. Note that the validity of $\mathbf{P}$ only depends on the assignment for propositional variables in $\mathbf{P}$.

Lemma 5.4.2 (Rule P) *If $\mathbf{A}_o$ is a tautology, then $\vdash_{\mathfrak{T}} \mathbf{A}$ and $\vdash_{\mathfrak{TE}} \mathbf{A}$.*

Proof sketch: Let $\mathbf{P}_o$ be the proposition, such that $\mathbf{A} = \sigma(\mathbf{A})$ and $\mathbf{P}$ only contains propositional variables and connectives. It is well known that the propositional part of $\mathfrak{T}$ and $\mathfrak{TE}$ is complete (see for instance [And86]), so there is a $\mathfrak{T}$-proof $\mathcal{D}: \vdash_{\mathfrak{T}} \mathbf{P}$. One application of the substitution rule now gives the assertion. □

Theorem 5.4.3 *The class $\Delta := \{\Phi \subseteq cwff_o(\Sigma) \mid S$ is $\mathfrak{TE}$-consistent$\}$ is an extensional abstract consistency class.*

Proof: Obviously Δ is closed under subsets, since if a set Φ is $\mathfrak{TE}$-consistent, then every subset of Φ is $\mathfrak{TE}$-consistent. Also by definition no well-formed formula $\mathbf{A}_o$ can be in a $\mathfrak{TE}$-consistent set along with its negation $\neg\mathbf{A}$, this establishes 5.1.5(1). To verify 5.1.5(3), 5.1.5(5) and 5.1.5(6) we note that if $\vdash_{\mathfrak{TE}} \mathbf{C} \Rightarrow \mathbf{D}^1 \wedge \ldots \wedge \mathbf{D}^n$, $\mathbf{C} \in \Phi$ and furthermore Φ is $\mathfrak{TE}$-consistent, then $\Phi \cup \{\mathbf{D}^1, \ldots, \mathbf{D}^n\}$ must be $\mathfrak{TE}$-consistent. For if $\Phi \cup \{\mathbf{D}^1, \ldots, \mathbf{D}^n\}$ were $\mathfrak{TE}$-inconsistent, then $\Phi \cup \{\neg\mathbf{D}^1, \ldots, \neg\mathbf{D}^n\}$ must be $\mathfrak{TE}$-inconsistent by 4.1.13(3), so $\Phi \vdash_{\mathfrak{TE}} \mathbf{C}$, which would entail that Φ were $\mathfrak{TE}$-inconsistent. The observation that $\mathbf{A} \wedge \mathbf{B} \Leftrightarrow .\mathbf{A} \Leftrightarrow \mathbf{B}$ is tautologous can extend this argument to a proof of 5.1.8(9) and 5.1.8(10).

If Φ is $\mathfrak{TE}$-consistent and $\Phi * \mathbf{A}_o$ and $\Phi * \mathbf{B}_o$ are both $\mathfrak{TE}$-inconsistent, then $\Phi \vdash_{\mathfrak{TE}} \neg\mathbf{A}$ and $\Phi \vdash_{\mathfrak{TE}} \neg\mathbf{B}$, so $\Phi \vdash_{\mathfrak{TE}} \neg[\mathbf{A} \vee \mathbf{B}]$ by rule P, therefore $[\mathbf{A} \vee \mathbf{B}] \notin \Phi$ which is just the contrapositive of 5.1.5(4).

To establish 5.1.5(7) we assume that Φ is $\mathfrak{TE}$-consistent but $\Phi * \neg[\mathbf{A}c]$ is $\mathfrak{TE}$-inconsistent, where c_α is a constant or variable that does not occur in Φ or $\mathbf{A}$. So there is a $\mathfrak{TE}$-derivation $\mathcal{D}: \Phi \vdash_{\mathfrak{TE}} \mathbf{A}c$ by 4.1.13 and rule P. Let $\mathcal{D}'$ be the proof tree that is obtained by exchanging all occurrences of c in $\mathcal{D}$ with a new variable X_α. Since c does not occur in Φ and $\mathbf{A}$ it is immediate that $\mathcal{D}'$ is a $\mathfrak{TE}$-derivation of $\mathbf{A}X$ from Φ. By adding an application of the existential generalization rule to the root of $\mathcal{D}'$ we obtain a $\mathfrak{TE}$-derivation of $\Pi\mathbf{A}$ from Φ, so $\neg[\mathbf{A}c] \notin \Phi$.

To establish 5.1.8(8) we suppose that Φ is $\mathfrak{TE}$-consistent but $\Phi * \neg[\mathbf{A}_{\alpha\beta}c_\beta = \mathbf{B}_{\alpha\beta}c]$ is $\mathfrak{TE}$-inconsistent, where c is a constant or variable that does not occur in Φ, $\mathbf{A}$ or $\mathbf{B}$. By an analogous argument as in the case above we obtain a $\mathfrak{TE}$-derivation $\mathcal{D}'$ of $\mathbf{A} = \mathbf{B}$ from Φ, and conclude that $\neg[\mathbf{A}c = \mathbf{B}c] \notin \Phi$. □

Theorem 5.4.4 (Henkin's Theorem for $\mathfrak{TE}$) *Every $\mathfrak{TE}$-consistent set of sentences has a countable general model.*

Proof: Let Φ be a set of $\mathfrak{TE}$-consistent sentences. First we pass from the signature Σ to a signature $\Sigma^+ := \Sigma \cup \Delta$, where each Δ_α is a countably infinite set of new constant symbols. Obviously Φ does not contain constants from Δ and therefore is sufficiently pure in $\mathit{wff}(\Sigma^+)$. Since $(\mathit{wff}(\Sigma^+), \mathfrak{TE})$ is a conservative extension of $(\mathit{wff}(\Sigma), \mathfrak{TE})$ we have that H is also $(\mathit{wff}(\Sigma^+), \mathfrak{TE})$-consistent (cf. 4.1.10). By 5.4.3 we know that the class of sets of $\mathfrak{TE}$-consistent propositions constitute an extensional abstract consistency class Φ with $\Phi \in \Gamma$ 5.3.6 guarantees a countable general model for Φ. □

Corollary 5.4.5 (Completeness Theorem for $\mathfrak{TE}$) $\mathbf{A}_o \vdash_{\mathfrak{TE}} \mathbf{B}$, *iff* $\mathbf{A} \models_{\mathfrak{TE}} \mathbf{B}$.

Remark 5.4.6 In the light of the previous theorem it is not surprising that we can prove the formula $\clubsuit_o$ of 5.3.8 in $\mathfrak{TE}$. Here we sketch the direct proof. We have $\vdash_{\mathfrak{TE}} b_o \Leftrightarrow .\neg\neg b$ and by extensionality $\vdash_{\mathfrak{TE}} b = .\neg\neg b$, which expands to $\vdash_{\mathfrak{TE}} \forall P_{oo}.Pb \Rightarrow P.\neg\neg b$ and by substitution $\vdash_{\mathfrak{TE}} c_{oo}b \Rightarrow c.\neg\neg b = \clubsuit_o$.

6 Conclusion and Further Work

We have presented a unifying principle for $\mathcal{Q}$, that can serve as a tool for investigating the completeness of machine-oriented calculi for higher-order logic.

In appendix A we give some ideas for a machine-oriented calculus for $\mathcal{Q}$. Even though we do not have a completeness proof, we are confident that the proposed calculus will at least solve the problem of the two-valuedness.

In [And71] P. Andrews has given a simple cut-elimination proof for a system $\mathcal{G}^+$ of higher-order logic without extensionality, by showing that both the system $\mathcal{G}^+$ with cut and the cut-free system $\mathcal{G}$ are complete relative to $\mathfrak{T}$. We conjecture that along these lines it should be easy to construct a cut elimination proof for simple type theory with extensionality. In particular the method above would lead to a proof of cut-elimination in a formulation of type theory with function symbols. The author only knows of proofs in formulations of classical higher-order logic without function symbols (cf. [Tak87, Tak68, Tak70]). There is a cut-elimination for intuitionistic type theory with extensionlity and function symbols in [asabPGar]. Note that the results in [And71] are abstractions of the cut-elimination proof for simple type theory in [Tak67], which was extended to the extensional case in [Tak68]. Therefore we believe that the unifying principle for general models can be correspondingly used.

References

[ALCMP84] Peter B. Andrews, Eve Longini-Cohen, Dale Miller, and Frank Pfenning. Automating higher order logics. *Contemp. Math*, 29:169–192, 1984.

[And71] Peter B. Andrews. Resolution in type theory. *Journal of Symbolic Logic*, 3(36), 1971.

[And72] Peter B. Andrews. General models and extensionality. *Journal of Symbolic Logic*, 37(2):395–397, 1972.

[And86] Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof.* Academic Press, 1986.

[And89] Peter B. Andrews. On connections and higher order logic. *Journal of Automated Reasoning*, 5:257–291, 1989.

[asabPGar] Unknown Author (a student advised by Professor Gandy). *Unknown Title (manuscript without title page in my possession).* PhD thesis, Worcester College, Unknown Year.

[Chu40] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[dB72] Nicolas G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with an application to the church-rosser theorem. *Indagationes Mathematicae*, 34, 1972.

[Fit90] Melvin Fitting. *First-Order Logic and Automated Threorem Proving.* Springer Verlag, 1990.

[Hen50] Leon Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15(2):81–91, 1950.

[Hin55] K. J. J. Hintikka. Form and content in quantification theory. *Acta Philosophica Fennica*, 8:7–55, 1955.

[HS86] J. Hindeley and J. Seldin. *Introduction to Combinators and Lambda Calculus.* Cambridge University Press, 1986.

[Hue72] Gérard P. Huet. *Constrained Resolution: A Complete Method for Higher Order Logic.* PhD thesis, Case Western Reserve University, 1972.

[Hue73] Gérard P. Huet. A mechanization of type theory. In *Proceedings of the Third International Joint Conference on Artificial Intelligence*, pages 139–146, 1973.

[Koh93] Michael Kohlhase. Resolution in order-sorted type theory. submitted to LICS'93, 1993.

[Mil83] Dale Miller. *Proofs in Higher-Order Logic.* PhD thesis, Carnegie-Mellon University, 1983.

[Rus08] Bertrand Russell. Mathematical logic as based on the theory of types. *American Jounal of Mathematics*, XXX:222–262, 1908.

[Smu63] Raymond M. Smullyan. A unifying principle for quantification theory. *Proc. Nat. Acad Sciences*, 49:828–832, 1963.

[Smu68] Raymond M. Smullyan. *First-Order Logic.* Springer Verlag, 1968.

[Sny91] Wayne Snyder. *A Proof Theory for General Unification.* Progress in Computer Science and Applied Logic. Birkhäuser, 1991.

[Tak67] Moto-o Takahashi. A proof of cut-elimination in simple type theory. *Journal of the Mathematical Society of Japan*, 19:399–410, 1967.

[Tak68] Moto-o Takahashi. Cut-elimination in simple type theory with extensionality. *Journal of the Mathematical Society of Japan*, 19, 1968.

[Tak70] Moto-o Takahashi. A system of simple type theory of Gentzen style with inference on extensionality and the cut-elimination in it. *Commentarii Mathematici Universitatis Sancti Pauli*, XVIII(II):129–147, 1970.

[Tak87] Gaisi Takeuti. *Proof Theory.* North Holland, 1987.

Index

Σ-algebra, 9
Σ-congruence, 10
Σ-endomorphism, 9
Σ-epimorphism, 9
Σ-monomorphism, 9
Σ-quasi-algebra, 9
α-conversion, 7
β-equality, 7
β-reduction, 7
$\mathcal{C}$-derivation of $\mathbf{A}$ from $\mathbf{C}^1, \ldots, \mathbf{C}^n$, 13
$\mathcal{H}$-congruent, 21

abstract consistency class, 17, 18
admissible rule of inference, 14
alphabetical variant, 7
Andrews Calculus, 15
antecedent, 13
assertion of a proof node, 13
assignment, 9
atomic formula, 30
axioms, 13

base type symbols, 4
beta-conversion, 15
bound in, 5

calculus for $\mathcal{Q}(\Sigma)$, 13
canonical projection, 10
carrier set, 9
clause, 30
closed formula, 5
closed under subsets, 17
co-extensive, 15
comprehension closed, 9
connective, 6
conservative extension, 13
consistent w.r.t. a higher-order theory, 14
consistent with, 14
constrained clause, 30
constraint, 30
contradiction w.r.t. a higher-order theory, 14
currying, 8

deduction theorem, 14
denotation, 9
derivable rule of inference, 14
domain of a partial function, 8

entails, 12
equational system, 30
expansion, 13
expansion of a theory, 13
extension, 13
extensional Σ-congruence, 10
extensionality axioms, 15
extensionality of equality, 8

finite character, 17
frame, 9
free in, 5
full Σ-algebra, 9
fully invariant, 10
function, 8

general model for $\mathcal{Q}(\Sigma)$, 11
ground term model, 22

higher-order Hintikka set, 19
higher-order theory, 13
homomorphic Extension, 9
homomorphism of Σ-algebras, 9

idempotent substitution, 7
identity substitution, 7
image of a function, 8
implication elimination, 14
implication introduction, 14
inconsistent w.r.t. a higher-order theory, 14
indiscernability, 5
inference rule, 13
interpretation of constants, 9

justification of a proof node, 13

language of a higher-order theory, 13
Leibniz' formulation of equality, 6
literals, 30
logical constants, 6

maximal in Γ, 19
Modus Ponens, 14, 15

parameter, 4
partial function, 8
primitive symbols, 4
proof tree, 13
proposition, 5

quantor, 6
quotient algebra, 11

renaming substitution, 7

satisfiable, 12
satisfies, 12
sentence, 5
signature, 4
solved form, 30
sound w.r.t. a class of models, 14
standard model, 11
substitutable for, 6
substitution, 7
substitution rule, 15
succedent, 13
sufficiently pure, 17
support of a proof node, 13
support of a substitution, 7

tautology, 23
term algebras, 9
theorem of $\mathcal{T}$, 13
total function, 8
type symbol, 4
typed binary relation, 8
typed collection of sets, 8
typed map, 8

unifier, 30
universal generalization, 15

valid, 12
value, 9
variable capture, 6
variables, set of, 4

well-formed formulae, 4

7 The Index of defined symbols

$\mathcal{T}_0$ base type symbols 4

$\mathcal{T}$ set of type symbols 4

Σ, Σ_α Signature (of type α) 4

$\mathcal{V}, \mathcal{V}_\alpha$ set of variables (of type α) 4

$q_{o\alpha\alpha}$ equality constant 4

$\mathit{wff}(\Sigma), \mathit{wff}_\alpha(\Sigma)$ set of well-formed formulae (of type α) 4

$\mathbf{Var}(\mathbf{A})$ set of variables of $\mathbf{A}$ 5

$\mathbf{Free}(\mathbf{A})$ set of free variables of $\mathbf{A}$ 5

$\mathbf{Bound}(\mathbf{A})$ set of bound variables of $\mathbf{A}$ 5

$\mathit{cwff}_\alpha(\Sigma), \mathit{cwff}(\Sigma)$ set of closed well-formed formulae 5

$\doteq$ syntactic equality 5

$\mathrm{supp}(\sigma)$ support of the substitution σ 7

$\mathbf{SUB}(\Sigma)$ set of substitutions 7

$\mathbf{A}{=_\alpha}\mathbf{B}$ $\mathbf{A}$ and $\mathbf{B}$ are alphabetical variants 7

$=_\alpha$ α-conversion 7

$=_\beta$ β-equality 7

$\mathbf{Dom}(f)$ domain of the function f 8

$\mathbf{Im}(f)$ image of the function f 8

$\mathcal{F}(A_1, \ldots, A_n; B)$ family of n-ary total functions 8

$\mathcal{I}_\varphi$ homomorphic extension of an assignment φ to $\mathit{wff}(\Sigma)$ 9

$\mathcal{A}/_\sim$ quotient algebra for the Σ-congruence $\sim$ 11

$\mathcal{M} \models_\varphi \mathbf{A}$ φ satisfies $\mathbf{A}$ in a model $\mathcal{M}$ 12

$\mathcal{M} \models \mathbf{A}$ $\mathbf{A}$ is valid in a model $\mathcal{M}$ 12

$\models_\mathcal{K} \mathbf{A}$ $\mathbf{A}$ is valid in a class $\mathcal{K}$ of models 12

$\mathbf{A} \models_\mathcal{K} \mathbf{B}$ $\mathbf{A}$ entails $\mathbf{B}$ in a class of models $\mathcal{K}$ 12

$\mathcal{L}(\mathcal{T})$ language of a theory $\mathcal{T}$ 13

$\Phi \vdash_\mathcal{T} \mathbf{A}$ derivation of $\mathbf{A}$ from Φ in a higher-order theory $\mathcal{T}$ 13

$\vdash_{\mathcal{T}}$ **A** **A** is a theorem of $\mathcal{T}$ 13

$\mathfrak{T}$ Andrews Calculus 15

A $\sim_{\mathcal{H}}$ **B** **A** is $\mathcal{H}$-congruent to **B** 21

$\clubsuit_o$ principal counterexample 23

A A Resolution Calculus for $\mathcal{Q}$

In this section we will present a variant of Huet's *Constrained Resolution* calculus [Hue72] has additional inference rules to deal with extensionality.

As we have seen in Example 5.3.8 in extensional calculi we have to deal with propositions that in the arguments of function constants. The simplest approach to build a calculus that can refute $\neg\clubsuit_o$ is simply to add the equational theory $b = \neg\neg b$ to higher-order unification, This approach is intuitive, but it does not solve the general problem of incorporating extensionality into resolution. In fact we can generalize $\clubsuit_o = [cb] \vee \neg[c[\neg\neg b]]$ to $\clubsuit_o{}'[c\mathbf{A}_o] \vee \neg[c\mathbf{B}_o]$, where **A** and **B** are arbitrary proposition. Now $\clubsuit_o{}'$ is valid, iff $\mathbf{A} \Leftrightarrow \mathbf{B}$ is valid. So the approach of enhancing the unification would require augmenting the unification procedure with the theory of logical equivalence, which would enable the unification procedure to prove any theorem by unifying it with $\top_o$.

To make these ideas more precise let us digress to to a more general look at automatic theorem proving. Theorem proving is a syntactic process of making judgments about the validity of formulae in all models.

In propositional logic formulae are built up from propositional variables and the logical connectives $\neg$ and $\vee$. While the variables can be arbitrarily interpreted (to be either T or F), the connectives $\neg$ and $\vee$ are interpreted to denote the negation and disjunction functions on the set of truth values. Thus the class of models consists only of the $\{\neg, \vee\}$-algebra with carrier set $\mathcal{D}_o = \{\mathtt{T}, \mathtt{F}\}$, where $\mathcal{I}(\neg)$ and $\mathcal{I}(\vee)$ are the well-known functions.

In first order logic there is a clear conceptual distinction between terms (syntactic objects that denote individuals) and formulae (syntactic objects that denote truth values). Formulae are built up from atoms, the symbols $\neg$ and $\vee$ and quantification. Atoms take the place of propositional variables whereas $\neg$, $\vee$ and quantification have fixed interpretations. Atoms are built up from predicate symbols and terms, which in turn are built up from function symbols, individual constants and variables, all of which can be freely interpreted. Thus the class of models for first-order logic consists of some universe $\mathcal{D}_\iota$ of individuals and $\mathcal{D}_o$ with a fixed interpretation for $\neg$, $\vee$ and quantification.

By the use of skolemization in refutation-based theorem proving, the quantification can be eliminated into a preprocess. For instance resolution-based calculi consist of the propositional rules (computation in the fixed part $\mathcal{D}_o$) and the unification procedure, which amounts to solving term equations in all models. Since the term algebra is the free algebra, it is sufficient to solve the term equations there.

Let us summarize. Due to the strong division of the model theory into a fixed part $\mathcal{D}_o$ and a free part $\mathcal{D}_\iota$, first-order theorem proving can be divided into a propositional part (acting on formulae) and a term part (unification), which do not interfere.

In higher-order logic (here simple type theory) we do not have this clear division. In particular there are formulae, where symbols with fixed interpretation are dominated (in the scope or subterms of arguments) by symbols with flexible interpretation.

We propose a calculus where the unfication procedure calls the theorem proving procedure recursively on demand that is whenever it encounters a propositional pair. This approach makes it necessary to break down the distinction between unification and resolution and treat both processes in one uniform calculus.

A.1 Resolution in Extensional Type Theory

First we will need some facts and notation for higher-order unification, for a discussion see eg. [Sny91].

Definition A.1.1 (Equational System) An **equational systemis** a finite set of pairs of well-formed formulae of the form $\Gamma = \{\langle \mathbf{A}^i, \mathbf{B}^i \rangle \mid i = 1, \ldots, n\}$. A substitution $\sigma \in \mathbf{SUB}(\Sigma)$ is called a **unifier of a pair** $\langle \mathbf{A}, \mathbf{B} \rangle$, if $\sigma(\mathbf{A}) = \sigma(\mathbf{B})$ and a **unifier of** Γ, iff σ is a unifier of all pairs $\langle \mathbf{A}^i, \mathbf{B}^i \rangle$. We denote the set of Σ-unifiers of Γ by $\mathbf{U}\Gamma$.

A pair $\langle X_\alpha, \mathbf{B} \rangle$ is in **solved form** in an equational system Γ, iff X_α is a variable, which does not occur anywhere else in Γ. Obviously a system $\Gamma = \{\langle X^1, \mathbf{A}^1 \rangle, \ldots, \langle X^n, \mathbf{A}^n \rangle\}$ is in solved form corresponds to a substitution $\theta := \{X^1 \mapsto \mathbf{A}^1, \ldots, X^n \mapsto \mathbf{A}^n\}$. We write $\langle \theta \rangle := \Gamma$ and $\sigma_\Gamma := \theta$ and note that σ_Γ is the most general unifier for Γ.

Theorem A.1.2 (Completeness Theorem for Higher-Order Unification) *For any unification problem Γ and any $\theta \in \mathbf{U}\Gamma$, there is a sequence $\Gamma \overset{*}{\Longrightarrow} \Delta$ of unification transformations, such that Δ is in Σ-solved form and σ_Δ is more general than θ.*

Definition A.1.3 (Constrained Clause) Let $\mathbf{A}_o$ be an application, constant or variable of type o, such that $\mathbf{head}(\mathbf{A})$ is not one of the logical constants, then $\mathbf{A}$ is called **atomic**. Atoms and their negations are together called **literals**, and finite sets of literals are called **clauses**.

A pair $\mathcal{C} = C || \Gamma$, where C is a clause and Γ is an equational system is called a **constrained clause**, and C is called the **clause** of $\mathcal{C}$ and Γ the **constraint** of $\mathcal{C}$.

Just as in first-order logic we have that each set Φ of sentences can effectively be transformed into a set of clauses $\mathbf{ICCNF}(\Phi)$ that are satisfiable, iff Φ is.

Definition A.1.4 ($\mathcal{HR}$-Refutation) Let $\Box$ stand for any constrained clause $\emptyset || \Gamma$, where Γ is an equational system in pre-solved form. A derivation of $\Box$ from a set $\mathcal{C}$ of constrained clauses with the inference rules below is called a **refutation of** $\mathcal{C}$.

$$\frac{\{\mathbf{N}^1, \ldots, \mathbf{N}^n\} || \Gamma \quad \{[\neg \mathbf{M}^1], \mathbf{M}^2, \ldots, \mathbf{M}^m\} || \Delta}{\{\mathbf{N}^2, \ldots, \mathbf{N}^n, \mathbf{M}^2, \ldots, \mathbf{M}^m\} || \Gamma \cup \Delta * \langle \mathbf{N}^1, \mathbf{M}^1 \rangle} \; \mathcal{HR}(Res)$$

$$\frac{\{\mathbf{N}^1, \ldots, \mathbf{N}^n\} || \Gamma}{\{\mathbf{N}^2, \ldots, \mathbf{N}^n\} || \Gamma * \langle \mathbf{N}^1, \mathbf{N}^2 \rangle} \; \mathcal{HR}(Fac)$$

$$\frac{\{\mathbf{M}^1, \ldots, \mathbf{M}^m\} || \Gamma}{\{\mathbf{M}^1, \ldots, \mathbf{M}^m\} || \Gamma \cup \langle \sigma \rangle} \; \mathcal{HR}(Prim)$$

where in $\mathcal{HR}(Prim)$ one $\mathbf{M}^i$ is a flexible literal of the form $P\overline{\mathbf{U}^k}$ and $\sigma = \{P \mapsto \mathbf{P}\}$ and $\mathbf{P} \in \{[\lambda \overline{X^k}.[H^1\overline{X}] \vee [H^2\overline{X}]], [\lambda \overline{X^k}.\neg[H^1\overline{X}]], [\lambda \overline{X^k}.\forall Y\, H^1\overline{X}Y], [\lambda \overline{X^k}.X^j\overline{X}]\}$

$$\frac{\{\mathbf{M}^1,\ldots,\mathbf{M}^m\}||\Gamma}{\{\mathrm{CNF}(\sigma(\mathbf{M}^1) \vee \ldots \vee \sigma(\mathbf{M}^m))\}||\Gamma}\ \mathcal{HR}(Solv) \qquad \frac{\mathbf{C}||\Gamma}{\mathbf{C}||\Gamma'}\ \mathcal{HR}(\Sigma\mathcal{PU})$$

if $\langle\sigma\rangle \subseteq \Gamma$ and X is free in the $\mathbf{M}^i$ and Γ' can be obtained from Γ by a unification transformation from [Sny91]. Note that in contrast to Huet's calculus we use Andrews' primitive substitutions [And89] instead of splitting rules and that we are able to perform unification everywhere in the deduction in contrast to only at the end. The rule $\mathcal{HR}(Solv)$ will propagate partial solutions from the constraints to the clause part and thus help detect clashes early. Since the substitution may well change the propositional structure of the clause by instantiating a predicate variable we have to renormalize the clause on the fly.

To account for extensionality we propose the following two rules:

$$\frac{\mathcal{C}||\Gamma * \langle\mathbf{A}_o, \mathbf{B}_o\rangle}{\mathrm{CNF}(\mathbf{A} \Leftrightarrow \mathbf{B} \Rightarrow \bigvee\mathcal{C})||\Gamma}\ \mathcal{ER}(Ref)$$

$$\frac{\mathcal{C} * \neg P_{o(\alpha\beta)}\mathbf{A}_{\alpha\beta} * P\mathbf{B}_{\alpha\beta}||\Gamma}{\mathcal{C} * \neg Q_{o\alpha}\mathbf{A}c_\alpha * Q\mathbf{B}c||\Gamma}\ \mathcal{ER}(Ext)$$

where Q is a new variable and c is a new constant. Obviously the first rule amounts to the recursive call of the refutation procedure.

We will call a set Φ of well-formed sentences $\mathcal{HR}$**-refutable**, iff $\Box$ is derivable from the set of constrained clauses $\mathrm{CNF}(\Phi)||\emptyset$. Note that $\Box$ denotes falsehood or contradiction, so by A.1.5 a refutation of a set of sentences Φ proves the unsatisfiability of Φ.

Theorem A.1.5 (Soundness) *The $\mathcal{HR}$ calculus is sound.*

Proof sketch: It is well known that naive skolemization in type theory is not sound. In fact it is possible to prove an instance of the axiom of choice, which is known to be independent, in a resolution system with naive skolemization. In his thesis [Mil83] Dale Miller gives a sound version of skolemization in the context of expansion trees and higher-order matings. The idea is to restrict the unification algorithm, such that only formulae can be produced by substitution where the Skolem functions always have enough arguments. This method can also be utilized in the resolution context and yields a soundness theorem. $\Box$