Higher-Order Automated Theorem Proving

Michael Kohlhase
Fachbereich Informatik, Universität des Saarlandes, Germany
kohlhase@@cs.uni-sb.de

December 7, 2008

1 Introduction

The history of building automated theorem provers for higher-order logic is almost as old as the field of deduction systems itself. The first successful attempts to mechanize and implement higher-order logic were those of Huet [13] and Jensen and Pietrzykowski [17]. They combine the resolution principle for higher-order logic (first studied in [1]) with higher-order unification. The unification problem in typed λ-calculi is much more complex than that for first-order terms, since it has to take the theory of $\alpha\beta\eta$-equality into account. As a consequence, the higher-order unification problem is undecidable and sets of solutions need not even always have most general elements that represent them. Thus the mentioned calculi for higher-order logic have take special measures to circumvent the problems posed by the theoretical complexity of higher-order unification.

In this paper, we will exemplify the methods and proof- and model-theoretic tools needed for extending first-order automated theorem proving to higher-order logic. For the sake of simplicity take the tableau method as a basis (for a general introduction to first-order tableaux see part I.1) and discuss the higher-order tableau calculi $\mathcal{HT}$ and $\mathcal{HTE}$ first presented in [19]. The methods in this paper also apply to higher-order resolution calculi [1, 13, 6] or the higher-order matings method of Peter [3], which extend their first-order counterparts in much the same way.

Since higher-order calculi cannot be complete for the standard semantics by Gödel's incompleteness theorem [11], only the weaker notion of Henkin models [12] leads to a meaningful notion of completeness in higher-order logic. It turns out that the calculi in [1, 13, 3, 19] are not Henkin-complete, since they fail to capture the extensionality principles of higher-order logic. We will characterize the deductive power of our calculus $\mathcal{HT}$ (which is roughly equivalent to these calculi) by the semantics of functional Σ-models.

To arrive at a calculus that is complete with respect to Henkin models, we build on ideas from [6] and augment $\mathcal{HT}$ with tableau construction rules that use the extensionality principles in a goal-oriented way.

2 Higher-Order Logic

We will use a formulation $\mathcal{HOL}$ of higher-order logic which is essentially the Andrews/Henkin version [12, 1, 2] of simple type theory.

2.1 The System $\mathcal{HOL}$

For the logical system $\mathcal{HOL}$, we specialize the simply typed λ-calculus (see Chapter I.2.12 Section 2) by assuming special types and constants. In particular, we assume that the set $\mathcal{B}$ of *base types* is $\{o, \iota\}$ where the base type ι stands for the set of *individuals*, and the type o for the *truth values*. These types alleviate the need for the syntactic categories of "terms" and "formulae" that are necessary for first order logic, since in $\mathcal{HOL}$, the equivalent of first-order "terms" are just the formulae of type ι and that of first-order "formulae", those of type o. Correspondingly, we call a well-formed formula of type o a *proposition*, and a closed proposition a *sentence*. In contrast to the exposition in I.2.12, we will denote the types of formulae in the index unless they are clear from the context.

Furthermore, we will assume the existence of the usual connectives and quantifiers. In particular, we assume all signatures to contain the declarations

$$\Sigma^{\mathcal{HOL}} := \{[\neg\colon o \to o], [\vee\colon o \to o \to o]\} \cup \{[\Pi^\alpha\colon (\alpha \to o) \to o] \big| \alpha \in \mathcal{T}\}$$

We call the constants $\neg, \vee, \Pi^\alpha$ *logical constants*, since they will have a fixed interpretation in the respective semantics. More specifically $\neg, \vee$ are called *connectives* and the Π^α a *quantifier*. We can obtain the connectives $\vee, \Rightarrow, \Leftrightarrow$ from the connectives defined so far, for instance, we take $\mathbf{A} \Rightarrow \mathbf{B}$ as an abbreviation for $\neg(\mathbf{A} \wedge \neg\mathbf{B})$. Finally, we use the infix notation for connectives, and write $\forall X_\alpha.\mathbf{A}$ as an abbreviation for $\Pi^\alpha(\lambda X_\alpha.\mathbf{A})$.

Non-logical constants are called *parameters*, since the choice of parameters determines the particular formulation of the logical system $\mathcal{HOL}$. To simplify the model constructions, we assume the existence of infinitely many parameters $w^i_\alpha \in \mathcal{W}_\alpha$ which we will call *witness parameters* per type α. We will denote the set of witness parameters occurring in a formula $\mathbf{A}$ with $\mathcal{W}(\mathbf{A})$ and call all other parameters *proper*.

In order to get a feeling for the expressivity of higher-order logic, let us look at a few mathematical examples.

Example 2.1 (Peano Axioms) The natural numbers are formalized using the constant 0_ι for the number zero, and the constant $s_{\iota\to\iota}$ for the successor function. The set of natural numbers is represented by the predicate constant $\mathbb{N}_{\iota\to o}$ and the following set of axioms.

1. $\mathbb{N}0$ (zero is a natural number)
2. $\forall X_\iota.(\mathbb{N}X) \Rightarrow \mathbb{N}(sX)$ (the successor of a natural number is a natural number)
3. $\neg\exists X_\iota.(\mathbb{N}X) \wedge sX = 0$ (zero has no predecessor)

4. $\forall X_\iota, Y_\iota.(sX = sY) \Rightarrow (X = Y)$ (the successor function is one-one)

5. $\forall P_{\iota\to o}.(P0 \wedge \forall Y_\iota.PX \Rightarrow P(sX)) \Rightarrow (\forall Z_\iota.\mathbb{N}Z \Rightarrow PZ)$
   (Induction Axiom: all properties P that hold on zero and with every natural number hold on its successor must hold on all natural numbers.)

Even complex assertions, such as Cantor's theorem have a simple representation in $\mathcal{HOL}$.

Example 2.2 (Cantor's Theorem) The following formula is a variant of Cantor's theorem of the uncountability of $\mathbb{R}$.

$$\neg\exists F_{\iota\to\iota\to\iota}.(\forall Z_\iota, W_\iota.\mathbb{N}Z \wedge \mathbb{N}W \Rightarrow \mathbb{N}(FZW) \wedge \\ \forall G_{\iota\to\iota}.\forall Z_\iota.(\mathbb{N}Z \Rightarrow \mathbb{N}(GZ)) \Rightarrow \exists J_\iota.\mathbb{N}J \wedge FJ = G)$$

The mapping F – that is claimed not to exist – maps natural numbers (type ι) to functions (type $\iota \to \iota$), thus it has type $\iota \to \iota \to \iota$. The subformula beginning with the first universal quantifier says that F is onto (surjective: every function G has a pre-image J_ι).

2.2 Semantics

When evaluating calculi for higher-order logic, the classical notion of completeness becomes problematic, since higher-order logic cannot admit complete calculi according to Gödel's first incompleteness theorem [11]. At closer view, Gödel's theorem only applies to the so-called standard semantics, where a model consists of a given universe $\mathcal{D}_\iota$ of individuals, the set $\mathcal{D}_o$ of truth values, and universes $\mathcal{D}_{\alpha\to\beta}$ for the function types that are just the sets of *all* functions with domain $\mathcal{D}_\alpha$ and codomain $\mathcal{D}_\beta$. While this semantics is indeed the intuitive semantics for mathematics, it does not yield a reasonable measure for the completeness of a calculus. If we consider a generalized notion of model theory, the so-called *Henkin models*, where the universes of functional type are only required to be subsets of the set of all functions such that there exists a denotation for any well-formed formula, then appropriate generalizations of first-order calculi are complete [12]. Clearly each standard model is a Henkin model. Moreover, there are now so many new models, that all propositions that are valid (in the standard sense) but not provable, now have a counterexample. Thus this semantics yields an appropriate measure of completeness for higher-order calculi. Fortunately, the corresponding notion of soundness entails that of standard soundness, since each standard model is a Henkin model by definition. Furthermore, by Gödel's second incompleteness theorem, formal methods cannot characterize the standard models in the class of Henkin models.

We will now formally develop the semantics of $\mathcal{HOL}$. For this, we first give a set-theoretic semantics (Σ-algebras) for the simply typed λ-calculus and use that as a basis for the definition of Σ-models, which we further specialize to Henkin- and standard models. For details see [5].

Definition 2.3 (Σ-Algebra) Let $\mathcal{D} = \{\mathcal{D}_\alpha \big| \alpha \in \mathcal{T}\}$ be a collection of sets, such that $\mathcal{D}_{\alpha\to\beta} \subseteq \mathcal{F}(\mathcal{D}_\alpha;\mathcal{D}_\beta) = \{\mathsf{f}\big|\mathsf{f}\colon\mathcal{D}_\alpha \to \mathcal{D}_\beta\}$ and let $\mathcal{I}\colon\Sigma \to \mathcal{D}$ be a type-preserving mapping ($\mathcal{I}(\Sigma_\alpha) \subseteq \mathcal{D}_\alpha$)), then we call the pair $\mathcal{A}\colon= (\mathcal{D},\mathcal{I})$ a *pre-Σ-algebra*. The collection $\mathcal{D}$ is called the *frame*, the set $\mathcal{D}_\alpha$ the *universe* of type α, and the function $\mathcal{I}$ the *interpretation* of constants.

We call a type-preserving function φ from the set V of variables into $\mathcal{D}$ a *variable assignment*. We will denote the assignment ψ with $\psi(X) = \mathsf{a}$ that coincides with φ everywhere else with $\varphi,\{X \mapsto \mathsf{a}\}$. The *homomorphic extension* $\mathcal{I}_\varphi$ of φ is inductively defined to be a type-preserving partial function from well-formed formulae to $\mathcal{D}$, 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. If there is a function f in $\mathcal{D}_{\alpha\to\beta}$ such that $\mathsf{f}(\mathsf{a})\colon= \mathcal{I}_{\varphi,\{X\mapsto\mathsf{a}\}}(\mathbf{B})$, then $\mathcal{I}_\varphi(\lambda X_\alpha.\mathbf{B}_\beta) = \mathsf{f}$ else it is undefined.

We call $\mathcal{I}_\varphi(\mathbf{A}_\alpha) \in \mathcal{D}_\alpha$ the *value* or *denotation* of $\mathbf{A}_\alpha$ in $\mathcal{A}$ for φ.

A pre-Σ-algebra $\mathcal{A} = (\mathcal{D},\mathcal{I})$ is called Σ-*algebra*, iff for each assignment φ into $\mathcal{A}$ the homomorphic extension $\mathcal{I}_\varphi$ is defined for any well-formed formula. These closure conditions for the carrier set $\mathcal{D}$ of $\mathcal{A}$ assures that the universes of functions $\mathcal{D}_{\alpha\to\beta}$ are rich enough to contain a value for all well-formed formulae $\mathbf{A}$. Note that this requirement directly corresponds to the so-called *comprehension axioms* of higher-order logic and set-theory.

Example 2.4 (Term Algebra) Let $\mathcal{D}$ be the collection sets of of well-formed formulae in $\beta\eta$-normal form, let function application be defined such that $\mathbf{A}(\mathbf{B})$ is the $\beta\eta$-normal form of $\mathbf{AB}$, and $\mathcal{I}\colon= Id_\Sigma$, then we call $\mathcal{TS}(\Sigma)\colon= (\mathcal{D},\mathcal{I})$ the *term algebra* for Σ. Note that assignments into $\mathcal{TS}(\Sigma)$ are just substitutions, and $\mathcal{I}_\sigma(\mathbf{A})$ is the $\beta\eta$-normal form of $\sigma(\mathbf{A})$.

Definition 2.5 (Valuation) Let $\mathcal{A} = (\mathcal{D},\mathcal{I})$ be a Σ-algebra, then a surjective total function $v\colon\mathcal{D}_o \to \{\mathsf{T},\mathsf{F}\}$ is called a *valuation* for $\mathcal{A}$, iff

1. $v(\mathcal{I}(\neg)(\mathsf{a})) = \mathsf{T}$, iff $v(\mathsf{a}) = \mathsf{F}$,
2. $v(\mathcal{I}(\vee)(\mathsf{a},\mathsf{b})) = \mathsf{T}$, iff $v(\mathsf{a}) = \mathsf{T}$ or $v(\mathsf{b}) = \mathsf{T}$,
3. $v(\mathcal{I}(\Pi^\alpha)(\mathsf{f})) = \mathsf{T}$, iff $v(\mathsf{f}(\mathsf{a})) = \mathsf{T}$ for each $\mathsf{a} \in \mathcal{D}_\alpha$

The notion of valuation intuitively gives a truth-value interpretation to the domain $\mathcal{D}_o$ of a Σ-algebra, which is consistent with the intuitive interpretations of the logical constants. Since models are semantic entities that are constructed to make statements about truth and falsity of formulae, the requirement that there exists a valuation is perhaps the most general condition under which one

wants to speak of a model.[1] Thus we will define our most general notion of semantics as Σ-algebras that have valuations.

Definition 2.6 (Σ-model) Let $\mathcal{A} = (\mathcal{D}, \mathcal{I})$ be a Σ-algebra and υ be a valuation for $(\mathcal{D}, \mathcal{I})$, then we call the triple $\mathcal{M} := (\mathcal{D}, \mathcal{I}, \upsilon)$ a Σ-*model*.

If for every type α, the universe $\mathcal{D}_{\alpha\to\alpha\to o}$ contains a relation r, such that for all $\mathsf{a}, \mathsf{b} \in \mathcal{D}_\alpha$, $\upsilon(\mathsf{r}(\mathsf{a}, \mathsf{b})) = \mathsf{T}$, iff $\mathsf{a} = \mathsf{b}$, then we call $\mathcal{M}$ a Σ-*model with equality*. Σ-models without equality can have counterintuitive properties (see [2] for a discussion in the context of Henkin models and [5] for a full development of the theory), so we will always consider Σ-models with equality in this paper.

It is a matter of folklore that in $\mathcal{HOL}$, equality can be defined from the connectives and quantifiers.

Definition 2.7 (Leibniz' Formulation for Equality) We define the *Leibniz formula* for equality by

$$\mathbf{Q}^\alpha := (\lambda X_\alpha Y_\alpha . \forall P_{\alpha\to o} . PX \Rightarrow PY)$$

With this definition, $\mathbf{Q}^\alpha\mathbf{AB}$ β-reduces to $\forall P_{\alpha\to o}.(P\mathbf{A}) \Rightarrow (P\mathbf{B})$, which can be interpreted as: formulae $\mathbf{A}$ and $\mathbf{B}$ are not equal, iff there exists a discerning property P. In other words, $\mathbf{A}$ and $\mathbf{B}$ are equal, if they are indiscernible. Note that we do not need equivalence in the definition, since P can be instantiated with $\lambda Z_\alpha.\neg R_{\alpha\to o} Z$ for some new variable R, which gives the converse direction of the implication.

Since the formula $\mathbf{Q}^\alpha$ is intended to denote the equality relation, we use $\mathbf{A} =^\alpha \mathbf{B}$ or even $\mathbf{A} = \mathbf{B}$ as an abbreviation for $(\mathbf{Q}^\alpha\mathbf{AB})$. Note that we have not included a constant for *primitive equality* in $\Sigma^{\mathcal{HOL}}$, so that all equalities occurring in formulae in this paper are abbreviations for $\mathbf{Q}^\alpha$.

If $\mathcal{M} = (\mathcal{D}, \mathcal{I}, \upsilon)$ is a Σ-model with equality, then $\upsilon(\mathcal{I}_\varphi(\mathbf{Q}^\alpha\mathbf{AB})) = \mathsf{T}$, iff $\mathcal{I}_\varphi(\mathbf{A}) = \mathcal{I}_\varphi(\mathbf{B})$. Note, that the proof of this crucially hinges on the availability of the identity relation in $\mathcal{D}_{\alpha\to\alpha\to o}$, which ensures the singleton sets in $\mathcal{D}_{\alpha\to o}$ which are needed as discerning properties. Thus in Σ-models without equality, $\mathbf{Q}^\alpha$ is not a faithful representation of equality (for details see [5]).

We now present two special classes of Σ-models, which model the intended understanding of $\mathcal{HOL}$. The class of standard models is in some way the most natural notion of semantics for $\mathcal{HOL}$, however, with the notion of completeness induced with this semantics there cannot be complete calculi, a fact that makes it virtually useless for our purposes. The class of Henkin models allows complete calculi and, in fact, we will exhibit one (cf. Section 4).

Definition 2.8 (Henkin model) Let $\mathcal{M} = (\mathcal{D}, \mathcal{I}, \upsilon)$ be a Σ-model with equality, such that $\mathcal{A} = (\mathcal{D}, \mathcal{I})$ is a Σ-algebra, then $\mathcal{M}$ is called a *Henkin model*, iff

[1] There are more general notions, for instance, Peter Andrews uses models, where functionality can break down in [1] (see [5] for a discussion) to establish his "unifying principle", but we will not pursue this in this paper.

$\mathcal{D}_o$ is the set $\{\mathsf{T}, \mathsf{F}\}$ of truth values and v is the identity function on $\mathcal{D}$. Thus v is fixed in Henkin models, and we can fully describe $\mathcal{M}$ by its carrier set $\mathcal{D}$ and its interpretation $\mathcal{I}$. We are striving for a general notion of algebraic model, so we only require $\mathcal{M}$ to be *comprehension-closeds* ($\mathcal{A}$ is a Σ-algebra) and do not require $\mathcal{M}$ to be *full* ($\mathcal{D}_{\alpha\to\beta} = \mathcal{F}(\mathcal{D}_\alpha; \mathcal{D}_\beta)$). A full Henkin model is called a *standard model*.

Remark 2.9 If we were only interested in analyzing $\mathcal{HOL}$ with respect to Henkin semantics, we could have simplified the presentation of the theory by only assuming the equality constants q^α for all types $\alpha \in \mathcal{T}$, defining the other logical constants by the following definitions (due to Peter Andrews), and treating them as defined formulae:

$$\begin{array}{cc} \mathbf{T}_o := q^{\alpha\to\alpha\to o} q^\alpha q^\alpha & \mathbf{F}_o := q^{o\to o}(\lambda X_o.X)(\lambda X_o.\mathbf{T}_o) \\ \neg := (q^o \mathbf{F}_o) & \Pi^\alpha := (q^{\alpha\to o}(\lambda X_\alpha.\mathbf{T}_o)) \\ \multicolumn{2}{c}{\wedge := (\lambda X_o Y_o.q^{(o\to o\to o)\to o}(\lambda G_{o\to o\to o}.G\mathbf{T}_o\mathbf{T}_o)(\lambda G.GXY))} \end{array}$$

Furthermore, we could have weakened the conditions for v to be a valuation by only requiring that $v \circ \mathcal{I}(q^\alpha)$ is the identity relation on $\mathcal{D}_\alpha$, since the definitions above entail that $v := Id_{\mathcal{D}_o}$ is a valuation. Note that a construction like the one above is not possible in the case of Σ-models, since the proof of the condition for $\neg$ requires that $\mathcal{D}_o$ has exactly the elements T and F, as the definition takes all elements that are not T to be false.

Definition 2.10 (Extensionality) We call the following formula schemata

$$\begin{array}{rcl} \mathbf{Ext}^{\alpha\to\beta} & = & \forall F_{\alpha\to\beta}.\forall G_{\alpha\to\beta}(\forall X_\alpha.FX = GX) \Rightarrow F = G \\ \mathbf{Ext}^o & = & \forall F_o.\forall G_o.(F \Leftrightarrow G) \Leftrightarrow F = G \end{array}$$

the *extensionality axioms*. $\mathbf{Ext}^{\alpha\to\beta}$ is called the axiom of *functional extensionality* and specifies that two (mathematical) functions are equal, iff they give the same values on all arguments. This principle is universally assumed in mathematics, but not for instance in the theory of computation, since it would for instance make any two correct sorting algorithms indiscernible.

The second axiom is called the axiom of *propositional extensionality* or *substitutivity of equivalence* and entails that logically equivalent propositions can be substituted for each other in any context. This principle is also usually assumed in higher-order logic, but not for instance in natural language semantics, where the facts that the queen is always the head of the Commonwealth and that Elisabeth is the queen do not entail that Elisabeth is always the head of the Commonwealth.

Note that the equalities in these formulae are abbreviations for the Leibniz formula.

It is easy to check that $\mathbf{Ext}^{\alpha\to\beta}$ is valid in all Σ-models, while $\mathbf{Ext}^o$ is only valid in Henkin Models. In fact we have specifically constructed the Σ-models in order to allow $\mathbf{Ext}^o$ to break down: The first version of our higher-order

tableau calculus will not be able to handle full extensionality. This is not a problem special to our system, since this is also a problem for the calculi of higher-order resolution [13] and for higher-order mating-search [3].

2.3 Model Existence Theorems

Now we introduce model existence theorems for Σ-models and Henkin models as an important tool for proving completeness results in higher-order logic. A more detailed discussion together with further model existence theorems in connection with additional notions of models can be found in [5]. Such theorems were first introduced by Smullyan (who calls them unifying principles) in [24] based on work by Hintikka and Beth and later generalized to higher-order logic by Andrews in [1]. Since there is no simple Herbrand theorem in higher-order logic, Andrews "unifying principle for type theory" from [1] has become the standard method for completeness proofs in higher-order logic, even though it only yields completeness relative to a certain Hilbert-style calculus (that is not complete with respect to Henkin models, since it lacks extensionality). For this paper, we will use a stronger version.

Definition 2.11 (Abstract Consistency Class) Let ∇_Σ be a family of sets of propositions, then ∇_Σ is called an *abstract consistency class*, iff each ∇_Σ is closed under subsets, and satisfies conditions (1) to (8) for all sets $\Phi \in \nabla_\Sigma$. If it also satisfies (9), then we call it *extensional*.

1. If $\mathbf{A}$ is atomic, then $\mathbf{A} \notin \Phi$ or $\neg\mathbf{A} \notin \Phi$.
2. If $\mathbf{A} \in \Phi$ and if $\mathbf{B}$ is the $\beta\eta$-normal form of $\mathbf{A}$, then $\mathbf{B} * \Phi \in \nabla_\Sigma$[2].
3. If $\neg\neg\mathbf{A} \in \Phi$, then $\mathbf{A} * \Phi \in \nabla_\Sigma$.
4. If $\mathbf{A} \vee \mathbf{B} \in \Phi$, then $\Phi * \mathbf{A} \in \nabla_\Sigma$ or $\Phi * \mathbf{B} \in \nabla_\Sigma$.
5. If $\neg(\mathbf{A} \vee \mathbf{B}) \in \Phi$, then $\Phi * \neg\mathbf{A} * \neg\mathbf{B} \in \nabla_\Sigma$.
6. If $\Pi^\alpha\mathbf{A} \in \Phi$, then $\Phi * \mathbf{AB} \in \nabla_\Sigma$ for each closed formula $\mathbf{B} \in \mathit{wff}_\alpha(\Sigma)$.
7. If $\neg\Pi^\alpha\mathbf{A} \in \Phi$, then $\Phi * \neg(\mathbf{A}w_\alpha) \in \nabla_\Sigma$ for any witness constant $w_\alpha \in \mathcal{W}$ that does not occur in Φ.
8. If $\neg(\mathbf{A} =^{\alpha\to\beta} \mathbf{B}) \in \Phi$, then $\Phi * \neg(\mathbf{A}w_\alpha = \mathbf{B}w) \in \nabla_\Sigma$ for any witness constant $w_\alpha \in \mathcal{W}$ that does not occur in Φ.
9. If $\neg(\mathbf{A} =^o \mathbf{B}) \in \Phi$, then $\Phi \cup \{\mathbf{A}, \neg\mathbf{B}\} \in \nabla_\Sigma$ or $\Phi \cup \{\neg\mathbf{A}, \mathbf{B}\} \in \nabla_\Sigma$.

Here, we treat equality as an abbreviation for Leibniz definition. We call an abstract consistency class *saturated*, iff for all $\Phi \in \nabla_\Sigma$ and all propositions $\mathbf{A} \in \mathit{wff}_o(\Sigma)$ we have $\Phi * \mathbf{A} \in \nabla_\Sigma$ or $\Phi * \neg\mathbf{A} \in \nabla_\Sigma$.

[2] We use $A * a$ as an abbreviation for $A \cup \{a\}$.

Theorem 2.12 (Model Existence Theorem) *Let $H \in \nabla_\Sigma$ and ∇_Σ be a saturated abstract consistency class, then there is a Σ-model $\mathcal{M}$ (with equality), such that $\mathcal{M} \models H$. If ∇_Σ is extensional, then there is even a Henkin model for H.*

Proof sketch: For details see [5]. The proof of the first assertion has two stages: First the set H is extended to a *Hintikka set*, i.e. a maximal element $\mathcal{H} \in \nabla_\Sigma$, where for each sentence $\mathbf{D} \in \nabla_\Sigma$ with $\mathcal{H} * \mathbf{D} \in \nabla_\Sigma$, we already have $\mathbf{D} \in \mathcal{H}$. This limit construction, that iteratively adds consistent sentences and witness formulae, is a generalized version of Henkin's in [12]. In the second stage, the special properties of Hintikka sets are exploited to construct a valuation v for the term algebra $\mathcal{TS}(\Sigma)$ ($v(\mathbf{C}) = \mathsf{T}$, if $\mathbf{C} \in \mathcal{H}$ and $v(\mathbf{C}) = \mathsf{F}$, if $\neg\mathbf{C} \in \mathcal{H}$). For instance, in any Hintikka set $\mathcal{H}$ we have $\mathbf{A} \in \mathcal{H}$ iff $\neg\mathbf{A} \notin \mathcal{H}$, so v is a total function. Here we have used the assumption that ∇_Σ is saturated, since the result is false otherwise. Similarly, for Hintikka sets we have $\mathbf{A} \wedge \mathbf{B} \in \mathcal{H}$, iff $\mathbf{A}, \mathbf{B} \in \mathcal{H}$, and similar properties for the other connectives and quantifiers, which entails that v is a valuation $\mathcal{TS}(\Sigma)$. Thus $\mathcal{M} := (\mathcal{TS}(\Sigma), v)$ is a Σ-model with $\mathcal{M} \models \mathcal{H}$ and thus $\mathcal{M} \models H$, since $H \subseteq \mathcal{H}$.

Note that this proof does not give us a Henkin model, since the set $\mathcal{D}_o$ is the set of all $\beta\eta$-normal sentences and not $\{\mathsf{T}, \mathsf{F}\}$. For a Henkin model, we extract a congruence $\sim_\mathcal{H}$ from the Hintikka set $\mathcal{H}$ and take the quotient algebra of $\mathcal{TS}(\Sigma)$ with respect to $\sim_\mathcal{H}$. For an extensional Hintikka set $\mathcal{H}$ we have $\forall X.(\mathbf{A}X = \mathbf{B}X) \in \mathcal{H}$, implies $(\mathbf{A} = \mathbf{B}) \in \mathcal{H}$ and $(\mathbf{A} \Leftrightarrow \mathbf{B}) \in \mathcal{H}$ implies $(\mathbf{A} = \mathbf{B}) \in \mathcal{H}$, which in turn entails that for any propositions $\mathbf{A}$, and $\mathbf{B}$ either $\mathbf{A} = \mathbf{B} \in \mathcal{H}$ or $\mathbf{A} = \neg\mathbf{B} \in \mathcal{H}$. So, if we say that formulae $\mathbf{A}$ and $\mathbf{B}$ are *$\mathcal{H}$-congruent* ($\mathbf{A} \sim_\mathcal{H} \mathbf{B}$), iff the universal closure of $\mathbf{A} = \mathbf{B}$ is a member of $\mathcal{H}$, then $\sim_\mathcal{H}$ is a congruence relation on *wff*(Σ). Thus the desired model is the the quotient algebra $\mathcal{M}^\mathcal{H} = \mathcal{TS}(\Sigma)/_{\sim_\mathcal{H}} = (\mathcal{D}^\mathcal{H}, \mathcal{I}^\mathcal{H})$ with respect to the $\mathcal{H}$-congruence $\sim_\mathcal{H}$. We have $\mathcal{D}_o = \{\mathsf{T}, \mathsf{F}\}$, if we define $\mathsf{T} := [\![\mathbf{A} \vee \neg\mathbf{A}]\!]$ and $\mathsf{F} := [\![\mathbf{A} \wedge \neg\mathbf{A}]\!]$ for some atomic sentence. Then we verify that $\mathcal{I}(\mathbf{Q}^\alpha)$ is indeed the identity relation on $\mathcal{D}_\alpha$, and $\mathcal{M}^\mathcal{H}$ is a Henkin model.

We have $\mathcal{I}_\varphi(\mathcal{H}) = \{\mathsf{T}\}$ for each assignment φ into $\mathcal{D}$, since $\mathbf{A} \vee \neg\mathbf{A} \in \mathcal{H}$. Furthermore, we have $H \subseteq \mathcal{H}$, hence we get $\mathcal{I}_\varphi(H) = \{\mathsf{T}\}$, and therefore $\mathcal{M} \models H$. □

3 Higher-Order Tableaux

Now that we have specified the semantics we can turn to the exposition of our tableau calculi $\mathcal{HT}$ and $\mathcal{HTE}$. Weaker variants of these calculi have been first considered in [19], they differ from the present version in the treatment of functional extensionality. In this respect, the calculi from [19] were not complete with respect to Σ-models Christoph Benzmüller has noted this shortcoming and proposed the rule $\mathcal{HT}$(*leib*) in [4].

The case of standard tableaux for higher-order logic is a simple extension of first-order tableau methods to the higher-order language. The only significant

difference is that $\alpha\beta\eta$-equality has to be built in by keeping formulae in normal form. Therefore we will only concern ourselves with free variable tableaux. Here there are two main differences to the first-order case.

Higher-order unification is undecidable, therefore we cannot simply use it as a sub-procedure that is invoked for closing branches in tableaux. The solution for this problem is to treat the unification problem as a constraint and residuate it using an explicit $\mathcal{HT}(cut)$ tableau rule.

The second difference is that naive Skolemization is not sound for higher-order logic. It is possible to prove an instance of the axiom of choice that is known to be independent of $\mathcal{HOL}$ using naive Skolemization. Dale Miller has investigated this problem in depth in [22].

For $\mathcal{HT}$, we use a direct encoding of dependencies induced by the sequencing of quantifiers (witnesses for existential quantifications in the scope of a universal quantifier $\forall X$ may not occur in any formula instantiated for X) in form of an explicit relation $\mathcal{R}$ which is similar, but not identical to the approaches in [22, 8]..

Definition 3.1 (Variable Condition, $\mathcal{R}$-Substitution) We will call a partial function $\mathcal{R}$ that maps witness constants to sets of variables a *variable condition*. We will call a witness constant $w \in \mathcal{W}$ $\mathcal{R}$-*illegal* for a variable $X \in V$, iff $X \in \mathcal{R}(w)$ and a formula $\mathbf{A}$ $\mathcal{R}$-legal for X, if $\mathbf{A}$ does not contain witness constants that are $\mathcal{R}$-illegal for X. Finally, a substitution σ is called an $\mathcal{R}$-*substitution*, if $\sigma(X)$ is $\mathcal{R}$-legal for $X \in \mathbf{Dom}(\sigma)$.

For a given variable condition $\mathcal{R}$ and an $\mathcal{R}$-substitution $\{X \mapsto \mathbf{A}\}$ we will often need the following variable condition

$$\mathcal{R}(X \mapsto \mathbf{A})(w) := \begin{cases} \mathcal{R}(w) & \text{if } X \notin \mathcal{R}(w) \\ (\mathcal{R}(w) \setminus \{X\}) \cup \mathbf{Free}(\mathbf{A}) \cup \mathcal{R}(\mathcal{W}(\mathbf{A})) & \text{if } X \in \mathcal{R}(w) \end{cases}$$

For the tableau calculus $\mathcal{HT}$, we will need the notion of general bindings, which we will recapitulate in order to be self-contained.

Definition 3.2 (General Binding) A *general binding* $\mathbf{G}^h_\alpha$ of type $\alpha = \overline{\beta_n} \to \gamma$ with head h is a formula of the following form

$$\mathbf{G}^h_\alpha = \lambda\overline{X^n_\beta}.h(H^1\overline{X})\ldots(H^m\overline{X})$$

where h has the type $\overline{\delta_m} \to \gamma$ and the H^i are new variables of types $\overline{\beta_n} \to \delta_i$. If the head h is the bound variable X^j, then we call $\mathbf{G}$ the *j-projection* binding and we will denote it with $\mathbf{G}^j_\alpha$, else an *imitation* binding. Note that for a given type $\alpha = \overline{\beta_n} \to \gamma$ there are at most n projection bindings. Thus the set

$$\mathcal{A}^h_\alpha(\Sigma) := \{\mathbf{G}^j_\alpha \big| j \leq n\} * \mathbf{G}^h_\alpha$$

of *approximating bindings* is finite for all heads $h \in \Sigma \cup V$ and types α.

The name of general bindings is justified by the following theorem.

Theorem 3.3 (General Binding Theorem) *Let* $\mathbf{A} \in wff_\alpha(\Sigma)$ *be a formula with head* h*, then there exists a substitution* ρ*, such that* $\rho(\mathbf{G}^h_\alpha) =_{\beta\eta} \mathbf{A}$ *where* $\mathbf{G}^h_\alpha$ *is the general binding for* h *and* α*. Moreover, if* $\mathbf{A}$ *is a head normal form, then the depth of* ρ *(i.e. the maximum depth of* $\rho(X)$ *for* $X \in \mathbf{Dom}(\rho)$*) is strictly less than that of* $\mathbf{A}$*.*

Definition 3.4 (Higher-Order Tableau) We will call a proposition $\mathbf{A}^v$, where $v \in \{\mathsf{T}, \mathsf{F}\}$ a *labelled proposition* and a *literal* if $\mathbf{A}$ is *atomic*, i.e. if the head of $\mathbf{A}$ is a parameter or a variable. We will call a pair $\mathbf{A}_\alpha \neq^? \mathbf{B}_\alpha$ of formulae a *unification constraint*.

Let $\mathcal{R}$ be a variable condition. We will call a pair $\mathcal{R}.\mathcal{T}$, where $\mathcal{T}$ is a tree whose nodes are labeled with labeled propositions or unification constraints a *higher-order tableau*, iff it can be constructed by the tableau *construction rules* in figures 1 – 3. See 3.5 for an example.

The structural rules in figure 1 recursively build up the tableau tree by decomposing the logical structure of proper labeled formulae and adding new nodes and branches, managing the variable condition along the way.

$$\frac{(\mathbf{A} \vee \mathbf{B})^\mathsf{T}}{\mathbf{A}^\mathsf{T} \mid \mathbf{B}^\mathsf{T}}\,\mathcal{HT}(\wedge) \qquad \frac{(\neg\mathbf{A})^\mathsf{T}}{\mathbf{A}^\mathsf{F}}\,\mathcal{HT}(\neg^\mathsf{F}) \qquad \frac{(\Pi^\alpha\mathbf{A})^\mathsf{T}}{(\mathbf{A}X_\alpha)^\mathsf{T}}\,\mathcal{HT}(all)$$

$$\frac{(\mathbf{A} \vee \mathbf{B})^\mathsf{F}}{\mathbf{A}^\mathsf{F}\;\mathbf{B}^\mathsf{F}}\,\mathcal{HT}(\vee) \qquad \frac{(\neg\mathbf{A})^\mathsf{F}}{\mathbf{A}^\mathsf{T}}\,\mathcal{HT}(\neg^\mathsf{T}) \qquad \frac{(\Pi^\alpha\mathbf{A})^\mathsf{F}}{(\mathbf{A}w_\alpha)^\mathsf{F}}\,\mathcal{HT}(ex)^1$$

[1] w_α new and $\mathcal{R}^1 = \mathcal{R} * \{w_\alpha \mapsto \mathbf{Free}(\mathbf{A})\}$

Figure 1: Structural Rules of $\mathcal{HT}$

Since higher-order unification is undecidable, we need an explicit rule $\mathcal{HT}(cut)$ (see figure 2) for cutting complementary formulae: If $\mathbf{A}^v$ and $\mathbf{B}^w$ occur in a branch $\mathcal{B}$ of $\mathcal{R}.\mathcal{T}$ where $v \neq w$, then the rule $\mathcal{HT}(cut)$ introduces the constraint $\mathbf{A} \neq^? \mathbf{B}$, which has to be processed before $\mathcal{B}$ can be closed in order to conserve soundness. Tableau instantiation is only permitted for $\mathcal{R}$-solved pairs, where the most general unifier is obvious ($\mathcal{HT}(subst)$ in figure 2). We will call a pair $X \neq^? \mathbf{C}$ *$\mathcal{R}$-solved*, iff $X \notin \mathbf{Free}(\mathbf{C})$ and $\mathbf{C}$ is $\mathcal{R}$-legal for X. For a given tableau $\mathcal{R}.\mathcal{T}$, we will call the combined substitution $\theta_\mathcal{T}$ generated by all applications of $\mathcal{HT}(subst)$ in $\mathcal{T}$ the substitution *induced* by $\mathcal{R}.\mathcal{T}$. Note that σ is a $\mathcal{R}$-substitution.

The $\mathcal{HT}(leib)$ rule allows to interpret unification constraints as Leibniz equalities. This rule has been introduced by Christoph Benzmüller in [4] and can be seen as a restricted form of primitive substitution introducing the necessary structure for Leibniz equality. Note that the inverse of this rule is admissible in

$\mathcal{HT}$, as we can see in figure 7. Since it is not clear, whether all instantiations that are necessary for a proof can be found by the unification rules in figure 3, $\mathcal{HT}$ utilizes a *primitive substitution* rule that allows to instantiate flexible literals with general bindings for arbitrary connectives and quantifiers.

Unification constraints can be structurally simplified by decomposition of applications (rule $\mathcal{SIM}(dec)$) and functional extensionality (rule $\mathcal{SIM}(fun)$). The latter rule is licensed by functional extensionality ($\mathbf{EXT}^{\alpha\to\beta}$) and supplies new arguments that witness the difference of functions $\mathbf{A}_{\alpha\to\beta}$ and $\mathbf{B}_{\alpha\to\beta}$. Note that this rule eliminates λ-binders from unification problems, so that it is sufficient to restrict the rule $\mathcal{SIM}(dec)$ to applications. All other substitution pairs have

$$\frac{\mathbf{A}^v \quad \mathbf{B}^w}{\mathbf{A} \neq^? \mathbf{B}}\,\mathcal{HT}(cut) \qquad \frac{X \neq^? \mathbf{C} \quad \mathcal{R}-\text{solved}}{\mathbf{C} \neq^? \mathbf{C}}\,\mathcal{HT}(subst)^1$$

$$\frac{\mathbf{A} \neq^? \mathbf{B}}{\mathbf{A} = \mathbf{B}^{\mathsf{F}}}\,\mathcal{HT}(leib) \qquad \frac{X_\alpha \overline{\mathbf{U}^n}^v \quad \mathbf{K} \in \mathcal{A}^k(\Sigma) \quad k \in \Sigma^{\mathcal{HOL}}}{X\overline{\mathbf{U}^n}^v \;\Big|\; X \neq^? \mathbf{K}}\,\mathcal{HT}(prim)$$

[1] the resulting tableau is of the form $\mathcal{R}(X \mapsto \mathbf{C}).\{X \mapsto \mathbf{C}\}\mathcal{T}$

Figure 2: Constraint and instantiation rules in $\mathcal{HT}$

to be treated by the unification rule from . Note that the rules from figure 3 together with $\mathcal{HT}(subst)$ directly correspond to the rules of higher-order preunification [14]. With these rules we use the tableau mechanism to construct unification trees (see Theorem 3.10 or Chapter I.2.12, Section 9). All of these rules are used with the understanding that all formulae are reduced $\beta\eta$-normal form after each rule application.

$$\frac{F_\alpha \overline{\mathbf{U}} \neq^? h\overline{\mathbf{V}} \quad \mathbf{G} \in \mathcal{A}^h_\alpha(\Sigma)}{F \neq^? \mathbf{G}_\alpha \;\Big|\; \lambda\overline{X}.F\overline{\mathbf{U}} \neq^? \lambda\overline{X}.h\overline{\mathbf{V}}}\,\mathcal{HT}(flex-rigid)$$

$$\frac{h\overline{\mathbf{U}^n} \neq^? h\overline{\mathbf{V}^n} \quad h \in \Sigma}{\mathbf{U}^1 \neq^? \mathbf{V}^1 \;\Big|\; \ldots \;\Big|\; \mathbf{U}^n \neq^? \mathbf{V}^n}\,\mathcal{SIM}(dec) \qquad \frac{\mathbf{A}_{\alpha\to\beta} \neq^? \mathbf{B}_{\alpha\to\beta}}{\mathbf{A}w \neq^? \mathbf{B}w}\,\mathcal{SIM}(fun)^1$$

[1] $w_\alpha \in \mathcal{W}$ new and $\mathcal{R}^1 = \mathcal{R} * \{w \mapsto \mathbf{Free}(\mathbf{A}) \cup \mathbf{Free}(\mathbf{B})\}$

Figure 3: Unification rules of $\mathcal{HT}$

We call a branch $\mathcal{B}$ in a higher-order tableau $\mathcal{T}$ *closed*, iff $\mathcal{B}$ ends in a flex-flex pair or a trivial pair $\mathbf{A} \neq^? \mathbf{A}$. Note that closed branches need not stay closed,

since the heads of flex-flex pairs[3] can be instantiated making them rigid. In this case unification has to be resumed on the particular branch. The $\mathcal{HT}(subst)$ rule immediately closes the branch $\mathcal{B}$ that ends in a $\mathcal{R}$-solved pair.

A tableau $\mathcal{R}.\mathcal{T}$ is called *closed*, iff each branch of $\mathcal{R}.\mathcal{T}$ is closed. For a proposition $\mathbf{A} \in wff_o(\Sigma)$ we write $\vdash_{\mathcal{HT}} \mathbf{A}$, if there is a closed higher-order tableau $\mathcal{R}.\mathcal{T}$ with $\mathbf{A}^{\mathsf{F}}$ at the root. In this case we call $\mathcal{R}.\mathcal{T}$ a $\mathcal{HT}$ *refutation* for $\mathbf{A}^{\mathsf{F}}$ and a $\mathcal{HT}$*-proof* for $\mathbf{A}$.

Example 3.5 (A $\mathcal{HT}$-Proof of Cantor's Theorem) We show that there cannot be a surjective mapping from the natural numbers to infinite sequences of natural numbers (see example 2.2). In order to simplify the problem we take the type ι to be the set of natural numbers ($\mathbb{N} = \lambda X_\alpha.\mathbf{T}_o$), thus Cantor's theorem has the form

$$\neg\exists F_{\iota\to\iota\to\iota}.\forall G_{\iota\to\iota}.\exists J_\iota.FJ = G$$

To be able to prove the theorem we need a property of the natural numbers (instead of the Peano Axioms we add the fact that $\forall X_\iota.\neg X = sX$ that the successor function has no fixed point). To keep the presentation short, we will add an extensionality axiom, which is not strictly needed (the HOT system discussed in Section 5 finds a proof without it).

1	$(\neg\exists F_{\iota\to\iota\to\iota}.\forall G_{\iota\to\iota}.(\exists J_\iota.FJ = G))^{\mathsf{F}}$	initial
2	$(\forall H_{\iota\to\iota}.\forall K_{\iota\to\iota}.H = K \Rightarrow .\forall N_\iota.HN = KN)^{\mathsf{T}}$	initial
3	$(\forall X_\iota.\neg X = sX)^{\mathsf{T}}$	initial
4	$\forall G_{\iota\to\iota}.(\exists J_\iota.fJ = G)^{\mathsf{T}}$	$[\mathcal{HT}(ex)1]$
5	$(\exists J_\iota.fJ = G)^{\mathsf{T}}$	$[\mathcal{HT}(all)4]$
6	$(fj = G)^{\mathsf{T}}$	$[\mathcal{HT}(ex)5]$
7	$(H = K \Rightarrow .\forall N_\iota.HN = KN)^{\mathsf{T}}$	$[\mathcal{HT}(all)2]$

At this stage $\mathcal{R} = \{f \mapsto \emptyset, j \mapsto \{G\}$. Figure 4 completes the tableau proof of cantor's theorem. Note that if we collect the bindings for G, then we obtain the instance $\lambda Z_\iota.s(fZZ)$, which corresponds to the incremented diagonal sequence that serves as a counterexample to countability in Cantor's original proof.

3.1 Soundness

We now proceed to give a definition of validity for labeled formulae that is the basis of the soundness considerations. This notion of validity takes positive variables implicitly, universally quantified, and uses the notion of $\mathcal{R}$-correspondences as a semantic counterpart of variable conditions that specify the dependencies of variables recorded during the clause normal form transformation.

Definition 3.6 ($\mathcal{R}$-Correspondence) Let $\mathcal{M} = (\mathcal{D}, \mathcal{I}, \upsilon)$ be a Σ-model and $\mathcal{R}$ a variable condition. If $\mathcal{R}(w_\beta) = \{X_{\alpha_1}, \ldots, X_{\alpha_n}\}$, then a total function

$$\mathsf{f}_w : \mathcal{D}_{\alpha_1} \times \cdots \times \mathcal{D}_{\alpha_n} \to \mathcal{D}_\beta$$

[3]Recall that a formula is called *flexible*, if the head is a free (positive) variable and *rigid* otherwise.

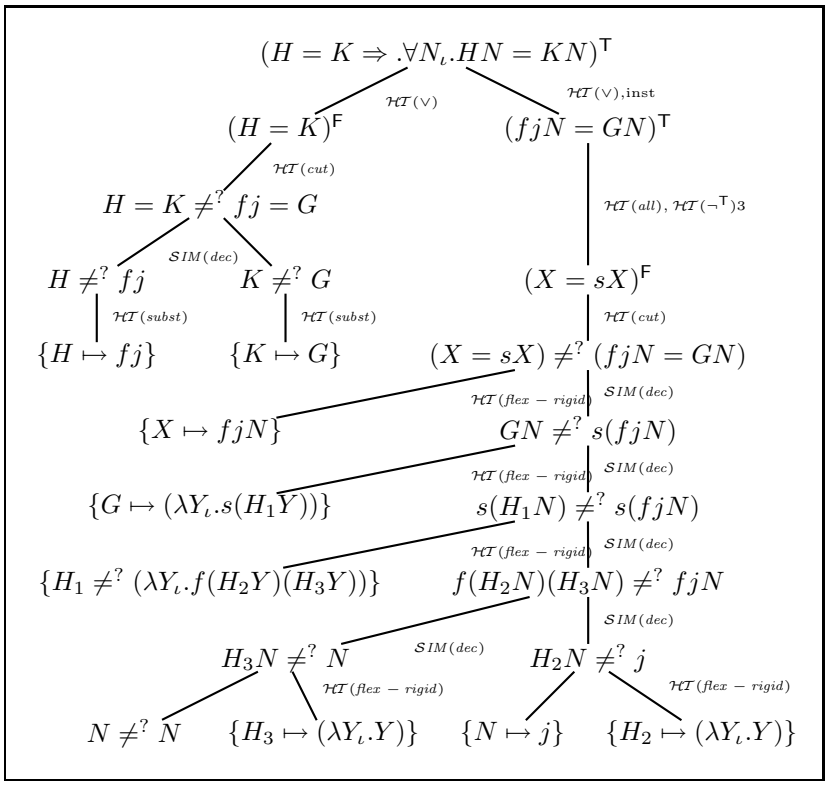

Figure 4: Unification for Cantor's Theorem

is called an $\mathcal{R}$-*function* for w_β in $\mathcal{M}$. We call a complete set $\mathcal{F} := \{\mathsf{f}_w \big| w_\beta \in \mathbf{Dom}(\mathcal{R})\}$ of $\mathcal{R}$-functions an $\mathcal{R}$-*correspondence* for $\mathcal{M}$. Note that we need a value $\mathsf{f}_w \in \mathcal{D}_\beta$ in $\mathcal{F}$, even if $n = 0$.

If $\mathcal{F}$ is an $\mathcal{R}$-correspondence for $\mathcal{M}$ and φ is an assignment into $\mathcal{M}$, then we define the alternative value function $\mathcal{I}^{\mathcal{F}}_\varphi$ by the new base case in the definition of homomorphic extension (cf. 2.3) for witness constants $w_\beta \in \mathcal{W}$

$$\mathcal{I}^{\mathcal{F}}_\varphi(w_\beta) := \begin{cases} \mathcal{I}(w_\beta), & \text{if} \quad w_\beta \notin \mathbf{Dom}(\mathcal{R}) \\ \mathsf{f}_w(\overline{\varphi(X_n)}), & \text{if} \quad \mathcal{R}(w_\beta) = \{X^1, \ldots, X^n\} \end{cases}$$

Note that $\mathcal{I}^{\mathcal{F}}_\varphi(\mathbf{A}) = \mathcal{I}_\varphi(\mathbf{A})$, if $\mathbf{A}$ does not contain witness constants in $\mathbf{Dom}(\mathcal{R})$.

Let $\mathbf{A}$ be a formula with $\mathbf{Free}(\mathbf{A}) = \{Y^1, \ldots, Y^m\}$ and $\mathcal{W}(\mathbf{A}) = \{w^1, \ldots, w^k\}$, and $\mathcal{R}(w_i) = \{Z^{i1}, \ldots, Z^{in_i}\}$, then we define the $\mathcal{R}(X \mapsto \mathbf{A})$-correspondence

$$\begin{array}{l} \mathcal{F}(X \mapsto \mathbf{A}) := \{f'_w \big| f_w \in \mathcal{F}\} \\ f'_w(a^1, \ldots, a^n, b^1, \ldots, b^m, c^{11}, \ldots, c^{kn_k}) := f_w(a^1, \ldots, a^n, \mathcal{I}^{\mathcal{F}}_\varphi(\mathbf{A})) \\ \varphi := [b^1/Y^1], \ldots, [b^m/Y^m], [c^{11}/Z^{11}], \ldots, [c^{kn_k}/Z^{kn_k}] \end{array}$$

Clearly, $\mathcal{F}(X \mapsto \mathbf{A})$ is an $\mathcal{R}(X \mapsto \mathbf{A})$-correspondence by construction, since $\mathcal{R}(X \mapsto \mathbf{A})(w) = \mathcal{R}(w) \cup \{Y^1, \ldots, Y^m, Z^{11}, \ldots, Z^{kn_k}\}$

For this variant notion of evaluation, we have a substitution-value theorem that is similar to that for the original value function $\mathcal{I}_\varphi$.

Theorem 3.7 (Substitution-Value Theorem for $\mathcal{I}^{\mathcal{F}}_\varphi$) *Let $\mathcal{M} = (\mathcal{D}, \mathcal{I})$ be a Σ-algebra, $\mathcal{R}$ a variable condition for $\mathcal{M}$ and $\mathbf{A}$ is $\mathcal{R}$-legal for X, then*

$$\mathcal{I}^{\mathcal{F}(X\mapsto\mathbf{A})}_\varphi(\{X \mapsto \mathbf{A}\}\mathbf{B}) = \mathcal{I}^{\mathcal{F}}_{\varphi,\{X\mapsto\mathcal{I}^{\mathcal{F}}_\varphi(\mathbf{A})\}}(\mathbf{B})$$

Proof: We prove the assertion by an induction over the structure of $\mathbf{B}$. Since the inductive cases are standard, we will only present the base cases of the induction. If $\mathbf{B}$ is a proper constant c, then

$$\mathcal{I}^{\mathcal{F}(X\mapsto\mathbf{A})}_\varphi(\{X \mapsto \mathbf{A}\}\mathbf{B}) = \mathcal{I}^{\mathcal{F}(X\mapsto\mathbf{A})}_\varphi(c) = \mathcal{I}(c) = \mathcal{I}^{\mathcal{F}}_{\varphi,\{X\mapsto\mathcal{I}^{\mathcal{F}}_\varphi(\mathbf{A})\}}(c)$$

and analogously if $\mathbf{B} \neq X$ is a variable. If $\mathbf{B} = X$, then

$$\begin{aligned}
\mathcal{I}^{\mathcal{F}(X\mapsto\mathbf{A})}_\varphi(\{X \mapsto \mathbf{A}\}\mathbf{B}) &= \mathcal{I}^{\mathcal{F}(X\mapsto\mathbf{A})}_\varphi(\mathbf{A}) \\
&= \varphi, \{X \mapsto \mathcal{I}^{\mathcal{F}(X\mapsto\mathbf{A})}_\varphi(\mathbf{A})\}X \\
&= \mathcal{I}^{\mathcal{F}}_{\varphi,\{X\mapsto\mathcal{I}^{\mathcal{F}(X\mapsto\mathbf{A})}_\varphi(\mathbf{A})\}}(X) \\
&= \mathcal{I}^{\mathcal{F}}_{\varphi,\{X\mapsto\mathcal{I}^{\mathcal{F}}_\varphi(\mathbf{A})\}}(X)
\end{aligned}$$

The last step is correct, since $\{X \mapsto \mathbf{A}\}$ is a $\mathcal{R}$-substitution: We have $X \notin \mathcal{R}(w)$ for any $w \in \mathcal{W}(\mathbf{A})$, therefore $f_w \in \mathcal{F}$ and $f'_w \in \mathcal{F}(X \mapsto \mathbf{A})$ coincide for $w \in \mathcal{W}(\mathbf{A})$. Thus we have $\mathcal{I}^{\mathcal{F}(X\mapsto\mathbf{A})}_\varphi(\mathbf{A}) = \mathcal{I}^{\mathcal{F}}_\varphi(\mathbf{A})$, which concludes the proof for the $\mathbf{B} = X$ case.

Now let $\mathbf{B} = w \in \mathcal{W}$ and $\mathcal{R}(w) = \{X^1, \ldots, X^n\}$, then we have

$$\mathcal{R}(X \mapsto \mathbf{A})(w) = \{X^1, \ldots, X^n, Y^1, \ldots, Y^m, Z^{11}, \ldots, Z^{kn_k}$$

and (by construction of $\mathcal{F}(X \mapsto \mathbf{A})$)

$$\begin{aligned}
&\mathcal{I}^{\mathcal{F}(X\mapsto\mathbf{A})}_\varphi(\{X \mapsto \mathbf{A}\}\mathbf{B}) \\
&\quad = \mathcal{I}^{\mathcal{F}(X\mapsto\mathbf{A})}_\varphi(w) \\
&\quad = f'_w(\varphi(X^1), \ldots, \varphi(X^n), \varphi(Y^1), \ldots, \varphi(Y^m), \varphi(Z^{11}), \ldots, \varphi(Z^{kn_k})) \\
&\quad = f_w(\varphi(X^1), \ldots, \varphi(X^n), \mathcal{I}^{\mathcal{F}}_\varphi(\mathbf{A})) \\
&\quad = \mathcal{I}^{\mathcal{F}}_{\varphi,\{X\mapsto\mathcal{I}^{\mathcal{F}}_\varphi(\mathbf{A})\}}(w) \\
&\quad = \mathcal{I}^{\mathcal{F}}_{\varphi,\{X\mapsto\mathcal{I}^{\mathcal{F}}_\varphi(\mathbf{A})\}}(\mathbf{B})
\end{aligned}$$

□

Definition 3.8 (Tableau Satisfiability) We say that a labeled formula $\mathbf{A}^v$ is $\mathcal{R}$-*satisfiable* in a Σ-model $\mathcal{M} = (\mathcal{D}, \mathcal{I}, \upsilon)$, iff there is an $\mathcal{R}$-correspondence $\mathcal{F}$ for $\mathcal{M}$, such that $\upsilon(\mathcal{I}^{\mathcal{F}}_{\varphi}(\mathbf{A})) = v$ for all assignments φ; analogously for a pair $\mathbf{A} \neq^{?} \mathbf{B}$, iff $\mathcal{I}^{\mathcal{F}}_{\varphi}(\mathbf{A}) \neq \mathcal{I}^{\mathcal{F}}_{\varphi}(\mathbf{B})$. We call a tableau $\mathcal{R}.\mathcal{T}$ satisfiable, iff all of its formulae $\mathcal{R}$-satisfiable.

Theorem 3.9 (Soundness of $\mathcal{HT}$) *$\mathcal{HT}$-theorems are valid in the class of Σ-models.*

Proof: Let $\mathcal{R}'.\mathcal{T}'$ be a tableau obtained from $\mathcal{R}.\mathcal{T}$ by a tableau construction rule and $\mathcal{M}$ be a Σ-model, then we show that $\mathcal{M} \models \mathcal{R}.\mathcal{T}$, iff $\mathcal{M} \models \mathcal{R}'.\mathcal{T}'$. We only present the proof for the rules $\mathcal{HT}(ex)$ and $\mathcal{HT}(subst)$, since $\mathcal{SIM}(fun)$ is analogous to $\mathcal{HT}(ex)$ and all others are unproblematic, because the variable condition is not altered by the transformation.

In the the case of $\mathcal{HT}(ex)$ we have $\mathcal{R}' = \mathcal{R} * \{w \mapsto \mathbf{Free}(\mathbf{A})\}$. If $\mathcal{M} = (\mathcal{D}, \mathcal{I}, \upsilon) \models \mathcal{R}.\mathcal{T}$, then there is an $\mathcal{R}$-correspondence $\mathcal{F}$ for $\mathcal{M}$ such that for all assignments φ there is a labeled proposition or unification constraint in $\mathcal{R}.\mathcal{T}$ that is satisfied by φ in $\mathcal{M}$. Let $\Pi^{\alpha}\mathbf{A}^{\mathsf{F}}$ be the formula in $\mathcal{R}.\mathcal{T}$ that the rule $\mathcal{HT}(ex)$ acts on. Since $\mathcal{M} \models (\Pi^{\alpha}\mathbf{A})^{\mathsf{F}}$ we have $\upsilon(\mathcal{I}^{\mathcal{F}}_{\varphi}(\Pi^{\alpha}\mathbf{A})) = \mathsf{F}$, thus there is an $\mathsf{a} \in \mathcal{D}_{\alpha}$ such that $\mathcal{I}^{\mathcal{F}}_{\varphi}(\mathbf{A})(\mathsf{a}) = \mathcal{I}_{\psi}(\mathbf{A}Y) = \mathsf{F}$, where Y is a new variable and $\psi := \varphi, \{Y \mapsto \mathsf{a}\}$. Since $\mathcal{I}^{\mathcal{F}}_{\psi}(\mathbf{A}) = \mathcal{I}^{\mathcal{F}}_{\psi'}(\mathbf{A})$ for any assignment ψ' that agrees with ψ on $\mathbf{Free}(\mathbf{A}) = \{X_1, \ldots, X_n\}$, this a only depends on $\psi|_{\mathbf{Free}(\mathbf{A})} = \varphi|_{\mathbf{Free}(\mathbf{A})}$. Since we have made no assumptions on φ, we can construct an n-ary function F_w that maps each tuple $(\psi(X_1), \ldots, \psi(X_n))$, such that $\upsilon(\mathcal{I}^{\mathcal{F}}_{\varphi}(\mathbf{A})(\mathsf{a})) = \mathsf{F}$. These sets are always nonempty, so by the axiom of choice (not the one on the object level, which we have not included into $\mathcal{HOL}$ but the one on the meta level we are reasoning on) there is a total function f_w mapping each tuple $(\psi(X_1), \ldots, \psi(X_n))$ to some element of $\mathsf{F}_w(\psi(X_1), \ldots, \psi(X_n))$. Thus $\mathcal{F}' := \mathcal{F} * \mathsf{f}_w$ is an $\mathcal{R}'$-correspondence. Furthermore, we have

$$\mathcal{I}^{\mathcal{F}'}_{\varphi}(\mathbf{A}w) = \mathcal{I}^{\mathcal{F}}_{\varphi}(\mathbf{A})(\mathsf{f}_w(\varphi(X_1), \ldots, \varphi(X_n))) = \mathsf{F}$$

for all assignments φ into $\mathcal{M}$ and thus $\mathcal{M} \models \mathcal{D}$ by definition.

For the converse direction let $\mathcal{M} \models \mathcal{D}$. We assume the existence of an $\mathcal{R}'$-correspondence $\mathcal{F}'$ for $\mathcal{M}$ such that $\upsilon(\mathcal{I}^{\mathcal{F}'}_{\varphi}(\mathbf{A}w)) = \mathsf{F}$ for all assignments φ. Since $w \in \mathbf{Dom}(\mathcal{R}')$ there must be a function $\mathsf{f}_w : \mathcal{D}_{\alpha_1} \times \cdots \times \mathcal{D}_{\alpha_n} \to \mathcal{D}_{\beta}$ in $\mathcal{F}'$. Let $\mathcal{F} := \mathcal{F}' \setminus \{\mathsf{f}_w\}$, then $\mathcal{F}$ is an $\mathcal{R}$-correspondence and $\mathcal{I}^{\mathcal{F}}_{\varphi}(\mathbf{A}w) = \mathcal{I}^{\mathcal{F}}_{\varphi}(\mathbf{A})(\mathsf{f}_w(\varphi(X_1), \ldots, \varphi(X_n)))$, and therefore $\upsilon(\mathcal{I}^{\mathcal{F}}_{\varphi}(\Pi^{\alpha}\mathbf{A})) = \mathsf{F}$, since υ is a valuation. Since we have taken φ to be an arbitrary valuation, we have $\mathcal{M} \models \mathcal{R}.\mathcal{T}$.

For the $\mathcal{HT}(subst)$ case we have $\mathcal{I}^{\mathcal{F}}_{\varphi}(\mathbf{A}) = \mathcal{I}^{\mathcal{F}(X \mapsto \mathbf{A})}_{\varphi}(\{X \mapsto \mathbf{B}\}\mathbf{A})$ by the substitution value theorem 3.7. Note that in particular, the truth values of the formulae labelling the tableau are not changed by the instantiation with $\mathcal{HT}(subst)$, so tableau satisfiability is conserved, which entails the assertion. □

3.2 Higher-Order Unification with $\mathcal{HT}$

We will now use higher-order unification theory to establish a partial completeness result for $\mathcal{HT}$: The $\mathcal{HT}$ rules for the unification constraints constitute a complete higher-order pre-unification algorithm, or in other words, $\mathcal{HT}$ is complete on the fragment of formulae of the form $\mathcal{U}:= \overline{\mathcal{Q}}.\mathbf{A}^1 = \mathbf{B}^1 \wedge \ldots \wedge \mathbf{A}^n = \mathbf{B}^n$, where $\overline{\mathcal{Q}}$ is an arbitrary quantifier prefix.

The quantifier rules transform $\mathcal{U}^{\mathsf{F}}$ into a tableau, $\mathcal{R}.\mathcal{T}$ that has the unification constraints[4] $\mathbf{A}^i \neq^? \mathbf{B}^i$ at the leaves, and where the variable condition $\mathcal{R}$ encodes the information from the quantifier prefix $\overline{\mathcal{Q}}$. Clearly any unifier of this set of constraints corresponds to a tableau proof of $\mathcal{U}$.

For a given tableau $\mathcal{R}.\mathcal{T}$, let us call the pair $\mathcal{E}:= \mathcal{R}.\mathcal{E}_{\mathcal{T}} = \mathcal{R}.\{\mathbf{A}^1 \neq^? \mathbf{B}^1, \ldots, \mathbf{A}^n \neq^? \mathbf{B}^n\}$ – where the $\mathbf{A}^i \neq^? \mathbf{B}^i$ are the unification constraints at the leaves of $\mathcal{T}$ – the *unification problem* associated with $\mathcal{R}.\mathcal{T}$. We will say that θ is a *higher-order unifier* for $\mathcal{R}.\mathcal{T}$, iff it is for $\mathcal{R}.\mathcal{E}_{\mathcal{T}}$, i.e. a $\mathcal{R}$-substitution, such that $\theta(\mathbf{A}^i) =_{\beta\eta} \theta(\mathbf{B}^i)$. So the completeness result says that $\mathcal{HT}$ can find some higher-order unifier of a tableau $\mathcal{R}.\mathcal{T}$, if there is one.

In unification theory, one is generally interested in a even stronger completeness result: For any unifier θ of $\mathcal{E}$, find a unifier (called *most general unifier*), from which other unifiers can be generated by further instantiation. However, for higher-order logic, such most general unifiers need not exist in general. Even though complete sets of unifiers can be calculated, they are in general very redundant, and the search spaces involved are enormous [14]. Especially, in the case of unification constraints, where the heads of both sides are variables (so-called *flex-flex pairs*). Fortunately, for theorem proving purposes (especially in a refutation calculus) it is more important to know about the existence of a unifier, than to have it itself. Therefore, it is sufficient to for our application consider flex-flex pairs as solved, since they are guaranteed to have unifiers. A flex-flex pair $F\overline{\mathbf{U}^n} =^? \mathbf{G}\overline{\mathbf{V}^m}$, can be solved by the substitution $\sigma = \{F \mapsto (\lambda\overline{Z^n}.W), G \mapsto (\lambda\overline{Z^m}.W)\}$ which binds the head variables F and G to constant functions that absorb their arguments. The corresponding unification problem is called the *pre-unification* problem (i.e. find a substitution σ, such that $\sigma(\mathbf{A}^i) =_{\beta\eta} \sigma(\mathbf{B}^i)$ or $\sigma(\mathbf{A}^i) =^? \sigma(\mathbf{B}^i)$ is a flex-flex pair). We will denote the set of pre-unifiers for a unification problem $\mathcal{E}$ with $\mathbf{PU}(\mathcal{E})$. Now the completeness result for pre-unification

Theorem 3.10 ($\mathcal{HT}$ is a complete for pre-unification) *If θ is a pre-unifier of a tableau $\mathcal{R}.\mathcal{T}$, then there is an extension $\mathcal{R}'.\mathcal{T}'$ of $\mathcal{R}.\mathcal{T}$, such that its induced substitution σ is also a pre-unifier of $\mathcal{R}.\mathcal{T}$, and furthermore, σ is more general than θ.*

Proof sketch: The proof of the assertion is carried out in two parts: We first convince ourselves that the rules $\mathcal{SIM}(dec)$, $\mathcal{SIM}(fun)$, and $\mathcal{HT}(subst)$ are terminating and confluent, and furthermore conserve sets of pre-unifiers.

[4] Recall the inverse of $\mathcal{HT}(leib)$, which is admissible in $\mathcal{HT}$.

This only leaves the rule $\mathcal{HT}(\mathit{flex}-\mathit{rigid})$, which need not be terminating in general[5]. So the completeness result hinges on a semi-termination lemma, which states that given a unification problem $\mathcal{E}$ that is not solved but $\mathcal{SIM}$-normal, there is a transformation with $\mathcal{HT}(\mathit{flex}-\mathit{rigid})$ (which approximates θ) that is measure-decreasing. So any unifier θ of $\mathcal{R}.\mathcal{T}$ determines a terminating sequence of $\mathcal{HT}$-transformations that ends in a closed tableau. Since all instantiations generated by $\mathcal{HT}(\mathit{subst})$ along the way approximate θ, the resulting unifier also approximates θ, i.e. is more general than it. □

3.3 Completeness of $\mathcal{HT}$

A central part of the completeness proofs for unification-based refutation calculi are the lifting properties. The central lifting theorem for $\mathcal{HT}$ states that any given tableau refutation for $\theta(\mathbf{A})$, can be lifted to a tableau refutation for $\mathbf{A}$. For the construction of a lifted $\mathcal{HT}$-tableau $\mathcal{R}.\mathcal{T}$ for a proposition $\mathbf{A}$ from a tableau $\mathcal{R}_\theta.\mathcal{T}_\theta$ for $\theta(\mathbf{A})$ it is crucial to maintain a tight correspondence $\omega\colon\mathcal{T}_\theta \to \mathcal{T}$ between $\mathcal{R}.\mathcal{T}_\theta$ and $\mathcal{R}_\theta.\mathcal{T}$ that respects labels and is compatible with θ, i.e. for any node $\mathcal{N}$ in $\mathcal{T}$ with labeled formula $\mathbf{A}^v$ we have $\omega_{\mathcal{N}}(\theta(\mathbf{A})) = \mathbf{A}$. The main difficulty with lifting properties in higher-order logic is the fact that due to the existence of predicate variables at the head of formulae, the propositional structure of formulae can change during instantiation. For instance if $\theta(F_{\alpha\to o}) = \lambda X_\alpha.GX \vee p_o$, and $\mathbf{A}^{\mathsf{T}} = Fa^{\mathsf{T}}$ is the formula of $\mathcal{N}$, then $\mathcal{HT}(\vee)$ is applicable in $\mathcal{T}_\theta$ but not in the fragment of $\mathcal{T}$ already constructed. The solution of this problem is to apply $\mathcal{HT}(\mathit{prim})$ with a suitable imitation binding $\mathcal{G}^{\vee}_{\alpha\to o} = \lambda X_\alpha.(H^1X) \vee (H^2X)$ and to obtain a node $\mathcal{N}'$ with formula $(H^1a \vee H^2a)^{\mathsf{T}}$, to which $\mathcal{HT}(\vee)$ can be applied. Since $\mathcal{G}^{\vee}_{\alpha\to o}$ is more general than $\theta(F)$ there is a substitution ρ, such that $\theta(F) = \rho(\mathcal{G}^{\vee}_{\alpha\to o})$, therefore $\omega_{\mathcal{N}'}(\theta'(H^1a \vee H^2a)^{\mathsf{T}}) = (H^1a \vee H^2a)^{\mathsf{T}}$ where $\theta' = \theta \cup \rho$.

Definition 3.11 (Tableau Embedding) Let $\mathcal{R}.\mathcal{T}$ and $\mathcal{R}'.\mathcal{T}'$ be higher-order tableaux such that Φ and Φ' are the respective sets of nodes in $\mathcal{R}.\mathcal{T}$ and $\mathcal{R}'.\mathcal{T}'$, then we call an injection $\omega\colon\Phi \to \Phi'$ a *tableau embedding*, iff

- it preserves dominance in trees (i.e. if node $\mathcal{M}$ dominates node $\mathcal{N}$ in $\mathcal{T}$, then $\omega(\mathcal{M})$ dominates $\omega(\mathcal{N})$ in $\mathcal{T}'$) and furthermore
- it preserves truth values, i.e. $\omega(\mathbf{M}^\alpha)$ is of the form $\mathbf{N}^\alpha$. Here (and in the following) we also use ω for the mapping on formulae induced by ω ($\mathbf{M}$ and $\mathbf{N}$ are the formulae at nodes $\mathcal{M}$ and $\mathcal{N} = \omega(\mathcal{M})$).

For the lifting argument it will also be important to monitor the solvability of the unification constraints represented by the collection of leaves of a tableau. In particular, we will have to relate the sets of constraints of the basic and the lifted tableau modulo the connecting substitution θ. The condition we want to

[5] Pre-unifiers can be extended to unifiers, therefore pre-unification problem cannot be decidable, since otherwise the higher-order unification problem would also be.

maintain is that the set of solutions of the lifted constraints, together with those induced by θ should always be the set of solutions of the ground constraints.

Definition 3.12 (Equivalent mod θ) Let Φ and Φ' be sets of substitutions and $\tau(\rho) := \rho \circ \theta$ for $\rho \in \Phi$, then we say that the sets Φ and Φ' are *equivalent mod* θ, iff τ is a bijection between Φ' and Φ. In this case we write $\Phi \Leftrightarrow_\theta \Phi'$.

Clearly τ is injective by construction, so it is only necessary to check that $\tau(\Phi') = \Phi$ in order to verify that Φ and Φ' are equivalent mod θ.

Lemma 3.13 *For a substitution σ let $\mathcal{E}_\sigma := \{X \stackrel{?}{\neq} \sigma(X) \big| X \in \mathbf{Dom}(\sigma)\}$, then we have $\mathbf{PU}(\mathcal{R}.\mathcal{E} \vee \mathcal{E}_\theta) \Leftrightarrow_\theta \mathbf{PU}(\mathcal{R}(\theta).\theta(\mathcal{E}))$ for any $\mathcal{R}$-substitution θ.*

Proof: We have $\tau(\mathbf{PU}(\mathcal{R}(\theta).\theta(\mathcal{E})) \subseteq \mathbf{PU}(\mathcal{R}.\mathcal{E} \vee \mathcal{E}_\theta)$, since for any substitution $\sigma \in \mathbf{PU}(\mathcal{R}(\theta).\theta(\mathcal{E}))$ we have $\tau(\sigma) = \sigma \circ \theta$ is a pre-unifier of $\mathcal{E}_\theta$. On the other hand, any pre-unifier ρ of $\mathcal{R}.\mathcal{E} \vee \mathcal{E}_\theta$ is also one of $\mathcal{R}.\mathcal{E}_\theta$, so we have $\rho = \rho \circ \theta$, since θ is the most general unifier of $\mathcal{R}.\mathcal{E}_\theta$. Thus we have $\mathbf{PU}(\mathcal{R}.\mathcal{E} \vee \mathcal{E}_\theta) \subseteq \tau(\mathbf{PU}(\mathcal{R}(\theta).\theta(\mathcal{E}))$, which completes the proof. □

Definition 3.14 (θ-Compatible) Let $\mathcal{R}.\mathcal{T}$ and $\mathcal{R}'.\mathcal{T}'$ be higher-order tableaux with unification constraints $\mathcal{R}.\mathcal{E}$ and $\mathcal{R}'.\mathcal{E}'$, and let θ be an $\mathcal{R}$-substitution. We say that a tableau embedding $\omega: \mathcal{T} \to \mathcal{T}'$ is θ-*compatible*, iff $\omega(\theta(\mathbf{M})) = \mathbf{M}$ and moreover $\mathbf{PU}(\mathcal{R}'.\mathcal{E}' \vee \mathcal{E}_\theta) \Leftrightarrow_\theta \mathbf{PU}(\mathcal{R}.\mathcal{E})$

Lemma 3.15 *For any $\mathbf{A} \in wff_o(\Sigma)$ and any substitution θ, such that the head of $\theta(\mathbf{A})$ is $\vee$, there is a tableau $\mathcal{T}_\vee$, such that*

- *$\mathcal{T}_\vee$ has substitution $\theta_\vee$, root $\mathbf{A}^\mathsf{T}$ and the two open leaves $\mathbf{B}^\mathsf{T}$ and $\mathbf{C}^\mathsf{T}$*
- *a substitution $\rho_\vee$, such that $\theta = \rho_\vee \circ \theta_\vee[\mathbf{Dom}(\theta_\vee)]$ and $\theta \cup \rho_\vee(\mathbf{A}) = \mathbf{B}' \vee \mathbf{C}'$, where $\mathbf{B}' = \theta \cup \rho_\vee(\mathbf{B})$ and $\mathbf{C}' = \theta \cup \rho_\vee(\mathbf{C})$*

Proof: Let $\mu(\theta, \mathbf{A})$ be the multiset of term depths in $\theta(\mathbf{Free}(\mathbf{A}))$. We prove the assertion by well-founded induction over the multi-set ordering for natural numbers. If $\mathbf{A} = \mathbf{B} \vee \mathbf{C}$, then $\mathcal{T}$ consists of a single application of $\mathcal{HT}(\vee)$ and $\theta_\vee = \rho_{\mathcal{HT}(\vee)} = \emptyset$.

If $\mathbf{head}(\mathbf{A}) \neq \vee$, then $\mathbf{A}$ must be flexible, i.e. $\mathbf{A} = P_\alpha \overline{\mathbf{U}^n}$ for some variable $P_\alpha \in \mathbf{Dom}(\theta)$. The head h of $\theta(P)$ is either $\vee$ or $\theta(P)$ is a j-projection. Therefore, by the general binding theorem 3.3 there is a partial binding $\mathbf{G}^h_\alpha$, a substitution ρ' such that $\rho'(\mathbf{G}^h_\alpha) =_{\beta\eta} \theta(P_\alpha)$. Now let $\theta' := \theta \cup \rho'$, then $\theta'(\{P \mapsto \mathbf{G}^h_\alpha\}\mathbf{A}) = \theta(\mathbf{A})$ and $\theta = \rho' \circ \{P \mapsto \mathbf{G}^h_\alpha\}$. Furthermore $\mu(\theta, \{P \mapsto \mathbf{G}\}\mathbf{A}) < \mu(\theta, \mathbf{A})$, since $\mathbf{dp}(\rho) < \mathbf{dp}(\theta(P))$ and $P \notin \mathbf{Free}(\{P \mapsto \mathbf{G}^h_\alpha\}\mathbf{A})$. So by inductive hypothesis, there is a tableau $\mathcal{T}'_\vee$ with substitution $\theta'_\vee$ and root $\{P \mapsto \mathbf{G}^h_\alpha\}\mathbf{A}$ and a substitution $\rho'_\vee$, such that $\theta' \cup \rho'_\vee(\mathbf{A}) = \theta' \cup \rho'_\vee(\mathbf{B}) \vee \theta' \cup \rho'_\vee(\mathbf{C})$. If we combine $\mathcal{T}'_\vee$ with a $\mathcal{HT}(prim)$ step at the root, we have the tableau shown in figure 5 which has root $\mathbf{A}$. Now let $\theta_\vee := \theta'_\vee \cup \{P \mapsto \mathbf{G}^h_\alpha\}$ and $\rho_\vee := \rho'_\vee \cup \rho'$,

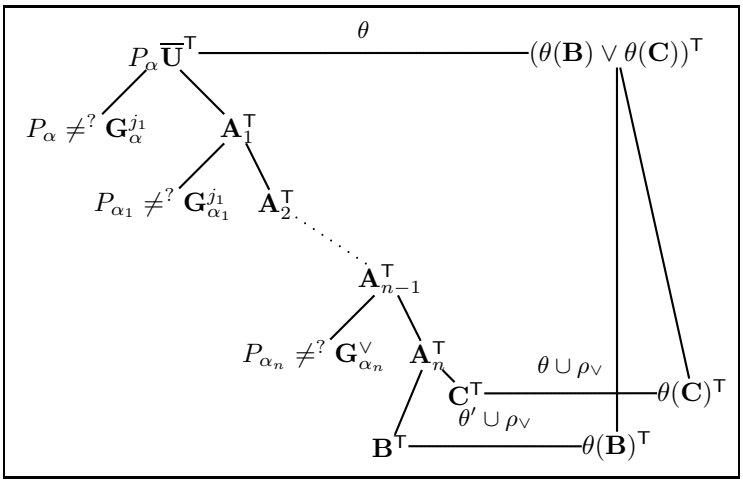

Figure 5: Tableau Lifting

then

$$\begin{aligned}\rho_\vee \circ \theta_\vee &= (\theta'_\vee \cup \{P \mapsto \mathbf{G}_\alpha^h\}) \circ (\rho'_\vee \cup \rho') \\ &= (\theta'_\vee \circ \rho'_\vee) \cup (\rho' \circ \{P \mapsto \mathbf{G}_\alpha^h\}) \\ &= \theta[\mathbf{Dom}(\theta_\vee) = \mathbf{Dom}(\theta'_\vee) * P]\end{aligned}$$

Furthermore $\theta'_\vee(\mathbf{B}) = \theta_\vee(\mathbf{B})$, since $P \notin \mathbf{Free}(\mathbf{B})$ and similarly for $\mathbf{C}$. □

Clearly, similar results for the other structural $\mathcal{HT}$ *rules* can be shown by exactly the same methods.

Theorem 3.16 (Tableau Lifting) *Let* $\mathbf{A} \in wff_o(\Sigma)$ *and* θ *a substitution, then* $\mathbf{A}$ *has a* $\mathcal{HT}$*-proof provided* $\theta(\mathbf{A})$ *has one.*

Proof: Let $\mathcal{R}_\theta.\mathcal{T}_\theta$ be a tableau refutation for $\theta(\mathbf{A})$, the claim is proven by an induction on the construction of $\mathcal{R}_\theta.\mathcal{T}_\theta$ constructing a tableau refutation $\mathcal{R}.\mathcal{T}$ for $\mathbf{A}$, a substitution ρ and a $\theta \cup \rho$-compatible tableau embedding $\omega\colon \mathcal{T}_\theta \to \mathcal{T}$. This is sufficient for the proof of the assertion, since the leaves of $\mathcal{R}.\mathcal{T}$ can only be unification constraints as $\mathcal{T}_\theta$ is closed. $\mathcal{R}.\mathcal{E}_\mathcal{T}$ must be pre-unifiable, since ω is $\theta \cup \rho$-compatible $\mathcal{R}_\theta.\mathcal{T}_\theta$ is closed:

$$\emptyset \neq \mathbf{PU}(\mathcal{R}_\theta.\mathcal{E}_{\mathcal{T}_\theta}) \Leftrightarrow_{\theta\cup\rho} \mathbf{PU}(\mathcal{R}.\mathcal{E}_\mathcal{T} \vee \mathcal{E}_{\theta\cup\rho}) \subseteq \mathbf{PU}(\mathcal{R}.\mathcal{E}_\mathcal{T})$$

So let us now construct $\mathcal{R}.\mathcal{T}$, ρ and ω inductively on the structure of $\mathcal{T}_\theta$. To make the induction go through, we need to treat the slightly stronger assertion, where we do not assume $\mathcal{R}_\theta.\mathcal{T}_\theta$ to be closed. If $\mathcal{R}_\theta.\mathcal{T}_\theta$ is an initial tableau, then we we take ω to be the obvious tableau embedding and ρ the empty substitution. ω is $\theta \cup \rho$-compatible, since there are no unification constraints present.

Now let us assume that the assertion holds for $\omega{:}\,\mathcal{R}.\mathcal{T}_\theta \to \mathcal{R}.\mathcal{T}_\theta$ and $\theta \cup \rho$ and look at possible extensions of $\mathcal{T}_\theta$. If $\mathcal{T}_\theta$ is extended by $\mathcal{HT}(\vee)$ at node $\mathcal{N}_\theta$ to $\mathcal{T}'_\theta$, then $\mathcal{N}_\theta$ must be labelled with $\theta(\mathbf{A})^{\mathsf{T}}$. As ω is $\theta \cup \rho$-compatible, the node $\omega(\mathcal{N}_\theta)$ in $\mathcal{T}$ must have the label $\mathbf{A}^{\mathsf{T}}$. Lemma 3.15 guarantees a tableau $\mathcal{T}_\vee$ with root $\theta(\mathbf{A})$, the two open leaves $\mathbf{B}$ and $\mathbf{C}$.

Let $\mathcal{R}'.\mathcal{T}'$ be $\mathcal{T}$ extended with $\mathcal{T}_\vee$ at $\omega(\mathcal{N}_\theta)$, then $\mathcal{R}' = \mathcal{R}(\theta_\vee)$ and $\mathcal{E}'_{\mathcal{T}} = \theta_\vee(\mathcal{E}_{\mathcal{T}})$, since $\theta_\vee$ is the substitution of $\mathcal{T}_\vee$ and $\mathcal{T}_\vee$ does not have open constraints. Now let ω' be the obvious extension of ω that maps the disjuncts of $\theta(\mathbf{A})$ to the leaves $\mathbf{B}$ and $\mathbf{C}$ of $\mathcal{T}_\vee$ and $\widetilde{\theta} := \theta \cup \rho \cup \rho_\vee$, then it only remains to show that ω' is $\widetilde{\theta}$-compatible. Now let $\theta = \theta' \cup \theta''$, where $\mathbf{Dom}(\theta' = \theta_\vee$ and $\mathbf{Dom}(\theta'') \cap \mathbf{Dom}(\theta_\vee) = \emptyset$. Then $\theta' = \rho_\vee \circ \theta_\vee$ and we have

$$\begin{array}{rcl}
\mathbf{PU}(\mathcal{R}'.\mathcal{E}_{\mathcal{T}'} \vee \mathcal{E}_{\widetilde{\theta}}) & = & \mathbf{PU}(\mathcal{R}(\theta_\vee).\theta_\vee(\mathcal{E}_{\mathcal{T}}) \vee \mathcal{E}_\theta \vee \mathcal{E}_\rho \vee \mathcal{E}_{\rho_\vee}) \\
& \Leftrightarrow_{\rho_\vee} & \mathbf{PU}(\mathcal{R}(\rho_\vee \circ \theta_\vee).\rho_\vee \circ \theta_\vee(\mathcal{E}_{\mathcal{T}}) \vee \mathcal{E}_\theta \vee \mathcal{E}_\rho) \\
& = & \mathbf{PU}(\mathcal{R}(\theta').\theta'(\mathcal{E}_{\mathcal{T}}) \vee \mathcal{E}_{\theta''} \vee \mathcal{E}_\rho) \\
& \Leftrightarrow_{\theta'} & \mathbf{PU}(\mathcal{R}.\mathcal{E}_{\mathcal{T}} \vee \mathcal{E}_\theta \vee \mathcal{E}_\rho) \\
& \Leftrightarrow_{\theta \cup \rho} & \mathbf{PU}(\mathcal{R}_\theta.\mathcal{E}_{\mathcal{T}_\theta})
\end{array}$$

by inductive hypothesis. It is easy to check that $\Phi \Leftrightarrow_\theta \Phi'$ and $\Phi' \Leftrightarrow_\rho \Phi''$ entail $\Phi \Leftrightarrow_{\rho\circ\theta} \Phi''$ and that $\Phi \Leftrightarrow_\theta \Phi'$ and $\Phi' \Leftrightarrow_\rho \Phi''$ entail $\Phi \Leftrightarrow_\rho \Phi''$ if $\theta \subseteq \rho$, so that we have

$$\mathbf{PU}(\mathcal{R}'.\mathcal{E}_{\mathcal{T}'} \vee \mathcal{E}_{\widetilde{\theta}}) \Leftrightarrow_{\theta\cup\rho\cup\rho_\vee} \mathbf{PU}(\mathcal{R}_\theta.\mathcal{E}_{\mathcal{T}_\theta})$$

This completes the assertion for the $\mathcal{HT}(\vee)$ case, since $\widetilde{\theta} = \theta \cup \rho \cup \rho_\vee$.

The cases for the other structural $\mathcal{HT}$ rules can be shown by exactly the same methods, using the respective analogs of Lemma 3.15.

In the $\mathcal{HT}(cut)$ case, we have have the left situation below, so we can extend $\mathcal{T}$ by the right tableau.

$$\dfrac{\begin{array}{c}\theta(\mathbf{A})^v \\ \theta(\mathbf{B})^w\end{array}}{\theta(\mathbf{A}) \neq^? \theta(\mathbf{B})}\,\mathcal{HT}(cut) \qquad \dfrac{\begin{array}{c}\mathbf{A}^v \\ \mathbf{B}^w\end{array}}{\mathbf{A} \neq^? \mathbf{B}}\,\mathcal{HT}(cut)$$

Since tableau embeddings do not extend to unification pairs, we can take $\widetilde{\omega} = \omega$ and $\widetilde{\theta} = \theta$. This extension adds a new pair $\theta(\mathbf{A}) \neq^? \theta(\mathbf{B})$ to the unification constraints of $\mathcal{T}_\theta$ and $\mathbf{A} \neq^? \mathbf{B}$ of those of $\mathcal{T}$. Now, lemma 3.13 gives

$$\mathbf{PU}(\mathcal{R}.\mathcal{E} \wedge \mathcal{E}_\theta \wedge \mathbf{A} \neq^? \mathbf{B}) \Leftrightarrow_\theta \mathbf{PU}(\mathcal{R}.\mathcal{E}' \wedge \theta(\mathbf{A}) \neq^? \theta(\mathbf{B}))$$

Similarly, the simplification rule $\mathcal{SIM}(dec)$ does not affect the tableau embedding ω, since that is restricted to labeled formulae and θ, since $\mathcal{SIM}(dec)$ it conserve sets of pre-unifiers. Finally, for $\mathcal{HT}(subst)$ and $\mathcal{HT}(prim)$ which instantiate the tableau by a substitution $\{X \mapsto \mathbf{A}\}$ where $X \notin \mathbf{Dom}(\theta)$, thus we can lift their applications directly (i.e. create the lifted tableau, by applying exactly the same substitution to $\mathcal{T}$).

Now we have examined all the cases of tableau construction and have thus completed the proof of the assertion. □

Example 3.17 (Tableau Lifting) Consider the concrete lifting situation in figure 6 with the instantiated version on the right hand side.

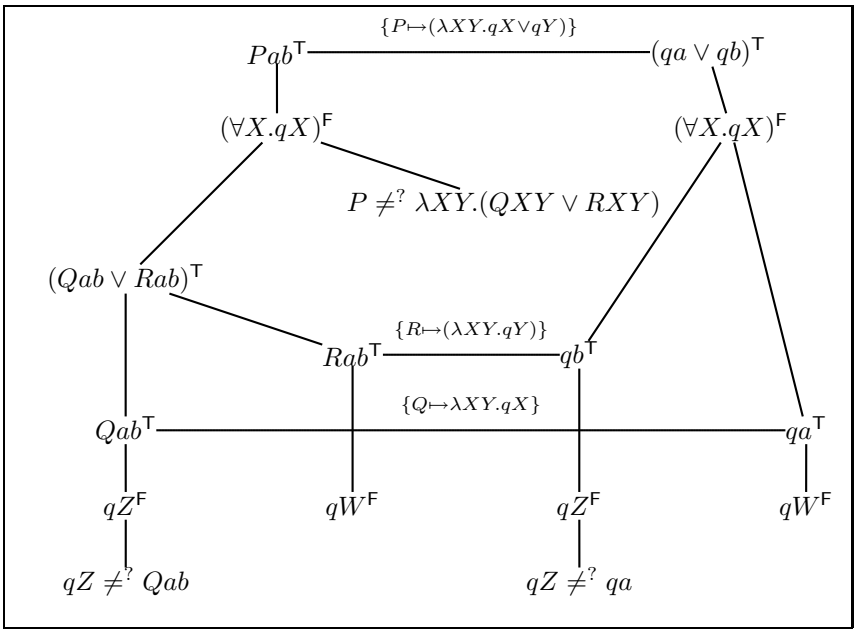

Figure 6: A concrete tableau lifting situation

Theorem 3.18 (Completeness) *$\mathcal{HT}$ is refutation complete with respect to the class of Σ-models, i.e. if Φ is a valid set of sentences, then there is a closed higher-order tableau for $\emptyset.\Phi^{\mathsf{F}}$.*

Proof: Completeness of $\mathcal{HT}$ can be proven using the model existence theorem 2.12 by verifying that the class ∇_Σ defined by

$$\nabla_\Sigma := \{\Phi \subset wff_o(\Sigma) \big| \not\vdash_{\mathcal{HT}} \Phi^{\mathsf{T}}\}$$

is a saturated abstract consistency class. For conditions 1. and 3.– 5., and 7. in the definition of abstract consistency class (cf. definition 2.11) this can be achieved with the usual techniques (see for instance [10]). Condition 2. is trivially met, since all formulae are kept in $\beta\eta$-normal form and we have treated condition 6. in the tableau lifting theorem above.

For 8. let $\Phi = \Phi' * \neg(\mathbf{A} =^{\alpha\to\beta} \mathbf{B})$ and let $\mathcal{R}.\mathcal{T}$ be a closed tableau for $\Phi * (\neg\mathbf{A}w = \mathbf{B}w)$ for some witness constant w that does not occur in Φ. The

Φ'	$(\mathbf{A} = \mathbf{B})^{\mathsf{F}}$
$\mathbf{A} =^{\alpha\to\beta} \mathbf{B}^{\mathsf{F}}$	$(\forall P_{\alpha\to o}.P\mathbf{A} \Rightarrow P\mathbf{B})^{\mathsf{F}}$
$\mathbf{A} \neq^{?} \mathbf{B}$	$r\mathbf{A}^{\mathsf{T}}$
$\mathbf{A}w_\alpha \neq^{?} \mathbf{B}w$	$r\mathbf{B}^{\mathsf{F}}$
$(\mathbf{A}w =^{\beta} \mathbf{B}w)^{\mathsf{F}}$	$r\mathbf{A} \neq^{?} r\mathbf{B}$
$\mathcal{T}'$	$\mathbf{A} \neq^{?} \mathbf{B}$
(The first step in the left tableau abbreviates the right.)	

Figure 7: Functional Extensionality for ∇_Σ.

first tableau in figure 7 is a $\mathcal{HT}$-refutation of Φ. Note that the witness constant r is eliminated again from the unification constraint by $\mathcal{SIM}(dec)$, therefore the intermediate variable condition $\mathcal{R}' := \mathcal{R} * \{r \mapsto \mathbf{Free}(\mathbf{A}) \cup \mathbf{Free}(\mathbf{B})\}$ in figure 7 can be simplified to $\mathcal{R}$.

To show saturation of ∇_Σ we assume that $\Phi * \mathbf{A} \notin \nabla_\Sigma$ and $\Phi * \neg\mathbf{A} \notin \nabla_\Sigma$ for some proposition $\mathbf{A} \in wff_o(\Sigma)$ and some $\Phi \in \nabla_\Sigma$. In other words, there are $\mathcal{HT}$-refutations for both $\Phi * \mathbf{A}$ and $\Phi * \neg\mathbf{A}$, and therefore there is one for $\Phi \cup \{\mathbf{A} \vee \neg\mathbf{A}\}$, and consequently there is a $\mathcal{HT}$-refutation for Φ, since tautologies can be eliminated in $\mathcal{HT}$. This contradicts our assumption that $\Phi \in \nabla_\Sigma$. □

3.4 Optimizations for Primitive Substitutions

The primitive substitution rules have originally been introduced by Peter Andrews in [3]. Note that the set of general bindings is infinite, since we need one for every quantifier Π^α and the set of types is infinite. Thus in contrast to the goal-directed search for instantiations in unification, the rule $\mathcal{HT}(prim)$ performs blind search and even worse, is infinitely branching. Therefore, the problem of finding instantiations for predicate variables is conceived as the limiting factor to higher-order automated theorem proving.

It has been a long-standing conjecture that in machine-oriented calculi it is sufficient to restrict the order of primitive quantifier substitutions to the order of the input formulae. In [5], we have established a finer-grained variant of theorem 2.12 that we can use as a basis to prove this conjecture. Let us now introduce the necessary definitions.

Definition 3.19 (Order) For a type $\alpha \in \mathcal{T}$, we define the *order* $\mathbf{ord}(\alpha)$ of α as $\mathbf{ord}(\iota) = \mathbf{ord}(o) = 0$, and $\mathbf{ord}(\alpha \to \beta) = \max\{\mathbf{ord}(\alpha), \mathbf{ord}(\beta)\} + 1$. Note that the set $\mathcal{T}^k = \{\alpha \in \mathcal{T} \,|\, \mathbf{ord}(\alpha) \leq k\}$ is finite for any order k. We will take the order of a formula to be the highest order of any type of any of its subformulae, and the order of a set of formulae to be the maximum of the orders of its members.

Theorem 3.20 (Model Existence with Order) *The model existence theorem (2.12) holds even if we weaken the condition 6. of an abstract consistency class (cf 2.11) to*

6. *If* $\Pi^\alpha \mathbf{A} \in \Phi$*, then* $\Phi * \mathbf{AB} \in \nabla_\Sigma$ *for each closed formula* $\mathbf{B} \in wff_\alpha(\Sigma)$ *with* $\mathbf{ord}(\mathbf{B}) \leq \mathbf{ord}(\Phi)$.

With this result [5] it is possible to restrict the order of the primitive substitutions in rule $\mathcal{HT}(prim)$ to the order of the initial formulae, since with this restriction, all $\mathcal{HT}$-rules are order-preserving. Thus, the $\mathcal{HT}$ tableau calculus is finitely branching. However, even with this restriction the search space for primitive substitutions is prohibitively large, so we have to restrict it even further.

Remark 3.21 (Atomic Cut and $\mathcal{HT}(prim)$ in $\mathcal{HT}$) From the completeness proof above we can directly verify, that the $\mathcal{HT}(cut)$ rule can be restricted to the case where the formulae $\mathbf{A}^\mathsf{T}$ and $\mathbf{B}^\mathsf{F}$ are literals. Indeed, the base cases of the definition of abstract consistency class only mention atomic formulae and the lifting argument lifts atomic cuts to atomic cuts. This restriction considerably restricts the possibilities of applications of $\mathcal{HT}(cut)$ and thus prunes the search spaces associated with proof search in $\mathcal{HT}$ considerably.

In Example 3.17, we have shown a special instance of the lifting process used in 3.16. This uses $\mathcal{HT}(prim)$ to instantiate the variable P with a disjunction binding so that the literal $P\overline{\mathbf{U}}^\mathsf{T}$ can be split into the two tableau branches present in the ground proof. Indeed this is necessary, if we want to prune search spaces by restricting $\mathcal{HT}(cut)$ to atomic cuts. However, if we do not impose this restriction, then $\exists P.(Pab) \Rightarrow \forall X.qX$ has a closed $\mathcal{HT}$-tableau that does not need $\mathcal{HT}(prim)$ (see figure 8). This suggests that without the restriction to atomic cuts, $\mathcal{HT}$ may well be complete without $\mathcal{HT}(prim)$[6]. Clearly, however, we have to extend the $\mathcal{HT}(cut)$ rule by a variant, that acts on arbitrary pairs of formulae $\mathbf{A}^v$, $\mathbf{B}^v$ with equal annotation in the branch (introducing a negation operator for soundness).

$$\frac{\mathbf{A}^v \quad \mathbf{B}^v}{\mathbf{A} \neq^? \neg\mathbf{B}} \qquad \frac{\mathbf{A}^v \quad \mathbf{B}^v}{\neg\mathbf{A} \neq^? \mathbf{B}}$$

Since the primitive substitution rule $\mathcal{HT}(prim)$ basically employs blind search for a propositional substitution, this variant of $\mathcal{HT}$ may well be more efficient, since unification generates the necessary instantiations in a goal-driven manner from the material already present in the tableau.

4 Extensional Tableaux

In the previous Section we have proven completeness of $\mathcal{HT}$ with respect to Σ-models, which characterizes traditional higher-order refutation calculi, but

[6] Note that this is a fairly strong conjecture, since the resulting calculus would be able to handle for instance generalizations in inductive proofs. Consider for instance the formula $\forall X_\iota.X+(X+X) = (X+X)+X$, which can only be proven by a detour to general associativity of $+$.

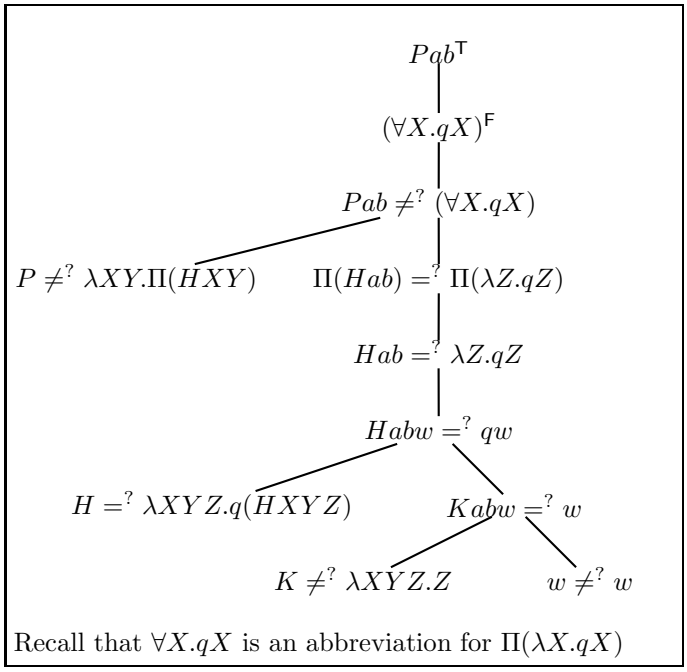

Figure 8: A closed tableau without $\mathcal{HT}(prim)$.

is not a very intuitive notion of semantics. Indeed $\mathcal{HT}$ is not complete with respect to Henkin models. Consider for instance the formula $\mathbf{C}{:=}\neg(c_{o\to o}b_o)\vee(c(\neg\neg b))$ which cannot be proven in $\mathcal{HT}$ – the middle tableau in figure 9 shows the maximal tableau for $\mathbf{C}$, which cannot be closed, since no $\mathcal{HT}$ construction rule applies to the last pair. This lack of completeness is unfortunate, since the class of Henkin models is the most intuitive one that admits complete calculi. In particular, our mathematical intuition which conforms to Henkin models would make us believe that $\mathbf{C}$ should be refutable, because $\neg\neg b$ is provably equivalent to b. This example shows us that in extensional calculi we have to deal with propositions that appear in the arguments of parameters. The simplest approach to build a calculus that can refute $\mathbf{C}$ is to add the equational theory $b=\neg\neg b$ to higher-order unification. Even though this approach is intuitive, it does not solve the general problem of incorporating extensionality into higher-order tableaux. In fact, we can generalize the formula $\mathbf{C}{:=}(cb)\vee\neg c(\neg\neg b)$ to $\mathbf{C}'{:=}(c\mathbf{A})\vee\neg(c\mathbf{B})$, where $\mathbf{A}$ and $\mathbf{B}$ are arbitrary propositions. Now $\mathbf{C}'$ is valid in the class of Henkin models, iff $\mathbf{A}\Leftrightarrow\mathbf{B}$ is valid. So the approach of enhancing the unification would require augmenting the unification procedure by the theory of logical equivalence, which would enable the unification procedure to prove any theorem by unifying it with some elementary tautology like $\forall X_o.X\vee\neg X$.

The semantic problem with completeness behind this example is maybe best illustrated by the fact that in Σ-models does not hold and thus "buried occurrences" of propositions cannot be substituted for equivalent ones. Thus the remedy for the incompleteness of $\mathcal{HT}$ is a tableau construction rule that relates equality in unification constraints with equivalence.

Definition 4.1 (Extensional Tableau Calculus $\mathcal{HTE}$) The tableau construction rules for $\mathcal{HTE}$ consist of those for $\mathcal{HT}$ augmented with the one from figure 9. This rule interprets unification constraints of type o as equivalences.

$$\frac{\mathbf{A}_o\neq^?\mathbf{B}_o}{\begin{array}{c|c}\mathbf{A}^{\mathsf{F}}&\mathbf{A}^{\mathsf{T}}\\ \mathbf{B}^{\mathsf{T}}&\mathbf{B}^{\mathsf{F}}\end{array}}\,\mathcal{HTE}(ext)$$

$$\begin{array}{c}(\neg(cb)\vee c(\neg\neg b))^{\mathsf{F}}\\ \neg(cb)^{\mathsf{F}}\\ c(\neg\neg b)^{\mathsf{F}}\\ cb^{\mathsf{T}}\\ cb\neq^? c(\neg\neg b)\\ b\neq^?\neg\neg b\end{array}$$

$$\begin{array}{c}b\neq^?\neg\neg b\\ \neg\neg b^{\mathsf{F}}\\ b^{\mathsf{T}}\\ b^{\mathsf{F}}\\ b\neq^? b\end{array}\qquad\begin{array}{c}\\ \neg\neg b^{\mathsf{T}}\\ b^{\mathsf{F}}\\ b^{\mathsf{T}}\\ b\neq^? b\end{array}$$

Figure 9: Extensional Tableaux in $\mathcal{HTE}$

In particular we can continue our example above with the right sub-tableau in figure 9. Note that $\mathcal{HTE}$ completely blurs the distinction between propositional reasoning by decomposition of formulae and substitution construction by unification constraint solving. In particular we can indeed prove a sentence $\mathbf{A}$ by unifying it with a simple tautology, such as $\forall X_o.X\vee\neg X$. We will now explore this symmetry of propositional reasoning and unification.

Theorem 4.2 (Soundness and Completeness) *$\mathcal{HTE}$ is sound and refutation complete, i.e. a proposition $\mathbf{A} \in wff_o(\Sigma)$ is valid in all Henkin models, iff there is a closed higher-order extensional tableau for $\emptyset.\mathbf{A}^{\mathsf{F}}$.*

Proof: Soundness of $\mathcal{HTE}(ext)$ is immediate: If $\mathbf{A}$ and $\mathbf{B}$ have different truth values, then one must be T, while the other must be F. Completeness is a consequence of the model existence theorem for Henkin models 2.12. To show that

$$\nabla_\Sigma := \{\Phi \subset wff_o(\Sigma) \big| \not\vdash_{\mathcal{HTE}} \Phi^{\mathsf{T}}\}$$

is a saturated abstract consistency class, we build on 3.18, so we now only have to verify the condition 9.

Φ'	
$(\mathbf{A} =^o \mathbf{B})^{\mathsf{F}}$	
$\mathbf{A} \neq^? \mathbf{B}$	
$\mathbf{A}^{\mathsf{F}}$	$\mathbf{A}^{\mathsf{T}}$
$\mathbf{B}^{\mathsf{T}}$	$\mathbf{B}^{\mathsf{F}}$
$\mathcal{T}^1$	$\mathcal{T}^2$

Figure 10: Extensionality for ∇_Σ

Let $\Phi = \Phi' * \neg(\mathbf{A} =^o \mathbf{B})$, and $\mathcal{R}.\mathcal{T}^1$ a closed tableau for $\Phi \cup \{\neg\mathbf{A}, \mathbf{B}\}$ and finally $\mathcal{R}.\mathcal{T}^2$ one for $\Phi \cup \{\mathbf{A}, \neg\mathbf{B}\}$ then thetableau in figure 10 is a $\mathcal{HTE}$-refutation for Φ. Here Q is a new negative variable of type $\alpha \to o$. □

Now that we have seen how we can achieve completeness with respect to Henkin models, let us note that completeness with respect Σ-models is still an interesting property for a higher-order calculus: For the fragment of formulae where every propositional subformula (i.e. one of type o) is dominated only by logical constants – let us call such formulae *non-degenerate* – the two notions of validity coincide. Since for most applications degenerate do not occur, if suffices to employ calculi without rules such as $\mathcal{HTE}(ext)$.

5 Implementation: Concurrent Higher-Order Tableaux

The $\mathcal{HTE}$-calculus has been used as the basis of the HOT system [21], a concurrent higher-order tableau theorem prover that is intended for applications in mathematics and computational linguistics [20]. The system is implemented in OZ [23], a constraint programming language based on a new computation model providing a uniform foundation for higher-order functional programming, constraint logic programming, and concurrent objects with multiple inheritance. OZ is a concurrent programming language, i.e., a procedure may start sub-processes, called *threads*, which are executed concurrently in a fair way. HOT uses this feature to implement a blackboard architecture for higher-order tableaux where multiple tableau agents work together in order to construct a proof.

We can conceptualize tableaux implementations as blackboard architectures [9] where tableau agents, equipped with abilities from their underlying calculus, manipulate a Blackboard-like data structure, the tableau. The proof search space is defined by each agent's possible nondeterministic decisions about which $\mathcal{HTE}$ inference rule it applies. An agent may implement search strategies by choosing an order of rule applications or restricting the location of rule applications.

Tableau agents analyze disjunctions in separate branches of the tree. Because tree expansion is a potentially infinite process, each agent has to respect suitable fairness conditions for branch expansion. Since common strategies of first-order systems cannot be fully performed for $\mathcal{HTE}$ (higher-order formulae may change their propositional structure by primitive substitution) we have implemented a *concurrent tableau expansion*. Instead of a single tableau agent that has to decide on the order of branches to visit, we analyze disjunctive branches by multiple agents, each one working autonomously on its own part of the tableau. The first agent which is able close a tableau branch decides on the important choice of the next global variable substitution to explore, hopefully inhibiting unnecessary unification attempts in other parts of the tableau. In this way, the agents communicate with each other by manipulating the variable substitutions that are part of the blackboard. As long as the concurrent execution of the agents is fair, we emulate a weak form of breadth-first expansion by concurrency.

The concurrent architecture gives us an efficient method to cope with flex-flex pairs. Recall that flex-flex leaves close their branch until one of the heads becomes determined by a variable substitution. Each branch of a unification problem is part of the tableau, and therefore a unique agent deals with it. In the case of a branch ending in a flex-flex pair, the agent related to the branch simply suspends and waits for one of the flexible heads to become determined. An instantiation of one of the flexible heads will reactivate the agent, and the extension/closing cycle of the branch continues.

6 Conclusion

We have presented two calculi for automated theorem proving in higher-order logic and have semantically characterized their deductive power. We have chosen the framework of analytic tableaux for this presentation because of its simplicity and mathematical elegance, so that we could concentrate on the (syntactical and semantical) peculiarities of higher-order logic. It should be clear, that the methods developed here, carry over to all other automated theorem proving paradigms known from first-order logic.

The first calculus $\mathcal{HT}$ is complete relative to a non-standard semantics that also characterizes completeness of known higher-order refutation calculi like Huet's constrained resolution or Andrews' higher-order matings method. All of these calculi are not complete in the presence of buried occurrences of propositions, since substitutivity of equivalence is not admissible in them and thus they are not complete with respect to Henkin models. We have remedied this situation by introducing a special rule that trades propositional unification pairs

for equivalences and gives a complete calculus.

The calculi presented in this chapter use Leibniz formulation of equality instead of primitive equality. While this is theoretically sufficient, it is practically infeasible, since positive Leibniz equalities lead to flexible literals that are subject to primitive substitutions and therefore to combinatorial explosion of search spaces. Therefore it will be crucial to apply the methods described in this chapter to first-order calculi with equality. Since reduction orderings tend to be very weak for higher-order logic, it seems advantageous to focus on difference-reducing calculi like those discussed in [15], particularly, since the generalization of the colored unification algorithms needed to manipulate the search restrictions has already worked out [16].

Acknowledgments

While most of the work reported here is based on [18, 19], important parts of the treatment of extensionality [5, 6] have been developed in collaboration with Christoph Benzmüller and was funded by the DFG in Project HOTEL.

Karsten Konrad has implemented of the HOT theorem prover [21] in the LISA project of the Sonderforschungsbereich 378.

The earlier implementation experiments and a higher-order resolution theorem prover LEO [7], that have yielded valuable intuition for the course of the proofs have been carried out in the deduction tool-box $\mathcal{KEIM}$ which already provides most of the primitives needed for higher-order theorem provers. This has been funded by the DFG in the Schwerpunktprogramm "Deduktion" in the years 1992-1996. Without this implementation platform, the cost of experimenting would have been prohibitively high. The author is grateful to Daniel Nesmith and Detlef Fehrer of the $\mathcal{KEIM}$ project and Karsten Konrad for stimulating discussions about implementation issues, which have helped clarify various theoretical details.

References

[1] Peter B. Andrews. Resolution in type theory. *Journal of Symbolic Logic*, 36(3):414–432, 1971.

[2] Peter B. Andrews. General models descriptions and choice in type theory. *Journal of Symbolic Logic*, 37(2):385–394, 1972.

[3] Peter B. Andrews. On Connections and Higher Order Logic. *Journal of Automated Reasoning*, 5:257–291, 1989.

[4] Christoph Benzmüller. A calculus and a system architecture for extensional higher-order resolution. Research Report 97-198, Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh,USA, June 1997.

[5] Christoph Benzmüller and Michael Kohlhase. Model existence for higher-order logic. SEKI-Report SR-97-09, Universität des Saarlandes, 1997.

[6] Christoph Benzmüller and Michael Kohlhase. Resolution for henkin models. SEKI-Report SR-97-10, Universität des Saarlandes, 1997.

[7] Christoph Benzmüller and Michael Kohlhase. LEO – a higher order theorem prover. In Claude Kirchner and Hélène Kirchner, editors, *Proceedings of the 15th Conference on Automated Deduction*, number 1421 in LNAI, pages 139–144. Springer Verlag, 1998.

[8] Wolfgang Bibel. *Automated Theorem Proving*. Vieweg, Braunschweig, 1982.

[9] R. Engelmore and T. Morgan, editors. *Blackboard Systems*. Addison-Wesley, 1988.

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

[11] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte der Mathematischen Physik*, 38:173–198, 1931. English Version in [25].

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

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

[14] Gérard P. Huet. An unification algorithm for typed λ-calculus. *Theoretical Computer Science*, 1:27–57, 1975.

[15] Dieter Hutter. Colouring terms to control equational reasoning. *Journal of Automated Reasoning*, 18:399–442, 1997.

[16] Dieter Hutter and Michael Kohlhase. A coloured version of the λ-calculus. In William McCune, editor, *Proceedings of the 14th Conference on Automated Deduction*, number 1249 in LNAI, pages 291–305, Townsville, Australia, 1997. Springer Verlag.

[17] D. C. Jensen and T. Pietrzykowski. Mechanizing ω-order type theory through unification. *Theoretical Computer Science*, 3:123–171, 1976.

[18] Michael Kohlhase. *A Mechanization of Sorted Higher-Order Logic Based on the Resolution Principle*. PhD thesis, Universität des Saarlandes, 1994.

[19] Michael Kohlhase. Higher-order tableaux. In *Proceedings of the Tableau Workshop*, pages 294–309, Koblenz, Germany, 1995.

[20] Michael Kohlhase and Karsten Konrad. Higher-order automated theorem proving for natural language semantics. Seki Report SR-98-04, Fachbereich Informatik, Universität Saarbrücken, 1998.

[21] Karsten Konrad. Hot: A concurrent automated theorem prover based on higher-order tableaux. Seki Report SR-98-03, Fachbereich Informatik, Universität Saarbrücken, 1998.

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

[23] Programming Systems Lab Saarbrücken, 1998. The Oz Webpage: `http://www.ps.uni-sb.de/oz/`.

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

[25] Jean van Heijenoort, editor. *From Frege to Gödel A Source Book in Mathematical Logic, 1879-1931*. Source Books in the History of the Sciences. Harvard University Press, 1967.