Higher-Order Multi-Valued Resolution

Michael Kohlhase
Fachbereich Informatik
kohlhase@cs.uni-sb.de,

Ortwin Scheja
Fachbereich Mathematik
scheja@math.uni-sb.de

Universität des Saarlandes, 66041 Saarbrücken, Germany

ABSTRACT. This paper introduces a multi-valued variant of higher-order resolution and proves it correct and complete with respect to a variant of Henkin's general model semantics. This resolution method is parametric in the number of truth values as well as in the particular choice of the set of connectives (given by arbitrary truth tables) and even substitutional quantifiers. In the course of the completeness proof we establish a model existence theorem for this logical system. The work reported in this paper provides a basis for developing higher-order mechanizations for many non-classical logics.

KEY WORDS: higher-order logic, resolution, multi-valued, λ-calculus

---

1 Introduction

From the first attempts of modeling everyday reasoning within the framework of classical first-order logic it has been known that many relevant aspects cannot be adequately expressed in it. This has lead to many specialized logics in the field of artificial intelligence that use more than two truth values in order to deal with notions like vagueness, uncertainty, undefinedness and even non-monotonicity. These logics primarily obtain their specialized behavior by utilizing nonstandard truth tables for connectives and quantifiers.

On the other hand, the past 25 years have seen a tremendous increase in the deductive power of automated reasoning systems for standard predicate logic. These systems have reached the ability to solve non-trivial theorems fully automatically. Stimulated by this development, multi-valued logics are now treated in the context of deduction systems, i.e. with an emphasis on mechanization [Car87, Car91, BF92, Häh94].

A sorted version of these methods [KK97] was used to mechanize Kleene's strong logic for partial functions [Kle52], thus giving a clean foundation for first-order automated theorem proving in mathematics, where most functions are only defined on parts of the universe.

Another problem that has been addressed by multi-valued logics is that of so-called *presuppositions*, where natural language allows to draw conclusions that classical logic does not warrant.[1] These phenomena have been widely studied in the philosophy of language from a semantic point of view, but lack an efficient mechanization, which is a primary concern of artificial intelligence. See [KK97] for an attempt using four- and five-valued formalisms.

In spite of these successful attempts, a first-order language and mechanization is not fully satisfactory for real applications in mathematics and linguistics. They cannot be adequate for mathematics, since quantification over functions or predicates is widespread and not even Peano arithmetic can be fully axiomatized in first-order logic. They are also insufficient for linguistics, where many modern analyses of natural language make use of higher-order language constructs [DSP91, GK96] and integrate world-knowledge into the semantics construction process by higher-order theorem proving [GKvL96]. Since higher-order reasoning with presuppositions is ubiquitous in natural language [Kra95, Mus89], a higher-order version of [KK97] based on the results of this paper would provide a computational basis for a more adequate integration of world knowledge.

Of course higher-order multi-valued logics can be easily formalized and reasoned about in logical frameworks like Isabelle [Pau94] or Elf [Pfe91]. Such a definition only gives a natural-deduction calculus for the logics, and not a universal automated theorem prover. But full automation of the basic reasoning procedures is an equally important goal for applications in mathematics, linguistics and artificial intelligence.

To facilitate the development of specialized higher-order logics geared to these applications, we need a generalization of the multi-valued logic framework mentioned above to higher-order logic. In this paper, we try to do just that: We define a multi-valued higher-order logic $\mathcal{HOL}^n$ and present an appropriate resolution calculus $\mathcal{HR}^n$. To prove completeness of this calculus we establish a multi-valued model existence theorem that is a joint generalization of Andrews' "unifying principle" for higher-order logic [And71] and Carnielli's [Car87] for multi-valued first-order logics. Just like the latter, the method is parametric in the choice of the sets of truth values, connectives and quantifiers and can be instantiated yielding a mechanization for a wide range of particular logics. As a running example for this general framework we present a higher-order version Kleene's stong three-valued logic for partial functions mentioned above.

The choice of the resolution paradigm for automated theorem proving is not significant; since our proof theory relies on a model existence theorem and not so much on the calculus, an adaption of the higher-order tableau calculus from [Koh98] would be straightforward. Naturally, the results reported here

[1] For instance, albeit in classical first order logic there is the assumption that the universe of discourse is non-empty, it is not assumed that restrictions as usually expressed by implications are non-empty. A straightforward translation of "All humans are mortal" into first-order logic does not presuppose that there are any humans at all.

are widely applicable, they extend to all logical systems that combine multiple truth values with λ-binding and $\beta\eta$-conversion. We exemplify this by instantiating the methods developed in this paper to simple three-valued higher-order logic $\mathbf{K}^3$ of partial functions. Even if the target logic does not contain higher-order features, the added expressivity of $\mathcal{HOL}^n$ can admit simple and efficient relativizations (especially for first-order target logics, such as modal logics). Thus in these cases $\mathcal{HR}^n$ makes mechanization of the target logic much simpler than the first-order multi-valued frameworks.

2 Higher-Order Logic

In this paper we study a higher-order logic $\mathcal{HOL}^n$, which is parametric in the number n of truth values and the choice of connectives and quantifiers. It is based on the simply typed lambda calculus which we will shortly review in the following.

2.1 Syntax

Definition 2.1 (Types) Let $\mathcal{BT} := \{o, \iota\}$, then the set $\mathcal{T}$ of **types** is inductively defined to be the set $\mathcal{BT}$ together with all expressions $\alpha \to \beta$, where α and β are types. Here the base type ι stands for the set of **individuals** and the type o for the **truth values**. The functional type $\alpha \to \beta$ denotes the type of functions with domain α and codomain β. The types in $\mathcal{BT} \subset \mathcal{T}$ are called **base types**, types of the form $\alpha \to \beta$ are called **functional types**. We use the convention of association to the right for omitting parentheses in functional types, thus $\alpha \to \beta \to \gamma$ is an abbreviation for $(\alpha \to (\beta \to \gamma))$. This way the type $\gamma := \beta_1 \to \ldots \to \beta_k \to \alpha$ denotes the type of k-ary functions, that take k arguments of the types $\beta_1, \ldots, \beta_k$ and have values of type α. To conserve even more space we use a kind of vector notation and abbreviate γ by $\overline{\beta_k} \to \alpha$.

We will write finite functions like substitutions or variable assignments as sets of pairs $\varphi := [a^1/X^1], \ldots, [a^k/X^k]$ with the intended meaning that $\varphi(X^i) = a^i$. Furthermore we use the convention that $\psi := \varphi, [a/X]$ assigns a to X and coincides with φ everywhere else.

For the definition of well-formed formulae we fix a **signature** and a **collection of variables**, i.e. a typed collection $\Sigma := \bigcup_{\alpha\in\mathcal{T}} \Sigma_\alpha$ and $\mathcal{V} := \bigcup_{\alpha\in\mathcal{T}} \mathcal{V}_\alpha$ of symbols, such that each $\mathcal{V}_\alpha$ is countably infinite.

We denote the constants by lower case letters and the variables by upper case letters and use bold upper case letters $\mathbf{A}_\alpha$, $\mathbf{B}_{\alpha\to\beta}$, $\mathbf{C}_\gamma \ldots$ as syntactical variables for well-formed formulae.

Definition 2.2 (Well-Formed Formulae) For each $\alpha \in \mathcal{T}$ we define the set $\mathit{wff}_\alpha(\Sigma)$ of **well-formed formulae of type** α inductively:

1. $\Sigma_\alpha \subseteq \mathit{wff}_\alpha(\Sigma)$ and $\mathcal{V}_\alpha \subseteq \mathit{wff}_\alpha(\Sigma)$.

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

3. If $\mathbf{A} \in \mathit{wff}_{\alpha}(\Sigma)$, then $(\lambda X_{\beta}.\mathbf{A}) \in \mathit{wff}_{\beta\to\alpha}(\Sigma)$.

We call formulae of the form **AB** **applications**, and formulae of the form $\lambda X_{\alpha}.\mathbf{A}$ **λ-abstractions**. We will often write the type as a subscript $\mathbf{A}_{\alpha}$, if it is not irrelevant or clear from the context.

We adopt the usual definition of **free** and **bound** (all occurrences of the variable X in $\lambda X_{\alpha}.\mathbf{A}$ are called bound), variables and call a formula **closed**, iff it does not contain free variables. As in first-order logic the names of bound variables have no meaning at all, thus we consider alphabetic variants as identical and use a notion of substitution that systematically renames bound variables in order to avoid variable capture. We refer to formulae of type o as **propositions** and as **sentences** if they are closed.

We assume fixed subsets $\mathcal{J} = \bigcup_{l\in\mathbb{N}} \mathcal{J}^l \subseteq \Sigma$ of **connectives** and $\mathcal{Q} = \bigcup_{\alpha\in\mathcal{T}} \mathcal{Q}^{\alpha} \subseteq \Sigma$ of **quantifiers**. Here $\mathcal{J}^l$ is the set of l-ary connectives of type $\overline{o_l} \to o$ and $\mathcal{Q}^{\alpha} \subseteq \Sigma_{(\alpha\to o)\to o}$. We generally apply the convention that quantified expression $QX_{\alpha}.\mathbf{A}$ is an abbreviation of $Q^{\alpha}(\lambda X_{\alpha}.\mathbf{A})$, which is a well-formed formula (in the λ-calculus, quantifiers can be represented by ordinary constants, since the λ-binding mechanism can be utilized).

Remark 2.3 (Classical Higher-Order Logic) The syntax of classical higher-order logic ($\mathcal{CHOL}$) can be recovered as an instance of $\mathcal{HOL}^2$, where we have $\mathcal{J} = \{\neg_{o\to o}, \vee_{o\to o\to o}\}$ and $\mathcal{Q} = \{\Pi^{\alpha} \big| \alpha \in \mathcal{T}\}$. According to our convention, $\forall X_{\alpha}.\mathbf{A}$ is an abbreviation for $\Pi^{\alpha}(\lambda X_{\alpha}.\mathbf{A})$. For the semantics we refer to Remark 2.5.

In order to make the notation of well-formed formulae more legible, we use the convention that the group brackets (and) associate to the left and that the square dot **.** denotes a left bracket, whose mate is as far right as consistent with the brackets already present. Additionally, we combine successive λ-abstractions, so that the formulae $\lambda X^1 \dots X^n.\mathbf{AE}^1 \dots \mathbf{E}^m$ and $\lambda\overline{X^n}.\mathbf{A}\overline{\mathbf{E}^m}$ stand for $(\lambda X^1(\lambda X^2 \dots (\lambda X^n(\mathbf{AE}^1)\mathbf{E}^2 \dots \mathbf{E}^m)\cdots)$.

Let $\lambda \in \{\beta, \beta\eta, \eta\}$. We say that a well-formed formula $\mathbf{B}$ is obtained from a well-formed formula $\mathbf{A}$ by a **one-step λ-reduction** ($\mathbf{A} \Rightarrow_{\lambda} \mathbf{B}$), if it is obtained by applying one of the following rules to a well-formed part of $\mathbf{A}$.

β-Reduction $(\lambda X.\mathbf{C})\mathbf{D} \Rightarrow_{\beta} [\mathbf{D}/X]\mathbf{C}$.

η-Reduction If X is not free in $\mathbf{C}$, then $(\lambda X.\mathbf{C}X) \Rightarrow_{\eta} \mathbf{C}$.

As usual we denote the transitive reflexive closure of a reduction relation $\Rightarrow_{\lambda}$ with $\to_{\lambda}^{*}$. These rules induce equivalence relations $=_{\beta}$, $=_{\eta}$, and $=_{\beta\eta}$ on $\mathit{wff}(\Sigma)$, which we call the **λ-equality** relations. A formula that does not contain a λ-redex, and thus cannot be reduced by λ-reduction, is called a **λ-normal form**.

The λ-reduction relations are terminating and confluent, as the reader can convince himself by looking at the proofs for instance in [HS86]. Thus for any formula $\mathbf{A}$ there is a sequence of λ-reductions $\mathbf{A} \rightarrow_\lambda^* \mathbf{A}{\downarrow}$ such that $\mathbf{A}{\downarrow}$ is a **λ-normal form**.

2.2 Semantics

For the semantics we first define the higher-order algebras, which will serve as models for the underlying simply typed λ-calculus. Then we will specialize the type o of truth values to give the system its meaning of a multi-valued higher-order logic.

Higher-order algebras are built up from a **carrier set** $\mathcal{D}$, i.e. a collection $\mathcal{D} = \{\mathcal{D}_\alpha \big| \alpha \in \mathcal{T}\}$ of sets, such that $\mathcal{D}_{\alpha\to\beta}$ is a subset of the set $\mathcal{F}(\mathcal{D}_\alpha; \mathcal{D}_\beta)$ of functions from $\mathcal{D}_\alpha$ to $\mathcal{D}_\beta$, and a (well-typed) **interpretation** $\mathcal{I}\colon \Sigma \longrightarrow \mathcal{D}$.

We call a function $\varphi\colon \mathcal{V} \longrightarrow \mathcal{D}$ an **assignment**, iff $\varphi(X_\alpha) \in \mathcal{D}_\alpha$ for all variables X_α. A pair $\mathcal{A} := (\mathcal{D}, \mathcal{I})$ is a higher-order **algebra**, iff for each assignment φ the interpretation function $\mathcal{I}$ can inductively be extended to a total **value function** $\mathcal{I}_\varphi$: *wff*$(\Sigma) \rightarrow \mathcal{D}$ by the following rules.

1. $\mathcal{I}_\varphi(X) = \varphi(X)$, if X is a variable,
2. $\mathcal{I}_\varphi(c) = \mathcal{I}(c)$, if c is a constant,
3. $\mathcal{I}_\varphi(\mathbf{AB}) = \mathcal{I}_\varphi(\mathbf{A})\,[\mathcal{I}_\varphi(\mathbf{B})]$,
4. $\mathcal{I}_\varphi(\lambda X_\alpha \mathbf{.} \mathbf{B}_\beta)$ is the function $f \in \mathcal{D}_{\alpha\to\beta}$ such that $f(a) := \mathcal{I}_{\varphi,[a/X]}(\mathbf{B})$ for all $a \in \mathcal{D}_\alpha$.

Remark 2.4 (Term Algebra) Maybe the most prominent example of a higher-order algebra is the set *wff*$(\Sigma){\big\downarrow}$ of closed well-formed formulae in $\beta\eta$-normal form, together with $\mathcal{I} = \mathrm{Id}_\Sigma$. Here we consider formulae $\mathbf{A}$ of type $\alpha \rightarrow \beta$ as functions, such that $\mathbf{A}(\mathbf{B}) = (\mathbf{AB}){\big\downarrow}$. In this setting, assignments are ground substitutions and $\mathcal{I}_\varphi(\mathbf{A}) = \varphi(\mathbf{A}){\big\downarrow}$. Note that η-equality is essential for obtaining a higher-order algebra, since otherwise the resulting functions would not be extensional: For instance $\lambda X\mathbf{.}\mathbf{A}X$ and $\mathbf{A}$ are not β-equal but $(\lambda X\mathbf{.}\mathbf{A}X)\mathbf{B}$ and $\mathbf{AB}$ have the same β-normal form for all possible arguments $\mathbf{B}$.

So far the semantical notions do not make any requirements on the special type o of truth values. In contrast to classical higher-order logic, $\mathcal{HOL}^n$ has a finite set $\mathfrak{B}$ of **truth values** that has $n \geq 2$ elements. In this, we have a designated, nonempty subset $\mathfrak{T} \subseteq \mathfrak{B}$ that denotes those truth values, which are considered as *true* (in the sense that formulae that evaluate to a member of $\mathfrak{T}$ are valid).

We have claimed that $\mathcal{HOL}^n$ is parametric in the choice of the set of connectives and quantifiers. Indeed, the semantics makes no assumptions on the value $\widetilde{j} = \mathcal{I}(j)\colon \mathfrak{B}^l \rightarrow \mathfrak{B}$ for a connective $j \in \mathcal{J}^l$.

In first-order multi-valued logics the intended meaning of a quantifier Q is traditionally given as a function $\widetilde{Q}: \mathscr{P}^*(\mathfrak{B}) \longrightarrow \mathfrak{B}$, where we write $\mathscr{P}(M)$ for the power set of a set M and $\mathscr{P}^*(M) := \mathscr{P}(M) \setminus \emptyset$. With this, the value of a quantified expression is computed by applying $\widetilde{Q}$ to the set of truth-values of all of the instances of its scope.

In higher-order logic this construction is all the more natural, since a truth-function $\widetilde{Q}$ induces the value $\mathcal{I}(Q)$ that is defined by $\mathcal{I}(Q)(p) = \widetilde{Q}(p(\mathcal{D}_\alpha))$ for all $p \in \mathcal{D}_{\alpha\to o}$. Thus we have

$$\begin{aligned}\mathcal{I}_\varphi(QX_\alpha\mathbf{.A}) = \mathcal{I}_\varphi(Q^\alpha(\lambda X_\alpha\mathbf{.A})) &= \widetilde{Q}(\mathcal{I}_\varphi(\lambda X_\alpha\mathbf{.A})(\mathcal{D}_\alpha)) \\ &= \widetilde{Q}(\{\mathcal{I}_{\varphi,[a/X]}(\mathbf{A}) \big| a \in \mathcal{D}_\alpha\})\end{aligned}$$

Remark 2.5 (Semantics of $\mathcal{CHOL}$) Note that these definitions generalize the classical case, where $\mathfrak{B} = \{\mathsf{T}, \mathsf{F}\}$ and $\mathfrak{T} = \{\mathsf{T}\}$. The connectives $\neg$ and $\vee$ are given the well-known classical truth-functions. The choice that $\widetilde{\forall}(M)$ is true, iff $M = \{\mathsf{T}\}$ gives us the following value for the quantifier Π^α: $\mathcal{I}(\Pi^\alpha)(p) = \widetilde{\forall}(p(\mathcal{D}_\alpha)) = \mathsf{T}$ iff $p(\mathcal{D}_\alpha) = \{\mathsf{T}\}$. In other words, $\mathcal{I}(\Pi^\alpha)$ is the predicate that checks whether its argument is the universal predicate.

In the following we will assume that the truth tables $\widetilde{j}$ of connectives and truth functions $\widetilde{Q}$ for quantifiers are fixed (given by the user).

Definition 2.6 (Henkin Model) A higher-order algebra $\mathcal{A}$ is called a **Henkin model**, iff $\mathcal{D}_o = \mathfrak{B}$, $\mathcal{I}(j) = \widetilde{j}$ and $\mathcal{I}(Q)(f) = \widetilde{Q}(\{f(a) \big| a \in \mathcal{D}_\alpha\})$ for any $j \in \mathcal{J}$ and $Q \in \mathcal{Q}^\alpha$.

The class of **standard models** (where we furthermore require that $\mathcal{D}_{\alpha\to\beta}$ is the set of *all* functions $\mathcal{D}_\alpha \longrightarrow \mathcal{D}_\beta$) is in some way the most natural notion of semantics for $\mathcal{HOL}^n$. However, Gödel's incompleteness result shows that there cannot be calculi that are complete with respect to the notion induced by this semantics, a fact that makes it virtually useless for our purposes.

In this paper we use an even weaker semantics than Henkin models which does not make the strong assertions about extensionality on $\mathcal{D}_o$ that Henkin models do[2]. Treating extensionality would lead to a more complex resolution calculus [BK98], and we want to concentrate on the issues of multiple truth values here. Hence we do not require that $\mathcal{D}_o = \mathfrak{B}$, but make the more general assumption that there exists a valuation (a mapping that respects the intended meaning of connectives and quantifiers) from $\mathcal{D}_o$ to $\mathfrak{B}$.

Definition 2.7 (Frege Model) Let $\mathcal{A} = (\mathcal{D}, \mathcal{I})$ be a higher-order algebra, then a surjective total function $\upsilon: \mathcal{D}_o \longrightarrow \mathfrak{B}$ with

1. $\upsilon(\mathcal{I}(j)(a_1, \ldots, a_l)) = \widetilde{j}(\upsilon(a_1), \ldots, \upsilon(a_l))$ for any $j \in \mathcal{J}^l$.

[2]The fact that in $\mathcal{CHOL}$ we have $\mathcal{D}_o = \{\mathsf{T}, \mathsf{F}\}$ implies that equivalent propositions can be substituted for each other.

2. $\upsilon\left(\mathcal{I}(Q)(f)\right) = \widetilde{Q}\left(\{\upsilon(f(a)) \middle| a \in \mathcal{D}_\alpha\}\right)$ for any $Q \in \mathcal{Q}^\alpha$.

is called a **valuation for** $\mathcal{A}$. In this case we call the triple $\mathcal{M} := (\mathcal{D}, \mathcal{I}, \upsilon)$ a **Frege model**. For a given assignment φ the evaluation of a formula $\mathbf{A}$ consists of the interpretation $\mathcal{I}_\varphi(\mathbf{A})$ in $\mathcal{A}$ and the subsequent valuation with υ. Thus we call a formula $\mathbf{A} \in \mathit{wff}_o(\Sigma)$ **valid** in $\mathcal{M}$ under an assignment φ ($\mathcal{M} \models_\varphi \mathbf{A}$), iff $\upsilon \circ \mathcal{I}_\varphi(\mathbf{A}) \in \mathfrak{T}$.

2.3 A Three-Valued Instance

We will use a three-valued instance $\mathbf{K}^3$ of our generic logic $\mathcal{HOL}^n$ as a running example[3]. $\mathbf{K}^3$ is a variant of Kleene's strong three-valued logic for recursive partial predicates on natural numbers [Kle52]. It shares its truth values and the truth-tables and truth-functions. In particular we have $\mathfrak{B} = \{\mathsf{T}, \mathsf{F}, \bot\}$ and $\mathfrak{T} = \{\mathsf{T}\}$. Here the third truth value $\bot$ is intended for atomic formulae that contain a non-denoting subformula, such as $\frac{1}{0}$ or the predecessor of zero. $\mathbf{K}^3$ has the same sets of connectives and quantifiers as classical logic, which have extended truth functions:

$\widetilde{\vee}$	F	$\bot$	T
F	F	$\bot$	T
$\bot$	$\bot$	$\bot$	T
T	T	T	T

$\widetilde{\neg}$	
F	T
$\bot$	$\bot$
T	F

$$\widetilde{\forall}(M) := \begin{cases} \mathsf{T} & \text{for } M = \{\mathsf{T}\} \\ \bot & \text{for } M = \{\mathsf{T}, \bot\} \text{or } \{\bot\} \\ \mathsf{F} & \mathsf{F} \in M \end{cases}$$

Even though $\mathbf{K}^3$ uses the same truth-function $\widetilde{\forall}$ for the universal quantifier, it differs from Kleene's logic in the definition of quantification itself: To model partial functions Kleene assumes an error element $\bot$ and excludes $\bot$ from the domain of quantification. This is necessary, since he furthermore assumes that all functions and predicates are strict with respect to $\bot$ (i.e. if $\mathcal{I}_\varphi(\mathbf{A}) = \bot$, then $\mathcal{I}_\varphi(f\mathbf{A}) = \bot$ and $\mathcal{I}_\varphi(p\mathbf{A}) = \bot$), which together with unrestricted quantification would lead to a logic without theorems.

To arrive at a higher-order logic for partial functions, we formalize Kleene's additional assumptions by an error constant $\bot_\alpha \in \Sigma_\alpha$ for all types $\alpha \in \mathcal{T}$ where $\mathcal{I}(\bot_o) = \bot$, and the following **strictness axioms**.

$$\forall F_{\alpha\to\beta}. F\bot_\alpha = \bot_\beta \tag{1}$$

Furthermore, to model restricted quantification, we define the restricted Kleene quantifier $\forall_r$, such that $\forall_r X_\alpha.\mathbf{A}$ abbreviates $\forall X_\alpha.(X \neq \bot_\alpha) \Rightarrow \mathbf{A}$.

Remark 2.8 (Equality in $\mathbf{K}^3$) For modelling partial functions, we need an equality that is a strict binary relation, i.e. $\mathbf{A} = \mathbf{B}$ should be undefined whenever $\mathbf{A}$ or $\mathbf{B}$ is; the primitive equality we have used above in the strictness axioms and to define restricted quantification is not, since $\bot = \bot$.

[3] The purpose of this logic is only to serve as a demonstration object for the methods developed in this paper, an efficient mechanization would need sort reasoning as we have shown in [KK97, KK94]

We have two options to formalize this kind of equality in $\mathbf{K}^3$:

- We can assume a second primitive equality constant $=_s^\alpha$ of type $\alpha \rightarrow \alpha \rightarrow o$ for each type α, such that $\mathcal{I}(=_s^\alpha)$ is the strict identity relation on $\mathcal{D}_\alpha$, i.e. the unique strict relation that is the identity on $\mathcal{D}_\alpha \setminus \{\mathcal{I}(\bot_\alpha)\}$. This essentially means that we have to augment the strictness axioms by axioms for transitivity, reflexivity, symmetry, and substitutivity of equality.

- We can define a strict equality relation via a variant of the well-known Leibniz formulation:

  $$=_s^\alpha \; := \lambda X_\alpha Y_\alpha \boldsymbol{.} \forall P_{\alpha\rightarrow o} \boldsymbol{.} (!PX \wedge !PY) \equiv (PX \Rightarrow PY)$$

  Here, we need an additional connective ! for definedness such that $\mathcal{I}(!)$ is true on T and F and false on $\bot$.

  The intuition behind the Leibniz formula is that two objects are equal, iff there are no discerning properties P. Note that $=_s$ is strict, since the formula $\mathbf{A} =_s \bot$ reduces to

  $$\forall P_{\alpha\rightarrow o} \boldsymbol{.} (!P\mathbf{A} \wedge !P\bot) \equiv (P\mathbf{A} \Rightarrow P\bot)$$

  This universal statement is undefined, since we can take P to be the property that is true on all its (defined) arguments: It makes $(!P\mathbf{A} \wedge !P\bot)$ false but makes $(P\mathbf{A} \Rightarrow P\bot)$ undefined, since it makes $P\mathbf{A}$ true and $P\bot$ undefined.

Remark 2.9 An alternative to this multi-valued approach to partial functions is the two-valued logic $\mathcal{PF}$ proposed by Bill Farmer [Far90]. $\mathcal{PF}$ is a variant of $\mathcal{CHOL}$, where the function universes $\mathcal{D}_{\alpha\rightarrow\beta}$ in Henkin models are sets of *partial functions* instead of total functions. The logic also models partiality using error elements, but interprets atomic propositions as false, if one of the arguments is undefined. While this leads to a simpler logical system, it has the severe disadvantage of yielding unwanted theorems such as

$$\forall_r X, Y \boldsymbol{.} \frac{X}{Y} = Z \Rightarrow X = Y {*} Z$$

This formula is considered problematic in mathematics, since it neglects the condition that $Y \neq 0$. It is a theorem of $\mathcal{PF}$ even for $Y = 0$, since then $X/Y = Z$ is false and thus the implication true as a whole.

Now we can use $\mathbf{K}^3$ to formalize a simple mathematical fact about function division, namely a cancellation law for real functions:

> **Theorem**: *For all real functions F and G, the product of F/G and G is F, provided that G is nowhere zero.*

Note the use of quantification over functions in this example.

Example 2.10 Of the real numbers we use the constants $0, 1$ and the functions **inv** and $*$. We will use the symbols $*$ and $\div$ on functions, defined by

$$\begin{aligned} * &= (\lambda F, G\lambda X\mathbf{.}(FX)*(GX)) & (2)\\ \div &= (\lambda F, G\mathbf{.}F*(\lambda X\mathbf{.inv}(GX))) & (3) \end{aligned}$$

In order to prove the theorem we need the following axioms of elementary calculus.

$$\begin{aligned} \forall_r X\mathbf{.}(\mathbf{inv}X =_s \bot_\iota) \equiv X =_s 0 \qquad (4)\\ \forall_r X\mathbf{.}(X \neq_s 0) \Rightarrow (\mathbf{inv}X*X) =_s 1 \qquad (5) \end{aligned}$$

together with associativity of $*$ and the unit axiom for 1 and $*$. In the theory defined by axioms (1)–(5), the theorem stated above has the form

$$\forall_r F, G\mathbf{.}(\forall_r X\mathbf{.}GX \neq_s 0 \wedge GX \neq \bot) \Rightarrow (*(\div FG)G) =_s F \qquad (6)$$

3 Model Existence

In this section we introduce an important tool for proving completeness results in higher-order logic. Model existence theorems state that sets which belong to a so-called abstract consistency class are satisfiable. With their help the completeness proof for a given logical system $\mathcal{C}$ is reduced to the (purely proof-theoretic) demonstration that the class of $\mathcal{C}$-consistent sets is an abstract consistency class. This proof technique was first introduced by Smullyan in [Smu68] based on work by Hintikka and Beth. It was later generalized to higher-order logic by Andrews in [And71] and to multi-valued first-order logics by Carnielli [Car87]. Since there is no simple Herbrand theorem in higher-order logic, Andrews' theorem has become the standard method for completeness proofs in higher-order logic.

We call a pair $\mathbf{A}^w$ a **labeled formula**, iff $\mathbf{A} \in \mathit{wff}_o(\Sigma)$ and $w \in \mathfrak{B}$. For a labeled formula $\mathbf{A}^w$ we require $\upsilon \circ \mathcal{I}_\varphi(\mathbf{A}) = w$. As usual we can derive a notion of satisfiability from this.

For the definition of an abstract consistency class we must consider the relation of satisfiability of a labeled formula $j\overline{\mathbf{A}}^w$ to the values of its subformulae $\mathbf{A}_i$. The immediate answer to this question is that $\mathcal{I}_\varphi(j\overline{\mathbf{A}}) = \widetilde{j}(\overline{\mathcal{I}_\varphi(\mathbf{A})})$ and thus $(\mathcal{I}_\varphi(\mathbf{A}_1), \ldots, \mathcal{I}_\varphi(\mathbf{A}_l)) \in \widetilde{j}^{-1}(w)$ is the relevant condition. However it is possible to optimize this condition, if $\widetilde{j}$ is constant on some argument. We formalize this in the notion of a $\mathcal{P}$-consequence, which has been introduced under the name Π-consequence by Carnielli [Car87]. We avoid the latter name for higher-order logic, since it could lead to confusion with the Π^α-quantifier.

Definition 3.1 ($\mathcal{P}$-Consequence) Let $\mathfrak{B}^* := \mathfrak{B} \cup \{*\}$ and $\vec{v}_l = (v_1, \ldots, v_l)$ and $\vec{w}_l = (w_1, \ldots, w_l)$ be members of ${\mathfrak{B}^*}^l$, then we say that $\vec{v}_l$ is **more general** than $\vec{w}_l$ ($\vec{v}_l \sqsubseteq \vec{w}_l$), if for some $\mathcal{N} \subseteq \{1, \ldots, n\}$ we have $v_k = *$ for all $k \in \mathcal{N}$ and $w_i = v_i$ for all $i \notin \mathcal{N}$. Intuitively, higher generality can be obtained by replacing some components of a vector by $*$. For a sequence $\overline{\mathbf{A}} = \mathbf{A}_1 \ldots \mathbf{A}_l$ of formulae we write $\overline{\mathbf{A}}^{\vec{v}_l}$ for the set $\{\mathbf{A}_i^{v_i} | v_i \neq *\}$. Asterisks mark positions without influence on the value of connective formulae; they can be left out of consideration while forming semantic consequences. Let us extend the function $\widetilde{j}$ to all of $\mathfrak{B}^*$ by inductively defining $\widetilde{j}(v_1, \ldots, v_{i-1}, *, v_{i+1}, \ldots, v_l) = v$ whenever $\widetilde{j}(v_1, \ldots, v_{i-1}, w, v_{i+1}, \ldots, v_l) = v$ for all $w \in \mathfrak{B}$ and undefined else. Then we call the set $\mathcal{H}_v(j) := \widetilde{j}^{-1}(v) \subseteq {\mathfrak{B}^*}^n$ the **propositional condition** for a connective j and the truth value v. From this we can choose a set $\mu_{\sqsubseteq}\mathcal{H}_v(j)$ of generators (a vector $\vec{v}_l$ generates the set of all $\vec{w}_l$ with $\vec{v}_l \sqsubseteq \vec{w}_l$). Now let $\mathbf{B}^v$ be a labeled formula of the form $(j\mathbf{A}_1 \ldots \mathbf{A}_l)^v$, then a set $\overline{\mathbf{A}}^{\vec{w}_l}$ is called a **$\mathcal{P}$-consequence of $\mathbf{B}^v$**, iff $\vec{w}_l \in \mu_{\sqsubseteq}\mathcal{H}_v(j)$.

Example 3.2 For the connectives of $\mathbf{K}^3$ we have the following $\mathcal{P}$-consequences:

	$\mathcal{P}$-cons.
$(\neg\mathbf{A})^{\mathsf{F}}$	$\{\mathbf{A}^{\mathsf{T}}\}$
$(\neg\mathbf{A})^{\perp}$	$\{\mathbf{A}^{\perp}\}$
$(\neg\mathbf{A})^{\mathsf{T}}$	$\{\mathbf{A}^{\mathsf{F}}\}$

	$\mathcal{P}$-consequences
$(\mathbf{A} \vee \mathbf{B})^{\mathsf{F}}$	$\{\mathbf{A}^{\mathsf{F}}\}, \{\mathbf{B}^{\mathsf{F}}\}$
$(\mathbf{A} \vee \mathbf{B})^{\perp}$	$\{\mathbf{A}^{\perp}, \mathbf{B}^{\perp}\}, \{\mathbf{A}^{\perp}, \mathbf{B}^{\mathsf{T}}\}, \{\mathbf{A}^{\mathsf{T}}, \mathbf{B}^{\perp}\}$
$(\mathbf{A} \vee \mathbf{B})^{\mathsf{T}}$	$\{\mathbf{A}^{\mathsf{T}}, \mathbf{B}^{\mathsf{T}}\}$

Note that the set of $\mathcal{P}$-consequences can be empty. Furthermore this construction is not necessarily unique for connectives of arity $k > 2$ (for a detailed discussion see [Häh94]). In order to have a unique notion of $\mathcal{P}$-consequence we fix some general method of constructing these sets in advance. This a priori choice makes the presentation of the calculus and the completeness arguments simpler; for an actual application and implementation of the calculus in section 4, where optimization is essential, it may be necessary to compute $\mathcal{P}$-consequences on a case-by-case basis. We do not consider the optimization of multi-valued calculi [Häh96, Sal96] here, since it is independent of higher-order logic.

For the construction of term models it is necessary to introduce formulae that contain witnesses for existential expressions. For this we assume a countably infinite set of new **witness** constants (which we will denote with $\mathtt{f}_\alpha, \mathtt{f}_\beta \ldots$) for each type.

Definition 3.3 (Set of witnesses) Let $M = \{w_1, \ldots, w_m\} \subseteq \mathfrak{B}$ be a set of truth values, $\mathbf{A} \in \mathit{wff}_{\alpha\to o}(\Sigma)$ a closed formula and Φ be a set of sentences. We call a set $\Xi_M(\mathbf{A}) := \{(\mathbf{A}\mathtt{f}_1)^{w_1}, \ldots, (\mathbf{A}\mathtt{f}_m)^{w_m}\}$ a **set of M-witnesses for $\mathbf{A}$ in environment** Φ, if $\mathtt{f}_1, \ldots, \mathtt{f}_m$ are new for Φ.

Definition 3.4 (Abstract Consistency Class) Let ∇ be a class of sets of labeled propositions, then ∇ is called an **abstract consistency class**, iff ∇ is closed under subsets, and for all sets $\Phi \in \nabla$ the following conditions hold:

$\nabla 1$ There are no truth values $v \neq w$, such that $\mathbf{A}^v, \mathbf{A}^w \in \Phi$ for an atomic formula $\mathbf{A} \in \mathit{wff}_o(\Sigma)$.

$\nabla 2$ If $\mathbf{A}^w \in \Phi$, then $\Phi \cup \{\, \mathbf{A}{\downarrow}^w\} \in \nabla$.

$\nabla 3$ If $\mathbf{B}^v = (j\mathbf{A}_1 \ldots \mathbf{A}_l)^v \in \Phi$, then there is a $\mathcal{P}$-consequence Ψ of $\mathbf{B}^v$, such that $\Phi \cup \Psi \in \nabla$.

$\nabla 4$ If $(Q^\alpha \mathbf{A})^v \in \Phi$, then there is a set $M \subseteq \mathfrak{B}$ of truth values with $\widetilde{Q}(M) = v$, such that for any set $\Xi_M(\mathbf{A})$ of M-witnesses in the environment Φ and for each closed formula $\mathbf{B} \in \mathit{wff}_\alpha(\Sigma)$ there is a truth value $w \in M$, such that $\Phi \cup \Xi_M(\mathbf{A}) \cup \{(\mathbf{AB})^w\} \in \nabla$.

We call an abstract consistency class **saturated**, iff for all $\Phi \in \nabla$ and all propositions $\mathbf{A} \in \mathit{wff}_o(\Sigma)$ we have $\Phi \cup \{\mathbf{A}^v\} \in \nabla$ for some $v \in \mathfrak{B}$.

Remember that abstract consistency is intended to be a notion of consistency that is independent of a particular calculus. Thus the first condition just states that there may not be elementary contradictions in "consistent" sets, whereas the second one is a closure condition for $\beta\eta$-equality. The remaining conditions state that a "consistent" set of propositions can be extended by certain logical preconditions without loosing "consistency". In contrast to the two-valued case, n-valued quantifiers have in general both existential and universal nature, thus it is necessary to extend by preconditions that contain arbitrary instances as well as witnesses. The condition $\nabla 2$ is the only one that is particular to higher-order logic, the others are analogous to their first-order counterparts (see for instance [Car91])

The significance of abstract consistency classes lies in the following theorem.

Theorem 3.5 (Existence of Frege Models) *Let ∇ be a saturated abstract consistency class and $H \in \nabla$, then there is a Frege model $\mathcal{M}$ with $\mathcal{M} \models H$.*

Proof: A class Γ of sets is said to be of finite character, if for any set H the condition $H \in \Gamma$ is equivalent to $K \in \Gamma$ for every finite subset K of H. We can assume that ∇ is of finite character, since (saturated) abstract consistency classes can be extended conserving abstract consistency and saturation.

The set H can be extended to a higher-order Hintikka set, i.e. a set $\mathcal{H}$ that is maximal in ∇ with $H \subseteq \mathcal{H}$ by the well-known Henkin completion procedure: For a given enumeration $\mathbf{A}_1, \mathbf{A}_2, \ldots$ of propositions, we construct a sequence H_i of finite supersets of H by adding $\mathbf{A}_i$ to H_i, iff $H_i \cup \{\mathbf{A}_i\} \in \nabla$. The set $\mathcal{H} := \bigcup H_i$ is in ∇, since we have assumed ∇ to be of finite character. It is also maximal, since for any proposition $\mathbf{A}_i$, such that $\mathcal{H} \cup \{\mathbf{A}_i\} \in \nabla$, we have $\mathbf{A}_i \in H_{i+1} \subset \mathcal{H}$; thus $\mathcal{H}$ is a Hintikka set.

From this we can build a Frege model $(\mathcal{D}, \mathcal{I}, \upsilon)$ that satisfies $\mathcal{H}$ by choosing $\mathcal{D}$ to be the set of closed formulae in $\beta\eta$-normal forms, and $\mathcal{I}$ to be the identity on constants and finally $\upsilon(\mathbf{A}_o) = w$, iff $\mathbf{A}^w \in \mathcal{H}$. It is well-known, that $(\mathcal{D}, \mathcal{I})$ is a higher-order algebra (Recall Remark 2.4: Assignments φ into $\mathcal{D}$ are just ground substitutions and $\mathcal{I}_\varphi(\mathbf{A})$ is the $\beta\eta$-normal form of $\varphi(\mathbf{A})$, which is closed, since φ is ground.), so it only remains to verify that υ is a valuation. This is a consequence of the maximality of Hintikka sets for saturated abstract consistency classes, which gives us stronger variants of the abstract consistency conditions.

$\mathcal{H}1$ For any proposition $\mathbf{A} \in \mathit{wff}_o(\Sigma)$ we have $\mathbf{A}^v \in \mathcal{H}$ for exactly one truth value $v \in \mathfrak{B}$.

(∇1 gives us the assertion for atoms and at most one truth value, saturation for at least one truth value. The generalization to arbitrary propositions is proven by structural induction.)

$\mathcal{H}2$ $\mathbf{A}^w \in \mathcal{H}$, iff $\mathbf{A}{\downarrow}^w \in \mathcal{H}$.

(The converse direction of ∇2 follows by maximality of $\mathcal{H}$ in ∇: If $\mathbf{A}^w \notin \mathcal{H}$, but $\mathbf{A}{\downarrow}^w \in \mathcal{H}$, then $\mathbf{A}^v \in \mathcal{H}$ by $\mathcal{H}1$ for some $v \neq w$, thus $\mathbf{A}{\downarrow}^v \in \mathcal{H}$ by ∇2 contradicting $\mathcal{H}1$.)

$\mathcal{H}3$ $\mathbf{B}^v = (j\mathbf{A}_1 \ldots \mathbf{A}_l)^v \in \mathcal{H}$, iff there is a $\mathcal{P}$-consequence Ψ of $\mathbf{B}^v$, such that $\Psi \subseteq \mathcal{H}$.

$\mathcal{H}4$ $(Q^\alpha \mathbf{A})^v \in \mathcal{H}$, iff there is a set $M \subseteq \mathfrak{B}$ of truth values with $\widetilde{Q}(M) = v$, such that for any set $\Xi_M(\mathbf{A})$ of M-witnesses in the environment $\mathcal{H}$ and for each closed formula $\mathbf{B} \in \mathit{wff}_\alpha(\Sigma)$ there is a truth value $w \in M$, such that $\Xi_M(\mathbf{A}) \in \mathcal{H}$ and $(\mathbf{AB})^w \in \mathcal{H}$.

The first two properties give totality and well-definedness of the function υ, while the second two guarantee the two valuation conditions from Definition 2.7. □

Note that with this construction we can only obtain Frege models, not Henkin models, since the set of closed formulae of sort o is different from $\mathfrak{B}$.

4 Resolution ($\mathcal{HR}^n$)

Now that we have specified the semantics we can turn to the exposition of our resolution calculus $\mathcal{HR}^n$. There are three main differences to the first-order case. First, higher-order unification is undecidable, therefore we cannot simply use it as a sub-procedure that is invoked during resolution. The solution for this problem is to treat the unification problem as a constraint and residuate it in the resolution and factoring rules. In fact we use negative equality literals that are disjunctively bound to the clause (cf. 4.2).

Not all instantiations for predicate variables can be found by unification. For completeness the instantiations of head variables of literals must contain logical constants, which cannot be supplied by unification, since they are not even present in the clauses set, as they have been eliminated in the clause normal form transformations.

Finally, naive Skolemization is not sound for higher-order logic: For instance in $\mathcal{CHOL}$ it is possible to use it to prove a version of the axiom of choice, which is known to be independent. In this paper we will follow Dale Miller's solution (cf. [Mil83]) to this problem: His idea is to introduce arities for the witness constants (we call the resulting pair $\mathtt{f}_\alpha^k$, where the arity k is smaller than the length of α a **Skolem constant**). Then the language is restricted to so-called **Skolem formulae**, where all Skolem constants $\mathtt{f}_\alpha^k$ have all their necessary arguments (i.e. at least k of them) and furthermore no variables occurring in necessary arguments of Skolem constants are bound outside. Note that the resulting fragment (Miller calls it the "Herbrand Universe") is well-defined and closed under (restricted) substitutions and β-reduction and long normal forms.

Remark 4.1 Alternatively, we could have used Dale Miller's quantifier "raising" [Mil92] which is dual to Skolemization, but does not need Skolem constants, or we could have directly encoded the dependency relation introduced by the quantifier sequencing into an explicit variable relation that is part of the clauses (for details see [Koh98]).

4.1 Clause Normal Form

In our definition of clauses, we will use disjunctions as meta-symbols for sets of formulae, in order to enhance legibility. Note that since the disjuncts are labeled formulae, these can easily be discerned from the disjunction constants which might appear in the signature.

Definition 4.2 (Clause) If $\mathbf{M}_i \in \mathit{wff}_o(\Sigma)$ and $v_i \in \mathfrak{B}$, then we call a formula $\mathcal{D} := \mathbf{C} \vee \mathcal{E}$ a **generalized clause**, if $\mathbf{C}$ is of the form $\mathbf{C} := \mathbf{M}_1^{v_1} \vee \ldots \vee \mathbf{M}_n^{v_n}$, and if $\mathcal{E}$ is a disjunction of pairs of the form $\mathbf{A}_1 \neq^? \mathbf{B}_1 \vee \ldots \vee \mathbf{A}_m \neq^? \mathbf{B}_m$ (we will consider unification pairs of the form $\mathbf{A}_\alpha \neq^? \mathbf{B}_\alpha$ as literals, since this will simplify the presentation). We call $\mathcal{D}$ a **clause**, iff the $\mathbf{M}_i^{v_i}$ are literals (a labeled formula $\mathbf{A}^v$ is called **literal**, if the head of $\mathbf{A}$ is a parameter or variable). In order to conserve space we will write disjunctions of the form $\bigvee_{v \in V} \mathbf{A}^v$ as $\mathbf{A}^V$, so $\mathbf{A}^v \vee \mathbf{A}^w$ becomes $\mathbf{A}^{vw}$.

In $\mathcal{HR}^n$ the transformation to clause normal form only need two parametric rules, one for the connectives

$$\frac{\mathbf{C} \vee (j\overline{\mathbf{A}})^v \quad \mathbf{D} \in \bigvee_{\vec{w}_l \in \mu_{\sqsubseteq} \mathcal{H}_v(j)} \overline{\mathbf{A}}^{\vec{w}_l}}{\mathbf{C} \vee \mathbf{D}} \mathcal{RC} : j$$

which basically transforms a labeled connective formula $(j\overline{\mathbf{A}})^v$ into the cross product of all it's $\mathcal{P}$-consequences $\overline{\mathbf{A}}^{\overline{w}_l}$; and one for the quantifiers

$$\frac{\mathbf{C} \vee (Q^\alpha \mathbf{A})^w \quad \mathbf{D} \in \bigvee_{M \in \widetilde{Q}^{-1}(w)} \beta(M)}{\mathbf{C} \vee \mathbf{D}} \ \mathcal{RC} : Q$$

where $\beta(M) = \{(\mathbf{A}(\mathtt{f}_i^k \overline{X_k}))^{w_i} | w_i \in M\} \cup \{(\mathbf{A} Z_\alpha)^M\}$ with $\{X_1, \ldots, X_k\}$ is the set of free variables of $\mathbf{A}$ and and where Z_α is a new variable. This set plays the role of the set $\Xi_M(\mathbf{A})$ of witnesses defined in 3.3. The key difference is that instead of arbitrary instances $\beta(M)$ uses variables that will be instantiated appropriately by unification.

For a a given set Ψ of generalized clauses we call the set $\mathbf{cnf}(\Psi)$ of clauses that is derivable from Ψ the **clause normal form** of Ψ. Since in order to show that a sentence $\mathbf{A} \in \mathit{wff}_o(\Sigma)$ is valid (i.e. obtains a truth value in $\mathfrak{T}$), it is sufficient to refute that $\mathbf{A}$ obtains a truth value in $\mathfrak{B} \setminus \mathfrak{T}$, we define the **refutation clause form** of a set Φ of sentences as

$$\mathbf{RCF}(\Phi) = \bigcup_{\mathbf{A} \in \Phi} \mathbf{cnf}\left(\mathbf{A}^{\mathfrak{B} \setminus \mathfrak{T}}\right)$$

If we apply the rules above to classical higher-order logic, we obtain the traditional clause normal form reductions for $\neg$ and $\vee$, but a quantifier reduction that is significantly less efficient. Fortunately, wide classes of naturally occurring quantifiers admit generic optimizations [Häh94, Sch94] that yield the classical rules for $\mathcal{CHOL}$. This also holds for our running example $\mathbf{K}^3$, where we obtain the following (optimized) transformation rules. For instance the $\forall^\perp$ rule, where the number of introduced clauses is decreased from six to two.

$$\frac{\mathbf{C} \vee (\mathbf{A} \vee \mathbf{B})^{\mathsf{T}}}{\mathbf{C} \vee \mathbf{A}^{\mathsf{T}} \vee \mathbf{B}^{\mathsf{T}}} \qquad \frac{\mathbf{C} \vee (\mathbf{A} \vee \mathbf{B})^{\mathsf{F}}}{\mathbf{C} \vee \mathbf{A}^{\mathsf{F}} \quad \mathbf{C} \vee \mathbf{B}^{\mathsf{F}}}$$

$$\frac{\mathbf{C} \vee (\mathbf{A} \vee \mathbf{B})^{\perp}}{\mathbf{C} \vee \mathbf{A}^{\perp\mathsf{F}} \quad \mathbf{C} \vee \mathbf{B}^{\perp\mathsf{F}} \quad \mathbf{C} \vee \mathbf{A}^{\perp} \vee \mathbf{B}^{\perp}}$$

$$\frac{\mathbf{C} \vee (\neg\mathbf{A})^{\mathsf{T}}}{\mathbf{C} \vee \mathbf{A}^{\mathsf{F}}} \qquad \frac{\mathbf{C} \vee (\neg\mathbf{A})^{\perp}}{\mathbf{C} \vee \mathbf{A}^{\perp}} \qquad \frac{\mathbf{C} \vee (\neg\mathbf{A})^{\mathsf{F}}}{\mathbf{C} \vee \mathbf{A}^{\mathsf{T}}}$$

$$\frac{\mathbf{C} \vee (\Pi^\alpha \mathbf{A})^{\mathsf{T}}}{\mathbf{C} \vee (\mathbf{A} X)^{\mathsf{T}}} \qquad \frac{\mathbf{C} \vee (\Pi^\alpha \mathbf{A})^{\mathsf{F}}}{\mathbf{C} \vee (\mathbf{A}(\mathtt{f}_i^k \overline{X_k}))^{\mathsf{F}}}$$

$$\frac{\mathbf{C} \vee (\Pi^\alpha \mathbf{A})^\perp}{\mathbf{C} \vee (\mathbf{A}(\mathtt{f}_i^k \overline{X_k}))^\perp \quad \mathbf{C} \vee (\mathbf{A}X)^{\perp\mathsf{T}}}$$

The regularity of $\mathbf{K}^3$ allows us to optimize this clause normal form even further: As first noticed by Rainer Hähnle [Häh94], clause normalization can be more efficient, if we process disjunctions $\mathbf{L}^{\{v_1,\ldots,v_n\}}$ (written as $\mathbf{L}^{v_1\cdots v_n}$) in one step. In particular for $\mathbf{K}^3$, labeled formulae containing literals $\mathbf{L}^{\mathsf{T}\perp\mathsf{F}}$ are tautologous and can be deleted and normalization rules acting on $\mathbf{A}^{\mathsf{T},\perp}$ or $\mathbf{A}^{\mathsf{F},\perp}$ (intuitively meaning that the formula $\mathbf{A}$ must not be F/T) are much more regular than the combination of the T, and $\perp$ rules induced by the disjunctions. For instance we have the following rules for sets of signs

$$\frac{\mathbf{C} \vee (\mathbf{A} \vee \mathbf{B})^{\mathsf{T}\perp}}{\mathbf{C} \vee \mathbf{A}^{\mathsf{T}\perp} \vee \mathbf{B}^{\mathsf{T}\perp}} \qquad \frac{\mathbf{C} \vee (\mathbf{A} \vee \mathbf{B})^{\mathsf{F}\perp}}{\mathbf{C} \vee \mathbf{A}^{\mathsf{F}\perp} \quad \mathbf{C} \vee \mathbf{B}^{\mathsf{F}\perp}}$$

$$\frac{\mathbf{C} \vee (\Pi^\alpha \mathbf{A})^{\mathsf{T}\perp}}{\mathbf{C} \vee (\mathbf{A}X)^{\mathsf{T}\perp}} \qquad \frac{\mathbf{C} \vee (\Pi^\alpha \mathbf{A})^{\mathsf{F}\perp}}{\mathbf{C} \vee (\mathbf{A}(\mathtt{f}_i^k \overline{X_k}))^{\mathsf{F}\perp}}$$

Let us now return to our example 2.10 to prove Theorem 6, we have to consider the clause normal form of the set of axioms (1)–(5) labeled with T, together with (6) labeled with $\mathsf{F}\perp$ (we have to refute that it obtains the truth values F or $\perp$). Using the optimized reduction rules above the refutation clause form of our example 2.10 has the following form:

A1 $(F_{\iota\to\iota}\perp_\iota = \perp_\iota)^{\mathsf{T}}$
A2 $X = \perp^{\mathsf{T}} \vee (\mathbf{inv}X = \perp)^{\mathsf{F}} \vee (X =_s 0)^{\mathsf{T}}$
A3 $X = \perp^{\mathsf{T}} \vee (\mathbf{inv}X = \perp)^{\mathsf{T}} \vee (X =_s 0)^{\mathsf{F}}$
A4 $(X = \perp)^{\mathsf{T}} \vee (X =_s 0)^{\mathsf{T}} \vee (\mathbf{inv}X{*}X) =_s 1^{\mathsf{T}}$
T1 $(\mathtt{f}^0 = \perp)^{\mathsf{F}\perp}$
T2 $(\mathtt{g}^0 = \perp)^{\mathsf{F}\perp}$
T3 $(X = \perp)^{\mathsf{T}\perp} \vee (\mathtt{g}^0 X =_s 0)^{\mathsf{F}\perp}$
T4 $(X = \perp)^{\mathsf{T}\perp} \vee (\mathtt{g}^0 X = \perp)^{\mathsf{F}\perp}$
T5 $((\lambda Y.(\mathtt{f}^0 Y){*}(\mathbf{inv}(\mathtt{g}^0 Y)){*}(\mathtt{g}^0 Y)) =_s \mathtt{f}^0)^{\mathsf{F}\perp}$

where A1 comes from strictness (1), A2-A3 from (4), and A4 comes from (5). The theorem clauses T1 to T5 have been obtained from (6) by eliminating definitions (2) and (3) and clause normalizing.

We will not execute the refutation here, since multi-valued resolution proofs look almost exactly the same as classical resolution proofs (the special features of the logic only come into play during the clause form transformation) and the particular refutation of our example is rather large (especially after expanding the equality with the Leibniz formula from 2.8).

To show the correctness of the normalization process, we first have to take a look at the concept of satisfiability for clauses. This is nearly straightforward:

a clause is satisfiable, iff one of its literals is. However, due to the non-standard nature of Skolem constants, they may not be interpreted as normal functions in the model, but rather in a **Skolem extension** of $\mathcal{M}$. This extends the carrier $\mathcal{D}$ of $\mathcal{M}$ by a carrier $\mathcal{S}$ for the Skolem constants, where $\mathcal{S} = \{\mathcal{S}_\alpha^k\}$ with

$$\mathcal{S}^k_{\overline{\alpha_k} \to \beta} = \mathcal{F}(\mathcal{D}_{\alpha_1} \times \cdots \times \mathcal{D}_{\alpha_k}; \mathcal{D}_\beta)$$

and extends $\mathcal{I}$, such that Skolem constants $\mathtt{f}_\alpha^k$ are interpreted in $\mathcal{S}_\alpha^k$. Note that since we only consider the restricted fragment, where all necessary arguments of Skolem constants are present, the obvious value function $\mathcal{I}_\varphi$ is well-defined. Thus a clause $\mathcal{C}$ is satisfiable in a Skolem model $\mathcal{M} = (\mathcal{D}, \mathcal{S}, \mathcal{I})$, iff there is either a literal $\mathbf{L}^v$ in $\mathcal{C}$, such that $\mathcal{I}_\varphi(\mathbf{L}) = v$ or a pair $\mathbf{A} \neq^? \mathbf{B}$, such that $\mathcal{I}_\varphi(\mathbf{A}) \neq \mathcal{I}_\varphi(\mathbf{B})$ for some assignment φ.

Theorem 4.3 (Refutation Clause Form Theorem) *A set Φ of sentences is valid, if* $\mathbf{RCF}(\Phi)$ *is unsatisfiable.*

Proof: Recall that a set Φ of sentences is valid in a model $\mathcal{M} = (\mathcal{D}, \mathcal{I})$, iff for all propositions $\mathbf{A} \in \Phi$ we have $\mathcal{I}_\varphi(\mathbf{A}) \in \mathfrak{T}$, or equivalently, if the initial generalized clause $\mathbf{I} := \mathbf{A}^{\mathfrak{B}\setminus\mathfrak{T}}$ is unsatisfiable in the Skolem extension $\mathcal{M}^S = (\mathcal{D}, \mathcal{S}, \mathcal{I})$ of $\mathcal{M}$. Note that since $\mathbf{I}$ does not contain any Skolem constants, the two notions of satisfiability coincide. Thus the proof of the assertion reduces to checking that the clause form transformation conserves satisfiability. For the connective cases, this is unproblematic:

$$\frac{\mathbf{C} \vee (j\overline{\mathbf{A}})^r \quad \mathbf{D} \in \bigvee_{\vec{w_l} \in \mu_{\sqsubseteq}\mathcal{H}_r(j)} \overline{\mathbf{A}}^{\vec{w_l}}}{\mathbf{C} \vee \mathbf{D}} \mathcal{RC} : j$$

Let $\mathcal{M}^S$ satisfy the $\mathbf{C} \vee (j\overline{\mathbf{A}})^r$, we can assume that $\mathcal{I}_\varphi(j\overline{\mathbf{A}}) = r$, since otherwise the assertion is trivial. Thus $\vec{w_l} = (\upsilon(\mathcal{I}_\varphi(\mathbf{A}_1)), \ldots, \upsilon(\mathcal{I}_\varphi(A_l))) \in \widetilde{j}^{-1}(r) = \mathcal{H}_r(j)$ and consequently one of the $\mu_{\sqsubseteq}\mathcal{H}_r(j)$ is more general than $\vec{w_l}$. Since the line of reasoning above only depends of the variable assignment φ, for any such φ one of the $\mathcal{P}$-consequences and thus $\mathbf{D}$ will be satisfied.

In the quantifier case, we are in the following situation

$$\frac{\mathbf{C} \vee (Q^\alpha \mathbf{A})^w \quad \mathbf{D} \in \bigvee_{M \in \widetilde{Q}^{-1}(w)} \beta(M)}{\mathbf{C} \vee \mathbf{D}} \mathcal{RC} : Q$$

Let $\mathcal{M}^S$ satisfy the $\mathbf{C} \vee (Q^\alpha \mathbf{A})^w$, as above, we can assume that $\mathcal{I}_\varphi(Q^\alpha \mathbf{A}) = w$. Thus it remains to show that we can extend $\mathcal{I}$ for the new Skolem constants, such that it satisfies the new clause $\mathbf{D}$ introduced in the rule. Since we have assumed that $\mathcal{I}_\varphi(Q^\alpha \mathbf{A}) = w$, we have $\widetilde{Q}(M) = w$, where $M = \{\upsilon(\mathcal{I}_\varphi(\mathbf{A})(a)) \big| a \in$

$\mathcal{D}_\alpha\}$ and thus $M \in \widetilde{Q}^{-1}(w)$. Clearly, the proof is complete, if we can show that every clause $\mathbf{K}$ in $\beta(M)$ is satisfiable. For this we have to consider two cases. If $\mathbf{K} = (\mathbf{A}Z_\alpha)^M$, then we know that $\upsilon(\mathcal{I}_\varphi(\mathbf{A}Z)) \in M$ for any variable assignment φ. In other words, $\mathbf{K}$ is satisfiable.

If $\mathbf{K} = \mathbf{A}(\mathtt{f}^k\overline{X_k})^r$ for some $r \in M$, where $\{X_1, \ldots, X_k\}$ are the free variables of $\mathbf{A}$ with types α_i, then we know that there is an $a = a_\varphi \in \mathcal{D}_\alpha$ with $\upsilon(\mathcal{I}_\varphi(\mathbf{A})(a)) = r$ by construction. Since $\mathcal{I}_\varphi(\mathbf{A})$ only depends on the φ-values on $\{X_1, \ldots, X_k\}$, we can choose $\mathcal{I}(\mathtt{f}^k) \in \mathcal{S}^k_{\overline{\alpha_k}\to\alpha}$ as the function $(\varphi(X_1), \ldots, \varphi(X_k)) \mapsto a_\varphi$. Note that as φ varies over all variable assignments, $(\varphi(X_1), \ldots, \varphi(X_k))$ covers $\mathcal{D}_{\alpha_1} \times \cdots \times \mathcal{D}_{\alpha_k}$. $\square$

4.2 Higher-Order Unification

Now we will briefly review higher-order unification and its properties, for details we refer the reader to [Sny91]. The algorithm consists of two parts. A deterministic, terminating simplification part decomposes terms, and establishes variables bindings for partial solutions and directly generalizes first-order unification. This leaves unification pairs in a form, where both formulae are applications and at least one has a variable at its head. The strategy of the non-deterministic part is to bind this head variable to a (most general) formula that enables further simplification. Thus the head of the binding must either match that of the other formula, or be a bound variable, that (upon β-reduction) projects up the head of a subformula. Technically, the right notion of binding is presented in Definition 4.4, which we give in detail, since we need it independently.

Definition 4.4 (General Binding) Let $\alpha = (\overline{\beta_l} \to \gamma)$, and h be a constant or variable of type $(\overline{\delta_m} \to \gamma)$, then $\mathbf{G} := \lambda\overline{X^l_{\alpha_l}}.h\overline{\mathbf{V}^m}$ is called a **general binding of type α and head h**, if $\mathbf{V}^i = H^i\overline{X^l_{\alpha_l}}$. If h is the Skolem constant f^k, then $\mathbf{V}^i = H^i$ for $i \leq k$, since otherwise we leave the Herbrand Universe.

The H^i are new variables of types $\overline{\beta_l} \to \delta^i$ (or δ_i for $i \leq k$if $h = f^k$). It is easy to show that general bindings indeed have the type and head claimed in the name and are most general in the class of all such terms. Moreover they are unique up to the choice of variable names.

General bindings, where the head is a bound variable $X^j_{\beta_j}$ are called **projection bindings** (we write them as $\mathcal{G}^j_\alpha(\Sigma)$) and **imitation bindings** (written $\mathcal{G}^h_\alpha(\Sigma)$) else. Since we need both imitation and projection bindings for higher-order unification, we collect them in the set of **Approximations**

$$\mathcal{A}^h_\alpha(\Sigma) := \{\mathcal{G}^h_\alpha(\Sigma)\} \cup \{\mathcal{G}^j_\alpha(\Sigma) \big| j \leq l\} .$$

The non-deterministic part becomes problematic in the case of so-called **flex-flex** pairs, where both formulae of a pair have head variables, since the head of the binding has to be guessed. Fortunately, in the application in automated theorem proving, it is sufficient to guarantee the existence of an arbitrary unifier, rather than calculate it. Thus for deduction purposes we only need pre-unification, a variant of higher-order unification that considers flex-flex pairs as solved (see [Sny91] for details).

4.3 The Resolution Calculus $\mathcal{HR}^n$

Now we turn to the actual resolution calculus $\mathcal{HR}^n$. The previous results set the stage by giving a semantic justification of a resolution calculus that proves sentences $\mathbf{A}$ by converting $\mathbf{A}^{\mathfrak{B}\setminus\mathfrak{T}}$ to clause normal form and then by deriving the empty clause from that. Intuitively, this refutes that possibility that $\mathbf{A}$ obtains a value in $\mathfrak{B} \setminus \mathfrak{T}$ in order to prove that it indeed obtains a value in $\mathfrak{T}$ and thus is valid.

Definition 4.5 (Higher-Order Resolution ($\mathcal{HR}^n$)) The calculus $\mathcal{HR}^n$ is a variant of Huet's resolution calculus from [Hue72], and has the following rules of inference:

$$\frac{\mathbf{N}^v \vee \mathbf{C} \quad \mathbf{M}^w \vee \mathbf{D} \quad v \neq w}{\mathbf{C} \vee \mathbf{D} \vee \mathbf{M} \neq^? \mathbf{N}} \mathcal{HR}:R$$

$$\frac{\mathbf{M}^v \vee \mathbf{N}^v \vee \mathbf{C}}{\mathbf{M}^v \vee \mathbf{C} \vee \mathbf{M} \neq^? \mathbf{N}} \mathcal{HR}:F$$

which operate on the clause part of clauses. For manipulating the unification constraints $\mathcal{HR}^n$ utilizes pre-unification rules (cf. [Sny91]) of which we will only state the most interesting one:

$$\frac{\mathbf{C} \vee F_\alpha \overline{\mathbf{U}} \neq^? h \overline{\mathbf{V}}}{\mathbf{C} \vee F \neq^? \mathbf{G} \vee F \overline{\mathbf{U}} \neq^? h \overline{\mathbf{V}}} \mathcal{HR}:f/r$$

Here $\mathbf{G}$ is a general binding in $\mathcal{A}^h_\alpha(\Sigma)$. The following inference rule

$$\frac{F_\alpha \overline{\mathbf{U}}^v \vee \mathbf{C}}{\mathbf{C} \vee F \overline{\mathbf{U}}^v \vee F \neq^? \mathbf{P}} \mathcal{HR}:P$$

generates instantiations for flexible literals, i.e. literals where the head symbol is a positive variable. Here $\mathbf{P} \in \mathcal{A}^k_\alpha(\Sigma)$ is a general binding of type α that approximates some logical constant $k \in \mathcal{J} \cup \mathcal{Q}$. $\mathcal{HR}^n$ has one further inference rule

$$\frac{\mathbf{C} \vee \mathcal{E} \vee X \neq^{?} \mathbf{A}}{\mathbf{D}} \mathcal{HR}: E$$

where $X \neq^{?} \mathbf{A}$ is solved in $\mathcal{E} \vee X \neq^{?} \mathbf{A}$ and $\mathbf{D} \in \mathbf{RCF}([\mathbf{A}/X]\mathbf{C} \vee [\mathbf{A}/X]\mathcal{E})$. This rule propagates partial solutions from the constraints to the clause part, and thus helps detect clashes early. Since the instantiation may well change the propositional structure of the clause by instantiating a predicate variable, we have to renormalize the resulting generalized clause on the fly.

We call a clause **empty**, iff it does not contain any proper literals and its unification constraint is pre-solved (i.e. contains only solved pairs $X \neq^{?} \mathbf{A}$ or flex/flex pairs). Clearly any empty clause is unsatisfiable with respect to Frege models, since the constraint is solvable. We will call a set Ψ of generalized clauses **refutable**, iff the empty clause is derivable from it and a set Φ of sentences **provable**, iff $\mathbf{RCF}(\Phi)$ is refutable.

In contrast to Huet's calculus we allow pre-unification transformations to be applied to clauses during the resolution process. This generalization allows us to investigate more realistic strategies than in Huet's calculus, which uses the "lazy unification" strategy, that only allows unification to happen after a clause has been derived that only consists of unification constraints.

Theorem 4.6 (Soundness) *If a set Φ of propositions is provable, then it is valid.*

Proof: The soundness is a simple consequence of the soundness of unification and the refutation clause form theorem 4.3 since the resolution and factoring rules residuate the appropriate unification constraint. □

Lemma 4.7 *Let Φ be a set of generalized clauses, θ a substitution, and $\mathcal{D}$ a refutation of $\theta(\Phi)$. Then there is a derivation $\mathcal{D}'\colon \Phi \vdash_{\mathcal{HR}} \mathcal{E}$, where $\mathcal{E}$ is a set of pairs. Furthermore there is an extension θ' of θ, such that*

- *θ' unifies $\mathcal{E}$, and*
- *the new variables in the domain of θ' do not occur in Φ.*

Proof sketch: The derivation $\mathcal{D}'$ is constructed along the line of $\mathcal{D}$. In order to do this, it is essential to maintain a close correspondence between the clause sets involved (see the notion of a clause set isomorphism in [Koh94]). Note that the clause normal form transformations from $\mathcal{D}$ can also be applied to the corresponding clauses in Φ with the exception of the case, where the clause in Φ contains a flexible literal, whose head θ instantiates with a formula whose head is a logical constant. Here the transformation from $\mathcal{D}$ must be mimicked by using the $\mathcal{HR}: P$ rule that introduces the appropriate constant. Since the

$\mathcal{HR} : P$ rule contains an application of $\mathcal{HR} : E$, the ensuing clause normal form transformation makes it possible to update the correspondence. Thus by a simple inductive argument we see that the clause normal form transformation part of $\mathcal{D}$ can be lifted to a $\mathcal{HR}$-derivation.

The rest of $\mathcal{D}$ can then be lifted one inference rule at a time. The only two interesting aspects of this:

- In the lifting of the $\mathcal{HR} : E$ rule, we can have the case, that again θ introduces logical constants in the codomain of the eliminated variables. Fortunately, this can be solved by exactly the argument above.

- The clause isomorphism can be destroyed by the fact that literals in $\mathcal{D}$ may correspond to more than one literal in $\mathcal{D}'$, then we use $\mathcal{HR} : F$ to collapse them (restoring the correspondence).

The results on θ' are obtained by maintaining θ along with the correspondence (updating it with the primitive substitutions) and carefully analyzing unifiability conditions. $\square$

Theorem 4.8 (Completeness) $\mathcal{HR}^n$ *is complete for Frege models.*

Proof: The proof is conducted by verifying that the property of clause sets not to be refutable is a saturated abstract consistency property. So by the model existence theorem 3.5 we see that non-refutable sets of generalized clauses are satisfiable in the class of Frege models. Since this is just the contrapositive of the statement of completeness, we have finished the proof. Thus it only remains to verify the conditions of 3.4.

$\nabla 1$ We prove the converse: Assume there are literals $\mathbf{A}^v, \mathbf{A}^w \in \Phi$ for $v \neq w$, then Φ is refutable, since there are unit clauses $\mathbf{A}^v$ and $\mathbf{A}^w$ in the clause normal form of Φ, which can be resolved to the empty clause as the resulting unification constraint $\mathbf{A} \neq^? \mathbf{A}$ is trivially solvable.

$\nabla 2$ This condition is trivially met, since the clause normal form is invariant under $\beta\eta$-equality.

$\nabla 3$ Again we prove the converse: Assume that for each $\mathcal{P}$-consequence $\mathbf{C}^i$ of a formula $(j\overline{\mathbf{A}})^w$ there is a refutation of $\Phi \cup \mathbf{C}$. We inductively merge these refutations together to a refutation $\mathcal{D}$ of $\Phi \cup \mathbf{C}^1 \otimes \cdots \otimes \mathbf{C}^l$ where $\Psi \otimes \Theta$ is the set

$$\{\mathbf{A} \vee \mathbf{B} \big| \mathbf{A} \in \mathbf{cnf}(\Psi); \mathbf{B} \in \mathbf{cnf}(\Theta)\}$$

For this construction we use a technical result (disjunction lemma) that refutations of $\Xi \cup \Psi$ and $\Xi \cup \Theta$ imply the existence of a refutation of $\Xi \cup (\Psi \otimes \Theta)$. We conclude the proof by remarking that $\mathbf{C}^1 \otimes \cdots \otimes \mathbf{C}^l$ is just the clause normal form of $(j\overline{\mathbf{A}})^w$.

∇4 Let $(Q\overline{\mathbf{A}})^v \in \Phi$. We have to show that the existence of a family of refutations $\mathcal{D}_M^w$ of $\Phi \cup \Xi_M(\mathbf{A}) \cup \{(\mathbf{AB})^w\}$, where $M \in \widetilde{Q}^{-1}(v)$ and $w \in M$ implies the existence of a refutation $\mathcal{D}$ of Φ.

Remember that in the clause normal form reduction, $(Q\overline{\mathbf{A}})^v$ is transformed to generalized clauses of the form $\mathbf{L}^1 \vee \ldots \vee \mathbf{L}^k$, where the $\mathbf{L}^i$ come from some $\beta(M^i)$. From refutations $\mathcal{D}_M^w$, $w \in M$, we will construct refutations $\mathcal{D}_M$ of $\Phi \cup \beta(M)$. With a disjunction lemma technique similar to the one above the $\mathcal{D}_M$ are combined to a refutation $\mathcal{D}$ of $\Phi \cup \{\mathbf{L}_1 \vee \ldots \vee \mathbf{L}_k \big| \mathbf{L}_i \in \beta(M_i)\}$, which has the same refutation clause form as Φ. Thus $\mathcal{D}$ is indeed the refutation needed to complete the proof.

Let us fix a $M = \{w_1, \ldots, w_m\} \in \widetilde{Q}^{-1}(v)$, then

$$\beta(M) = \{(\mathbf{A}(\mathtt{f}_i^k \overline{X_k}))^{w_i} \big| w_i \in M\} \cup \{(\mathbf{A}X)^M\}$$

For each $\mathcal{D}_M^w$, $w \in M$, the lifting lemma (cf. 4.7 take $\theta = [\mathbf{B}/X]$) guarantees a derivation $\mathcal{F}_M^w \colon \Phi \cup \Xi_M(\mathbf{A}) \cup \{(\mathbf{A}X)^w\} \vdash_{\mathcal{HR}} \mathcal{C}$, where the resulting clause $\mathcal{C}$ only contains a set $\mathcal{E}^w$ of pairs. Again by a disjunction lemma technique, we can combine these to a derivation

$$\mathcal{F}_M \colon \Phi \cup \Xi_M(\mathbf{A}) \cup \{(\mathbf{A}X)^{w_1} \vee \ldots \vee (\mathbf{A}X)^{w_k}\} \vdash_{\mathcal{HR}} \mathcal{E}^{w_1} \vee \ldots \vee \mathcal{E}^{w_n}$$

The solutions θ^w of the $\mathcal{E}^w$ from the lifting lemma can be combined to a substitution $\theta_M = \theta^{w_1} \cup \ldots \cup \theta^{w_k}$, since they agree on X. Thus $\mathcal{E}^{w_1} \vee \ldots \vee \mathcal{E}^{w_n}$ is pre-unifiable and hence (higher-order unification is complete) there is a derivation $\mathcal{H}_m$ (using only pre-unification steps) that derives the empty clause from $\mathcal{E}^{w_1} \vee \ldots \vee \mathcal{E}^{w_n}$. Finally we remark the Skolem subterms (i.e. the Skolem constants with all their necessary arguments) from the clause form transformation directly correspond to the witness constants in the abstract consistency property.

Finally, to convince ourselves that ∇ is saturated, we show that for any set $\Phi \in \nabla$ and any proposition $\mathbf{A}$, there must be a truth value $w \in \mathfrak{B}$, such that $\Phi \cup \{\mathbf{A}^w\} \in \nabla$. To prove the contraposition, let us assume that $\Phi \cup \{\mathbf{A}^w\} \notin \nabla$ for every $w \in \mathfrak{B}$, in other words that there are refutations for all $\Phi \cup \{\mathbf{A}^w\}$. By the disjunction lemma, there must be a refutation for $\Phi \cup \{\mathbf{A}^{\mathfrak{B}}\}$. This contradicts our assumption that $\Phi \in \nabla$, since the last clause is a tautology and cannot contribute to a refutation. $\square$

5 Conclusion

We have presented a multi-valued higher-order logic $\mathcal{HOL}^n$ and a higher-order resolution calculus $\mathcal{HR}^n$ that is sound and complete with respect to multi-valued Frege models. Since this logical system combines multiple truth values and parametric choice of connectives and quantifiers with higher-order features, such as λ-binding and $\beta\eta$-conversion, it is a suitable basis for the development of artificial intelligence logics. However, as we have seen in the example, $\mathcal{HOL}^n$ can only be a starting point for the development of a higher-order logic with partial functions. In order for an adequate treatment of quantification (which must exclude the undefined element for a higher-order account of partial functions) it will be necessary to combine it with the sort techniques of [Koh94] in the spirit of [KK97]. This will yield a suitable basis for formalizing and mechanizing informal mathematical vernacular. Similarly, given a more general treatment of generalized quantifiers we will obtain a higher-order mechanization of presuppositions, as a basis for an adequate integration of world knowledge and pragmatics in the process of natural language semantics construction.

References

[And71] Peter B. Andrews. Resolution in type theory. *Journal of Symbolic Logic*, 36(3):414–432, 1971.

[BF92] Matthias BaazChristian G. Fermüller. Resolution for many-valued logics. In *LPAR'92*, 624 *LNAI*, 107–118. Springer Verlag, 1992.

[BK98] C. Benzmüller and M. Kohlhase. Extensional Higher-Order Resolution. in *Proc. CADE-15*, LNAI, Springer Verlag, 1998.

[Car87] Walter A. Carnielli. Systematization of finite many-valued logics through the method of tableaux. *Journal of Symbolic Logic*, 52:473–493, 1987.

[Car91] Walter A. Carnielli. On sequents and tableaux for many-valued logics. *Journal of Non-Classical Logic*, 8(1):59–76, 1991.

[DSP91] Mary Dalrymple, Stuart ShieberFernando Pereira. Ellipsis and higher-order-unification. *Linguistics and Philosophy*, 14:399–452, 1991.

[Far90] William M. Farmer. A partial-function version of Church's simple theory of types. *Journal of Symbolic Logic*, 55:1269–1291, 1990.

[GK96] Claire GardentMichael Kohlhase. Higher–order coloured unification and natural language semantics. In *Proceedings of ACL'96*. Association for Computational Linguistics, Santa Cruz, 1996.

[GKvL96] Claire Gardent, Michael KohlhaseNoor van Leusen. Corrections and Higher-Order Unification. In *Proceedings of KONVENS'96*, 268–279. De Gruyter, 1996.

[Häh94] Reiner Hähnle. *Automated Deduction in Multiple-Valued Logics*, 10 *International Series of Monographs on Computer Science*. Oxford University Press, 1994.

[Häh96] Reiner Hähnle. Exploiting data dependencies in many-valued logics. *Journal of Applied Non-Classical Logics*, 6(1):49–69, 1996.

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

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

[KK94] Manfred KerberMichael Kohlhase. Mechanising Partiality without Re-Implementation. In Alan Bundy, , *KI'97*, 1303 in LNAI, Springer Verlag, 1997.

[KK97] Manfred KerberMichael Kohlhase. A mechanization of strong Kleene logic for partial functions. In Alan Bundy, , *CADE'94*, 814 in LNAI, 371–385, Springer Verlag, 1994.

[KK97] Manfred KerberMichael Kohlhase. Reasoning without believing: On the mechanization of presuppositions and partiality. To appear in Lopez, Manandhar, Nutt eds. *Computational Logic for Natural Language Understanding*, Springer Verlag.

[Kle52] Stephen C. Kleene. *Introduction to Meta-Mathematics.* North Holland, 1952.

[Koh94] Michael Kohlhase. *A Mechanization of Sorted Higher-Order Logic Based on the Resolution Principle.* , Universität des Saarlandes, 1994.

[Koh98] Michael Kohlhase. Higher-order automated theorem proving. In Wolfgang BibelPeter Schmitt, , *Automated Deduction – A Basis for Applications*, 2, Kluwer, 1998. .

[Kra95] Emiel Krahmer. *Discourse and Presupposition.* , University of Tilburg, 1995.

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

[Mil92] Dale Miller. Unification under a mixed prefix. *Journal of Symbolic Computation*, 14:321–358, 1992.

[Mus89] R.A. Muskens. *Meaning and Partiality.* , University of Amsterdam, 1989.

[Pau94] Lawrence C. Paulson. *Isabelle: A Generic Theorem Prover.* 828 of LNCS. Springer Verlag, 1994.

[Pfe91] Frank Pfenning. Logic programming in the LF logical framework. In Gérard P. HuetGordon D. Plotkin, , *Logical Frameworks.* Cambridge University Press, 1991.

[Sal96] Gernot Salzer. Optimal axiomatizations for multiple-valued operators and quantifiers based on semilattices. In M.A. McRobbieJ.K. Slaney, , *CADE'96*, 1104 in LNAI, 688–702, Springer Verlag, 1996.

[Sch94] Ortwin Scheja. Resolution in mehrwertigen Logiken höherer Stufe. SEKI-Working-Paper SWP-94-07, Universität des Saarlandes, 1994.

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

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