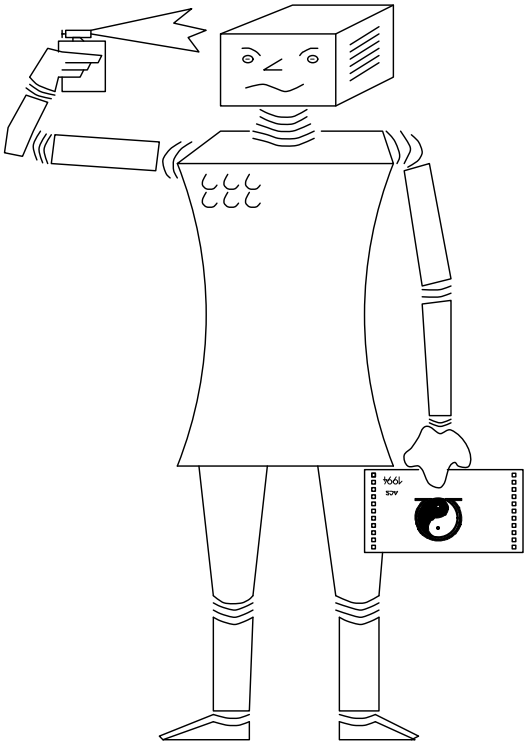
SEKI Report



Higher-Order Multi-Valued Resolution

Michael Kohlhase, Ortwin Scheja

SEKI Report SR-95-04

Also published as: short version submitted to ECAI'96

Higher-Order Multi-Valued Resolution

Michael Kohlhase
Fachbereich Informatik
kohlhase@cs.uni-sb.de,
tel/fax: (49)-681-302-4627/4421

Ortwin Scheja
Fachbereich Mathematik
scheja@math.uni-sb.de
tel: (49)-681-302-2230

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 natural multi-valued 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.

1 Introduction

Many specialized logics in the field of artificial intelligence use more than two truth values in order to deal with notions like vagueness, uncertainty, undefinedness and even non-monotonicity. Moreover, 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 two-valued first-order predicate logic. These systems have reached the ability to solve nontrivial theorems fully automatically. Stimulated by this development, multi valued logics, after they had been of theoretic interest in the field of mathematical logic in the past, are nowadays treated in the context of deduction systems, i.e. with an emphasis on mechanization.

For this, it is desirable to have a method that is parametric in the choice of the sets of truth values, connectives and quantifiers. Such a method can be instantiated with a particular choice and (possibly after optimization) would yield a mechanization for that logic system. In [Car87, Car91] Carnielli presents a tableau method, a resolution calculus is given by Baaz and Fermüller [BF92]. A sorted version of these methods [KK94, KK95] can be 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. However a first-order language cannot be adequate for mathematics, since quantification over functions or predicates is widespread. For instance the natural numbers cannot be fully characterized in first-order logic (the induction axiom quantifies over sets).

This has been the initial motivation to investigate the multi-valued framework for higher-order logic. In [Sch94] the second author presents an n–valued higher–order logic and gives a sound and complete resolution calculus by combining results of Huet [Hue72] on higher-order resolution with the first-order framework of multi-valued deduction mentioned above. Our system is further refined in this paper to a logic $\mathcal{HOL}^n$ and a resolution calculus $\mathcal{HR}^n$ using techniques and results from [Koh94]. Thus the proofs omitted in this paper can be easily adapted from those in these sources.

The resulting framework can be combined with the sort techniques developed in [Koh94] to obtain a higher-order formalization of mathematics in the spirit of [KK94]. Naturally the results reported here are much more widely applicable, they extend to all logical systems that combine multiple truth values with higher-order features, such as λ-binding and $\beta\eta$-conversion. Even if the target logic does not contain higher-order features, the added expressivity of $\mathcal{HOL}^n$ admits simple and efficient relativizations (especially for first-order target logics, such as modal logics). Thus $\mathcal{HR}^n$ 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.

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 $\mathcal{BT} \subseteq \mathcal{T}$ 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 $\gamma := \beta_1 \to \ldots \to \beta_n \to \alpha$ 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 **collection of variables**, i.e. 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 by

1. $\Sigma_\alpha \subseteq \mathit{wff}_\alpha(\Sigma)$
2. If $X \in \mathcal{V}_\alpha$, then $X \in \mathit{wff}_\alpha(\Sigma)$.
3. If $\mathbf{A} \in \mathit{wff}_{\beta\to\alpha}(\Sigma)$ and $\mathbf{B} \in \mathit{wff}_\beta(\Sigma)$, then $\mathbf{AB} \in \mathit{wff}_\alpha(\Sigma)$.
4. 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 $\mathcal{J} = \bigcup_{k\in\mathbf{N}} \mathcal{J}^k \subseteq \Sigma$ of **connectives** and $\mathcal{Q} = \bigcup_{\alpha\in\mathcal{T}} \mathcal{Q}^\alpha \subseteq \Sigma$ of **quantifiers**. Here $\mathcal{J}^k$ is the set of k-ary connectives of type $\overline{o_k} \to o$ 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 $Q(\lambda X_\alpha\mathbf{.A})$, 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).

The case of classical higher-order logic [Chu40] can be recovered as $\mathcal{HOL}^2$, where we have $\mathcal{J} = \{\neg_{o\to o}, \vee_{o\to o\to o}\}$ and $\mathcal{Q} = \{\Pi^\alpha \big| \alpha \in \mathcal{T}\}$ where $\forall X_\alpha\mathbf{.A}$ is an abbreviation for $\Pi^\alpha(\lambda X_\alpha\mathbf{.A})$.

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** $(\mathbf{A} \to_\lambda \mathbf{B})$, if it is obtained by applying one of the following rules to a well-formed part of $\mathbf{A}$.

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

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

As usual we denote the transitive closure of a reduction relation $\to_\lambda$ 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 $\mathbf{A} \to_\lambda^* \mathbf{A}{\downarrow}$ such that $\mathbf{A}{\downarrow}$ is a **λ-normal form**.

3 Semantics

For the semantics we first define the algebraic structure of model structures which will serve as a model for the underlying 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. Model structures are built up from a **carrier set** $\mathcal{D}_\mathcal{T}$, i.e. a collection $\mathcal{D}_\mathcal{T} = \{\mathcal{D}_\alpha \big| \alpha \in \mathcal{T}\}$ of sets, such that $\mathcal{D}_{\alpha\to\beta} \subseteq \mathcal{F}(\mathcal{D}_\alpha; \mathcal{D}_\beta)$, and a (well-typed) **interpretation of constants** $\mathcal{I}\colon \Sigma \longrightarrow \mathcal{D}$.

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. We call a pair $\mathcal{A} := (\mathcal{D}, \mathcal{I})$ a **model structure**, iff for each assignment φ the interpretation function $\mathcal{I}$ can inductively be extended to a function $\mathcal{I}_\varphi\colon \mathit{wff}(\Sigma) \to \mathcal{D}$ 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. $\mathcal{I}_\varphi(\lambda X_\alpha.\mathbf{B}_\beta)$ is the unique function in $f \in \mathcal{D}_{\alpha\to\beta}$ such that $f(z) := \mathcal{I}_{\varphi,[z/X]}(\mathbf{B})$, else undefined.

Maybe the most prominent example of a model structure is the set $\mathit{wff}(\Sigma)\downarrow$ of well-formed formulae in $\beta\eta$-normal form, together with $\mathcal{I} = \mathrm{Id}_\Sigma$. Here we consider formulae $\mathbf{A}$ of type $\alpha \to \beta$ as functions, such that $\mathbf{A}(\mathbf{B}) = (\mathbf{AB})\downarrow$. In this setting, assignments are substitutions and $\mathcal{I}_\varphi(\mathbf{A}) = \varphi(\mathbf{A})\downarrow$.

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 [Chu40] $\mathcal{HOL}^n$ 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 $\tilde{\jmath} = \mathcal{I}(j)\colon \mathfrak{B}^k \to \mathfrak{B}$ for a connective $j \in \mathcal{J}^k$.

In first-order multi-valued logics the intended meaning of a quantifier Q is traditionally given as a function $\tilde{Q}\colon \mathbf{P}^*(\mathfrak{B}) \longrightarrow \mathfrak{B}$, where we write $\mathbf{P}(M)$ for the power set of a set M and $\mathbf{P}^*(M) := \mathbf{P}(M) \setminus \emptyset$. With this, the value of a quantified expression is computed by applying $\tilde{Q}$ to the set of truth-values of all of the instances of its scope. Note that these definitions generalize the classical case, where $\mathfrak{B} = \{\mathtt{T}, \mathtt{F}\}$ and $\mathfrak{T} = \{\mathtt{T}\}$ and for instance $\tilde{\forall}(M)$ is true, iff $M = \{\mathtt{T}\}$.

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

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

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

We will use Kleene's strong logic $\mathbf{K}^3$ [Kle52] as a running example. There we have $\mathfrak{B} = \{\mathtt{T}, \mathtt{F}, \bot\}$ and $\mathfrak{T} = \{\mathtt{T}\}$. 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, which will however have extended truth functions:

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

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

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

Now we can use $\mathbf{K}^3$ to formalize a simple mathematical fact about function division, namely a cancelation law for real functions: *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 3.1. Of the real numbers we use the constants $0, 1$ and the functions **inv** and $*$. Since we want **inv** to be partial (undefined on 0), we assume an error element $\bot_\alpha$ for all types $\alpha \in \mathcal{T}$ and all functions and predicates to be strict with respect to it (where $\alpha \neq o$ and β does not have o as an argument type):

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

We will also use the symbols **inv** and $*$ on functions, defined by

$$\mathbf{inv} = (\lambda F\lambda X.\mathbf{inv}(FX)) \tag{2}$$

$$* = (\lambda F, G\lambda X.(FX)*(GX)) \tag{3}$$

This allows us to define function division by

$$\div = (\lambda F, G.F*(\mathbf{inv}G)) \tag{4}$$

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

$$\forall X.(\mathbf{inv}X = \bot_\iota) \equiv X = 0 \tag{5}$$

$$\forall X.(X \neq 0) \implies (\mathbf{inv}X*X) = 1 \tag{6}$$

together with associativity of $*$ and the unit axiom for 1 and $*$. In the theory defined by axioms (1)–(6), the theorem stated above has the form

$$\forall F, G.(\forall X.GX \neq 0 \wedge GX \neq \bot) \implies (*(\div FG)G) = F \tag{7}$$

In fact, this axiomatization is not yet correct, since the universal quantifications include the error-element $\bot$. In order to arrive at a correct representation, we have to change all quantified expressions of the form $\forall X.\mathbf{A}$ into $\forall X.(X \neq \bot) \implies \mathbf{A}$.

Let us now turn to the definition of models.

Definition 3.2 (Henkin Model). A model structure $\mathcal{A}$ is called a **Henkin model**, iff $\mathcal{D}_o = \mathfrak{B}$, $\mathcal{I}(j) = \tilde{j}$ and $\mathcal{I}(Q)(f) = \tilde{Q}(\{f(a) \big| a \in \mathcal{D}_\alpha\})$ 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, with the notion of completeness induced by this semantics there cannot be complete calculi [Göd31], a fact that makes it virtually useless for our purposes. Unfortunately, we cannot even use Henkin models directly, since they make too strong assertions about extensionality on $\mathcal{D}_o$[1] that higher-order refutation calculi cannot yet handle. So we cannot require that $\mathcal{D}_o = \mathfrak{B}$, but we have to 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 $\mathfrak{B}$.

[1] the fact that $\mathcal{D}_o = \{\mathtt{T}, \mathtt{F}\}$ implies that equivalent propositions can be substituted for each other

Definition 3.3 (Frege Structure). Let $\mathcal{A} = (\mathcal{D}, \mathcal{I})$ be a model structure, then a surjective total function $v\colon \mathcal{D}_o \longrightarrow \mathfrak{B}$ with

1. $v(\mathcal{I}(j)[a_1, \ldots, a_n]) = \tilde{j}\,[v(a_1), \ldots, v(a_k)]$ for any $j \in \mathcal{J}^k$.
2. $v(\mathcal{I}(Q)[f]) = \tilde{Q}(\{v(f[a]) \big| a \in \mathcal{D}_\alpha\})$ for any $Q \in \mathcal{Q}^\alpha$.

is called a **valuation for** $\mathcal{A}$. In this case we call the triple $\mathcal{M} := (\mathcal{D}, \mathcal{I}, v)$ a **Frege structure**. For a given assignment φ the evaluation of a formula $\mathbf{A}$ consists of interpretation $\mathcal{I}_\varphi(\mathbf{A})$ in $\mathcal{A}$ and the subsequent valuation with v. 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 $v \circ \mathcal{I}_\varphi(\mathbf{A}) \in \mathfrak{T}$.

4 Model Existence

We now 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' model existence theorem for higher-order logic [And71] 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 $v \circ \mathcal{I}_\varphi(\mathbf{A}) = w$. 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 $\mathcal{I}_\varphi(j\overline{\mathbf{A}}) = \tilde{j}(\overline{\mathcal{I}_\varphi(\mathbf{A})})$ and thus $(\mathcal{I}_\varphi(\mathbf{A}_1), \ldots, \mathcal{I}_\varphi(\mathbf{A}_n)) \in \tilde{j}^{-1}(w)$ is the relevant condition. However it is possible to optimize this condition, if $\tilde{j}$ is constant on some argument. We formalize this in the notion of a Π-consequence, which has been introduced by Carnielli [Car87].

Definition 4.1 (Π-Consequence). Let $\mathfrak{B}^* := \mathfrak{B} \cup \{*\}$ and $\vec{v}_n = (v_1, \ldots, v_n)$ and $\vec{w}_n = (w_1, \ldots, w_n)$ be members of $\mathfrak{B}^{*n}$, then we say that $\vec{v}_n$ is **more general** than $\vec{w}_n$ ($\vec{v}_n \sqsubseteq \vec{w}_n$), if for some $\mathcal{J} \subseteq \{1, \ldots, n\}$ we have $v_k = *$ for all $k \in \mathcal{J}$ and $w_i = v_i$ for all $i \notin \mathcal{J}$. Intuitively, higher generality can be obtained by replacing some components of a vector by $*$. For a sequence $\overline{\mathbf{A}} = \mathbf{A}_1 \ldots \mathbf{A}_n$ of formulae we write $\overline{\mathbf{A}}^{\vec{v}_n}$ for the set $\{\mathbf{A}_i^{v_i} \big| v_i \neq *\}$. 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 $\tilde{j}$ to all of $\mathfrak{B}^*$ by inductively defining $\tilde{j}(v_1, \ldots, v_{i-1}, *, v_{i+1}, \ldots, v_n) = v$ whenever $\tilde{j}(v_1, \ldots, v_{i-1}, w, v_{i+1}, \ldots, v_n) = v$ for all $w \in \mathfrak{B}$ and undefined else. Then we call the set

$\mathcal{H}_v(j) := \tilde{j}^{-1}(v) \subseteq \mathfrak{B}^{*n}$ 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 $\vec{v}_n$ generates the set of all $\vec{w}_n$ with $\vec{v}_n \sqsubseteq \vec{w}_n$). Now let $\mathbf{B}^v$ be a labeled formula of the form $(j\mathbf{A}_1 \dots \mathbf{A}_n)^v$, then a set $\overline{\mathbf{A}}^{\vec{w}_n}$ is called a **Π-consequence of $\mathbf{B}^v$**, iff $\vec{w}_n \in \mu_{\sqsubseteq}\mathcal{H}_v(j)$.

Example 4.2. For the connectives of $\mathbf{K}^3$ we have the following Π-consequences:

	Π-cons.
$(\neg\mathbf{A})^{\mathtt{F}}$	$\{\mathbf{A}^{\mathtt{T}}\}$
$(\neg\mathbf{A})^{\perp}$	$\{\mathbf{A}^{\perp}\}$
$(\neg\mathbf{A})^{\mathtt{T}}$	$\{\mathbf{A}^{\mathtt{F}}\}$

	Π-consequences
$(\mathbf{A} \vee \mathbf{B})^{\mathtt{F}}$	$\{\mathbf{A}^{\mathtt{F}}\}, \{\mathbf{B}^{\mathtt{F}}\}$
$(\mathbf{A} \vee \mathbf{B})^{\perp}$	$\{\mathbf{A}^{\perp}, \mathbf{B}^{\perp}\}, \{\mathbf{A}^{\perp}, \mathbf{B}^{\mathtt{T}}\}, \{\mathbf{A}^{\mathtt{T}}, \mathbf{B}^{\perp}\}$
$(\mathbf{A} \vee \mathbf{B})^{\mathtt{T}}$	$\{\mathbf{A}^{\mathtt{T}}, \mathbf{B}^{\mathtt{T}}\}$

Note that the set of Π-consequences can be empty. Furthermore this construction is not necessarily unique for connectives of arity $k > 2$ (for a detailed discussion see [Häh92]). In order to have a unique notion of Π-consequence we fix some general method of constructing these sets in advance. 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 $\mathtt{f}_\alpha, \mathtt{f}_\beta \dots$) for each type.

Definition 4.3 (Set of witnesses). Let $M = \{w_1, \dots, w_m\} \subseteq \mathfrak{B}$ 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 $\Xi_M(\mathbf{A}) := \{(\mathbf{A}\mathtt{f}_1)^{w_1}, \dots, (\mathbf{A}\mathtt{f}_m)^{w_m}\}$ a **set of M-witnesses for A in environment** Φ.

Definition 4.4 (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 $\mathbf{B}^v = (j\mathbf{A}_1 \dots \mathbf{A}_n)^v \in \Phi$, then there is a Π-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 $\tilde{Q}(M) = v$, 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 atomic 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 just 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 (negative variables).

The significance of abstract consistency classes lies in the following theorem, which we cite from [Sch94].

Theorem 4.5 (Existence of Frege Structures).
Let $H \in \nabla$ and ∇ be a saturated abstract consistency class, then there is a model structure $\mathcal{M}$ with $\mathcal{M} \models H$.

Proof sketch: The set H can be extended to higher-order Hintikka set,i.e. a maximal set $\mathcal{H} \in \nabla$, such that $H \subseteq \mathcal{H}$. For this we can build a Frege structure $(\mathcal{D}, \mathcal{I}, v)$ 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. Since $\mathcal{H} \in \nabla$, the function $v\colon \mathcal{D}_o \to \mathfrak{B}$ defined by $v(\mathbf{A}_o) = w$, iff $\mathbf{A}^w \in \mathcal{H}$ is a valuation. Note that with this construction we can only obtain Frege structures, not Henkin models, since the set of closed formulae of sort o is different from $\mathfrak{B}$. □

5 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.

- Higher-order unification is undecidable, therefore we cannot simply use it as a subprocedure 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. 5.1).

- 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.

- Naive Skolemization is not sound for higher-order logic. In this paper we will use Dale Miller's solution to this problem: His idea is to introduce arities for the witness constants (we call the resulting pair $\mathtt{f}^k_\alpha$, 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 $\mathtt{f}^k_\alpha$ 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. For lack of space, we will not go into depth here; the reader is referred to [Mil83] for details.

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 symbols are different from the disjunction constants in the signature.

Definition 5.1 (Clause). If $\mathbf{M}_i \in wff_o(\Sigma)$ 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 $\mathcal{C}$ 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

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

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

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

where $\beta(M) = \{(\mathbf{A}(\mathtt{f}_i^k \overline{X_k}))^{w_i} \,|\, w_i \in M\} \cup \{\bigvee_{w_i \in M} (\mathbf{A}X)^{w_i}\}$, where $\mathbf{Free}(\mathbf{A}) = \{X_1, \ldots, X_k\}$. This set plays the role of the set $\Xi_M(\mathbf{A})$ of witnesses defined in 4.3. 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 $\mathbf{A} \in wff_o(\Sigma)$ 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 clause normal form of a set Φ of sentences as

$$\mathbf{CNF}(\Phi) = \bigcup_{\mathbf{A} \in \Phi} \mathbf{cnf}(\bigvee_{w \in \mathfrak{B} \setminus \mathfrak{T}} \mathbf{A}^w)$$

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 occuring quantifiers admit generic optimizations [Sch94] that yield the classical rules for $\mathcal{HOL}^2$. 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.

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

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

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

The regularity of $\mathbf{K}^3$ allows us to optimize this clause normal form even further: As first noticed by Rainer Hähnle [Häh92], clause normalization can be more efficient, if we process disjunctions $\mathbf{L}^{\{v_1,\ldots,v_n\}}$ (written as $\mathbf{L}^{v_1\ldots v_n}$) in one step. In particular for $\mathbf{K}^3$, labeled formulae containing literals $\mathbf{L}^{\mathtt{T}\perp\mathtt{F}}$ are tautologous and can be deleted and normalization rules acting on $\mathbf{A}^{\mathtt{T},\perp}$ or $\mathbf{A}^{\mathtt{F},\perp}$ (intuitively meaning that the formula $\mathbf{A}$ must not be $\mathtt{F}/\mathtt{T}$) are much more regular than the combination of the $\mathtt{T}$, and $\perp$ rules induced by the disjunctions. For instance we have the following rules for sets of signs

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

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

Let us now return to our example 3.1 to prove the theorem 7, we have to consider the clause normal form of the set of axioms (1)–(6) labeled with $\mathtt{T}$, together with (7) labeled with $\mathtt{F}\perp$ (we have to refute that it obtains the truth values $\mathtt{F}$ or $\perp$). Using the optimized reduction rules above the clause normal form of our example 3.1 has the following form:

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

where $A1$ comes from strictness (1), $A2-A3$ from (5), and $A4$ comes from (6). The theorem clauses $T1$ and $T2$ have been obtained from (7) by eliminating definitions (2) – (4) and clause normalizing.

To prove the theorem from this clause set, we have two possibilities: We can either axiomatize transitivity, reflexivity, symmetry, and substitutivity of equality or or eliminate the equality predicate via the well-known Leibniz formulation

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

We will execute neither 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).

Theorem 5.2 (Clause Normal Form Theorem).
A set Φ of sentences is valid, if $\mathbf{CNF}(\Phi)$ is unsatisfiable.

Proof sketch: The notion of satisfiability for clauses in this theorem is nearly straightforward, a clause $\mathcal{C}$ is satisfiable in $\mathcal{M} = (\mathcal{D}, \mathcal{I})$, iff there is a literal $\mathbf{L}^v$ in $\mathcal{C}$ such that $\mathcal{I}_\varphi(\mathbf{L}) = v$ for some assignment φ. However, due to the non-standard nature of Skolem constants, they may not be interpreted as normal functions in the model; but for our restricted class of Skolem formulae interpretation is unproblematic. With this and the definition of Π-consequences the proof of the assertion reduces to a standard argument about Skolemization. For details about the model construction see [Sch94]. □

5.1 Higher-Order Unification

The higher-order unification problem can be reduced to the problem of finding most general formulae of a given type and a given head symbol.

Definition 5.3 (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 $\mathbf{V}^i = H^i\overline{X^l_{\alpha_l}}$. The H^i are new variables of types $\overline{\beta_l} \to \delta^i$. 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** $\mathcal{A}^h_\alpha(\Sigma) := \{\mathcal{G}^h_\alpha(\Sigma)\} \cup \{\mathcal{G}^j_\alpha(\Sigma) \big| j \leq l\}$.

5.2 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 $\bigvee_{v\in\mathfrak{B}\setminus\mathfrak{T}} \mathbf{A}^v$ 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 5.4 (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:

$$\frac{\mathbf{N}^v \vee \mathbf{C} \quad \mathbf{M}^w \vee \mathbf{D} \quad v \neq w}{\mathbf{C} \vee \mathbf{D} \vee \mathbf{M} \neq^? \mathbf{N}} \mathcal{HR}(Res)$$

$$\frac{\mathbf{M}^v \vee \mathbf{N}^v \vee \mathbf{C}}{\mathbf{M}^v \vee \mathbf{C} \vee \mathbf{M} \neq^? \mathbf{N}} \mathcal{HR}(Fac)$$

which operate on the clause part of clauses. For manipulating the unification constraints $\mathcal{HR}^n$ utilizes pre-unification rules (cf. [Koh94]) of which we will only state the most interesting one:

$$\frac{\mathbf{C} \vee F_\alpha \overline{\mathbf{U}} \neq^? h\overline{\mathbf{V}}}{\mathbf{C} \vee F \neq^? \mathbf{G} \vee F\overline{\mathbf{U}} \neq^? h\overline{\mathbf{V}}} \mathcal{HR}(flex - rigid)$$

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

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

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

$$\frac{\mathbf{C} \vee \mathcal{E} \vee X \neq^? \mathbf{A}}{\mathcal{C}} \mathcal{HR}(Solv)$$

where $X \neq^? \mathbf{A}$ is solved in $\mathcal{E} \vee X \neq^? \mathbf{A}$ and $\mathcal{C} \in \mathbf{CNF}([\mathbf{A}/X]\mathbf{C} \vee [\mathbf{A}/X]\mathcal{E})$. 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). Since these clauses play the role of the empty clause in our resolution calculus we denote them collectively by $\Box$. Clearly any empty clause $\Box$ is unsatisfiable with respect to Frege structures, since the constraint is solvable. We will call a set Ψ of generalized clauses **refutable**, iff $\Box$ is derivable from it and a set Φ of sentences **provable**, iff $\mathbf{CNF}(\Phi)$ 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 terminal clause has been derived.

Theorem 5.5 (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 clause normal form theorem 5.2 since the resolution and factoring rules residuate the appropriate unification constraint. □

Lemma 5.6. *Let Φ be a set of generalized clauses, θ a substitution $\mathcal{D}$ a refutation of $\theta(\Phi)$. Then there is a derivation $\mathcal{D}'$: $\Phi \vdash_{\mathcal{HR}} \mathcal{E}$, 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 $\mathcal{HR}(Prim)$ rule that introduces the appropriate constant. Since the $\mathcal{HR}(Prim)$ rule contains an application of $\mathcal{HR}(Solv)$, 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 $\mathcal{HR}(Solv)$ 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 $\mathcal{HR}(Fac)$ 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. □

Theorem 5.7 (Completeness). *$\mathcal{HR}^n$ is complete with respect to Frege structures.*

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 4.5 we see that non-refutable sets of generalized clauses are satisfiable in the class of Frege structures. 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 4.4.

$\nabla 1$ 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.

$\nabla 2$ 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 Π-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 $\Phi \cup \mathbf{C}^1 \otimes \cdots \otimes \mathbf{C}^k$ 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 concluded the proof by remarking that $\mathbf{C}^1 \otimes \cdots \otimes \mathbf{C}^k$ is just the clause normal form of $(j\overline{\mathbf{A}})^w$.

$\nabla 4$ Let $(Q\overline{\mathbf{A}})^v \in \Phi$. We have to show that the existence of a family of refutations $\mathcal{D}^w_M$ of $\Phi \cup \Xi_M(\mathbf{A}) \cup \{(\mathbf{AB})^w\}$, where $M \in \tilde{Q}^{-1}(v)$ 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 $\mathbf{L}^i$ comes from some $\beta(M^i)$. From refutations $\mathcal{D}^w_M$, $w \in M$, 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 clause normal form as Φ. Thus $\mathcal{D}$ is indeed the refutation needed to complete the proof.

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

$$\beta(M) = \{(\mathbf{A}(\mathfrak{f}^k_i\overline{X_k}))^{w_i} \big| w_i \in M\} \cup \{\bigvee_{w_i \in M} (\mathbf{A}X)^{w_i}\}$$

For each $\mathcal{D}^w_M$, $w \in M$, the lifting lemma (cf. 5.6 take $\theta = [\mathbf{B}/X]$) guarantees a derivation $\mathcal{F}^w_M \colon \Phi \cup \Xi_M(\mathbf{A}) \cup \{(\mathbf{A}X)^w\} \vdash_{\mathcal{HR}} \mathcal{C}$, 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. □

6 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 structures. 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. Even if the target logic does not contain higher-order features, the added expressivity of $\mathcal{HOL}^n$ admits simple and efficient relativizations (especially for first-order target logics, such as modal logics). Thus $\mathcal{HR}^n$ makes mechanization of the target logic much simpler than the first-order multi-valued frameworks.

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) it will be necessary to combine it with the sort techniques of [Koh94] in the spirit of [KK94]. This will yield a suitable basis for formalizing and mechanizing informal mathematical vernacular.

References

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

[BF92] Matthias Baaz and Christian G. Fermüller. Resolution for many-valued logics. In 1992, editor, *Proceedings of the International Conference on Logic Programming and Automated Reasoning LPAR'92*, volume 624 of *LNAI*, pages 107–118. Springer Verlag, St. Petersburg, Russia, 1992.

[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.

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

[Göd31] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte der Mathematischen Physik*, 38:173–198, 1931. English Version in [vH67].

[Häh92] Rainer Hähnle. *Tableaux-Based Theorem Proving in Multiple-Valued Logics*. PhD thesis, Universität Karlsruhe, Germany, 1992.

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

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

[KK94] Manfred Kerber and Michael Kohlhase. A mechanization of strong Kleene logic for partial functions. In Alan Bundy, editor, *Proceedings of the 12th Conference on Automated Deduction*, LNAI, pages 371–385, Nancy, France, 1994.

[KK95] Manfred Kerber and Michael Kohlhase. A tableau calculus for partial functions. In *Annals of the Kurt-Gödel-Society.* Springer Verlag, 1995.

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

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

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

[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.

[vH67] Jean van Heijenoort, editor. *From Frege to Göel A Soruce Book in Mathematical Logic, 1879-1931.* Source Books in the History of the Sciences. Harvard University Press, 1967.