A Mechanization of Strong Kleene Logic for Partial Functions*

Manfred Kerber Michael Kohlhase

Fachbereich Informatik, Universität des Saarlandes
66041 Saarbrücken, Germany
+49-681-302-{4628|4627}
{kerber|kohlhase}@cs.uni-sb.de

Abstract. Even though it is not very often admitted, partial functions do play a significant role in many practical applications of deduction systems. Kleene has already given a semantic account of partial functions using three-valued logic decades ago, but there has not been a satisfactory mechanization. Recent years have seen a thorough investigation of the framework of many-valued truth-functional logics. However, strong Kleene logic, where quantification is restricted and therefore not truth-functional, does not fit the framework directly. We solve this problem by applying recent methods from sorted logics. This paper presents a resolution calculus that combines the proper treatment of partial functions with the efficiency of sorted calculi.

1 Introduction

Many practical applications of deduction systems in mathematics and computer science rely on the proper treatment of partial functions. Although there are workarounds for most concrete situations, there has been a considerable interest in the community for clean formalizations of partial functions.

One of the key problems to be solved when formalizing partial functions is to decide what happens if partial functions are applied to arguments not in their domain. In mathematical practice expressions like $\frac{0}{0} = 1$ or $odd(predecessor(0))$ are thought to be neither true nor false. This phenomenon can be handled in the well-known systems for intuitionistic logic, where the law of the excluded middle does not hold, hence $\frac{0}{0} = 1$ can be (and in fact is) neither true nor false, since neither the truth nor the falsehood of this expression can be shown. However, most mathematicians do not want to give up the law of the excluded middle, because it is basic for a strong proof technique, the indirect proof. The standard way to deal with this situation in classical mathematics is to reject expressions like $\frac{0}{0}$ as "meaningless". This phenomenon of "truth value gaps" is studied in various systems of free logic [22, 10, 15]. A related approach that seems more available for mechanization has first been advocated by Kleene in [14]. He introduces an additional individual $\perp$ denoting meaningless individuals and a third truth value u, standing for the "undefined" truth value. At first glance this seems to be a great deviation from mathematical practice, which only acknowledges two truth

* This work was supported by the Deutsche Forschungsgemeinschaft (SFB 314)

values, but the third truth value simply labels situations that would be rejected in mathematical practice anyway.

In recent years, methods for the operationalization of many-valued logics have been developed by Carnielli [6], Hähnle [12], Baaz and Fermüller [2]. All of these logics have in common that they are *truth-functional*, that is, composed formulae obtain their truth values from their components and (for quantifiers) from *all* instances of the scope. Therefore a direct utilization of these methods is impossible for Kleene logic, since his quantifiers only range over defined values, that is, not over $\bot$. Kleene's approach has been utilized by Tichy [21], Lucio-Carrasco and Gavilanes-Franco [16] to give logical systems for partial functions. Both approaches offer unsorted operationalizations of the systems in sequent calculi by providing special subcalculi for reasoning about definedness.

Other authors (cf. [5, 8, 19, 24]) have avoided the problems that accompany treating a third truth value, and simply consider all atomic expressions containing a meaningless term as false. This has the advantage that partial functions can be handled within the classical two-valued framework. However, the serious drawback is that the results of these logic systems can be unintuitive to the working mathematician. For instance it is mathematical consensus that the following equation should only hold provided that y is not 0:

$$\forall x_{\mathbb{R}}, y_{\mathbb{R}}, z_{\mathbb{R}}.\ z = \frac{x}{y} \Rightarrow x = y * z$$

However, in the abovementioned systems this is a theorem, since for $y = 0$ the atom $z = \frac{x}{0}$ obtains the truth value f which in turn makes the implication true. We formalize Kleene's ideas for partial functions in an order-sorted three-valued logic, called $\mathcal{SKL}$, that uses the Kleene's strong interpretation of connectives and quantifiers and adapts techniques from Weidenbach's logic [24] to handle definedness information. It will turn out (cf. example 39) that the formula above is not a theorem in our formalization, since the case $y = 0$ is a counterexample.

2 Strong Order-Sorted Kleene Logic ($\mathcal{SKL}$)

In [14] Kleene presents a logic, which he calls *strong three-valued logic* for reasoning about partial recursive predicates on the set of natural numbers. He argues that the intuitive meaning of the third truth value should be "undefined" or "unknown" and introduces the truth tables shown in definition 26. Similarly Kleene enlarges the universe of discourse by an element $\bot$ denoting the undefined number. In his exposition the quantifiers only range over natural numbers, in particular he does not quantify over the undefined individual (number).

The approach of this paper is to make Kleene's meta-level discussion of defined and undefined individuals explicit by structuring the universe of discourse with the sort $\mathfrak{D}$ for all defined individuals. Furthermore all functions and predicates are strict, that is, if one of the arguments of a compound term or an atom evaluates to $\bot$, then the term evaluates to $\bot$ or the truth value of the atom is u. Just as in Kleene's system, our quantifiers only range over individuals in $\mathfrak{D}$, that is, individuals that are not undefined. This is in contrast to the well-understood framework for truth-functional many-valued logics, where the concept of definedness and defined quantification cannot be easily introduced, since quantification

is truth-functional and depends on the truth values for all (even the undefined) instantiations of the scope. Kleene's concept of bounded quantification is essential for our program of representing partial functions, since in a truth-functional approach no proper universally quantified expression can evaluate to the truth value t (dually for the existential quantifier), since all functions and predicates are assumed strict.

In the following we present the logic system $\mathcal{SKL}$, which is a sorted version of what we believe to be a faithful formalization of Kleene's ideas from [14]. We treat the sorted version here, since we need the machinery for dynamic sorts in the calculus to be able to treat the sort $\mathfrak{D}$ (sort techniques as that from [24, 25] give us the bounded quantification). We will call formulations of $\mathcal{SKL}$ where $\mathfrak{D}$ is the only sort in the signature *strong unsorted Kleene logic*. The further use of sorts gives the well-known advantages of sorted logics for the conciseness of representation and reduction of search spaces.

Syntax

Definition 21 (Signature) A *signature* $\Sigma := (\mathcal{S}, \mathcal{V}, \mathcal{F}, \mathcal{P})$ consists of the following disjoint sets

- $\mathcal{S}$ is a finite set of *sorts* including the sort $\mathfrak{D}$. We define $\mathcal{S}^* := \mathcal{S} \setminus \{\mathfrak{D}\}$
- $\mathcal{V}$ is a set of *variable symbols*. Each variable x is associated with a unique sort S, which we write in the index, i.e. x_S. We assume that for each sort $S \in \mathcal{S}$ there is a countably infinite supply of variables of sort S in $\mathcal{V}$.
- $\mathcal{F}$ is a set of *function symbols*.
- $\mathcal{P}$ is the set of *predicate symbols*.

The sets $\mathcal{F}$ and $\mathcal{P}$ are subdivided into the sets $\mathcal{F}^k$ of *function symbols of arity k* and $\mathcal{P}^k$ of *predicate symbols of arity k*. Note that individual constants are just nullary functions. We call a signature *unsorted* if $\mathcal{S}^*$ is empty, that is, if $\mathfrak{D}$ is the only sort.

Definition 22 (Terms and Formulae) We define the set of *terms* to be the set of variables together with *compound terms* $f(t^1, \ldots, t^k)$ for terms $t^1, \ldots, t^k$ and $f \in \mathcal{F}^k$.

If $P \in \mathcal{P}^k$, then $P(t^1, \ldots, t^k)$ is a *proper atom*. If t is a term and S a sort then $t \leqslant S$ is a *sort atom*. The set of *formulae* contains all atoms and with formulae A and B the formulae $A \wedge B$, $A \vee B$, $A \Rightarrow B$, $\neg A$, $!A$, $\forall x_S.\, A$ and $\exists x_S.\, A$. Here the intended meaning of $!A$ is that A is defined.

Semantics

In this section we will define the three valued semantics for $\mathcal{SKL}$ by postulating an "undefined individual" $\perp$ in the universe of discourse. Note that this is similar to the classical flat CPO construction [20], but Kleene's interpretation of truth values does not make u minimal. Since we are not interested in least fix-points, monotonicity does not play a role in this paper.

Definition 23 (Strict Σ-Algebra) Let Σ be a signature, then a pair $(\mathcal{A}, \mathcal{I})$ is called a *strict Σ-algebra*, iff

1. the *carrier set* $\mathcal{A}$ is a set of at least two elements that contains $\perp$,

2. the *interpretation function* $\mathcal{I}$ obeys the following restrictions:
   (a) For all function symbols f, the function $\mathcal{I}(f)\colon \mathcal{A}^k \longrightarrow \mathcal{A}$ is strict for $\bot$, that is, $\mathcal{I}(f)(a_1, \ldots, a_k) = \bot$, if $a_i = \bot$ for (at least) one i.
   (b) If P is a predicate symbol, then the relation $\mathcal{I}(P) \subseteq \mathcal{A}^k$ is strict for $\bot$, that is, $\mathcal{I}(P)(a_1, \ldots, a_k) = \mathsf{u}$, if $a_i = \bot$ for (at least) one i.
   (c) If $S \neq \mathfrak{D}$ is a sort, then $\mathcal{I}(S)$ is a total, unary, and strict relation, that is, $\mathcal{I}(S)(a) \in \{\mathsf{f}, \mathsf{t}\}$, if $a \neq \bot$ and $\mathcal{I}(S)(\bot) = \mathsf{u}$.
   (d) $\mathcal{I}(\mathfrak{D})(\bot) = \mathsf{f}$ and $\mathcal{I}(\mathfrak{D})(a) = \mathsf{t}$, if $a \neq \bot$. Note that in contrast to all other sorts and predicates, the denotation of $\mathfrak{D}$ is not a strict relation.

We define the *carrier* $\mathcal{A}_S$ *of sort* S as $\mathcal{A}_S := \{a \in \mathcal{A} \mid \mathcal{I}(S)(a) = \mathsf{t}\}$. Note that in contrast to other sorted logics, it is not assumed that the $\mathcal{A}_S$ are non-empty. This fact will require special treatments in the transformation to clause normal form and for instantiations in the resolution calculus. Furthermore $\bot \notin \mathcal{A}_S$ for any $S \in \mathcal{S}$.

By systematically deleting $\bot$ and u from the carrier and the truth values we can canonically transform strict Σ-algebras into algebras of partial functions. These are an algebraic account of the standard interpretation in mathematics, where partiality of functions is directly modeled by right-unique relations. Obviously these notions of algebras have a one-to-one correspondence, so both approaches are equivalent.

Definition 24 (Σ-assignment) Let $(\mathcal{A}, \mathcal{I})$ be a strict Σ-algebra, then we call a total mapping $\varphi\colon \mathcal{V} \longrightarrow \mathcal{A}$ a *Σ-assignment*, iff $\varphi(x_S) \in \mathcal{A}_S$, provided $\mathcal{A}_S$ is non-empty and $\varphi(x_S) = \bot$ if $\mathcal{A}_S = \emptyset$. We denote the Σ-assignment that coincides with φ away from x and maps x to a with $\varphi, [a/x]$.

Definition 25 Let φ be a Σ-assignment into a strict Σ-algebra $(\mathcal{A}, \mathcal{I})$ then we define the *value function* $\mathcal{I}_\varphi$ *from formulae to* $\mathcal{A}$ inductively to be

1. $\mathcal{I}_\varphi(f) := \mathcal{I}(f)$, if f is a function or a predicate.
2. $\mathcal{I}_\varphi(x) := \varphi(x)$, if x is a variable.
3. $\mathcal{I}_\varphi(f(t^1, \ldots, t^k)) := \mathcal{I}(f)(\mathcal{I}_\varphi(t^1), \ldots, \mathcal{I}_\varphi(t^k))$, if f is a function or predicate.
4. $\mathcal{I}_\varphi(t{\Leftarrow}S) = \mathcal{I}(S)(\mathcal{I}_\varphi(t))$.

Note that this definition applies to $\mathcal{P}$ and $\mathcal{F}$ alike, thus we have given the semantics of all atomic formulae. The semantic status of sorts is that of total unary predicates; in particular in $\mathcal{A}$ we have $\mathcal{I}_\varphi(t{\Leftarrow}S) = \mathsf{u}$, iff $\mathcal{I}_\varphi(t) = \bot$.

Definition 26 The value of a formula dominated by a connective is obtained from the value(s) of the subformula(e) in a truth-functional way. Therefore it suffices to define the truth tables for the connectives:

$\wedge$	f	u	t
f	f	f	f
u	f	u	u
t	f	u	t

$\vee$	f	u	t
f	f	u	t
u	u	u	t
t	t	t	t

$\Rightarrow$	f	u	t
f	t	t	t
u	u	u	t
t	f	u	t

$\neg$	
f	t
u	u
t	f

!	
f	t
u	f
t	t

Kleene does not use the ! operator as a connective but treats it on the meta-level. Note while it is useful it is not necessary for the treatment. Furthermore,

even this connective does not render $\mathcal{SKL}$ truth-functionally complete, since, just like negation and conjunction, ! is normal, that is, when restricted to $\{\mathsf{f},\mathsf{t}\}$ all connectives yield values in $\{\mathsf{f},\mathsf{t}\}$.

The semantics of the quantifiers is defined with the help of function $\widetilde{\forall}$ and $\widetilde{\exists}$ from the non-empty subsets of the truth values in the truth values. We define

$$\mathcal{I}_\varphi(\mathsf{Q}x_S.\ A) := \widetilde{\mathsf{Q}}(\{\mathcal{I}_{\varphi,[a/x]}(A) \mid a \in \mathcal{A}_S\})$$

where $\mathsf{Q} \in \{\forall, \exists\}$ and furthermore

$$\widetilde{\forall}(T) := \begin{cases} \mathsf{t} & \text{for } T = \{\mathsf{t}\} \\ \mathsf{u} & \text{for } T = \{\mathsf{t},\mathsf{u}\} \text{ or } \{\mathsf{u}\} \\ \mathsf{f} & \text{for } \mathsf{f} \in T \end{cases} \qquad \text{and} \qquad \widetilde{\exists}(T) := \begin{cases} \mathsf{t} & \text{for } \mathsf{t} \in T \\ \mathsf{u} & \text{for } T = \{\mathsf{f},\mathsf{u}\} \text{ or } \{\mathsf{u}\} \\ \mathsf{f} & \text{for } T = \{\mathsf{f}\} \end{cases}$$

Note that with this definition quantification is separated into a truth-functional part $\widetilde{\forall}$ and an instantiation part that only considers members of $\mathcal{A}_S$. Since $\bot$ is not a member of any $\mathcal{A}_S$, quantification never considers it and therefore cannot be truth-functional even for the unsorted case.

For lack of space we will in the following often only treat the (sufficient) subset $\{\wedge, \neg, !, \forall\}$ of logical symbols, since all others can be defined from these.

Definition 27 (Σ-Model) Let A be a formula, then we call a strict Σ-algebra $\mathcal{M} := (\mathcal{A}, \mathcal{I})$ a *Σ-model for A* (written $\mathcal{M} \models A$), iff $\mathcal{I}_\varphi(A) = \mathsf{t}$ for all Σ-assignments φ. With this notion we can define the notions of *validity*, *(un)-satisfiability*, and *entailment* in the usual way.

Remark 28 The "tertium non datur" principle of classical logic is no longer valid, since formulae can be undefined, in which case they are neither true nor false. We do however have a "quartum non datur" principle, that is, formulae are either true, false, or undefined, which allows us to derive the validity of a formula by refuting that it is false or undefined. We will use this observation in our resolution calculus.

Extended Example

We will formalize an extended example from elementary algebra that shows the basic features of $\mathcal{SKL}$. Here the sort $\mathbb{R}^*$ denotes the real numbers without zero. Note that we use the sort information to encode definedness information for inversion: $\frac{1}{x}$ is defined for all $x \in \mathbb{R}^*$, since $\mathbb{R}^*$ is subsort of $\mathfrak{D}$ by definition. Naturally, we give only a reduced formalization of real number arithmetic that is sufficient for our example. (For instance, we could add expressions like $\frac{1}{0} \not< \mathfrak{D}$.) Consider the formula $A := (\mathrm{A1} \wedge \mathrm{A2} \wedge \mathrm{A3} \wedge \mathrm{A4} \wedge \mathrm{A5}) \Rightarrow \mathrm{T}$ with

$$\begin{array}{ll} \mathrm{A1} & \forall x_{\mathbb{R}}\ x \neq 0 \Rightarrow x < \mathbb{R}^* \\ \mathrm{A2} & \forall x_{\mathbb{R}^*}\ \frac{1}{x} < \mathbb{R}^* \\ \mathrm{A3} & \forall x_{\mathbb{R}^*}\ x^2 > 0 \end{array} \qquad \begin{array}{ll} \mathrm{A4} & \forall x_{\mathbb{R}}\ \forall y_{\mathbb{R}}\ x - y < \mathbb{R} \\ \mathrm{A5} & \forall x_{\mathbb{R}}\ \forall y_{\mathbb{R}}\ x - y = 0 \Rightarrow x = y \\ \mathrm{T} & \forall x_{\mathbb{R}}\ \forall y_{\mathbb{R}}\ x \neq y \Rightarrow \left(\frac{1}{x-y}\right)^2 > 0 \end{array}$$

An informal mathematical argumentation why T is entailed by $\mathrm{A1} \wedge \ldots \wedge \mathrm{A5}$ can be as follows:

Let x and y be arbitrary elements of $\mathbb{R}$. If $x = y$, the premise of T is false, hence the whole expression true (in this case the conclusion evaluates to u). If

$x \neq y$, then the premise is true and the truth value of the whole expression is equal to that of the conclusion $\left(\frac{1}{x-y}\right)^2 > 0$. Since $x \neq y$ we get by A5 that $x-y \neq 0$ and by A4 that $x-y \leqslant \mathbb{R}$, hence by A1 $x-y \leqslant \mathbb{R}^*$ and by A2 $\frac{1}{x-y} \leqslant \mathbb{R}^*$, which finally gives $\left(\frac{1}{x-y}\right)^2 > 0$ together with A3.

However, if we analyze the justification of this argumentation, we see that there is a hidden assumption, namely the totality of the binary predicate $>$ on $\mathbb{R} \times \mathbb{R}$. In fact the formula A is not a tautology, since it is possible to interpret the $>$ predicate as undefined for the second argument being zero, so that A3 as well as T evaluate to u, while the other Ai evaluate to t, hence the whole expression evaluates to u. There are two solutions of this problem, namely adding further formulae Ai, in which the definiteness of the predicates are specified, or – what is normally done in mathematics – to start with a formula where the Ai are assumed to be true, that is, neither false nor undefined. We will discuss the alternatives in remark 310 and modify the example accordingly.

Relativization into Truth-Functional Logic

In this section we show that we can always systematically transform $\mathcal{SKL}$ formulae to formulae in an unsorted truth-functional three-valued logic $\mathbf{K}^3$ in a way that respects the semantics. However, we will see that this formulation will lose much of the conciseness of the presentation and enlarge the search spaces involved with automatic theorem proving.

At first glance it may seem that $\mathcal{SKL}$ is only an order-sorted variant of a three-valued instance of the truth-functional many-valued logics that were very thoroughly investigated by Carnielli, Hähnle, Baaz and Fermüller [2, 6, 12]. However, since all instances of this framework are truth-functional, even unsorted $\mathcal{SKL}$ does not fit into this paradigm, since quantification excludes the undefined element. In $\mathcal{SKL}$ quantification is bounded by sorts, which are all subsorts of the sort $\mathfrak{D}$ of defined objects. Therefore relativization not only considers sort information, it also has to care about definedness aspects in quantification.

Informally, $\mathbf{K}^3$-formulae are just first-order formulae (with the additional unary connective !). While the three-valued semantics of the connectives is just that given in definition 26, the semantics of quantifications uses unrestricted instantiation:

$$\mathcal{I}_\varphi(\forall x.\ A) := \widetilde{\forall}(\{\mathcal{I}_{\varphi,[a/x]}(A) \mid a \in \mathcal{A}\})$$

Definition 29 We define transformations $\mathbf{Rel}^{\mathcal{S}}$ and $\mathbf{Rel}^{\mathfrak{D}}$, that map $\mathcal{SKL}$-sentences to unsorted $\mathcal{SKL}$-sentences and further to $\mathbf{K}^3$-sentences. $\mathbf{Rel}^{\mathcal{S}}$ is the identity on terms and atoms, homomorphic on connectives and

$$\mathbf{Rel}^{\mathcal{S}}(\forall x_S.\ \Phi) := \forall x_{\mathfrak{D}}.\ S(x) \Rightarrow \mathbf{Rel}^{\mathcal{S}}(\Phi)$$

Note that in order for these sentences to make sense in unsorted $\mathcal{SKL}$ we have to extend the set of predicate symbols by unary predicates S for all sorts $S \in \mathcal{S}^*$. Furthermore, for any of these new predicates we need the axiom: $\forall x_{\mathfrak{D}}.\ !S(x)$. The set of all these axioms is denoted by $\mathbf{Rel}^{\mathcal{S}}(\Sigma)$.

We define $\mathbf{Rel}^{\mathfrak{D}}$ to be the identity (only dropping the sort references from the variables) on terms and proper atoms and

$$\mathbf{Rel}^{\mathfrak{D}}(t{\leqslant}\mathfrak{D}) := \mathfrak{D}(t) \qquad \text{and} \qquad \mathbf{Rel}^{\mathfrak{D}}(\forall x_{\mathfrak{D}}.\, A) := \forall x.\, \mathfrak{D}(x) \Rightarrow \mathbf{Rel}^{\mathfrak{D}}(A)$$

Just as above we have to extend the set of predicate symbols by a unary predicate $\mathfrak{D}$ and need a set $\mathbf{Rel}^{\mathfrak{D}}(\Sigma)$ of signature axioms, which contains the axioms

$$\forall x_1, \ldots, x_n.\, P^n(x_1, \ldots, x_n) \vee \neg P^n(x_1, \ldots, x_n) \Rightarrow (\mathfrak{D}(x_1) \wedge \ldots \wedge \mathfrak{D}(x_n))$$
$$\forall x_1, \ldots, x_n.\, \mathfrak{D}(f(x_1, \ldots, x_n)) \Rightarrow (\mathfrak{D}(x_1) \wedge \ldots \wedge \mathfrak{D}(x_n))$$

for any predicate symbol $P \in \mathcal{P}^n$, such that $P \neq \mathfrak{D}$ and for any function symbol $f \in \mathcal{F}^n$, together with the axioms

$$\forall x.\, !\mathfrak{D}(x)^1 \qquad \text{and} \qquad \exists x.\, \mathfrak{D}(x)$$

These axioms axiomatize the $\mathcal{SKL}$ notion of definedness in $\mathbf{K}^3$. In particular the last axioms state that the predicate $\mathfrak{D}$ is two-valued and non-empty, in contrast to all other sort predicates which are strict, thus three-valued, and may be empty. The other axioms force all functions and predicates to be interpreted strictly with respect to the $\mathfrak{D}$ predicate. Note that in the case of nullary function symbols the signature axioms have the form $\mathfrak{D}(c^0)$, which state that individual constants are defined.

Theorem 210 (Sort Theorem) *Let Φ be a set of sentences, then the following statements are equivalent*

1. *Φ has a Σ-model.*
2. *$\mathbf{Rel}^{\mathcal{S}}(\Phi)$ has a $\Sigma \cup \mathcal{S}^*$-model that satisfies $\mathbf{Rel}^{\mathcal{S}}(\Sigma)$.*
3. *$\mathbf{Rel}^{\mathfrak{D}} \circ \mathbf{Rel}^{\mathcal{S}}(\Phi)$ has a $\mathbf{K}^3$-model that satisfies $\mathbf{Rel}^{\mathfrak{D}}(\Sigma \cup \mathcal{S}^*) \cup \mathbf{Rel}^{\mathfrak{D}}(\mathbf{Rel}^{\mathcal{S}}(\Sigma))$.*

Proof sketch: Using standard sort techniques we can use the signature axioms of Σ to identify sorted models in the class of unsorted models, for details see [13]. □

As a consequence of the sort theorem, the standard operationalization for many-valued logics [2, 6, 12] can be utilized to mechanize strong order-sorted Kleene logic and in fact the system of Lucio-Carrasco and Gavilanes-Franco [16] can be seen as a standard many-valued tableau operationalization [12, 3] of the relativization of $\mathcal{SKL}$. However, as the extended example shows, we can do better by using sorted methods, since relativization expands the size and number of input formulae and furthermore expands the search spaces involved in automatic theorem proving by building up many meaningless branches. Note that already the formulation of unsorted $\mathcal{SKL}$ where we only have the required sort $\mathfrak{D}$ is more concise than the relativized version and, as we will see, the theory of definedness is treated by the sorts in the $\mathcal{RPF}$ calculus (cf. section 3). Thus the $\mathcal{RPF}$ calculus is closer to informal practice than the relativization in this respect.

[1] Since this is an axiom (see remark 310), we could also have used $\forall x.\, \mathfrak{D}(x) \vee \neg\mathfrak{D}(x)$ here.

Extended Example (continued)

The relativization $\mathbf{Rel}^{\mathcal{S}}(\mathbf{Rel}^{\mathfrak{D}}(A))$ of the formula A in the extended example is the $\mathbf{K}^3$-formula $(\mathrm{R1} \wedge \mathrm{R2} \wedge \mathrm{R3} \wedge \mathrm{R4} \wedge \mathrm{R5}) \Rightarrow \mathrm{RT}$.

$$
\begin{array}{ll}
\mathrm{R1} & \forall x.\, \mathfrak{D}(x) \Rightarrow (\mathbb{R}(x) \Rightarrow (x \neq 0 \Rightarrow \mathbb{R}^*(x))) \\
\mathrm{R2} & \forall x.\, \mathfrak{D}(x) \Rightarrow (\mathbb{R}^*(x) \Rightarrow \mathbb{R}^*(\tfrac{1}{x})) \\
\mathrm{R3} & \forall x.\, \mathfrak{D}(x) \Rightarrow (\mathbb{R}^*(x) \Rightarrow x^2 > 0) \\
\mathrm{R4} & \forall x.\, \mathfrak{D}(x) \Rightarrow (\mathbb{R}(x) \Rightarrow (\forall y.\, \mathfrak{D}(y) \wedge \mathbb{R}(y) \Rightarrow \mathbb{R}(x - y))) \\
\mathrm{R5} & \forall x.\, \mathfrak{D}(x) \Rightarrow (\mathbb{R}(x) \Rightarrow (\forall y.\, \mathfrak{D}(y) \Rightarrow (\mathbb{R}(y) \Rightarrow (x - y = 0 \Rightarrow x = y)))) \\
\mathrm{RT} & \forall x.\, \mathfrak{D}(x) \Rightarrow (\mathbb{R}(x) \Rightarrow (\forall y.\, \mathfrak{D}(y) \Rightarrow (\mathbb{R}(y) \Rightarrow (x \neq 0 \Rightarrow \left(\tfrac{1}{x}\right)^2 > 0))))
\end{array}
$$

The set of signature axioms $\mathbf{Rel}^{\mathfrak{D}}(\Sigma \cup \mathcal{S}^*) \cup \mathbf{Rel}^{\mathfrak{D}}(\mathbf{Rel}^{\mathcal{S}}(\Sigma))$ is the following set of $\mathbf{K}^3$-formulae:

$$
\begin{array}{ll}
\mathrm{R}^{=} & \forall x, y.\, (x = y \vee x \neq y) \Rightarrow \mathfrak{D}(x) \wedge \mathfrak{D}(y) \\
\mathrm{R}^{>} & \forall x, y.\, (x > y \vee x \not> y) \Rightarrow \mathfrak{D}(x) \wedge \mathfrak{D}(y) \\
\mathrm{R}^{-} & \forall x, y.\, \mathfrak{D}(x - y) \Rightarrow \mathfrak{D}(x) \wedge \mathfrak{D}(y) \\
\mathrm{R}^{/} & \forall x.\, \mathfrak{D}(\tfrac{1}{x}) \Rightarrow \mathfrak{D}(x) \\
\mathrm{R}^{0} & \mathfrak{D}(0)
\end{array}
\qquad
\begin{array}{ll}
\mathrm{R}^{2} & \forall x.\, \mathfrak{D}(x^2) \Rightarrow \mathfrak{D}(x) \\
\mathfrak{D}^{!} & \forall x.\, \mathfrak{D}(x) \vee \neg\mathfrak{D}(x) \\
\mathfrak{D}^{\emptyset} & \exists x.\, \mathfrak{D}(x) \\
\mathrm{R}^{\mathbb{R}} & \forall x.\, \mathfrak{D}(x) \Rightarrow !\mathbb{R}(x) \\
\mathrm{R}^{\mathbb{R}^*} & \forall x.\, \mathfrak{D}(x) \Rightarrow !\mathbb{R}^*(x)
\end{array}
$$

3 Resolution

In this section we present a resolution calculus $\mathcal{RPF}$ with dynamic sorts that is a generalization of Weidenbach's work [24, 25] with ideas from [2, 12]. In the literature [23, 7, 18, 11] there are various calculi for sorted logics that vary in deductive power but have in common that the sort information available to sort reasoning remains static over the length of a proof. These methods are not sufficient for our purposes, since definedness[2] cannot in general be decided by syntactic means only, but is usually given in the form of logical axioms that have to be reasoned about in the calculus itself. In contrast to these Weidenbach's logics allows the declaration of conditional sort (and thus definedness) information. When these conditions are proved in the course of the proof, additional sort information becomes available for the sort reasoning mechanism. There are two variants of this calculus (unsorted unification [24] and sorted unification [25]), we have generalized both for our purposes, but in this paper we only present the first (simpler) version due to the lack of space. We refer the reader to the full version of this paper [13] for the other variant.

Clause Normal Form

Definition 31 Let A be a formula, then we call A^α (the formula A indexed with the intended truth value $\alpha \in \{\mathsf{f}, \mathsf{u}, \mathsf{t}\}$), a *labeled formula*. We will call a labeled atom L^α a *literal* and a set of literals $\{L_1^{\alpha_1}, \ldots, L_n^{\alpha_n}\}$ a *clause*. We say that a Σ-model $\mathcal{M}$ *satisfies* a clause C, iff it satisfies one of its literals $L^\alpha \in C$, that is,

[2] Cohn's Boolean lattice of sorts [7] has $\bot$ elements for ill-sorted terms and formulae. While the notation is similar to ours, this concept should not be confused with undefinedness. In contrast to Kleene's interpretation, all of Cohn's connectives are strict and no expression containing ill-sorted elements can be a tautology, making his calculus and his notion of ill-sortedness inappropriate for the treatment of undefinedness.

$\mathcal{I}_\varphi(L^\alpha) = \alpha$. $\mathcal{M}$ satisfies a set of clauses iff it satisfies each clause. In order to conserve space, we employ the "," as the operator for the disjoint union of sets, so that C, L^α means $C \cup \{L^\alpha\}$ and L^α is not a member of C. Furthermore we adopt Hähnle's notion of multi-labels in the form $C, A^{\alpha\beta}$ to mean C, A^α, A^β.

Definition 32 (Transformations to Clause Normal Form)

$$\frac{C, (A \wedge B)^{\mathsf{t}}}{C, A^{\mathsf{t}} \quad C, B^{\mathsf{t}}} \qquad \frac{C, (A \wedge B)^{\mathsf{u}}}{C, A^{\mathsf{ut}} \quad C, B^{\mathsf{ut}} \quad C, A^{\mathsf{u}}, B^{\mathsf{u}}} \qquad \frac{C, (A \wedge B)^{\mathsf{f}}}{C, A^{\mathsf{f}}, B^{\mathsf{f}}}$$

$$\frac{C, (\neg A)^{\mathsf{t}}}{C, A^{\mathsf{f}}} \qquad \frac{C, (\neg A)^{\mathsf{u}}}{C, A^{\mathsf{u}}} \qquad \frac{C, (\neg A)^{\mathsf{f}}}{C, A^{\mathsf{t}}}$$

$$\frac{C, (\forall x_S.\, A[x_S])^{\mathsf{t}}}{C, A[x_S]^{\mathsf{t}}} \qquad \frac{C, (\forall x_S.\, A[x_S])^{\mathsf{u}}}{C, A[f(y^1, \ldots, y^n)]^{\mathsf{u}} \quad C, A[x_S]^{\mathsf{ut}} \quad C, (f(y^1, \ldots, y^n) \leqslant S)^{\mathsf{t}}}$$

$$\frac{C, (\forall x_S.\, A[x_S])^{\mathsf{f}}}{C, A[f(y^1, \ldots, y^n)]^{\mathsf{f}} \quad C, (f(y^1, \ldots, y^n) \leqslant S)^{\mathsf{t}}}$$

$$\frac{C, (!A)^{\mathsf{t}}}{C, A^{\mathsf{tf}}} \qquad \frac{C, (!A)^{\mathsf{u}}}{C} \qquad \frac{C, (!A)^{\mathsf{f}}}{C, A^{\mathsf{u}}} \qquad \frac{C, (t \leqslant \mathfrak{D})^{\mathsf{u}}}{C} \qquad \frac{C, (t \leqslant S)^{\mathsf{u}}}{C, (t \leqslant \mathfrak{D})^{\mathsf{f}}}$$

where $\{x_S, y^1, \ldots, y^n\} = \mathbf{Free}(A)$ and f is a new function symbol of arity n. Here $\mathbf{Free}(A)$ denotes the set of free variables of A.

Note that this set of transformations is confluent, therefore any total reduction of a set Φ of labelled sentences results in a unique set of clauses. We will denote this set with $\mathbf{CNF}(\Phi)$.

General Assumption 33 The clause normal form transformations as presented above are not complete, that is, they do not transform every given labelled formula into clause form, since the rules for quantified formulae insist that the bound variable occurs in the scope. In fact the handling of degenerate quantifications poses some problems in the presence of possibly empty sorts, as quantification over empty sets are vacuously true. In this situation we have three possibilities, either to forbid degenerate quantifications, or empty sorts, or treat degenerate quantifications in the clause normal form transformations. For this paper we chose the first, since degenerate quantifications do not make much sense mathematically and do not appear in informal mathematics. See [13] for the other possibilities. Thus we will asssume that in all formulae in this paper the bound variables of quantifications occur in the scopes.

As usual the reduction to clause normal form conserves satisfiability.

Theorem 34 *Let Φ be a set of labelled sentences, then the clause normal form $\mathbf{CNF}(\Phi)$ is satisfiable, iff Φ is.*

Resolution Calculus ($\mathcal{RPF}$)

Now we proceed to give a simple resolution calculus, which utilizes unsorted unification. However, despite its name the calculus still utilizes the sort information present in the clause set and therefore gives considerably improved search behavior over unsorted methods as in [16]. In [13], we have further improved the calculus by using a sorted unification algorithm, which delegates parts of the search into the unification algorithm. For unsorted substitutions a naive resolution rule is unsound. Therefore we have to add a residual (the sort constraint) that ensures the well-sortedness of the unifier.

Definition 35 (Sort Constraints) Let $\sigma = [t^1/x^1_{S_1}], \ldots, [t^n/x^n_{S_n}]$ be a substitution, then we define the *sort constraint for* σ to be the clause

$$\mathscr{C}(\sigma) := \{(t^1 \leqslant S_1)^{\mathsf{fu}}, \ldots, (t^n \leqslant S_n)^{\mathsf{fu}}\}$$

Definition 36 (Resolution Inference Rules ($\mathcal{RPF}$))

$$\frac{L^\alpha, C \qquad M^\beta, D}{\sigma(C), \sigma(D), \mathscr{C}(\sigma)}\, Res \qquad \frac{L^\alpha, M^\alpha, C}{\sigma(L^\alpha), \sigma(C), \mathscr{C}(\sigma)}\, Fac$$

$$\frac{(t \leqslant \mathfrak{D})^{\mathsf{f}}, C \qquad L^\gamma, D}{\rho(C), \rho(D), \mathscr{C}(\rho)}\, Strict$$

where $\alpha \neq \beta$ and $\gamma \in \{\mathsf{t}, \mathsf{f}\}$. For *Res* and *Fac* the substitution σ is the most general (unsorted) unifier of L and M and for *Strict* there exists a subterm s of L, such that ρ is a most general unifier of t and s. Here we have assumed α, β and γ to be single truth values, naturally the rules can be easily extended to sets of truth values.

Remark 37 Note that clauses containing A^{fut} are tautologous and can therefore be deleted in the generation of the clause normal form as well as in the deduction process. The calculus can be extended by the usual subsumption rule, allowing to delete clauses that are subsumed (super-sets).

Definition 38 Let A be a sentence and Φ be the clause normal form of the set $\{\{A^{\mathsf{f}}\}, \{A^{\mathsf{u}}\}\}$ then we say that A can be *proven in* $\mathcal{RPF}$ ($\vdash A$), iff there is a derivation of the empty clause $\square$ from Φ with the inference rules above.

Example 39 Now we can come back to the example from the exposition. The assertion is not a theorem of $\mathcal{SKL}$, since the clause normal form of the instance $\{\{(1 = \frac{1}{0} \Rightarrow 1 = 0 * 1)^{\mathsf{f}}\}, \{(1 = \frac{1}{0} \Rightarrow 1 = 0 * 1)^{\mathsf{u}}\}\}$, namely:

$$(1 = \tfrac{1}{0})^{\mathsf{u}}, (1 = \tfrac{1}{0})^{\mathsf{t}}$$
$$(1 = 0 * 1)^{\mathsf{u}}, (1 = 0 * 1)^{\mathsf{f}}$$

is satisfiable. In fact in any reasonable formalization of elementary algebra $1 = \frac{1}{0}$ is undefined, whereas $1 = 0 * 1$ is false. Thus, since $\mathcal{RPF}$ is sound (cf. 311), the example cannot be a theorem.

Remark 310 In practical applications most problems will be of the form $A := (A_1 \wedge \ldots \wedge A_n \Rightarrow C)$ where the A_i are the assumptions and C is the intended conclusion. In contrast to classical first-order predicate logic where it suffices to take the clause normal form of $\{\{A_1^{\mathsf{t}}\}, \ldots, \{A_n^{\mathsf{t}}\}, \{C^{\mathsf{f}}\}\}$ the situation here is more complex, since in $\mathcal{SKL}$ we also have to refute the case that A gets the value u. It is however easy to see, that we can start the calculation of the clause normal form with the set

$$\{\{A_1^{\mathsf{ut}}\}, \ldots, \{A_n^{\mathsf{ut}}\}, \{C^{\mathsf{fu}}\}\} \qquad \text{or with the sets}$$

$$\{\{A_1^{\mathsf{ut}}\}, \ldots, \{A_n^{\mathsf{ut}}\}, \{A_1^{\mathsf{u}}, \ldots, A_n^{\mathsf{u}}\}, \{C^{\mathsf{u}}\}\} \qquad (*)$$
$$\{\{A_1^{\mathsf{t}}\}, \ldots, \{A_n^{\mathsf{t}}\}, \{C^{\mathsf{fu}}\}\} \qquad (**)$$

which have to be refuted by the resolution calculus independently. In the second case the refutation can be split in two independent proofs, thus reducing the search space considerably. Nevertheless, the refutation of the set $(*)$ is impractical except for trivial examples. Fortunately in mathematical practice the assumptions A_i often have the status of axioms, which are assumed to be true independently of the theorem[3]. Then the problem is really of the form

$$A' := (A_1 \wedge !A_1 \wedge \ldots \wedge A_n \wedge !A_n \Rightarrow C)$$

The clause normal form of A' is just that of $(**)$, which is close to the classical case in derivational complexity. In particular the background theory formalized by the A_i results in exactly the same clauses as in the classical case.

Extended Example (continued)

Following the discussion above we will continue our extended example with the calculation of the clause normal form $(**)$ of A1$\wedge$!A1 $\wedge \ldots \wedge$ A5$\wedge$!A5 $\Rightarrow$ T. Since $\mathbb{R}$ and $\mathbb{R}^*$ are not empty, we can use slightly simplified quantification rules in the clause normal form transformations (see [13]).

A1 $(x_{\mathbb{R}} = 0)^{\mathsf{t}}, (x_{\mathbb{R}} \lessdot \mathbb{R}^*)^{\mathsf{t}}$
A2 $(\frac{1}{x_{\mathbb{R}^*}} \lessdot \mathbb{R}^*)^{\mathsf{t}}$
A3 $(x^2_{\mathbb{R}^*} > 0)^{\mathsf{t}}$
A4 $(x_{\mathbb{R}} - y_{\mathbb{R}} \lessdot \mathbb{R})^{\mathsf{t}}$
A5 $(x_{\mathbb{R}} - y_{\mathbb{R}} = 0)^{\mathsf{f}}, (x_{\mathbb{R}} = y_{\mathbb{R}})^{\mathsf{t}}$

The price for the formal treatment of three-valued partiality has to be paid in the complicated clause normal form of the formula T with the label fu.

T1 $(c \lessdot \mathbb{R})^{\mathsf{t}}$
T2 $(d \lessdot \mathbb{R})^{\mathsf{t}}$
T3 $(e \lessdot \mathbb{R})^{\mathsf{t}}$
T4 $(f \lessdot \mathbb{R})^{\mathsf{t}}$
T5 $(g(y_{\mathbb{R}}) \lessdot \mathbb{R})^{\mathsf{t}}$
T6 $(c = d)^{\mathsf{f}}, (e = f)^{\mathsf{fu}}$
T7 $(c = d)^{\mathsf{f}}, \left(\left(\frac{1}{e-f}\right)^2 > 0\right)^{\mathsf{fu}}$
T8 $\left(\left(\frac{1}{c-d}\right)^2 > 0\right)^{\mathsf{f}}, (e = f)^{\mathsf{fu}}$
T9 $\left(\left(\frac{1}{c-d}\right)^2 > 0\right)^{\mathsf{f}}, \left(\left(\frac{1}{e-f}\right)^2 > 0\right)^{\mathsf{fu}}$

Eight further clauses resulting from the theorem are not shown here, four are tautologies, four others not needed for the derivation below.

[3] This is also the very idea of the set of support strategy in resolution theorem proving.

T6 & A5→R1 $(c-d=0)^{\mathsf{f}}, (e=f)^{\mathsf{fu}}, (c{\leqslant}\mathbb{R})^{\mathsf{fu}}, (d{\leqslant}\mathbb{R})^{\mathsf{fu}}$

R1 & A1→R2 $(c-d{\leqslant}\mathbb{R}^*)^{\mathsf{t}}, (e=f)^{\mathsf{fu}}, (c-d{\leqslant}\mathbb{R})^{\mathsf{fu}}, (c{\leqslant}\mathbb{R})^{\mathsf{fu}}, (d{\leqslant}\mathbb{R})^{\mathsf{fu}}$

R2 & A4→R3 $(c-d{\leqslant}\mathbb{R}^*)^{\mathsf{t}}, (e=f)^{\mathsf{fu}}, (c{\leqslant}\mathbb{R})^{\mathsf{fu}}, (d{\leqslant}\mathbb{R})^{\mathsf{fu}}$

R3 & T1→R4 $(c-d{\leqslant}\mathbb{R}^*)^{\mathsf{t}}, (e=f)^{\mathsf{fu}}, (d{\leqslant}\mathbb{R})^{\mathsf{fu}}$

R4 & T2→R5 $(c-d{\leqslant}\mathbb{R}^*)^{\mathsf{t}}, (e=f)^{\mathsf{fu}}$

T8 & A3→R6 $(e=f)^{\mathsf{fu}}, \left(\left(\frac{1}{c-d}\right){\leqslant}\mathbb{R}^*\right)^{\mathsf{fu}}$

R5 & A2→R7 $(e=f)^{\mathsf{fu}}, (c-d{\leqslant}\mathbb{R}^*)^{\mathsf{fu}}$

R7 & R5→R8 $(e=f)^{\mathsf{fu}}$

Analogously, clause T7 can be reduced with T9 to R16.

… & … →R16 $\left(\left(\frac{1}{e-f}\right)^2>0\right)^{\mathsf{fu}}$

R16 & A3 →R17 $\left(\frac{1}{e-f}{\leqslant}\mathbb{R}^*\right)^{\mathsf{fu}}$

R17 & A2 →R18 $(e-f{\leqslant}\mathbb{R}^*)^{\mathsf{fu}}$

R18 & A1 →R19 $(e-f=0)^{\mathsf{t}}, (e-f{\leqslant}\mathbb{R})^{\mathsf{fu}}$

R19 & A4 →R20 $(e-f=0)^{\mathsf{t}}, (e{\leqslant}\mathbb{R})^{\mathsf{fu}}, (f{\leqslant}\mathbb{R})^{\mathsf{fu}}$

R20 & A5 →R21 $(e=f)^{\mathsf{t}}, (e{\leqslant}\mathbb{R})^{\mathsf{fu}}, (f{\leqslant}\mathbb{R})^{\mathsf{fu}}$

R21 & T3 →R22 $(e=f)^{\mathsf{t}}, (f{\leqslant}\mathbb{R})^{\mathsf{fu}}$

R22 & T4 →R23 $(e=f)^{\mathsf{t}}$

R8 & R23→R24 $\square$

Soundness and Completeness

Theorem 311 (Soundness) *$\mathcal{RPF}$ is sound.*

Proof sketch: The soundness of the resolution and factoring rules is established in the usual way taking into account that the sort constraints make the substitutions "well-sorted" and thus compatible with the semantics: The sort constraints add two sort literals $(t{\leqslant}S)^{\mathsf{f}}, (t{\leqslant}S)^{\mathsf{u}}$ per component of the substitution, which only can be refuted if indeed $(t{\leqslant}S)^{\mathsf{t}}$.

The *Strict* rule is sound, because functions and predicates in $\mathcal{SKL}$ are strict and thus undefined subterms of a literal make the literal undefined. □

Definition 312 Let $C \coloneqq \{L_1^{\alpha_1}, \ldots, L_n^{\alpha_n}\}$ be a clause, then the *conditional instantiation $\sigma{\downarrow}(C)$ of σ to C* is defined by

$$\sigma{\downarrow}(C) \coloneqq \{\sigma(L_1^{\alpha_1}), \ldots, \sigma(L_n^{\alpha_n})\} \cup \mathscr{C}(\sigma|_{\mathbf{Free}(C)})$$

The following result from [24] is independent of the number of truth values.

Lemma 313 *Conditional instantiation is sound: for any clause C, substitution σ and Σ-model $\mathcal{M}$ we have that $\mathcal{M} \models \sigma{\downarrow}(C)$, whenever $\mathcal{M} \models C$.*

Definition 314 Let A be a sentence and $\mathbf{CNF}(A)$ be the clause normal form of A, then we define the *Herbrand set of clauses* $\mathbf{CNF}_H(A)$ *for* A as
$\mathbf{CNF}_H(A) \coloneqq \{\sigma{\downarrow}(C) \mid C \in \mathbf{CNF}(A), \sigma \text{ ground substitution}, \mathbf{Dom}(\sigma) = \mathbf{Free}(C)\}$

Definition 315 We will call two literals L^α and L^β *complementary*, if $\alpha \neq \beta$ and literals L^γ and $(t{\leqslant}\mathfrak{D})^{\mathsf{f}}$ *$\perp$-complementary*, if t is a subterm of L and $\gamma \in \{\mathsf{t}, \mathsf{f}\}$.

Definition 316 (Herbrand Model) Let Φ be a set of clauses, then the *Herbrand base $\mathcal{H}(\Phi)$ of Φ* is defined to be the set of all ground atoms containing only function symbols that appear in the clauses of Φ. If there is no individual constant in Φ, we add a new constant c. A *valuation* ν is a function $\mathcal{H}(\Phi) \longrightarrow \{\mathsf{f}, \mathsf{u}, \mathsf{t}\}$, such that for all atoms $L, M \in \mathcal{H}(\Phi)$ the literals $L^{\nu(L)}$ and $M^{\nu(M)}$ are not $\perp$-complementary. Note that these literals are not complementary since ν is a function. The *Σ-Herbrand model $\mathcal{H}$ for Φ and ν* is the set $\mathcal{H} \coloneqq \{L^\alpha \mid \alpha = \nu(L), L \in \mathcal{H}(\Phi)\}$.

We say that a Σ-Herbrand model $\mathcal{H}$ *satisfies a clause set Φ* iff for all ground substitutions σ and clauses $C \in \Phi$ we have $\sigma{\downarrow}(C) \cap \mathcal{H} \neq \emptyset$. A clause set is called *Σ-Herbrand-unsatisfiable* iff there is no Σ-Herbrand-model for Φ.

Theorem 317 (Herbrand Theorem) *Let A be a formula, then the clause normal form $\mathbf{CNF}(A)$ has a Σ-model iff $\mathbf{CNF}_H(A)$ has a Σ-Herbrand-model.*

Proof: Let $\mathcal{M} = (\mathcal{A}, \mathcal{I})$ be a Σ-model for $\Phi \coloneqq \mathbf{CNF}(A)$. The set

$$\mathcal{H} \coloneqq \{L^\alpha \mid L \in \mathcal{H}(\Phi), \alpha = \mathcal{I}_\varphi(L)\}$$

is a Σ-Herbrand model for $\Psi \coloneqq \mathbf{CNF}_H(A)$ if φ is an arbitrary Σ-assignment, since obviously $\mathcal{I}_\varphi$ is a valuation. To show that indeed $\mathcal{H}$ is a Σ-Herbrand model for Ψ, we assume the opposite, that is, there is a clause $C \in \Psi$, such that

$\mathcal{H} \cap C = \emptyset$. Since $C \in \Psi$ there is a substitution $\sigma = [t^i/x^i_{S_i}]$ and a clause $D \in \Phi$, such that $C = \sigma\downarrow(D) = \sigma(D) \cup \mathcal{SC}(\sigma)$.

Without loss of generality we can assume that $\mathcal{I}(S_i)(\mathcal{I}_\varphi(t^i)) = \mathsf{t}$, since otherwise $\mathcal{I}_\varphi(t^i \leqslant S^i) \in \{\mathsf{f}, \mathsf{u}\}$, and therefore $(t^i \leqslant S^i)^\gamma \in \mathcal{H}$ for $\gamma \in \{\mathsf{f}, \mathsf{u}\}$, which contradicts the assumption. Thus the mapping $\psi := \varphi, [\mathcal{I}_\varphi(t^i)/x^i]$ is a Σ-assignment.

Note that since $\mathcal{M}$ is a model of Φ, we have that $\mathcal{M} \models D$ and therefore there is a literal $L^\alpha \in D$, such that $\alpha = \mathcal{I}_\psi(L) = \mathcal{I}_\varphi(\sigma(L))$, hence $\sigma(L) \in \mathcal{H}$, which contradicts the assumption.

For the converse direction let $\mathcal{H}$ be a Σ-Herbrand model for Ψ. To construct a Σ-model $\mathcal{M}$ for Φ we first construct an algebra of partial functions $(\mathcal{A}, \mathcal{I})$ (and then pass to the corresponding strict Σ-algebra). Let

$$\mathcal{A} := \{t \mid \exists L^\alpha \in \mathcal{H} \text{ where } \alpha \in \{\mathsf{f}, \mathsf{t}\} \text{ and } t \text{ subterm of } L\}$$

and let $\mathcal{I}(S)$, $\mathcal{I}(f^n)$ and $\mathcal{I}(P^n)$ be partial functions, such that

$$\begin{aligned} \mathcal{I}(S)(t) = \mathsf{t} \text{ iff } (t \leqslant S)^\mathsf{t} \in \mathcal{H} \\ \mathcal{I}(f^n)(t^1, \ldots, t^n) := f^n(t^1, \ldots, t^n) \text{ iff } f^n(t^1, \ldots, t^n) \in \mathcal{A} \\ \mathcal{I}(P^n)(t^1, \ldots, t^n) := \alpha \text{ iff } (P^n(t^1, \ldots, t^n))^\alpha \in \mathcal{H} \end{aligned}$$

Now let $\mathcal{M}$ be the strict Σ-algebra corresponding to the partial Σ-algebra $(\mathcal{A}, \mathcal{I})$. We proceed by convincing ourselves that $\mathcal{M} \models \Phi$. Let $C \in \Phi$ and $\varphi := [t^i/x^i_{S_i}]$ be an arbitrary Σ-assignment. Since $\mathcal{A}$ is a set of ground terms φ is also a ground substitution and moreover $(t^i \leqslant S_i)^\mathsf{t} \in \mathcal{H}$ by construction of $\mathcal{I}$.

$\mathcal{H}$ is a Σ-Herbrand model for Ψ and thus $\varphi\downarrow(C) \cap \mathcal{H} = (\varphi(C) \cup \mathcal{SC}(\varphi)) \cap \mathcal{H} \neq \emptyset$. Because $\mathcal{H}$ cannot contain complementary literals we must already have a literal $\varphi(L^\alpha) \in \varphi(C) \cap \mathcal{H}$. Now let ν be the valuation associated with $\mathcal{H}$. Since $\varphi(L^\alpha) \in \mathcal{H}$ we have $\alpha = \nu(\varphi(L)) = \mathcal{I}_\varphi(L)$, which implies $\mathcal{M} \models_\varphi L^\alpha$. We have taken C and φ arbitrary, so we get the assertion. □

Corollary 318 *A set Φ of ground unit clauses is unsatisfiable iff it contains two complementary or $\bot$-complementary literals.*

Theorem 319 (Ground Completeness) *Let Φ be an unsatisfiable set of ground clauses, then there exists a $\mathcal{RPF}$ derivation of the empty clause from Φ.*

Proof: The proof is analogous to the standard k-parameter proof of Anderson and Bledsoe [1]. We show be induction on $k := \sum_{C \in \Phi}(\text{card}(C) - 1)$ that there exists a refutation for Φ.

If $k = 0$ then Φ is a set of ground unit clauses. Therefore by Corollary 318 and the assumed unsatisfiability there has to be a pair of complementary or $\bot$-complementary literals in Φ. Thus a single application of the rule *Res* or *Strict* yields the empty clause.

If $k > 0$, then there is a non-unit clause $C =: C_1 \cup C_2 \in \Phi$. If $\Phi = \Phi' \cup \{C\}$ then the k parameters for $\Phi_1 := \Phi' \cup \{C_1\}$ and $\Phi_2 := \Phi' \cup \{C_2\}$ are smaller than k and therefore by inductive hypothesis there are refutations for Φ_1 and Φ_2 which can be combined to a refutation for Φ, since Φ is ground. □

Theorem 320 (Completeness) *$\mathcal{RPF}$ is complete.*

Proof sketch: For the proof of this assertion we combine the completeness result from the ground case with a lifting argument. It turns out that the lifting property can be established by methods from [24], since they are independent of the number of truth values. □

4 Conclusion

We have developed an order sorted three-valued logic for the formalization of informal mathematical reasoning with partial functions. This system formalizes and generalizes the system proposed by Kleene in [14] for the treatment of partial functions over natural numbers to general first-order logic. In fact we believe that the unsorted version of our system without the ! operator is a faithful formalization of Kleene's ideas. Furthermore we have presented a sound and complete resolution calculus with dynamic sorts for our system, which uses the sort mechanism to capture the fact that in Kleene's logic quantification only ranges over defined individuals.

Our calculus can be seen as an extension of classical logic that combines methods from many-valued logics (cf. [2, 12]) for a correct treatment of the undefined and order-sorted logics (see [24, 25]) for an adequate treatment of the defined. It differs from the sequent calculus in [16] in that the use of dynamic sort techniques greatly simplifies the calculus. In particular in the version with sorted unification as described in [13] most definedness preconditions can be taken care of in the unification. Thus we believe that our system is not only more faithful to Kleene's ideas (definedness inference is handled in the unification at a level below the calculus) but also more efficient for the sort techniques involved.

Of course further extensions of the system described here have to be considered in order to be feasible for practical mathematics. In particular this calculus does not address the question of the efficient mechanization of equality, here paramodulation (cf. [17]) or even superposition ([4]) methods would be interesting to study. However, we believe that this endeavor will mainly involve the development of the sort aspects for these calculi, because we think that the aspects of three-valuedness will not be critical.

On the other hand, the mechanization of higher-order features is essential for the formalization of mathematical practice. Higher-order logics are especially suitable for formalizing partial functions, since functions are first class objects of the systems, that can even be quantified over. In this direction the work of Farmer et al. [8, 9] has shown that partial functions are a very natural and powerful tool for formalizing mathematics. We expect that our three-valued approach, which remedies some problems of their simpler two-valued approach (see the discussion in the introduction and in example 39) can be generalized in much the same manner and will be a useful tool for formalizing mathematics.

Acknowledgments: We would like to thank Christian Fermüller and Ortwin Scheja for stimulating discussions and the anonymous referees for useful comments.

References

1. R. Anderson and W.W. Bledsoe. A linear format for resolution with merging and a new technique for establishing completeness. *Journal of the ACM*, **17**:525–534, 1970.
2. Matthias Baaz and Christian G. Fermüller. Resolution for many-valued logics. In A. Voronkov, editor, *Proceedings of International Conference on Logic Programming and Automated Reasoning*, pages 107–118, St. Petersburg, Russia, 1992. Springer Verlag. LNAI 624.
3. Matthias Baaz, Christian G. Fermüller, and Richard Zach. Dual systems of sequents and tableaux for many-valued logics. Technical Report TUW-E185.2BFZ.2-92, Technische Universität Wien, 1993. Short version in *Proceedings of the 23rd International Symposium on Multiple Valued Logic*, Sacramento, California, USA, 1993. IEEE Press.
4. Leo Bachmair and Harald Ganzinger. Non-clausal resolution and superposition with selection and redundancy criteria. In A. Voronkov, editor, *Proceedings of International Conference on Logic Programming and Automated Reasoning*, pages 273–284, St. Petersburg, Russia, 1992. Springer Verlag. LNAI 624.
5. Michael J. Beeson. *Foundations of Constructive Mathematics*. Springer Verlag, 1985.
6. Walter A. Carnielli. On sequents and tableaux for many-valued logics. *Journal of Non-Classical Logic*, **8**(1):59–76, 1991.
7. Anthony G. Cohn. A more expressive formulation of many sorted logics. *Journal of Automated Reasoning*, **3**:113–200, 1987.
8. William M. Farmer. A partial functions version of Church's simple theory of types. Technical Report M88-52, The MITRE Corporation, Bedford, Massachusetts, USA, February 1990.
9. William M. Farmer, Joshua D. Guttman, and F. Javier Thayer. IMPS: An Interactive Mathematical Proof System. *Journal of Automated Reasoning*, **11**(2):213–248, October 1993.
10. Bas C. van Fraassen. Singular terms, truth-value gaps, and free logic. *The Journal of Philosophy*, LXIII(17):481–495, 1966.
11. Alan M. Frisch. The substitutional framework for sorted deduction: Fundamental results on hybrid reasoning. *Artificial Intelligence*, **49**:161–198, 1991.
12. Reiner Hähnle. Automated Deduction in Multiple-Valued Logics, Oxford University Press, 1994.
13. Manfred Kerber and Michael Kohlhase. A mechanization of strong Kleene logic for partial functions. SEKI-Report SR-93-20 (SFB), Universität des Saarlandes, Saarbrücken, Germany, 1993.
14. Stephen Cole Kleene. *Introduction to Metamathematics*. Van Nostrand, 1952.
15. H. Leblanc and R. Thomason. Completeness theorems for some presupposition-free logics. *Fundamenta Mathematicae*, **62**:125–164, 1968.
16. Francisca Lucio-Carrrasco and Antonio Gavilanes-Franco. A first order logic for partial functions. In *Proceedings STACS'89*, pages 47–58. Springer Verlag, 1989. LNCS 349.
17. Arthur Robinson and Larry Wos. Paramodulation and TP in first order theories with equality. *Machine Intelligence*, **4**:135–150, 1969.
18. Manfred Schmidt-Schauß. *Computational Aspects of an Order-Sorted Logic with Term Declarations*. Springer Verlag, 1989. LNAI 395.
19. R. Schock. *Logics without Existence Assumptions*. Almquist & Wisell, 1968.
20. Dana S. Scott. Outline of a mathematical theory of computation. Technical Monograph PRG-2, Oxford University Computing Laboratory, November 1970.

21. Pawel Tichy. Foundations of partial type theory. *Reports on Mathematical Logic*, **14**:59–72, 1982.

22. Bertrand Russell. On denoting. *Mind (New Series)*, **14**:479–493, 1905.

23. Christoph Walther. *A Many-Sorted Calculus Based on Resolution and Paramodulation*. Research Notes in Artificial Intelligence. Pitman and Morgan Kaufmann, 1987.

24. Christoph Weidenbach. A resolution calculus with dynamic sort structures and partial functions. SEKI-Report SR-89-23, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1989. Short version in ECAI'90, p.668–693.

25. Christoph Weidenbach. A sorted logic using dynamic sorts. Technical Report MPI-I-91-218, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 1991. Short version in IJCAI'93, p.60–65.

Errata

On page 379 in Definition 38
wrong: $\{\{A^{\mathsf{f}}\},\{A^{\mathsf{u}}\}\}$
right: $\{\{A^{\mathsf{f}},A^{\mathsf{u}}\}\}$

On page 379 in Example 39
wrong: $\{\{(1=\frac{1}{0}\Rightarrow 1=0*1)^{\mathsf{f}}\},\{(1=\frac{1}{0}\Rightarrow 1=0*1)^{\mathsf{u}}\}\}$
right: $\{\{(1=\frac{1}{0}\Rightarrow 1=0*1)^{\mathsf{f}},(1=\frac{1}{0}\Rightarrow 1=0*1)^{\mathsf{u}}\}\}$