A Resolution Calculus for Presuppositions

Manfred Kerber and Michael Kohlhase

Published as: ECAI-96, Proceedings of the 12th European Conference on Artificial Intelligence, p.375–379, John Wiley & Sons

A Resolution Calculus for Presuppositions

Manfred Kerber[1] and **Michael Kohlhase**[2]

Abstract. The semantics of everyday language and the semantics of its naive translation into classical first-order language considerably differ. An important discrepancy that is addressed in this paper is about the implicit assumption what exists. For instance, in the case of universal quantification natural language uses restrictions and presupposes that these restrictions are non-empty, while in classical logic it is only assumed that the whole universe is non-empty. On the other hand, all constants mentioned in classical logic are presupposed to exist, while it makes no problems to speak about hypothetical objects in everyday language. These problems have been discussed in philosophical logic and some adequate many-valued logics were developed to model these phenomena much better than classical first-order logic can do. An adequate calculus, however, has not yet been given. Recent years have seen a thorough investigation of the framework of many-valued truth-functional logics. Unfortunately, restricted quantifications are not truth-functional, hence they do not fit the framework directly. We solve this problem by applying recent methods from sorted logics.

1 Introduction

From the first attempts of modelling everyday reasoning within the framework of classical logic, it has been known that many relevant aspects cannot be adequately expressed in classical first-order logic. The attempts to cope with these have lead to a variety of logics.

In this paper we address one of these problems, namely that of so-called *presuppositions*, where natural language allows to draw conclusions that classical logic does not warrant (for instance implicit consensus that universally quantified statements range over non-empty domains). These phenomena have been widely studied in the philosophy of language from a semantic point of view, but lack an efficient mechanisation, which is a primary concern of artificial intelligence. One of the more logic-oriented ways to cope with this phenomenon is to use a four-valued logic [3]. We take this logic as a starting point of a mechanisation by a resolution calculus.

There are two different kinds of presuppositions: the *quantificational* ones presuppose that the domain of quantifications is non-empty and the *existential ones* assume the existence of constants. In natural language, the first ones are mandatory, whereas the second kind is defeasible (it is possible to talk about non-existing entities in natural language). Surprisingly enough, the standard semantics of classical logic treats the two kinds almost the opposite way: constants always must have denotations, that is, just speaking about an object means that is must exist (for instance, speaking about a dragon, means that there is one), while quantifications are unrestricted and therefore always range over the whole (non-empty) universe. In classical logic the standard way to restrict a quantification is the use of an implication, which may, however, have an antecedent with empty domain.

A first attempt to overcome this problem is to employ three-valued Kleene logic, where a third truth value $\mathtt{undefined}$ is assumed which is given to every atomic formula containing a non-determined object like a dragon. This approach has been disputed since it does not allow hypothetical reasoning of the kind "Let us assume that all dragons can fly and that Tabaluga is a dragon, hence Tabaluga can fly." If we assume that Tabaluga does not exist, in a representation of Kleene logic the last statement $\mathtt{can\_fly}(\mathtt{Tabaluga})$ would be evaluated to $\mathtt{undefined}$ and not to true at all.

In [3] Bergmann proposes a four-valued logic to cope with presuppositions. She essentially argues that the semantical status of a formula has two independent dimensions, first a classical truth value, i.e., $\mathtt{t}$ or $\mathtt{f}$, and second a value, which tells whether the formula is secure (i.e. talks about existing objects) or not. In the case $\mathtt{can\_fly}(\mathtt{Tabaluga})$, the formula should be true but insecure.

While the presupposition that all mentioned objects exist is adequate unless the opposite is explicitly said, the quantificational presuppositions of everyday languages differ from those in classical logic. For instance, an everyday sentence like "All children of John are sleeping" presuppose that John really has children. Therefore the representation in classical first-order logic $\forall x.\ \mathtt{child\_of}(x,\mathtt{John}) \rightarrow \mathtt{sleeps}(x)$ is not adequate, since this sentence is true even when John has no children at all.

To overcome this problem Bergmann proposes a restricted quantification of the syntactic form $\forall x_{\mathtt{child\_of}(x,\mathtt{John})}.\ \mathtt{sleeps}(x)$, where the semantics of the quantifier is defined such that for a true and secure universally quantified statement the restriction expression is assumed to be non-empty.

Our mechanisation is based on the work of Carnielli [2], Hähnle [5], Baaz and Fermüller [1], who have developed methods for the operationalisation of many-valued first-order logics. However all of these approaches 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 utilisation of these methods is impossible for Bergmann's logic, since the quantifiers range only over a restricted domain.

2 Logic

The main feature of Bergmann's logic for presuppositions [3] is a two-dimensional set of truth values, where the classical two are replaced by four truth-values which are represented by pairs, where the first component consists of the values true and false, and the second of the values secure and insecure. In the following we denote these truth values by $\mathtt{t}^+$, $\mathtt{f}^+$, $\mathtt{t}^-$, and $\mathtt{f}^-$.

[1] School of Computer Science, The University of Birmingham, Birmingham B15 2TT, England, E-mail: M.Kerber@cs.bham.ac.uk

[2] Fachbereich Informatik, Universität des Saarlandes, 66041 Saarbrücken, Germany, E-mail: kohlhase@cs.uni-sb.de

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

In this paper, we further formalise Bergmann's logic and in particular present a resolution calculus for this logic. Starting from an approach like that in [6] where we have presented a mechanisation of three-valued Kleene logic, the main problem of this work is to give a proper treatment of restricted quantification and their presuppositions. The range of the quantifiers is restricted and assumed to be non-empty.

In the following we present the logic system $\mathcal{PL}$, which is a variant of Bergmann's ideas from [3]. The treatment of the restriction part of a quantification is very similar to the sort techniques developed in [8].

Definition 2.1 (Signature) A *signature* $\Sigma := (\mathcal{V}, \mathcal{F}, \mathcal{P})$ consists of the following disjoint sets: $\mathcal{V}$ is a countably infinite set of *variable symbols*, $\mathcal{F}$ is a set of *function symbols*, and $\mathcal{P}$ is a set of *predicate symbols* that contains a special predicate $\mathcal{D}$, called security predicate. 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.

Definition 2.2 (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 $f \in \mathcal{F}^k$. The set of *formulae* consists of *atoms* ($P(t^1, \ldots, t^k)$, where $P \in \mathcal{P}$) and of *compound formulae* $A \wedge B$, $A \vee B$, $A \to B$, $\neg A$, $!A$, $\mathbf{T}A$, $\forall x_S.\, A$, and $\exists x_S.\, A$, where A, B, and S are formulae.

The intended meaning of the *restricted quantification* $\forall x_S.\, A$ is that A holds for the set of all x for which S holds, and that furthermore this set is nonempty. The meaning of $!A$ is that A is secure, and that of $\mathbf{T}A$ is that A holds, but may be insecure.

Note that the concept of restricted quantification is a generalisation of sorted logics, where variables are restricted by so-called sorts, i.e. unary predicates: For any unary predicate $P \in \mathcal{P}$ the restricted quantification $\forall x_{Px}.\, A$ is equivalent to the sorted quantification $\forall x_P.\, A$ as it can be found in sorted logics.

For an intuitive treatment of presuppositions for terms (corresponding to questions whether Pegasus exists, whether it is a horse, or about the nature and existence of it's left front hoof) we use a set of so-called term declarations.

Definition 2.3 (Term Declarations) Let A be a formula, then we call A^α (the formula A indexed with the intended truth value $\alpha \in \{\mathtt{t}^+, \mathtt{f}^+, \mathtt{t}^-, \mathtt{f}^-\}$), a *labelled formula*. The set $\mathcal{TD}$ of *term declarations* is a set of labelled formulae.

We now will define the four-valued, two-dimensional semantics for $\mathcal{PL}$ by decorating the truth value of a formula with a "security value". Thus the set of truth values contains $\mathtt{t}^+$ and $\mathtt{f}^+$ for secure truth and falsity and $\mathtt{t}^-, \mathtt{f}^-$ for the insecure ones.

Definition 2.4 (Σ-Algebra) Let Σ be a signature, then a pair $(\mathcal{A}, \mathcal{I})$ is called a Σ-*algebra* with *carrier set* $\mathcal{A}$, iff the *interpretation function* $\mathcal{I}$ maps $\mathcal{F}$ and $\mathcal{P}$ to functions and predicates of the appropriate arity over $\mathcal{A}$. The only restriction we pose is that $\mathcal{I}(\mathcal{D}) \subset \{\mathtt{t}^+, \mathtt{f}^+\}$.

We call elements $a \in \mathcal{A}$ *secure*, if $\mathcal{I}(\mathcal{D})(a) = \mathtt{t}^+$, else *insecure*, and we subdivide $\mathcal{A}$ into subsets $\mathcal{A}^+$ of secure and $\mathcal{A}^-$ of insecure elements. Our definition of semantics entails that $\mathcal{A} = \mathcal{A}^+ \cup \mathcal{A}^-$ and $\mathcal{A}^+ \cap \mathcal{A}^- = \emptyset$.

Note that our treatment of undefined elements differs from the Kleene approach taken in [6], where all undefined elements are identified, since here we want to be able to reason about properties of undefined objects instead of only stating undefinedness.

Definition 2.5 (Σ-assignment) Let $(\mathcal{A}, \mathcal{I})$ be a Σ-algebra, then we call a total mapping $\varphi \colon \mathcal{V} \longrightarrow \mathcal{A}$ a Σ-*assignment*. We denote the Σ-assignment that coincides with φ away from x and maps x to a with $\varphi, [a/x]$.

Definition 2.6 Let φ be a Σ-assignment into a 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.

Note that this definition applies to $\mathcal{P}$ and $\mathcal{F}$ alike, thus we have given the semantics of all atomic formulae.

Definition 2.7 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:

$\wedge$	$\mathtt{t}^+$	$\mathtt{f}^+$	$\mathtt{t}^-$	$\mathtt{f}^-$
$\mathtt{t}^+$	$\mathtt{t}^+$	$\mathtt{f}^+$	$\mathtt{t}^-$	$\mathtt{f}^-$
$\mathtt{f}^+$	$\mathtt{f}^+$	$\mathtt{f}^+$	$\mathtt{f}^-$	$\mathtt{f}^-$
$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{f}^-$	$\mathtt{t}^-$	$\mathtt{f}^-$
$\mathtt{f}^-$	$\mathtt{f}^-$	$\mathtt{f}^-$	$\mathtt{f}^-$	$\mathtt{f}^-$

$\vee$	$\mathtt{t}^+$	$\mathtt{f}^+$	$\mathtt{t}^-$	$\mathtt{f}^-$
$\mathtt{t}^+$	$\mathtt{t}^+$	$\mathtt{t}^+$	$\mathtt{t}^-$	$\mathtt{t}^-$
$\mathtt{f}^+$	$\mathtt{t}^+$	$\mathtt{f}^+$	$\mathtt{t}^-$	$\mathtt{f}^-$
$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{t}^-$
$\mathtt{f}^-$	$\mathtt{t}^-$	$\mathtt{f}^-$	$\mathtt{t}^-$	$\mathtt{f}^-$

$\neg$	
$\mathtt{t}^+$	$\mathtt{f}^+$
$\mathtt{f}^+$	$\mathtt{t}^+$
$\mathtt{t}^-$	$\mathtt{f}^-$
$\mathtt{f}^-$	$\mathtt{f}^-$

$\to$	$\mathtt{t}^+$	$\mathtt{f}^+$	$\mathtt{t}^-$	$\mathtt{f}^-$
$\mathtt{t}^+$	$\mathtt{t}^+$	$\mathtt{f}^+$	$\mathtt{t}^-$	$\mathtt{f}^-$
$\mathtt{f}^+$	$\mathtt{t}^+$	$\mathtt{t}^+$	$\mathtt{t}^-$	$\mathtt{t}^-$
$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{f}^-$	$\mathtt{t}^-$	$\mathtt{f}^-$
$\mathtt{f}^-$	$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{t}^-$

$\mathbf{T}$	
$\mathtt{t}^+$	$\mathtt{t}^+$
$\mathtt{f}^+$	$\mathtt{f}^+$
$\mathtt{t}^-$	$\mathtt{t}^+$
$\mathtt{f}^-$	$\mathtt{f}^+$

$!$	
$\mathtt{t}^+$	$\mathtt{t}^+$
$\mathtt{f}^+$	$\mathtt{t}^+$
$\mathtt{t}^-$	$\mathtt{f}^+$
$\mathtt{f}^-$	$\mathtt{f}^+$

For formula S and each variable x (we call the pair (x, S) a *restriction*) let

$$\begin{aligned} \mathcal{A}^\pm_\varphi(S, x) &= \{a \in \mathcal{A} \mid \mathcal{I}_{\varphi,[a/x]} S \in \{\mathtt{t}^+, \mathtt{t}^-\}\} \\ \mathcal{A}^+_\varphi(S, x) &= \{a \in \mathcal{A} \mid \mathcal{I}_{\varphi,[a/x]} S = \mathtt{t}^+\} \end{aligned}$$

We call a restriction (x, S) *empty*, if $\mathcal{A}^\pm_\varphi(S, x)$ is. With this we can define the semantics of the universal quantifier by requiring $\mathcal{I}_\varphi(\forall x_S.\, A)$ to be

- $\mathtt{t}^+$, if $\mathcal{I}_{\varphi,[a/x]} A = \mathtt{t}^+$ for all $a \in \mathcal{A}^\pm_\varphi(S, x)$ and $\mathcal{A}^+_\varphi(S, x) \neq \emptyset$
- $\mathtt{f}^+$, if there is an $a \in \mathcal{A}^\pm_\varphi(S, x)$ with $\mathcal{I}_{\varphi,[a/x]} A = \mathtt{f}^+$
- $\mathtt{t}^-$, if $\mathcal{I}_{\varphi,[a/x]} A = \{\mathtt{t}^+, \mathtt{t}^-\}$ for all $a \in \mathcal{A}^\pm_\varphi(S, x)$,
  but $\mathcal{I}_{\varphi,[a/x]} A = \mathtt{t}^-$ for some $a \in \mathcal{A}^\pm_\varphi(S, x)$.
- $\mathtt{f}^-$, if there is an $a \in \mathcal{A}^\pm_\varphi(S, x)$ with $\mathcal{I}_{\varphi,[a/x]} A = \mathtt{f}^-$

Note that with this definition, the condition that $\varphi(x) \in \mathcal{A}^\pm_\varphi(S, x)$ is conserved. We call this condition **well-sortedness** of assignments. Consequently, all assignments in the construction of the semantics of a sentence are well-sorted, if we start from the empty assignment (which we can always do, since like in classical logic the value of a formula only depends on those for it's free variables). Thus we will restrict ourselves to well-sorted assignments.

With the specification of the behaviours of the connectives and quantifiers we have completed the definition of the semantics of formulae. We say that a labelled formula A^α is **satisfied** by φ, iff $\mathcal{I}_\varphi(A) = \alpha$ and **valid**, iff it is satisfied by all well-sorted assignments.

Remark 2.8 Now we can further study the relation of restricted quantification to sorted logics. Those usually define the *carrier* $\mathcal{A}_P \subseteq \mathcal{A}$ for any sort (unary predicate $P \in \mathcal{P}$) as $\mathcal{A}_P := \{a \in \mathcal{A} \mid \mathcal{I}(P)(a) = \mathtt{t}\}$ and use that to define sorted quantification as $\mathcal{I}_\varphi(\forall x_P.\, A)$ to be true, iff $\mathcal{I}_{\varphi,[a/x]}(A)$ is true for all $a \in \mathcal{A}_P$. Note that sorted logics usually assume that the $\mathcal{A}_S$ are non-empty[3] and therefore lead to the same presuppositions as $\mathcal{PL}$ on the sorted fragment.

We exploit this similarity in this paper by generalising sort techniques for the mechanisation of $\mathcal{PL}$.

Definition 2.9 (Σ-Model) Let A be a formula, then we call a Σ-algebra $\mathcal{M} := (\mathcal{A}, \mathcal{I})$ a *Σ-model for A* (written $\mathcal{M} \models A$), iff $\mathcal{I}_\varphi(A) = \mathtt{t}^+$ for all Σ-assignments φ. With this notion we can define the notions of *validity*, *(un)-satisfiability*, and *entailment* in the usual way. For a set $\mathcal{TD}$ of term declarations, we say that $\mathcal{M}$ is a $\mathcal{TD}$-Model, iff all labelled formulae in $\mathcal{TD}$ are valid (cf. 2.7).

In the following we will only consider $\mathcal{TD}$-models. From a purely theoretical point of view, term declarations do not yield more expressivity, since they can be axiomatised (any intended truth value can be characterised by combinations of the connectives ! and **T**). However, from a practical point of view, the term declarations provide a convenient means of specifying the belief about existence and sortality in the world. Furthermore, the term declarations can be used for optimisations of the calculus by sorted unification as in [7].

Remark 2.10 The "tertium non datur" principle of classical logic is no longer valid, since formulae can be insecure, in which case they are neither true nor false. We do however have a "quintum non datur" principle, that is, formulae are either true or false, but independently they can be secure or not, which allows us to derive the validity (i.e. that it is true and secure in all models) of a formula by refuting that it is false or insecure or both. We will use this observation in our resolution calculus below.

3 Resolution Calculus ($\mathcal{RPL}$)

In this section we present a resolution calculus $\mathcal{RPL}$ that is a generalisation of the resolution calculus for partial functions [6], which in turn is a joint generalisation of Weidenbach's logics with dynamic sorts [8] with ideas from [1, 5]. There are two variants of the sorted calculus, we have generalised both for our purposes, but in this paper we only present the first (simpler) version due to the lack of space.

Definition 3.1 We will call a labelled 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, $\mathcal{I}_\varphi(L^\alpha) = \alpha$. $\mathcal{M}$ 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.

Now we are in the position to give a set of transformations that take a set of labelled formulae to a refutationally equivalent set of clauses.

Definition 3.2 (Transformations to Clause Normal Form)

$$\frac{C, (A \wedge B)^{\mathtt{t}^+}}{C, A^{\mathtt{t}^+} \quad C, B^{\mathtt{t}^+}} \qquad \frac{C, (A \wedge B)^{\mathtt{f}^+}}{C, A^{\mathtt{f}^+}, B^{\mathtt{f}^+}}$$

$$\frac{C, (A \wedge B)^{\mathtt{t}^-}}{C, A^{\mathtt{t}^-}, B^{\mathtt{t}^+} \quad C, A^{\mathtt{t}^-}, B^{\mathtt{t}^-} \quad C, A^{\mathtt{t}^+}, B^{\mathtt{t}^-}}$$

$$\frac{C, (\neg A)^{\mathtt{t}^+}}{C, A^{\mathtt{f}^+}} \qquad \frac{C, (\neg A)^{\mathtt{f}^+}}{C, A^{\mathtt{t}^+}} \qquad \frac{C, (\neg A)^{\mathtt{t}^-}}{C,} \qquad \frac{C, (\neg A)^{\mathtt{f}^-}}{C, A^{\mathtt{t}^-, \mathtt{f}^-}}$$

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

$$\frac{C, (\forall x_S.\ A[x_S])^{\mathtt{f}^+}}{C, A[f(y^1, \ldots, y^n)]^{\mathtt{f}^+} \quad C, ([f(y^1, \ldots, y^n)/x]S)^{\mathtt{t}^+\mathtt{t}^-}}$$

$$\frac{C, (\forall x_S.\ A[x_S])^{\mathtt{f}^-}}{C, A[f(y^1, \ldots, y^n)]^{\mathtt{f}^-} \quad C, ([f(y^1, \ldots, y^n)/x]S)^{\mathtt{t}^+\mathtt{t}^-}}$$

$$\frac{C, (!A)^{\mathtt{t}^+}}{C, A^{\mathtt{t}^+\mathtt{f}^+}} \qquad \frac{C, (!A)^{\mathtt{f}^+}}{C, A^{\mathtt{t}^-\mathtt{f}^-}} \qquad \frac{C, (!A)^{\mathtt{t}^-}}{C} \qquad \frac{C, (!A)^{\mathtt{f}^-}}{C}$$

$$\frac{C, (\mathbf{T}A)^{\mathtt{t}^+}}{C, A^{\mathtt{t}^+\mathtt{t}^-}} \qquad \frac{C, (\mathbf{T}A)^{\mathtt{f}^+}}{C, A^{\mathtt{f}^+\mathtt{f}^-}} \qquad \frac{C, (\mathbf{T}A)^{\mathtt{t}^-}}{C} \qquad \frac{C, (\mathbf{T}A)^{\mathtt{f}^-}}{C}$$

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.

The transformations can be directly derived from the semantics of the connectives and quantifiers. Due to space restrictions we have not presented all of them above. Note that the transformations for the universal quantifier have to associate the restriction S with the variable x, that is, in the resolution setting, we assume variables to be pairs, consisting of a symbol and a restriction. Furthermore Skolem functions have to conserve security and insecurity. In particular Skolem constants are always secure.

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 $\mathbf{CNF}(\Phi)$.

Assumption 3.3 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 restrictions, as quantification over empty sets are vacuously true. In this situation we have three possibilities, either to forbid degenerate quantifications, or empty restrictions, 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 and do not appear in everyday language. See [7] for the other possibilities. Thus we will assume 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 3.4 *Let Φ be a set of labelled sentences, then the clause normal form* $\mathbf{CNF}(\Phi)$ *is satisfiable, iff Φ is.*

[3] The logics of Cohn and Weidenbach [4, 8] do away with this restriction that has always been considered as a technical anomaly that has alleviated the need of special treatments in the transformation to clause normal form and for instantiations in the resolution calculus: A unifier that contains variables of sorts that are empty does not lead to a correct refutation.

Proof sketch: The assertion critically depends on the fact that the notion of satisfiability employed there takes the restrictions into account: A clause is valid in a Σ-model $\mathcal{M}$, iff for one literal L^α $\mathcal{I}_\varphi(L) = \alpha$ for all well-sorted assignments φ into $\mathcal{M}$. With this notion, the assertion can be reduced to the standard argumentation about Skolemisation and a tedious calculation with the truth tables from 2.7. □

Now we proceed to give a simple resolution calculus, which utilises standard (unsorted) unification. In [7], we have further improved a similar 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 restriction constraint) that ensures the soundness (with respect to the restrictions on the variables) of the unifier.

Definition 3.5 (Restriction Constraints)
Let $\sigma = [t^1/x^1_{S_1}], \ldots, [t^n/x^n_{S_n}]$ be a substitution, then we define the *restriction constraint for* σ to be the clause

$$\mathcal{RC}(\sigma) := \{([t^1/x^1]S_1)^{\mathtt{f}^+\mathtt{f}^-}, \ldots, ([t^n/x_n]S_n)^{\mathtt{f}^+\mathtt{f}^-}\}$$

These labelled formulae are residuated in the $\mathcal{RPL}$ rules and have to be refuted in order to guarantee that $\mathcal{A}^{\pm}_\varphi(S,t)$ holds (cf. definition 2.7) for every instance t instantiated for a variable x with restriction S.

Definition 3.6 (Resolution Inference Rules ($\mathcal{RPL}$))

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

where $\alpha \neq \beta$ and σ is the most general unifier of L and M. Here we have assumed α and β to be single truth values, naturally the rules can be easily extended to sets of truth values.

Remark 3.7 Note that clauses containing $A^{\mathtt{t}^+\mathtt{f}^+\mathtt{t}^-\mathtt{f}^-}$ are tautological 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 3.8 Let A be a sentence and Φ be the clause normal form of the set $\{\{A^{\mathtt{f}^+,\mathtt{t}^-,\mathtt{f}^-}\}\}$ then we say that A can be *proven in* $\mathcal{RPL}$ ($\vdash A$), iff there is a derivation of the empty clause □ from Φ with the inference rules above.

Theorem 3.9 (Soundness) $\mathcal{RPL}$ is sound.

Proof sketch: The soundness of the resolution and factoring rules is established in the usual way taking into account that the restriction constraints make the substitutions "well-sorted" and thus compatible with the semantics: The restriction constraints add two literals $([t/x]S)^{\mathtt{f}^+}$, $([t/x]S)^{\mathtt{f}^-}$ per component of the substitution, which only can be refuted if indeed $([t/x]S)^{\mathtt{t}^+}$ or $([t/x]S)^{\mathtt{t}^-}$ are valid. □

Definition 3.10 Let $C := \{L_1^{\alpha_1}, \ldots, L_n^{\alpha_n}\}$ be a clause, then the *conditional instantiation* $\sigma{\downarrow}(C)$ *of* σ *to* C is defined by

$$\sigma{\downarrow}(C) := \{\sigma(L_1^{\alpha_1}), \ldots, \sigma(L_n^{\alpha_n})\} \cup \mathcal{RC}(\sigma|_{\mathbf{Free}(C)})$$

The following result from [8] is independent of the number of truth values.

Lemma 3.11 *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 3.12 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 $\{\sigma{\downarrow}(C) \mid C \in \mathbf{CNF}(A), \sigma \text{ ground}, \mathbf{Dom}(\sigma) = \mathbf{Free}(C)\}$

Definition 3.13 We will call two literals L^α and L^β *complementary*, if $\alpha \neq \beta$.

Definition 3.14 (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 $\mathcal{H}(\Phi) \longrightarrow \{\mathtt{t}^+, \mathtt{f}^+, \mathtt{t}^-, \mathtt{f}^-\}$. 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 $\mathcal{H} := \{L^\alpha \mid \alpha = \nu(L), L \in \mathcal{H}(\Phi)\}$.

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 3.15 (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 $\Phi := \mathbf{CNF}(A)$. The set

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

is a Σ-Herbrand model for $\Psi := \mathbf{CNF}_H(A)$ 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 $C = \sigma{\downarrow}(D) = \sigma(D) \cup \mathcal{RC}(\sigma)$.

Without loss of generality we can assume that $\mathcal{I}(S_i)(\mathcal{I}_\varphi(t^i)) \in \{\mathtt{t}^+, \mathtt{t}^-\}$, since otherwise $\mathcal{I}_\varphi([t^i/x_i]S^i) \in \{\mathtt{f}^+, \mathtt{f}^-\}$, and therefore $([t^i/x_i]S^i)^\gamma \in \mathcal{H}$ for $\gamma \in \{\mathtt{f}^+, \mathtt{f}^-\}$, 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 and $\mathcal{A}$ the Herbrand base for $\mathcal{H}$. Furthermore let $\mathcal{I}(f^n)$ and $\mathcal{I}(P^n)$ be partial functions, such that

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

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 well-sorted Σ-assignment. Since $\mathcal{A}$ is a set of ground terms φ is also a ground substitution and moreover $([t^i/x_i]S_i)^{\mathtt{t}^+} \in \mathcal{H}$ or $([t^i/x_i]S_i)^{\mathtt{t}^-} \in \mathcal{H}$ by construction of $\mathcal{I}$ and the fact that φ is well-sorted.

$\mathcal{H}$ is a Σ-Herbrand model for Ψ and thus $\varphi{\downarrow}(C) \cap \mathcal{H} = (\varphi(C) \cup \mathcal{RC}(\varphi)) \cap \mathcal{H} \neq \emptyset$. 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 3.16 *A set Φ of ground unit clauses is unsatisfiable iff it contains two complementary literals.*

Theorem 3.17 (Ground Completeness) *Let Φ be an unsatisfiable set of ground clauses, then there exists a $\mathcal{RPL}$ derivation of the empty clause from Φ.*

Theorem 3.18 (Completeness) *$\mathcal{RPL}$ 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 [8], since they are independent of the number of truth values. □

4 Example

At first we want to give an example for quantificational presuppositions and then shortly discuss existential presuppositions.

Let us assume the following information. There is a company `TheCompany` which wants to fire people, but they have a social touch and don't fire any persons which have children. We are worried whether John will be fired, but then we hear that his children are sleeping. Implicitly we can conclude from this information that John has children and hence will not be fired.

This can be encoded in $\mathcal{PL}$ by the following statements:[4]

A $\forall x_{\mathcal{D}}.\, \mathtt{fires}(\mathtt{TheCompany}, x) \to \neg\mathtt{parent}(x)$
B $\forall x_{\mathcal{D}}.\, (\exists y_{\mathcal{D}}.\, \mathtt{child}(y, x)) \to \mathtt{parent}(x)$
C $\forall x_{\mathtt{child}(x,\mathtt{John})}.\, \mathtt{sleeps}(x)$
T $\neg\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John})$

with the term declarations $(\mathcal{D}(\mathtt{TheCompany}))^{\mathtt{t}^+}$ and $(\mathcal{D}(\mathtt{John}))^{\mathtt{t}^+}$. In order to prove the theorem T, the following generalised clause set has to be refuted:

A $(\forall x_{\mathcal{D}}.\, \mathtt{fires}(\mathtt{TheCompany}, x) \to \neg\mathtt{parent}(x))^{\mathtt{t}^+}$
B $(\forall x_{\mathcal{D}}.\, (\exists y_{\mathcal{D}}.\, \mathtt{child}(y, x)) \to \mathtt{parent}(x))^{\mathtt{t}^+}$
C $(\forall x_{\mathtt{child}(x,\mathtt{John})}.\, \mathtt{sleeps}(x))^{\mathtt{t}^+}$
T $(\neg\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{t}^-,\mathtt{f}^+,\mathtt{f}^-}$

By the rules for forming a clause normal form we get the clauses:

A1 $(\mathtt{fires}(\mathtt{TheCompany}, x_{\mathcal{D}}))^{\mathtt{f}^+}, (\mathtt{parent}(x_{\mathcal{D}}))^{\mathtt{f}^+}$
A2 $\mathcal{D}(c_1)^{\mathtt{t}^+}$
B1 $(\mathtt{child}(y_{\mathcal{D}}, x_{\mathcal{D}}))^{\mathtt{f}^+}, (\mathtt{parent}(x))^{\mathtt{t}^+}$
B2 $\mathcal{D}(c_2)^{\mathtt{t}^+}$
B3 $\mathcal{D}(c_3)^{\mathtt{t}^+}$
C1 $(\mathtt{sleeps}(x_{\mathtt{child}(x,\mathtt{John})}))^{\mathtt{t}^+}$
C2 $(\mathtt{child}(c_4, \mathtt{John}))^{\mathtt{t}^+}$
C3 $\mathcal{D}(c_4)^{\mathtt{t}^+}$
T $(\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{f}^-}$,
$(\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{t}^+}$,
$(\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{t}^-}$

By resolution we get from Res(B1,C2):

R1 $(\mathtt{parent}(\mathtt{John}))^{\mathtt{t}^+}, (\mathcal{D}(c_4))^{\mathtt{f}^+}, (\mathcal{D}(c_4))^{\mathtt{f}^-}$

Two-times resolving with C3 results in:

R2 $(\mathtt{parent}(\mathtt{John}))^{\mathtt{t}^+}$

which in turn can be resolved with A1:

R3 $(\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{f}^+}$,
$(\mathcal{D}(\mathtt{John}))^{\mathtt{f}^+}, (\mathcal{D}(\mathtt{John}))^{\mathtt{f}^-}$

The last two literals can be resolved away using the term declaration $(\mathcal{D}(\mathtt{John}))^{\mathtt{t}^+}$. T can be resolved three times with the resulting unit $(\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{f}^+}$, whereby finally the empty clause is derived.

Please note that in a direct first-order translation of the above text, the essential information in C2 that John has a child cannot be derived and hence no proof can be found.

The second form of presuppositions concern the fact that all constants of classical logic exist just because of mentioning them. For instance, classical logic is not a good tool for a dispute of a theist and an atheist about the existence of God, since if the atheist only mentions God, he would admit the existence of God. In $\mathcal{PL}$, however, the status of statements about constants can be insecure and in particular no existence is assumed, unless otherwise specified by term declarations.

5 Conclusion

We have developed a four-valued logic for the formalisation of everyday reasoning with presuppositions. This system generalises the system proposed by Bergmann in [3]. Furthermore we have presented a sound and complete resolution calculus for our system, which uses the sort mechanism to capture Bergmann's restricted quantifications.

Our calculus can be seen as an extension of classical logic that combines methods from many-valued logics (cf. [1, 5]) for a correct treatment of the secure and insecure information and order-sorted logics (see [8]) for an adequate treatment of restricted domains. In contrast to the partial function calculi in [6, 7] $\mathcal{PL}$ does not identify the insecure objects. However, just like in these logics, most definedness preconditions can be taken care of in the unification, making inferencing quite efficient.

Even though the research on presuppositions in linguistics has nowadays turned to dynamic and more pragmatically driven analyses, and away from the multi-valued treatment, this is not a counterargument to our system. In contrast to classical logic $\mathcal{PL}$ makes it possible to specify (and reason with) presuppositions, so that once the linguistic analyses are used for reasoning, some system like our's will be indispensable.

REFERENCES

[1] Matthias Baaz and Christian G. Fermüller, 'Resolution for many-valued logics', in *Proceedings of LPAR*, ed., A. Voronkov, pp. 107–118, St. Petersburg, Russia, (1992). Springer LNAI 624.

[2] Walter A. Carnielli, 'Systematization of finite many-valued logics through the method of tableaux', *The Journal of Symbolic Logic*, **52**, 473–493, (1987).

[3] Merrie Bergmann, 'Presupposition and two-dimensional logic', *Journal of Philosophical Logic*, **10**, 27-53, (1981).

[4] Anthony G. Cohn, 'A more expressive formulation of many sorted logics', *Journal of Automated Reasoning*, **3**, 113–200, (1987).

[5] Reiner Hähnle, *Automated Theorem Proving in Multiple Valued Logics*, Oxford University Press, 1994.

[6] Manfred Kerber and Michael Kohlhase, 'A mechanization of strong Kleene logic for partial functions', in *Proceedings of the 12th CADE*, ed., Alan Bundy, pp. 371–385, (1994). Springer LNAI 814.

[7] Manfred Kerber and Michael Kohlhase, 'A tableau calculus for partial functions', in *Annals of the Kurt-Gödel-Society*, Springer Verlag, (1996). forthcoming.

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

[4] There is a typo in the treatment of formula B in the proceedings, which is corrected here.