THE UNIVERSITY OF BIRMINGHAM
School of Computer Science
Edgbaston, Birmingham B15 2TT, England
URL: http://www.cs.bham.ac.uk/

Reasoning without Believing – On the Mechanization of Presuppositions and Partiality

Manfred Kerber and Michael Kohlhase

Published as: Forthcoming in LNCS, Suresh Manandhar, ed., *Computational Logic for Natural Language Understanding*

Reasoning without Believing: On the Mechanization of Presuppositions and Partiality

Manfred Kerber[1] and Michael Kohlhase[2]

[1]School of Computer Science
The University of Birmingham
Birmingham B15 2TT, England
e-mail: M.Kerber@cs.bham.ac.uk
http://www.cs.bham.ac.uk/~mmk

[2]Fachbereich Informatik
Universität des Saarlandes
66041 Saarbrücken, Germany
e-mail: kohlhase@cs.uni-sb.de
http://www.ags.uni-sb.de/~kohlhase

Abstract. It is well-known that many relevant aspects of everyday reasoning based on natural language cannot be adequately expressed in classical first-order logic. In this paper we address two of the problems, firstly that of so-called *presuppositions*, expressions from which it is possible to draw implicit conclusion, which classical logic normally does not warrant, and secondly the related problem of *partiality* and the adequate treatment of undefined expressions. In natural language, presuppositions are quite common, they can, however, only insufficiently be modeled in classical first-order logic. For instance, in the case of universal quantification one normally uses restrictions in natural language 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. Similarly, undefined expressions can be handled in natural language discourses and utterances are not only classified into the two categories 'true' and 'false'. This has led to the development of various better-suited many-valued logics.

By combining different approaches we can now give a static description of presuppositions and undefinedness within the same framework. Additionally, we have developed an efficient mechanization of the induced consequence relation (which has been missing in the literature) by combining methods from many-valued truth-functional logics and sort techniques developed for search control in automated theorem proving.

1 Introduction

Classical first-order logic (FOL) has often been used as a basis for natural language semantics. But there are considerable discrepancies between the meaning of everyday language and the formal semantics of its naive translation into FOL. In this paper we address two important ones, the first is about the so-called presuppositions[1], that is, implicit assumptions in everyday reasoning about what exists and what not, the second is about the treatment of undefined expressions.

These phenomena have been widely studied in the philosophy of language from a semantic point of view. But the resulting logical formalisms do not cover the observed phenomena completely. At the same time, they lack an efficient mechanization. One of the more logic-oriented ways to cope with this phenomenon is to use a four-valued logic [Ber81]. We take this logic as a starting point, extend it to a five-valued version that is capable of capturing undefinedness as well, and finally mechanize this logic by a resolution calculus.

We will use the remainder of the introduction to pinpoint the contribution of this paper, review the problem and related work in presupposition logics[2], logical treatment of undefinedness and automated theorem proving providing background information for the formal development of our logical system, presented in Section 2. Then we will present a resolution calculus and briefly sketch the soundness and completeness results (Section 3). Finally, we explain the method by two simple examples (Section 4) before we conclude with a discussion of further work.

1.1 Contribution of this paper

We present a five-valued logic for the formalisation of everyday reasoning with presuppositions. This system enhances the four-valued system proposed by Bergmann in [Ber81] and mechanized by the authors in [KK96a] by providing a fifth truth value for syntactical undefinedness (for instance, application of functions to objects outside their domain). It adds a further degree of partiality to the system and is treated in a Kleene way as described in [KK94,KK96b]. Furthermore we present a sound and complete resolution calculus for our system, which uses a sort mechanism to capture Bergmann's restricted quantifications.

This paper is *not* intended as a linguistic account of the semantics of natural language sentences involving presuppositions. Such accounts (see for instance [vdS92,Kra95,Bea95]) have nowadays mostly turned away from multi-valued logics towards discourse representation theories (DRT, see e.g. [KR93]. Rather than a linguistic account, it is an investigation of the underlying logical principles and the rôle of partiality, which also plays a major rôle in the linguistic theories (cf. [Kra95] for a discussion). In particular, this paper does not make any claim about the process of constructing logical representations from natural language utterances.

Since the discourse representation theories used in the linguistic accounts are essentially dynamified versions of FOL, we are convinced that the methods and

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

[2] Since we are mainly interested in the inference aspect of presuppositions, we will not go into the large body of linguistic literature on presuppositions.

principles presented in this paper can be transported to the linguistic theories to give them more leverage. Our logic $\mathcal{PL}$ is a generalization of FOL in a direction that is orthogonal to dynamification, so that the ideas from both systems can be combined into a joint logical system. In fact Krahmer's work already takes a step in that direction: it uses a version of DRT that is partialized using ideas from Kleene's weak three-valued logic, which has well-known disadvantages for modeling presuppositions.

Once reasoning calculi and in particular automated theorem proving techniques have been developed for DRT, it is also possible to generalize our resolution calculus to such a logic. This would alleviate one central methodological weakness of the linguistic analyses, which use (intuitive) reasoning on the meta-level to argue about presupposition (failure) in the presence of world knowledge without being able to make this formal and thus part of the analysis.

Since direct inferences on DRT (first steps have been taken in [Sau93]) seem to be a difficult problem, the present paper studies the static case in isolation.

1.2 Presuppositions

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 FOL treats the two kinds almost the opposite way: constants always must have denotations, that is, just speaking about an object means that it must exist (for instance, speaking about a dragon, means that there is one), while quantifications are unrestricted and therefore always range over the whole (albeit 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.

This has been recognized quite early in the field of philosophical logics, in particular, that there is a distinction between different kinds of utterances. For instance Strawson [Str73, p.203 ff] analyses the sentence "The king of France is wise." and relates his view to that of Russell [Rus05].

> *Firstly the sentence is significant, that is, its meaning can be understood. Secondly it is true only if there existed in fact one and only one king of France, and if he were wise.*

Strawson argues that – unlike to this view of Russell – the sentence is not necessarily true or false. He underpins this view as follows:

> *Now suppose someone were in fact to say to you with a perfectly serious air: 'The king of France is wise.' Would you say, 'That's untrue' ... suppose he went on to* ask *you whether you thought that what he had just said was true, or was false ... You might ... say something like: 'I'm afraid you must be under a misapprehension. France is not a monarchy. There is no king of France.'*

Later on Strawson explains, why the sentence is significant, namely, since

> *the sentence* could *be used, in certain circumstances, to say something true or false, that the expression* could *be used, in certain circumstances, to mention a particular person...*

In order to cope with such kinds of phenomena, many different logics have been developed. They can be roughly categorized according to the truth values they use. There is classical two-valued logic, which is not particularly well-suited according to the discussion of Strawson. In order to overcome such problems three-valued approaches and approaches with truth-value gaps have been adapted and been developed, these are in particular the logics developed by Kleene [Kle52] and van Fraassen [vF66]. These logics are quite well-suited to indicate that something is wrong with a certain kind of statement (like "The king of France is wise") and assign to it either the truth value `undefined` or no truth value at all. However, they do not enable any kind of hypothetical reasoning (like "The king of France is wise.", "If somebody is wise, he is gracious", hence "The king of France is gracious.") In order to cope with corresponding phenomena Herzberger [Her73] has developed a two-dimensional approach, in which essentially the truth values consist of pairs of values, the first value is a traditional truth value, that is, either `true` or `false` and the second is a kind of presupposition value, this value is `1` if all presuppositions are fulfilled and `0` else. This approach was modified and further developed by Bergmann [Ber81], who kept the two-dimensional approach, but interpreted the values slightly differently so that she got different truth tables. While the first component in Bergmann's approach represents the classical two values, the second expresses whether the sentence is secure or insecure. For instance, in any reasonable axiomatization "The king of France is wise" would be insecure, but may be true or false.

The presupposition that all mentioned objects exist is adequate unless the opposite is explicitly said. However, everyday language allows to explicitly disable certain presuppositions and is well-suited for instance for a discussion between a theist and an atheist about the properties of God, *assuming he exists.* Classical logic, however, is not well-suited as basis for formalizing such a dispute, since the fact that all constants denote something, means if the atheist only uses the word "God", he would admit the existence of God. In the formal system we are going to present, however, the status of statements about constants can be insecure and in particular no existence is assumed, unless otherwise specified.

While this kind of presuppositions (existence of constants) is reasonable for most cases, quantificational presuppositions of everyday languages fundamentally differ from those in classical logic. For instance, an everyday sentence like "All children of John are sleeping" presupposes that John really has children. Therefore the representation in FOL $\forall x.\mathtt{child\_of}(x, \mathtt{John}) \rightarrow \mathtt{sleeps}(x)$ 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 operator of the syntactic form $\forall x_{\mathtt{child\_of}(x,\mathtt{John})}.\ \mathtt{sleeps}(x)$, as is used in generalized quantifier theory [BC81,Mos57]. Here 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.

1.3 Undefinedness

In this paper we argue that the possibility to go for a more extended form of semantics that allows for hypothetical reasoning, like those developed by Herzberger and Bergmann, is quite important, however, it does not – at least when it is compared to the Kleene approach – make the point obsolete that certain utterances do not make any sense at all (either because they are not understandable, e.g. since certain things are not defined, or because the speech is not serious at all). Sentences like "The present king of France is wise" or "God is almighty" should be distinguished from sentences like "Madagascar is east of the equator" or even "Kraba bla shle". For the first category, Strawson's criterion that it could be true (or false) in certain circumstances holds, while sentences of the second category do not make any sense. So we would like to assign to those a fifth truth value, marking them as `undefined`.

Again, if we consider the semantics of a quantification in presence of undefined values, we have to be precise what an expression like $\forall x.\ P(x)$ should mean. In general we do not want this sentence to tell us anything about undefined objects (since there isn't much to tell about them, except that they are undefined), so we would like to interpret this formula as "for all defined expressions x, $P(x)$ holds". In order to be precise about this, here again, we use only restricted quantifications, that is, expressions of the kind $\forall x_{\mathtt{human}(x)}.\ \mathtt{mortal(x)}$. If we do not want to make any restriction on x, that is, to quantify over the whole universe, we write $\forall x_{\mathfrak{D}(x)}.\ \ldots$, where $\mathfrak{D}$ stands for "defined". In the same spirit we will use a predicate $\mathfrak{S}$ that stands for "secure".

For modeling undefinedness, we have to choose among various approaches. We advocate here for Kleene's approach, in which undefinedness is contagious for all terms and atomic formulae, that is, whenever a term contains an undefined expression like "East of the equator", that is, is of the form "North of east of the equator", then it is undefined too (and in particular never covered by a quantification). Any atomic formula containing such a term (like "Madagascar is east of the equator") is mapped to a fifth truth value u, the truth value "undefined".

Formulae may, however, be true, even if they do contain an undefined expression. For instance, "Madagascar is east of the equator *or* south of the equator" is true. While the first part evaluates to "undefined", the second to "true" (more precisely to "true and secure"), the disjunction evaluates to true too (actually to "true and insecure" to indicate that something is strange about the sentence).

1.4 Automated Theorem Proving

One of the cornerstones in the development of logic in the twentieth century is the development of sound and complete calculi, that is, of calculi which exactly correspond to a given semantics. This introduces a fully syntactical notion of proof which represents, but does not have to refer to the semantics. By this, a mechanical treatment of the semantical consequence relation and in particular of theorem-hood becomes possible at least in principle. The simplest calculi are so-called Hilbert calculi, which derive true statements from axiom schemata by sound calculus rules. While this is quite convenient for proving theoretical results about the calculus, it does not correspond to usual mathematical practice. Therefore, Gentzen [Gen35] developed the so-called calculi of Natural Deduction, which also allow to make temporary assumptions. Variants of this calculus are much better suited for practical theorem proving than Hilbert calculi and are very popular in advanced *interactive* proof development environments.

The most important breakthrough in the development of *automated* theorem proving was Robinson's invention of the resolution principle in 1965 [Rob65]. Its three important features compared to the calculi above are: It works with formulae in a conjunctive normal form, the so-called clause normal form, which simplifies the calculus. It is a refutation calculus, which tries to prove a theorem by refuting the negation of it; this is the basis for automating the theorem process as goal-oriented search for the (unsatisfiable) empty clause. Finally, but most importantly, it uses a procedure called unification, which essentially computes for terms the most general substitution that makes formulae equal; this makes it possible to avoid instantiation of universal quantifiers and thus make the search space finitely branching.

In the sequel the idea of unification was transferred to other calculi, furthermore the basic calculi were very much refined in order to get very efficient implementations, such as the OTTER [McC90] theorem prover.

An important refinement to theorem proving systems was the introduction of sorts into the representation language[3], by which the universe of discourse can be structured. This results in a significant improvement of the search complexity in many cases. For certain problem classes the refutation/unification-based framework provides a very powerful tool for automated theorem proving and in fact it has even been possible to prove mechanically several mathematical theorems that had been open previously. See [WGR96] for the recent developments important to this paper.

On top of the operationalization of FOL Carnielli [Car87], Hähnle [Häh94], Baaz and Fermüller [BF92] developed methods for the operationalization of *many-valued* first-order logics. In this paper we will advocate a many-valued logic that is much better suited to describe certain phenomena of presuppositions and undefinedness than classical logic is. However, the above-mentioned operationalizations for many-valued logics can not directly be applied in our context since their quantifications are unrestricted, that is, quantified formulae obtain their truth values from those of *all* instances of the scope, while we need a restricted form of quantification. Therefore a direct utilization of these methods is impossible for Bergmann's logic as well as for Kleene's logic, since the quantifiers range only over a restricted domain.

[3] For instance, in a sorted logic it is possible to say $\forall x_{\mathtt{human}}.\, \mathtt{mortal(x)}$, instead of $\forall x.\, \mathtt{human}(x) \rightarrow \mathtt{mortal}(x)$.

Once this problem is mastered, it is relatively straightforward to build a suitable resolution calculus by employing standard techniques from multi-valued theorem proving: Only the standard first-order translation from formulae to a clause normal form has to be adapted, so that it respects the many-valued semantics of the connectives and quantifiers. In order to master the restriction problem, we employ a sorted version of the multi-valued logic. For our purpose the method developed by Weidenbach [Wei91] is most appropriate, since it allows arbitrary sort expressions that may occur anywhere in formulae.

2 Presupposition Logic

In this section, we develop a logic system $\mathcal{PL}$ by combining ideas from the presupposition logic of Bergmann [Ber81] with methods from our mechanization of partial functions [KK94] by a sorted variant of three-valued Kleene logic.

The main problem is to give a proper treatment of restricted quantification and their presuppositions. But let us first fix the notation for the logical language.

Definition 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 special predicates $\mathfrak{D}$ and $\mathfrak{S}$, called the **definedness** and **security** predicates.

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.

From this set of basic material, the terms and formulae are built up in a rather standard manner, with the only exception that quantification is generalized to *restricted* quantifiers, where the quantifier involves not only a variable binding, but also a formula S for restricting the domain of quantification.

Definition 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 $P \in \mathcal{P}^k$) and of **compound formulae** $A \wedge B$, $A \vee B$, $A \rightarrow B$, $\neg A$, $\mathbf{S}A$, $\mathbf{T}A$, $\mathbf{D}A$, $\forall x_S\textbf{.}\ A$, and $\exists x_S\textbf{.}\ A$, where A, B, and S are formulae.

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

Note that the concept of restricted quantification is a generalization 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 $\forall x_{P(x)}\textbf{.}\ A$ is equivalent to the sorted quantification $\forall x_P\textbf{.}\ A$ as it can be found in sorted logics.

We now will define the five-valued, two-dimensional semantics for $\mathcal{PL}$. The main feature of Bergmann's logic for presuppositions 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. Intuitively, the truth values are augmented 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. Furthermore, we add a value $\mathtt{u}$ well-known from the Kleene-type logics. In the following we denote the set of **truth values** by $\mathcal{B} = \{\mathtt{t}^+, \mathtt{f}^+, \mathtt{t}^-, \mathtt{f}^-, \mathtt{u}\}$. This set serves as the semantical domain for formulae. The meaning of terms can best be formalized with a more algebraic notion for the universe of individuals, which we define now.

In defining the semantics we have different options, one is to assume strictness for all function and predicate symbols. While strictness with respect to undefinedness makes sense in almost all cases, the strictness assumption for security may be too strong for some applications. For instance, we might want to model the fans of Sherlock Holmes as secure even if the detective is purely fictional. Although we assume strictness for security too in the following, this assumption can be lifted by simply skipping part 2d of the following definition. In such a case the burden of specifying where strictness is wanted would be posed in the term declarations (cf. Definition 8). Clearly, the second strictness rule would have to be dropped from the resolution calculus too (see Definition 18).

Definition 3 (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 an arbitrary set that contains a special element $\perp$,
2. the **interpretation function** $\mathcal{I}$ obeys the following restrictions:
   (a) $\mathcal{I}(\mathfrak{D})(\perp) = \mathtt{f}^+$ and $\mathcal{I}(\mathfrak{D})(a) = \mathtt{t}^+$ else. Furthermore $\mathcal{I}(\mathfrak{S})(a) \in \{\mathtt{f}^+, \mathtt{t}^+\}$, but $\mathcal{I}(\mathfrak{S})(\perp) = \mathtt{u}$.
   We call elements $a \in \mathcal{A}$ **secure** if $\mathcal{I}(\mathfrak{S})(a) = \mathtt{t}^+$ and **insecure** if $\mathcal{I}(\mathfrak{S})(a) = \mathtt{f}^+$, and else **undefined**. Thus we can subdivide $\mathcal{A}$ into subsets $\mathcal{A}^+$ of secure and $\mathcal{A}^-$ of insecure elements. Note that $\mathcal{A} = \mathcal{A}^+ \cup \mathcal{A}^- \cup \{\perp\}$ and $\mathcal{A}^+ \cap \mathcal{A}^- = \emptyset$.
   (b) For all function symbols f, the function $\mathcal{I}(f): \mathcal{A}^k \longrightarrow \mathcal{A}$ is strict for $\perp$, that is, $\mathcal{I}(f)(a_1, \ldots, a_k) = \perp$, if $a_i = \perp$ for at least one i.
   (c) If $p \neq \mathfrak{D}$ is a predicate symbol, then the five-valued relation $\mathcal{I}(p)$ is a mapping from $\mathcal{A}^k$ into $\mathcal{B}$, the set of truth values (see also Example 9). It is strict for $\perp$, that is, $\mathcal{I}(p)(a_1, \ldots, a_k) = \mathtt{u}$, if $a_i = \perp$ for at least one i.
   (d) All predicates and functions are strict for $\mathfrak{S}$, i.e., $\mathcal{I}(f)(a_1, \ldots, a_k) \in \mathcal{A}^- \cup \{\perp\}$, if $a_i \in \mathcal{A}^-$ and $\mathcal{I}(p)(a_1, \ldots, a_k) \notin \{\mathtt{t}^+, \mathtt{f}^+\}$, if $a_i \in \mathcal{A}^-$ for at least on i.[4]

[4] Note that $\mathcal{I}(p)(a_1, \ldots, a_k)$ may be $\mathtt{u}$. This is in particular the case when one of the arguments evaluates to $\perp$. If, however, an argument of an expression is insecure the whole expression cannot be secure. For instance, the expression "mother of Sherlock Holmes" would not be evaluated in $\mathcal{A}^+$. We do not require an "if and only if" in the definition, that is, for a real person like "Conan Doyle", the term "guardian angel of Conan Doyle" can be interpreted in $\mathcal{A}^-$.

Note that this treatment differs from the Kleene approach taken in [KK94], where all undefined elements are identified with the semantic object $\perp$, since here we distinguish between secure, insecure, and undefined objects. In particular we want to be able to reason about insecure objects, but not about undefined ones.

While the meaning of material from the signature Σ is given by the interpretation function of a strict Σ-algebra, that of the variables is locally given by an assignment function φ, which can then be combined to a value function $\mathcal{I}_\varphi$, which gives the value of a term or formula, assuming that the variables are evaluated as prescribed by φ.

Definition 4 (Value of Terms and Atoms). 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 $\varphi, [a/x]$. Now we can inductively define the **value function** $\mathcal{I}_\varphi$ for terms and atoms 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. $\mathcal{I}_\varphi(f(t^1, \ldots, t^k)) := \mathcal{I}(f)(\mathcal{I}_\varphi(t^1), \ldots, \mathcal{I}_\varphi(t^k)) \in \mathcal{A}$, for $f \in \mathcal{F}^k$.
4. $\mathcal{I}_\varphi(p(t^1, \ldots, t^k)) := \mathcal{I}(p)(\mathcal{I}_\varphi(t^1), \ldots, \mathcal{I}_\varphi(t^k)) \in \mathcal{B}$, if $p \in \mathcal{P}^k$.

Definition 5. 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:

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

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

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

$\rightarrow$	$\mathtt{t}^+$	$\mathtt{f}^+$	$\mathtt{t}^-$	$\mathtt{f}^-$	$\mathtt{u}$
$\mathtt{t}^+$	$\mathtt{t}^+$	$\mathtt{f}^+$	$\mathtt{t}^-$	$\mathtt{f}^-$	$\mathtt{u}$
$\mathtt{f}^+$	$\mathtt{t}^+$	$\mathtt{t}^+$	$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{t}^-$
$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{f}^-$	$\mathtt{t}^-$	$\mathtt{f}^-$	$\mathtt{u}$
$\mathtt{f}^-$	$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{t}^-$	$\mathtt{t}^-$
$\mathtt{u}$	$\mathtt{t}^-$	$\mathtt{u}$	$\mathtt{t}^-$	$\mathtt{u}$	$\mathtt{u}$

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

S	
$\mathtt{t}^+$	$\mathtt{t}^+$
$\mathtt{f}^+$	$\mathtt{t}^+$
$\mathtt{t}^-$	$\mathtt{f}^+$
$\mathtt{f}^-$	$\mathtt{f}^+$
$\mathtt{u}$	$\mathtt{f}^+$

D	
$\mathtt{t}^+$	$\mathtt{t}^+$
$\mathtt{f}^+$	$\mathtt{t}^+$
$\mathtt{t}^-$	$\mathtt{t}^+$
$\mathtt{f}^-$	$\mathtt{t}^+$
$\mathtt{u}$	$\mathtt{f}^+$

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

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

We call a restriction (x,S) **empty** under φ, 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 $a \in \mathcal{A}^+_{\varphi}(S,x)$ and $\mathcal{A}^+_{\varphi}(S,x) \neq \emptyset$ and if $\mathcal{I}_{\varphi,[a/x]}A = \mathtt{t}^-$ for all $a \in \mathcal{A}^-_{\varphi}(S,x)$
- $\mathtt{f}^+$ if there is an $a \in \mathcal{A}^+_{\varphi}(S,x)$ with $\mathcal{I}_{\varphi,[a/x]}A = \mathtt{f}^+$ and $\mathcal{I}_{\varphi,[a/x]}A \in \{\mathtt{t}^+,\mathtt{f}^+\}$ for all $a \in \mathcal{A}^+_{\varphi}(S,x)$.
- $\mathtt{t}^-$ if $\mathcal{I}_{\varphi,[a/x]}A \in \{\mathtt{t}^+,\mathtt{t}^-\}$ for all $a \in \mathcal{A}^{\pm}_{\varphi}(S,x)$, but either $\mathcal{A}^+_{\varphi}(S,x) = \emptyset$ or $\mathcal{I}_{\varphi,[a/x]}A = \mathtt{t}^-$ for some $a \in \mathcal{A}^+_{\varphi}(S,x)$.
- $\mathtt{f}^-$ if there is an $a \in \mathcal{A}^{\pm}_{\varphi}(S,x)$ with $\mathcal{I}_{\varphi,[a/x]}A = \mathtt{f}^-$
- $\mathtt{u}$ else

With the specification of the behaviors of the connectives and quantifiers we have completed the definition of the semantics of formulae. Let us now come back to the presuppositions of a formula A, which we can now determine from the semantics of A.

Remark 6 (Presuppositions and Preimplications). Following Bergmann [Ber81], we will determine presuppositions in terms of semantic anomaly (insecurity) rather than truth; we will use the logical concept of **preimplication** as a formalization of the intuitive concept of presuppositions. Let us say that formula A **preimplies** formula B (under an assignment φ) iff $\mathcal{I}_{\varphi}(A) \in \{\mathtt{t}^+,\mathtt{f}^+\}$ entails that $\mathcal{I}_{\varphi}(B) \in \{\mathtt{t}^+\mathtt{t}^-\}$, in other words, if the security of A entails the truth of B. With this, we can see that the sentence "The present king of France is wise" (which is insecure) preimplies the existence of a king of France irrespective of the truth value, i.e., irrelevant of whether we generally think that kings of France are wise or not. Furthermore, we can see that preimplication passes the classical linguistic test for presuppositions: The set of presuppositions must be invariant under negation. This condition is trivially met by preimplication, since negation does not change the security value of a formula.

Bergmann uses a negation connective $\sim$ that is identical to ours, except that $\mathcal{I}(\sim)(\mathtt{f}^-) = \mathtt{f}^-$. This is motivated by a slightly different understanding of presuppositions (cf. [Ber81]). Our choice of negation, which is more pleasing to a logician, and indeed more suited to hypothetical reasoning is not a restriction in principle, since we could have both connectives in our system. Moreover, the two connectives are interdefinable; therefore, we only need to treat one of them in our paper: It is an easy exercise to see that $\neg A = \sim A \vee ((\sim \mathbf{T}A) \wedge \mathbf{D}A)$ and $\sim A = \neg A \wedge (\mathbf{S}A \vee \neg \mathbf{D}A)$.

Definition 7 (Σ-Model). Let A be a formula, then we call A^{α} (the formula A indexed with the intended truth value $\alpha \in \mathcal{B}$), a **labeled formula**.

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. Analogously, $\mathcal{M} \models A^{\alpha}$ iff $\mathcal{I}_{\varphi}(A) = \alpha$ for all assignments φ. With this notion we can define the notions of **validity**, **(un)-satisfiability**, and **entailment** in the usual way.

We are normally only interested in classes of models that satisfy some common background knowledge about definedness of concepts (corresponding to questions whether Pegasus exists, whether he is a horse, or about the nature and existence of his left front hoof). For this we restrict our class of models and variable assignments to those that satisfy a set of so-called term declarations that can be supplied by the user to specify just this background knowledge.

Definition 8 (Term Declarations). In the following, we will assume a set $\mathcal{TD}$ of labeled formulae that we will call **term declarations** and restrict our attention to $\mathcal{TD}$-models, i.e., models that satisfy all labeled formulae in $\mathcal{TD}$.

From a purely theoretical point of view, term declarations do not yield more expressivity, since they can be axiomatized (any intended truth value can be characterized by combinations of the connectives $\neg$, **S**, **T**, and **D**). 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 optimizations of the calculus by sorted unification as in [KK96b].

Example 9. As simple example that also gives an intuition about five-valued predicates let us try to specify some definedness knowledge about the property of being male. The set of term declarations

$$male(doyle)^{\mathtt{t}^+}, male(holmes)^{\mathtt{t}^-}, male(austen)^{\mathtt{f}^+},$$
$$male(emma)^{\mathtt{f}^-}, male(baker\_street)^{\mathtt{u}}$$

tells us about our belief about famous fictional personalities and the authors that invented them[5]. This shows us that five-valued predicates subdivide the universe of individuals into five regions: one for every defined truth value (we have given examples for each of them) and the undefined region, which only contains the value $\bot$.

Remark 10. 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[6] and therefore lead to the same presuppositions as $\mathcal{PL}$ on the sorted fragment.

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

[5] In particular these tell us that the Names "Jane Austen" and "Arthur Conan Doyle" were not pen-names, but also that the predicate *male* doesn't make sense for a street, irrespective of whether it may exist or not.

[6] The logics of Cohn and Weidenbach [Coh87,Wei91] 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 – in general – lead to a correct refutation.

Remark 11. Bergmann's generalization of the restrictions to formulae adds practical expressivity to the logical system. For instance, the domain of quantification in the formula $A = \forall x_{\mathtt{son}(x,y)}.B$ will be sensitive to the assignment to the variable y. In particular, $\mathcal{I}_{[a/y]}(A)$ will make an assertion about the sons of a, while $\mathcal{I}_{[b/y]}(A)$ one about the sons of b. The negative side of this generalization is that the property of well-sortedness (and thus presupposition failure) becomes undecidable; this further illustrates the need for inference calculi for the entailment relation of presupposition logics.

Remark 12. The "tertium non datur" principle of FOL is no longer valid, since formulae can indeed be evaluated to five different truth values. We do however have a "sextum non datur" principle, that is, formulae cannot get a truth value different from the five in $\mathcal{B}$. This allows us to derive the validity of a formula (i.e., that it is true and secure in all models) by refuting that it is $\mathtt{t}^-$, $\mathtt{f}^+$, $\mathtt{f}^-$, or $\mathtt{u}$. (In the same line, if we are not interested in security, we can show the truth of a formula by refuting that it is $\mathtt{f}^+$, $\mathtt{f}^-$, or $\mathtt{u}$.) 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 generalization of the resolution calculi for partial functions [KK94], and that for Bergmann's logic [KK96a], which are in turn joint generalizations of Weidenbach's logics with dynamic sorts [Wei91] with ideas from [BF92,Häh94].

Definition 13 (Clause Normal Form). 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**. In order to enhance legibility, 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 [Häh94] notion of multi-labels in the form $C, A^{\alpha\beta}$ to mean C, A^α, A^β. The clause normal form of a labeled formula set is generated by the application of the following rules.

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

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

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

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

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

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

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

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

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

$$\frac{C, (\mathbf{S}A)^{\mathtt{t}^+}}{C, A^{\mathtt{t}^+\mathtt{f}^+}} \quad \frac{C, (\mathbf{S}A)^{\mathtt{f}^+}}{C, A^{\mathtt{t}^-\mathtt{f}^-\mathtt{u}}} \quad \frac{C, (\mathbf{S}A)^{\mathtt{t}^-}}{C} \quad \frac{C, (\mathbf{S}A)^{\mathtt{f}^-}}{C} \quad \frac{C, (\mathbf{S}A)^{\mathtt{u}}}{C}$$

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

$$\frac{C, (\mathbf{D}A)^{\mathtt{t}^+}}{C, A^{\mathtt{t}^+\mathtt{t}^-\mathtt{f}^+\mathtt{f}^-}} \quad \frac{C, (\mathbf{D}A)^{\mathtt{f}^+}}{C, A^{\mathtt{u}}} \quad \frac{C, (\mathbf{D}A)^{\mathtt{t}^-}}{C} \quad \frac{C, (\mathbf{D}A)^{\mathtt{f}^-}}{C} \quad \frac{C, (\mathbf{D}A)^{\mathtt{u}}}{C}$$

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

Note that the transformations for the universal quantifier introduce new free variables x_S^*, consisting of a variable x, a mode specifier $\in \{+, -, \pm\}$ (see Definition 17) and a restriction S.

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

The clause normal form transformations as presented above are not complete, that is, they do not transform every given labeled 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 normally do not appear in everyday language. See [KK96b] for the other possibilities. Thus we will assume that in all formulae in this paper the bound variables of quantifications occur in the scopes. Note that this assumption is not about the restrictions. In particular, $\forall x_S.A$ is allowed, even if $x \notin \mathbf{Free}(S)$. Intuitively, this means that x is unrestricted in a context that makes S true and the whole formula carries the presupposition that S is indeed true there.

Remark 14. As in many regular multi-valued logics like Kleene logic it is possible to define combined transformation rules for truth value sets. For instance, the transformation for $\forall$ with the set $\{\mathtt{t}^+, \mathtt{t}^-\}$ has the following form:

$$\frac{C, (\forall x_S.\, A[x_S])^{\mathtt{t}^+\mathtt{t}^-}}{C, A[x_S^{\pm}]^{\mathtt{t}^+\mathtt{t}^-}}$$

Using such transformations can result in a much shorter clause normal form[7], which in turn is a much better starting point for any refutation. Moreover, in some regular multi-valued logics (cf. [Häh94]) a careful analysis using sets of signs can result in a calculus that can be realized as a restriction strategy to existing theorem provers such as SPASS [WGR96]. We have executed this for three-valued Kleene-logic in [KK97], obtaining a simple path to a highly efficient implementation. Since $\mathcal{PL}$ is very similar to Kleene logic, it seems plausible that a similar reduction is possible for $\mathcal{RPL}$ as well, but we leave this to further research.

Definition 15. Let $\mathcal{M} = (\mathcal{A}, \mathcal{I})$ be a Σ-model, then we will call a variable assignment φ **well-sorted** iff $\varphi(x_S^*) \in \mathcal{A}_\varphi^*(S, x)$ for all $x \in \mathbf{Dom}(\varphi)$ and say that φ **satisfies** a clause C in $\mathcal{M}$, 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.

Theorem 16. *Let Φ be a set of labeled sentences, then the clause normal form* $\mathbf{CNF}(\Phi)$ *is satisfiable iff Φ is.*

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 we have $\mathcal{I}_\varphi(L) = \alpha$ for all well-sorted assignments φ into $\mathcal{M}$. With this notion, the assertion can be reduced to the

[7] A computation using only the rules from Definition 13 would have yielded 4 clauses in the succedent of the rule shown above.

standard argumentation about Skolemization and a tedious calculation with the truth tables from Definition 5. □

Now we proceed to give a simple resolution calculus, which utilizes standard (unsorted) unification. In [KK96b], 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 17 (Restriction Constraints).
The **restriction constraint** of a substitution $[t/x_S^*]$, depends on the mode $*$.

$$\begin{aligned}
\mathcal{RC}([t/x_S^+]) &:= \{([t/x^+]S)^{\mathtt{t}^-\mathtt{f}^+\mathtt{f}^-\mathtt{u}}, \mathfrak{S}(t)^{\mathtt{f}^+}\},\\
\mathcal{RC}([t/x_S^-]) &:= \{([t/x^-]S)^{\mathtt{t}^+\mathtt{f}^+\mathtt{f}^-\mathtt{u}}, \mathfrak{S}(t)^{\mathtt{t}^+}\}, \text{ and}\\
\mathcal{RC}([t/x_S^\pm]) &:= \{([t/x^\pm]S)^{\mathtt{f}^+\mathtt{f}^-\mathtt{u}}\}
\end{aligned}$$

For a substitution $\sigma = \sigma', [t/x_S^*]$ we have $\mathcal{RC}(\sigma) := \mathcal{RC}(\sigma') \cup \mathcal{RC}[t/x_S^*]$.

These clauses are residuated in the $\mathcal{RPL}$ rules and have to be refuted in order to guarantee that the restriction $\mathcal{A}_\varphi^*(S,t)$ holds (cf. Definition 5) for every instance t instantiated for a free variable x_S^*.

Definition 18 (Resolution Inference Rules ($\mathcal{RPL}$)).

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

$$\frac{\mathfrak{D}(t)^{\mathtt{f}^+}, C \qquad L^\gamma, D}{\rho(C), \rho(D), \mathcal{RC}(\rho)}\,Strict^{\mathfrak{D}} \qquad \frac{\mathfrak{S}(t)^{\mathtt{f}^+}, C \qquad L^\delta, D}{\rho(C), \rho(D), \mathcal{RC}(\rho)}\,Strict^{\mathfrak{S}}$$

where $\alpha \neq \beta$ and $\gamma \neq \mathtt{u}$, and $\delta \in \{\mathtt{t}^+, \mathtt{f}^+\}$. For *Res* and *Fac*, the substitution σ is the most general unifier of L and M; and for $Strict^{\mathfrak{D}}$ and $Strict^{\mathfrak{S}}$ there exists a subterm s of L, such that ρ is a most general unifier of t and s.

Here we have assumed $\alpha, \beta, \gamma, \delta$ to be single truth values, naturally the rules can be easily extended to sets of truth values. For instance, the resolution of L^α, C and M^β, D would lead to the clause $\sigma(L)^{\alpha\cap\beta}, \sigma(C), \sigma(D), \mathcal{RC}(\sigma)$.

Remark 19. Note that clauses containing $A^{\mathtt{t}^+\mathtt{f}^+\mathtt{t}^-\mathtt{f}^-\mathtt{u}}$ 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 20. Let A be a sentence and Φ be the clause normal form of the set $\{\{A^{\mathtt{f}^+\mathtt{t}^-\mathtt{f}^-\mathtt{u}}\}\}$ then we say that A can be **proved in** $\mathcal{RPL}$ ($\vdash A$) iff there is a derivation of the empty clause □ from Φ with the inference rules above.

Theorem 21 (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 literals $([t/x]S)^\alpha$ (with some truth value set α) per component of the substitution, which only can be refuted if indeed $([t/x]S)^\beta$ with $\alpha \cap \beta = \emptyset$.

The soundness of the rules $Strict^{\mathfrak{S}}$ and $Strict^{\mathfrak{D}}$ hinges on the fact that an atom must be insecure (undefined), iff it has an insecure (undefined) subterm, since we have assumed strictness of functions and predicates. □

Definition 22. 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 [Wei91] is independent of the number of truth values.

Lemma 23. *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 24 (Herbrand Model). 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 $\{\sigma{\downarrow}(C) \mid C \in \mathbf{CNF}(A), \sigma \text{ ground}, \mathbf{Dom}(\sigma) = \mathbf{Free}(C)\}$

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 $\mathcal{H}(\Phi) \longrightarrow \mathcal{B}$. The **$\Sigma$-Herbrand model $\mathcal{H}$ for Φ and** ν is the set $\mathcal{H} := \{L^\alpha \mid \alpha = \nu(L), L \in \mathcal{H}(\Phi)\}$. Note that $\mathcal{H}$ cannot contain **complementary** literals (i.e., two literals L^α and L^β, where $\alpha \neq \beta$) since ν is a function.

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.

The following theorem which links Herbrand satisfiability to satisfiability in Σ-models is one key ingredient to all completeness proofs in $\mathcal{PL}$, therefore we will give the proof in full detail.

Theorem 25 (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 $\Phi := \mathbf{CNF}(A)$ and $\Psi := \mathbf{CNF}_H(A)$. We will show that the set $\mathcal{H} := \{L^\alpha \mid L \in \mathcal{H}(\Phi), \alpha = \mathcal{I}_\emptyset(L)\}$ is a Σ-Herbrand model for Ψ. For this we assume the opposite, that is, there is a clause $C \in \Psi$, such that $\mathcal{H} \cap C = \emptyset$, this is sufficient, since C is ground and therefore $\sigma{\downarrow}(C) = C$. Since $C \in \Psi$ there is a ground substitution $\sigma = [t^i/x^{*_i}_{S_i}]$ and a clause $D \in \Phi$, such that $C = \sigma{\downarrow}(D) = \sigma(D) \cup \mathcal{RC}(\sigma)$ by definition.

Without loss of generality we can assume that the assignment $\psi := [\mathcal{I}_\emptyset(t^i)/x^i]$ is well-sorted, i.e. $\psi(x^*_S) \in \mathcal{A}^*_\varphi(S, x) = \{a \in \mathcal{A} \mid \mathcal{I}_{\psi,[a/x]}(S) = *\}$ or equivalently $\mathcal{I}_\psi(S) = *$, since otherwise $\mathcal{I}_\psi(S) \in \mathcal{B} \setminus *$, and therefore $([t^i/x_i]S^i)^\gamma \in \mathcal{H}$ for some $\gamma \in \mathcal{B} \setminus *$, which contradicts the assumption that $\mathcal{H} \cap C = \emptyset$, since $([t^i/x_i]S^i)^\gamma \in \mathcal{RC}(\sigma)$ by construction.

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 $\mathcal{HB}$ the Herbrand base for $\mathcal{H}$ together with $\bot$. In order to construct the carrier $\mathcal{A}$ or the intended model, we have to identify all elements in $\mathcal{HB}$ that are undefined $((t{<}\mathfrak{D})^{\mathfrak{f}^+} \in \mathcal{H})$ and identify them with $\bot$. Traditional construction of the model define the meanings $\mathcal{I}(f^n)$ and $\mathcal{I}(P^n)$ be of functions and predicates, such that

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

However, this definition does not make $\mathcal{I}(f)$ strict for $\mathfrak{D}$, since $\mathcal{I}(f)(\bot) = f(\bot) \neq \bot$. To repair this defect we take the carrier $\mathcal{A}$ to be the quotient of $\mathcal{T}_\bot$ with respect to the equality theory $=_\bot$ induced by the set

$$E_\bot = \{t =_\bot \bot \mid (t{<}\mathfrak{D})^{\mathfrak{f}} \in \mathcal{H}\} \cup \{f^k(x_1, \ldots, \bot, \ldots, x_k) = \bot \mid f^k \in \Sigma^k\}$$

of equations. Thus $\mathcal{A}$ is the set of equivalence classes $[\![t]\!]_\bot = \{s \mid E_\bot \models s =_\bot t\}$. The function $f_\bot\colon ([\![t_1]\!]_\bot, \ldots, [\![t_1]\!]_\bot) \mapsto [\![f(t_1, \ldots, t_n)]\!]_\bot$ is a well-defined function, since $=_\bot$ is a congruence relation. We define $\mathcal{I}(f) := f_\bot$ and note that the special construction of $E_\bot$ entails the strictness of $f_\bot$ for $\mathfrak{D}$. Strictness for $\mathfrak{S}$ is not a problem in this construction, since it does not collapse the meanings of terms.

For $P \in \mathcal{P}^n$ let $P_\mathcal{H}([\![t^1]\!]_\bot, \ldots, [\![t^n]\!]_\bot) = \alpha$ iff $P(t^1, \ldots, t^n)^\alpha \in \mathcal{H}$. Clearly $P_\mathcal{H}$ is well-defined, since the definition only depends on $=_\bot$-equivalence classes, furthermore it is strict for $\mathfrak{D}$ and $\mathfrak{S}$ by construction.

Note that any assignment into $\mathcal{M}$ has the form $[[\![t_i]\!]/x_i]$, where the t_i are ground terms, so the sets of substitutions and that of assignments are isomorphic (we will neglect to make this isomorphism explicit in the following). Moreover, a simple structural induction shows that $\mathcal{I}_\varphi(t) = [\![\varphi(t)]\!]$. In fact, the argument can be extended to show that the isomorphism respects well-sortedness.

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. So we have $\varphi(S_i)^\gamma \in \mathcal{H}$ for some $\gamma \in *_i$. $\mathcal{H}$ 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. $\square$

Theorem 26 (Refutation Completeness). *$\mathcal{RPL}$ is refutation complete, i.e. if Φ is an unsatisfiable set of ground clauses then there exists a $\mathcal{RPL}$ derivation of the empty clause from Φ.*

Proof sketch: The proof of completeness for unification-based refutation calculi is usually in two parts. First completeness for ground clauses is established, and then the result is generalized with a lifting argument. It turns out that the lifting property can be established by methods from [Wei91], since they are independent of the number of truth values.

Thus we only have to establish ground completeness. We do this using the so-called k-parameter induction method. For a clause set Φ let $k(\Phi)$ be the number of literals minus the number of clauses in Φ. If $k = 0$, then Φ contains two complementary literals, and is unsatisfiable by the Herbrand theorem above, since Φ is ground. If $k > 0$, then there is a non-unit clause $C \in \Phi$, which we can divide into two subclauses D and E. By inductive hypothesis there is a $\mathcal{RPL}$-refutation $\mathcal{D}_D$ of $\Phi_D = \Phi \setminus C \cup \{D\}$. The refutation of Φ_E can be used to construct a $\mathcal{RPL}$-derivation $\mathcal{D}_D^E$ of the clause E out of Φ (the rest E of D in C is conserved during the proof). Again by inductive hypothesis, there is a $\mathcal{RPL}$ derivation and $\mathcal{D}_E$ of of Φ_E. Thus the combined derivation $\mathcal{D}_D^E, \mathcal{D}_E$ is a refutation of Φ and we have completed the proof for the ground case. □

Corollary 27 (Completeness). *Any valid $\mathcal{PL}$ formula can be proved by $\mathcal{PL}$.*

Proof sketch: Let A be a valid formula, then the labelled formula $A^{\mathtt{f}^+\mathtt{t}^-\mathtt{f}^-\mathtt{u}}$ must be unsatisfiable. By Theorem 16, $\Phi := \mathbf{CNF}(\{\{A^{\mathtt{f}^+\mathtt{t}^-\mathtt{f}^-\mathtt{u}}\}\})$ must be unsatisfiable as well, so by the refutation completeness result above, there is a refutation of Φ, which is a proof of A by definition. □

4 Example: Quantificational Presuppositions

Let us assume the following information. There is a company `TheCompany` which wants to fire people, but they have a social touch and do not fire any persons with 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:

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

T $\neg\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John})$

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

A $(\forall x_{\mathfrak{D}(x)}.\ \mathtt{fires}(\mathtt{TheCompany}, x) \rightarrow \neg\mathtt{parent}(x))^{\mathtt{t}^+}$

B $(\forall x_{\mathfrak{D}(x)}.\ \exists y_{\mathfrak{D}(y)}.\ \mathtt{child}(\mathtt{y},\mathtt{x}) \rightarrow \mathtt{parent}(\mathtt{x}))^{\mathtt{t}^+}$

C $(\forall x_{\mathtt{child}(x,\mathtt{John})}.\ \mathtt{sleeps}(x))^{\mathtt{t}^+}$

T $(\neg\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{t}^-\mathtt{f}^+\mathtt{f}^-\mathtt{u}}$

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

A1 $(\mathtt{fires}(\mathtt{TheCompany}, x^+_{\mathfrak{D}}))^{\mathtt{f}^+}, (\mathtt{parent}(x^+_{\mathfrak{D}}))^{\mathtt{f}^+}$

A2 $\mathfrak{D}(c_1)^{\mathtt{t}^+}$ A3 $\mathfrak{S}(c_1)^{\mathtt{t}^+}$

B1 $(\mathtt{child}(y^+_{\mathfrak{D}}, x^+_{\mathfrak{D}}))^{\mathtt{f}^+}, (\mathtt{parent}(x^+_{\mathfrak{D}}))^{\mathtt{t}^+}$

B2 $\mathfrak{D}(c_2)^{\mathtt{t}^+}$ B3 $\mathfrak{S}(c_2)^{\mathtt{t}^+}$

B4 $\mathfrak{D}(c_3)^{\mathtt{t}^+}$ B5 $\mathfrak{S}(c_3)^{\mathtt{t}^+}$

C1 $(\mathtt{sleeps}(x^+_{\mathtt{child}(x,\mathtt{John})}))^{\mathtt{t}^+}$

C2 $(\mathtt{child}(c_4, \mathtt{John}))^{\mathtt{t}^+}$

C3 $\mathfrak{D}(c_4)^{\mathtt{t}^+}$ C4 $\mathfrak{S}(c_4)^{\mathtt{t}^+}$

T $(\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{f}^-}, (\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{t}^+}$, $(\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{t}^-}, (\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{u}}$

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

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

Two-times resolving with clauses from the term declarations and resolving with C3 and C4 results in:

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

which in turn can be resolved with A1:

R3 $(\mathtt{fires}(\mathtt{TheCompany}, \mathtt{John}))^{\mathtt{f}^+}, (\mathfrak{D}(\mathtt{John}))^{\mathtt{t}^-\mathtt{f}^+\mathtt{f}^-\mathtt{u}}, (\mathfrak{S}(\mathtt{John}))^{\mathtt{f}^+}$

The last two literals can again be resolved away using the term declarations. T can be resolved four 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 translation of the above text into FOL, the essential information in C2 that John has a child cannot be derived and hence no proof can be found.

5 Example: Existential Presupposition

In this example we show how we can *avoid* the presupposition in classical logic that all constants 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 properties of God, supposed he existed, 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.

Let us look at a simplification of Leibniz's argument that God cannot be almighty, since if he were then there could not be any evil in the world. The argument is as follows: *Assume God existed (1) and God were good (2). Furthermore we assume that everything that is created by an almighty and good being must be good (3). Since God has created the world (4) and the world is not good at all (5), God cannot be almighty (T).*

In our logic this can be formalized in a way that can be accepted by a theist and an atheist alike:
Term declarations:

(1) $\mathfrak{S}(\mathtt{God})^{\mathtt{t}^+\mathtt{f}^+}$ $\mathfrak{D}(\mathtt{God})^{\mathtt{t}^+}$ $\mathfrak{S}(\mathtt{World})^{\mathtt{t}^+}$ $\mathfrak{D}(\mathtt{World})^{\mathtt{t}^+}$

Propositions:

(2) $\mathbf{T}(\mathtt{good}(\mathtt{God}))$
(3) $\mathbf{T}(\forall x_{\mathfrak{D}(x)}.\, \forall y_{\mathfrak{D}(y)}.\, \mathtt{almty}(x) \wedge \mathtt{good}(x) \wedge \mathtt{crtd}(x, y) \to \mathtt{good}(y))$
(4) $\mathbf{T}(\mathtt{crtd}(\mathtt{God}, \mathtt{World}))$
(5) $\neg(\mathtt{good}(\mathtt{World}))$

Theorem:

(T) $\mathbf{T}\neg\mathtt{almty}(\mathtt{God})$

Note that most of the statements are prefixed by the $\mathbf{T}$ connective, what essentially says that the security value is of no concern. A clause normal form of the assumptions and the negated theorem would look like (using combined transformations as discussed in Remark 14):

(2) $(\mathtt{good}(\mathtt{God}))^{\mathtt{t}^+\mathtt{t}^-}$
(3) $(\mathtt{almty}(x^{\pm}_{\mathfrak{D}}))^{\mathtt{f}^+\mathtt{f}^-}, (\mathtt{good}(x^{\pm}_{\mathfrak{D}}))^{\mathtt{f}^+\mathtt{f}^-}, (\mathtt{crtd}(x^{\pm}_{\mathfrak{D}}, y^{\pm}_{\mathfrak{D}}))^{\mathtt{f}^+\mathtt{f}^-}, (\mathtt{good}(y^{\pm}_{\mathfrak{D}}))^{\mathtt{t}^+\mathtt{t}^-}$
(4) $(\mathtt{crtd}(\mathtt{God}, \mathtt{World}))^{\mathtt{t}^+\mathtt{t}^-}$
(5) $(\mathtt{good}(\mathtt{World}))^{\mathtt{f}^+}$
(¬T) $(\mathtt{almty}(\mathtt{God}))^{\mathtt{t}^+\mathtt{t}^-}$

It is now possible to resolve clause (3) with the unit clauses (2), (4), (5), and (¬T). Because of the term declarations $\mathfrak{D}(\mathtt{God})^{\mathtt{t}^+}$ and $\mathfrak{D}(\mathtt{World})^{\mathtt{t}^+}$, the generated restriction constraints for the variables $x^{\pm}_{\mathfrak{D}}$ and $y^{\pm}_{\mathfrak{D}}$ can be resolved away too, so that we finally can derive the empty clause.

6 Conclusion

We have developed a five-valued logic for the formalisation of everyday reasoning with presuppositions and undefinedness. Our logic is a common generalization of the system proposed by Bergmann in [Ber81] for dealing with presuppositions and the one proposed by Kleene [Kle52] for dealing with undefinedness.

Furthermore we have presented a sound and complete resolution calculus for our system, which uses the sort mechanism to capture bounded quantifications. Our calculus can be seen as an extension of classical logic that combines methods from many-valued logics (cf. [BF92,Häh94]) for a correct treatment of the secure and insecure or undefinedness information and order-sorted logics (see [Wei91]) for an adequate treatment of restricted domains. In contrast to the mere partial function calculi in [KK94,KK96b], from which we have drawn most of the technical ideas, in this paper we strictly distinguish between undefined objects which are all identified, and insecure objects, which can be different from each other. However, just like in the mere Kleene case, most definedness preconditions can be taken care of in the unification, making inferring quite efficient.

Although certainly the phenomenon of presuppositions cannot be fully captured by a static analysis and in fact research on presuppositions in linguistics has nowadays turned mostly to dynamic and more pragmatically driven analyses, and away from the multi-valued treatment, this is not a counter-argument to our system. In contrast to FOL our system $\mathcal{PL}$ makes it possible to specify (and reason with) presuppositions and undefinedness, so that once the dynamic logics underlying the linguistic analyses can be used for reasoning, techniques like those developed in this paper will be indispensable.

References

[Ber81] Merrie Bergmann. Presupposition and two-dimensional logic. *Journal of Philosophical Logic*, **10**:27–53, 1981.

[Bea95] David Beaver. *Presupposition and Assertion in Dynamic Semantics.* PhD thesis, University of Edinburgh, 1995.

[BC81] J. Barwise and R. Cooper Generalized quantifiers and natural language *Linguistics and Philosophy*, **4**:159–219, 1981.

[BF92] Matthias Baaz and Christian G. Fermüller. Resolution for many-valued logics. In A. Voronkov, editor, *Proceedings of LPAR*, pages 107–118, St. Petersburg, Russia, 1992. Springer Verlag, LNAI 624.

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

[Coh87] Anthony G. Cohn. A more expressive formulation of many sorted logics. *Journal of Automated Reasoning*, **3**:113–200, 1987.

[vF66] Bas C. van Fraassen. Singular terms, truth-value gaps, and free logic. *The Journal of Philosophy*, LXIII(17):481–495, 1966.

[Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen I & II. *Mathematische Zeitschrift*, **39**:176–210, 572–595, 1935.

[Häh94] Reiner Hähnle. *Automated Deduction in Multiple-Valued Logics.* Oxford University Press, 1994.

[Her73] H. Herzberger. Dimensions of truth. *Journal of Philosophical Logic*, **2**:335–356, 1973.

[KK94] Manfred Kerber and Michael Kohlhase. A mechanization of strong Kleene logic for partial functions. In Alan Bundy, editor, *Proceedings of the 12th CADE*, pages 371–385, Nancy, France, 1994. Springer Verlag, LNAI 814.

[KK96a] Manfred Kerber and Michael Kohlhase. A resolution calculus for presuppositions. In Wolfgang Wahlster, editor, *Proceedings of the 12th ECAI*, pages 375–379, Budapest, Hungary, August 1996. John Wiley & Sons.

[KK96b] Manfred Kerber and Michael Kohlhase. A tableau calculus for partial functions. *Collegium Logicum – Annals of the Kurt-Gödel-Society*, **2**:21–49, 1996.

[KK97] Manfred Kerber and Michael Kohlhase. Mechanising Partiality without Re-Implementation. In Gerhard Brewka, Christopher Habel, and Bernhard Nebel, editors, *Proceedings of the 21st Annual German Conference on Artificial Intelligence, KI'97*, pages 123–134, Freiburg, Germany 1997. Springer Verlag, LNAI 1303.

[Kle52] Stephen Cole Kleene. *Introduction to Metamathematics.* Van Nostrand, 1952.

[KR93] H. Kamp and U. Reyle. *From Discourse to Logic.* Kluwer, 1993.

[Kra95] Emiel Jacques Krahmer. *Discourse and Presupposition.* PhD thesis, Institute for Language Technology & Artificial Intelligence, Tilburg University, Tilburg, The Netherlands, 1995.

[McC90] William McCune. Otter 2.0. In Mark E. Stickel, editor, *Proceedings of the 10th CADE*, pages 663–664, Kaiserslautern, Germany, 1990. Springer Verlag, LNAI 449.

[Mos57] Andrzej Mostowski. On a generalization of quantifiers. *Fundamenta Mathematicae*, **44**:12–36, 1957.

[Rob65] John Alan Robinson. A machine oriented logic based on the resolution principle. *Journal of the ACM*, **12**:23–41, 1965.

[Rus05] Bertrand Russell. On denoting. *Mind*, **14**:479–493, 1905.

[Sau93] W. Saurer. A natural deduction system for discourse representation theory. *Journal of Philosophical Logic*, **22**, 1993.

[Str73] P. F. Strawson. On referring. In János S. Petöfi and Dorothea Franck, editors, *Presuppositions in Philosophy and Linguistics*, pages 193–220. Athenäum Verlag, 1973.

[vdS92] R. van der Sandt. Presupposition projection as anaphora resolution. *Journal of Semantics*, **9**(4), 1992.

[Wei91] 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.

[WGR96] Christoph Weidenbach, Bernd Gaede, and Georg Rock. SPASS & FLOTTER, Version 0.42. In M.A. McRobbie and J.K. Slaney, editors, *Automated Deduction—CADE 13, Proceedings*, pages 141–145. Springer-Verlag, LNAI 1104, 1996.