



# When iota meets lambda

Andrzej Indrzejczak<sup>1</sup> · Michał Zawidzki<sup>1,2</sup>

Received: 19 April 2022 / Accepted: 13 January 2023 / Published online: 15 February 2023  
© The Author(s) 2023

## Abstract

Definite descriptions are widely discussed in linguistics and formal semantics, but their formal treatment in logic is surprisingly modest. In this article we present a sound, complete, and cut-free tableau calculus  $\mathbf{TC}_{R_\lambda}$  for the logic  $\mathbf{L}_{R_\lambda}$  being a formalisation of a Russell-style theory of definite descriptions with the iota-operator used to construct definite descriptions, the lambda-operator forming predicate-abstracts, and definite descriptions as genuine terms with a restricted right of residence. We show that in this setting we are able to overcome problems typical of Russell's original theory, such as scoping difficulties or undesired inconsistencies. We prove the Craig interpolation property for the proposed theory, which, through the Beth definability property, allows us to check whether an individual constant from a signature has a definite description-counterpart under a given theory.

**Keywords** Definite descriptions · Lambda-abstraction · Analytic tableaux · Completeness · Craig interpolation

## 1 Introduction

In natural languages naming expressions are used not only, and even not always, to refer to objects. They are also important information-conveying ingredients of every discourse. One of the most important class of such expressions are *definite descriptions*, expressions of the form ‘the so and so’, such as ‘the author of «On Denoting»’. Definite descriptions are ubiquitous not only in natural languages but also in mathematics and

---

A. Indrzejczak and M. Zawidzki have contributed equally to this work.

---

✉ Michał Zawidzki  
michal.zawidzki@cs.ox.ac.uk

Andrzej Indrzejczak  
andrzej.indrzejczak@filhist.uni.lodz.pl

<sup>1</sup> Department of Logic, University of Lodz, Lindleya 3/5, 90-131 Łódź, Poland

<sup>2</sup> Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, UK

science ('The sum of 7 and 5'). In artificial languages it is the iota-operator,  $\iota$ , due to Peano, that is usually applied to formalise definite descriptions: 'the  $F$ ' is represented formally as  $\iota x F$ .

The investigation of definite descriptions since Russell's paper "On Denoting" (1905) occupies a central place in logical, semantical, and linguistic research and many deep and detailed studies have been carried out ever since. The second half of the 20<sup>th</sup> century saw the development of new approaches to this phenomenon based on non-classical logics, in particular free logic in which, contrary to Frege's and Russell's assumptions, characteristic for classical logic, it is not assumed that every term refers. Such solutions found a place in the application to intensional logics where the distinctions between terms which denote existent or non-existent objects, in a rigid or non-rigid way, or simply are non-denoting, are crucial. Yet despite the long history of research into definite descriptions and a variety of solutions to the problems they pose, we can hardly say that any theory of definite descriptions is obvious or commonly accepted. In some approaches definite descriptions are not even treated as naming expressions, but rather as binary quantifiers (Neale, 1990; Kürbis, 2019a, 2021a) or predicative expressions (Graff, 2001).

In modern approaches, the category of *singular terms* is usually limited to proper names and functional expressions formed from the former. As a consequence, proof-theoretic tools are badly prepared to deal with reasoning tasks in natural languages, where subsentential parts, including several complex naming expressions, play a privileged role. The inclusion of a wider category of terms was characteristic for the earlier stage of the development of mathematical logic in the works of Frege (1893), Hilbert and Bernays (1968), Bernays (1958), Carnap (1947), Quine (1982), and Rosser (1978). Later, the restriction to functional terms became a norm, especially in automated deduction. In particular, definite descriptions, despite their importance in logic and formal semantics, are largely ignored. Possibly, the fact that modern treatments of theories of definite descriptions are on the market and it is not always evident which of them is fit for purpose.

Since the role of definite descriptions in the field of proof theory and automated deduction has so far been underestimated, it is important to stress some advantages that using them may bring. First of all, definite descriptions are more expressive than functional terms:  $f^n(t_1, \dots, t_n)$  can be represented as  $\iota x F^n(t_1, \dots, t_n, x)$ . Getting rid of functional terms by means of definite description is also in the spirit of Russell, who, for philosophical reasons, did not introduce the former to his logic. On the other hand, not every (proper) description can be expressed using functional terms. Consider descriptions like 'the winner of the ultimate fight', 'the bear we have seen recently'. It is not at all clear which parts of these expressions could be treated as functors and which as arguments. Moreover, in the case of using definite descriptions instead of functional terms we do not need extra bridge principles showing how the information encoded by the latter is represented by predicates (e.g.,  $f(x) = y \leftrightarrow F(x, y)$ ). In languages with functional terms such bridge principles are usually necessary as enthymematic premises in an analysis of obviously valid arguments. If we use definite descriptions, such bridge principles become superfluous. Moreover, the presence of functions in formal languages often easily leads to generating infinite Herbrand domains even

when finite domains are allowed. Let us illustrate this with a simple example. From  $\forall x(a = f(x))$  we infer  $a = f(a)$ ,  $a = f(f(a))$ ,  $a = f(f(f(a)))$ ,  $\dots$ . On the other hand, from the corresponding  $\forall x(a = \iota y F(x, y))$  we can obtain a single-element model satisfying  $a = a$ ,  $F(a, a)$ , e.g., by means of tableau rules proposed by Indrzejczak and Zawidzki (2021). Additionally, definite descriptions can be used to provide smooth definitions of new terms, and even new operators, in formal languages. For example, one may define the abstraction operator in set theory in an elegant way (Tennant, 2004). Finally, it is also noteworthy that definite descriptions have recently found applications in other areas such as verification of hybrid systems (Bohrer et al., 2019), formal ontology (Oppenheimer & Zalta, 2011; Blumson, 2020), or knowledge representation (Artale et al., 2021).

These virtues of definite descriptions have not hitherto been thoroughly examined mainly because of a lack of good formal systems expressing their theories. An enormous number of books and papers have been devoted to offering solutions to logical and linguistic problems connected with definite descriptions, but the number of formal systems and their studies is relatively modest. Moreover, most approaches to the formalisation of theories of definite descriptions follow the axiomatic approach. Only comparatively few natural deduction (ND), tableau systems, or sequent calculi (SC) have been formulated. One may mention here ND systems of: Kalish and Montague (1957), Kalish et al. (1980), Stenlund (1973, 1975), Tennant (1978, 2004), Garson (2006), Carlström (2005), Francez and Więckowski (2014), and Kürbis (2019a, b). But only a few of them (namely Tennant's and Kürbis' works) deal with definite descriptions by means of rules which allow for finer proof analysis and provide normalisation proofs. SC systems were proposed by Czermak (1974), Gratzl (2015), Orlandelli (2021), Kürbis (2021a, b, c), and Indrzejczak (2018, 2019, 2020a, b, 2021b). There are also a few tableau calculi, due to Bencivenga et al. (1986), Gumb (2001), Bostock (1997), Fitting and Mendelsohn (1998), and Indrzejczak and Zawidzki (2021). However, with the exception of the last one, these systems introduce definite descriptions by means of rather complex rules, which makes them quite difficult to maneuver.

Given the importance of definite descriptions in formal semantics and of deductive systems such as natural deduction and sequent or tableau calculi in formal logic, it is rather surprising that so little attention has been paid to what each can contribute to the other. In this paper we focus on the problem of decent formalisation of Russell's approach to definite descriptions which initiated the whole line of research in formal logic. However, the semantics and syntactical formalisation we offer departs from the original one, which helps us avoid some recognised drawbacks of the latter while saving its essential features, and enables its application to automated deduction.

In Sect. 2 we briefly characterise the Russellian approach to definite descriptions, then, in Sect. 3, we explain in what way the introduction of predicate-abstracts can help in avoiding some of the known pitfalls of Russell's original account. Section 4 introduces the syntax and semantics of the considered logic, whereas in Sect. 5 we present a suitable cut-free tableau calculus. Soundness and completeness of the calculus is proved in Sect. 6. Section 7 contains a proof of the interpolation theorem and shows how, through the Beth definability property, it can be checked whether an individual constant from a given signature can be replaced by an appropriate definite description under a considered theory. The paper is briefly concluded in Sect. 8.

## 2 The Russellian approach to definite descriptions

Almost 120 years have passed since Bertrand Russell published his groundbreaking article “On denoting” (1905). It deals with many semantical questions of a general character, but it became famous mainly for the considerations on the problem of definite descriptions. Russell’s work stands, with the works of Frege, at the beginning of the development of modern mathematical logic, and his theory of definite descriptions, presented first in the cited paper and elaborated in “Principia Mathematica” (Whitehead & Russell, 1910) and in Russell’s later work (1920), plays a central role in the foundations of logic and formal semantics. Although Russell’s theory of definite descriptions was not the first one (pride of place for the first formalisation of definite descriptions in logic belongs to Frege (1893, § 11, § 18)), it gave an enormous impact on all further studies in this area. Despite of its weak and controversial points, widely criticised (see, e.g., Strawson (1950), Donnellan (1966), Kripke (1977)), Russell’s theory has become a standard point of reference of almost all works devoted to the analysis of definite descriptions, even those which are very critical towards its applicability to definite descriptions in natural language (see, e.g., Ludlow (2021)). Moreover, it is still widely accepted by formal logicians as a proper way of handling descriptions; the scores of textbooks that use it as their official theory of definite descriptions count as witnesses for this claim. Russell’s theory has also strong affinities to logics closely connected with applications in constructive mathematics and computer science like the logic of the existence predicate by Scott (1979) or the logic of partial terms by Beeson (1985), also called the basic system of definedness logics. The latter is treated as a wider family of systems specialised to deal with partial untyped combinatorics and lambda calculi (Feferman, 1995).

Russell was prompted by reflections on improper definite descriptions: what do sentences like ‘The present King of France is bald’ mean? His approach to definite descriptions (shortly RDD) was developed in opposition to Meinong’s and Frege’s views. Both Frege and Russell assumed that terms must denote but reacted in different ways to the failure of denotation (the phenomenon of *improper* descriptions). Frege considered four different attempts, analysed by Pelletier and Linsky (2009), and in general he urged that all definite descriptions must have designates, even if they are to be arbitrary. On the other hand, Russell refused definite descriptions the status of terms and left variables as the only terms of the formal language of mathematics. In contrast to Frege’s view, he treated descriptions as a kind of incomplete signs and showed how to get rid of them by means of contextual definitions of the form:

$$\psi(\iota x\varphi(x)) := \exists x(\forall y(\varphi(y) \leftrightarrow y = x) \wedge \psi(x)). \quad (R)$$

A characteristic feature of Russell’s approach is that no meaning is assigned to the expression ‘the  $F$ ’ standing alone, but only in the context of complete sentences in which it occurs. ‘The  $F$  is  $G$ ’ means ‘There is exactly one  $F$  and it is  $G$ ’. Thus the complex term ‘The  $F$ ’ disappears upon logical analysis. Other distinctive properties of RDD, which mainly follow from Russell’s policy of omitting singular terms from the primitive expressions of a formal language altogether, are:

- (1) a fixed truth-value (falsity) of all atomic formulas with non-denoting terms;

- (2) quantification rules of universal instantiation and existential generalisation restricted to variables;  
 (3) identity with reflexivity restricted to variables.

In fact the most important point is (1), as (2) and (3) are its consequences. According to (2), when the universal quantifier is eliminated, we cannot unrestrictedly substitute the quantified variable with an arbitrary term satisfying the conditions of proper substitution. For instance, for a non-denoting term we can deduce a formula which is false because of (1). Therefore the rule of universal elimination (or existential generalisation) must be restricted to terms known in advance to be denoting, that is, to variables as instantiated terms. Similarly in (3): if  $t$  is non-denoting, then even  $t = t$  is false according to (1). Therefore, reflexivity must be restricted to variables as well. Summing up: these three desiderata constitute a logic which is currently identified as a kind of *negative free logic* rather than classical logic.

Nevertheless, Russell's account has also serious formal drawbacks. As it stands, (R) must be restricted to atomic  $\psi$  or it is necessary to add means for marking scope distinctions. For example, let  $\psi$  be  $\neg G$ , then the result is ambiguous between the internal and external negation of  $\psi(\iota x\varphi(x))$ , that is, between 'It is not the case that the F is G' and 'The F is not G'. If we still admit complex  $\psi$  in (R), the situation is even worse. If we use a valid  $\psi$  and an inconsistent  $\varphi$ , we run into a contradiction. For example:

$$\begin{aligned} [A(\iota y(B(y) \wedge \neg B(y))) \rightarrow A(\iota y(B(y) \wedge \neg B(y)))] \\ \Leftrightarrow \exists x[\forall y((B(y) \wedge \neg B(y)) \Leftrightarrow y = x) \wedge (A(y) \rightarrow A(y))] \end{aligned}$$

leads to  $B(a) \wedge \neg B(a)$  (modulo constant renaming).

To avoid the problems, Whitehead and Russell introduced a rather clumsy method for drawing scope distinctions that is far from ideal, which is demonstrated by the fact that almost no one uses it. Instead, a restricted binary quantifier is used to overcome scope difficulties (Bostock, 1997; Kürbis, 2019a, 2021a). One is even tempted to conjecture that a reason for the lack of proof-theoretic investigations into definite descriptions is a perceived tension between the need for drawing scope distinctions in the symbolism and the difficulties encountered in doing so.<sup>1</sup>

Even without scoping problems, the Russellian policy of reducing the category of terms only to variables seems to be inadequate, or at least problematic and artificial, with respect to processing reasoning tasks in natural languages. In fact, there is a tension between the theoretical treatment of definite descriptions by Russell and their practical applications. Whitehead and Russell (1910) formulate several theorems with definite descriptions occurring as terms. Also Kalish et al. (1980), as well as Pelletier and Linsky (2009) provide lists of theorems and rules correct in RDD and containing occurrences of definite descriptions. This shows that it is possible to save some features of the original Russellian approach to definite descriptions, yet to treat them as genuine terms. Henceforth, by RDD we mean a theory where all the abovementioned features of the Russellian approach are saved except that definite descriptions are treated as

<sup>1</sup> For some examples of how Russell's scope distinctions can be put into practice, and what difficulties this may cause, see Gratzl (2015).

terms. In this case ( $R$ ) is replaced with an object-language biconditional which can serve as a characteristic axiom of a logic. Such a logic is essentially equivalent to Feferman's definedness logic, but expressed in a language in which the definedness predicate is eliminated. This equivalence was proved by Indrzejczak (2021a). In such a form RDD was formalised as ND by Kalish et al. (1980) and recently by Francez and Więckowski (2014). However, all these formalisations involve complex rules or axioms. Recently Indrzejczak (2021b) provided an analytic and cut-free SC equivalent to RDD as formalised by Kalish et al. (1980). In all these systems ( $R$ ) is restricted to atomic predicates and special rules are required to evaluate all formulas with improper definite descriptions as false. The necessity of such a restriction of ( $R$ ) can be seen as another pitfall of RDD, in particular in the context of natural language and its formalisations. Bostock rightly noticed that the qualification of some predicates as atomic is to some extent arbitrary, though reflected in signatures of non-logical symbols for a given logic. Can we offer any improvement to the state of the art? Below we introduce a variant of RDD based on the use of predicate abstracts, which avoids the discussed shortcomings while saving all plausible features of the Russellian approach.

### 3 Enter lambda

To sum up: despite of the linguistic and philosophical problems with the Russellian approach to definite descriptions, which have enormous literature devoted to it, we have also some significant formal problems described above. It seems that we can avoid the latter problems if we enrich the language with the lambda-operator and restrict the range of what can be predicated of definite descriptions to predicate abstracts of the form  $\lambda x\varphi$ , where  $\varphi$  is any formula. Thus,  $\lambda x\varphi$  is a unary predicate and  $(\lambda x\varphi)(t)$ , where  $t$  is a term, is a formula called a *lambda-atom*. Accordingly, atomic formulas in the strict sense are built from predicate symbols and individual variables or constants only; definite descriptions can only be attached as arguments to lambda-atoms.

Predicate abstracts built by means of the lambda-operator were first proposed in the context of studies on modal predicate logic by Stalnaker and Thomason (1968) and then the idea was further developed by Bressan (1972) and Fitting (1975). Independently, this technique was used by Scales (1969) in his formulation of attributional logic, where Aristotle's distinction between the negation of a sentence and of a predicate is formally expressible. In fact, Scales seems to be the first one to apply predicate abstraction to formalise a theory of definite descriptions which utilises Russell's principle ( $R$ ) without restricting it to atomic predicates. However, in other respects his system is considerably far from RDD, being a kind of negative logic with some extra constraints. Predicate abstracts were also successfully applied by Fitting and Mendelsohn (1998) to obtain a theory of definite descriptions in a modal setting. Such an approach, with slight modifications, was further developed by Indrzejczak (2020a) to obtain a cut-free SC for hybrid modal logic.

In the area of modal logic this tool is mainly used for taking control over scoping difficulties concerning modal operators but also complex terms like definite descriptions. From the standpoint of proof theory it has additional advantages. In general, introducing complex terms leads to serious problems revolved around unrestricted

instantiation of such terms for variables. Unrestricted instantiation in quantifier rules usually destroys the subformula property and interferes with a cut-elimination proof. Applying predicate abstracts opens a way to avoid such problems by drawing a distinction between direct predication restricted to variables and indirect predication via the lambda-operators. Since definite descriptions may appear only as arguments of predicate abstracts, respective rules for quantifiers may be restricted to individual variables or constants as the only allowed instantiated terms without losing generality or introducing some additional instantiation rules (see Indrzejczak (2020b)).

Since definite descriptions are used only as arguments of predicate abstracts, the modified version of  $(R)$  is  $(R_\lambda)$ :

$$(\lambda x\psi)(ty\varphi) \rightarrow \exists x(\forall y(\varphi \leftrightarrow y = x) \wedge \psi), \quad (R_\lambda)$$

where  $\varphi$  does not contain free occurrences of  $x$ . In this way we avoid problems with scope and inconsistency described above. Not only atomic predicates but also arbitrary formulas may be used to constitute lambda atoms and definite descriptions. Moreover, no additional denotation principles for atomic formulas are needed. Indeed, for variables they hold by the fact that all variables denote, and for definite descriptions they follow from  $(R_\lambda)$ . This is important if someone agrees, e.g., with the abovementioned criticism of Bostock about the status of atomic predicates.

To sum up: we define the logic  $\mathbf{L}_{R_\lambda}$  as pure FOLI (first-order logic with identity) with added  $(R_\lambda)$  and  $\beta$ -conversion for the lambda-operator. More concretely, its axiomatisation is the standard axiomatisation of pure FOLI augmented with  $(R_\lambda)$  and  $(\lambda x\psi)(b) \leftrightarrow \psi[x/b]$ , where  $b$  is a free variable (parameter), substituted for  $x$  in  $\psi$ . In general, we will consider it as defined for an arbitrary signature (the set of predicates and individual constants, but with *no* function symbols). Notice, however, that if any individual constants are present, then the rules for quantifier elimination and the principle of  $\beta$ -conversion apply to both variables and individual constants.

$\mathbf{L}_{R_\lambda}$  saves the essential features of the Russellian approach. Moreover, it avoids several problems like arbitrary restriction of  $(R)$  to atomic predicates symbols, scoping difficulties, or running into contradictions. However, in the semantics it retains the eliminativist Russellian flavour in the sense that definite descriptions are not characterised in the model by an interpretation function like in other approaches. Instead they are handled only by a specific clause for lambda-atoms in the definition of the forcing relation. Such a solution considerably simplifies a formulation of the semantics and completeness proof, since we avoid simultaneous induction on the complexity of terms and formulas.

## 4 Preliminaries

### 4.1 Syntax

We consider sentences, that is, formulas with no free variables, built in the standard first-order language  $\mathcal{L}$  with identity treated as a logical constant. The vocabulary of  $\mathcal{L}$  consists of:

- a countably infinite set of bound individual variables  $\text{VAR} = \{x, y, z \dots\}$ ,
- a countably infinite set of parametric (free) individual variables  $\text{PAR} = \{a, b, c, \dots\}$ ,
- the contradiction symbol  $\perp$ ,
- a set of propositional connectives:  $\neg, \wedge$ ,
- the universal quantifier  $\forall$ ,
- the definite description operator  $\iota$ ,
- the abstraction operator  $\lambda$ ,
- the identity relation  $=$ ,
- left and right parentheses:  $(, )$ .

A *signature*  $\Sigma$  for the language  $\mathcal{L}$  is a triple  $(\text{PRED}, \text{CONS}, \text{ar})$ , where  $\text{PRED}$  is a (possibly empty) set of predicates,  $\text{CONS}$  is a (possibly empty) set of individual constants, and  $\text{ar} : \text{PRED} \rightarrow \mathbb{N}_+$  is a function assigning a (positive) arity to each predicate.

Given a signature  $\Sigma = (\text{PRED}, \text{CONS}, \text{ar})$ , a set of terms  $\text{TERM}$  and a set of formulas  $\text{FOR}$  over  $\Sigma$  are defined simultaneously by the following context-free grammars:

$$\begin{aligned} \text{TERM} &::= x \mid a \mid c \mid \iota x \varphi, \\ \text{FOR} &::= \perp \mid P(s_1, \dots, s_n) \mid s_1 = s_2 \mid \neg \varphi \mid \varphi \wedge \psi \mid \forall x \varphi \mid (\lambda x \psi)(t), \end{aligned}$$

where  $x \in \text{VAR}, a \in \text{PAR}, c \in \text{CONS}, \varphi, \psi \in \text{FOR}, s_1, \dots, s_n \in \text{VAR} \cup \text{PAR} \cup \text{CONS}, t \in \text{TERM}$  such that  $x$  does not occur in  $t$ , and  $P \in \text{PRED}$  with  $\text{ar}(P) = n$ . Henceforth, we will refer to the set  $\text{VAR} \cup \text{PAR} \cup \text{CONS}$  as  $\text{TERM}^-$ . An expression  $\varphi[s]$  indicates that  $s$  occurs (freely if it is a variable) in  $\varphi$ . A formula  $\varphi[s_1/s_2]$  is the result of a uniform substitution of  $s_1$  with  $s_2$  in  $\varphi$ , whereas  $\varphi[s_1//s_2]$  is a result of replacing some occurrences of  $s_1$  in  $\varphi$  with  $s_2$ . Note that this notation is restricted to  $s_1, s_2 \in \text{TERM}^-$  so we can make substitutions and replacements only using variables, parameters, or individual constants, but not definite descriptions. In practice, when constructing a tableau proof, variables are substituted only with parameters or constants, however in the formulation of the semantics and in metalogical proofs it may happen that variables are substituted for variables. In such cases it is assumed that the substituted variable is free after the substitution.

Finally, we also use the following standard abbreviations:

$$\begin{aligned} s_1 \neq s_2 &::= \neg(s_1 = s_2) & \top &::= \neg \perp \\ \exists x \varphi &::= \neg \forall x \neg \varphi & \varphi \vee \psi &::= \neg(\neg \varphi \wedge \neg \psi) \\ \varphi \rightarrow \psi &::= \neg \varphi \vee \psi & \varphi \leftrightarrow \psi &::= (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi), \end{aligned}$$

where  $s_1, s_2 \in \text{TERM}^-$ .

Note that the absence of function symbols in considered signatures for  $\mathcal{L}$  is due to the fact, already mentioned, that they can be simulated by using the operator  $\iota$ .

### 4.2 Semantics

Given a signature  $\Sigma = (\text{PRED}, \text{CONS}, \text{ar})$ , a *model over*  $\Sigma$  is a structure  $\mathcal{M} = (\mathcal{D}, \mathcal{I})$ , where  $\mathcal{D} \neq \emptyset$  is called a *domain* and  $\mathcal{I}$  is an *interpretation*. For each predicate



$P \in \text{PRED}$ ,  $\mathcal{I}(P) \subseteq \mathcal{D}^n$ , where  $n$  is the arity of  $P$ . For each constant  $c \in \text{CONS}$ ,  $\mathcal{I}(c) \in \mathcal{D}$ . An *assignment*  $v$  is defined as a function mapping variables and parameters to elements of the domain. An  $x$ -variant  $v'$  of  $v$  agrees with  $v$  on all arguments, save, possibly,  $x$ . We will write  $v_o^x$  to denote the  $x$ -variant of  $v$  with  $v_o^x(x) = o$ .

Given a signature  $\Sigma = (\text{PRED}, \text{CONS}, \text{ar})$ , sets TERM and FOR over  $\Sigma$ , a model  $\mathcal{M} = (\mathcal{D}, \mathcal{I})$  over  $\Sigma$ , and an assignment  $v$ , let  $\mathcal{I}_v(s)$  be  $v(s)$  if  $s \in \text{VAR} \cup \text{PAR}$ , or  $\mathcal{I}(s)$  if  $s \in \text{CONS}$ . Then the notion of *satisfaction of a formula  $\varphi$  in  $\mathcal{M}$  under  $v$* , in symbols  $\mathcal{M}, v \models \varphi$ , is defined inductively as follows:

$$\begin{aligned} & \mathcal{M}, v \not\models \perp \\ \mathcal{M}, v \models P(s_1, \dots, s_n) & \text{ iff } \langle \mathcal{I}_v(s_1), \dots, \mathcal{I}_v(s_n) \rangle \in \mathcal{I}(P) \\ \mathcal{M}, v \models s_1 = s_2 & \text{ iff } \mathcal{I}_v(s_1) = \mathcal{I}_v(s_2) \\ \mathcal{M}, v \models (\lambda x \psi)(s) & \text{ iff } \mathcal{M}, v_o^x \models \psi \text{ and } o = \mathcal{I}_v(s) \\ \mathcal{M}, v \models (\lambda x \psi)(t y \varphi) & \text{ iff there exists } o \in \mathcal{D} \text{ such that } \mathcal{M}, v_o^x \models \varphi[y/x], \\ & \mathcal{M}, v_o^x \models \psi, \text{ and for any } y\text{-variant } v' \text{ of } v_o^x, \text{ if} \\ & \mathcal{M}, v' \models \varphi, \text{ then } v'(y) = o \\ \mathcal{M}, v \models \neg \varphi & \text{ iff } \mathcal{M}, v \not\models \varphi, \\ \mathcal{M}, v \models \varphi \wedge \psi & \text{ iff } \mathcal{M}, v \models \varphi \text{ and } \mathcal{M}, v \models \psi, \\ \mathcal{M}, v \models \forall x \varphi & \text{ iff } \mathcal{M}, v_o^x \models \varphi, \text{ for all } o \in \mathcal{D}, \end{aligned}$$

where  $P \in \text{PRED}$  with  $\text{ar}(P) = n$ ,  $s, s_1, \dots, s_n \in \text{TERM}^-$ ,  $x, y \in \text{VAR}$ , and  $\varphi, \psi \in \text{FOR}$ , and  $x$  is not free in  $\varphi$  in condition for  $(\lambda x \psi)(t y \varphi)$ .

A formula  $\varphi$  over a signature  $\Sigma$  is called *satisfiable* if there exists a model  $\mathcal{M}$  over  $\Sigma$  and a valuation  $v$  such that  $\mathcal{M}, v \models \varphi$ .  $\varphi$  is *valid*, in symbols  $\models \varphi$ , if, for all models  $\mathcal{M}$  and valuations  $v$ ,  $\mathcal{M}, v \models \varphi$ . In the remainder of the paper, instead of writing  $\mathcal{M}, v \models \varphi_1, \dots, \mathcal{M}, v \models \varphi_n$ , we will write  $\mathcal{M}, v \models \varphi_1, \dots, \varphi_n$ . Semantically we identify  $\mathbf{L}_{R_\lambda}$  as the set of all valid formulas.

**Example 1** Consider the following sentence: ‘The oldest daughter of Anne got married to some businessman and is the richest woman in the family (of Anne)’. Let  $d_1$  abbreviate the first and  $d_2$  the second description, which after unpacking are:  $t x(D(x, a) \wedge \forall y(D(y, a) \wedge y \neq x \rightarrow O(x, y)))$  and  $t x \forall y(F(y, a) \wedge W(y) \wedge y \neq x \rightarrow R(x, y))$ . Then the sentence in question is formalised as  $(\lambda z(\exists y(B(y) \wedge M(z, y)) \wedge (\lambda u(z = u))(d_2)))(d_1)$ . It can be shown that it implies, e.g., ‘Some daughter of Anne got married to a businessman’. Let us notice that the formalisation of the above sentence in Russell’s original theory leads to a much more convoluted formula.

Although definite descriptions are only allowed in the scope of  $\lambda$ -expressions, this does not preclude the possibility of them interacting with other referring expressions such as individual constants in a single formula.

**Example 2** Consider the following sentence: ‘Scott is the author of «Waverley».’ It states identity of a proper name (individual constant) and a definite description. The logic  $\mathbf{L}_{R_\lambda}$  provides us with a means to formally capture this sentence. The definite description involved can be written as  $t y A(y, w)$ , where  $w$  is the individual constant representing “Waverley” and  $A$  represents the relation of authorship. Then the discussed identity is expressed by the following formula:  $(\lambda x s = x)(t y A(y, w))$ , where  $s$  represents Scott.

The syntax and semantics of  $\mathbf{L}_{R_\lambda}$  make it possible to distinguish between different usages of negation, namely internal and external, without resorting to any external tools for scope demarcation, which was not possible in Russell's original setting.

**Example 3** Consider the sentences: 'It is not the case that the least integer is even' and 'The least integer is not even'. Let the set of all integers be the domain of our interest. Then the definite description 'the least integer' can be formalised as  $\iota y(\forall z(z \neq y \rightarrow z > y))$ . The first of the discussed sentences can be written as  $\neg(\lambda x D(x, 2))(\iota y \forall z(z \neq y \rightarrow z > y))$ , where  $D$  is the relation of divisibility. According to the relevant semantic condition it is false that there exists a unique object  $o$  simultaneously satisfying  $D(o, 2)$  and  $\forall z(z \neq o \rightarrow z > o)$ . In other words, for each object  $o$  in the domain, either it is the case that  $\neg D(o, 2)$ , or  $\exists z(z \neq o \wedge z < o)$ , or there exists  $o' \neq o$  in the domain such that  $\forall z(z \neq o' \rightarrow z > o')$ . On the other hand, the latter of the considered sentences can be formalised as  $(\lambda x \neg D(x, 2))(\iota y \forall z(z \neq y \rightarrow z > y))$ . By the semantics of  $\mathbf{L}_{R_\lambda}$ , this formula being true implies that there exists a unique object  $o$  in the domain satisfying  $\neg D(o, 2)$  and  $\forall z(z \neq o \rightarrow z > o)$ . An obvious consequence of the difference between the semantics of internally and externally used negation is that the equivalence  $\neg(\lambda x D(x, 2))(\iota y \forall z(z \neq y \rightarrow z > y)) \leftrightarrow (\lambda x \neg D(x, 2))(\iota y \forall z(z \neq y \rightarrow z > y))$  is not a theorem of  $\mathbf{L}_{R_\lambda}$ .

## 5 Calculus

In this section we focus on the construction of a tableau calculus for the logic of Russellian descriptions  $\mathbf{L}_{R_\lambda}$ , henceforth abbreviated as  $\mathbf{TC}_{R_\lambda}$ .

A *tableau*  $\mathcal{T}$  generated by a calculus  $\mathbf{TC}_{R_\lambda}$  is a *derivation tree* whose nodes are assigned formulas in the language of deduction. A *branch* of  $\mathcal{T}$  is a simple path from the root to a leaf of  $\mathcal{T}$ . For simplicity, we will identify each branch  $\mathcal{B}$  with the set of formulas assigned to nodes on  $\mathcal{B}$ .

A general form of rules is as follows:  $\frac{\Phi}{\Psi_1 | \dots | \Psi_n}$ , where  $\Phi$  is the set of *premises* and each  $\Psi_i$ , for  $i \in \{1, \dots, n\}$ , is a set of *conclusions*. If a rule has more than one set of conclusions, it is called a *branching* rule. Otherwise it is *non-branching*. Thus, if a rule  $\frac{\Phi}{\Psi_1 | \dots | \Psi_n}$  is applied to  $\Phi$  occurring on  $\mathcal{B}$ ,  $\mathcal{B}$  splits into  $n$  branches:  $\mathcal{B} \cup \{\Psi_1\}, \dots, \mathcal{B} \cup \{\Psi_n\}$ . A rule (R) with  $\Phi$  as the set of its premises is *applicable* to  $\Phi$  occurring on a branch  $\mathcal{B}$  if it has not yet been applied to  $\Phi$  on  $\mathcal{B}$ . A set  $\Phi$  is called (R)-*expanded* on  $\mathcal{B}$  if (R) has already been applied to  $\Phi$  on  $\mathcal{B}$ . We say that (R) is *sound* if, whenever  $\Phi$  is satisfiable, then  $\Phi \cup \Psi_i$  is satisfiable, for some  $i \in \{1, \dots, n\}$ . (R) is *invertible* if it is sound and, for each  $i \in \{1, \dots, n\}$ , if  $\Psi_i$  is satisfiable, then  $\Phi \cup \Psi_i$  is satisfiable. A term  $t$  is called *fresh* on a branch  $\mathcal{B}$  if it has not yet occurred on  $\mathcal{B}$ . We call a branch  $\mathcal{B}$  *closed* if the inconsistency symbol  $\perp$  occurs on  $\mathcal{B}$ . If  $\mathcal{B}$  is not closed, it is *open*. A branch is *fully expanded* if it is closed or no rules are applicable to (sets of) formulas occurring on  $\mathcal{B}$ . Note that due to the fact that new parameters can be introduced to a branch by, e.g.,  $(\neg\forall)$  or  $(\iota_1)$ , fully expanded branches do not need to be finite. A tableau  $\mathcal{T}$  is called *closed* if all of its branches are closed. Otherwise  $\mathcal{T}$  is called *open*. Finally,  $\mathcal{T}$  is *fully expanded* if all its branches are fully expanded. A *tableau proof* of a formula  $\varphi$  is a closed tableau with  $\neg\varphi$  at its root. A formula  $\varphi$

is tableau-valid (with respect to the calculus  $\mathbf{TC}_{R_\lambda}$ ) if there exists a tableau proof of  $\varphi$ . It is known that a tableau calculus is *confluent*, that is, every partial tableau proof of a formula  $\varphi$  can be expanded into a full proof (Hähnle & Beckert, 1999), if all rules are invertible. In the case of  $\mathbf{TC}_{R_\lambda}$  all rules are trivially invertible, as all formulas introduced to a branch at some point remain available for further rule applications. This makes  $\mathbf{TC}_{R_\lambda}$  a confluent tableau calculus, yet the order of rule application imposed by a decision procedure based on  $\mathbf{TC}_{R_\lambda}$  may affect its efficiency. We will not be pursuing this problem any further, as it is beyond the scope of this paper.

In Fig. 1 we present the rules constituting  $\mathbf{TC}_{R_\lambda}$ . We transfer the notation from the previous section with the caveat that  $a$  denotes a parameter that is *fresh* on the branch, whereas  $b, b_1, b_2$  denote parameters or constants that have already been present on the branch. Finally,  $b_1 \approx b_2$  stands for either  $b_1 = b_2$  or  $b_2 = b_1$ . While the closure, propositional, quantifier rules, Leibniz’s rule and the non-empty domain rule are standard and need no further explanation, a commentary on the specific rules for the  $\iota$  and  $\lambda$  operators is in order.

$(\iota_1)$  handles the scenario where a definite description occurs in the scope of a  $\lambda$ -expression. Then  $(\iota_1)$  stipulates that both the formulas hold of the same fresh parameter. If, moreover, a formula constituting a definite description occurs independently on the branch, then  $(\iota_2)$  guarantees that all the parameters it holds of are identical. If a definite description  $\iota y\varphi$  in the scope of a  $\lambda$ -expression  $\lambda x\psi$  occurs in the negated form, then for any parameter  $b$  present on the branch, either  $\varphi$  does not hold of  $b$ , or  $\psi$  does not hold of  $b$ , or we can introduce a fresh parameter  $a$  distinct from  $b$  such that  $\varphi$  holds of  $a$ . Finally,  $(\lambda)$  and  $(\neg\lambda)$  are tableau-counterparts of standard  $\beta$ -reduction known from  $\lambda$ -calculus. Their application is restricted to parameters and individual constants. Note that the rules  $(\forall)$  and  $(\neg\iota)$  both have an enthymematic meta-premise ‘a parameter  $b$  is present on the branch’, which manifests itself in the presence of  $b$  in the conclusions of both of them. Consequently, like any other rule,  $(\forall)$  and  $(\neg\iota)$  can be applied only once to a given combination of its premise and a parameter present on the branch. Observe that if there are no parameters on a branch, then neither  $(\forall)$  nor  $(\neg\iota)$  can be applied. In such a case, if no other rules are applicable, we apply the rule (NE), which corresponds to the fact that we only consider non-empty domains. By that means we introduce to the branch a fresh parameter which can then fire the aforementioned rules. What is also worth noting is that whenever an application of a rule results with a formula  $\varphi$  where a term  $b_2$  was substituted for a term  $b_1$ , or  $b_2$  replaced  $b_1$ , there is no risk that  $b_2$  will be bound in  $\varphi$  thanks to the fact that VAR and PAR (as well as VAR and CONS) are disjoint sets.

**Example 4** For the sake of example, let us consider a  $\mathbf{TC}_{R_\lambda}$ -proof of  $(R_\lambda)$ , where  $\psi = P(x)$  and  $\varphi = Q(y)$ :

$$(\lambda x P(x))(\iota y Q(y)) \wedge \forall x \neg(\forall y [(\neg(Q(y) \wedge y \neq x) \wedge \neg(y = x \wedge \neg Q(y)))] \wedge P(x))$$

Closure rules

$$(\perp_1) \frac{\varphi, \neg\varphi}{\perp} \quad (\perp_2) \frac{b \neq b}{\perp}$$

Leibniz’s rule

$$(L) \frac{b_1 \approx b_2, \varphi[b_1]}{\varphi[b_1//b_2]}$$

Non-empty domain rule

$$(NE) \frac{}{a = a}$$

Propositional rules

$$(\neg\neg) \frac{\neg\neg\varphi}{\varphi} \quad (\wedge) \frac{\varphi \wedge \psi}{\varphi} \quad (\neg\wedge) \frac{\neg(\varphi \wedge \psi)}{\neg\varphi \mid \neg\psi}$$

$$\psi$$

Quantifier rules

$$(\forall) \frac{\forall x\varphi}{\varphi[x/b]} \quad (\neg\forall) \frac{\neg\forall x\varphi}{\neg\varphi[x/a]}$$

$\iota$ -rules

$$(\iota_1) \frac{(\lambda x\psi)(\iota y\varphi)}{\varphi[y/a]} \quad (\iota_2) \frac{(\lambda x\psi)(\iota y\varphi), \varphi[y/b_1], \varphi[y/b_2]}{b_1 = b_2}$$

$$\psi[x/a]$$

$$(\neg\iota) \frac{\neg(\lambda x\psi)(\iota y\varphi)}{\neg\psi[x/b] \mid \neg\varphi[y/b] \mid \varphi[y/a]}$$

$$a \neq b$$

$\lambda$ -rules

$$(\lambda) \frac{(\lambda x\psi)(b)}{\psi[x/b]} \quad (\neg\lambda) \frac{\neg(\lambda x\psi)(b)}{\neg\psi[x/b]}$$

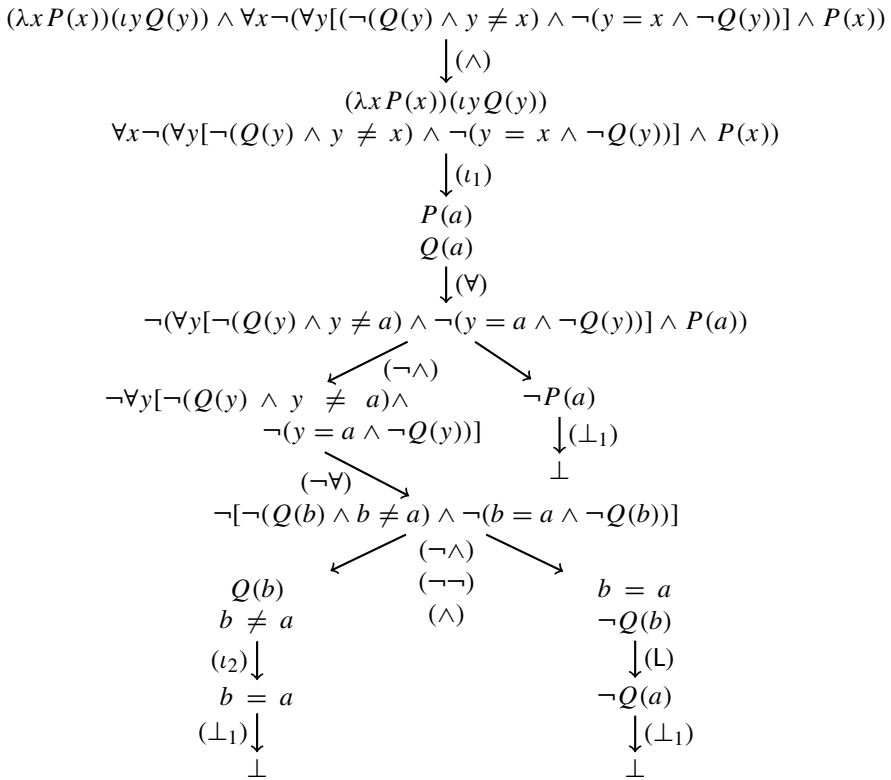
$b, b_1, b_2$  are parameters or constants present on the branch.

$a$  is a parameter fresh on the branch.

The rule (NE) can only be applied if there is no parameter or constant present on the branch and no other rules are applicable.

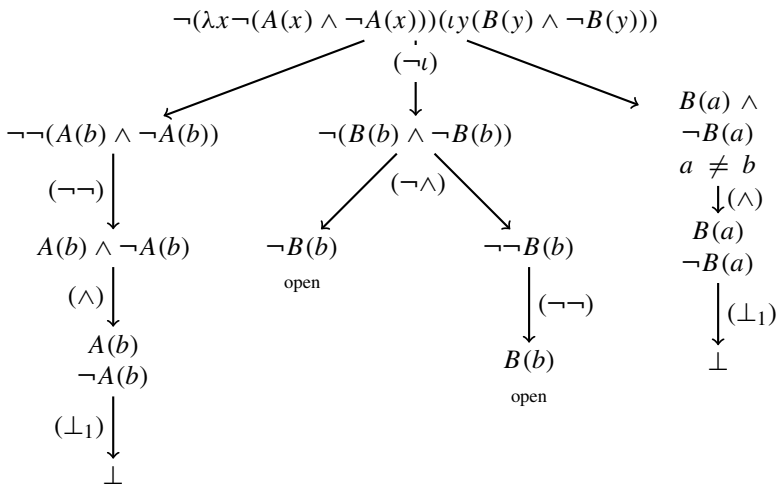
Fig. 1 Rules of the tableau calculus  $\mathbf{TC}_{R_\lambda}$

This is a pre-processed form involving only those operators which occur in  $\mathbf{TC}_{R_\lambda}$ ; the original implication is  $(\lambda x P(x))(\iota y Q(y)) \rightarrow \exists x (\forall y (Q(y) \leftrightarrow y = x) \wedge P(x))$ .



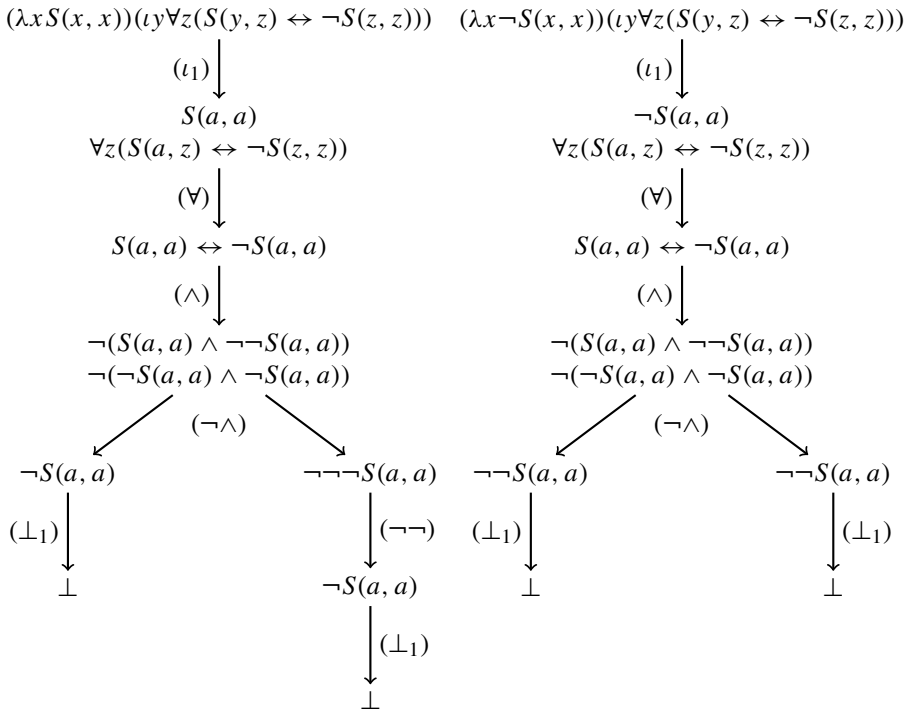
Note also that the presence of the  $\lambda$ -operator in the calculus prevents us from stumbling upon contradictions like in the example from p. 6.

**Example 5** Consider the following formulas:  $\psi(x) := A(x) \rightarrow A(x)$  and  $\varphi(y) := B(y) \wedge \neg B(y)$ . After suitably rewriting  $\psi(x)$ , we can show within  $\mathbf{TC}_{R_\lambda}$  that the formula  $(\lambda x \psi(x))(ty\varphi(y))$  is not a  $\mathbf{L}_{R_\lambda}$  tautology, even if we assume that the domain of individual objects is non-empty, that is, there occurs a parameter on the branch (say  $b$ ):

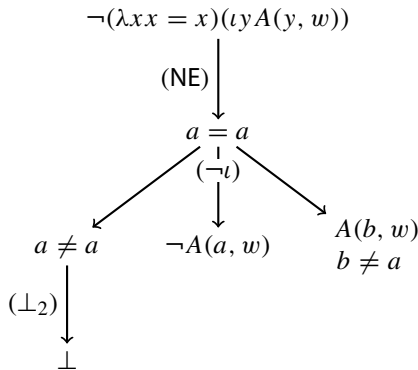


Consequently, the right-hand side of  $(R_\lambda)$  cannot be derived, which means that we will not get a contradiction of the form  $B(c) \wedge \neg B(c)$ .

Let us now consider the following pair of formulas:  $\chi(x) := S(x, x)$ ,  $\theta(y) := \forall z(S(y, z) \leftrightarrow \neg S(z, z))$ , where  $S(x, y)$  stands for ‘ $x$  shaves  $y$ ’. If any of the formulas  $(\lambda x \chi(x))(\iota y \theta(y))$  or  $(\lambda x \neg \chi(x))(\iota y \theta(y))$  turned out true, this would lead to a form of Russell’s paradox. By means of  $\mathbf{TC}_{R_\lambda}$ , however, after rewriting  $(\lambda x \chi(x))(\iota y \theta(y))$  in the language of  $\mathbf{L}_{R_\lambda}$  it is possible to prove that both of them are contradictory, and hence, false. Thus, the equivalence  $(\lambda x \chi(x))(\iota y \theta(y)) \leftrightarrow (\lambda x \neg \chi(x))(\iota y \theta(y))$  is indeed a theorem of  $\mathbf{L}_{R_\lambda}$ , but, by the argument discussed in Example 3,  $\mathbf{L}_{R_\lambda}$  does not make true the equivalence  $(\lambda x \chi(x))(\iota y \theta(y)) \leftrightarrow \neg((\lambda x \chi(x))(\iota y \theta(y)))$ , which helps us avoid the paradox. In the trees below we unpack the definition:  $\varphi \leftrightarrow \psi := \neg(\varphi \wedge \neg \psi) \wedge \neg(\psi \wedge \neg \varphi)$ .



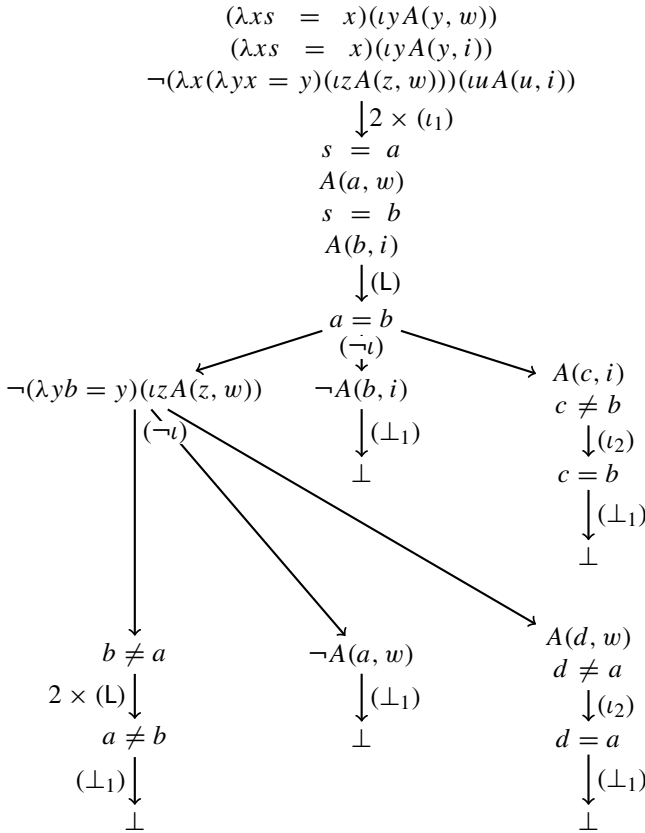
**Example 6** Consider the definite description from Example 2, namely ‘the author of “Waverley”’. Notice that  $\mathbf{TC}_{R_\lambda}$  does not prove the sentence ‘The author of “Waverley” is the author of ” Waverley”’, represented in  $\mathbf{L}_{R_\lambda}$  as  $(\lambda x x = x)(\iota y A(y, w))$ . Indeed, a derivation tree for  $\neg(\lambda x x = x)(\iota y A(y, w))$  looks as follows:



The formula  $(\lambda x x = x)(\iota y A(y, w))$  is not provable in  $\mathbf{TC}_{R_\lambda}$ , as we do not know whether  $\iota y A(y, w)$  is a proper definite description. This is consistent with Russell’s

assumptions. Notice, however, that over all theories where  $\iota_y A(y, w)$  is proper, that is, theories which entail  $\exists! x A(x, w)$ ,  $(\lambda x x = x)(\iota_y A(y, w))$  is provable in  $\mathbf{TC}_{R_\lambda}$ .

Now, consider two sentences: ‘Scott is the author of “Waverley”’ and ‘Scott is the author of “Ivanhoe”’, represented as  $(\lambda x s = x)(\iota_y A(y, w))$  and  $(\lambda x s = x)(\iota_y A(y, i))$ , respectively. In  $\mathbf{TC}_{R_\lambda}$  we can show that the sentence ‘The author of “Waverley” is the author of “Ivanhoe”’, formalised as  $(\lambda x(\lambda y x = y)(\iota_z A(z, w)))(\iota u A(u, i))$  is derivable from the above sentences. Indeed:



More generally speaking, using  $\mathbf{TC}_{R_\lambda}$  we can show, in a similar way, that from any two of the above-mentioned sentences the remaining one is derivable.

### 6 Soundness and Completeness

In this section we will rely on two well-known lemmas which we recall without proofs (see, e.g., Ebbinghaus, Flum, and Thomas (1994, Sections III.4 and III.8)).



**Lemma 1** (Coincidence Lemma) *Let  $\varphi \in \text{FOR}$ , let  $\mathcal{M} = (\mathcal{D}, \mathcal{I})$  be a model, and let  $v_1, v_2$  be assignments. If  $v_1(s) = v_2(s)$  for each free variable or parameter  $s$  occurring in  $\varphi$ , then  $\mathcal{M}, v_1 \models \varphi$  if and only if  $\mathcal{M}, v_2 \models \varphi$ .*

**Lemma 2** (Substitution Lemma) *Let  $\varphi \in \text{FOR}$ ,  $s \in \text{TERM}^-$ , and let  $\mathcal{M} = (\mathcal{D}, \mathcal{I})$  be a model. Then  $\mathcal{M}, v \models \varphi[x/s]$  if and only if  $\mathcal{M}, v_{\mathcal{I}_v(s)}^x \models \varphi$ .*

### 6.1 Soundness

We start the section with a lemma that relies on the notion of sound rule introduced in the preceding section.

**Lemma 3** *All rules of  $\text{TC}_{R_\lambda}$  are sound.*

**Proof** Since the closure, propositional, and quantifier rules, as well as Leibniz’s and the non-empty domain rule are standard, we restrict ourselves to proving soundness of  $\iota$ -rules and  $\lambda$ -rules. In what follows, we assume that  $b, b_1, b_2$  are parameters, as a proof for them being constants is analogous.

( $\iota_1$ ) Assume that  $(\lambda x\psi)(\iota y\varphi)$  is satisfiable. It means that there exists a model  $\mathcal{M} = (\mathcal{D}, \mathcal{I})$  and an assignment  $v$  such that  $\mathcal{M}, v \models (\lambda x\psi)(\iota y\varphi)$ . Hence, by the respective satisfaction condition, there exists an object  $o \in \mathcal{D}$  such that  $\mathcal{M}, v_o^x \models \varphi[y/x]$ ,  $\psi$  and for all  $y$ -variants  $v'$  of  $v_o^x$ , if  $\mathcal{M}, v' \models \varphi$ , then  $v'(y) = o$ . Without loss of generality let’s assume that  $a$  is a fresh parameter such that  $v(a) = o$ . By the Substitution Lemma, we get that  $\mathcal{M}, v \models \varphi[y/(x/a)]$ ,  $\psi[x/a]$ , where  $\varphi[y/(x/a)]$  is the result of substituting  $y$  with  $a$  which replaced  $x$ . The substitution of  $y$  with  $x$  follows from the semantic condition for lambda-atoms involving definite descriptions, whereas the substitution of  $x$  with  $a$  follows from the Substitution Lemma. Consequently,  $\mathcal{M}, v \models \varphi[y/a]$ ,  $\psi[x/a]$ , as desired.

( $\iota_2$ ) Assume that  $(\lambda x\psi)(\iota y\varphi)$ ,  $\varphi[y/b_1]$ , and  $\varphi[y/b_2]$  are jointly satisfiable. It means that there exists a model  $\mathcal{M} = (\mathcal{D}, \mathcal{I})$  and an assignment  $v$  such that  $\mathcal{M}, v \models (\lambda x\psi)(\iota y\varphi)$ ,  $\varphi[y/b_1]$ ,  $\varphi[y/b_2]$ . Hence, by the respective satisfaction condition, there exists an object  $o \in \mathcal{D}$  such that  $\mathcal{M}, v_o^x \models \varphi[y/x]$ ,  $\psi$  and, for any  $y$ -variant  $v'$  of  $v_o^x$ , if  $\mathcal{M}, v' \models \varphi$ , then  $v'(y) = o$ . Let  $v(b_1) = o'$  and  $v(b_2) = o''$ . By the Substitution Lemma,  $\mathcal{M}, v_{o'}^y \models \varphi$  and  $\mathcal{M}, v_{o''}^y \models \varphi$ . Since  $x$  does not occur freely in  $\varphi$ , by the Coincidence Lemma we get  $\mathcal{M}, (v_o^x)_{o'}^y \models \varphi$  and  $\mathcal{M}, (v_o^x)_{o''}^y \models \varphi$ . Since both  $(v_o^x)_{o'}^y$  and  $(v_o^x)_{o''}^y$  are  $y$ -variants of  $v_o^x$ , we obtain  $(v_o^x)_{o'}^y(y) = o$  and  $(v_o^x)_{o''}^y(y) = o$ , and so,  $o = o' = o''$ . As  $o = v(b_1) = v(b_2)$ , the respective satisfaction condition yields  $\mathcal{M}, v \models b_1 = b_2$ , as expected.

( $\neg\iota$ )d Assume that  $\neg(\lambda x\psi)(\iota y\varphi)$  is satisfiable. Then there exists a model  $\mathcal{M} = (\mathcal{D}, \mathcal{I})$  and an assignment  $v$  such that  $\mathcal{M}, v \models \neg(\lambda x\psi)(\iota y\varphi)$ . By the respective satisfiability conditions it means that for all objects  $o \in \mathcal{B}$  (at least) one of the following three conditions holds: (1)  $\mathcal{M}, v_o^x \not\models \psi$ ; (2)  $\mathcal{M}, v_o^x \not\models \varphi[y/x]$ ; (3) there exists a  $y$ -variant  $v'$  of  $v_o^x$  such that  $\mathcal{M}, v' \models \varphi$  and  $v'(y) \neq o$ . Let  $b$  be a parameter present on the branch and  $v(b) = o'$ . If (1) holds for  $o'$ , i.e.,  $\mathcal{M}, v_{o'}^x \not\models \psi$ , then, by the Substitution

Lemma,  $\mathcal{M}, v \not\models \psi[x/b]$ , whence, by the respective satisfaction condition, we get  $\mathcal{M}, v \models \neg\psi[x/b]$ . Let (2) hold for  $o'$ , i.e.,  $\mathcal{M}, v_{o'}^x \not\models \varphi[y/x]$ . By the Substitution Lemma we get  $\mathcal{M}, v \not\models \varphi[y/(x/b)]$ , which is equivalent to  $\mathcal{M}, v \not\models \varphi[y/b]$ . By the satisfaction condition for negation we obtain  $\mathcal{M}, v \models \neg\varphi[y/b]$ . Assume that (3) holds for  $o'$ , that is, there exists a  $y$ -variant  $v'$  of  $v_{o'}^x$  such that  $\mathcal{M}, v' \models \varphi$  and  $v'(y) \neq o'$ . Let  $\mathcal{D} \ni o'' \neq o'$  be such that  $\mathcal{M}, (v_{o''}^x)^y_{o''} \models \varphi$ . Without loss of generality we may assume that there exists  $a \in \text{PAR}$  such that  $a$  does not occur freely in  $\varphi$  and  $v(a) = o''$ . Since  $x$  does not occur freely in  $\varphi$ , we can apply the Substitution Lemma twice and from  $\mathcal{M}, (v_{o''}^x)^y_{o''} \models \varphi$  obtain  $\mathcal{M}, v \models \varphi[y/a]$ , as desired.

( $\lambda$ ) Let  $b$  be a parameter present on the branch. Assume that  $(\lambda x\psi)(b)$  is satisfiable. Then there exists a model  $\mathcal{M} = (\mathcal{D}, \mathcal{I})$  and an assignment  $v$  such that  $\mathcal{M}, v \models (\lambda x\psi)(b)$ . By the respective satisfaction condition it means that  $v(b) = o$ , for some  $o \in \mathcal{D}$ , and  $\mathcal{M}, v_o^x \models \psi$ . By the Substitution Lemma it holds that  $\mathcal{M}, v \models \psi[x/b]$ , hence  $\psi[x/b]$  is satisfiable.

( $\neg\lambda$ ) Let  $b$  be a parameter present on the branch. Assume that  $\neg(\lambda x\psi)(b)$  is satisfiable. Then there exists a model  $\mathcal{M} = (\mathcal{D}, \mathcal{I})$  and an assignment  $v$  such that  $\mathcal{M}, v \models \neg(\lambda x\psi)(b)$ . By the respective satisfaction condition it means that  $\mathcal{M}, v \not\models (\lambda x\psi)(b)$ . Assume that  $v(b) = o$  for some  $o \in \mathcal{D}$ . Then by the respective satisfaction condition  $\mathcal{M}, v_o^x \not\models \varphi$ . By the Substitution Lemma we get that  $\mathcal{M}, v \not\models \psi[x/b]$ . Again, by the satisfaction condition for negation it follows that  $\mathcal{M}, v \models \neg\psi[x/b]$ , hence  $\neg\psi[x/b]$  is satisfiable.  $\square$

Now we are ready to prove the following theorem.

**Theorem 4** (Soundness) *The tableau calculus  $\text{TC}_{R_\lambda}$  is sound.*

**Proof** Let  $\varphi$  be a formula over a signature  $\Sigma$ . Let  $\mathcal{T}$  be a  $\text{TC}_{R_\lambda}$ -proof of  $\varphi$ . Each branch of  $\mathcal{T}$  is closed. By Lemma 3 all the rules of  $\text{TC}_{R_\lambda}$  preserve satisfiability, and so, going from the bottom to the top of  $\mathcal{T}$ , we can “push” unsatisfiability of  $\perp$  upwards, eventually reaching the root. Since at the root we have  $\neg\varphi$ , we conclude that it is unsatisfiable. By the well-known duality between satisfiability and validity we obtain that  $\varphi$  is valid.  $\square$

### 6.2 Completeness

Below we will show that  $\text{TC}_{R_\lambda}$  is complete, that is, for each formula  $\varphi$ , if  $\models \varphi$ , then there exists a  $\text{TC}_{R_\lambda}$ -proof of  $\varphi$ . In order to do that, we will show the contrapositive, i.e., that whenever  $\text{TC}_{R_\lambda}$  yields an open tableau  $\mathcal{T}$  for a formula  $\neg\varphi$ , with an open and fully expanded branch  $\mathcal{B}$ , then there exists a model  $\mathcal{M}_{\mathcal{B}} = (\mathcal{D}_{\mathcal{B}}, \mathcal{I}_{\mathcal{B}})$  and a valuation  $v_{\mathcal{B}}$ , that can be constructed using the data from  $\mathcal{B}$ , such that  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg\varphi$ . Thus, in short, we will show that if there is no  $\text{TC}_{R_\lambda}$ -proof for  $\varphi$ , then  $\not\models \varphi$ . We will assume that the rules are used *fairly* in the sense that whenever one of them can be applied, it will eventually be applied. Moreover, let us observe that the symmetry rule  $\frac{s_1=s_2}{s_2=s_1}$  is derivable in  $\text{TC}_{R_\lambda}$ . Indeed, it suffices to apply (L) to  $s_1 = s_2$  (which plays a double role here: an identity formula and a formula within which we replace identical parameters)

twice to obtain, subsequently,  $s_1 = s_1$  and  $s_2 = s_1$ . Thus, in this section we will use expressions of the forms  $s_1 = s_2$  and  $s_2 = s_1$  interchangeably whenever necessary.

Let  $\varphi$  be a formula over a signature  $\Sigma = (\text{PAR}, \text{CONS}, \text{ar})$ . Let  $\mathcal{T}$  be an open  $\text{TC}_{R_\lambda}$ -tableau for  $\varphi$ , where  $\mathcal{B}$  is an open and fully expanded branch. Let  $\text{CONS}(\mathcal{B})$ ,  $\text{VAR}(\mathcal{B})$ , and  $\text{PAR}(\mathcal{B})$  be the sets of, respectively, all constants, all bound variables, and all parameters occurring on  $\mathcal{B}$ . We define a binary relation  $\sim_{\mathcal{B}}$  on  $\text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$ . Let  $s_1, s_2 \in \text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$ . Then:

$$s_1 \sim_{\mathcal{B}} s_2 \text{ iff } s_1 = s_2 \text{ occurs on } \mathcal{B} \text{ or } s_1 \text{ is identical to } s_2.$$

**Proposition 5**  $\sim_{\mathcal{B}}$  is an equivalence relation.

**Proof** Of course,  $\sim_{\mathcal{B}}$  is reflexive. Indeed, every  $s \in \text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$  is identical to itself, hence, by the definition of  $\sim_{\mathcal{B}}$ ,  $s \sim_{\mathcal{B}} s$ . Symmetry of  $\sim_{\mathcal{B}}$  is a consequence of  $\mathcal{B}$  being fully expanded and the rule  $\frac{s_1=s_2}{s_2=s_1}$  being derivable in  $\text{TC}_{R_\lambda}$ . Indeed, if  $s_1, s_2 \in \text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$  are distinct and  $s_1 \sim_{\mathcal{B}} s_2$ , it means that  $s_1 = s_2$  was on  $\mathcal{B}$ . Hence  $s_2 = s_1$  is also on  $\mathcal{B}$ , and so,  $s_2 \sim_{\mathcal{B}} s_1$ . For transitivity assume that  $s_1, s_2, s_3 \in \text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$ ,  $s_1 \sim_{\mathcal{B}} s_2$ , and  $s_2 \sim_{\mathcal{B}} s_3$ . If  $s_1$  is identical to  $s_2$  or  $s_2$  is identical to  $s_3$ , we immediately obtain  $s_1 \sim_{\mathcal{B}} s_3$ . If  $s_1, s_2, s_3$  are pairwise distinct, the identities  $s_1 = s_2$  and  $s_2 = s_3$  must have occurred on  $\mathcal{B}$ . A single application of (L) to both of them yields  $s_1 = s_3$  and since  $\mathcal{B}$  is fully expanded,  $s_1 = s_3 \in \mathcal{B}$ . Hence,  $s_1 \sim_{\mathcal{B}} s_3$ . □

Obviously, thanks to the rule (L) and expandedness of  $\mathcal{B}$ ,  $\sim_{\mathcal{B}}$  is also a congruence, that is, for any  $s_1, s_2 \in \text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$  such that  $s_1 \sim_{\mathcal{B}} s_2$ , it holds that  $\varphi[s_1] \in \mathcal{B}$  if and only if  $\varphi[s_1/s_2] \in \mathcal{B}$ .

**Definition 1 (Branch model and valuation)** Let  $\mathcal{B}$  be an open and fully expanded branch of a  $\text{TC}_{R_\lambda}$ -tableau  $\mathcal{T}$  with  $\varphi$  over a signature  $\Sigma = (\text{PRED}, \text{CONS}, \text{ar})$  at the root.

A branch model is a tuple  $\mathcal{M}_{\mathcal{B}} = (\mathcal{D}_{\mathcal{B}}, \mathcal{I}_{\mathcal{B}})$  such that:

- $\mathcal{D}_{\mathcal{B}}$  is a set of  $\sim_{\mathcal{B}}$ -equivalence classes over  $\text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$  (note that thanks to the rule (NE),  $\mathcal{D}_{\mathcal{B}}$  is non-empty, as required);
- For each  $c \in \text{CONS}(\mathcal{B})$ ,  $\mathcal{I}(c) = [c]_{\sim_{\mathcal{B}}}$ ;
- For each  $P \in \text{PRED}$  and  $n \in \mathbb{N}_+$  such that  $\text{ar}(P) = n$ ,  $\mathcal{I}_{\mathcal{B}}(P) = \{(\mathcal{I}_{v_{\mathcal{B}}}(s_1), \dots, \mathcal{I}_{v_{\mathcal{B}}}(s_n)) \mid P(s_1, \dots, s_n) \in \mathcal{B}\}$ .

A branch valuation is a function  $v_{\mathcal{B}} : \text{PAR}(\mathcal{B}) \cup \text{VAR}(\mathcal{B}) \rightarrow \mathcal{D}_{\mathcal{B}}$  defined as follows:

$$v_{\mathcal{B}}(s) = \begin{cases} [s]_{\sim_{\mathcal{B}}}, & \text{if } s \in \text{PAR}(\mathcal{B}); \\ [s_0]_{\sim_{\mathcal{B}}}, & \text{if } s \in \text{VAR}(\mathcal{B}), \end{cases}$$

where  $s_0$  is the parameter that occurred on  $\mathcal{B}$  first.

Of course, all branch models and branch valuations are, respectively, models and valuations in the sense of Sect. 4.2.

We are now ready to prove the key results of this section.

**Lemma 6** *Let  $\varphi$  be a formula over a signature  $\Sigma = (\text{PRED}, \text{CONS}, \text{ar})$ ,  $\mathcal{T}$  be an open  $\text{TC}_{R_\lambda}$ -tableau with  $\neg\varphi$  at the root, and  $\mathcal{B}$  an open and fully expanded branch of  $\mathcal{T}$ . Then for each formula  $\psi$ :*

$$\text{If } \psi \in \mathcal{B}, \text{ then } \mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \psi.$$

**Proof** The proof is by induction on complexity of  $\psi$ . Note that since  $\mathcal{B}$  is open,  $\perp$  could not have occurred on  $\mathcal{B}$ .

$\psi = P(s_1, \dots, s_n)$  Assume that  $P(s_1, \dots, s_n) \in \mathcal{B}$ . Then  $s_1, \dots, s_n \in \text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$  and by the definition of  $\mathcal{M}_{\mathcal{B}}$  and  $v_{\mathcal{B}}$ ,  $\mathcal{I}_{v_{\mathcal{B}}}(s_i) = [s_i]_{\sim_{\mathcal{B}}}$ ,  $1 \leq i \leq n$ ,  $\langle \mathcal{I}_{v_{\mathcal{B}}}(s_1), \dots, \mathcal{I}_{v_{\mathcal{B}}}(s_n) \rangle \in \mathcal{I}_{\mathcal{B}}(P)$ , and so,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models P(s_1, \dots, s_n)$ .

$\psi = s_1 = s_2$  Assume that  $s_1 = s_2 \in \mathcal{B}$ . It means that  $s_1, s_2 \in \text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$ . By the definition of  $\sim_{\mathcal{B}}$ ,  $[s_1]_{\sim_{\mathcal{B}}} = [s_2]_{\sim_{\mathcal{B}}}$ , and so,  $\mathcal{I}_{v_{\mathcal{B}}}(s_1) = \mathcal{I}_{v_{\mathcal{B}}}(s_2)$ . Thus,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models s_1 = s_2$ .

$\psi = (\lambda x \chi)(s)$  Assume that  $(\lambda x \chi)(s) \in \mathcal{B}$ . Since  $\mathcal{B}$  is fully expanded, the rule  $(\lambda)$  must have been applied to  $\psi$ , yielding  $\chi[x/s] \in \mathcal{B}$ . By the inductive hypothesis,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \chi[x/s]$ . By the Substitution Lemma,  $\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[s]_{\sim_{\mathcal{B}}}}^x \models \chi$ , which, together with the fact that  $\mathcal{I}_{v_{\mathcal{B}}}(s) = [s]_{\sim_{\mathcal{B}}}$ , gives  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models (\lambda x \chi)(s)$ .

$\psi = (\lambda x \chi)(\iota y \theta)$  Assume that  $(\lambda x \chi)(\iota y \theta) \in \mathcal{B}$ . Due to expandedness of  $\mathcal{B}$ , the rule  $(\iota_1)$  must have been applied to  $\psi$ , yielding  $\chi[x/a], \theta[y/a] \in \mathcal{B}$ . By the inductive hypothesis,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \chi[x/a], \theta[y/a]$ . By the Substitution Lemma,  $\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[a]_{\sim_{\mathcal{B}}}}^x \models \chi, \theta[y/a]$ . Now, let  $b$  be a parameter such that  $\theta[y/b] \in \mathcal{B}$ . Then  $(\iota_2)$  was applied to  $\theta[y/a]$  and  $\theta[y/b]$  yielding  $a = b \in \mathcal{B}$ . By the inductive hypothesis  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \theta[y/b]$ . Since, by the construction of  $\mathbf{L}_{R_\lambda}$ -atoms involving definite descriptions,  $x$  does not occur in  $\theta[y/b]$ , by the Coincidence Lemma we obtain  $\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[a]_{\sim_{\mathcal{B}}}}^x \models \theta[y/b]$ . Further, by the Substitution Lemma, it follows that  $\mathcal{M}_{\mathcal{B}}, ((v_{\mathcal{B}})_{[a]_{\sim_{\mathcal{B}}}}^x)_{[b]_{\sim_{\mathcal{B}}}}^y \models \theta$ . Also by the inductive hypothesis  $v_{\mathcal{B}}(b) = v_{\mathcal{B}}(a)$ . Since  $b$  is arbitrary and  $((v_{\mathcal{B}})_{[a]_{\sim_{\mathcal{B}}}}^x)_{[b]_{\sim_{\mathcal{B}}}}^y$  is a  $y$ -variant of  $(v_{\mathcal{B}})_{[a]_{\sim_{\mathcal{B}}}}^x$ , we get  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models (\lambda x \chi)(\iota y \theta)$ .

$\psi = \chi \wedge \theta$  Assume that  $\chi \wedge \theta \in \mathcal{B}$ . Since  $\mathcal{B}$  is fully expanded,  $(\wedge)$  must have been applied to  $\chi \wedge \theta$ , and so  $\chi, \theta \in \mathcal{B}$ . By the inductive hypothesis,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \chi, \theta$ , and so, by the satisfaction condition for  $\wedge$ ,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \chi \wedge \theta$ .

$\psi = \forall x \chi$  Assume that  $\forall x \chi \in \mathcal{B}$  and that  $b \in \text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$ . By expandedness of  $\mathcal{B}$ ,  $(\forall)$  must have been applied to  $\forall x \chi$  and  $b$  yielding  $\chi[x/b] \in \mathcal{B}$ . By the inductive hypothesis we obtain  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \chi[x/b]$ , whence, by the Substitution Lemma, we get  $\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[b]_{\sim_{\mathcal{B}}}}^x \models \chi$ . Since  $b$  was arbitrary and  $\mathcal{D}_{\mathcal{B}}$  is defined over the set of all constants and parameters on  $\mathcal{B}$ , we arrive at  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \forall x \chi$ .

$\psi = \neg \perp$  Assume that  $\neg \perp \in \mathcal{B}$ . Since  $\mathcal{M}_{\mathcal{B}}$  is a well-defined  $\mathbf{L}_{R_\lambda}$  model and  $v_{\mathcal{B}}$  is a well-defined  $\mathbf{L}_{R_\lambda}$  valuation, we have  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \not\models \perp$ . By the satisfaction condition for  $\neg$  we obtain  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg \perp$ .

$\psi = \neg P(s_1, \dots, s_n)$  Assume that  $\neg P(s_1, \dots, s_n) \in \mathcal{B}$ . Then  $s_1, \dots, s_n \in \text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$  and since  $\mathcal{B}$  is open,  $P(s_1, \dots, s_n) \notin \mathcal{B}$ . Then, by the definition of  $\mathcal{M}_{\mathcal{B}}$  and  $v_{\mathcal{B}}$ , it holds that  $\mathcal{I}_{v_{\mathcal{B}}}(s_i) = [s_i]_{\sim_{\mathcal{B}}}$ ,  $1 \leq i \leq n$ ,  $\langle \mathcal{I}_{v_{\mathcal{B}}}(s_1), \dots, \mathcal{I}_{v_{\mathcal{B}}}(s_n) \rangle \notin \mathcal{I}_{\mathcal{B}}(P)$ . Thus,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \not\models P(s_1, \dots, s_n)$ , and so,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg P(s_1, \dots, s_n)$ .

$\psi = s_1 \neq s_2$  Assume that  $s_1 \neq s_2 \in \mathcal{B}$ . It means that  $s_1, s_2 \in \text{CONS}(\mathcal{B}) \cup \text{PAR}(\mathcal{B})$ .

By openness of  $\mathcal{B}$ ,  $s_1 = s_2 \notin \mathcal{B}$ , and by the definition of  $\sim_{\mathcal{B}}$ ,  $[s_1]_{\sim_{\mathcal{B}}} \neq [s_2]_{\sim_{\mathcal{B}}}$ .

Therefore,  $\mathcal{I}_{v_{\mathcal{B}}}(s_1) \neq \mathcal{I}_{v_{\mathcal{B}}}(s_2)$ , and so,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \not\models s_1 = s_2$  and  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models s_1 \neq s_2$ .

$\psi = \neg(\lambda x \chi)(s)$  Assume that  $\neg(\lambda x \chi)(s) \in \mathcal{B}$ . Since  $\mathcal{B}$  is fully expanded, the rule  $(\neg\lambda)$  was applied to  $\psi$ , giving  $\neg\chi[x/s] \in \mathcal{B}$ . By the inductive hypothesis,

$\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg\chi[x/s]$ . By the Substitution Lemma,  $\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[s]_{\sim_{\mathcal{B}}}}^x \models \neg\chi$ , and

so,  $\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[s]_{\sim_{\mathcal{B}}}}^x \not\models \chi$ . Since  $\mathcal{I}_{v_{\mathcal{B}}}(s) = [s]_{\sim_{\mathcal{B}}}$ , the satisfaction condition for  $\lambda$ -expressions (without  $\iota$ -expressions) yields  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg(\lambda x \chi)(s)$ .

$\psi = \neg(\lambda x \chi)(\iota y \theta)$  Assume that  $\neg(\lambda x \chi)(\iota y \theta) \in \mathcal{B}$  and that  $b$  is a parameter present on  $\mathcal{B}$ . Since  $\mathcal{B}$  is fully expanded, the rule  $(\neg\iota)$  was applied to  $\psi$ , making, for any  $b$

on the branch, one of the following three hold: (1)  $\neg\chi[x/b] \in \mathcal{B}$ , (2)  $\neg\theta[y/b] \in \mathcal{B}$ ,

(3) there is a fresh  $a$  such that  $\theta[y/a]$ ,  $a \neq b \in \mathcal{B}$ . Assume (1) is the case. By the inductive hypothesis we get  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg\chi[x/b]$ . By the Substitution Lemma we

obtain  $\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[b]_{\sim_{\mathcal{B}}}}^x \models \neg\chi$ , and so,  $\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[b]_{\sim_{\mathcal{B}}}}^x \not\models \chi$ . If (2) holds, then by, the inductive hypothesis,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg\theta[y/b]$ . By the Substitution Lemma we obtain

$\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[b]_{\sim_{\mathcal{B}}}}^y \models \neg\theta$ , and so,  $\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[b]_{\sim_{\mathcal{B}}}}^x \not\models \theta[y/x]$ . Finally, let (3) hold. Then,

by the inductive hypothesis,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \theta[y/a]$ ,  $a \neq b$ , whence  $[a]_{\sim_{\mathcal{B}}} \neq [b]_{\sim_{\mathcal{B}}}$ . Since  $x$  does not occur freely in  $\theta[y/a]$ , it holds that  $\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[b]_{\sim_{\mathcal{B}}}}^x \models \theta[y/a]$

By the Substitution Lemma we obtain  $\mathcal{M}_{\mathcal{B}}, ((v_{\mathcal{B}})_{[b]_{\sim_{\mathcal{B}}}}^x)_{[a]_{\sim_{\mathcal{B}}}}^y \models \theta$ . As previously

noted,  $[a]_{\sim_{\mathcal{B}}} \neq [b]_{\sim_{\mathcal{B}}}$ , which means, by the respective satisfaction condition,

that taking these three possibilities together, we obtain  $\mathcal{M}_{\mathcal{B}}, ((v_{\mathcal{B}})_{[b]_{\sim_{\mathcal{B}}}}^x)_{[a]_{\sim_{\mathcal{B}}}}^y \not\models$

$(\lambda x \chi)(\iota y \theta)$  and  $\mathcal{M}_{\mathcal{B}}, ((v_{\mathcal{B}})_{[b]_{\sim_{\mathcal{B}}}}^x)_{[a]_{\sim_{\mathcal{B}}}}^y \models \neg(\lambda x \chi)(\iota y \theta)$ . Neither  $x$  nor  $y$  occurs

freely in  $\neg(\lambda x \chi)(\iota y \theta)$ , so after applying the Substitution Lemma twice we obtain  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg(\lambda x \chi)(\iota y \theta)$ .

$\psi = \neg\neg\chi$  Assume that  $\neg\neg\chi \in \mathcal{B}$ . Since  $\mathcal{B}$  is fully expanded,  $(\neg\neg)$  was applied

to  $\neg\chi$ , which resulted in  $\chi \in \mathcal{B}$ . By the inductive hypothesis,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \chi$ .

Applying the satisfaction condition for  $\neg$  twice, we obtain  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg\neg\chi$ .

$\psi = \neg(\chi \wedge \theta)$  Assume that  $\neg(\chi \wedge \theta) \in \mathcal{B}$ . By expandedness of  $\mathcal{B}$  we know that

$(\neg\wedge)$  was applied to  $\neg(\chi \wedge \theta)$ , which resulted in  $\neg\chi \in \mathcal{B}$  or  $\neg\theta \in \mathcal{B}$ . Assume the

former. By the inductive hypothesis,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg\chi$ , and therefore,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \not\models \chi$ .

By the satisfaction condition for  $\wedge$  it follows that  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \not\models \chi \wedge \theta$ . Consequently,

$\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg(\chi \wedge \theta)$ . The reasoning for the latter case is analogous.

$\psi = \neg\forall x \chi$  Assume that  $\neg\forall x \chi \in \mathcal{B}$ . Since  $\mathcal{B}$  is fully expanded,  $(\neg\forall)$  was applied

to  $\forall x \chi$  resulting in  $\neg\varphi[x/a] \in \mathcal{B}$  for some then fresh parameter  $a \in \text{PAR}(\mathcal{B})$ . By the inductive hypothesis we obtain  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg\varphi[x/a]$  and  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \not\models \varphi[x/a]$ ,

whence, by the Substitution Lemma, we get  $\mathcal{M}_{\mathcal{B}}, (v_{\mathcal{B}})_{[a]_{\sim_{\mathcal{B}}}}^x \not\models \varphi$ . By the satisfaction

condition for  $\forall$  we obtain  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \not\models \forall x \chi$ , and thus,  $\mathcal{M}_{\mathcal{B}}, v_{\mathcal{B}} \models \neg\forall x \chi$ .  $\square$

Finally, we are able to establish the completeness result.

**Theorem 7** (Completeness)  $\text{TC}_{R_{\lambda}}$  is complete.

**Proof** We show that if a formula  $\varphi$  does not have a  $\text{TC}_{R_{\lambda}}$ -proof, then it is not valid.

Assume that  $\varphi$  does not have a  $\text{TC}_{R_{\lambda}}$ -proof. Then each fully expanded tableau  $\mathcal{T}$  with  $\neg\varphi$  at the root has an open and fully expanded branch. Let  $\mathcal{B}$  be such a branch. By

Lemma 6 we know that  $\mathcal{M}_B, v_B \models \neg\varphi$ . Since  $\mathcal{M}_B$  is a well-defined model and  $v_B$  is a well-defined valuation, we obtain  $\mathcal{M}_B, v_B \not\models \varphi$ , hence  $\varphi$  is not valid.  $\square$

### 7 Interpolation

In this section we will show that  $\mathbf{L}_{R_\lambda}$  satisfies two important properties: the Craig interpolation property and the Beth definability property. We say that a logic  $L$  satisfies the *Craig interpolation property* if, for all  $L$ -formulas  $\varphi$  and  $\psi$  (over any signature), if  $\models \varphi \rightarrow \psi$ , then we can construct an interpolant formula, that is, a formula  $\chi$  such that  $\models \varphi \rightarrow \chi$ ,  $\models \chi \rightarrow \psi$  and  $\chi$  contains only predicates, constants and parameters common to both  $\varphi$  and  $\psi$ . Let  $L$  be an extension of first-order logic with equality and let  $Th$  be an  $L$ -theory over  $\Sigma = (\text{PRED}, \text{CONS}, \text{ar})$ . We say that a constant  $c$  is *implicitly definable* with respect to  $Th$  if, for any two models  $\mathcal{M}, \mathcal{M}'$  of  $Th$  such that  $\mathcal{M} \upharpoonright_{\Sigma \setminus \{c\}} = \mathcal{M}' \upharpoonright_{\Sigma \setminus \{c\}}$ , and any formula  $\psi$  over  $\Sigma$ ,  $\mathcal{M} \models \psi$  if and only if  $\mathcal{M}' \models \psi$ . In other words,  $c$  is implicitly definable if whenever  $\mathcal{M}$  and  $\mathcal{M}'$  are identical with respect to all formulas over the reduced signature  $\Sigma' = (\text{PRED}, \text{CONST} \setminus \{c\}, \text{ar})$ , then they are identical with respect to all formulas over the full signature  $\Sigma$ . A constant  $c$  is said to be *explicitly definable* with respect to  $Th$  if there exists a formula  $\psi$  over  $\Sigma' = (\text{PRED}, \text{CONS} \setminus \{c\}, \text{ar})$  such that  $Th \models \forall x, \bar{y}(x = c \leftrightarrow \psi(x, \bar{y}))$ . Implicit and explicit definability of predicates is defined analogously.  $L$  has the *Beth definability property* if, for any signature  $\Sigma = (\text{PRED}, \text{CONS}, \text{ar})$ , any  $L$ -theory  $Th$  over  $\Sigma$ , and any  $\alpha \in \text{PRED} \cup \text{CONS}$  implicit and explicit definability of  $\alpha$  are equivalent. Since for the purpose of this section we are interested in constants rather than predicates,<sup>2</sup> we provide a formalisation of the condition for implicit definability restricted to constants (Hermes, 1973). Let  $L$  have the Beth definability property and let  $c$  be a constant like characterised above. Then the following conditions are equivalent:

- (i)  $Th \cup Th' \models c = c'$
- (ii)  $Th \models \forall x, \bar{y}(x = c \leftrightarrow \psi(x, \bar{y}))$ ,

where  $Th'$  is a theory resulting from replacing all occurrences of  $c$  in  $Th$  with  $c'$  and  $\psi$  is a formula over  $\Sigma' = (\text{PRED}, \text{CONS} \setminus \{c\}, \text{ar})$  and  $\bar{y}$  is a tuple of variables of arbitrary length.

A nice feature of our calculus  $\mathbf{TC}_{R_\lambda}$  is that it can be used to prove that the logic  $\mathbf{L}_{R_\lambda}$  enjoys the Craig interpolation property by actually showing, for a valid implication  $\varphi \rightarrow \psi$ , how to construct an interpolant. To this end we exploit a technique introduced by Smullyan (1968) and further adjusted to the tableaux setting by Fitting (1996). To take full advantage of this method we need to modify  $\mathbf{TC}_{R_\lambda}$  so that all the rules, save  $(\perp_1)$ , are single-premise rules. Consider the following two transformed rules:

$$(L') \frac{\varphi[b_1]}{b_1 \neq b_2 \mid \varphi[b_1/b_2]}$$

<sup>2</sup> We focus our attention on constants, because the presence of definite descriptions in  $\mathbf{L}_{R_\lambda}$  makes it possible to rewrite a theory  $Th$  where a constant  $c$  is explicitly definable into an equivalent theory  $Th'$  in which  $c$  is simply replaced with a suitable definite description (and ordinary predicates are replaced with suitable lambda-atoms if necessary). Nevertheless, all results seamlessly transfer to the case of predicates.

$$(l'_2) \frac{(\lambda x \psi)(ty\varphi)}{\neg\varphi[y/b_1] \mid \neg\varphi[y/b_2] \mid b_1 = b_2}$$

Let  $\mathbf{TC}'_{R_\lambda}$  be  $\mathbf{TC}_{R_\lambda}$  with (L) and  $(l_2)$  replaced with (L') and  $(l'_2)$ , respectively. In what follows we want to show that  $\mathbf{TC}_{R_\lambda}$  and  $\mathbf{TC}'_{R_\lambda}$  coincide. To that end we will exploit the cut rule:

$$(\text{cut}) \frac{}{\varphi \mid \neg\varphi}$$

and the following proposition.

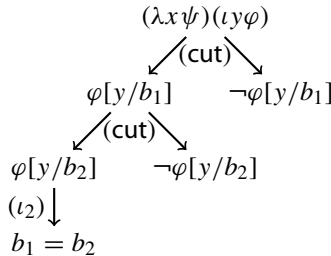
**Proposition 8** *(cut) is admissible in  $\mathbf{TC}_{R_\lambda}$ .*

**Proof** We will show that whatever is provable in  $\mathbf{TC}_{R_\lambda}$  with the added (cut)-rule, is provable in sole  $\mathbf{TC}_{R_\lambda}$ . It follows from completeness of  $\mathbf{TC}_{R_\lambda}$  and the fact that (cut) is a sound rule. Assume that  $\varphi$  is not provable in  $\mathbf{TC}_{R_\lambda}$ , although it has a proof in  $\mathbf{TC}_{R_\lambda}$  with (cut). Hence, by completeness of  $\mathbf{TC}_{R_\lambda}$ ,  $\neg\varphi$  is satisfiable. Since (cut) is sound, then every application of it to an open branch in a tableau for  $\neg\varphi$  leads to a satisfiable extension, contrary to our assumption.  $\square$

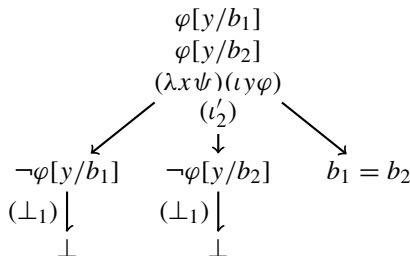
Therefore we can apply (cut) safely in  $\mathbf{TC}_{R_\lambda}$  to show derivability of other rules.

**Lemma 9**  *$\mathbf{TC}_{R_\lambda}$  and  $\mathbf{TC}'_{R_\lambda}$  are equivalent.*

**Proof** First we prove that  $(l'_2)$  is derivable in  $\mathbf{TC}_{R_\lambda}$ :



Below we show that  $(l_2)$  is derivable in  $\mathbf{TC}'_{R_\lambda}$ :



In a similar way we prove derivability of (L') in  $\mathbf{TC}_{R_\lambda}$  (again with (cut)) and derivability of (L) in  $\mathbf{TC}'_{R_\lambda}$  (without (cut)). Hence the two calculi are equivalent, both cut-free and analytic.  $\square$

Thanks to Lemma 9 we can use  $\mathbf{TC}'_{R_\lambda}$  for proving the interpolation property for  $\mathbf{L}_{R_\lambda}$ .

Let us consider closed tableaux for valid implications in  $\mathbf{TC}'_{R_\lambda}$ . Each tableau can be mechanically transformed into a *biased* tableau in the following way. Without loss of generality we assume that at the root of a proof tree of a valid implication we have a formula of the form  $\varphi \wedge \neg\psi$ . We delete the root, replace  $\varphi$  with  $\mathbf{L} \varphi$  and  $\neg\psi$  with  $\mathbf{R}\neg\psi$ , and continue the process of assigning prefixes  $\mathbf{L}, \mathbf{R}$ : for each application of a rule we precede with  $\mathbf{L}$  all conclusions of the premise prefixed with  $\mathbf{L}$  and with  $\mathbf{R}$  all conclusions of the  $\mathbf{R}$ -premise. This way all formulas, save  $\perp$  at the end of each branch, are signed in a way that makes explicit their ancestry: they follow either from the antecedent  $\mathbf{L} \varphi$  or from the succedent  $\mathbf{R} \psi$  of the original implication.

**Theorem 10** (Craig interpolation)  $\mathbf{L}_{R_\lambda}$  enjoys the Craig interpolation property.

**Proof** Let us recall that we are dealing with sentences only, so  $\varphi, \psi$  contain no free variables. To simplify the presentation, we will be using freely non-primitive constants  $\vee, \rightarrow, \exists$ , assuming tacitly that in respective tableau proofs their definitions are used. We build up an interpolant constructively, starting from each occurrence of  $\perp$  at the end of a branch, and going up the tree. In general, at each stage we consider the last applied rule and having already established interpolants for conclusions of the applied rule we extract an interpolant for the premise with respect to all formulas which are above on the branch. Thus, the general scheme is:

If  $\chi_1, \dots, \chi_k$  are interpolants for  $\Gamma \cup \{\Psi_1\}, \dots, \Gamma \cup \{\Psi_k\}$ , then  $\mathbf{I}(\chi_1, \dots, \chi_k)$  is an interpolant for  $\Gamma \cup \{\varphi\}$ , where  $\varphi$  is the premise of the applied rule and  $\Psi_1, \dots, \Psi_k$  are all the (sets of) conclusions, and  $\Gamma$  is the set of all formulas on the branch above the premise.

Clearly, the specific rules for calculating interpolants are in two versions for each rule: the  $\mathbf{L}$ -variant with  $\mathbf{L}$ , or the  $\mathbf{R}$ -variant with  $\mathbf{R}$  assigned to the premise and conclusions (in the case of  $(\perp_1)$  there are four combinations). Let  $\gamma_1, \dots, \gamma_n$  be the set of all formulas such that  $\mathbf{L} \gamma_i \in \Gamma$ , and let  $\delta_1, \dots, \delta_m$  be the set of all formulas such that  $\mathbf{R} \delta_i \in \Gamma$ . For each rule we are showing that for its  $\mathbf{L}$ -variant:

If, for every  $i \leq k, \models \psi_i \wedge \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \chi_i$  and  $\models \chi_i \rightarrow \neg\delta_1 \vee \dots \vee \neg\delta_m$ , then  $\models \varphi \wedge \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \mathbf{I}(\chi_1, \dots, \chi_k)$

and for the  $\mathbf{R}$ -variant:

If, for every  $i \leq k, \models \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \chi_i$  and  $\models \chi_i \rightarrow \neg\delta_1 \vee \dots \vee \neg\delta_m \vee \neg\psi_i$ , then  $\models \mathbf{I}(\chi_1, \dots, \chi_k) \rightarrow \neg\delta_1 \vee \dots \vee \neg\delta_m \vee \neg\varphi$

Below we state the principles for calculating interpolants for the specific rules of  $\mathbf{TC}'_{R_\lambda}$ . For the remaining rules they can be found in Fitting's work (1996).

$(\perp_2^{\mathbf{L}})$   $\perp$  is an interpolant for  $\Gamma \cup \{\mathbf{L} b \neq b\}$ .



- $(\perp_2^R)$   $\top$  is an interpolant for  $\Gamma \cup \{R b \neq b\}$ .
- $(L^L)$  If  $\chi_1$  is an interpolant for  $\Gamma \cup \{L b_1 \neq b_2\}$  and  $\chi_2$  is an interpolant for  $\Gamma \cup \{L \varphi[b_1//b_2]\}$ , then  $\forall x(\chi_1 \vee \chi_2[b_2/x])$  is an interpolant for  $\Gamma \cup \{L \varphi[b_1]\}$ .
- $(L^R)$  If  $\chi_1$  is an interpolant for  $\Gamma \cup \{R b_1 \neq b_2\}$  and  $\chi_2$  is an interpolant for  $\Gamma \cup \{R \varphi[b_1//b_2]\}$ , then  $\exists x(\chi_1 \wedge \chi_2[b_2/x])$  is an interpolant for  $\Gamma \cup \{R \varphi[b_1]\}$ .
- $(L^L)$  If  $\chi$  is an interpolant for  $\Gamma \cup \{L \psi[x/b]\}$ , then  $\chi$  is an interpolant for  $\Gamma \cup \{L (\lambda x \psi)(b)\}$ .
- $(L^R)$  If  $\chi$  is an interpolant for  $\Gamma \cup \{R \psi[x/b]\}$ , then  $\chi$  is an interpolant for  $\Gamma \cup \{R (\lambda x \psi)(b)\}$ .
- $(\neg L^L)$  If  $\chi$  is an interpolant for  $\Gamma \cup \{L \neg\psi[x/b]\}$ , then  $\chi$  is an interpolant for  $\Gamma \cup \{L \neg((\lambda x \psi)(b))\}$ .
- $(\neg L^R)$  If  $\chi$  is an interpolant for  $\Gamma \cup \{R \neg\psi[x/b]\}$ , then  $\chi$  is an interpolant for  $\Gamma \cup \{R \neg((\lambda x \psi)(b))\}$ .
- $(l_1^L)$  If  $\chi$  is an interpolant for  $\Gamma \cup \{L \psi[x/a], L \varphi[x/a]\}$ , then  $\chi$  is an interpolant for  $\Gamma \cup \{L (\lambda x \psi)(\iota y \varphi)\}$ .
- $(l_1^R)$  If  $\chi$  is an interpolant for  $\Gamma \cup \{R \psi[x/a], R \varphi[x/a]\}$ , then  $\chi$  is an interpolant for  $\Gamma \cup \{R (\lambda x \psi)(\iota y \varphi)\}$ .
- $(l_2^L)$  If  $\chi_1$  is an interpolant for  $\Gamma \cup \{L \neg\varphi[y/b_1]\}$ ,  $\chi_2$  is an interpolant for  $\Gamma \cup \{L \neg\varphi[y/b_2]\}$  and  $\chi_3$  is an interpolant for  $\Gamma \cup \{L b_1 = b_2\}$ , then  $\forall x \forall y(\chi_1 \vee \chi_2 \vee \chi_3[b_1/x, b_2/y])$  is an interpolant for  $\Gamma \cup \{L (\lambda x \psi)(\iota y \varphi)\}$ .
- $(l_2^R)$  If  $\chi_1$  is an interpolant for  $\Gamma \cup \{R \neg\varphi[y/b_1]\}$ ,  $\chi_2$  is an interpolant for  $\Gamma \cup \{R \neg\varphi[y/b_2]\}$  and  $\chi_3$  is an interpolant for  $\Gamma \cup \{R b_1 = b_2\}$ , then  $\exists x \exists y(\chi_1 \wedge \chi_2 \wedge \chi_3[b_1/x, b_2/y])$  is an interpolant for  $\Gamma \cup \{R (\lambda x \psi)(\iota y \varphi)\}$ .
- $(\neg l_1^L)$  If  $\chi_1$  is an interpolant for  $\Gamma \cup \{L \neg\psi[y/b]\}$ ,  $\chi_2$  is an interpolant for  $\Gamma \cup \{L \neg\varphi[y/a]\}$  and  $\chi_3$  is an interpolant for  $\Gamma \cup \{L a \neq b\}$ , then  $\forall x(\chi_1 \vee \chi_2 \vee \chi_3[b/x])$  is an interpolant for  $\Gamma \cup \{L \neg((\lambda x \psi)(\iota y \varphi))\}$ .
- $(\neg l_1^R)$  If  $\chi_1$  is an interpolant for  $\Gamma \cup \{R \neg\psi[y/b]\}$ ,  $\chi_2$  is an interpolant for  $\Gamma \cup \{R \neg\varphi[y/a]\}$  and  $\chi_3$  is an interpolant for  $\Gamma \cup \{R a \neq b\}$ , then  $\exists x(\chi_1 \wedge \chi_2 \wedge \chi_3[b/x])$  is an interpolant for  $\Gamma \cup \{R \neg((\lambda x \psi)(\iota y \varphi))\}$ .

Let us check the hardest case of  $(l_2')$ . The remaining ones can be shown by using an analogous argument. For the L-variant we assume that:

- 1  $\models \neg\varphi[y/b_1] \wedge \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \chi_1$
- 2  $\models \chi_1 \rightarrow \neg\delta_1 \vee \dots \vee \neg\delta_m$
- 3  $\models \neg\varphi[y/b_2] \wedge \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \chi_2$
- 4  $\models \chi_2 \rightarrow \neg\delta_1 \vee \dots \vee \neg\delta_m$
- 5  $\models b_1 = b_2 \wedge \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \chi_3$
- 6  $\models \chi_3 \rightarrow \neg\delta_1 \vee \dots \vee \neg\delta_m$

On this basis we can show that  $\forall x \forall y(\chi_1 \vee \chi_2 \vee \chi_3[b_1/x, b_2/y])$  is the required interpolant. Let us assume that one or both of  $b_1, b_2$  occur in any of  $\chi_1, \chi_2, \chi_3$ , but not in  $\gamma_1, \dots, \gamma_n$ . It follows that they must also occur in  $\neg\delta_1 \vee \dots \vee \neg\delta_m$ . If the assumption is not satisfied, then the universal quantification of  $\chi_1 \vee \chi_2 \vee \chi_3$  is either unnecessary (if they occur also in  $\gamma_1, \dots, \gamma_n$ ) or void (if they are not in  $\chi_1, \chi_2, \chi_3$ ), and hence, also unnecessary. From 2, 4, 6 it obviously follows that

$\models \forall x \forall y (\chi_1 \vee \chi_2 \vee \chi_3 [b_1/x, b_2/y]) \rightarrow \neg \delta_1 \vee \dots \vee \neg \delta_m$ . We show, using  $\mathbf{TC}_{R_\lambda}$ , that from 1, 3, 5 it follows that  $\models (\lambda x \psi)(ty\varphi) \wedge \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \forall x \forall y (\chi_1 \vee \chi_2 \vee \chi_3 [b_1/x, b_2/y])$ . Assume, towards a contradiction, that this implication is not valid, that is, by completeness of  $\mathbf{TC}_{R_\lambda}$ , there is an open branch in the tableau containing  $(\lambda x \psi)(ty\varphi), \gamma_1, \dots, \gamma_n, \neg \forall x \forall y (\chi_1 \vee \chi_2 \vee \chi_3 [b_1/x, b_2/y])$ . Since  $b_1, b_2$  are not present on the branch, we can add  $\neg \chi_1, \neg \chi_2, \neg \chi_3$  to the branch with these parameters as fresh. Again, by completeness of  $\mathbf{TC}_{R_\lambda}$  we know that each of 1, 3, 5 has a proof, hence applying (cut) thrice with these implications and their negations as cut formulas we obtain their occurrences on the open branch. By applying systematically branching rules to them we obtain  $\varphi[y/b_1], \varphi[y/b_2], b_1 \neq b_2$ ; all other branches containing  $\neg \gamma_1, \dots, \neg \gamma_n, \chi_1, \chi_2, \chi_3$  must be closed. Eventually, an application of  $(t_2)$  yields  $b_1 = b_2$  on the branch, thus making it closed, so  $\models (\lambda x \psi)(ty\varphi) \wedge \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \forall x \forall y (\chi_1 \vee \chi_2 \vee \chi_3 [b_1/x, b_2/y])$ , contrary to our assumption. Since  $\forall x \forall y (\chi_1 \vee \chi_2 \vee \chi_3 [b_1/x, b_2/y])$  contains only predicates and parameters which are common to  $(\lambda x \psi)(ty\varphi) \wedge \gamma_1 \wedge \dots \wedge \gamma_n$  and  $\neg \delta_1 \vee \dots \vee \neg \delta_m$ , we are done.

For the R-variant we assume that:

- 1'.  $\models \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \chi_1$ ;
- 2'.  $\models \chi_1 \rightarrow \neg \delta_1 \vee \dots \vee \neg \delta_m \vee \varphi[y/b_1]$ ;
- 3'.  $\models \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \chi_2$ ;
- 4'.  $\models \chi_2 \rightarrow \neg \delta_1 \vee \dots \vee \neg \delta_m \vee \varphi[y/b_2]$ ;
- 5'.  $\models \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \chi_3$ ;
- 6'.  $\models \chi_3 \rightarrow \neg \delta_1 \vee \dots \vee \neg \delta_m \vee b_1 \neq b_2$ ;

On this basis we can show that  $\exists x \exists y (\chi_1 \wedge \chi_2 \wedge \chi_3 [b_1/x, b_2/y])$  is the required interpolant. Let us assume that one or both of  $b_1, b_2$  occur in any of  $\chi_1, \chi_2, \chi_3$ , but not in  $\delta_1, \dots, \delta_m$ . It follows that they must also occur in  $\gamma_1 \wedge \dots \wedge \gamma_n$ . If, on the other hand, our assumption is false, then the existential quantification of  $\chi_1 \wedge \chi_2 \wedge \chi_3$  is either unnecessary (if one or both of  $b_1, b_2$  occur also in  $\delta_1, \dots, \delta_m$ ) or void (if they are not in  $\chi_1, \chi_2, \chi_3$ ), and hence, also unnecessary. From 1', 3', 5' it straightforwardly follows that  $\models \gamma_1 \wedge \dots \wedge \gamma_n \rightarrow \exists x \exists y (\chi_1 \wedge \chi_2 \wedge \chi_3 [b_1/x, b_2/y])$ . We show, using  $\mathbf{TC}_{R_\lambda}$ , that 2', 4', 6' imply  $\models \exists x \exists y (\chi_1 \wedge \chi_2 \wedge \chi_3 [b_1/x, b_2/y]) \rightarrow \neg \delta_1 \vee \dots \vee \neg \delta_m \vee \neg (\lambda x \psi)(ty\varphi)$ . For the sake of contradiction assume that the implication is false. Hence, by completeness of  $\mathbf{TC}_{R_\lambda}$ , there is an open branch in a tableau containing  $\exists x \exists y (\chi_1 \wedge \chi_2 \wedge \chi_3 [b_1/x, b_2/y]), \delta_1, \dots, \delta_m, (\lambda x \psi)(ty\varphi)$ . Since  $b_1, b_2$  are absent from the branch, we can add  $\chi_1, \chi_2, \chi_3$  to the branch with these parameters as fresh. Again, from completeness of  $\mathbf{TC}_{R_\lambda}$  we derive that each of 2', 4', 6' has a proof, hence by applying (cut) three times with these implications and their negations as cut formulas we make them occur on an open branch. By systematic applications of branching rules to them we obtain  $\varphi[y/b_1], \varphi[y/b_2], b_1 \neq b_2$  on an open branch. All other branches containing  $\neg \chi_1, \neg \chi_2, \neg \chi_3, \neg \delta_1, \dots, \neg \delta_m$  must be closed. Finally, an application of  $(t_2)$  introduces  $b_1 = b_2$  to the branch and closes it, therefore  $\models \exists x \exists y (\chi_1 \wedge \chi_2 \wedge \chi_3 [b_1/x, b_2/y]) \rightarrow \neg \delta_1 \vee \dots \vee \neg \delta_m \vee \neg (\lambda x \psi)(ty\varphi)$ , contradicting the assumption. Since  $\exists x \exists y (\chi_1 \wedge \chi_2 \wedge \chi_3 [b_1/x, b_2/y])$  contains only predicates and parameters occurring both in  $\gamma_1 \wedge \dots \wedge \gamma_n$  and  $\neg \delta_1 \vee \dots \vee \neg \delta_m \vee \neg (\lambda x \psi)(ty\varphi)$ , it concludes the proof.  $\square$

As a consequence of Theorem 10 we get:

**Theorem 11** (Beth definability)  $L_{R_\lambda}$  enjoys the Beth definability property.

Implicit definability follows almost immediately from explicit definability. If we assume (ii), then the equivalence form (i) can be obtained by the assumption that  $c$  (and thus,  $c'$ ) does not occur in the defining formula. For the converse implication we employ the Craig interpolation property and the deduction theorem for  $L$  to show that an interpolant obtained in a series of derivations can play the role of the formula on the right-hand side of the equivalence in (ii). See Andrews (2002, Theorem 4200) for details.

Knowing that  $L_{R_\lambda}$  satisfies the Beth definability property results in a straightforward method of determining, for any signature  $\Sigma = (\text{PRED}, \text{CONS}, \text{ar})$ , any theory  $\text{Th}$  over  $\Sigma$  and any constant  $c \in \text{CONS}$ , whether  $c$  can be defined by a formula  $\psi$  over  $\Sigma' = (\text{PRED}, \text{CONS} \setminus \{c\}, \text{ar})$  under  $\text{Th}$ . In other words, we can decide whether there exists a formula that can form a definite description satisfied by the object that  $c$  denotes. Indeed, it suffices to check for implicit definability of  $c$ , that is, check if a tableau with the following formula at the root is closed:

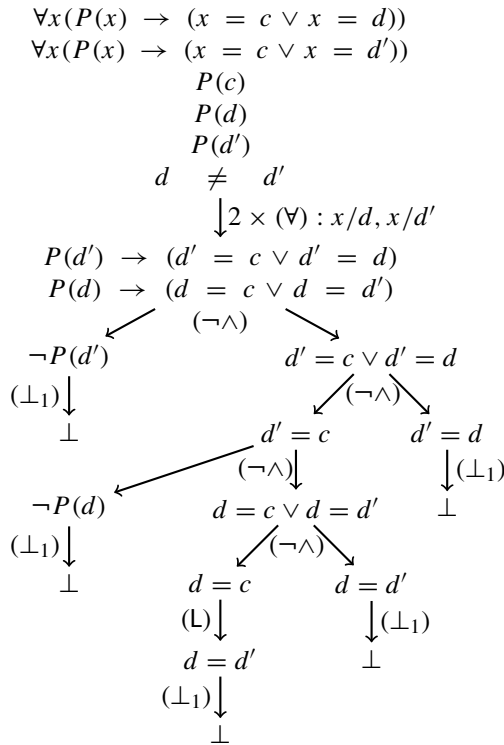
$$\bigwedge (\text{Th} \cup \text{Th}') \wedge c \neq c',$$

where  $\text{Th}'$  is defined like at the beginning of the section. Since, by Theorem 11, implicit definability implies explicit definability, we can replace such a constant with definite description  $\iota x \psi$ , where  $\psi$  is a definiens of  $c$ , whenever such a replacement results in a syntactically correct expression.

**Example 7** Consider a theory  $\text{Th}$  which provides characteristics of two individuals: Charles and Dana. The one thing that  $\text{Th}$  stipulates is that only Charles and Dana are politicians. Formally:

$$\text{Th} = \{\forall x (P(x) \rightarrow (x = c \vee x = d)), P(c), P(d)\}.$$

Using  $\text{TC}_{R_\lambda}$  it is not difficult to check that  $d$  is implicitly definable in  $\text{Th}$  (in the proof tree below we exploit the definitions:  $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$  and  $\varphi \rightarrow \psi := \neg(\varphi \wedge \neg\psi)$ ):



One can easily verify that, since Th does not specify whether Charles and Dana are the same person,  $d$  in Th is unambiguously characterised by the formula  $P(x) \wedge (x \neq c \vee \neg \exists y(y \neq x \wedge P(y)))$  saying that either Dana is a politician distinct from Charles or the only politician that exists, thus it can be replaced with  $\iota x(P(x) \wedge (x \neq c \vee \neg \exists y(y \neq x \wedge P(y))))$  in every syntactically permitted context. In fact, the above tableau reveals that even in the reduct of Th only consisting of  $\forall x(P(x) \rightarrow (x = c \vee x = d))$  and  $P(d)$ ,  $d$  is implicitly (and therefore, explicitly) definable, for  $P(c)$  does not take active part in closing the tree. Indeed, in such a reduced theory  $d$  can be defined explicitly by the same formula. If, on the other hand, we remove  $P(d)$  from Th and keep  $P(c)$  instead,  $d$  will no longer be implicitly (and thus, explicitly) definable in such a reduct, as we will no longer have a means to close the branches with  $\neg P(d)$  and  $\neg P(d')$ .

### 8 Discussion and future work

$TC_{R_\lambda}$  is an accessible, well-behaved tableau calculus which represents essential features of the Russellian approach to definite descriptions and at the same time avoids its shortcomings. Moreover, the proposed methodology is open for further extensions to several other theories of definite descriptions. In particular, we can use it to formalise the approaches to definite descriptions developed in the area of free logics.

Indrzejczak and Zawidzki (2021) proposed tableau systems for the minimal theory of definite descriptions expressed in several variants of free logics. The language of these systems is standard in the sense that predicate abstracts are not admitted and definite descriptions are directly used as arguments of predicates on a par with variables and individual constants. Moreover, the characterisation of definite descriptions is based on the weaker principle, often called *Lambert's axiom*. Only in negative free logic Lambert's axiom can be proved equivalent to Russell's axiom being the basis of the approach presented in this paper. Rules for definite descriptions involve equalities with definite descriptions as arguments and to obtain completeness we need a restricted form of cut on equalities with definite descriptions already present on the branch. Introducing the lambda-operator and restricting occurrences of definite descriptions to arguments of lambda-atoms has a potential of simplifying the mentioned tableau systems for these theories. Roughly, it seems that to adjust  $\mathbf{TC}_{R_\lambda}$  to positive free logics we would only need to slightly modify the rules for quantifiers and for lambda-atoms. If we want negative free logics to be covered by a transformed version of  $\mathbf{TC}_{R_\lambda}$ , we must additionally alter the rules for equality and add some extra rules expressing special denotation principles. An advantage offered by the setting based on a modified version of  $\mathbf{TC}_{R_\lambda}$  for free logics over previous tableau systems by Indrzejczak and Zawidzki (2021) is that (the restricted form of) cut is no longer required, which decreases the search space in a proof.

Another promising field of application of the presented setting is modal logic, where tableau systems with lambda- and iota-terms were already examined by Fitting and Mendelsohn (1998) and then characterised by means of sequent calculi by Orlandelli (2021) and Indrzejczak (2020a). The application of lambda-abstracts to the problem of distinguishing between *de dicto* and *de re* reading of a modality is direct:  $\Box(\lambda x\varphi)(t)$  and  $(\lambda x\Box\varphi)(t)$ . Fitting and Mendelsohn introduced labelled tableaux and applied the lambda-operator, but did not characterise definite descriptions by means of well-behaved rules. Both Orlandelli's and Indrzejczak's approaches rely on the same theory of definite descriptions, again based on Lambert's axiom, but, contrary to Fitting and Mendelsohn, provide sequent calculi with well-defined rules. The difference between them is that in the former external labels are used and rules for definite descriptions are built on the basis of a special denotation predicate, whereas in the latter hybrid logic and a special equality predicate are exploited to characterise definite descriptions. In both cases cut elimination is proved and lambda-predicates are introduced mainly for dealing with modal scoping difficulties. In the case of the system introduced in this paper the role of the lambda-operator is rather to solve the known problems caused by the Russellian approach to definite descriptions and to introduce tighter constraints on treating descriptions as terms. In particular, in all mentioned alternative approaches definite descriptions are semantically treated as terms, whereas in the present approach, they are characterised semantically only as arguments of lambda atoms. This is what makes metalogical proofs almost standard in contrast to proofs conducted within approaches where definite descriptions are semantically characterised as independent terms. Nevertheless, since the solutions proposed in this paper, with particular focus on restricting the part of the syntax revolved around definite descriptions, were not considered in the aforementioned works, it opens a path for cross-fertilisation of the techniques developed in all presented approaches.

Also, notice that to prove interpolation we had to refer to admissibility of cut on the basis of completeness of the cut-free version of the calculus. From the theoretical standpoint it is desirable to have the cut elimination theorem proved constructively for this calculus. This problem has recently been solved by Indrzejczak and Kürbis (2022).

An interesting task for future research would be extending the present approach to intuitionistic and, possibly, intermediate logics. Intuitionistic logic could be handled by simply adding labels representing states in an S4 modal frame, and slightly modifying the rules to account for this extra dimension. Feasibility of this enterprise becomes evident when we use sequent calculus instead of tableaux. Since all rules specific for definite descriptions have never more than one formula in the succedent (in the tableau setting: negated formulas), we can simply add them to the standard intuitionistic version of sequent calculus. Tackling some known intermediate logics which have modal frames based on an order relation in a uniform fashion also seems achievable.

Finally, from the practical standpoint the presented tableau system requires a proper implementation for its efficiency to be firmly tested.

**Acknowledgements** The research leading to these results received funding from the National Science Centre, Poland, under Grant Agreement No. DEC- 2017/25/B/HS1/01268). The second author is supported by the EPSRC projects OASIS (EP/S032347/1), AnaLOG (EP/P025943/1), and UK FIRES (EP/S019111/1), the SIRIUS Centre for Scalable Data Access, and Samsung Research UK.

## Declarations

**Conflict of interest** The authors have no other competing interests to declare that are relevant to the content of this article.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Andrews, P. B. (2002). *An introduction to mathematical logic and type theory: To truth through proof* (Vol. 27). Springer. <https://doi.org/10.1007/978-94-015-9934-4>
- Artale, A., Mazzullo, A., Ozaki, A., Wolter, F. (2021, 11). On Free Description Logics with Definite Descriptions. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning* (pp. 63–73). IJCAI Organization. <https://doi.org/10.24963/kr.2021/7>.
- Beeson, M. J. (1985). *Foundations of constructive mathematics* (Vol. 6). Springer. <https://doi.org/10.1007/978-3-642-68952-9>
- Bencivenga, E., Lambert, K., & Van Fraassen, B. C. (1986). *Logic, bivalence and denotation*. Ridgeview Publishing Company.
- Bernays, P. (1985). *Axiomatic set theory* (Vol. 21). North-Holland. [https://doi.org/10.1016/S0049-237X\(08\)71565-9](https://doi.org/10.1016/S0049-237X(08)71565-9)

- Blumson, B. (2020). Anselm's God in Isabelle/HOL. [https://www.isa-afp.org/browser\\_info/current/AFP/AnselmGod/document.pdf](https://www.isa-afp.org/browser_info/current/AFP/AnselmGod/document.pdf).
- Bohrer, B., Fernández, M., & Platzer, A. (2019).  $dL_1$ : Definite descriptions in differential dynamic logic. In P. Fontaine (Ed.), *Automated Deduction—CADE 27* (pp. 94–110). Springer International Publishing. [https://doi.org/10.1007/978-3-030-29436-6\\_6](https://doi.org/10.1007/978-3-030-29436-6_6)
- Bostock, D. (1997). *Intermediate logic*. Clarendon Press.
- Bressan, A. (1972). *A general interpreted modal calculus*. Yale University Press.
- Carlström, J. (2005). Interpreting descriptions in intensional type theory. *The Journal of Symbolic Logic*, 70(2), 488–514. <https://doi.org/10.2178/jsl/1120224725>
- Carnap, R. (1947). *Meaning and necessity. A study in semantics and modal logic*. The University of Chicago Press.
- Czermak, J. (1974). A logical calculus with descriptions. *Journal of Philosophical Logic*, 3, 211–228. <https://doi.org/10.1007/BF00247223>
- Donnellan, K. S. (1966). Reference and definite descriptions. *The Philosophical Review*, 75(3), 281–304. <https://doi.org/10.2307/2183143>
- Ebbinghaus, H.-D., Flum, J., & Thomas, W. (1994). *Mathematical logic*. Springer. <https://doi.org/10.1007/978-1-4757-2355-7>
- Feferman, S. (1995). Definedness. *Erkenntnis*, 43, 295–320. <https://doi.org/10.1007/BF01135376>
- Fitting, M. (1975). A modal logic epsilon-calculus. *Notre Dame Journal of Formal Logic*, 16(1), 1–16. <https://doi.org/10.1305/ndjfl/1093891609>
- Fitting, M. (1996). *First-order logic and automated theorem proving*. Springer. <https://doi.org/10.1007/978-1-4612-2360-3>
- Fitting, M., & Mendelsohn, R. L. (1998). *First-order modal logic* (Vol. 277). Springer. <https://doi.org/10.1007/978-94-011-5292-1>
- Francez, N., & Więckowski, B. (2014). A proof-theoretic semantics for contextual definiteness. In E. Moriconi & L. Tesconi (Eds.), *Proceedings of the second pisa colloquium in logic, language and epistemology* (pp. 181–212). Edizioni ETS.
- Frege, G. (1893). *Grundgesetze der Arithmetik. Begriffsschriftlich abgeleitet. I. Band*. Hermann Pohle.
- Garson, J. W. (2006). *Modal logic for philosophers*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511617737>
- Graff, D. (2001). Descriptions as predicates. *Philosophical Studies*, 102, 1–42. <https://doi.org/10.1023/A:1010379409594>
- Gratzl, N. (2015). Incomplete symbols—Definite descriptions revisited. *Journal of Philosophical Logic*, 44, 489–506. <https://doi.org/10.1007/s10992-014-9339-1>
- Gumb, R. D. (2001). An extended joint consistency theorem for a nonconstructive logic of partial terms with definite descriptions. *Studia Logica*, 69(2), 279–292. <https://doi.org/10.1023/A:1013822008159>
- Hähnle, R., & Beckert, B. (1999). Proof confluent tableau calculi. In N. V. Murray (Ed.), *Automated reasoning with analytic tableaux and related methods* (pp. 34–35). Springer. [https://doi.org/10.1007/3-540-48754-9\\_7](https://doi.org/10.1007/3-540-48754-9_7)
- Hermes, H. (1973). *Introduction to mathematical logic*. Springer. <https://doi.org/10.1007/978-3-642-87132-0>
- Hilbert, D., & Bernays, P. (1968). *Grundlagen der Mathematik I* (Vol. 40). Springer. <https://doi.org/10.1007/978-3-642-86894-8>
- Indrzejczak, A. (2018). Cut-free modal theory of definite descriptions. In N. Bezhanishvili, M. D'Agostino, & T. Studer (Eds.), *Advances in Modal Logic 12* (pp. 359–378). College Publications.
- Indrzejczak, A. (2019). Fregean description theory in proof-theoretic setting. *Logic and Logical Philosophy*, 28(1), 137–155. <https://doi.org/10.12775/LLP.2018.008>
- Indrzejczak, A. (2020). Existence, definedness and definite descriptions in hybrid modal logic. In N. Olivetti, R. Verbrugge, S. Negri, & G. Sandu (Eds.), *Advances in Modal Logic 13*. College Publications.
- Indrzejczak, A. (2020). Free definite description theory—Sequent calculi and cut elimination. *Logic and Logical Philosophy*, 29(4), 505–539. <https://doi.org/10.12775/LLP.2020.020>
- Indrzejczak, A. (2021). Free logics are cut-free. *Studia Logica*, 109(4), 859–886. <https://doi.org/10.1007/s11225-020-09929-8>
- Indrzejczak, A. (2021). Russellian definite description theory—A proof-theoretic approach. *The Review of Symbolic Logic online first*. <https://doi.org/10.1017/S1755020321000289>
- Indrzejczak, A., & Kürbis, N. (2022). A cut-free, sound and complete formalisation of a Russellian theory of definite descriptions. submitted.

- Indrzejczak, A., & Zawidzki, M. (2021). Tableaux for free logics with descriptions. In *Proceedings of the 30th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods* (pp. 56–73). Springer-Verlag, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-030-86059-2\\_4](https://doi.org/10.1007/978-3-030-86059-2_4).
- Kalish, D., & Montague, R. (1957). Remarks on descriptions and natural deduction. *Archiv für Mathematische Logik und Grundlagenforschung*, 3, 50–64. <https://doi.org/10.1007/BF01977152>
- Kalish, D., Montague, R., & Mar, G. (1980). *Logic. Techniques of formal reasoning* (2nd ed.). Oxford University Press.
- Kripke, S. (1977). Speaker' reference and semantic reference. *Midwest Studies in Philosophy*, 2(1), 255–276. <https://doi.org/10.1111/j.1475-4975.1977.tb00045.x>
- Kürbis, N. (2019). A binary quantifier for definite descriptions in intuitionist negative free logic: Natural deduction and normalization. *Bulletin of the Section of Logic*, 48(2), 81–97. <https://doi.org/10.18778/0138-0680.48.2.01>
- Kürbis, N. (2019). Two treatments of definite descriptions in intuitionist negative free logic. *Bulletin of the Section of Logic*, 48(4), 299–317. <https://doi.org/10.18778/0138-0680.48.4.04>
- Kürbis, N. (2021). A binary quantifier for definite descriptions for cut free free logics. *Studia Logica*. <https://doi.org/10.1007/s11225-021-09958-x>
- Kürbis, N. (2021). Definite descriptions in intuitionist positive free logic. *Logic and Logical Philosophy*, 30(2), 327–358. <https://doi.org/10.12775/LLP.2020.024>
- Kürbis, N. (2021c). Proof-theory and semantics for a theory of definite descriptions. In *Proceedings of the 30th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods* (pp. 95–111). Springer-Verlag, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-030-86059-2\\_6](https://doi.org/10.1007/978-3-030-86059-2_6).
- Ludlow, P. (2021). Descriptions. E.N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Fall 2021 ed.). Metaphysics Research Lab, Stanford University. <https://plato.stanford.edu/archives/fall2021/entries/descriptions/>.
- Neale, S. (1990). *Descriptions*. MIT Press.
- Oppenheimer, P. E., & Zalta, E. N. (2011). A computationally-discovered simplification of the ontological argument. *Australasian Journal of Philosophy*, 89(2), 333–349. <https://doi.org/10.1080/00048401003674482>
- Orlandelli, E. (2021). Labelled calculi for quantified modal logics with definite descriptions. *Journal of Logic and Computation*, 31(3), 923–946. <https://doi.org/10.1093/logcom/exab018>
- Pelletier, F. J., & Linsky, B. (2009). Russell vs. Frege on definite descriptions as singular terms. In F. J. Pelletier & B. Linsky (Eds.), *Russell vs. Meinong: The legacy of On denoting* (pp. 40–64). London: Routledge.
- Quine, W. V. O. (1982). *Mathematical logic* (Revised). Harvard University Press.
- Rosser, J. B. (1978). *Logic for mathematicians*. Dover Publications.
- Russell, B. (1905). On denoting. *Mind*, XIV(4), 479–493. <https://doi.org/10.1093/mind/XIV.4.479>
- Russell, B. (1920). *Introduction to mathematical philosophy* (2nd ed.). George Allen & Unwin, Ltd., London. <http://people.umass.edu/klement/imp/imp.pdf>.
- Scales, R. (1969). *Attribution and existence* (Unpublished doctoral dissertation). University of California.
- Scott, D. (1979). Identity and existence in intuitionistic logic. In M. Fourman, C. Mulvey, & D. Scott (Eds.), *Applications of Sheaves: Proceedings of the Research Symposium on Applications of Sheaf Theory to Logic, Algebra, and Analysis, Durham, July 9–21, 1977* (pp. 660–696). Berlin, Heidelberg: Springer. <https://doi.org/10.1007/BFb0061839>
- Smullyan, R. M. (1968). *First-order logic*. Springer. <https://doi.org/10.1007/978-3-642-86718-7>
- Stalnaker, R. C., & Thomason, R. H. (1968). Abstraction in first-order modal logic I. *Theoria*, 34(3), 203–207. <https://doi.org/10.1111/j.1755-2567.1968.tb00351.x>
- Stenlund, S. (1973). The logic of description and existence (No. 18). Filosofiska Föreningen och Filosofiska Institutionen vid Uppsala Universitet, Uppsala.
- Stenlund, S. (1975). Descriptions in intuitionistic logic. S. Kanger (Ed.), In *Proceedings of the Third Scandinavian Logic Symposium* (Vol. 82, pp. 197–212). Elsevier, Amsterdam, Oxford. [https://doi.org/10.1016/S0049-237X\(08\)70732-8](https://doi.org/10.1016/S0049-237X(08)70732-8).
- Strawson, P. F. (1950). On referring. *Mind*, LIX(235), 320–344. <https://doi.org/10.1093/mind/LIX.235.320>
- Tennant, N. (1978). *Natural logic*. Edinburgh University Press.
- Tennant, N. (2004). A general theory of abstraction operators. *The Philosophical Quarterly*, 54(214), 105–133. <https://doi.org/10.1111/j.0031-8094.2004.00344.x>
- Whitehead, A. N., & Russell, B. (1910). *Principia mathematica* (Vol. I). Cambridge University Press. <https://doi.org/10.1017/CBO9780511623585>



**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.