A Tableau System for First-Order Epistemic Logic with Standard Names

Jens Claßen[1], Torben Braüner[1]

[1]Department of People and Technology, Roskilde University, Denmark

Abstract

Levesque and Lakemeyer proposed a first-order epistemic logic called $\mathcal{KL}$ for the purpose of knowledge representation and reasoning in knowledge-based systems. A characteristic feature of this logic is that it uses a countably infinite set of so-called standard names, which are syntactically treated like constants, but which are also assumed to comprise the fixed universe of discourse, under a substitutional interpretation of quantification. So far, the only sound and complete proof system for $\mathcal{KL}$ is a Hilbert-style axiom system due to Levesque and Lakemeyer. Tableau systems have been presented by Rosati for its propositional fragment, and recently by the authors of this paper for its non-modal fragment. In this paper, we present a tableau system for the full first-order version of $\mathcal{KL}$, and show its soundness and completeness.

Keywords

Tableau, First-Order Epistemic Logic, Standard Names, Substitutional Quantification

1. Introduction

Levesque and Lakemeyer proposed a first-order modal logic of knowledge called $\mathcal{KL}$ [1, 2]. Among other things, it is the basis for their logic of "only-knowing" called $\mathcal{OL}$, which has been found useful to capture various forms of non-monotonic reasoning [3, 4], and variants thereof have been developed that deal with actions and change [5, 6], multi-agent systems [7, 8], limited belief [9, 10], probabilistic belief [11, 12], and more. A characteristic feature of this family of logics is that they make use of a countably infinite set of so-called standard names, which serve a dual purpose: On the one hand, they are part of the language and can be used syntactically like constants; on the other hand, they are also assumed to comprise the fixed universe of discourse, under a substitutional interpretation of quantification. As we will see, this assumption not only significantly simplifies proofs for meta-theoretic properties such as soundness and completeness, but it can also be exploited for the purpose of defining specific reasoning techniques.

Due to not being compact, these logics over standard names cannot be reduced to classical first-order logic under standard Tarskian semantics, and so it is somewhat surprising that so far there has been little work on examining them from the perspective of proof theory. In their book [2], Levesque and Lakemeyer present a sound and complete Hilbert-style axiom system for $\mathcal{KL}$, which is not suited for actual reasoning, but only discussed as an alternative characterization of its valid sentences. Rosati [13] studied tableau systems for the propositional fragments of $\mathcal{KL}$ and $\mathcal{OL}$, which can be viewed as variants of the standard modal logic K45; to the best of our knowledge these were never extended to the full first-order language. Also, the paper [14] gives a tableau system for K45, similar to the propositional fragment of our system, but again, the system in [14] has not been extended to the full first-order language.

Recently [15] we developed a tableau-based proof system for $\mathcal{L}$, the non-modal fragment of $\mathcal{KL}$, and showed its soundness and completeness. In this paper, we extend it to the full (first-order) version of $\mathcal{KL}$, again proving its soundness and completeness. For the latter, we adapt the notions of Hintikka sets and

ARQNL 2026: Automated Reasoning in Quantified Non-Classical Logics, 24 July 2026, Lisbon, Portugal
classen@ruc.dk (J. Claßen); torben@ruc.dk (T. Braüner)
https://jens-classen.net/ (J. Claßen); https://forskning.ruc.dk/da/persons/torben/ (T. Braüner)
0000-0002-0395-1907 (J. Claßen); 0000-0003-4582-1702 (T. Braüner)


systematic tableaux. The completeness proof in the present paper is simpler and more straight-forward than the proof in [15], even though the latter is only for the non-modal fragment.

The remainder of this paper is organized as follows. The following section introduces $\mathcal{KL}$ formally and presents Levesque and Lakemeyer's axiom system for it. In Section 3, we then provide our tableau system for $\mathcal{KL}$ and show its soundness and completeness. The paper finishes with concluding remarks in Section 4.

2. The Logic $\mathcal{KL}$

In this section, we give a brief formal description of the logic $\mathcal{KL}$ and an associated axiom system, based on the presentation in [2].

2.1. Syntax

In terms of syntax, $\mathcal{KL}$ is a first-order modal language of knowledge with a single modal operator $\mathbf{K}$. It furthermore includes functions, equality, and standard names. The latter can be used in the same way as constant symbols, but have, as we will see, a fixed meaning. Formally:

Definition 2.1 (Vocabulary)**.** Formulas are built from a set of symbols consisting of:

1. a countably infinite supply of *variables*, written as x, y, z;
2. a countably infinite supply of *standard names* $\#1, \#2, \ldots$, written schematically as n, m;
3. countably many *predicate symbols* $P, Q, R, \ldots$ with associated arity $k \geq 0$;
4. countably many *function symbols* $f, g, h, \ldots$ with associated arity $k \geq 0$;
5. *equality* $=$ and *logical connectives* $\exists$, $\vee$, $\neg$;
6. modal operator $\mathbf{K}$.

Definition 2.2 (Terms)**.** The *terms* are the smallest set such that

1. every variable is a term;
2. every standard name is a term;
3. if f is a k-ary function symbol and $t_1, \ldots, t_k$ are terms, then $f(t_1, \ldots, t_k)$ is a term.

A term without variables is a *ground term*, and a ground term with exactly one function symbol is a *primitive term*.

Definition 2.3 (Formulas)**.** The *formulas* are the smallest set such that

1. if P is a k-ary predicate symbol and $t_1, \ldots, t_k$ are terms, then $P(t_1, \ldots, t_k)$ is an (atomic) formula;
2. if t_1, t_2 are terms, then $(t_1 = t_2)$ is a formula;
3. if ϕ, ψ are formulas, then so are $\neg\phi$ and $\phi \vee \psi$;
4. if x is a variable and ϕ a formula, then $\exists x\phi$ is a formula;
5. if ϕ is a formula, then so is $\mathbf{K}\phi$.

We treat $\phi \wedge \psi$, $\phi \supset \psi$, $\phi \equiv \psi$, and $\forall x\phi$ as the usual abbreviations. The notions of bound and free variables as well as their scope are defined as usual. By ϕ_t^x we denote the result of simultaneously replacing all free occurrences of variable x by term t. A *sentence* is a formula without free variables. A *ground atom* is an atomic formula with no variables. A *primitive atom* is a ground atom where every t_i is a standard name. We call a sentence *objective* if it does not mention $\mathbf{K}$, and *subjective* if all predicate and function symbols occur within the scope of a $\mathbf{K}$ operator.

2.2. Semantics

$\mathcal{KL}$ employs a non-standard, model-theoretic semantics based on possible worlds as described below. Intuitively, a world is a valuation of the primitive atoms $P(n_1, \ldots, n_k)$ and primitive terms $f(n_1, \ldots, n_k)$, and an epistemic state is the set of worlds the agent considers possible. Since standard names are assumed to be the (fixed) universe of discourse, any standard name n_i is interpreted by itself. Using this, we assign a meaning to any complex term and formula:

Definition 2.4 (Semantics)**.** A *world* w is a mapping from primitive terms to standard names, and from primitive atoms to truth values $\{0, 1\}$. The *value of a ground term* t *at a world* w, written as $w(t)$, is defined by

1. $w(n) = n$ for every standard name n;
2. $w(f(t_1, \ldots, t_k)) = w[f(n_1, \ldots, n_k)]$, where $n_i = w(t_i)$.

Let $\mathcal{W}$ be the set of all worlds. An *epistemic state* e is a set of worlds, i.e., $e \subseteq \mathcal{W}$. A semantic model is then given by an epistemic state e and a ("real") world w. The truth of a sentence ϕ wrt. e, w is defined inductively as:

1. $e, w \models P(t_1, \ldots, t_k)$ iff $w[P(n_1, \ldots, n_k)] = 1$, where $n_i = w(t_i)$;
2. $e, w \models (t_1 = t_2)$ iff $w(t_1)$ is the same as $w(t_2)$;
3. $e, w \models \neg\alpha$ iff it is not the case that $e, w \models \alpha$;
4. $e, w \models \alpha \vee \beta$ iff $e, w \models \alpha$ or $e, w \models \beta$;
5. $e, w \models \exists x\alpha$ iff for some name n, $e, w \models \alpha^x_n$;
6. $e, w \models \mathbf{K}\alpha$ iff for every $w' \in e$, $e, w' \models \alpha$.

We often write $e \models \sigma$ for subjective sentences σ, and $w \models \phi$ for objective sentences ϕ, since the truth of the former does not depend on w, and the truth of the latter does not depend on e (both of these intuitive claims are straightforward to prove by induction). A set of sentences Γ is called *satisfiable* just in case there is some world w and epistemic state e such that $e, w \models \alpha$ for every $\alpha \in \Gamma$. We say that α is *valid*, written $\models \alpha$, if for every e and w, $e, w \models \alpha$. Finally, a set of sentences Γ *logically entails* a sentence α, written $\Gamma \models \alpha$, if $\Gamma \cup \{\neg\alpha\}$ is unsatisfiable.

The second clause for sentences above formalizes equality as a built-in predicate of $\mathcal{KL}$ with a fixed meaning, namely identity over co-referring standard names. Furthermore, the fifth clause defines quantification to be interpreted substitutionally: A quantified formula like $\forall x\mathbf{K}P(x)$ is true in a model e, w just in case all of $\mathbf{K}P(\#1)$, $\mathbf{K}P(\#2)$,...are true in it. The truth of a quantified sentence is hence reduced to the truth of sentences of smaller size, and there is no need for using variable maps or the like. As we will see, this principle allows for proving properties about sentences by a simple induction over their structure, and is incorporated into our tableau system in a straightforward manner. The sixth clause formalizes the idea that knowing a formula α amounts to α being true in all worlds the agent considers possible. Note that it is not required that the real world w is contained in e, i.e., the agent's belief may be incorrect. Also, e can be empty, corresponding to the case where the agent's knowledge is inconsistent.

2.3. Axiom System for $\mathcal{KL}$

Levesque and Lakemeyer's axiom system for $\mathcal{KL}$ is given in Figure 1. The six axioms listed under the first item are the same as for $\mathcal{L}$, the non-modal fragment of $\mathcal{KL}$, with an additional proviso for the specialization axiom (Ax1.5) that instantiating the universally quantified variable x by a ground term t does not introduce any function symbols in the scope of a $\mathbf{K}$ modality. To see why the latter is needed, consider the formula $\forall x.\, \mathbf{K}P(x) \vee \mathbf{K}\neg P(x)$, expressing that for every individual, it is known whether it is a P or not. This does *not* logically entail that $\mathbf{K}P(a) \vee \mathbf{K}\neg P(a)$, because the denotation of the constant a may differ from world to world. Intuitively, this corresponds to a situation where the agent has complete knowledge about P for all individuals, but where the *identity* of the individual denoted

Axioms:

(Ax1) Axioms of $\mathcal{L}$, with proviso on (Ax1.5):

- **(Ax1.1)** $\alpha \supset (\beta \supset \alpha)$
- **(Ax1.2)** $(\alpha \supset (\beta \supset \gamma)) \supset ((\alpha \supset \beta) \supset (\alpha \supset \gamma))$
- **(Ax1.3)** $(\neg\beta \supset \neg\alpha) \supset ((\neg\beta \supset \alpha) \supset \beta)$
- **(Ax1.4)** $\forall x(\alpha \supset \beta) \supset (\alpha \supset \forall x\beta)$, provided that x does not occur freely in α
- **(Ax1.5)** $\forall x\alpha \supset \alpha^x_t$, provided no function symbol is introduced in the scope of a $\mathbf{K}$
- **(Ax1.6)** $(n = n) \wedge (n \neq m)$ for any distinct n, m

(Ax2) Knowledge of axioms:
$\mathbf{K}\alpha$, where α is an instance of an axiom from (Ax1);

(Ax3) Knowledge closed under Modus Ponens:
$\mathbf{K}(\alpha \supset \beta) \supset (\mathbf{K}\alpha \supset \mathbf{K}\beta)$;

(Ax4) Knowledge closed under universal generalization:
$\forall x\mathbf{K}\alpha \supset \mathbf{K}\forall x\alpha$;

(Ax5) Complete knowledge of subjective truths:
$(\sigma \supset \mathbf{K}\sigma)$, where σ is subjective.

Rules:

(MP) From α and $\alpha \supset \beta$, infer β.

(UG) From $\alpha^x_{n_1}, \ldots, \alpha^x_{n_k}$, infer $\forall x\alpha$, provided the n_i range over all names in α and at least one not in α.

Figure 1: Axiom System for $\mathcal{KL}$ [2].

by a is unknown, and hence it is unknown whether P applies to a or not. Axiom (Ax1.6) formalizes equality as identity over standard names.

Moreover, axioms (Ax2)–(Ax5) formalize the notion that the $\mathbf{K}$ modality represents a fully introspective, logically omniscient agent. That is to say the agent knows what it knows, knows what it does not know, and is aware of all the logical consequences of its knowledge. Here, the term "knowledge" is to be understood in the wider sense and interchangeable with "belief", i.e., what is known/believed may or may not be true in the real, outside world. Consequently, $\mathbf{K}$ corresponds to a K45 modality in terms of standard modal logic terminology.

In addition to the standard Modus Ponens rule, Levesque and Lakemeyer's system uses a Universal Generalization inference rule that only requires finitely many instances. The idea here is that if we can prove α^x_n for every standard name n, it can obviously be inferred that $\forall x\alpha$. Since standard names do not have any inherent structure other than being distinct from one another, it is however enough to have such a proof for every name occurring in α, plus a single proof for a name n that does *not* occur. The latter can then serve as proof for any other unmentioned name n', as all we would need to do is substitute n' for n. The same idea will be used by our tableau system to handle existential quantifiers.

Figures 2–4 show example proofs in the axiom system. The formula proven in Figure 2 says that as long as the agent's knowledge is consistent, it cannot know both a formula and its negation. In the figure, a line labelled "$\mathcal{L}$" means the corresponding formula can be derived purely by means of using the six axioms for $\mathcal{L}$, together with the derivation rules. If "KG" is given as justification, it means that the formula is obtained by the admissible rule "from α infer $\mathbf{K}\alpha$" [2, Theorem 4.4.4]. "MP" refers to the Modus Ponens rule. Next, the formula derived in Figure 3 expresses the fact that the agent knows all valid equalities over standard names. Here, "Ax" denotes an instance of an axiom, while "UG" is the Universal Generalization rule. The proofs in Figures 2 and 3 were originally given in [2] and are reproduced here for illustration purposes.

Now, axiom (Ax4) is usually called the Barcan formula and the converse implication is called the

1. $\beta \supset (\neg\beta \supset \alpha)$	$\mathcal{L}$
2. $\mathbf{K}(\beta \supset (\neg\beta \supset \alpha))$	KG
3. $\mathbf{K}\beta \supset (\mathbf{K}\neg\beta \supset \mathbf{K}\alpha)$	MP
4. $\neg\mathbf{K}\alpha \supset (\mathbf{K}\beta \supset \neg\mathbf{K}\neg\beta)$	$\mathcal{L}$

Figure 2: Axiomatic proof of not knowing a formula and its negation [2].

1. $(\#1 = \#1) \supset \mathbf{K}(\#1 = \#1)$	Ax
2. $(\#1 = \#2) \supset \mathbf{K}(\#1 = \#2)$	Ax
3. $\forall y.\, (\#1 = y) \supset \mathbf{K}(\#1 = y)$	UG
4. $\forall x \forall y.\, (x = y) \supset \mathbf{K}(x = y)$	UG

Figure 3: Axiomatic proof of knowing equality of names [2].

1. $\mathbf{K}(\forall x\alpha \supset \alpha^x_n)$	(Ax2.5)
2. $(\mathbf{K}\forall x\alpha) \supset (\mathbf{K}\alpha^x_n)$	MP from 1, (Ax3)
3. $((\mathbf{K}\forall x\alpha) \supset (\mathbf{K}\alpha))^x_n$	rewriting of 2
4. $\forall x.\, (\mathbf{K}\forall x\alpha) \supset (\mathbf{K}\alpha)$	UG from 3
5. $(\mathbf{K}\forall x\alpha) \supset (\forall x\mathbf{K}\alpha)$	MP from 4, (Ax1.4)

Figure 4: Axiomatic proof of the Converse Barcan formula. n ranges over all standard names occurring in α, plus one name that does not occur in α. Step 3 is justified by the fact that x does not occur freely in $\forall x\alpha$.

Converse Barcan formula, see the account in Section 8.10 of the book [16]. The Barcan and Converse Barcan formulas express interaction between quantifiers and modal operators, that is, quantifiers and modal operators can be permuted in one way or the other. Semantically, the Barcan formula says that the domain decreases (or stays the same), whereas the Converse Barcan formula says that the domain increases (or stays the same). In $\mathcal{KL}$, the domain is the same in any world, namely the set of standard names, so both the Barcan and Converse Barcan formulas are valid in it, and in fact, the Barcan formula is taken as an axiom. As shown in Figure 4, the Converse Barcan formula is provable, even without using the Barcan formula.[1] The proof is quite simple, involving axioms (Ax2.5), (Ax3), and (Ax1.4) but not (Ax4); note however that the part of the proof constituted by lines 1-3 is repeated once for each standard name occurring in α, plus one new standard name. This has to do with the fact that both the Barcan and Converse Barcan formulas are *axiom schemes*, with an instance for each possible substitution of α. The proof in Figure 4, and in particular its size, depends on the instance in question, so strictly speaking, we have not shown that the Converse Barcan scheme is *derivable*,[2] but we have shown that any instance of the scheme is provable. Compare to the tableau proof of the Converse Barcan formula in Figure 8 where it is the number of branches that depends on the instance in question.

Theorem 2.5 ([2]). *The axiom system for $\mathcal{KL}$ is sound and complete.*

We remark that, as a first-order logic, the set of valid sentences of $\mathcal{KL}$ is semi-decidable, but not decidable. Hence, in a manner of speaking, a sound and complete proof system is the best we can hope for (yielding a semi-decision procedure for validity), unless we consider restricted, decidable fragments. The latter is beyond the scope of the current paper, and left for future work.

[1] A similar observation also applies to some axiom systems for standard first-order modal logic, see Sections 2.5 and 2.9 of the handbook chapter [17].

[2] In fact, this cannot be hoped for since the number of premises of the rule (UG) also depends on the formula instance in question.

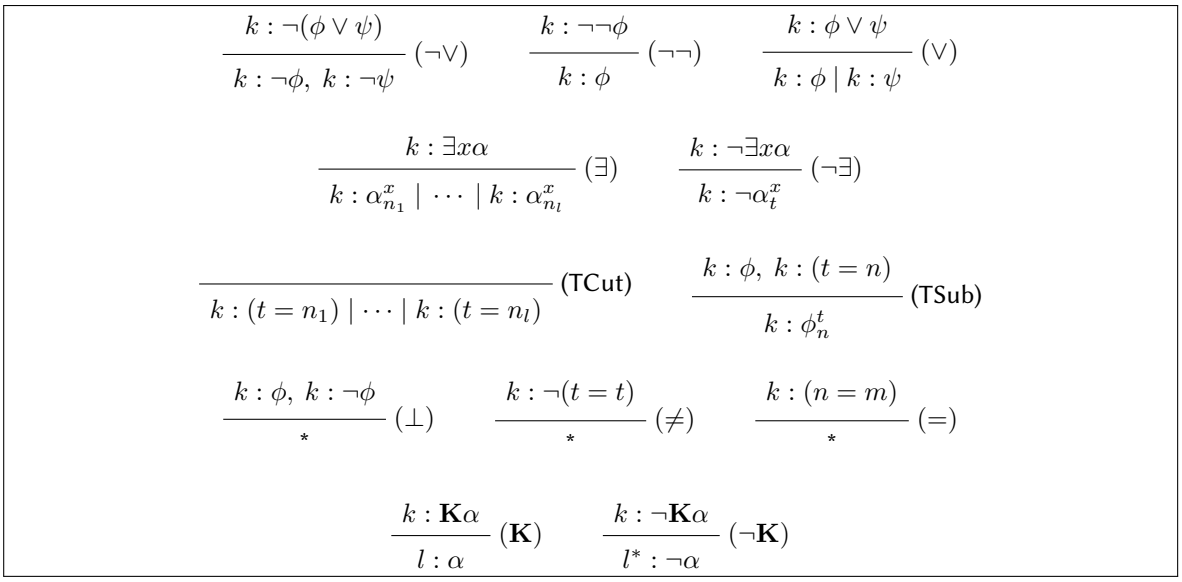

Figure 5: Tableau rules for $\mathcal{KL}$. In rules $(\exists)$ and (TCut), $n_1, \ldots, n_l$ range over all standard names in the branch, plus *one extra*. In rules $(\neg\exists)$ and $(\neq)$, t can be *any* ground term. In rule (TCut), t can be any ground term occurring on the branch. Rule $(\neg\exists)$ is only applicable if no function symbol is introduced in the scope of a $\mathbf{K}$. In rule (TSub), ϕ is a formula that mentions ground term t, and ϕ^t_n then means ϕ with every occurrence of t simultaneously replaced by n. Rule (TSub) is only applicable if t is a standard name, or t does not occur in the scope of a $\mathbf{K}$ in ϕ. In rule $(=)$, n and m are *distinct* names. In rule $(\mathbf{K})$, l is an index on the branch other than 0. In rule $(\neg\mathbf{K})$, l^* is an index that is new on the branch.

3. Sentence Tableaux for $\mathcal{KL}$

Drawing inspiration from the axiom system for $\mathcal{KL}$, and based on our earlier system [15] for the non-modal fragment $\mathcal{L}$, we devised a tableau proof system for $\mathcal{KL}$ as described below and depicted in Figure 5.

Definition 3.1 (Tableau System for $\mathcal{KL}$)**.** Every formula will be prefixed by a natural number k, intuitively denoting the index of a possible world under consideration. The index 0 is used to denote the "real" world, and any tableau for a given formula ϕ will start with a single node labelled $0 : \phi$. The rules for the system are depicted in Figure 5.

The first row in Figure 5 are the standard rules for Boolean connectives. The second row contains rules for quantified formulas, where the one for existential quantification incorporates the same idea used in the Universal Generalization inference rule, namely that a single additional standard name is enough to serve as representative for all unmentioned names. For universal quantification, we have the same proviso as for Axiom (Ax1.5), for the same reason as stated in the previous section.

The third row lists rules for handling terms that include functions, one for making a case distinction over their possible values (again with the "trick" of using an extra name), and one for substituting a term for its value. The substitution rule requires a variant of the aforementioned proviso again, in this case to prevent a term mentioning a function inside the scope of a $\mathbf{K}$ being substituted by a value the term has outside of $\mathbf{K}$. The reason for this is similar as before: Having a formula such as $k : \mathbf{K}P(a)$ means that it is believed that the individual denoted by a is a P, and having $k : (a = n)$ means that a is identical to n at world k. This does however *not* entail that a is identical to n in all worlds the agent considers possible; on the contrary, different possible worlds might assign different values (names) to a, meaning that a's identity is unknown.

The fourth row lists the standard rule for closing a branch due to complementary formulas, a standard rule for closing a branch with an inequality over the same term, and a non-standard rule for closing a branch based on equality over distinct standard names.

The first four rows together correspond to prefixed versions of the rules for $\mathcal{L}$ as described in [15]. The fifth row in Figure 5 contains new rules dealing with the $\mathbf{K}$ modality. The $(\mathbf{K})$ rule is based on the intuition that $\mathbf{K}$ represents a fully introspective agent, so believing α means α must hold in any accessible world, but not necessarily the real world 0. Similarly, *not* believing α means that there has to be one accessible world where α is false. As this world may or may not be one of the worlds already mentioned on the branch, the $(\neg\mathbf{K})$ rule hence introduces a new index. Note that this excludes world 0, since 0 is present on every branch due to being used in the root node.

Our propositional tableau rules are similar to rules given in [13],[3] with the difference that $(\mathbf{K})$ and $(\neg\mathbf{K})$ are not restricted to world 0, but are applicable to arbitrary worlds k. This restriction is valid in [13] due to the fact that every *propositional* $\mathcal{KL}$ formula can be reduced to an equivalent one without nested modalities [2, Theorem 4.6.1]. In general, this is however not the case for the full first-order language [2, Theorem 4.6.2]. Note that since we do not (yet) consider the $\mathbf{N}$ and $\mathbf{O}$ modalities of $\mathcal{OL}$, we do not require an extra metalinguistic component to distinguish accessible from inaccessible worlds, as done in [13]. Also, note that we do not use prefix *sequences* as in [18], the reason being that the accessibility relation in a K45 logic is such that any world in the epistemic state e is accessible from any other world in e.

Let us look at a few example proofs in our system. Figure 6 shows the tableau proof for the formula proven axiomatically in Figure 2. While both proofs are relatively straightforward, the tableau proof is perhaps better for illustrating why the $\neg\mathbf{K}\alpha$ part is needed, namely for ensuring that at least one world (here with index 1) is included in the epistemic state. Otherwise, note that $\mathbf{K}\beta \wedge \mathbf{K}\neg\beta$ is satisfiable, namely in cases where $e = \emptyset$. Accordingly, in our system it is not possible to close a branch solely from formulas $0 : \mathbf{K}\beta$ and $0 : \mathbf{K}\neg\beta$.

Figure 7 shows the tableau proof for the formula proven axiomatically in Figure 3. Observe the similarity in how the (UG) and $(\exists)$ rules, respectively, are applied twice. In both cases, one application is to a formula/branch without standard names, and hence only requires one instance by means of standard name $\#1$. The second application then involves two standard names, namely $\#1$ together with $\#2$ as new name.

Figure 8 shows a tableau proof for the Converse Barcan formula. Note that the proviso for $(\neg\exists)$ is satisfied for each instance since the variable x is only substituted by standard names. Due to the analytic nature of the tableau system, the proof is arguably more straightforward compared to the corresponding axiomatic one given in Figure 4.

Finally, Figure 9 presents an example proof with formulas that involve nested functions and quantifying into modal operators. Notice how after branching over the possible values for the primitive term $f(\#1)$ using the (TCut) rule in line 8, the left branch uses two applications of (TSub) do decompose $f(f(\#1))$ into smaller parts. The right branch illustrates that we do not need additional rules for handling equality, such as a rule for inferring $(t = s)$ from $(s = t)$, but it is rather enough to decompose terms into their parts, substitute, and close the branch based on equalities over the involved standard names. As our proofs will show, this approach works in general for arbitrarily nested terms in all parts of a formula, including equalities, so indeed our system is sound and complete for the full language.

3.1. Soundness

To prove soundness, we first define what it means to satisfy a set of prefixed formulas. As usual, we identify a branch in a tableau with the set of prefixed formulas it contains.

Definition 3.2 (Satisfying a Branch). [4] Let B be a set of prefixed formulas. An epistemic state e and a world w *satisfy* B, written $e, w \models B$, iff for every index i occurring in B there is a world w_i such that $w_0 = w$, $w_j \in e$ for all $j > 0$, and for all $k : \phi \in B$, $e, w_k \models \phi$.

[3] Note, however, that we do not need the $\mathcal{M}$-rule considered in that paper.

[4] Strictly speaking, we here define a function $w_{(-)}$ which for each index picks out a world satisfying a certain requirement, and this requires the well-known Axiom of Choice. The function $w_{(-)}$ is convenient in our soundness proof, but we remark that we could get along without this function, and hence the Axiom of Choice, by defining satisfaction of a branch differently, somewhat along the lines of Definition 4 in the paper [13], page 182.

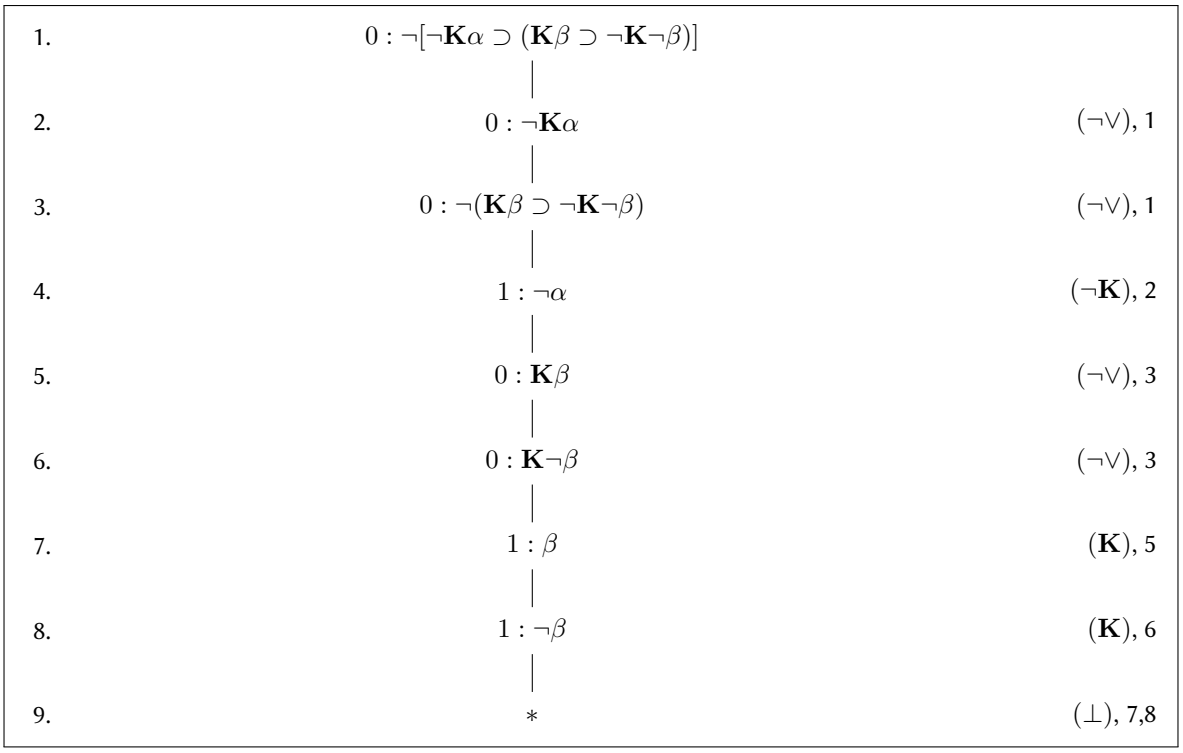

Figure 6: Tableau proof for not knowing a formula and its negation. Applications of the $(\neg\neg)$ rule are not shown.

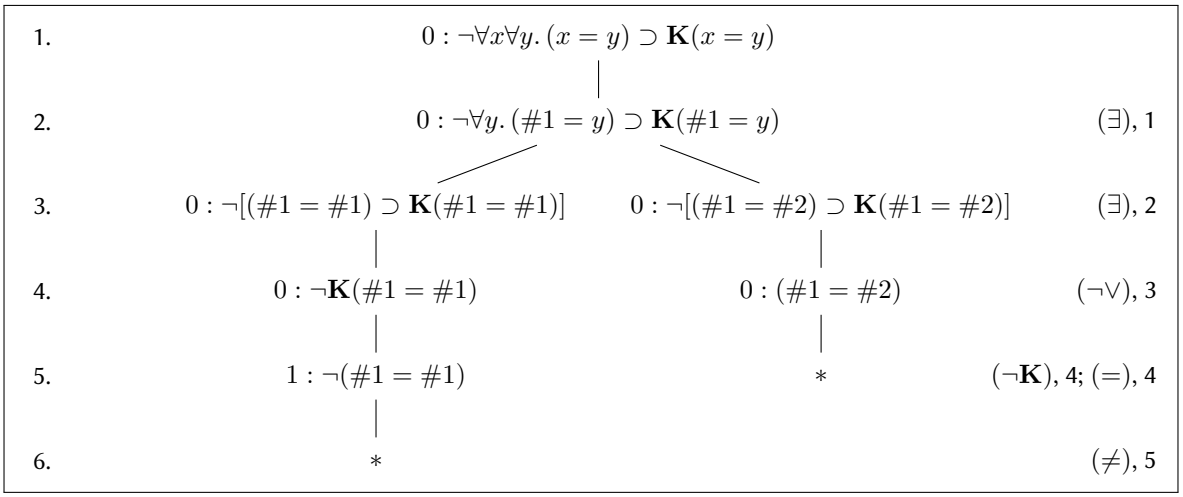

Figure 7: Tableau proof for knowing the equality of names. Applications of the $(\neg\neg)$ rule are not shown and a couple of insignificant formulas are omitted.

Theorem 3.3 (Soundness)**.** *The system given in Definition 3.1 is sound, i.e., if a $\mathcal{KL}$ sentence ϕ is satisfiable, then every tableau with root $0 : \phi$ has an open branch.*

Proof. We prove by induction that the expansion rules preserve the property that when being applied to a satisfiable branch, at least one of the resulting branches is satisfiable as well. Let B be a satisfiable branch where $e, w \models B$. We consider each rule case-by-case. Clearly, none of the closure rules are applicable in a satisfiable branch.

For the $(\neg\vee)$ rule, let $k : (\phi \vee \psi) \in B$. Then $e, w_k \models \phi$ or $e, w_k \models \psi$. In the former case, $e, w \models B \cup \{k : \phi\}$, in the latter case, $e, w \models B \cup \{k : \psi\}$. In either case, the claim follows. The rules $(\neg\neg)$ and $(\vee)$ work similarly.

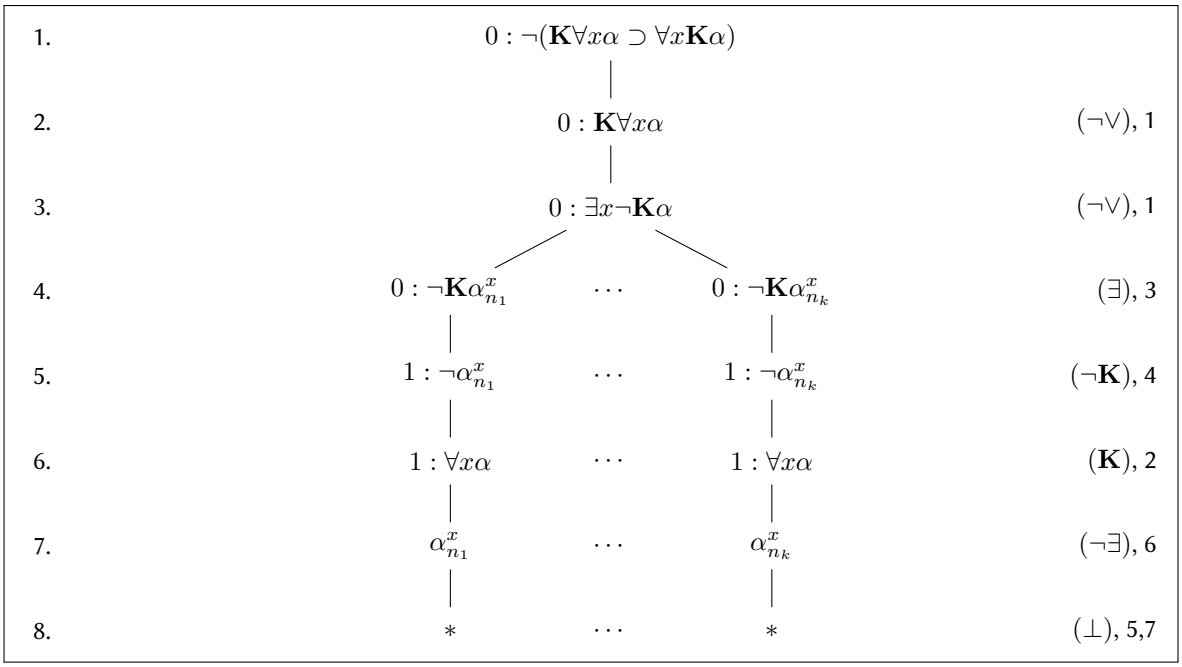

Figure 8: Tableau proof for the converse Barcan formula. Applications of the $(\neg\neg)$ rule are not shown. $n_1, \ldots, n_k$ range over all standard names in α, plus one name not occurring in α.

For the $(\exists)$ rule, let $k : \exists x\alpha \in B$. Then $e, w_k \models \alpha^x_n$ for some name n. If n is one of the names $n_1, \ldots, n_k$, then by assumption $e, w \models B \cup \{k : \alpha^x_n\}$, and we are done. In case where n is not one of the names $n_1, \ldots, n_k$, then n is not mentioned in B and n is distinct from the extra name (let's call it n'), and so let $*$ be a bijection from standard names to standard names that swaps n with n' and leaves all other names unchanged. Similar as in the proof for Theorem 4.4.2 in [2] (with β instead of α to avoid name confusion), let w^* be the world such that $w^*[\beta] = w[\beta^*]$ for every primitive formula β, and $w^*[t] = w[t^*]$ for every primitive term t, where β^* and t^* denote the results of simultaneously replacing every name by its mapping under $*$ in β and t, respectively. Let $e^* = \{w^* \mid w \in e\}$. It follows by induction that for any formula ϕ and any index j, it is the case that $e^*, w^*_j \models \phi$ iff $e, w_j \models \phi^*$. Given $e, w_k \models \alpha^x_n$, and $\alpha^x_n = (\alpha^x_{n'})^*$, we then get $e^*, w^*_k \models \alpha^x_{n'}$. Since B does not mention n', $B^* = B$. Hence $e^*, w^* \models B \cup \{k : \alpha^x_{n'}\}$, so the claim follows. The proof for the (TCut) rule works similarly.

For the $(\neg\exists)$ rule, let $k : \neg\exists x\alpha \in B$. Then $e, w_k \models \forall x\neg\alpha$. Let t be a term such that replacing x by t in α does not introduce a function symbol in the scope of $\mathbf{K}$. By Theorem 4.3.6 in [2], we get that $e, w_k \models \neg\alpha^x_t$, and so $e, w \models B \cup \{k : \neg\alpha^x_t\}$.

For the (TSub) rule, let $k : \phi \in B$ and $k : (t = n) \in B$, where t is a standard name, or t does not occur within the scope of $\mathbf{K}$ in ϕ. Then $e, w_k \models \phi$ and $w_k(t) = n$. It follows by induction on the structure of ϕ that $e, w_k \models \phi^t_n$, and so $e, w \models B \cup \{k : \phi^t_n\}$.

For the $(\mathbf{K})$ rule, let $k : \mathbf{K}\alpha \in B$. Then $e, w_k \models \mathbf{K}\alpha$, i.e., $e \models \mathbf{K}\alpha$. Since l is an index occurring in B, $w_l \in e$, hence $e, w_l \models \alpha$, and so $e, w \models B \cup \{l : \alpha\}$.

For the $(\neg\mathbf{K})$ rule, let $k : \neg\mathbf{K}\alpha \in B$. Then $e, w_k \models \neg\mathbf{K}\alpha$, i.e., $e \models \neg\mathbf{K}\alpha$. Hence there is $w^* \in e$ such that $e, w^* \models \neg\alpha$. If l^* is an index not occurring in B, we get $e, w \models B \cup \{l^* : \neg\alpha\}$. □

3.2. Completeness

Our completeness result is based on a variant of Hintikka sets and the construction of a systematic tableau. In the following, by a *literal* we mean a sentence that is a predicate atom, a negated predicate atom, an equality, or a negated equality. A literal is called *primitive* if it does not mention any function symbols, and *non-primitive* if it is not primitive.

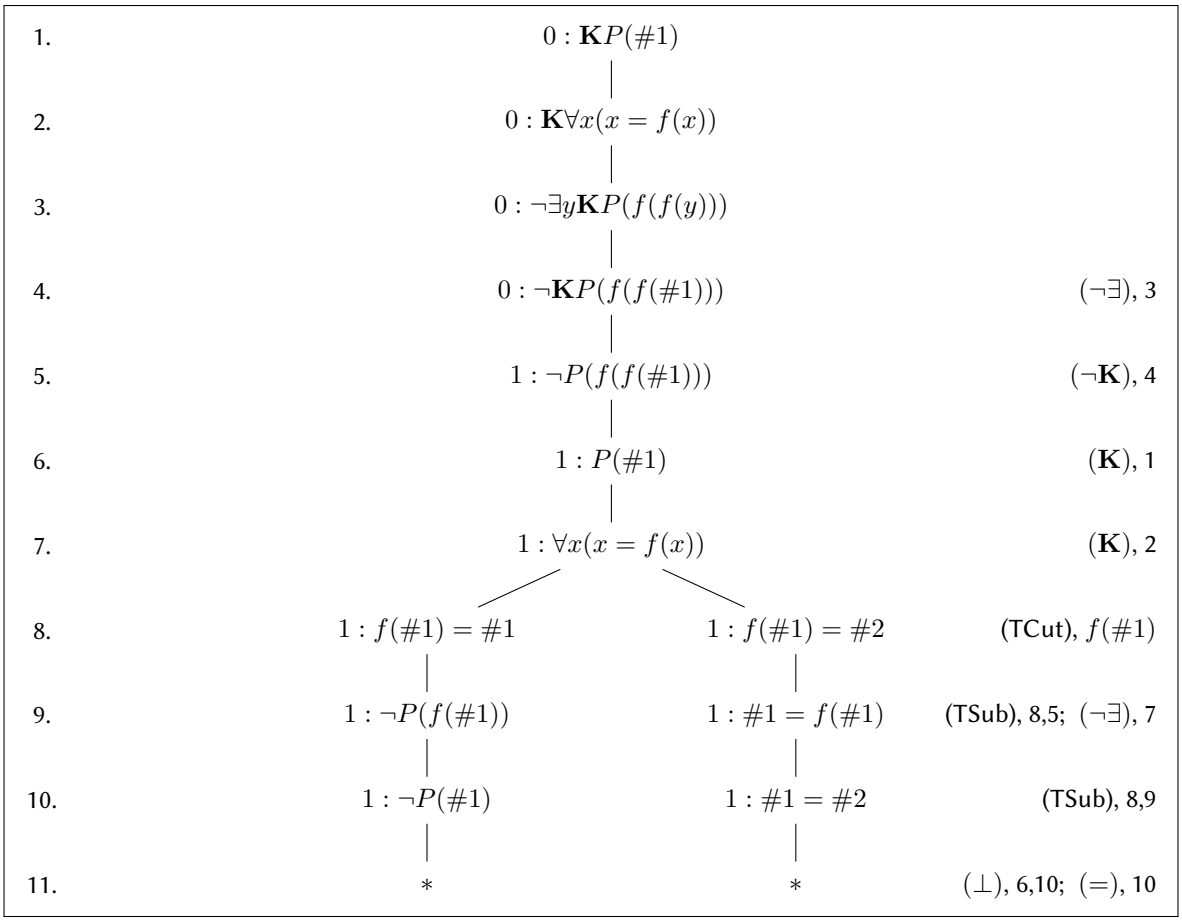

Figure 9: Tableau for $\{\mathbf{K}P(\#1), \mathbf{K}\forall x(x = f(x)), \neg\exists y\mathbf{K}P(f(f(y)))\}$.

Definition 3.4 (Hintikka Set)**.** A set S of prefixed $\mathcal{KL}$ sentences is called a *Hintikka set* if it satisfies the following conditions:

1. S does not contain both a primitive atom $k : P(n_1, \ldots, n_l)$ and its negation $k : \neg P(n_1, \ldots, n_l)$.
2. S does not contain an inequality $k : \neg(n = n)$ for any standard names n.
3. S does not contain an equality $k : (n = m)$ for any distinct names n, m.
4. If S contains $k : \neg(\phi \vee \psi)$, then it contains $k : \neg\phi$ and $k : \neg\psi$.
5. If S contains $k : \neg\neg\phi$, then it contains $k : \phi$.
6. If S contains $k : \phi \vee \psi$, then it contains $k : \phi$ or it contains $k : \psi$.
7. If S contains $k : \exists x\phi$, then it contains $k : \phi^x_n$ for some standard name n.
8. If S contains $k : \neg\exists x\phi$, then it contains $k : \neg\phi^x_n$ for all standard names n.
9. If S contains a literal $k : \phi$ that mentions a primitive term t, then S contains $k : (t = n)$ for some name n.
10. If S contains a literal $k : \phi$ that mentions a primitive term t and S contains $k : (t = n)$ for some name n, then S contains ϕ^t_n.
11. If S contains $k : \mathbf{K}\alpha$, then it contains $l : \alpha$ for all $l > 0$ appearing in S.
12. If S contains $k : \neg\mathbf{K}\alpha$, then it contains $l : \neg\alpha$ for some $l > 0$.

We remark that the definition above corresponds to what is called an *atomic* Hintikka set by Letz [19] and others, since only inconsistencies among or within literals are considered. Obviously, a set S that contains both ϕ and $\neg\phi$ for a non-literal formula is also unsatisfiable, which is why the $(\perp)$ rule allows to close corresponding branches. However, as we will see, for proving completeness, it suffices to restrict our attention to literals and (in)equalities when closing branches in a tableau. For the sake

of brevity, we will still just write "Hintikka set", "closed branch" etc. instead of "atomic Hintikka set", "atomically closed branch" etc.

Note that the restriction to primitive terms is sufficient here, as non-primitive ground terms are handled by unnesting them in a recursive fashion. For example, if a set S contains $k : P(f(a))$ and $k : f(a) = n$, where a is a constant (and hence a primitive term), then by clause 9, S has to include $k : (a = m)$ for some name m, and by clause 10 also $k : P(f(m))$ and $k : f(m) = n$. Thus $f(m)$ is another primitive term occurring in a literal in S with prefix k, and so S has to include $k : P(n)$. Similarly, arbitrarily nested terms in (in)equalities get subsequently decomposed and substituted by standard names, reducing them to (in)equalities over standard names only.

Lemma 3.5 (Hintikka's Lemma)**.** *Every Hintikka set is satisfiable.*

Proof. Let S be a Hintikka set. We construct a model e, w as follows. For any index $k \geq 0$ appearing in S, let w_k be a world such that

- $w_k[P(n_1, \ldots, n_l)] = 1$ if $k : P(n_1, \ldots, n_l) \in S$;
- $w_k[P(n_1, \ldots, n_l)] = 0$ if $k : \neg P(n_1, \ldots, n_l) \in S$;
- $w_k[f(n_1, \ldots, n_l)] = n$ if $k : (f(n_1, \ldots, n_l) = n) \in S$.

Observe that w_k is well defined: By the first clause of Definition 3.4, there can be no primitive atom $P(n_1, \ldots, n_l)$ and index k such that both $k : P(n_1, \ldots, n_l) \in S$ and $k : \neg P(n_1, \ldots, n_l) \in S$. Moreover, there cannot be any primitive term $f(n_1, \ldots, n_l)$ and index k such that $k : (f(n_1, \ldots, n_l) = n) \in S$ and $k : (f(n_1, \ldots, n_l) = m) \in S$ for two distinct standard names n and m. If this were the case, then by the tenth clause of Definition 3.4, $k : (n = m) \in S$, thus violating the third clause. Let e, w be so that $w = w_0$ and $e = \{w_l \mid l > 0 \text{ occurs in } S\}$. We prove by induction on the structure of prefixed formulas $k : \phi$ in S that $e, w_k \models \phi$.

- For the base case of ϕ being a primitive atom $P(n_1, \ldots, n_l)$, $e, w_k \models \phi$ immediately by assumption. Similar for a negated primitive atom $\neg\phi$.
- For the base case of ϕ being an equality $(n = m)$ over standard names, by assumption n and m are not distinct, and so clearly $e, w_k \models (n = m)$. Similarly, if ϕ is an inequality $\neg(n = m)$ over standard names, then by assumption, n and m must be distinct, and hence $e, w_k \models \neg(n = m)$.
- If ϕ is an atom $P(t_1, \ldots, t_l)$ that is *not* primitive, then it must contain some primitive term t, and by the ninth clause of Definition 3.4, $k : (t = n) \in S$ for some name n, and so $k : \phi^t_n \in S$ by the tenth clause. Note that ϕ^t_n is structurally simpler than ϕ, since it has one level of nesting removed for at least one term. It follows by induction that $e, w_k \models \phi^t_n$ and $e, w_k \models (t = n)$, and hence $e, w_k \models \phi$. Similar for negated atoms.
- If ϕ is an equality $(t_1 = t_2)$ or inequality $\neg(t_1 = t_2)$ where at least one of t_1, t_2 is not a standard name, the argument is exactly as in the previous item.
- The cases for the Boolean connectives follow immediately by induction.
- If ϕ is of the form $\exists x\alpha$, then by the seventh clause of Definition 3.4, $k : \alpha^x_n \in S$ for some name n. By induction, $e, w_k \models \alpha^x_n$, and hence by the semantics, $e, w_k \models \exists x\alpha$.
- If ϕ is of the form $\neg\exists x\alpha$, then by the eighth clause of Definition 3.4, $k : \neg\alpha^x_n \in S$ for all names n. By induction, $e, w_k \models \neg\alpha^x_n$ for all such that n, and hence by the semantics, $e, w_k \models \neg\exists x\alpha$.
- If ϕ is of the form $\mathbf{K}\alpha$, then by the eleventh clause of Definition 3.4, $l : \alpha \in S$ for all $l > 0$ appearing in S. By induction, $e, w_l \models \alpha$ for all such indexes l, i.e., for every world $w_l \in e$. But this means that $e, w_k \models \mathbf{K}\alpha$.
- If ϕ is of the form $\neg\mathbf{K}\alpha$, then by the twelfth clause of Definition 3.4, $l : \neg\alpha \in S$ for some $l > 0$. By induction, $e, w_l \models \neg\alpha$, i.e., $e, w_l \not\models \alpha$ for some $w_l \in e$. In terms of the semantics, this means that $e, w_k \models \neg\mathbf{K}\alpha$.

Hence $e, w \models S$, and so S is satisfiable. □

In what follows, we need total orders over the set of standard names and the set of primitive terms. For standard names, we can use the one where $\#1$ is the first name, $\#2$ the second, and so on. For the primitive terms, we can use any term order.

The construction of the systematic tableau shown below is inspired by the one presented in [16], in particular due to the fact that it works in stages. Intuitively, a stage corresponds to expanding all nodes in a current tableau (if possible), before proceeding to the next stage. This corresponds to a "breadth-first" expansion that might not be optimal in practice, but suffices for showing completeness.

We will use notation borrowed from [19], where $B \oplus (k : \phi)$ means extending B and adding the indexed formula $(k : \phi)$, and $B \oplus [(k : \phi) \mid (l : \psi)]$ means splitting B into two branches, where $(k : \phi)$ is added to one of the obtained branches and $(l : \psi)$ is added to the other.

Definition 3.6 (Systematic Tableau Construction)**.** We assume that every node in a tableau T is optionally marked as "completed". The *systematic tableau sequence* $T_0, T_1, T_2, \ldots$ for a sentence ϕ is given by:

- The first tableau T_0 is a one-node tableau with root formula $(0 : \phi)$, not marked as completed.
- Given tableau T_i at stage i, if there are any open branches, the next tableau T_{i+1} at stage $i + 1$ is obtained as follows. Considering open branches from left to right, each branch B is processed from top to bottom by considering every node on it. Processing each such node may add new indexed formulas at the end, or even split the branch, but the formulas added in this way will only be processed in the next stage, if any. For every node N at B, if its formula is not marked as completed, we proceed by expanding B as stated below:
  - $B \oplus (k : \neg\phi) \oplus (k : \neg\psi)$, if the formula at N is $k : \neg(\phi \vee \psi)$;
  - $B \oplus (k : \phi)$, if the formula at N is $k : \neg\neg\phi$;
  - $B \oplus [(k : \phi) \mid (k : \psi)]$, if the formula at N is $k : (\phi \vee \psi)$;
  - $B \oplus [(k : \alpha^x_{n_1}) \mid \cdots \mid (k : \alpha^x_{n_l})]$, if the formula at N is $k : \exists x\alpha$, $n_1, \ldots, n_{l-1}$ is the ordered sequence of all standard names occurring in B, and n_l is the least (wrt the name order) standard name not occurring in B;
  - $B \oplus (k : \neg\alpha^x_n)$, if the formula at N is $k : \neg\exists x\alpha$ and n is the least standard name such that $(k : \neg\alpha^x_n) \notin B$;
  - $B \oplus [k : (t = n_1) \mid \cdots \mid k : (t = n_l)]$, if the formula $k : \phi$ at N is a literal that mentions at least one primitive term, t is the smallest (wrt the term order) primitive term such that no $(t = n)$ is in B for any n, and where $n_1, \ldots, n_{l-1}$ is the ordered sequence of all standard names occurring in B, and n_l is the least standard name not occurring in B;
  - $B \oplus (k : \phi^{t_1}_{n_1}) \oplus \cdots \oplus (k : \phi^{t_j}_{n_j})$, if the formula $k : \phi$ at N is a non-primitive literal and $(t_1, n_1), \ldots, (t_j, n_j)$ is the lexicographically ordered sequence of all pairs of primitive terms t and names n such that t occurs in ϕ, $k : (t = n)$ is on B, and $k : \phi^t_n$ is not on B;
  - $B \oplus (l_1 : \alpha) \oplus \cdots \oplus (l_j : \alpha)$, if the formula at N is $k : \mathbf{K}\alpha$ and $l_1, \ldots, l_j$ is the ordered sequence of all indexes l occurring in B such that $l > 0$ and $(l : \alpha) \notin B$;
  - $B \oplus (l : \neg\alpha)$, if the formula at N is $k : \neg\mathbf{K}\alpha$, and $l > 0$ is the smallest index not occurring in B.

  In addition, every application of a rule marks the node's formula as completed, *except* in the case of literals and $\neg\exists x\alpha$ and $\mathbf{K}\alpha$ formulas. Afterwards, close every branch that can be closed.
- If T_i has no open branches, it is the last tableau in the sequence.

Definition 3.7 (Saturated Systematic Tableau)**.** Given a systematic tableau sequence $T_0, T_1, \ldots$ for formula ϕ, the *saturated systematic tableau* of ϕ, denoted T^*, is the smallest tree containing all T_i as initial segments. Note that T^* is guaranteed to exist because our tableau system is *non-destructive*, i.e., if we obtain T' by applying a rule to T, T is an initial segment of T'. Furthermore, T^* is infinite just in case the sequence is.

Proposition 3.8. *If B is an open branch of a saturated systematic tableau T^*, then the set of formulas it contains is a Hintikka set.*

Proof. Let B be an open branch in T^*. Obviously, clauses 1–3 in Definition 3.4 apply, as otherwise B would have been closed at some stage.

Hence assume $k : \varphi$ is an indexed formula on B that is not a primitive literal. Let T_i be the tableau of the stage where $k : \varphi$ was introduced, and B_i the subbranch of B at that stage. If one of the clauses 4–7 or 12 applies, the appropriate formulas are added to the branch in the next stage.

If clause 8 in Definition 3.4 applies, then φ is of the form $\neg\exists x\phi$. It follows by induction on the natural numbers that for any standard name n, if n is the j-th standard name wrt the name order, $k : \neg\phi^x_n$ is added to the branch no later than stage $i + j$ (note that a formula is only added if it does not occur already). Hence B contains $k : \neg\phi^x_n$ for every name n.

If clause 9 in Definition 3.4 applies, then φ is a literal mentioning at least one primitive term. Consequently, there must be some primitive term t in φ that is the smallest wrt the term order. If B_i already contains some $k : (t = n)$, we are done. Otherwise, the corresponding construction rule guarantees the addition of some $k : (t = n)$ in T_{i+1}.

If clause 10 in Definition 3.4 applies, then φ is a literal ϕ mentioning a primitive term t so that $k : (t = n) \in B$ for some name n. Let j be the stage where $k : \phi$ was introduced to B, or where $k : (t = n)$ was introduced to B, whichever happened later. Then by the construction above, $k : \phi^t_n$ is added to B at stage $j + 1$.

If clause 11 in Definition 3.4 applies, then φ is of the form $\mathbf{K}\alpha$. Let $l > 0$ be any index occurring in B. If $l : \alpha$ is not already present in B_i, let $j \geq i$ be the earliest stage containing l. By construction, $l : \alpha$ is added to B in stage $j + 1$. Hence B contains $l : \alpha$ for every index $l > 0$ occurring in B.

Therefore, the set of indexed formulas in B is a Hintikka set. □

Theorem 3.9 (Completeness)**.** *The system shown in Figure 5 is complete, i.e., if a sentence ϕ is unsatisfiable, then there is a finite tableau for ϕ all of whose branches are closed.*

Proof. Let T be a saturated systematic tableau for ϕ. Then T must be closed: Suppose it is not, then there is an open branch B, whose formulas by Proposition 3.8 form a Hintikka set. By Lemma 3.5, this set would be satisfiable. Since the set includes $0 : \phi$, this would mean that ϕ is satisfiable, contradicting the assumption that this formula is unsatisfiable. Furthermore, T being closed implies it being finite: When closing a branch, it will not be extended anymore, so its length remains finite. Because every node has only finitely many successors, the tree T is also finite by König's Lemma. □

4. Concluding Remarks

We have presented a tableau system for $\mathcal{KL}$, Levesque and Lakemeyer's first-order epistemic logic over standard names, and showed its soundness and completeness. Our results here generalize our previous work [15] on a tableau system for $\mathcal{L}$, which is the fragment obtained from $\mathcal{KL}$ when leaving out the modal operator $\mathbf{K}$. Interestingly, the completeness proof we give in this paper turns out to be simpler and less technical, one of the reasons being that the construction of the systematic tableau uses a "breadth-first" approach similar as in [16], rather than expanding a single node at a time as in [19], which requires some additional "bookkeeping".

Given how we handle modal subformulas in our system, it clearly falls into the category of single-step, prefixed tableaux [18, 16]. Ours is simpler in comparison since it does not require index *sequences* for prefixes, but simple numeric labels suffice. The reason for this lies in the fact that $\mathcal{KL}$ is not based on general Kripke structures with arbitrary accessibility relations, but uses sets of worlds to represent epistemic states. Since every world in an epistemic state is accessible from any other world in it, the index in front of subjective sentences like $\mathbf{K}\alpha$ and $\neg\mathbf{K}\alpha$ is actually irrelevant, and could hence even be left out entirely (for the same reason we allow to write $e \models \mathbf{K}\alpha$ instead of $e, w \models \mathbf{K}\alpha$). As mentioned before, $\mathcal{KL}$'s $\mathbf{K}$ operator behaves like the modality in the modal logic K45. The latter is part of a family of logics that also includes KB4, KD45, and S5, all of which allow for a simplified set-based Kripke-style semantics, and correspondingly simpler tableau systems as studied by Petrukhin and Zawidzki [14]. As

a matter of fact, when restricted to the propositional fragment of $\mathcal{KL}$, our system coincides with the one they provide for K45.

This leads to the question whether we can replace the K45 system by any of the KB4, KD45, and S5 systems, such that sound and complete first-order logics are obtained? This would be of interest not only from a theoretical, but also a more application-oriented point of view, since all four modal logics are used to model knowledge and belief in knowledge representation. Moreover, the paper [14, p. 130] remarks that "[in] the future we would like to investigate whether modal logics with simplified semantics give rise to a class of sequent or hypersequent calculi which is uniform in any respect." As we interpret this remark, the authors of that paper will investigate whether their strategy applied to K45, KB4, KD45 and S5 can be generalized uniformly to a broader range of logics, perhaps involving other formats of proof system. Of course, we would like to investigate whether such a broad range of logics can be equipped with first-order machinery as we have done in the present paper. Along a different line, adding hybrid-logical machinery is a well-known way to enable uniform proof systems, see [20], this would be interesting to investigate, and also, we would like to investigate how (or if) hybridization interacts with the strategy pursued in [14].

As a final remark, note that $\mathcal{KL}$ is not compact. Consequently, the tableau system only works for finite inputs, which is arguably not a restriction in terms of practical applications, but actually simplifies certain matters considerably. In particular, some texts [16] introduce the notions of local and global assumptions when defining logical consequence. Here, a local assumption is a formula taken to hold only at the possible world currently under consideration, while global assumptions hold everywhere. Since in our case these sets need to be finite, the local assumptions can be identified with a sentence λ, and the global assumptions with a sentence γ. Answering the question whether a sentence α is entailed by these assumptions then corresponds to checking whether $\lambda \wedge \mathbf{K}\gamma \models \alpha$, or equivalently whether there is a closed tableau for $\{\lambda, \mathbf{K}\gamma, \neg\alpha\}$.

References

[1] H. J. Levesque, Foundations of a functional approach to knowledge representation, Artificial Intelligence 23 (1984) 155–212. doi:10.1016/0004-3702(84)90009-2.

[2] H. J. Levesque, G. Lakemeyer, The Logic of Knowledge Bases, 2nd ed., College Publications, 2022.

[3] H. J. Levesque, All I know: a study in autoepistemic logic, Artificial Intelligence 42 (1990) 263–309. doi:10.1016/0004-3702(90)90056-6.

[4] G. Lakemeyer, H. J. Levesque, Only-knowing meets nonmonotonic modal logic, in: G. Brewka, T. Eiter, S. A. McIlraith (Eds.), Proceedings of the Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2012), AAAI Press, 2012, pp. 350–357.

[5] G. Lakemeyer, The situation calculus: A case for modal logic, Journal of Logic, Language and Information 19 (2010) 431–450. doi:10.1007/s10849-009-9117-6.

[6] G. Lakemeyer, H. J. Levesque, A semantic characterization of a useful fragment of the situation calculus with knowledge, Artificial Intelligence 175 (2010) 142–164.

[7] J. Y. Halpern, G. Lakemeyer, Multi-agent only knowing, Journal of Logic and Computation 11 (2001) 41–70. doi:10.1093/logcom/11.1.41.

[8] V. Belle, G. Lakemeyer, Multi-agent only-knowing revisited, in: F. Lin, U. Sattler, M. Truszczynski (Eds.), Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010), AAAI Press, 2010, pp. 49–59.

[9] G. Lakemeyer, H. J. Levesque, Decidable reasoning in a logic of limited belief with introspection and unknown individuals, in: F. Rossi (Ed.), Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 2013), AAAI Press, 2013, pp. 969–975.

[10] G. Lakemeyer, H. J. Levesque, A tractable, expressive, and eventually complete first-order logic of limited belief, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019), ijcai.org, 2019, pp. 1764–1771. doi:10.24963/ijcai.2019/244.

[11] D. Liu, Q. Feng, On the progression of belief, Artif. Intell. 322 (2023) 103947. doi:10.1016/J.ARTINT.2023.103947.
[12] D. Liu, G. Lakemeyer, A framework for belief-based programs and their verification, J. Artif. Intell. Res. 82 (2025) 1205–1242. doi:10.1613/JAIR.1.15796.
[13] R. Rosati, A sound and complete tableau calculus for reasoning about only knowing and knowing at most, Studia Logica 69 (2001) 171–191. doi:10.1023/A:1013894629540.
[14] Y. I. Petrukhin, M. Zawidzki, From simplified Kripke-style semantics to simplified analytic tableaux for some normal modal logics, in: M. Alviano, G. Greco, F. Scarcello (Eds.), Proceedings of the XVIIIth International Conference of the Italian Association for Artificial Intelligenc (AI*IA 2019), volume 11946 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 116–131. doi:10.1007/978-3-030-35166-3_9.
[15] J. Claßen, T. Braüner, A tableau system for first-order logic with standard names, in: G. L. Pozzato, T. Uustalu (Eds.), Proceedings of the 34th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2025), volume 15980 of *Lecture Notes in Computer Science*, Springer, 2025, pp. 22–38. doi:10.1007/978-3-032-06085-3_2.
[16] M. Fitting, R. L. Mendelsohn, First-Order Modal Logic, 2nd ed., Courier Dover Publications, 2023.
[17] T. Braüner, S. Ghilardi, First-order modal logic, in: P. Blackburn, J. van Benthem, F. Wolter (Eds.), Handbook of Modal Logic, Elsevier, 2007, pp. 549–620.
[18] F. Massacci, Single step tableaux for modal logics, Journal of Automated Reasoning 24 (2000) 319–364. doi:10.1023/A:1006155811656.
[19] R. Letz, First-order tableau methods, in: M. D'Agostino, D. M. Gabbay, R. Hähnle, J. Posegga (Eds.), Handbook of Tableau Methods, Springer Netherlands, Dordrecht, 1999, pp. 125–196. doi:10.1007/978-94-017-1754-0_3.
[20] T. Braüner, Hybrid Logic and its Proof-Theory, volume 37 of *Applied Logic Series*, Springer, 2011.