University of Pennsylvania
ScholarlyCommons

IRCS Technical Reports Series
Institute for Research in Cognitive Science

May 1996

A Decidable Predicate Logic of Knowledge

Giorgi Japaridze
University of Pennsylvania

Follow this and additional works at: http://repository.upenn.edu/ircs_reports

Japaridze, Giorgi, "A Decidable Predicate Logic of Knowledge" (1996). *IRCS Technical Reports Series*. 91.
http://repository.upenn.edu/ircs_reports/91

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-96-06.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/ircs_reports/91
For more information, please contact libraryrepository@pobox.upenn.edu.

A Decidable Predicate Logic of Knowledge

Abstract

The language we consider is that of classical first order logic augmented with the unary modal operator □. Sentences of this language are regarded as true or false in a *knowledge-base KB*, which is any finite set of □-free formulas. Truth of □α in *KB* is understood as that α is true in all classical models of *KB* and this interpretation is intended to capture the intuition "we *know* that α" behind □α.

The resulting logic is, in general, undecidable and not even semidecidable. However, there is a natural fragment of the above language, called the *constructive language*, which yields a decidable logic. The only syntactic constraint in the constructive language is that there exists x should always be followed by □. That is, we are not allowed to simply say "there is *x* such that ..." and we can only say "there is *x* for which we know that ...". Under this constraint, truth of there existsxα(x) will always imply that an object *x* for which α(x) holds not only exists, but can be effectively found. This is generally what we want of there exists in practical applications: knowing that "there exists a combination *c* that opens safe *S*" has no significance unless such a combination *c* can actually be found, which, in our semantics, will be equivalent to saying that there is *c* for which we know that *c* opens *S*. So, it is only truth of the sentence there existsc□*OPENS*(c,*S*) that really matters, and the latter, unlike there existsc□*OPENS*(c,*S*) is a perfectly legal formula of the constructive language.

I introduce a decidable sequent system *C K N* in the constructive language and prove its soundness and completeness with respect to the above semantics.

Comments

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-96-06.

This technical report is available at ScholarlyCommons: http://repository.upenn.edu/ircs_reports/91

Institute for Research in Cognitive Science

A Decidable Predicate Logic of Knowledge

Giorgi Japaridze

University of Pennsylvania
3401 Walnut Street, Suite 400C
Philadelphia, PA 19104-6228

May 1996

Site of the NSF Science and Technology Center for Research in Cognitive Science

IRCS Report 96--06

A decidable predicate logic of knowledge

Giorgi Japaridze *

Department of Computer and Information Science
University of Pennsylvania
200 S. 33rd Street
Philadelphia, PA 19104-6389, USA
giorgi@saul.cis.upenn.edu

May 6, 1996

Abstract

The language we consider is that of classical first order logic augmented with the unary modal operator $\Box$. Sentences of this language are regarded as true or false in a *knowledge-base* KB, which is any finite set of $\Box$-free formulas. Truth of $\Box\alpha$ in KB is understood as that α is true in all classical models of KB, and this interpretation is intended to capture the intuition "we *know* that α" behind $\Box\alpha$.

The resulting logic is, in general, undecidable and not even semi-decidable. However, there is a natural fragment of the above language, called the *constructive language*, which yields a decidable logic. The only syntactic constraint in the constructive language is that $\exists x$ should always be followed by $\Box$. That is, we are not allowed to simply say "there is x such that ...", and we can only say "there is x for which we know that ...". Under this constraint, truth of $\exists x\alpha(x)$ will always imply that an object x for which $\alpha(x)$ holds not only exists, but can

*The author is grateful for support from the University of Pennsylvania, the National Science Foundation (on grants CCR-9403447 and CCR-9057570), and the the Institute for Research in Cognitive Science at the University of Pennsylvania.

be effectively found. This is generally what we want of $\exists$ in practical applications: knowing that "there exists a combination c that opens safe S" has no significance unless such a combination c can actually be found, which, in our semantics, will be equivalent to saying that there is c for which we know that c opens S. So, it is only truth of the sentence $\exists c \Box OPENS(c, S)$ that really matters, and the latter, unlike $\exists c\, OPENS(c, S)$, is a perfectly legal formula of the constructive language.

I introduce a decidable sequent system CKN in the constructive language and prove its soundness and completeness with respect to the above semantics.

1 Introduction

The nonconstructive character of classical existential quantifier has many times been criticized. Letting alone the philosophy on the right of "existence" of the classical notion of existence, I will only point out that it has no practical meaning. Consider the sentence

$$\exists c OPENS(c, S),$$

asserting that there is a combination c that opens safe S. Knowing that this sentence is true has little significance unless we can actually find a particular combination which opens S. In other words, there must be a combination C such that we *know* that $OPENS(C, S)$ is true. This can be expressed by the sentence

$$\exists c \Box OPENS(c, S),$$

where $\Box$ is read as "we know that...".

This consideration suggests an idea how to make classical first order logic constructive and practically meaningful: first add to the language of the latter a knowledge operator $\Box$, and then restrict the resulting language by allowing usage of quantifiers only in combination with $\Box$ as in the above example. That is, we should not be allowed to simply say "there is x such that ...", and we can only say "there is x for which we *know* that ...".

On the second thought, existential quantifier is nothing but a "big disjunction", and one might ask the question why we don't impose similar restrictions on the usage of $\vee$. The point is that the disjunction

$$OPENS(C1, S) \vee OPENS(C2, S),$$

although not as good as

$$\Box OPENS(C1, S) \vee \Box OPENS(C2, S),$$

is still reasonably constructive as it envisages only a bounded number of (in particular, two) possibilities; if this disjunction is true, all we need to do to open S is to try both combinations $C1$ and $C2$, whereas knowing the truth of $\exists c OPENS(c, S)$ doesn't save our day unless dialing infinitely many, or, say, 2^{100} combinations, is feasible.

Our approach, on one hand, extends the expressive power of classical first order logic by adding the knowledge operator to it and, on the other hand, restricts some expressiveness of the latter by limiting the usage of quantifiers; as I tried to convince the reader, however, this restriction can be viewed as just cleansing classical logic of practically meaningless constructs.

Most importantly, as we will see later, our approach induces a decidable predicate logic, which nicely contrasts with the undecidability of classical logic, to say nothing about the non-semidecidability of the syntactic logics of knowledge ([3]) or epistemic logics studied within the framework of non-monotonic logics ([1], [2]).

2 The full language

We start by defining the syntax and semantics of the full language $\mathcal{L}$ of the predicate modal logic of knowledge.

$\mathcal{L}$ has an infinite set $\mathcal{V}$ of *variables*, a nonempty (finite or infinite) set $\mathcal{C}$ of *constants* and a nonempty (finite or infinite) set $\mathcal{R}$ of *predicate letters* together with a function that assigns to every $R \in \mathcal{R}$ a natural number called the *arity* of R. We also define the set of *terms* as $\mathcal{V} \cup \mathcal{C}$.

The set of *formulas* of $\mathcal{L}$ is the smallest set of expressions such that:

- $R(t_1, \ldots, t_n)$ is an (atomic) formula, for any n-ary relation symbol $R \in \mathcal{R}$ and any terms $t_1, \ldots, t_n$;
- if α is a formula, then $\neg(\alpha)$ is a formula;

- if α and β are formulas, then $(\alpha) \vee (\beta)$ is a formula;
- if α is a formula, then $\Box(\alpha)$ is a formula;
- if α is a formula and x is a variable, then $\exists x(\alpha)$ is a formula.

When this does not lead to confusions, we will be omitting some parentheses in formulas.

We will be using $\wedge$, $\rightarrow$, $\leftrightarrow$, $\forall$, $\Diamond$ (where $\Diamond = \neg\Box\neg$) as defined operators.

We also adopt the following standard notational convention: If $\alpha(x_1, \ldots, x_n)$ denotes a formula, where the x_i are variables (which do not necessarily have to have free occurrence in the formula, as well as not all free variables of the formula have to be among $x_1, \ldots, x_n$), then $\alpha(t_1, \ldots, t_n)$, where the t_i are terms, denotes the result of substituting each (free occurrence of each) x_i by t_i in $\alpha(x_1, \ldots, x_n)$.

Formulas without free variables will be called *sentences*, and formulas not containing $\Box$ will be said to be *pure*.

If $\alpha(x_1, \ldots, x_n)$ is a formula with exactly $x_1, \ldots, x_n$ free and $c_1, \ldots, c_n$ are constants, then $\alpha(c_1, \ldots, c_n)$ is said to be an *instance* of $\alpha(x_1, \ldots, x_n)$.

Definition 2.1 A *world* is a function w which assigns to each atomic sentence $R(\vec{c})$ one of the values $\{T(rue), F(alse)\}$. We write $\models_w \alpha$ for $w(\alpha) = T$.

The relation $\models_w$ is extended to all pure sentences in the following way:

- $\models_w \neg\alpha$ iff $\not\models_w \alpha$;
- $\models_w \alpha \vee \beta$ iff $\models_w \alpha$ or $\models_w \beta$;
- $\models_w \exists x \alpha(x)$ iff there is a constant c such that $\models_w \alpha(c)$.

Thus, a world w is nothing but a classical structure with the universe $\mathcal{C}$ and, for a pure sentence α, $\models_w \alpha$ means nothing but that α is classically true in this structure. Note the two simplifying assumptions we make vs the traditional approach: First, we assume that every object of the universe has a unique name in our language (a constant). Second, we identify these objects with their names. These assumptions make life much easier.

Definition 2.2 A *knowledge-base* is a finite (possibly empty) set of pure formulas.

Definition 2.3 A world w is said to be a *possible world* for a knowledge-base KB iff for every instance α' of every $\alpha \in KB$, $\models_w \alpha'$. This means nothing but that w, as a classical structure, is a model of KB.

A knowledge-base KB is said to be *consistent* iff it has at least one possible world, and KB is *complete* iff it has at most one possible world.

Intuitively, the knowledge-base is all our knowledge of the world. This knowledge is usually only partial unless the knowledge-base is complete. Different possible worlds correspond to different possible completions of the missing information, and they are equal candidates to be *the* (real) world.

The reason why we don't allow non-pure formulas in a knowledge-base is simple: the definition of the exact semantics of $\Box$ as a knowledge operator is going to appeal to what is contained in our knowledge-base, and including formulas containing $\Box$ in the latter would make that kind of definition intuitively circular. Also, we want our knowledge-base to contain only *objective information* — information about the outside world; such information is stable and we can safely expand it by adding new true facts to the knowledge-base, whereas, if we had, say, the formula $\neg\Box\alpha$ there, then adding, at some point, the knowledge α would make the knowledge-base intuitively inconsistent.

Definition 2.4 Let KB be a knowledge-base and w be a world. We say that a sentence ϕ is *true in KB with respect to w*, — and write $KB \models_w \phi$, iff one of the following conditions holds:

- ϕ is atomic and $\models_w \phi$;
- $\phi = \neg\alpha$ and $KB \not\models_w \alpha$;
- $\phi = \alpha \vee \beta$ and $KB \models_w \alpha$ or $KB \models_w \beta$;
- $\phi = \Box\alpha$ and for every possible world u for KB, $KB \models_u \alpha$;
- $\phi = \exists x\alpha(x)$ and, for some constant $c \in \mathcal{C}$, $KB \models_w \alpha(c)$.

And we say that a sentence ϕ is (simply) *true* in KB, — and write $KB \models \phi$, iff for every possible world w for KB, $KB \models_w \phi$. In other words, ϕ is true in KB iff $KB \models_w \Box\phi$ for any (or some) w.

Thus, intuitively, $\Box\alpha$ is true if we *know* that α, where knowing α means that the truth of α follows exclusively from our knowledge-base, so that it doesn't matter which of the possible worlds is *the* real world.

Note that if α is a pure sentence, then its truth in KB with respect to w does not depend on KB and $KB \models_w \alpha$ iff $\models_w \alpha$.

3 The constructive language

The *constructive language* $\mathcal{L}^c$, whose formulas will be referred to as *constructive formulas*, is the fragment of $\mathcal{L}$ where formulas are allowed to contain $\exists x$ only if it is immediately followed by $\Box$.

And a *constructive knowledge-base* is a knowledge-base consisting only of constructive formulas.

For a philosophy on why this fragment is natural and what it is good for see the Introduction.

Another way to present the constructive language is to take the full language $\mathcal{L}$ without any syntactic constraints but change the semantics of it so that $\exists x$ is simply understood as $\exists x\Box$. This might look more impressive but not quite fair, and we will not do that.

The above syntactic constraint may seem too inconvenient: nesting of quantifiers induces nesting of modal operators, and the meaning of a formula with deeply nested $\Box$'s becomes not very intuitive. However, one can show that every such formula is logically equivalent to a formula without nested modal operators. This is natural taking into account that our modal operator is in fact an $S5$-modality which, as it is well known, allows to eliminate nesting of $\Box$'s.

Also, theorem 3.1 below establishes that the constructive language has the same expressive power as the much bigger language called the *relaxed constructive language*, $\mathcal{L}^{rc}$, which is defined as the fragment of $\mathcal{L}$ where, whenever $\exists x$ is applied to a (sub)formula $\alpha(x)$, all free occurrences of x in the latter should be in the scope of $\Box$.

We say that two formulas $\alpha(x_1, \ldots, x_n)$ and $\beta(x_1, \ldots, x_n)$, whose all free variables are among $x_1, \ldots, x_n$, are (logically) *equivalent*, — and write $\alpha(x_1, \ldots, x_n) \equiv \beta(x_1, \ldots, x_n)$, iff for every knowledge-base KB, world w and tuple $c_1, \ldots, c_n$ of constants,

$$KB \models_w \alpha(c_1, \ldots, c_n) \;\Leftrightarrow\; KB \models_w \beta(c_1, \ldots, c_n).$$

For two sublanguages $L1$ and $L2$ of $\mathcal{L}$ we read $L1 \preceq L2$ as saying that there is an effective function $f: L1 \to L2$, called an *interpreter*, such that for every formula $\alpha \in L1$, $\alpha \equiv f(\alpha)$.

And we say that $L1$ and $L2$ are *equivalent* (in expressive power), iff $L1 \preceq L2$ and $L2 \preceq L1$.

Theorem 3.1 *The languages $\mathcal{L}^c$ and $\mathcal{L}^{rc}$ are equivalent.*

(Proof is given in Section 8.)

In view of this theorem, it suffices to study only $\mathcal{L}^c$, and we can safely use the more relaxed formulas of $\mathcal{L}^{rc}$, viewing them as shorthands for their equivalent $\mathcal{L}^c$-formulas and entrusting their legalization to the interpreter.

Allowing only constructive knowledge-bases means that the knowledge-bases (unlike queries) we consider cannot use quantifiers, because a constructive formula containing a quantifier should also contain a $\Box$, whereas a knowledge-base should consist of only pure formulas. This, too, may seem restrictive. However, the effect of external universal quantifiers in a constructive knowledge-base can be achieved by using free variables (which, we know, is legal), and most of the basic scientific or everyday knowledge, — whether it be general rules or individual facts, — does not require any other sort of quantification.

E.g., where $A(x, y, z)$ means $x + y = z$ and $S(x, y)$ means $x' = y$ (i.e. $x+1 = y$), the recursive definition of addition in terms of successor: $0+y = y$; $x' + y = (x + y)'$, — can be captured by the constructive knowledge-base consisting of the following two formulas:

- $A(0, y, y)$;
- $S(x_1, x_2) \wedge S(z_1, z_2) \wedge A(x_1, y, z_1) \to A(x_2, y, z_2)$.

To see possible applications of our logic in knowledge-base or database systems, consider an example knowledge-base of a dating service, which consists of the following constructive formulas:

1. $LIKES(Jon, x) \leftrightarrow BLONDE(x) \wedge GOODLOOKING(x)$ (a necessary and sufficient condition for Jon to like someone is that the someone is blonde and good-looking);
2. $LIKES(Bob, x) \rightarrow BLONDE(x)$ (Bob likes only blondes);
3. $LIKES(Bob, x) \rightarrow ASIAN(x)$ (Bob likes only Asians);
4. $ASIAN(x) \rightarrow \neg BLONDE(x)$ (no Asian is blonde);
5. $BLONDE(Ann)$;
6. $GOODLOOKING(Ann)$;
7. $ASIAN(Sue)$;
8. $BLONDE(Peg)$.

Is there an undoubted match for Jon? This query is expressedby

$$\exists x \Box LIKES(Jon, x),$$

and a system based on our logic would answer "YES" to this question. Then, as I promised that existential quantifier was going to be constructive in our logic, we could confidently ask the system to find a particular x for which $\Box LIKES(Jon, x)$ holds, and we would get $\Box LIKES(Jon, Ann)$ (Jon will *definitely* like Ann), so we would recommend Jon to meet Ann. We will also infer $\Diamond LIKES(Jon, Peg)$ (Jon *might* like Peg), so that it makes sense for Jon to try to find out more about Peg. And we will infer $\Box \neg LIKES(Jon, Sue)$ (Jon definitely will not like Sue), so Jon should not waste time on Sue. As for Bob, he will never find a match unless he reconsiders his taste: we can infer the (relaxed constructive) sentence $\forall x \neg \Diamond LIKES(Bob, x)$.

4 Logic CKN

We now describe a sequent calculus CKB. The singularity of CKN is that it has two sorts, — positive and negative, — of sequents.

A *sequent* is a triple $\Gamma \Rightarrow \Delta$ (positive sequent) or $\Gamma \not\Rightarrow \Delta$ (negative sequent), where Γ is a constructive knowledge-base and Δ is a finite set of constructive sentences.

The intended meaning of $\Gamma \Rightarrow \Delta$ (resp. $\Gamma \not\Rightarrow \Delta$) is that the disjunction of the elements of Δ is (resp. is not) true in the knowledge-base Γ.

"*Level-3 sequent*" is a synonym of "sequent".

A *level-2 sequent* is a sequent containing only pure formulas.

A *level-1 sequent* is a sequent containing only pure sentences.

Finally, a *level-0 sequent* is a sequent containing only atomic sentences.

By the standard abuse of notation, if Θ is a set of formulas and α is a formula, we will write "Θ, α" or "α, Θ" for $\Theta \cup \{\alpha\}$.

Without loss of generality we may assume that $\mathcal{C} = \{0, \ldots, n\}$ or $\mathcal{C} = \{0, 1, 2, \ldots\}$. Then we say that a constant c is *active* in a sequent S, if c occurs in some formula of S or c is the least constant not occurring in S. And c is *strictly active*, if c occurs in S or there are no constants in S and $c = 0$.

The *inference rules* listed below have the form

$$\frac{S_1 \ \ldots \ S_n}{S_0},$$

possibly $n = 0$, and possibly with some additional conditions on $S_0, S_1, \ldots, S_n$. S_0 is called the *conclusion* and $S_1, \ldots, S_n$ the *premises* of the rule.

We say that a set Sq of sequents is *closed under* a set Rl of rules, if, whenever

$$\frac{S_1 \ \ldots \ S_n}{S_0}$$

is a rule of Rl, $S'_0, S'_1, \ldots, S'_n$ are sequents of the form $S_0, S_1, \ldots, S_n$, respectively, and they satisfy all additional conditions (if any) stated in the rule, and if $n = 0$ or $S'_1, \ldots, S'_n \in Sq$, then $S'_0 \in Sq$.

In the rules below, $\leadsto$ is a variable ranging over $\{\Rightarrow, \not\Rightarrow\}$, so that each rule with $\leadsto$ in fact represents two rules, one with $\Rightarrow$ and the other with $\not\Rightarrow$. Also, all the sequents in a level-i rule ($i = 0, 1, 2, 3$) are assumed to be level-i sequents.

The *logic* CKN is defined as the smallest set of sequents closed under the following rules:

LEVEL-0 RULES (AXIOMS):

R0($\Rightarrow$):

$$\overline{\Gamma \Rightarrow \Delta},$$

where $\Gamma \cap \Delta$ is nonempty.

R0($\not\Rightarrow$):

$$\overline{\Gamma \not\Rightarrow \Delta},$$

where $\Gamma \cap \Delta$ is empty.

LEVEL-1 RULES:

R1($\leadsto \neg$):

$$\frac{\Gamma, \alpha \leadsto \Delta}{\Gamma \leadsto \neg\alpha, \Delta}.$$

R1($\neg \leadsto$):

$$\frac{\Gamma \leadsto \alpha, \Delta}{\Gamma, \neg\alpha \leadsto \Delta}.$$

R1($\leadsto \vee$):

$$\frac{\Gamma \leadsto \alpha_1, \alpha_2, \Delta}{\Gamma \leadsto \alpha_1 \vee \alpha_2, \Delta}.$$

R1($\vee \Rightarrow$):

$$\frac{\Gamma, \alpha_1 \Rightarrow \Delta \qquad \Gamma, \alpha_2 \Rightarrow \Delta}{\Gamma, \alpha_1 \vee \alpha_2 \Rightarrow \Delta}.$$

R1($\vee \not\Rightarrow$):

$$\textbf{a)}\ \frac{\Gamma, \alpha_1 \not\Rightarrow \Delta}{\Gamma, \alpha_1 \vee \alpha_2 \not\Rightarrow \Delta}; \qquad \textbf{b)}\ \frac{\Gamma, \alpha_2 \not\Rightarrow \Delta}{\Gamma, \alpha_1 \vee \alpha_2 \not\Rightarrow \Delta}.$$

LEVEL-2 RULES:

R2($\rightsquigarrow$):

$$\frac{\Gamma, \alpha(c_1), \ldots, \alpha(c_n) \rightsquigarrow \Delta}{\Gamma, \alpha(x) \rightsquigarrow \Delta},$$

where $c_1, \ldots, c_n$ are all the strictly active constants of the conclusion.

LEVEL-3 RULES:

R3($\rightsquigarrow \neg\neg$):

$$\frac{\Gamma \rightsquigarrow \alpha, \Delta}{\Gamma \rightsquigarrow \neg\neg\alpha, \Delta}.$$

R3($\rightsquigarrow \vee$):

$$\frac{\Gamma \rightsquigarrow \alpha_1, \alpha_2, \Delta}{\Gamma \rightsquigarrow \alpha_1 \vee \alpha_2, \Delta}.$$

R3($\Rightarrow \neg\vee$):

$$\frac{\Gamma \Rightarrow \neg\alpha_1, \Delta \qquad \Gamma \Rightarrow \neg\alpha_2, \Delta}{\Gamma \Rightarrow \neg(\alpha_1 \vee \alpha_2), \Delta}.$$

R3($\not\Rightarrow \neg\vee$):

$$\textbf{a)}\ \frac{\Gamma \not\Rightarrow \neg\alpha_1, \Delta}{\Gamma \not\Rightarrow \neg(\alpha_1 \vee \alpha_2), \Delta}; \qquad \textbf{b)}\ \frac{\Gamma \not\Rightarrow \neg\alpha_2, \Delta}{\Gamma \not\Rightarrow \neg(\alpha_1 \vee \alpha_2), \Delta}.$$

R3($\Rightarrow \Box$):

$$\textbf{a)}\ \frac{\Gamma \Rightarrow \alpha}{\Gamma \Rightarrow \Box\alpha, \Delta}; \qquad \textbf{b)}\ \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Box\alpha, \Delta}.$$

R3($\not\Rightarrow \Box$):

$$\frac{\Gamma \not\Rightarrow \alpha \qquad \Gamma \not\Rightarrow \Delta}{\Gamma \not\Rightarrow \Box\alpha, \Delta}.$$

R3($\Rightarrow \neg\Box$):

$$\textbf{a)}\ \frac{\Gamma \not\Rightarrow \alpha}{\Gamma \Rightarrow \neg\Box\alpha, \Delta}; \qquad \textbf{b)}\ \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \neg\Box\alpha, \Delta}.$$

R3($\not\Rightarrow \neg\Box$):

$$\frac{\Gamma \Rightarrow \alpha \qquad \Gamma \not\Rightarrow \Delta}{\Gamma \not\Rightarrow \neg\Box\alpha, \Delta}.$$

R3($\Rightarrow \exists$):

$$\frac{\Gamma \Rightarrow \alpha(c), \Delta}{\Gamma \Rightarrow \exists x\alpha(x), \Delta},$$

where c is an active constant of the conclusion.

R3($\not\Rightarrow \exists$):

$$\frac{\Gamma \not\Rightarrow \alpha(c_1), \Delta \qquad \cdots \qquad \Gamma \not\Rightarrow \alpha(c_n), \Delta}{\Gamma \not\Rightarrow \exists x\alpha(x), \Delta},$$

where $c_1, \ldots, c_n$ are all the active constants of the conclusion.

R3($\Rightarrow \neg\exists$):

$$\frac{\Gamma \Rightarrow \neg\alpha(c_1), \Delta \qquad \cdots \qquad \Gamma \Rightarrow \neg\alpha(c_n), \Delta}{\Gamma \Rightarrow \neg\exists x\alpha(x), \Delta},$$

where $c_1, \ldots, c_n$ are all the active constants of the conclusion.

R3($\not\Rightarrow \neg\exists$):

$$\frac{\Gamma \not\Rightarrow \neg\alpha(c), \Delta}{\Gamma \not\Rightarrow \neg\exists x\alpha(x), \Delta},$$

where c is an active constant of the conclusion.

5 The main results

The relation $KB \models \alpha$ is naturally extended to $KB \models \Delta$, where Δ is any finite set of sentences, in the following way: Let $\vee\Delta$ be the disjunction of all the elements of Δ. We may assume that we have an always-false atomic sentence $\perp$ in the language and, if Δ is empty, understand $\vee\Delta$ as $\perp$. Then we define $KB \models \Delta$ as $KB \models \vee\Delta$. Our original relation $KB \models \alpha$ is thus a special case of $KB \models \Delta$ where $\Delta = \{\alpha\}$. Notice also that $KB \models \perp$ means nothing but that KB is inconsistent.

As CKN is in fact a deductive system (with the conclusions of the level-0 rules as axioms and all the other rules as proper rules of inference), we will write $CKN \vdash S$ for $S \in CKN$.

Lemma 5.1 (Dual soundness of CKN) *For any sequent $KB \Rightarrow \Delta$,*

- **a)** *If $CKN \vdash KB \Rightarrow \Delta$, then $KB \models \Delta$.*
- **b)** *If $CKN \vdash KB \not\Rightarrow \Delta$, then $KB \not\models \Delta$.*

(Proof is given in Section 6.)

Lemma 5.2 (Syntactic completeness of CKN) *For any sequent $KB \Rightarrow \Gamma$, either $CKN \vdash KB \Rightarrow \Gamma$ or $CKN \vdash KB \not\Rightarrow \Gamma$.*

(Proof is given in Section 7.)

Theorem 5.3 *CKN is decidable.*

Proof: This is an immediate consequence of the above two lemmas, taking into account that the rules of CKN are effective. **End of proof.**

Theorem 5.4 (Soundness and completeness of CKN) *For any sequent $KB \Rightarrow \Delta$,*

$$KB \models \Delta \quad \textit{iff} \quad CKN \vdash KB \Rightarrow \Delta.$$

Proof: The "if" part has been established in Lemma 5.1a. For the "only if" part, suppose $CKN \not\vdash KB \Rightarrow \Delta$. Then, by Lemma 5.2, $CKN \vdash KB \not\Rightarrow \Delta$, whence, by Lemma 5.1b, $KB \not\models \Delta$. **End of proof.**

Fact 5.5 (Constructiveness of $\exists$) *There is an effective method which, for any constructive knowledge-base KB and constructive sentence $\exists x\alpha(x)$ with $KB \models \exists x\alpha(x)$, finds a constant c such that $KB \models \alpha(c)$.*

Proof: If $KB \models \exists x\alpha(x)$, then, by 5.4, CKN proves $KB \Rightarrow \exists x\alpha(x)$. The last rule in that proof can be only **R3($\Rightarrow \exists$)**, which means that $CKN \vdash KB \Rightarrow \alpha(c)$ for some constant c active in $KB \Rightarrow \exists x\alpha(x)$. Check whether $CKN \vdash KB \Rightarrow \alpha(c)$ for each such constant c, and return a c for which you get a positive answer. In view of the decidability of CKN, this can be done effectively. **End of proof.**

6 Proof of Lemma 5.1

We proceed by induction on the length of a CKN-proof of the sequent. $KB \Rightarrow \Delta$ or $KB \not\Rightarrow \Delta$ should be the conclusion of one of the 26 rules of CKN, and, correspondingly, we need to consider 26 cases.

For better readability, we will identify Δ with $\vee\Delta$.

Recall that when α is a pure sentence (and so are all the formulas in level-0 and level-1 rules, as well as the instances of formulas in level-2 rules), then $KB \models_w \alpha$ iff $\models_w \alpha$.

Case **R0($\Rightarrow$)**: Let $\alpha \in \Gamma \cap \Delta$ (since $\Gamma \cap \Delta$ is nonempty in this rule, such an α exists). Then, for every possible world w for Γ, we have $\models_w \alpha$, which implies that $\Gamma \models \Delta$ because α is a disjunct of Δ.

Case **R0($\not\Rightarrow$)**: Let w be the world such that, for every atomic sentence α, we have $\models_w \alpha$ iff $\alpha \in \Gamma$. Thus, w is a possible world for Γ. On the other hand, $\not\models_w \Delta$ because, since $\Gamma \cap \Delta$ is empty, for no disjunct β of Δ do we have $\models_w \beta$. Thus, $\Gamma \not\models \Delta$.

Case **R1($\Rightarrow \neg$)**: Suppose $\Gamma, \alpha \models \Delta$ (the induction hypothesis). We need to show that $\Gamma \models \neg\alpha, \Delta$. Let w be an arbitrary possible world for Γ. It suffices to show that $\models_w \neg\alpha, \Delta$. If $\models_w \neg\alpha$, we are done; otherwise we have $\models_w \alpha$, which means that w is a possible world for Γ, α, whence (as $\Gamma, \alpha \models \Delta$) $\models_w \Delta$, and we are done again.

Case **R1($\not\Rightarrow \neg$)**: Suppose $\Gamma, \alpha \not\models \Delta$ (the induction hypothesis). We need to show that $\Gamma \not\models \neg\alpha, \Delta$. Let w be a possible world for Γ, α such that $\not\models_w \Delta$. But notice that $\not\models_w \neg\alpha$ and, therefore, $\not\models_w \neg\alpha, \Delta$, which (as we deal with pure sentences) means that $\Gamma \not\models \neg\alpha, \Delta$.

Cases of the remaining level-1 rules are similar.

Case **R2($\Rightarrow$)**: It suffices to observe that every possible world for $\Gamma, \alpha(x)$ is a possible world for $\Gamma, \alpha(c_1), \ldots, \alpha(c_n)$.

Case **R2($\not\Rightarrow$)**: Suppose $\Gamma, \alpha(c_1), \ldots, \alpha(c_n) \not\models \Delta$. We need to show that $\Gamma, \alpha(x) \not\models \Delta$. Let w be a possible world for $\Gamma, \alpha(c_1), \ldots, \alpha(c_n)$ such that

$\not\models_w \Delta$. For every formula β, let β^* denote the result of replacing, in β, every constant $c \notin \{c_1, \ldots, c_n\}$ by c_1. Let u be the world such that for every atomic sentence γ, $\models_u \gamma$ iff $\models_w \gamma^*$. It is easy to verify, by induction on the complexity of σ, that for any (pure constructive) sentence σ,

$$\models_u \sigma \textit{ iff } \models_w \sigma^*. \tag{1}$$

Therefore, since $\Delta^* = \Delta$ and $\not\models_w \Delta$, we have $\not\models_u \Delta$. So, it remains to show that u is a possible world for $\Gamma, \alpha(x)$.

First, consider an arbitrary $\gamma(x_1, \ldots, x_m) \in \Gamma$, whose free variables are exactly $x_1, \ldots, x_m$. Let $d_1, \ldots, d_m$ be any constants. We need to show that $\models_u \gamma(d_1, \ldots, d_m)$, i.e., in view of (1), that $\models_w \gamma(d_1, \ldots, d_m)^*$. But notice that $\gamma(d_1, \ldots, d_m)^*$ is an instance of $\gamma(x_1, \ldots, x_m)$, and since w is a possible world for Γ, we, indeed, have $\models_w \gamma(d_1, \ldots, d_m)^*$.

Now it remains to consider instances of $\alpha(x)$. Suppose all the free variables of $\alpha(x)$ are among $x, x_1, \ldots, x_m$, so that $\alpha(x) = \alpha(x, x_1, \ldots, x_m)$. Let $d, d_1, \ldots, d_m$ be arbitrary constants. We need to show that $\models_u \alpha(d, d_1, \ldots, d_m)$, i.e., in view of (1), that $\models_w \alpha(d, d_1, \ldots, d_m)^*$. But notice that if $d = c_i$ for some $c_i \in \{c_1, \ldots, c_n\}$, then $\alpha(d, d_1, \ldots, d_m)^*$ is an instance of $\alpha(c_i)$, and otherwise it is an instance of $\alpha(c_1)$. In either case, since w is a possible world for $\Gamma, \alpha(c_1), \ldots, \alpha(c_n)$, we have $\models_w \alpha(d, d_1, \ldots, d_m)^*$.

Cases **R3**($\Rightarrow \neg\neg$), **R3**($\not\Rightarrow \neg\neg$), **R3**($\Rightarrow \vee$), **R3**($\not\Rightarrow \vee$), **R3**($\Rightarrow \neg\vee$), **R3**($\not\Rightarrow \neg\vee$) are rather straightforward.

Case **R3**($\Rightarrow \Box$): The subcase **(b)** is straightforward and for the subcase **(a)** it suffices to observe that $\Gamma \models \alpha$ implies $\Gamma \models \Box\alpha$.

Case **R3**($\not\Rightarrow \Box$): Suppose $\Gamma \not\models \alpha$ and $\Gamma \not\models \Delta$. Let w be a possible world for Γ such that $\Gamma \not\models_w \Delta$. Observe that then $\Gamma \not\models_w \Box\alpha, \Delta$. Hence, $\Gamma \not\models \Box\alpha, \Delta$.

Case **R3**($\Rightarrow \neg\Box$): The subcase **(b)** is straightforward and for the subcase **(a)** it suffices to observe that $\Gamma \not\models \alpha$ implies $\Gamma \models \neg\Box\alpha$.

Case **R3**($\not\Rightarrow \neg\Box$): Similar to case **R3**($\not\Rightarrow \Box$).

Case **R3**($\Rightarrow \exists$) is straightforward.

Case **R3($\not\Rightarrow$ ∃)**: Suppose $\Gamma \not\models \alpha(c_1), \Delta$ and ... and $\Gamma \not\models \alpha(c_n), \Delta$. Since we deal with constructive sentences, $\alpha(x)$ must have the form $\Box\beta(x)$. Thus, we have

$$\Gamma \not\models \Delta \tag{2}$$

and

$$\Gamma \not\models \Box\beta(c_1), \quad \ldots, \quad \Gamma \not\models \Box\beta(c_n). \tag{3}$$

We claim that

$$\textit{For every constant } c, \;\; \Gamma \not\models \Box\beta(c). \tag{4}$$

Indeed, if $c \in \{c_1, \ldots, c_n\}$, then $\Gamma \not\models \Box\beta(c)$ by (3). Suppose now $c \notin \{c_1, \ldots, c_n\}$. We may suppose that c_n is the constant that does not appear in the conclusion of the rule. Let w be a possible world for Γ such that $\Gamma \not\models_w \beta(c_n)$. By (3), such a world exists. Let then u be the world that evaluates every atom just as w does, only with the roles of c and c_n interchanged. Since neither c nor c_n appear in Γ or $\beta(x)$, it is clear that u, just as w, is a possible world for Γ and also (as $\Gamma \not\models_w \beta(c_n)$) we have $\Gamma \not\models_u \beta(c)$. Hence, $\Gamma \not\models \Box\beta(c)$ and (4) is thus proved.

Clearly (4) implies that for every world v, $\Gamma \not\models_v \exists x \Box\beta(x)$, and this, together with (2), implies that $\Gamma \not\models \exists x \Box\beta(x), \Delta$.

Case **R3($\Rightarrow$ ¬∃)**: As in the previous case, $\alpha(x)$ must have the form $\Box\beta(x)$. So, suppose $\Gamma \models \neg\Box\beta(c_1), \Delta$ and ... and $\Gamma \models \neg\Box\beta(c_n), \Delta$. If $\Gamma \models \Delta$, then $\Gamma \models \exists x \neg\Box\beta(x), \Delta$ and we are done. Otherwise, let w be a world such that $\Gamma \not\models_w \Delta$. Consider any $c_i \in \{c_1, \ldots, c_n\}$. We have $\Gamma \models_w \neg\Box\beta(c_i), \Delta$ and $\Gamma \not\models_w \Delta$. Hence, $\Gamma \models_w \neg\Box\beta(c_i)$. Consequently, there is a possible world u for Γ such that $\Gamma \not\models_u \beta(c_i)$, and this implies that $\Gamma \models \neg\Box\beta(c_i)$. Thus, we have:

$$\Gamma \models \neg\Box\beta(c_1), \quad \ldots, \quad \Gamma \models \neg\Box\beta(c_n).$$

Using an argument similar to the one employed in the proof of (4), we get that for every constant c, $\Gamma \models \neg\Box\beta(c)$. This implies that $\Gamma \models \neg\exists x \Box\beta(x)$, and thus $\Gamma \models \neg\exists x \Box\beta(x), \Delta$.

Case **R3($\not\Rightarrow$ ¬∃)** is simple.

Lemma 5.1 is proved.

7 Proof of Lemma 5.2

Define the *complexity of a formula* α as the number of occurrences of logical operators in α plus the number of distinct free variables of α. Next, define the *complexity of a sequent* S as the infinite sequence $\langle a_0, a_1, \ldots \rangle$, where each a_i is the number of formulas of S of complexity i. Define the well-ordering relation $\prec$ on such complexities by: $\langle a_0, a_1, \ldots \rangle \prec \langle b_0, b_1, \ldots \rangle$ iff there is i such that $a_i < b_i$ and, for all j with $j > i$, $a_j = b_j$.[1]

Now we can prove the lemma by induction on the complexity of $KB \Rightarrow \Delta$.

Suppose $KB \Rightarrow \Delta$ is a level-0 sequent. $KB \cap \Delta$ is either empty or nonempty. In the first case $CKN \vdash KB \not\Rightarrow \Delta$ by **R0**($\not\Rightarrow$), and in the second case $CKN \vdash KB \Rightarrow \Delta$ by **R0**($\Rightarrow$).

Suppose now $KB \Rightarrow \Delta$ is a level-i sequent but not level-$(i-1)$ sequent for some $i \in \{1,2,3\}$. Note that then it matches the conclusion of one of the level-i rules with a positive sequent in the conclusion. There are thus 12 cases to consider: **R1**($\Rightarrow \neg$), **R1**($\neg \Rightarrow$), **R1**($\Rightarrow \vee$), **R1**($\vee \Rightarrow$), **R2**($\Rightarrow$), **R3**($\Rightarrow \neg\neg$), **R3**($\Rightarrow \vee$), **R3**($\Rightarrow \neg\vee$), **R3**($\Rightarrow \Box$), **R3**($\Rightarrow \neg\Box$), **R3**($\Rightarrow \exists$), **R3**($\Rightarrow \neg\exists$). We will consider only one of them, **R1**($\Rightarrow \neg$), as an example, and all the other cases can be handled in a rather similar way.

So, suppose $KB \Rightarrow \Delta$ is a level-1 sequent of the form $\Gamma \Rightarrow \neg\alpha, \Delta'$, where (we may suppose) $\neg\alpha \notin \Delta'$. If CKN does not prove this sequent, then, in view of **R1**($\Rightarrow \neg$), $CKN \not\vdash \Gamma, \alpha \Rightarrow \Delta'$. Note that $\Gamma, \alpha \Rightarrow \Delta'$ has a strictly lower complexity than $\Gamma \Rightarrow \neg\alpha, \Delta'$. Therefore, by the induction hypothesis, $CKN \vdash \Gamma, \alpha \not\Rightarrow \Delta'$. But then, by **R1**($\not\Rightarrow$), $CKN \vdash \Gamma \not\Rightarrow \neg\alpha, \Delta'$.

Lemma 5.2 is proved.

8 Proof of Theorem 3.1

Let us say that two formulas α and β are *mutually safe* if they have exactly the same free variables, and for every such variable x, if all free occurrences of x in α are in the scope of $\Box$, then so are they in β, and vice versa.

We will say that α and β are *safely equivalent*, — and write $\alpha \equiv\equiv \beta$, if α and β are mutually safe and $\alpha \equiv \beta$.

[1]Thus, $\prec$ is the standard ordering relation on ordinals less than ω^ω, where each complexity $\langle a_0, a_1, a_2, \ldots \rangle$ is represented by the ordinal $\ldots + a_2 \cdot \omega^2 + a_1 \cdot \omega^1 + a_0 \cdot \omega^0$.

The following lemma can be verified by a routine analysis of the appropriate definitions, and we state it without a proof:

Lemma 8.1 *Let α and β be any formulas of $\mathcal{L}$ and x be any variable.*

1. *If $\alpha \equiv\equiv \beta$ and the formula $A(\beta)$ is the result of replacing α by β in the formula $A(\alpha)$, then $A(\alpha) \equiv\equiv A(\beta)$.*

2. *If $\alpha \leftrightarrow \beta$ is a classical propositional tautology, then $\alpha \equiv \beta$; if, at the same time, α and β are mutually safe, then $\alpha \equiv\equiv \beta$.*

3. *$\exists x(\alpha \vee \beta) \equiv\equiv \exists x\alpha \vee \exists x\beta$.*

4. *If α does not contain x free, then $\exists x(\alpha \wedge \beta) \equiv\equiv \alpha \wedge \exists x\beta$.*

5. *$\Box(\alpha \wedge \beta) \equiv\equiv \Box\alpha \wedge \Box\beta$.*

6. *$\Box\Box\alpha \equiv\equiv \Box\alpha$.*

7. *$\Box\neg\Box\alpha \equiv\equiv \neg\Box\alpha$.*

8. *$\Box\exists x\Box\alpha \equiv\equiv \exists x\Box\alpha$.*

9. *$\Box\neg\exists x\Box\alpha \equiv\equiv \neg\exists x\Box\alpha$.*

We now start proving Theorem 3.1. $\mathcal{L}^c \preceq \mathcal{L}^{rc}$ holds trivially, so we only need to show that $\mathcal{L}^{rc} \preceq \mathcal{L}^c$.

Let ϕ be an arbitrary formula of $\mathcal{L}^{rc}$. Below we give an interpreter's strategy converting ϕ into a safely equivalent constructive formula. The correctness of this strategy is verified by induction on the complexity of ϕ. We will be using 8.1.1 without explicitly referring to it.

If ϕ is atomic, return ϕ unchanged.

If $\phi = \neg\alpha$, then convert α into a safely equivalent constructive formula α' (which, by the induction hypothesis, can be done), and return $\neg\alpha'$. By 8.1.1, $\neg\alpha \equiv\equiv \neg\alpha'$.

Similarly if $\phi = \alpha \vee \beta$ or $\phi = \Box\alpha$.

Now, suppose $\phi = \exists x\alpha$. First convert α into a safely equivalent constructive formula α_1. Next, convert α_1 into a formula α_2 such that $\alpha \leftrightarrow \alpha_2$ is a tautology and

$$\alpha_2 = \beta_1 \vee \ldots \vee \beta_n$$

where, for each $1 \leq i \leq n$,

$$\beta_i = \gamma_1^i \wedge \ldots \wedge \gamma_{k_i}^i \wedge \delta_1^i \wedge \ldots \wedge \delta_{m_i}^i,$$

where each γ_j^i is an atom with or without negation, and each δ_j^i is of the form $\Box\delta$, $\neg\Box\delta$, $\exists y \Box\delta$ or $\neg\exists y \Box\delta$. That is, convert α_1 into a tautologically equivalent disjunctive normal form, where formulas of the form $\Box\delta$ and $\exists y \Box\delta$ are treated as propositional atoms. Naturally, we suppose that each such "atom" actually has an occurrence in α_1 and that occurrence is not in the scope of a non-Boolean operator ($\exists$ or $\Box$). In view of this, note that

$$\textit{no } \gamma_j^i \textit{ contains } x, \tag{5}$$

for otherwise α_1 would have an occurrence of x not in the scope of $\Box$ and (as α_1 and α are mutually safe) so would have α, which would contradict our assumption that $\exists x\alpha$ is a formula of $\mathcal{L}^{rc}$.

Clearly α_1 and α_2 are mutually safe and therefore, by 8.1.2, $\alpha_2 \equiv\equiv \alpha_1$, whence $\alpha_2 \equiv\equiv \alpha$. Note also that, since α_1 is constructive, so is every (γ_j^i and) δ_j^i.

For each $1 \leq i \leq n$, let

$$\sigma_i = \gamma_1^i \wedge \ldots \wedge \gamma_{k_i}^i \wedge \exists x \Box(\delta_1^i \wedge \ldots \wedge \delta_{m_i}^i).$$

Thus, σ_i is constructive. We claim that

$$\sigma_i \equiv\equiv \exists x \beta_i. \tag{6}$$

To show this, first note that, by (5) and 8.1.4,

$$\exists x \beta_i \equiv\equiv \gamma_1^i \wedge \ldots \wedge \gamma_{k_i}^i \wedge \exists x(\delta_1^i \wedge \ldots \wedge \delta_{m_i}^i). \tag{7}$$

By 8.1.6-9,

$$\delta_1^i \wedge \ldots \wedge \delta_{m_i}^i \equiv\equiv \Box\delta_1^i \wedge \ldots \wedge \Box\delta_{m_i}^i,$$

whence, by 8.1.5,

$$\delta_1^i \wedge \ldots \wedge \delta_{m_i}^i \equiv\equiv \Box(\delta_1^i \wedge \ldots \wedge \delta_{m_i}^i).$$

Hence,

$$\exists x(\delta_1^i \wedge \ldots \wedge \delta_{m_i}^i) \equiv\equiv \exists x \Box(\delta_1^i \wedge \ldots \wedge \delta_{m_i}^i)$$

which, together with (7), implies that $\sigma_i \equiv\equiv \exists x\beta_i$. (6) is thus proved.

Let

$$\phi' = \sigma_1 \vee \ldots \vee \sigma_n.$$

In view of (6),

$$\phi' \equiv\equiv \exists x\beta_1 \vee \ldots \vee \exists x\beta_n,$$

whence, by 8.1.3,

$$\phi' \equiv\equiv \exists x(\beta_1 \vee \ldots \vee \beta_n),$$

i.e. $\phi' \equiv\equiv \exists x\alpha_2$. But we know that $\alpha_2 \equiv\equiv \alpha$. Hence, $\phi' \equiv\equiv \exists x\alpha$. And as the σ_i's are constructive, ϕ' is constructive, too.

So, let the interpreter return ϕ' for our initial formula $\exists x\alpha$.

This completes the proof of Theorem 3.1.

References

[1] Dov M.Gabbay, C.J.Hogger, J.A.Robinson, D.Nute, eds. **Handbook of Logic in AI and Logic Programming**, vol.3. Clarendon Press, Oxford, 1994.

[2] K.Konolige, *On the relation between default and autoepistemic logic*, in: **Proceedings of the International Joint Conference on Artificial Intelligence**, Detroit, MI, 1989.

[3] L.Morgenstern, *Foundations of a Logic of Knowledge, Action and Communication*. Dissertation. New York University, 1988.