Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

UNIVERSAL WISER PUBLISHER

Research Article

Cut-Free Sequent Calculus for Multi-Agent Logic of Common Knowledge

Romas Alonderis[1], Haroldas Giedra[2*]

[1] Institute of Data Science and Digital Technologies, Vilnius University, Vilnius, Lithuania
[2] Institute of Computer Science, Vilnius University, Vilnius, Lithuania
E-mail: haroldas.giedra@mif.vu.lt

Received: 2 September 2024; **Revised:** 29 October 2024; **Accepted:** 6 November 2024

Abstract: Cut-free sequent calculi are handy tools for backward proof-search of logical formulas or sequents. In the present paper, we introduce a Gentzen-type sequent calculus for the logic of common knowledge. To maintain a deterministic backward proof-search process, we do not include cut or cut-like rules in the introduced calculus. Also, derivation loops are used to define provable sequents and to establish termination of backward proof-search. Using this sound and complete finitary loop-type sequent calculus we construct a decision procedure for the logic of common knowledge. The procedure allows to efficiently determine whether an arbitrary formula or sequent is valid in the logic.

***Keywords*:** common knowledge, sequent calculus, loops, decidability

MSC: 03B42, 03B70, 03F52

Abbreviation

LCK	Logic of common knowledge
GLCK	Gentzen-type sequent calculus for Logic of Common Knowledge

1. Introduction

The logic of common knowledge was introduced in [1]. **LCK** is based on multi-modal logics extended with the common knowledge operator. Common knowledge of a proposition p can be characterized by the following infinitary conjunction: everybody knows p and everybody knows that everybody knows p and everybody knows that everybody knows that everybody knows p and so on. **LCK** has many applications and plays a significant role in computer science, artificial intelligence, game theory, etc. [2].

A cut-free sequent calculus is a handy tool for backward proof-search of logical formulas or sequents. We choose a formula in the antecedent or succedent of a considered sequent and detect the outermost operator (logical or modal) of the formula. According to the operator and its location (antecedent or succedent), a unique derivation rule of the sequent calculus is chosen and backward applied to the sequent. As a result, one or several sequents, depending on the shape of

DOI: https://doi.org/10.37256/cm.6220255621

applied rule, are obtained. The process is continued for the obtained sequents, etc. A backward proof-search tree with some sequent S at the root is generated. The tree is a proof of S if all leaves of the tree satisfy some conditions, e.g., the leaves are axioms. The rule of cut complicates this deterministic backward proof-search process. It has the following shape:

$$\frac{\Gamma \Rightarrow \Delta,\ c \quad c,\ \Pi \Rightarrow \Lambda}{\Pi,\ \Gamma \Rightarrow \Delta,\ \Lambda}\ (cut).$$

In general, it is not clear what the formula c, called the cut-formula, should be so that to be convinced that a considered sequent is provable or non-provable. That makes the rule non-deterministic from the backward perspective. The same is true for cut-like rules such as the invariant rule in [3]. Hilbert-type calculi contain so called Modus Ponens rule

$$\frac{A \quad A \Rightarrow B}{B}.$$

This rule is an analogue of the cut rule. Such calculi are usually not used for efficient proof-search of formulas, e.g., in decision procedures. Hilbert-type calculi for **LCK** are considered in [4–6].

It was shown in [7] that common Gentzen-type sequent calculi for **LCK** are not complete without a cut rule. A complete cut-free calculus with an infinitary rule (a rule with infinitely many premises) is considered in [7, 8]. A sequent calculus with such an infinitary rule for a fragment of first order linear time logic was presented in [9]. It was proved in [10] that the infintary rule in [7] can be turned into a finitary one by restricting the number of its premises. A survey of proof theory of common knowledge can be found in [11].

Cut-free sequent calculi commonly possess the sub-formula property: each sequent in a backward proof-search tree with a sequent S at the root consists of the subformulas of formulas in S. This property is used to establish termination of backward proof-search in cases when the premises of some derivation rules are not simpler than the corresponding conclusions.

Cut-free sequent calculi for modal and temporal logics are considered in the literature, e.g., [12–16]. The loop-type sequent calculi for the logic of likelihood and the linear tense temporal logic are introduced in [17, 18], respectively.

The Gentzen-type sequent calculus **GLCK** for **LCK** is introduced in the present paper. The calculus does not contain cut and cut-like rules. In addition to axioms, we use so-called derivation loops to define provable in **GLCK** sequents. The derivation loops are also used to terminate potentially infinite backward proof-search. We prove that **GLCK** is sound and complete for **LCK**. Using the introduced calculus **GLCK**, a decision procedure for **LCK** is constructed. The procedure allows us to determine efficiently whether an arbitrary formula or sequent is valid in **LCK**.

The present paper is structured as follows. Syntax and semantics of **LCK** are presented in Section 2. The sequent calculus **GLCK** is introduced in Section 3. The derivation loops and connected with them notions are defined in Section 4. The soundness and completeness of **GLCK** are proved in Sections 5 and 6, respectively. The validity-check procedure for **LCK** based on the calculus **GLCK** is described in Section 7.

2. Syntax and semantics of LCK

We consider a language with μ agents for some natural number $\mu > 0$. Formulas of **LCK** consist of propositional variables, the logical connectives $\vee$, $\wedge$, $\rightarrow$, $\neg$, the modal operators K_i ($1 \leq i \leq \mu$), and C. Each propositional variable is a formula, and such a formula is called atomic. Formulas of **LCK** are defined as follows

$$\phi := p|(\phi \vee \psi)|(\phi \wedge \psi)|(\phi \rightarrow \psi)|\neg\phi|K_i\phi|C\phi,$$

where $1 \leq i \leq \mu$ and p is an atomic formula.

The formula $C\phi$ is understood intuitively as the infinite conjunction $\bigwedge_{i=1}^{\infty} E^i\phi$, where $E \overset{\text{def}}{=} \bigwedge_{i=1}^{\mu} K_i\phi$ and

$$E^m\phi = \underbrace{E \dots E}_{m}\phi.$$

The formula $K_i\phi$ means "agent i knows ϕ", $E\phi$ means "every agent knows ϕ", and $C\phi$ means "ϕ is common knowledge of all agents".

Subformulas of a formula are defined in the traditional way, e.g., the subformulas of $(\phi \to \psi)$ are $(\phi \to \psi)$ and all subformulas of ϕ and ψ. We do not burden formulas with parentheses if it causes no ambiguities, e.g., we write $(\phi \wedge \psi) \to \varphi$ instead of $((\phi \wedge \psi) \to \varphi)$.

In the paper, we use (1) the letters p, q, and r, possibly subscripted, to denote propositional variables and (2) the letters ϕ, φ, and ψ, possibly subscripted, to denote arbitrary formulas.

Sequents are objects of the type $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite, possibly empty, multisets of formulas. If $\Gamma = \phi_1, \dots, \phi_n$, then, as usual, $\Theta\Gamma = \Theta\phi_1, \dots, \Theta\phi_n$, where $\Theta \in \{K_i, C\}$. The letter S (possibly subscripted) is used in the paper to denote sequents. Any sequent

$$\phi_1, \dots, \phi_m \Rightarrow \psi_1, \dots, \psi_n$$

is understood informally as the formula

$$(\phi_1 \wedge \dots \wedge \phi_m) \to (\psi_1 \vee \dots \vee \psi_n).$$

The sequent $|S|$ is obtained from S by contracting equal members in the antecedent and succedent, e.g., if $S = (p, p \Rightarrow p, q, q)$, then $|S| = (p \Rightarrow p, q)$.

Given a countable set of atomic propositions Φ, a Kripke interpretation M is a tuple $\langle \mathcal{S}, R_1, \dots, R_\mu, \pi \rangle$, where:

(i) $\mathcal{S} \neq \emptyset$ is a set of states,

(ii) $R_i \subseteq \mathcal{S} \times \mathcal{S}$ $(1 \leq i \leq \mu)$,

(iii) $\pi : \mathcal{S} \to 2^{\Phi}$.

We say that $t \in \mathcal{S}$ is reachable from $s \in \mathcal{S}$, iff there is a finite chain $s_1, \dots, s_n$ $(n \geq 2)$ such that (1) $s_1 = s$, (2) $s_n = t$, and (3) $(s_i, s_{i+1}) \in R_{j_i}$ for each $1 \leq i \leq (n-1)$, where $j_i \in \{1, \dots, \mu\}$. If $n = 2$, then t is also called K-reachable from s. We use the expression $s \mapsto s'$ to denote that s' is K-reachable from s.

The binary relation $\models$ between pairs (M, s) and **LCK** formulas, where $s \in \mathcal{S}$, is defined as follows:

$(M, s) \models p$, iff $p \in \pi(s)$,

$(M, s) \models \neg\phi$, iff $(M, s) \not\models \phi$,

$(M, s) \models \phi \vee \psi$, iff $(M, s) \models \phi$ or $(M, s) \models \psi$,

$(M, s) \models \phi \wedge \psi$, iff $(M, s) \models \phi$ and $(M, s) \models \psi$,

$(M, s) \models \phi \to \psi$, iff $(M, s) \not\models \phi$ or $(M, s) \models \psi$,

$(M, s) \models K_i\phi$, iff $(M, t) \models \phi$ for all $t \in \mathcal{S}$ such that $(s, t) \in R_i$,

$(M, s) \models C\phi$, iff $(M, t) \models \phi$ for all $t \in \mathcal{S}$ reachable from s.

We extend the relation $\models$ for sequents: $(M, s) \models \phi_1, \dots, \phi_m \Rightarrow \psi_1, \dots, \psi_n$, iff there is $i \in \{1, \dots, m\}$ such that $(M, s) \not\models \phi_i$ or there is $j \in \{1, \dots, n\}$ such that $(M, s) \models \psi_j$. It is true that $(M, s) \models \phi_1, \dots, \phi_n \Rightarrow \psi_1, \dots, \psi_m$ implies $(M, s) \models (\phi_1 \wedge \dots \wedge \phi_n) \to (\psi_1 \vee \dots \vee \psi_m)$, and vice versa. In particular, $(M, s) \models \Rightarrow \psi$, iff $(M, s) \models \psi$.

If $(M,\ s) \models \nabla$ for all possible pairs $(M,\ s)$, where ∇ is a formula or a sequent, then we write $\models \nabla$ and call ∇ valid (in **LCK**). It is true that, e.g., the formula $C\phi \rightarrow K_i\phi$ is valid.

Proposition 1 $(M,\ s) \models S$, iff $(M,\ s) \models |S|$.

Proof. The proof of the proposition is obvious. □

3. Deductive system

We use the letter ε to denote the empty string, where $\varepsilon\chi = \chi\varepsilon = \chi$ for any expressions χ; also, $(AB_iC)_{1\leq i\leq m} = AB_1C,\ \ldots,\ AB_mC$, where A, B, and C are any, possibly ε, expressions.

The sequent calculus **GLCK** consists of an axiom schema and propositional and modal rules:

1. Axiom schema: $\Gamma,\ \phi \Rightarrow \phi,\ \Delta$.
2. Propositional rules:

$$\frac{|\Gamma \Rightarrow \phi, \Delta||\psi, \Gamma \Rightarrow \Delta|}{\phi \rightarrow \psi, \Gamma \Rightarrow \Delta}(\rightarrow\Rightarrow), \quad \frac{|\Gamma, \phi \Rightarrow \psi, \Delta|}{\Gamma \Rightarrow \phi \rightarrow \psi, \Delta}(\Rightarrow\rightarrow),$$

$$\frac{|\phi, \psi, \Gamma \Rightarrow \Delta|}{\phi \wedge \psi, \Gamma \Rightarrow \Delta}(\wedge \Rightarrow), \quad \frac{|\Gamma \Rightarrow \phi, \Delta||\Gamma \Rightarrow \psi, \Delta|}{\Gamma \Rightarrow \phi \wedge \psi, \Delta}(\Rightarrow \wedge),$$

$$\frac{|\phi, \Gamma \Rightarrow \Delta||\psi, \Gamma \Rightarrow \Delta|}{\phi \vee \psi, \Gamma \Rightarrow \Delta}(\vee \Rightarrow), \quad \frac{|\Gamma \Rightarrow \phi, \psi, \Delta|}{\Gamma \Rightarrow \phi \vee \psi, \Delta}(\Rightarrow \vee),$$

$$\frac{|\Gamma \Rightarrow \phi, \Delta|}{\neg\phi, \Gamma \Rightarrow \Delta}(\neg \Rightarrow), \quad \frac{|\Gamma, \phi \Rightarrow \Delta|}{\Gamma \Rightarrow \neg\phi, \Delta}(\Rightarrow \neg).$$

Modal rules:

$$\frac{|\Gamma \Rightarrow \phi|}{\Sigma_1,\ \Omega^{K_i},\ K_i\Gamma \Rightarrow K_i\phi,\ \Delta^K,\ \Sigma_2}\ (K_i),$$

$$\frac{|(K_i\phi)_{1\leq i\leq\mu},\ (K_iC\phi)_{1\leq i\leq\mu},\ \Gamma \Rightarrow \Delta|}{C\phi,\ \Gamma \Rightarrow \Delta}\ (C \Rightarrow),$$

$$\frac{(|\Gamma \Rightarrow \Delta,\ K_i\phi|)_{1\leq i\leq\mu}(|\Gamma \Rightarrow \Delta,\ K_iC\phi|)_{1\leq i\leq\mu}}{\Gamma \Rightarrow \Delta,\ C\phi}\ (\Rightarrow C).$$

Here: $\Gamma,\ \Delta,\ \Pi$ denote finite, possibly empty, multisets of formulas; in the rule (K_i): (1) Σ_1 and Σ_2 denote finite, possibly empty, multisets of atomic formulas; (2) each member of Ω^{K_i} is of the type $K_j\phi$, where $j \in \{1,\ \ldots,\ \mu\} \setminus \{i\}$; (3) each member of Δ^K is of the type $K_j\phi$, where $j \in \{1,\ \ldots,\ \mu\}$. It is required that the conclusion of (K_i) is not an axiom. Premises of the shape $|\Gamma \Rightarrow \Delta,\ K_i\phi|$, $1 \leq i \leq \mu$, are called k-premises of $(\Rightarrow C)$. Premises of the shape $|\Gamma \Rightarrow \Delta,\ K_iC\phi|$, $1 \leq i \leq \mu$, are called kc-premises of $(\Rightarrow C)$.

Remark 1 We prove in Section 6 that all derivation rules of **GLCK**, except (K_i), are invertible. The restriction that $\Sigma_1,\ \Sigma_2$ consist of atomic formulas implies the following strategy of backward proof-search: apply invertible rules (i.e., rules other than (K_i)) as long as possible. That allows us to reduce backtracking. If we drop the restriction that $\Sigma_1,\ \Sigma_2$ consist of atomic formulas, then the following backward application of (K_i) is possible:

$$\frac{p \Rightarrow q}{p \wedge q,\ K_i p \Rightarrow K_i q,\ p}\ (K_i),$$

where the premise is not provable, though the conclusion can be proved by applying backwards rule $(\wedge \Rightarrow)$ to it.

Principal formulas of the derivation rules are defined in the usual way, e.g., $C\phi$ is the principal formula of rules $(C \Rightarrow)$ and $(\Rightarrow C)$. We use the expression $(\Rightarrow C\phi)$ to specify that $C\phi$ is the principal formula of $(\Rightarrow C)$.

4. Derivation loops and provable sequents

Backward **GLCK** proof-search trees are generated as usual by subsequently applying backward derivation rules to sequents.

We use the letter V to denote a backward proof-search tree, and the expression $V(S)$ to specify that S is the root of V.

$\sigma[\phi]$ is the sequence of symbols in ϕ, except parentheses, e.g.,

$$\sigma[(p \vee p) \rightarrow K_i q] = p,\ \vee,\ p,\ \rightarrow,\ K_i,\ q.$$

Definition 1 The length of a formula ϕ ($\lambda(\phi)$ in notation) is the number of elements in $\sigma[\phi]$ plus the number of 'C' in $\sigma[\phi]$. For example, $\lambda(p \wedge p) = 3$, $\lambda(K_i p) = 2$, and $\lambda(Cp) = 2 + 1 = 3$. The length of a sequent S ($\lambda(S)$ in notation) is the sum of lengths of all the members of antecedent and succedent of S.

Remark 2 We need to have $\lambda(C\phi) > \lambda(K_i\phi)$ in, e.g., the proof of Proposition 2, concerning k-premises of $(\Rightarrow C)$. Therefore, the definition of formula length is somewhat non-standard.

Definition 2 Given a proof-search tree, an upward path in the tree from some sequent S to S' inclusive is called a derivation loop ($[S \circ S']$ in notation) iff 1) the length of the path is greater than 0 and 2) S' and S coincide. The sequents S and S' are called the base and terminal sequents of $[S \circ S']$, respectively.

Example 1 Let consider the following backward proof-search tree, assuming that $\mu = 1$ for simplicity (read the tree from bottom):

$$\dfrac{\dfrac{p,\ K_1 p,\ K_1 Cp \Rightarrow K_1 q \qquad \dfrac{0':\ p,\ Cp \Rightarrow Cq}{p,\ K_1 p,\ K_1 Cp \Rightarrow K_1 Cq}\ (K_1)}{p,\ K_1 p,\ K_1 Cp \Rightarrow Cq}}{0:\ p,\ Cp \Rightarrow Cq}\ (C \Rightarrow)$$

Here '0:' and '$0'$:' are used as sequent labels. One can see that $[0 \circ 0']$ is a derivation loop, consisting of the sequence of sequents

$$p,\ Cp \Rightarrow Cq \quad p,\ K_1 p,\ K_1 Cp \Rightarrow Cq \quad p,\ K_1 p,\ K_1 Cp \Rightarrow K_1 Cq \quad p,\ Cp \Rightarrow Cq.$$

Formulas of the shape $C\phi$ are called universality formulas.

Definition 3 A derivation loop $[S \circ S']$ is called a derivation loop with the universality formula $C\phi$, iff: (1) $S = (\Gamma \Rightarrow \eta C\phi,\ \Delta)$, where $\eta \in \{\varepsilon,\ K_i\}$, (2) succedent of premise of any (K_i) application in $[S \circ S']$ consists of $C\phi$, and (3) $[S \circ S']$ contains a kc-premise and does not contain k-premises of $(\Rightarrow C\phi)$.

If there is no formula $C\phi$ in a derivation loop satisfying (1), (2), and (3), then the derivation loop is called α-void.

In Example 1, $[0 \circ 0']$ is a derivation loop with the universality formula Cq.

Proposition 2 Any derivation loop $[S \circ S']$ has at least one application of $(C \Rightarrow)$ or kc-premise of an application of $(\Rightarrow C)$.

Proof. The proof follows from the facts that $\lambda(S) = \lambda(S')$ and the lengths of premises are less than the lengths of corresponding conclusions of rules other than $(C \Rightarrow)$ and $(\Rightarrow C)$; in the case of $(\Rightarrow C)$, the lengths of k-premises are less than the length of conclusion. □

Proposition 3 Any derivation loop $[S \circ S']$ has at least one application of (K_i).

Proof. Let $n(C\phi)$ be the number of occurrences of the formula $C\phi$ that are not within the scope of 'K_i' in a sequent. It is true that $n(C\phi)$ for S' is not less than for S with respect to any formula $C\phi$.

Let us call an application of $(\Rightarrow C)$ in $[S \circ S']$ essential, if a kc-premise of that application is in $[S \circ S']$. It follows from Proposition 2 that $[S \circ S']$ has an application of $(C \Rightarrow)$ or (and) an essential application of $(\Rightarrow C)$. Let $\{C\phi_1, \ldots, C\phi_m\}$ be the set of all principal formulas of such applications. There is $\iota \in \{1, \ldots, m\}$ such that $\lambda(C\phi_\iota) \geq \lambda(C\phi_i)$, for $1 \leq i \leq n$. It is true that $n(C\phi_\iota)$ for the conclusion is strictly greater than for a premise of $(C \Rightarrow)$ or any kc-premise of $(\Rightarrow C)$ if the principal formula is $C\phi_\iota$. One can see that only (K_i) can increase $n(C\phi_\iota)$. Hence if (K_i) was not applied in $[S \circ S']$, then $n(C\phi_\iota)$ for S is strictly greater than for S'. □

Proposition 4 In any derivation loop with a universality formula $C\phi$, there is an application of $(\Rightarrow C\phi)$ between any two applications of (K_i).

Proof. The proof follows from Definition 3 and from the shape of (K_i). □

Definition 4 A maximal connected graph Υ in V all edges of which belong to derivation loops with universality formulas, is called a connected component. Υ is said to have a universality formula, iff all derivation loops in it have a common universality formula. Such a formula is called the universality formula of Υ. If Υ does not have a universality formula, then each derivation loop in it is called β-void.

Definition 5 A derivation loop is called void, iff it is α-void or β-void.

Sequents of the shape $\Sigma_1, \Gamma^K \Rightarrow \Sigma_2$, where Σ_i, Σ_2 consist of atomic formulas and Γ^K consists of formulas of the shape $K_i\Gamma$, are called *atomic-like*. No derivation rule is backwards applicable to atomic-like sequents.

Definition 6 A leaf L of a backward proof-search tree is called strongly-closing, iff it is an axiom or a terminal sequent of a derivation loop with a universality formula; L is called weakly-closing, iff it is a non-axiom atomic-like sequent or a terminal sequent of a void derivation loop; L is called closing, iff it is weakly-closing or strongly-closing.

It is assumed in the sequel that derivation rules are not applied backward to closing leaves of any V.

Proposition 5 If S is a terminal sequent of a derivation loop, then S is closing.

Proof. Any derivation loop is a derivation loop with a universality formula or, otherwise, an α- or β-void derivation loop. It is true that S is closing in all these cases. □

Lemma 1 Any backward proof-search tree $V(|S|)$ is finite.

Proof. Each sequent in $V(|S|)$ consists of the subformulas of formulas of the finite sequent $|S|$, including $K_i\phi$ and $K_iC\phi$ $(1 \leq i \leq \mu)$ if $C\phi$ occurs in $|S|$. Some of these formulas occur in the antecedent and the others in succedent of sequents. Let us denote the sets of such formulas $\mathbb{A}$ and $\mathbb{S}$, respectively. For example, if $S = (Cp \Rightarrow K_1q)$, then

$$\mathbb{A} = \{p, K_1p, \ldots, K_\mu p, K_1Cp, \ldots, K_\mu Cp, Cp\} \text{ and } \mathbb{S} = \{q, K_1q\}.$$

The antecedents and succedents of sequents in $V(|S|)$ are the subsets of $\mathbb{A}$ and $\mathbb{S}$, respectively. Hence we can have no more than $2^{n(\mathbb{A})}$ different antecedents and $2^{n(\mathbb{S})}$ different succedents, where $n(\mathbb{A})$ and $n(\mathbb{S})$ denote the numbers of elements of $\mathbb{A}$ and $\mathbb{S}$, correspondingly. It follows that the number of different sequents in $V(|S|)$ is not greater than $2^{n(\mathbb{A})} \times 2^{n(\mathbb{S})} = 2^{n(\mathbb{A})+n(\mathbb{S})}$. We get that there are at least two equal sequents on any branch l of $V(|S|)$, where the length of l is at least $2^{n(\mathbb{A})+n(\mathbb{S})}$. Hence the length of l cannot be greater than $2^{n(\mathbb{A})+n(\mathbb{S})}$ because, if not closed earlier, a terminal sequent of a derivation loop will close l in no more than $2^{n(\mathbb{A})+n(\mathbb{S})}$ rule applications counting from the root. □

Definition 7 A sequent S is called provable in **GLCK** ($\vdash S$ in notation), iff there exists a backward proof-search tree V with $|S|$ at the root such that (1) each leaf of V is an axiom or a terminal sequent of a derivation loop with a universality formula and (2) each connected component in V has a universality formula. V is called a proof of $|S|$ or a proof tree. If all leaves of V are axioms, then S is called axiomatically provable in **GLCK**. The notation $\vdash^{V} |S|$ means that V is a proof of $|S|$.

A formula ϕ is provable, iff the sequent $\vdash \Rightarrow \phi$ is provable.

Example 2 The following formulas are provable in **GLCK**:

F1. $K_i(p \to q) \to (K_i p \to K_i q)$

F2. $C(p \to q) \to (Cp \to Cq)$

F1 is axiomatically proved in **GLCK** as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{q,\ p \Rightarrow q \quad p \Rightarrow p,\ q}{(p \to q),\ p \Rightarrow q}(\to\Rightarrow)}{K_i(p \to q),\ K_i p \Rightarrow K_i q}(K_i)}{K_i(p \to q) \Rightarrow K_i p \to K_i q}(\Rightarrow\to)}{\Rightarrow K_i(p \to q) \to (K_i p \to K_i q)}(\Rightarrow\to)$$

The formula F2 is proved in **GLCK** as follows (we assume that $\mu = 1$ for simplicity):

$$\dfrac{\dfrac{S \qquad \dfrac{\dfrac{C(p \to q),\ p,\ Cp \Rightarrow Cq,\ p \quad 0' :\ q,\ C(p \to q),\ p,\ Cp \Rightarrow Cq}{p \to q,\ C(p \to q),\ p,\ Cp \Rightarrow Cq}(\to\Rightarrow)}{K_1(p \to q),\ K_1C(p \to q),\ K_1 p,\ K_1 Cp \Rightarrow K_1 Cq}(K_1)}{K_1(p \to q),\ K_1C(p \to q),\ K_1 p,\ K_1 Cp \Rightarrow Cq}(\Rightarrow C)}{\dfrac{\dfrac{\dfrac{K_1(p \to q),\ K_1C(p \to q),\ Cp \Rightarrow Cq}{0 :\ C(p \to q),\ Cp \Rightarrow Cq}(C\Rightarrow)}{C(p \to q) \Rightarrow Cp \to Cq}(\Rightarrow\to)}{\Rightarrow C(p \to q) \to (Cp \to Cq)}(\Rightarrow\to)}(C\Rightarrow)$$

Here $S = \big(K_1(p \to q),\ K_1C(p \to q),\ K_1 p,\ K_1 Cp \Rightarrow K_1 q\big)$ is axiomatically provable, applying backward (K_1) and $(\to\Rightarrow)$; $[0 \circ 0']$ is a derivation loop with the universality formula Cq.

5. Soundness of GLCK

Lemma 2 Let

$$\frac{|S_1| \quad (|S_2|,\ \ldots,\ |S_n|)}{S}\ (r)$$

$(n \in \{2,\ 2\mu\})$ be an application of any **GLCK** rule, except (K_i). If $(M,\ s) \models |S_1|$ (and $(M,\ s) \models |S_i|$, $2 \le i \le n$, respectively), then $(M,\ s) \models |S|$.

Proof. The lemma is proved by considering cases of (r):

$(r) = (C \Rightarrow)$:

$$\frac{|S_1| :\ |(K_i\phi)_{1 \le i \le \mu},\ (K_iC\phi)_{1 \le i \le \mu},\ \Gamma \Rightarrow \Delta|}{C\phi,\ \Gamma \Rightarrow \Delta}(C\Rightarrow).$$

We have that $(M,\ s) \models |S_1|$. If there is $\phi' \in \Gamma$ or $\psi' \in \Delta$ such that $(M,\ s) \not\models \phi'$ or $(M,\ s) \models \psi'$, respectively, then $(M,\ s) \models S$. Otherwise, $(M,\ s) \not\models K_i\phi$ or $(M,\ s) \not\models K_iC\phi$ $(i \in \{1,\ \ldots,\ \mu\})$, i.e., there is t reachable from s such that $(M,\ t) \not\models \phi$. Hence $(M,\ s) \not\models C\phi$. We get $(M,\ s) \models S$, according to the definition of $\models$.

$(r) = (\rightarrow\Rightarrow)$:

$$\frac{|S_1|:\ |\Gamma \Rightarrow \phi,\ \Delta| \quad |S_2|:\ |\psi,\ \Gamma \Rightarrow \Delta|}{S:\ \phi \rightarrow \psi,\ \Gamma \Rightarrow \Delta}\ (\rightarrow\Rightarrow).$$

If there is $\phi' \in \Gamma$ or $\psi' \in \Delta$ such that $(M,\ s) \not\models \phi'$ or $(M,\ s) \models \psi'$, respectively, then $(M,\ s) \models S$. Otherwise, $(M,\ s) \models |S_i|$ $(1 \leq i \leq 2)$ imply $(M,\ s) \models \phi$ and $(M,\ s) \not\models \psi$. Hence $(M,\ s) \not\models \phi \rightarrow \psi$. This fact yields $(M,\ s) \models S$.

The remaining cases when the principal formula of (r) is in antecedent, i.e., $(r) \in \{(\wedge \Rightarrow),\ (\vee \Rightarrow),\ (\neg \Rightarrow)\}$, are considered in the same way as the previous case.

$(r) = (\Rightarrow C)$:

$$\frac{(|S_i|:\ |\Gamma \Rightarrow \Delta,\ K_i\phi|)_{1 \leq i \leq \mu} \quad (|S_{\mu+i}|:\ |\Gamma \Rightarrow \Delta,\ K_iC\phi|)_{1 \leq i \leq \mu}}{\Gamma \Rightarrow \Delta,\ C\phi}\ (\Rightarrow C).$$

If there is $\phi' \in \Gamma$ or $\psi' \in \Delta$ such that $(M,\ s) \not\models \phi'$ or $(M,\ s) \models \psi'$, respectively, then $(M,\ s) \models S$. Otherwise, $(M,\ s) \models |S_i|$ $(1 \leq i \leq 2\mu)$ imply $(M,\ s) \models K_i\phi$ and $(M,\ s) \models K_iC\phi$ $(1 \leq i \leq \mu)$. We have that $(M,\ t) \models \phi$ for each t that is reachable from s, i.e., $(M,\ s) \models C\phi$. Hence $(M,\ s) \models S$.

$(r) = (\Rightarrow\rightarrow)$:

$$\frac{|S_1|:\ |\Gamma,\ \phi \Rightarrow \psi,\ \Delta|}{S:\ \Gamma \Rightarrow \phi \rightarrow \psi,\ \Delta}\ (\Rightarrow\rightarrow).$$

If there is $\phi' \in \Gamma$ or $\psi' \in \Delta$ such that $(M,\ s) \not\models \phi'$ or $(M,\ s) \models \psi'$, respectively, then $(M,\ s) \models S$. Otherwise, $(M,\ s) \models |S_1|$ implies $(M,\ s) \not\models \phi$ or $(M,\ s) \models \psi$, i.e., $(M,\ s) \models \phi \rightarrow \psi$. Hence $(M,\ s) \models S$.

The remaining cases when the principal formula of (r) is in succedent, i.e., $(r) \in \{(\Rightarrow \wedge),\ (\Rightarrow \vee),\ (\Rightarrow \neg)\}$, are considered in the same way as the previous case. $\square$

Corollary 1 If $(M,\ s) \not\models S$, then there is $j \in \{1,\ \ldots,\ n\}$ such that $(M,\ s) \not\models |S_j|$.

Lemma 3 Let

$$\frac{|S_1|:\ |\Gamma \Rightarrow \phi|}{S:\ \Sigma_1,\ \Omega^{K_i},\ K_i\Gamma \Rightarrow K_i\phi,\ \Delta^K,\ \Sigma_2}\ (K_i),$$

be an application of rule (K_i), $i \in \{1,\ \ldots,\ \mu\}$. If $(M,\ t) \models |S_1|$ for each t such that $(s,\ t) \in R_i$, then $(M,\ s) \models S$.

Proof. If there is $\psi \in \Gamma$ such that $(M,\ t) \not\models \psi$, where $(s,\ t) \in R_i$, then $(M,\ s) \not\models K_i\psi$. Hence $(M,\ s) \models S$. If there is no such formula ψ in Γ, then the condition of lemma implies $(M,\ t) \models \phi$ for all t such that $(s,\ t) \in R_i$. Hence $(M,\ s) \models K_i\phi$. It follows that $(M,\ s) \models S$. $\square$

Corollary 2 If $(M,\ s) \not\models S$, then there is t such that $(s,\ t) \in R_i$ and $(M,\ t) \not\models |S_1|$.

Let $\Gamma = \phi_1,\ \ldots,\ \phi_n$. If $n > 1$, then $\theta\Gamma = \phi_1\theta \ldots \theta\phi_n$, where $\theta \in \{\wedge,\ \vee\}$. If $n = 1$ or no formula occur in Γ, then $\theta\Gamma = \Gamma$.

Lemma 4 Let

$$\frac{|S_1| : |\Gamma \Rightarrow C\phi|}{S : \Sigma_1,\ \Omega^{K_i},\ K_i\Gamma \Rightarrow K_iC\phi,\ \Delta^K,\ \Sigma_2}\ (K_i)$$

be an application of rule (K_i). If $(M,\ s) \not\models S$ and $(M,\ s) \models K_i\phi$, then there is $t \neq s$ such that $(s,\ t) \in R_i$ and $(M,\ t) \not\models |S_1|$.

Proof. $(M,\ s) \models K_i\phi$ (a), according to the condition of lemma. $(M,\ s) \not\models S$ implies $(M,\ s) \models \wedge K_i\Gamma$ (b) and $(M,\ s) \not\models K_iC\phi$ (c). There is $t \neq s$ such that $(s,\ t) \in R_i$ and $(M,\ t) \not\models C\phi$, based on (a) and (c). It is true that $(M,\ t) \models \wedge\Gamma$, according to $(s,\ t) \in R_i$ and (b). Hence $(M,\ t) \not\models |S_1|$. □

In the proofs of Theorems 1 and 2, we use the method of strong mathematical induction that is described as follows: An induction parameter ip for a considered proposition (lemma, theorem) P is fixed. Base case: it is proved that P holds for the minimal value of ip, e.g., $ip = 0$. Step case: it is assumed that P holds for all values of ip that are less than some fixed number $n > 0$. Based on this assumption, it is proved that P holds also for n. That implies that P holds for all values of ip.

Theorem 1 The calculus **GLCK** is sound: if $\vdash S$, then $\models S$, where S is an arbitrary sequent.

Proof. The theorem is proved by induction on the number d of the leaves of $V(|S|)$ that are terminal sequents of derivation loops.

Assume that there is a sequent S such that $\vdash^V |S|$ and $\not\models S$. Hence $\not\models |S|$, according to Proposition 1. We have that there is $(M,\ s)$ such that $(M,\ s) \not\models |S|$.

It follows from Corollaries 1, 2 that V has a branch l such that for each sequent $\hat{S}$ on the branch it is true that

$$(M,\ t_i) \not\models \hat{S}_i. \tag{1}$$

Let S_1' be the topmost sequent on l.

If $d = 0$, then S_i' is a logical axiom. Hence $(M,\ t) \models S'$ for any pair $(M,\ t)$, which contradicts (1) and the initial assumption that $\vdash^V |S|$ and $\not\models S$.

Let $d > 0$ and S_1' be a terminal sequent of a derivation loop $[S_1 \circ S_1']$. It is true that $l = \pi\bar{\pi}$, where π is a possibly empty path, and $\bar{\pi} = [S_1 \circ S_1']$. Let the formula $C\phi$ be a universality formula of the connected component Υ to which $[S_1 \circ S_1']$ belongs, and

$$\frac{(|\Gamma \Rightarrow K_i\phi,\ \Delta|)_{1\leq i\leq \mu} \quad (|\Gamma \Rightarrow K_iC\phi,\ \Delta|)_{1\leq i\leq \mu}}{|\Gamma \Rightarrow C\phi,\ \Delta|}\ (\Rightarrow C) \tag{2}$$

be the first from the bottom application of $(\Rightarrow C\phi)$ in $[S_1 \circ S_1']$. We have:

$$(M,\ s_0) \not\models |\Gamma \Rightarrow C\phi,\ \Delta| \tag{3}$$

and

$$(M,\ s_0) \models |\Gamma \Rightarrow K_i\phi,\ \Delta| \quad (1 \leq i \leq \mu) \tag{4}$$

for some pair $(M,\ s_0)$. (4) holds according to the inductive hypothesis because any derivation loop lp that contains the considered occurrence of $|\Gamma \Rightarrow C\phi,\ \Delta|$, is in Υ and must have the universality formula $C\phi$. Hence the path corresponding to lp cannot go via a k-premise of $(\Rightarrow C\phi)$. It follows that the maximal sub-tree of V with any such a k-premise at the root is a proof tree with the inductive parameter $d' < d$.

It follows from (3) that: (a) $(M,\ s_0) \models \wedge\Gamma$ and (b) $(M,\ s_0) \not\models \vee C\phi,\ \Delta$; the latter fact yields $(M,\ s_0) \not\models C\phi$, i. e., there is s_n reachable from s_0 such that

$$(M,\ s_n) \not\models \phi. \tag{5}$$

We call such states s_n critical. It is true that

$$(M,\ s_0) \models K_i\phi,\ 1 \leq i \leq \mu, \tag{6}$$

based on (4), (a), and (b). Hence s_0 is not critical.

It follows from (3) and 6 that there is $j \in \{1,\ \ldots,\ \mu\}$ in (2) such that

$$(M,\ s_0) \not\models |S_j| = |\Gamma \Rightarrow K_jC\phi,\ \Delta|.$$

Let us consider the following fragment of V, including the semantical analysis:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\big((M,s_1) \models |\Gamma_2 \Rightarrow K_i\phi,\Delta_2|\big)_{1\leq i\leq\mu} \qquad \overset{\ldots}{\big(|\Gamma_2 \Rightarrow K_iC\phi,\Delta_2|\big)_{1\leq i\leq\mu}}}{(M,s_1) \not\models |\Gamma_2 \Rightarrow C\phi,\Delta_2|}\,(\Rightarrow C)}{\pi_1}}{(M,s_1) \not\models |\Gamma_1 \Rightarrow C\phi|}}{(M,s_0) \not\models |S_j'|}\,(K_i)}{\pi}$$

$$\dfrac{\ldots \qquad (M,s_0) \not\models |S_j|}{(M,s_0) \not\models |\Gamma \Rightarrow C\phi,\Delta|}\,(\Rightarrow C)$$

Here:

1. π is a path that does not have applications of (K_i);

2. $(M,\ s_0) \not\models |S_j'|$, based on Corollary 1;

3. $(M,\ s_1) \not\models |\Gamma_1 \Rightarrow C\phi|$, where $s_0 \mapsto s_1$, based on Corollary 2. We choose $s_1 \neq s_0$, such a choice is legitimate, based on Lemma 4;

4. π_1 is a path that has no applications of $(\Rightarrow C\phi)$. It follows that there are no applications of (K_i) in π_1, according to Proposition 4. Hence $(M,\ s_1) \not\models |\Gamma_2 \Rightarrow C\phi,\ \Delta_2|$, according to item 3 and Corollary 1;

5. $(M,\ s_1) \models |\Gamma_2 \Rightarrow K_i\phi,\ \Delta_2|$ $(1 \leq i \leq \mu)$ is obtained in the same way as (4).

6. Hence $(M,\ s_1) \not\models C\phi$ and $(M,\ s_1) \models K_i\phi$ $(1 \leq i \leq \mu)$, based on items 4 and 5.

We have $(M,\ s_0) \models K_i\phi \wedge \neg C\phi$ and $(M,\ s_1) \models K_i\phi \wedge \neg C\phi$, where $s_0 \mapsto s_1$ and $1 \leq i \leq \mu$. The process of generation of such states is continued above $(M,\ s_1) \not\models |\Gamma_2 \Rightarrow C\phi,\ \Delta_2|$, making use also of Proposition 3. Each leaf L' of V reached during the process is a terminal sequent of a derivation loop $[L \circ L']$ with the universality formula $C\phi$. It is true that $L = L'$. When L' is reached, we go down to L and continue the process, traversing through derivation loops in Υ and obtaining

$$(M,\ s_0) \models K_i\phi \wedge \neg C\phi,\ (M,\ s_1) \models K_i\phi \wedge \neg C\phi,\ \ldots,\ (M,\ s_k) \models K_i\phi \wedge \neg C\phi$$

$(1 \leq i \leq \mu)$ for any $k \in N$, where $s_0 \mapsto s_1 \mapsto \ldots \mapsto s_k$ and the states s_i in the sequence are unequal in pairs. $(M,\ s_k) \models K_i\phi$ implies $(M,\ t) \models \phi$ for each t such that $(s_k,\ t) \in R_i$ $(1 \leq i \leq \mu)$. It is clear that we will eventually get k such that $t \models \phi$, $(s_k,\ t) \in R_i$ $(i \in \{1,\ \ldots,\ \mu\})$, and t is a critical state. That contradicts the existence of critical states. Hence we get a contradiction to the initial assumption that $\vdash S$ and $\not\models S$. □

6. Completeness of GLCK

Definition 8 A non-proof tree V is called a complete-refutation tree, iff all its leaves are closing sequents.

The K_i-resolvent set of S, where $1 \leq i \leq \mu$, is defined as follows:

$$\left\{S'\ :\ S' \text{ is a premise of backward application of } (K_i)\ (1 \leq i \leq \mu) \text{ to } \ S\right\}.$$

We introduce the following rule:

$$\frac{\mathbb{S}^{K_i}}{S}\ (STEP),$$

where S is not an axiom, and $\mathbb{S}^{K_i}$ is K_i-resolvent set of S. The calculus **GLCK*** is obtained from **GLCK** by replacing (K_i) with rule $(STEP)$. Each **GLCK*** tree V can be transformed into a **GLCK** tree by leaving only a premise of an application of (K_i) in each application of $(STEP)$. We call such trees **GLCK**-*sub-trees of* V. In the proof of Theorem 2, we do not consider connected components that are not contained in an **GLCK**-sub-tree of the corresponding **GLCK*** tree.

A **GLCK*** backward proof-search tree $V(|S|)$ is called *fully-extended* if the branches of its **GLCK**-sub-trees $V_{sub}(|S|)$ cannot be extended by backward rule applications.

A **GLCK*** backward proof-search tree $V(|S|)$ is called *a proof tree* if it has a **GLCK**-sub-tree $V_{sub}(|S|)$ that is a proof-tree.

Definition 9 A **GLCK*** backward proof-search tree is called $(STEP)$-free, iff there is no application of $(STEP)$ in it.

Theorem 2 The calculus **GLCK** is complete: if $\models S$, then $\vdash S$, where S is an arbitrary sequent.

Proof. We prove that $\not\vdash S$ implies $\not\models S$. The proof uses a variant of Schütte's method of reduction trees [19]. Assume that $\not\vdash S$. A fully-extended **GLCK*** backward proof-search tree $V^*(|S|)$ is obtained as follows. First, a fully-extended $(STEP)$-free tree is generated. If the obtained tree is fully-extended, then the construction of $V^*(|S|)$ is stopped. Otherwise, the construction of $V^*(|S|)$ is continued by: (1) applying backward $(STEP)$ to each non-closing leaf, (2) generating fully-extended $(STEP)$-free trees for the obtained premises of $(STEP)$, and (3) applying backward $(STEP)$ again to the non-closing leaves of $V^*(|S|)$. The procedure is continued until we obtain a fully-extended tree $V^*(|S|)$. This tree consists of **GLCK**-sub-trees $V_{sub}(|S|)$. Each such $V_{sub}(|S|)$ is finite, according to Lemma 1. Hence $V^*(|S|)$ is finite. No $V_{sub}(|S|)$ is a proof tree because $\not\vdash S$.

In each $V_{sub}(|S|)$, we mark a leaf that is a terminal sequent of an α-void derivation loop or a non-axiom atomic-like sequent; if $V_{sub}(|S|)$ does not have such leaves, then we mark every leaf that is a terminal sequent of a β-void derivation loop. The tree $\tilde{V}(|S|)$ is obtained from $V^*(|S|)$ by dropping each sequent that is not on the path from $|S|$ to a marked sequent.

Making use of $\tilde{V}(|S|)$, the interpretation $M = \langle \mathcal{S},\ R_1,\ \ldots,\ R_\mu,\ \pi\rangle$ is obtained as follows:

1. If there is no application of $(STEP)$ in $\tilde{V}(|S|)$, then $\tilde{V}(|S|)$ has only one leaf $\Sigma_1,\ \Gamma^K \Rightarrow \Sigma_2$. We choose $\mathcal{S} = \{s_0\}$, $R_1 = \ldots = R_\mu = \emptyset$, and $\pi(s_0) = \{\Sigma_1\}$. The path from S to $\Sigma_1,\ \Gamma^K \Rightarrow \Sigma_2$ is denoted by $p(s_0)$.

2. Otherwise, we include a unique state s in $\mathcal{S}$ for each path $p(s)$ from S_{bp} to S_{ec} of the shape

$$
\begin{array}{ccc}
S_{ec} : \Sigma_1,\ \Gamma^K \Rightarrow \Sigma_2 & & \dfrac{K_i}{S_{ec} : \Sigma_1,\ \Gamma \Rightarrow \Delta,\ \Sigma_2} \\
\vdots & \text{and} & \vdots \\
S_{bp} & & S_{bp}
\end{array}
$$

in $\tilde{V}(|S|)$, where (1) Γ and Δ do not have atomic members and (2) $S_{bp} = S$ or S_{bp} is a premise of an application of $(STEP)$, and $(STEP)$ is not applied between S_{bp} and S_{ec}. A path from a premise S_{bp} of $(STEP)$ to a terminal sequent $S_t \neq S_{ec}$ of a derivation loop dl is reckoned to be contained in $p(s)$, where the base sequent of dl belongs to $p(s)$ and $(STEP)$ is not applied between S_{bp} and S_t. That is to say, if the considered part of the branch is

$$
p(s)p(s_1)\cdots p(s_n)S_{bp}\cdots S_t,
$$

then the path $S_{bp}\cdots S_t$ is reckoned to be contained in $p(s)$, where there is no (K_i) between S_{bp} and S_t.

We set $\pi(s) = \Sigma_1$. Two states s_1 and s_2 in $\mathcal{S}$ are called connected, iff the corresponding paths, denoted by $p(s_1)$ and $p(s_2)$, are adjacent and on the same branch of $\tilde{V}(|S|)$.

For each connected states $s_1,\ s_2$, we add $(s_1,\ s_2)$ in R_i, iff $p(s_2)$ is above $p(s_1)$, and $p(s_2)$ starts with a premise of (K_i) $(i \in \{1,\ \ldots,\ \mu\})$.

Finally, loops of reachable states are formed. For each $s_1,\ s_2 \in \mathcal{S}$, which can coincide, we include $(s_2,\ s_1)$ in θ, where: (1) there is no $s_3 \in \mathcal{S}$ such that $p(s_3)$ is above $p(s_2)$ in $\tilde{V}(|S|)$, (2) the path $p(s_1)$ contains the base sequent of derivation loop that contains $p(s_2)$, and (3) $\theta = R_i$ if the uppermost S_{bp} in that derivation loop is a premise of (K_i).

Using induction on the formula length λ, we prove that $(M,\ s) \models \phi$ $((M,\ s) \not\models \phi)$, for any member formula ϕ in the antecedent (succedent, respectively) of any sequent in $p(s)$ in $\tilde{V}(|S|)$.

If $\lambda = 1$ and ϕ is a member of the antecedent (succedent) of some sequent in $p(s)$, then ϕ is a member of Σ_1 (Σ_2, respectively) in the end sequent S_{ec} of $p(s)$, and the proof is obtained.

Suppose that $\lambda > 1$ and $\phi = \phi_1 \wedge \phi_2$ is a member of the succedent of some sequent S_1 in $p(s)$. It follows from the construction of $p(s)$ that there is a sequent S_2 in the path from S_1 to S_{ec} such that ϕ_1 or ϕ_2 is a member of the succedent of S_2. We have $(M,\ s) \not\models \phi_i$, where $i \in \{1,\ 2\}$, based on the inductive hypothesis. Hence $(M,\ s) \not\models \phi_1 \wedge \phi_2$.

Let $\phi = \phi_1 \wedge \phi_2$ be a member of the antecedent of some sequent S_1 in $p(s)$. It follows from the construction of $p(s)$ that there is a sequent S_2 in the path from S_1 to S_{ec} such that ϕ_1 and ϕ_2 are members of the antecedent of S_2. We have $(M,\ s) \models \phi_i$, where $1 \leq i \leq 2$, based on the inductive hypothesis. Hence $(M,\ s) \models \phi_1 \wedge \phi_2$.

We skip the remaining cases when the outermost connective of ϕ is a propositional one because they are considered in the same way as the two previous cases.

Suppose that $\phi = K_i\psi$ is a member of the antecedent of some sequent S_1 in $p(s)$. It follows from the construction of $\tilde{V}(|S|)$ and M that each path $p(s_1)$, where $(s,\ s_1) \in R_i$, starts with a sequent that has the member ψ in antecedent. Hence $(M,\ s_1) \models \psi$, according to the inductive hypothesis, and $(M,\ s) \models K_i\psi$, based on the definition of semantics of K_i.

Assume that $\phi = K_i\psi$ is a member of the succedent of some sequent S_1 in $p(s)$. It follows from the construction of $\tilde{V}(|S|)$ and M that there is a path $p(s_1)$, where $(s,\ s_1) \in R_i$, that starts with a sequent the succedent of which has the member ψ. We obtain $(M,\ s_1) \not\models \psi$, according to the inductive hypothesis. Hence $(M,\ s) \not\models K_i\psi$, based on the definition of semantics of K_i.

Let $\phi = C\psi$ be a member in the succedent of some sequent S_1 in $p(s)$. It follows from the construction of $\tilde{V}(|S|)$ and M that there is $s_1 \in \mathcal{S}$ such that s_1 is coincident with or reachable from s and $p(s_1)$ starts with a sequent the succedent of which has the member $K_i\psi$ $(i \in \{1,\ \ldots,\ \mu\})$. We have $(M,\ s_1') \not\models \psi$, where $s_1 \mapsto s_1'$, according to the inductive hypothesis. Hence $(M,\ s) \not\models C\psi$, based on the definition of semantics of the operator C.

Let $\phi = C\psi$ be a member in the antecedent of some sequent S_1 in $p(s)$. It follows from the construction of $\tilde{V}(|S|)$ and M that, for each $s_1 \in \mathbb{S}$ reachable from s, the path $p(s_1)$ contains a sequent the antecedent of which has the members $K_i\psi$ $(1 \leq i \leq \mu)$. This fact yields $(M,\ s_1') \models \psi$ for each s_1' such that $s_1 \mapsto s_1'$, according to the inductive hypothesis. We have $(M,\ s) \models C\psi$, based on the definition of semantics of C.

The proved fact yields $(M,\ s) \not\models S'$ for any sequent S' in any $p(s)$ in $\tilde{V}(|S|)$. Hence $\not\models |S|$. This fact yields $\not\models S$ by Proposition 1. □

A derivation rule is called invertible in **GLCK**, iff derivability of its conclusion implies derivability of its premise(s) in **GLCK**.

Lemma 5 If the conclusion of any **GLCK** derivation rule, except (K_i), is valid, then each premise of this rule is valid, as well

Proof. The lemma is proved similarly as Lemma 2, by showing for each of the considered rules that $(M,\ s) \models S_c$ implies $(M,\ s) \models S_p$, where S_c is the conclusion and S_p is any premise of a rule. □

Theorem 3 All derivation rules of **GLCK**, except (K_i), are invertible.

Proof. If a conclusion of a **GLCK** rule is derivable, then it is valid, according to Theorem 1. Hence each premise of the rule is valid, based on Lemma 5. This fact implies that the premise(s) is (are) derivable, according to Theorem 2. □

7. Validity check

Making use of the calculus **GLCK** and obtained from it calculus **GLCK***, we construct a method that allows us to verify if a given sequent is valid in **LCK**.

Definition 10 A **GLCK** backward proof-search tree V is called a refutation tree, iff it contains a weakly-closing leaf.

Proposition 6 Any refutation tree $V^r(|S|)$ can be extended to a complete-refutation tree by backward rule applications.

Proof. The tree $V^r(|S|)$ has a weakly-closing leaf. We apply backward derivation rules to the leaves of $V^r(|S|)$ that are non-closing until each leaf is closing. The obtained tree is finite, according to Lemma 1. One can see that it is a complete-refutation tree. □

A **GLCK*** proof-search tree $V(|S|)$ is called a refutation tree if all its **GLCK**-sub-trees $V_{sub}(|S|)$ are refutation-trees.

The decision tree (DT) for an arbitrary sequent S is described similarly as the fully-extended tree $V^*(|S|)$ in the proof of Theorem 2. First, a fully-extended $(STEP)$-free proof-search tree $V(|S|)$ is generated. If $V(|S|)$ is a refutation or proof tree, then DT is obtained. Otherwise, the construction of DT is continued by: (1) applying backward $(STEP)$ to each non-closing leaf of $V(|S|)$, (2) generating a fully-extended $(STEP)$-free tree for each premise of $(STEP)$, and (3) applying backward $(STEP)$ again to the non-closing leaves. DT is obtained as soon as a refutation or proof tree is obtained. DT consists of **GLCK**-sub-trees $V_{sub}(|S|)$. Each such $V_{sub}(|S|)$ is finite, according to Lemma 1. Hence DT is finite.

The validity-check procedure for **LCK** is defined as follows: given an arbitrary sequent S, generate DT for $|S|$. In some finite number of steps, we get that DT is a refutation or proof tree. In the former case, there is a complete-refutation tree, based on Proposition 6. Hence $\not\models S$, as it is shown in the proof of Theorem 2. In the latter case, $\vdash S$; hence $\models S$, according to Theorem 1.

Example 3 Let us consider the sequent at root of proof-search tree in Example 1. We construct DT for $S = (p,\ Cp \Rightarrow Cq)$. The fully-extended $(STEP)$-free proof-search tree is generated first:

$$\dfrac{\dfrac{\left(p,\ (K_i p)^i,\ (K_i Cp)^i \Rightarrow K_j q\right)^j \quad \left(p,\ (K_i p)^i,\ (K_i Cp)^i \Rightarrow K_j Cq\right)^j}{p,\ (K_i p)^i,\ (K_i Cp)^i \Rightarrow Cq}(\Rightarrow C)}{p,\ Cp \Rightarrow Cq}(C \Rightarrow)$$

(We assume that $\mu = 2$ and, e.g., $(K_i p)^i$ is understood as $(K_i p)_{1 \leq i \leq 2}$). None of the leaves is closing. We apply backward $(STEP)$ to the k-premise $j = 1$ of $(\Rightarrow Cq)$ and construct the fully-extended $(STEP)$-free proof-search tree for the obtained sequent:

$$\cfrac{\cfrac{p,\ (K_ip)^i,\ (K_iCp)^i \Rightarrow q}{p,\ Cp \Rightarrow q}\,(C\Rightarrow)}{p,\ (K_ip)^i,\ (K_iCp)^i \Rightarrow K_1 q}\,(\text{STEP})$$

$$\dots$$

(STEP) is (K_1) in our case. The leaf of this tree is a non-axiom atomic-like, i.e., weakly closing, sequent. We obtained a refutation tree because no other **GLCK**-sub-tree $V_{sub}(S)$ is left to consider. According to the validity-check procedure, the root sequent of V is not valid in **LCK**. We construct an interpretation $M = \langle \mathcal{S},\ R_1,\ R_2,\ \pi\rangle$ in the same way as we did in the proof of Theorem 2. We mark the weakly closing leaf and obtain $\tilde{V}(S)$:

$$\cfrac{\cfrac{\cfrac{\cfrac{\underline{p,\ (K_ip)^i,\ (K_iCp)^i \Rightarrow q}}{p,\ Cp \Rightarrow q}\,(K_1)}{p,\ (K_ip)^i,\ (K_iCp)^i \Rightarrow K_1 q}}{p,\ (K_ip)^i,\ (K_iCp)^i \Rightarrow Cq}}{p,\ Cp \Rightarrow Cq}$$

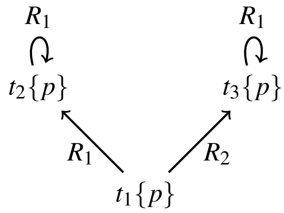
$S = \{s_1,\ s_2\}$, where $p(s_1)$ is the path from root to conclusion of (K_1), and $p(s_2)$ is the path from premise of (K_1) to the leaf. $\pi(s_1) = \pi(s_2) = \{p\}$ and $R_1 = \{(s_1,\ s_2)\}$. R_2 is the empty set. We have that $(M,\ s_i) \models p$ and $(M,\ s_i) \not\models q$ for $1 \le i \le 2$. Hence $(M,\ s_1) \not\models (p,\ Cp \Rightarrow Cq)$.

Example 4 Let $S = (p,\ C\phi \Rightarrow K_1\neg p,\ K_2\neg p)$, where $\phi = \neg K_1\neg p$. We assume that $\mu = 2$ for simplicity. A decision tree for this sequent is as follows:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{7:\ p, C\phi \Rightarrow K_1\neg p}{6:\ C\phi \Rightarrow K_1\neg p, \neg p}\,(\Rightarrow\neg)}{5:\ \neg K_1\neg p, C\phi \Rightarrow \neg p}\,(\neg\Rightarrow)}{4:\ p, (K_ip)^i, (K_iCp)^i \Rightarrow K_1\neg p}\,(K_1)}{3:\ p, C\phi \Rightarrow K_1\neg p}\,(C\Rightarrow)}{C\phi \Rightarrow K_1\neg p, \neg p}\,(\Rightarrow\neg)}{2:\ \neg K_1\neg p, C\phi \Rightarrow \neg p}\,(\neg\Rightarrow) \qquad \cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{7':\ p, C\phi \Rightarrow K_1\neg p}{6':\ C\phi \Rightarrow K_1\neg p, \neg p}\,(\Rightarrow\neg)}{5':\ \neg K_1\neg p, C\phi \Rightarrow \neg p}\,(\neg\Rightarrow)}{4':\ p, (K_ip)^i, (K_iCp)^i \Rightarrow K_1\neg p}\,(K_1)}{3':\ p, C\phi \Rightarrow K_1\neg p}\,(C\Rightarrow)}{C\phi \Rightarrow K_1\neg p, \neg p}\,(\Rightarrow\neg)}{2':\ \neg K_1\neg p, C\phi \Rightarrow \neg p}\,(\neg\Rightarrow)}{1:\ p, (K_i\phi)^i, (K_iC\phi)^i \Rightarrow K_1\neg p, K_2\neg p}\,(K_1)\,|\,(K_2)}{S:\ p, C\phi \Rightarrow K_1\neg p, K_2\neg p}\,(C\Rightarrow)$$

Instead of (STEP), we use here the names of rules that are actually applied. $(K_1)\,|\,(K_2)$ denotes the fact that the left premise 2 is obtained by applying (K_1), and the right one $2'$ is obtained by applying (K_2) backwards to 1.

It is true that $[3 \circ 7]$ and $[3' \circ 7']$ are α-void derivation loops. As in the previous example, we construct an interpretation $M = \langle \mathcal{S},\ R_1,\ R_2,\ \pi\rangle$. The obtained decision tree is, at the same time, the tree $\tilde{V}(S)$ in this case. $p(t_1) = p(S,\ 1)$, where $p(S,\ 1)$ denotes the path from S to 1 inclusive. $p(t_2)$ is the path from the premise 2 to conclusion 4 of (K_1) plus the sequents 5, 6, 7, i.e., $p(t_2) = p(2,\ 7)$. Also $p(t_3) = p(2',\ 7')$. We have $\mathcal{S} = \{t_1,\ t_2,\ t_3\}$; $\pi(t_1) = \pi(t_2) = \pi(t_3) = \{p\}$; $R_1 = \{(t_1,\ t_2),\ (t_2,\ t_2),\ (t_3,\ t_3)\}$; $R_2 = \{(t_1,\ t_3)\}$:



One can see that $(M, t_1) \not\models S$.

8. Conclusion

The finitary loop-type sequent calculus **GLCK** for **LCK** has been introduced in the present paper. We have proved that **GLCK** is sound and complete, and used it to obtain the validity-check procedure for **LCK**. The proof of Theorem 6. allows us to construct a pair (M, t) for any non-provable sequent S such that $(M, t) \not\models S$. Two-sided sequents and the non-minimal set of propositional connectives have been used to make the meaning of formulas, sequents, and derivation rules more evident in comparison to one-sided sequents and formulas kept in negation normal form.

The calculus **GLCK** has the subformula-like property: any backward proof-search tree contains only formulas that are subformulas of formulas in the root $|S|$, including also $K_i\phi$ and $K_iC\phi$ if $C\phi$ occurs in $|S|$.

Cut-free sequent calculus S_{CK}, containing annotated derivation rules, is considered for **LCK** in [20]. Since the calculus is Tait-style, formulas are kept in negation normal form. Annotated formulas make the subformula property less evident. In these aspects, S_{CK} proof analysis is harder in comparison to **GLCK**. A proof system with annotated sequents and analytical cut-rule for the logic of common knowledge over S5 is presented in [21].

An infinitary Tait style proof system S for **LCK** is presented in [6], where infinite branches of proof-search trees are axiomatically closed by so-called C-treads. Validity-check using the proof system is not dealt with in that work. A similar approach is used for an infinitary Tait style proof system for the linear time μ-calculus in [22], where two validity-check procedures based on the proof system are presented. Infinite branches of proof-search trees are axiomatically closed here by so-called ν-treads. The fact that there is a derivation loop in a branch does not imply that the branch can be closed by a ν-thread because the derivation loop can be β-void. It follows that larger proof-search trees must be generated by the decision procedures in the general case in comparison to DT in the present paper.

Two finitary cut-free Tait style sequent calculi for the modal μ-calculus are introduced in [23]. One of the calculi has the rule ind_S in the premise of which there is the formula $\overline{\Gamma}$, where Γ is the context of conclusion; the other operates with annotated sequents. Modal μ-calculi extend **LCK**, though one can conjecture that translation of **LCK** to the standard μ-calculi is not straightforward [24], based on the fact that **LCK** captures the induction principle of LTL. The subformula-like property for deductive systems of modal μ-calculi is less apparent in comparison with **GLCK** because of more intricate fixed points.

Backward proof-search of sequents using **GLCK** involves loop-check i.e., comparison of sequents. In some cases, the loop-check can be simplified or avoided. For example, any derivation loop has an application of (K_i), according to Proposition 3. Hence the fact that a path has no applications of (K_i) is enough to conclude that there are no derivation loops on the path. A possible future work is the specification of loop-check for the sake of more efficient backward proof-search. Also, one can consider how the derivation loop method works with other logics that capture the induction principle, such as propositional dynamic logic. Traditional sequent calculi of such logics usually contain a cut rule that essentially violates the subformula property, which makes them unsuitable for decision procedures of the logics.

Acknowledgement

We thank the anonymous reviewers for their useful comments and suggestions.

Conflict of interest

The authors declare no competing financial interest.

References

[1] Halpern JY, Moses Y. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*. 1990; 37(3): 549-587.

[2] Vanderschraaf P, Giacomo S. *The Stanford Encyclopedia of Philosophy: The Metaphysics Research Lab.* Philosophy Department Stanford University Stanford; 2023.

[3] Paech B. Gentzen-systems for propositional temporal logics. In: Börger E, Büning HK, Richter MM. (eds.) *Workshop on Computer Science Logic Duisburg, FRG*. Springer Berlin Heidelberg; 1989. p.240-253.

[4] Fagin R, Halpern JY, Moses Y, Vardi MY. *Reasoning About Knowledge*. Massachusetts: MIT Press; 1995.

[5] Meyer JJ Ch, van der Hoek W. *Epistemic Logic for AI and Computer Science*. Cambridge University Press; 1995.

[6] Bucheli S, Kuznets R, Studer T. Two ways to common knowledge. *Electronic Notes in Theoretical Computer Science*. 2010; 262: 83-98.

[7] Alberucci L, Jäger G. About cut elimination for logics of common knowledge. *Annals of Pure and Applied Logic*. 2005; 133: 73-99.

[8] Kretz M, Studer T. Deduction chains for common knowledge. *Journal of Applied Logic*. 2006; 4: 379-388.

[9] Pliuškevičius R. Infinitary calculus without loop rules for restricted sequents of the first-order linear temporal logic. *Lithuanian Mathematical Journal*. 2000; 40: 379-388.

[10] Jäger G, Kretz M, Studer T. Cut-free common knowledge. *Journal of Applied Logic*. 2007; 5(4): 681-689.

[11] Studer T, Marti M. The proof theory of common knowledge. In: van Ditmarsch H, Sandu G. (eds.) *Jaakko Hintikka on Knowledge and Game Theoretical Semantics*. Cham, Switzerland: Springer; 2018. p.433-455.

[12] Alonderis R, Sakalauskaitė J. A labelled sequent calculus for half-order modal logic. *Journal of applied logics-IFCoLog Journal of Logic and its Applications*. 2018; 5(1): 121-163.

[13] Hudelmaier J. A contraction-free sequent calculus for S4. In: Wansing H. *Proof Theory of Modal Logic*. Dordrecht: Springer Netherlands; 1996. p.3-15.

[14] Naoyuki N, Takata S. Deduction systems for BDI logics using sequent calculus. In: *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2*. New York, NY, USA: Association for Computing Machinery; 2002. p.928-935.

[15] Wansing H. Sequent calculi for normal modal propositional logics. *Journal of Logic and Computation*. 1994; 4(2): 125-142.

[16] Pliuškevičius R, Pliuškevičienė A. A new method to obtain termination in backward proof search for modal logic S4. *Journal of Logic and Computation*. 2010; 20(1): 353-379.

[17] Alonderis R, Giedra H. A proof-search system for the logic of likelihood. *Logic Journal of the IGPL*. 2019; 28(3): 261-280.

[18] Alonderis R, Pliuškevičius R, Pliuškevičienė A, Giedra H. Loop-type sequent calculi for temporal logic. *Journal of Automated Reasoning*. 2020; 64(8): 1663-1684.

[19] Schütte K. *Proof Theory*. Berlin and New York: Springer; 1997.

[20] Wehbe R. *Annotated Systems for Common Knowledge*. Universität Bern: CiteSeerX; 2010.

[21] Rooduijn J, Zenger L. An analytic proof system for common knowledge logic over S5. In: Fernández-Duque D, Palmigiano A, Pinchinat S, (eds.) *Proceedings of AiML*. Advances in modal logic. Bern: College Publications; 2022. p.659-680.

[22] Dax C, Hofman M, Lange M. A proof system for the linear time -calculus. In: Arun-Kumar S, Garg N, (eds.) *FSTTCS 2006: Foundations of Software Technology and Theoretical 23*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2006. p.273-284.

[23] Afshari B, Leigh GE. Cut-free completeness for modal mu-calculus. In: *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. Reykjavik, Iceland: Institute of Electrical and Electronics Engineers (IEEE); 2017. p.1-12.

[24] Cranen S, Groote JF, Reniers MA. *A Linear Translation From LTL to the First-Order Modal -Calculus*. Eindhoven, Netherlands: Technische Universiteit Eindhoven; 2010.