Proof Search and Certificates for Evidential Transactions

Vivek Nigam[1][0000−0003−4089−1218], Giselle Reis[2][0000−0002−5145−9829], Samar Rahmouni[2][0000−0003−1351−1515], and Harald Ruess[3][0000−0002−1405−2990]

[1] Huawei Munich Research Center, Germany vivek.nigam@gmail.com
[2] Carnegie Mellon University, Qatar giselle@cmu.edu, srahmoun@andrew.cmu.edu
[3] fortiss GmbH, Germany ruess@fortiss.org

Abstract. Attestation logics have been used for specifying systems with policies involving different principals. Cyberlogic is an attestation logic used for the specification of Evidential Transactions (ETs). In such transactions, evidence has to be provided supporting its validity with respect to given policies. For example, visa applicants may be required to demonstrate that they have sufficient funds to visit a foreign country. Such evidence can be expressed as a Cyberlogic proof, possibly combined with non-logical data (*e.g.*, a digitally signed document). A key issue is how to construct and communicate such evidence/proofs. It turns out that attestation modalities are challenging to use established proof-theoretic methods such as focusing. Our first contribution is the refinement of Cyberlogic proof theory with knowledge operators which can be used to represent knowledge bases local to one or more principals. Our second contribution is the identification of an executable fragment of Cyberlogic, called Cyberlogic programs, enabling the specification of ETs. Our third contribution is a sound and complete proof system for Cyberlogic programs enabling proof search similar to search in logic programming. Our final contribution is a proof certificate format for Cyberlogic programs inspired by Foundational Proof Certificates as a means to communicate evidence and check its validity.

Keywords: Attestation Logics · Proof Search · Sequent Calculus

1 Introduction

Attestation logics [1,14,21,15,6,5,29] have been used for the specification of policies of distributed systems, such as access control systems [1], distributed authorization policies [14,21], and evidential transactions (ETs) [15,5,6,6,29]. In these logics, one specifies policies involving attestation formulas of the form $\mathsf{K} :\triangleright F$, where K is a principal (or agent) in the system.

Cyberlogic is an attestation logic for ETs. In Cyberlogic, cryptographic keys K are identified with specific authorities, and attestations $\mathsf{K} :\triangleright A$ express the fact that principal K attests to statement A. For example, K may be a visa-granting authority and A the statement that the visa requester is authorized

to enter the specified country by the end of the year and at most once. An evidential transaction might issue a visa given that proof of sufficient funds has been provided in the form of a digital certificate whose validity can then be verified by customs authorities upon entry.

Formally, evidence in ETs can be expressed as a Cyberlogic proof. To carry out an ET, a Cyberlogic proof demonstrating policy compliance shall be produced and communicated. ETs therefore enable trust in, for example, distributed exchanges in electronic commerce, by enabling the exchange of various forms of *verifiable evidence*, such as evidence of funds in the visa example above.

The problem of producing attestation logic proofs (and proof objects) has not been given enough attention so far. Attestation logics have been formalized as Hilbert-style proof systems [1,15] that do not have the sub-formula property and therefore are not suitable for proof search. Other works on authorization logics [14,21] have proposed sequent calculi which do possess the sub-formula property. However, the search space is too great to enable efficient proof search.

The established proof-theoretic method for proof search is *focusing* [3,18]. Focusing distinguishes between inference rules that have "don't know" and "don't care" non-determinism to prune the proof search space. Interestingly, focused proof systems [7,18] provide a proof-theoretical justification for backward and forward-chaining, two proof-search strategies for Horn clauses (logic programs). Such justification, however, breaks when programs contain modalities, such as attestation modalities, *i.e.*, formulas of the form $\mathsf{K} :\triangleright F$. This is because focusing is lost whenever any of these formulas is encountered and therefore, improvements to the search space because of focusing is not so significant for attestation logics.

Our main goal is the study of Cyberlogic's proof theory in order to enable proof search (similar to the search involved in logic programming) and the generation of proof certificates for the communication of evidence in ETs.

Our first contribution, detailed in Section 2, is a Gentzen style proof system for Cyberlogic that admits cut elimination. A feature of the proof system is that it enables the combination of evidence represented as logical derivations as well as digital evidence, *e.g.*, signed hashes of documents, financial statements, medical records. The logic also includes a knowledge operator for sets of principals.

Our second contribution, detailed in Section 3, is the identification of a fragment of Cyberlogic, called Cyberlogic programs, akin to Horn clauses used in logic programming. This is motivated by the ongoing work on building distributed logic programming engines for ETs which extend existing engines [10] with attestations of the form $\mathsf{K} :\triangleright A$.

Our third contribution, also detailed in Section 3, addresses the challenge of how to efficiently construct Cyberlogic program proofs. We propose a focused inspired proof system for Cyberlogic programs and prove that it is sound and complete in this fragment. This system enables more efficient proof search.

Our last contribution, detailed in Section 4, addresses the challenge of how to efficiently communicate evidence. We propose a proof certificate format for Cyberlogic programs inspired by Foundational Proof Certificates (FPCs) [9]. FPCs enable the reconstruction of proofs by using simple logic programs as guides. This

$$\frac{}{\Gamma, A \longrightarrow A}\ \mathsf{init} \qquad \frac{\mathsf{evidence}_{\mathsf{K}} A}{\Gamma \longrightarrow \mathsf{K} :\rhd A}\ \mathsf{ext} \qquad \frac{}{\Gamma \longrightarrow \top}\ \top_r \qquad \frac{}{\Gamma, \bot \longrightarrow C}\ \bot_l$$

$$\frac{\Gamma, F_1, F_2 \longrightarrow G}{\Gamma, F_1 \wedge F_2 \longrightarrow G}\ \wedge_l \qquad \frac{\Gamma \longrightarrow F_1 \quad \Gamma \longrightarrow F_2}{\Gamma \longrightarrow F_1 \wedge F_2}\ \wedge_r \qquad \frac{\Gamma, F_1 \longrightarrow G \quad \Gamma, F_2 \longrightarrow G}{\Gamma, F_1 \vee F_2 \longrightarrow G}\ \vee_l \qquad \frac{\Gamma \longrightarrow F_i}{\Gamma \longrightarrow F_1 \wedge F_2}\ \vee_{r_i}$$

$$\frac{\Gamma, F_1 \supset F_2 \longrightarrow F_1 \quad \Gamma, F_2 \longrightarrow G}{\Gamma, F_1 \supset F_2 \longrightarrow G}\ \supset_l \qquad \frac{\Gamma, F_1 \longrightarrow F_2}{\Gamma \longrightarrow F_1 \supset F_2}\ \supset_r$$

$$\frac{\Gamma, \forall x.F, F[t/x] \longrightarrow G}{\Gamma, \forall x.F \longrightarrow G}\ \forall_l \qquad \frac{\Gamma \longrightarrow F[\alpha/x]}{\Gamma \longrightarrow \forall x.F}\ \forall_r \qquad \frac{\Gamma, F[\alpha/x] \longrightarrow G}{\Gamma, \exists x.F \longrightarrow G}\ \exists_l \qquad \frac{\Gamma \longrightarrow F[t/x]}{\Gamma \longrightarrow \exists x.F}\ \exists_r$$

$$\frac{\Gamma, F \longrightarrow \mathsf{K} :\rhd G}{\Gamma, \mathsf{K} :\rhd F \longrightarrow \mathsf{K} :\rhd G}\ :\rhd_l \qquad \frac{\Gamma \longrightarrow F}{\Gamma \longrightarrow \mathsf{K} :\rhd F}\ :\rhd_r \qquad \frac{\Gamma, \mathsf{kb}_{\mathcal{Q}} F, F \longrightarrow G}{\Gamma, \mathsf{kb}_{\mathcal{Q}} F \longrightarrow G}\ \mathsf{kb}_l \qquad \frac{\Gamma \mid_{\mathcal{Q}} \longrightarrow F}{\Gamma \longrightarrow \mathsf{kb}_{\mathcal{Q}} F}\ \mathsf{kb}_r$$

Fig. 1. $\mathsf{CL}_{\mathcal{K}}$ – Cyberlogic proof system for $\mathcal{K} = \{\mathsf{K}_1, \ldots, \mathsf{K}_n\}$. Here A is an atomic formula, $\mathcal{Q} \subseteq \mathcal{K}$, and $\Gamma \mid_{\mathcal{Q}} = \{\mathsf{kb}_{\mathcal{Q}'} F \mid \mathsf{kb}_{\mathcal{Q}'} F \in \Gamma \wedge \mathcal{Q}' \subseteq \mathcal{Q}\}$. Moreover, in rules $\exists_L$ and $\forall_R$, α is a fresh constant not appearing in Γ nor F.

means that such certificates can elide parts that can be easily reconstructed or which one is willing to reconstruct.

2 Cyberlogic Proof Theory

Cyberlogic [29] is an intuitionistic modal logic which can be used for specifying ETs. The logic is parametrized by a finite set of principals $\mathcal{K} = \{\mathsf{K}_1, \ldots, \mathsf{K}_n\}$, which are used in formulas as follows:

- $\mathsf{K}_i :\rhd F$: meaning that principal K_i attests the (Cyberlogic) formula F;
- $\mathsf{kb}_{\mathcal{Q}} F$, where $\mathcal{Q} \subseteq \mathcal{K}$: meaning that all principals in $\mathcal{Q}$ know F, or, alternatively, that the combined knowledge of principals in $\mathcal{Q}$ imply F; and
- $\mathsf{evidence}_{\mathsf{K}_i} A$: standing for an external evidence signed by principal K_i.

External evidences are left unspecified since they fall outside the logical scope and depend on the ET being formalized. For example, $\mathsf{evidence}_{\mathsf{K}_i} A$ could be signed hashes of tickets, financial statments, medical records, etc. In Cyberlogic the evidence associated with an ET is a combination of a formal proof (in sequent calculus) and a collection of external evidences.

Cyberlogic formulas are constructed according to the following grammar:

$$F, G ::= A \mid F \wedge G \mid F \vee G \mid F \supset G \mid \top \mid \bot \mid \mathsf{K} :\rhd F \mid \mathsf{kb}_{\mathcal{Q}} F \mid \forall x.F \mid \exists x.F$$

where A is an atom, $\mathsf{K} \in \mathcal{K}$, and $\mathcal{Q} \subseteq \mathcal{K}$. The formula $\mathsf{K} :\rhd F$ is read as "principal K attests F" and acts like the *says* modality in lax logics [13,27]. The formula $\mathsf{kb}_{\mathcal{Q}} F$ is read as "principals in $\mathcal{Q}$ know F" and is inspired by the *knows* modality used in linear authorization logics [14,21]. Different from that logic, Cyberlogic allows the direct specification of knowledge shared by multiple principals, as illustrated in Example 1.

Cyberlogic sequents are of the shape $\Gamma \longrightarrow G$, where Γ is a multiset of formulas. The Cyberlogic proof system, $\mathsf{CL}_{\mathcal{K}}$, is depicted in Figure 1. Rules for

the intuitionistic connectives $\wedge, \vee, \supset, \forall, \exists$ are as in LJ [30]. The new rules are the ones involving assertions $\mathsf{K} :\rhd F$ and $\mathsf{kb}_{\mathcal{Q}}$. Note that a "built-in" contraction of the main formula is needed on the left premise of $\supset_l$ and the premise of $\forall_l$, as expected in intuitionistic logics. Also, the rule kb_l has an explicit contraction on the premise. These contractions are needed for cut admissibility (Theorem 2).

Rules $:\rhd_l$ and $:\rhd_r$ specify that $:\rhd$ is a lax modality [27,21,24]. The intuition behind $:\rhd_l$ is: if an assertion G of a principal K is provable using F, then it is also provable if K attests F. Rule $:\rhd_r$ specifies that principals are rational, *i.e.*, they can always attest formulas that are derivable. Differently from existing systems with lax modalities, $\mathsf{CL}_{\mathcal{K}}$ has the rule ext. This rule allows a proof of an attestation $\mathsf{K} :\rhd A$ to be completed whenever a principal provides evidence $\mathsf{evidence}_{\mathsf{K}} A$ for the claim A. This formalizes the intuition that principals may use digital evidence signed by their private key. We leave the definition of evidence unspecified as it depends on the intended ET specified.

Rules kb_l and kb_r refine Cyberlogic by enabling the collection of logical theories known by a set of principals. Such theories act as *knowledge bases*. Rule kb_l specifies that any common knowledge can be part of a knowledge base. The interesting rule is kb_r, which specifies that $\mathsf{kb}_{\mathcal{Q}} F$ can only be proved using the local knowledge or evidence provided by principals in $\mathcal{Q}$. This is formally captured by restricting Γ in kb_r's premise to the set $\Gamma \mid_{\mathcal{Q}} = \{\mathsf{kb}_{\mathcal{Q}'} F \mid \mathsf{kb}_{\mathcal{Q}'} F \in \Gamma \wedge \mathcal{Q}' \subseteq \mathcal{Q}\}$. This is a powerful construct that increases the expressiveness of Cyberlogic. In particular, it is straightforward to specify that certain assertions can be concluded from the shared knowledge of a set of principals.

Proposition 1. *The following sequents are provable in* $\mathsf{CL}_{\mathcal{K}}$ *for all* $\mathsf{K} \in \mathcal{K}$ *and formulas* F_1, F_2. $F_1 \equiv F_2$ *represents the sequents* $(F_1 \longrightarrow F_2)$ *and* $(F_2 \longrightarrow F_1)$*:*

1. $F \longrightarrow \mathsf{K} :\rhd F$
2. $\mathsf{kb}_{\mathcal{Q}} F \longrightarrow F$
3. $\mathsf{kb}_{\{\mathsf{K}\}} F \longrightarrow \mathsf{K} :\rhd F$
4. $\mathsf{K} :\rhd \mathsf{K} :\rhd F \equiv \mathsf{K} :\rhd F$
5. $\mathsf{kb}_{\mathcal{Q}'} F \longrightarrow \mathsf{kb}_{\mathcal{Q}} F$, *if* $\mathcal{Q}' \subseteq \mathcal{Q}$. *In particular,* $\mathsf{kb}_{\mathcal{Q}_1} \mathsf{kb}_{\mathcal{Q}_2} F \longrightarrow \mathsf{kb}_{\mathcal{Q}_1 \cup \mathcal{Q}_2} F$.
6. $\mathsf{kb}_{\mathcal{Q}_1} F \wedge \mathsf{kb}_{\mathcal{Q}_2} F \longrightarrow \mathsf{kb}_{\mathcal{Q}_1 \cup \mathcal{Q}_2} F$
7. $\mathsf{K} :\rhd (F_1 \wedge F_2) \equiv \mathsf{K} :\rhd F_1 \wedge \mathsf{K} :\rhd F_2$
8. $\mathsf{kb}_{\mathcal{Q}}(F_1 \wedge F_2) \equiv \mathsf{kb}_{\mathcal{Q}} F_1 \wedge \mathsf{kb}_{\mathcal{Q}} F_2$
9. $(\mathsf{K} :\rhd F_1 \vee \mathsf{K} :\rhd F_2) \longrightarrow \mathsf{K} :\rhd (F_1 \vee F_2)$
10. $\mathsf{kb}_{\mathcal{Q}} A \vee \mathsf{kb}_{\mathcal{Q}} B \longrightarrow \mathsf{kb}_{\mathcal{Q}}(A \vee B)$
11. $\mathsf{K} :\rhd (F_1 \supset F_2) \longrightarrow (\mathsf{K} :\rhd F_1 \supset \mathsf{K} :\rhd F_2)$
12. $\mathsf{kb}_{\mathcal{Q}}(F_1 \supset F_2) \longrightarrow (\mathsf{kb}_{\mathcal{Q}} F_1 \supset \mathsf{K} :\rhd F_2)$
13. $\mathsf{K} :\rhd (\nabla x.F) \equiv \nabla x.\mathsf{K} :\rhd F$, $\nabla \in \{\forall, \exists\}$
14. $\mathsf{kb}_{\mathcal{Q}}(\nabla x.F) \equiv \nabla x.\mathsf{kb}_{\mathcal{Q}} F$, $\nabla \in \{\forall, \exists\}$

Moreover, the following sequents are not provable if $\mathsf{K}_1 \neq \mathsf{K}_2$ *and* $\mathcal{Q}_1 \neq \mathcal{Q}_2$*:*

1. $\mathsf{K} :\rhd F \not\longrightarrow F$
2. $F \not\longrightarrow \mathsf{kb}_{\mathcal{Q}} F$
3. $\mathsf{K} :\rhd F \not\longrightarrow \mathsf{kb}_{\{\mathsf{K}\}} F$
4. $\mathsf{K}_1 :\rhd (\mathsf{K}_2 :\rhd F) \not\longrightarrow \mathsf{K}_2 :\rhd (\mathsf{K}_1 :\rhd F)$
5. $\mathsf{kb}_{\mathcal{Q}_1}(\mathsf{kb}_{\mathcal{Q}_2} F) \not\longrightarrow \mathsf{kb}_{\mathcal{Q}_2}(\mathsf{kb}_{\mathcal{Q}_1} F)$
6. $\mathsf{kb}_{\mathcal{Q}_1 \cup \mathcal{Q}_2} F \not\longrightarrow \mathsf{kb}_{\mathcal{Q}_i} F$, $i \in \{1, 2\}$
7. $\mathsf{kb}_{\mathcal{Q}_1 \cup \mathcal{Q}_2} F \not\longrightarrow \mathsf{kb}_{\mathcal{Q}_1} F \wedge \mathsf{kb}_{\mathcal{Q}_2} F$
8. $\mathsf{kb}_{\mathcal{Q}} \mathsf{K} :\rhd A \not\longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A$
9. $\mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A \not\longrightarrow \mathsf{kb}_{\mathcal{Q}} \mathsf{K} :\rhd A$

In the remainder of the paper, we elide the set of principals $\mathcal{K}$ whenever it can be deduced from the context.

Example 1. **(Shared Knowledge)** The ability to use kb with multiple principals allows the derivation of facts that depend on the combination of knowledge of multiple principals. Consider that principal K_1 knows A and $B \supset C$, and principal K_2 knows $A \supset B$, then the following sequent is provable in CL:

$$\mathsf{kb}_{\{\mathsf{K}_1\}} A, \mathsf{kb}_{\{\mathsf{K}_1\}} B \supset C, \mathsf{kb}_{\{\mathsf{K}_2\}} A \supset B \longrightarrow \mathsf{kb}_{\{\mathsf{K}_1,\mathsf{K}_2\}} C$$

Remark 1. The original Cyberlogic paper [5] (and technical report [4]) proposed two kinds of attestations, $:\!\triangleright$ and $\triangleright$, to distinguish when an attestation is derived from a digital evidence or logical inferences. This combination, however, does not yield to a proof system with the cut-elimination property [28].

The meta-theory of CL has been analysed using the L-framework [25], which uses rewriting logic to automatically derive structural proofs of sequent calculi properties [26]. The following lemma was used in the proofs of cut-elimination and invertibility.

Lemma 1. *If* $\Gamma, \mathsf{K} :\!\triangleright F \longrightarrow G$*, then* $\Gamma, F \longrightarrow G$*.*

The proof proceeds by structural induction on the derivation of $\Gamma, \mathsf{K} :\!\triangleright F \longrightarrow G$. The proof has been mechanically checked using the the L-framework with some few cases proved by hand.

As expected, $\supset_r, \wedge_r, \wedge_l, \vee_l, \forall_r, \exists_l$ are invertible whereas $\vee_r, \supset_l, \forall_l, \exists_r$ are not invertible. In addition, the rules $:\!\triangleright_l$ and kb_l are invertible whereas the $:\!\triangleright_r$ and kb_r are not invertible.

Lemma 2. *If* $\Gamma, \mathsf{K} :\!\triangleright F \longrightarrow \mathsf{K} :\!\triangleright G$ *then* $\Gamma, F \longrightarrow \mathsf{K} :\!\triangleright G$*.*

This is a simple corollary of Lemma 1. Invertibility of kb_l is straightforward because of the contraction of the main formula.

Rules $:\!\triangleright_r$ and kb_r are not invertible. The counter examples are:

$$[:\!\triangleright_r] \quad \mathsf{K} :\!\triangleright a \longrightarrow \mathsf{K} :\!\triangleright a \;\text{ but }\; \mathsf{K} :\!\triangleright a \not\longrightarrow a$$
$$[\mathsf{kb}_r] \quad a, a \supset \mathsf{kb}_{\mathsf{K}} b \longrightarrow \mathsf{kb}_{\mathsf{K}} b \;\text{ but }\; \not\longrightarrow b$$

Weakening is height preserving admissible in CL.

Theorem 1 (Identity expansion). *$F \longrightarrow F$ is provable in CL for any cyberlogic formula F.*

The proof is by structural induction on F.

Theorem 2 (Cut elimination). *If $\Gamma \longrightarrow F$ and $\Gamma, F \longrightarrow C$, then $\Gamma \longrightarrow C$.*

The proof proceeds by a nested induction on the structure of the proofs of $\Gamma \longrightarrow F$ and $\Gamma, F \longrightarrow C$, and the formula F. The noteworthy cases are the ones where cut needs to permute over kb rules. For kb_l, contraction of the main formula is needed, and the permutation over kb_r can be done only if cut is principal on the left (which is a lemma that can be proved). Details about these transformations are in Appendix A.

3 Cyberlogic Programs

Cyberlogic programs are fragment of CL which resembles Horn clauses in logic programming. Section 3.2 proposes a proof search operational semantics for cyberlogic programs and proves its soundness and completeness. The proof search discipline relies on ideas from focusing [3]. Focused proof systems for LJ [18] provide a proof theoretical justification of forward and backward chaining search. Each technique is enforced by the choice of polarity of atomic formulas: positive atoms lead to forward chaining and negative atoms lead to backward chaining. This correspondence, however, does not extend to cyberlogic due to attestation formulas $\mathsf{K} :\rhd A$ which cause focusing to be lost [21]. Consider the following example where the formula under focus is in brackets:

$$\frac{\mathsf{K}_1 :\rhd a \longrightarrow [\mathsf{K}_1 :\rhd a] \quad \mathsf{K}_1 :\rhd a, [\mathsf{K}_2 :\rhd b] \longrightarrow \mathsf{K}_2 :\rhd b}{\mathsf{K}_1 :\rhd a, [\mathsf{K}_1 :\rhd a \supset \mathsf{K}_2 :\rhd b] \longrightarrow \mathsf{K}_2 :\rhd b} \supset_l$$

In focused proof systems, forward chaining can be enforced by disallowing focus to be lost on the right formula in the left premise, *i.e.* $[\mathsf{K}_1 :\rhd a]$. However, if $:\rhd_r$ is applied to this sequent the premise would be $\mathsf{K}_1 :\rhd a \longrightarrow a$, which is not provable (see Proposition 1). In fact, $[\mathsf{K}_1 :\rhd a]$ must lose focus on the right for the proof to be completed. Therefore, if $:\rhd$ modalities are used in logic programs, other strategies for proof search need to be analysed.

3.1 Cyberlogic Program Syntax

Cyberlogic programs can be divided into goals, knowledge bases, common knowledge, and attestation clauses.

Goals (G) Cyberlogic programs are used to derive a goal G, defined as:

$$G ::= \top \mid \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A \mid G_1 \wedge G_2 \mid \exists x.G$$

where A is an atomic formula. The restriction of $:\rhd \mathsf{kb}_{\mathcal{Q}}$ to atoms does not reduce the expressiveness of goals, given the equivalences in Proposition 1.

Knowledge Bases ($\mathcal{B}$): A knowledge base, written $\mathsf{kb}_{\{\mathsf{K}_i\}}\Gamma$, of a principal $\mathsf{K}_i \in \mathcal{K}$ is a set of formulas Γ not containing the connectives $:\rhd$ or kb. Here, $\mathsf{kb}_{\{\mathsf{K}_i\}}\Gamma$ represents the set of formulas $\{\mathsf{kb}_{\{K_i\}}F \mid F \in \Gamma\}$.

Intuitively, a knowledge base $\mathsf{kb}_{\{\mathsf{K}_i\}}\Gamma$ can be interpreted as K_i's local knowledge. This means that K_i may use its own prover to derive new facts. For example, if Γ is a collection of Horn-clauses, then K_i may deploy a Prolog engine to derive some goal. Alternatively if Γ is a set of formulas in CNF form, then K_i may use resolution provers. The absence of modal connectives in knowledge bases has important impacts on the design of the proof certificate described in Section 4, as those may rely on existing certificates for different provers [9].

Common Knowledge ($\mathcal{C}$): Common knowledge are knowledge bases that are known to all principals, written as $\mathsf{kb}_\emptyset\ \Gamma$. Since $\emptyset \subseteq \mathcal{Q}$ for every $\mathcal{Q}$, these formulas remain in the context when applying kb_r. In this sense they contain first order formulas that may be used by all principals.

Attestation Formulas ($\mathcal{D}$): Formulas of the form $\mathsf{K} :\rhd \mathsf{kb}_\mathcal{Q} A$ are derived by attestation formulas of the form below where for all $1 \leq i \leq n$, $\mathsf{K}_i \in \mathcal{K}$, $\mathcal{Q}_i \subseteq \mathcal{K}$, and $A_1, \ldots, A_n, A$ are atomic formulas and $\vec{X}$ are bounded by universal quantifiers:

$$\forall \vec{X}.\big(\mathsf{kb}_{\mathcal{Q}_1}(\mathsf{K}_1 :\rhd A_1) \wedge \cdots \wedge \mathsf{kb}_{\mathcal{Q}_n}(\mathsf{K}_n :\rhd A_n) \wedge G \supset \mathsf{K} :\rhd (\mathsf{kb}_\emptyset A)\big)$$
$$\forall \vec{X}.\big(\mathsf{kb}_{\mathcal{Q}_1}(\mathsf{K}_1 :\rhd A_1) \wedge \cdots \wedge \mathsf{kb}_{\mathcal{Q}_n}(\mathsf{K}_n :\rhd A_n) \wedge G \supset \mathsf{K} :\rhd (\mathsf{kb}_{\{\mathsf{K}\}} A)\big)$$

Intuitively, an attestation formula belongs to a principal, namely K in the right-hand side of $\supset$. Such formulas derive K's attestation of an atomic formula which is its own knowledge ($\mathsf{kb}_{\{\mathsf{K}\}} A$), or common knowledge ($\mathsf{kb}_\emptyset A$). This means that K's attestation formulas cannot derive knowledge belonging to other principals. Furthermore to derive an attestation, one can use the knowledge base of other principals, *i.e.* the formulas $\mathsf{kb}_{\mathcal{Q}_i}(\mathsf{K}_i :\rhd A_i)$ or additional goals, *i.e.* G. Finally notice that $\mathsf{K} :\rhd (\mathsf{kb}_\emptyset A)$ and $\mathsf{K} :\rhd (\mathsf{kb}_{\{\mathsf{K}\}} A)$ are attestation formulas themselves, where the left-hand side of $\supset$ is empty (denoting $\top$).

The difference between formulas $\mathsf{K} :\rhd A$ and $\mathsf{K} :\rhd (\mathsf{kb}_{\{\mathsf{K}\}} A)$ is subtle. Note that the former can be derived using the evidence rule ext, while the latter cannot. $\mathsf{K} :\rhd (\mathsf{kb}_{\{\mathsf{K}\}} A)$ is K's attestation that A follows from its local knowledge base. It is possible to specify that A can be derived from an external evidence, but this has to be made explicit by an attestation formula, *e.g.*, $\mathsf{kb}_{\{\mathsf{K}\}}(\mathsf{K} :\rhd A) \supset \mathsf{K} :\rhd (\mathsf{kb}_{\{\mathsf{K}\}} A)$. Note that this formula is not a tautology.

We are interested in proving goals from attestation formulas, knowledge bases, and common knowledge, which are formally represented by cyberlogic program sequents defined as follows.

Definition 1 (Cyberlogic Program Sequents (CPS)). *A* cyberlogic program sequent (CPS) *is a sequent $\mathcal{C}, \mathcal{B}, \mathcal{D} \longrightarrow G$, where $\mathcal{B}$ is a set of knowledge bases, $\mathcal{C}$ is a set of common knowledge formulas, $\mathcal{D}$ is a set of attestation formulas, and G is a goal formula.*

Example 2. **(Local Computations)** This example illustrates the use of kb to specify when parts of a derivation can be proved locally using a principal's knowledge. Consider that the following clause

$$\mathsf{kb}_{\{\mathsf{K}_1\}}(\mathsf{K}_1 :\rhd F_1) \wedge \mathsf{kb}_{\{\mathsf{K}_2\}}(\mathsf{K}_2 :\rhd F_2) \supset \mathsf{K} :\rhd \mathsf{kb}_{\{\mathsf{K}\}} G$$

specifies that for K to attest G, K_1 and K_2 have to attest F_1 and F_2 respectively, using *their own local theories, common knowledge, or evidence.* This means that computations carried out by K_1 and K_2 to derive their assertions $\mathsf{K}_1 :\rhd F_1$ and $\mathsf{K}_2 :\rhd F_2$ respectively, do not depend on other principals and therefore, the search for these derivations can be performed locally.

Example 3. (**Levels of Trust**) This example illustrates the use of kb to specify that some evidence should only be trusted if derived from trusted sources. Consider three principals $\mathcal{K} = \{\mathsf{K}_T, \mathsf{K}_U, \mathsf{K}\}$ where K trusts evidence from K_T, but not all evidence from K_U. Then the following clause

$$\mathsf{kb}_{\{\mathsf{K},\mathsf{K}_T\}}(\mathsf{K} :\rhd \mathsf{critical}(\mathsf{ok})) \wedge \mathsf{kb}_{\mathcal{K}}(\mathsf{K} :\rhd \mathsf{nonCritical}(\mathsf{ok})) \supset \mathsf{K} :\rhd \mathsf{kb}_{\emptyset}(\mathsf{all}(\mathsf{ok}))$$

specifies that K can attest that everything is ok as a common knowledge if all the non-critical and critical elements are ok. However, the check of critical parts can only be performed by principals K trusts, namely K itself or K_T. Information from K_U's knowledge bases cannot be used in the proof of critical(ok).

Example 4. (**Simplified Visa**) Consider a visa issuing scenario where an applicant applies to a consulate (cons) for an entry visa. This is an example of an ET as, to obtain the visa, evidence has to be provided that, for example, the applicant has no crime records, or that they have sufficient funds. We illustrate how such an ET can be specified in Cyberlogic.

The formula below labelled **main** specifies conditions for a visa to be issued:

$$\begin{aligned}\textbf{main: } & \forall \mathsf{Id}.\forall \mathsf{Doc}.\forall \mathsf{V}.\big(\mathsf{kb}_{\{\mathsf{cons}\}}(\mathsf{cons} :\rhd \mathsf{visitOk}(\mathsf{Id},\mathsf{Doc})) \\ & \wedge\ \mathsf{kb}_{\{\mathsf{cons}\}}(\mathsf{cons} :\rhd \mathsf{prepVisa}(\mathsf{Id},\mathsf{V})) \\ & \wedge\ \mathsf{cons} :\rhd \mathsf{kb}_{\{\mathsf{cons}\}}(\mathsf{sufFin}(\mathsf{Doc})) \wedge \mathsf{police} :\rhd \mathsf{kb}_{\{\mathsf{police}\}}(\mathsf{noCrimeRec}(\mathsf{Id})) \\ & \supset\ \mathsf{cons} :\rhd \mathsf{kb}_{\mathsf{cons}}(\mathsf{issVisa}(\mathsf{Id},\mathsf{Doc},\mathsf{V}))\big)\end{aligned}$$

The transaction for cons issuing a visa V to an applicant Id requires cons to attest validity of Id's visit by itself (visitOk(Id, Doc)) and Id's criminal record with the help of the police (noCrimeRec(Id)). In addition, cons also needs to attest Id's financial status (sufFin(Doc)).

The following two clauses expand on how cons can attest sufFin(Doc): either via an employment contract or a bank statement.

$$\begin{aligned}\textbf{cont: } & \mathsf{kb}_{\{\mathsf{cons}\}}\big(\forall \mathsf{Doc}.\forall \mathsf{Cont}.\big(\ \mathsf{empContract}(\mathsf{Doc},\mathsf{Cont}) \wedge \mathsf{valid}(\mathsf{Cont}) \\ & \qquad\qquad \supset\ \mathsf{sufFin}(\mathsf{Doc})\big)\big) \\ \textbf{bankStmt: } & \forall \mathsf{Doc}.\forall \mathsf{Stmt}.\big(\mathsf{kb}_{\{\mathsf{cons}\}}(\mathsf{cons} :\rhd \mathsf{bankStmt}(\mathsf{Doc},\mathsf{Stmt})) \\ & \qquad \wedge\ \mathsf{bank} :\rhd \mathsf{kb}_{\{\mathsf{bank}\}}(\mathsf{valid}(\mathsf{Stmt})) \supset \mathsf{cons} :\rhd \mathsf{kb}_{\{\mathsf{cons}\}}(\mathsf{sufFin}(\mathsf{Doc}))\big)\end{aligned}$$

The formula labeled **cont** belongs to cons's knowledge base. This means that cons can check the validity of an employment contract without evidence from other principals. For example, valid(Cont) may check the contract duration and salary. The formula labeled **bankStmt**, on the other hand, takes the bank statement Stmt from the given documents, Doc, and requires the bank to validate it using its knowledge base. This makes sense as Id's financial records are sensitive and do not need to be disclosed to anyone else apart from her financial institute.

These clauses also illustrate the subtle difference between goal formulas $\mathsf{K} :\rhd \mathsf{kb}_{\{K\}}F$ and knowledge base formulas $\mathsf{kb}_{\{K\}}\mathsf{K} :\rhd F$. For example, in the **main** clause, the fact that applicant has come to their appointment at the consulate does not depend on other agents and that is why we use a knowledge base formula. The same applies to the visa preparation. On the other hand, the fact that applicant has sufficient funds may require evidence from other parties, *e.g.*, the applicant's bank. Therefore this is specified as a goal.

Goal decomposition

$$\frac{}{\Theta;\Lambda;\Delta \longrightarrow [\top]}\ \top_r \qquad \frac{\Theta;\Lambda;\Delta \longrightarrow [G_1] \quad \Theta;\Lambda;\Delta \longrightarrow [G_2]}{\Theta;\Lambda;\Delta \longrightarrow [G_1 \wedge G_2]}\ \wedge_r \qquad \frac{\Theta;\Lambda;\Delta \longrightarrow [G[t/x]]}{\Theta;\Lambda;\Delta \longrightarrow [\exists x.G]}\ \exists_r$$

$$\frac{\Theta;\Lambda;[\Delta] \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A}{\Theta;\Lambda;\Delta \longrightarrow [\mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A]}\ G \Rightarrow :\rhd_l \qquad \frac{\Theta \mid^{\star}_{\mathcal{Q}} \longrightarrow A}{\Theta;\Lambda;\Delta \longrightarrow [\mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A]}\ :\rhd_r + \mathsf{kb}_r + \mathsf{kb}_l$$

$:\rhd_l$ application

$$\frac{\Theta, \mathsf{kb}_{\mathcal{Q}} A;\Lambda;[\Delta] \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}'} A'}{\Theta;\Lambda;[\Delta, \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A] \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}'} A'}\ :\rhd_l$$

$$\frac{\Theta;[\Lambda];\Delta^{\dagger} \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A}{\Theta;\Lambda;[\Delta^{\dagger}] \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A}\ :\rhd_l \Rightarrow \mathsf{att} \qquad \frac{\Theta;\Lambda;\Delta^{\dagger} \longrightarrow [\mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A]}{\Theta;\Lambda;[\Delta^{\dagger}] \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A}\ :\rhd_l \Rightarrow G$$

Attestation formula decomposition

$$\frac{\begin{array}{c}\Theta;\Lambda;\Delta \longrightarrow [G\sigma] \quad \Theta;\Lambda;[\Delta, \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A\sigma] \longrightarrow \mathsf{K}' :\rhd \mathsf{kb}_{\mathcal{Q}'} A' \\ \Theta \mid_{\mathcal{Q}_1};\cdot;\cdot \longrightarrow [\mathsf{K}_1 :\rhd A_1\sigma] \quad \cdots \quad \Theta \mid_{\mathcal{Q}_n};\cdot;\cdot \longrightarrow [\mathsf{K}_n :\rhd A_n\sigma]\end{array}}{\Theta;[\Lambda, \forall\vec{X}.(\mathsf{kb}_{\mathcal{Q}_1}(\mathsf{K}_1 :\rhd A_1) \wedge \cdots \wedge \mathsf{kb}_{\mathcal{Q}_n}(\mathsf{K}_n :\rhd A_n) \wedge G \supset \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A)];\Delta \longrightarrow \mathsf{K}' :\rhd \mathsf{kb}_{\mathcal{Q}'} A'}\ \mathsf{att}$$

$\mathsf{K} :\rhd A$ decomposition

$$\frac{\mathsf{evidence}_{\mathsf{K}} A}{\Theta;\cdot;\cdot \longrightarrow [\mathsf{K} :\rhd A]}\ \mathsf{ext} \qquad \frac{\Theta^{\star} \longrightarrow A}{\Theta;\cdot;\cdot \longrightarrow [\mathsf{K} :\rhd A]}\ :\rhd_r + \mathsf{kb}_l$$

First-order reasoning:

All first-order rules from CL on $\Theta^{\star} \longrightarrow A$ sequents

Fig. 2. $\mathsf{CL}_{\mathcal{P}}$ – Sequent calculus for cyberlogic programs. A, A' and A_i are atoms, $\Delta^{\dagger}$ is such that for all $\mathsf{K}' :\rhd \mathsf{kb}'_{\mathcal{Q}} A' \in \Delta^{\dagger}$, $\mathsf{K}' \neq \mathsf{K}$, and $\Theta^{\star} = \{F \mid \mathsf{kb}_{\mathcal{Q}} F \in \Theta\}$.

3.2 CPS Proof Search

Proof search of CPS can be divided into the following phases: goal decomposition, $:\rhd_l$ application, attestation formula decomposition, $\mathsf{K} :\rhd A$ decomposition, and first-order reasoning. We define a (focusing inspired) sequent calculus for the CPS fragment, called $\mathsf{CL}_{\mathcal{P}}$ (Figure 2) for enforcing this proof search discipline. Sequents in $\mathsf{CL}_{\mathcal{P}}$ have the following shape: $\Theta;\Lambda;\Delta \longrightarrow F$, where Θ contains kb formulas, Λ contains attestation formulas, Δ contains formulas of the form $\mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A$, and F is either a goal formula, $\mathsf{kb}_{\mathcal{Q}}(\mathsf{K} :\rhd A)$, $\mathsf{K} :\rhd A$ or A, where A is an atom. Moreover, the part of the sequent containing the formula that is being decomposed will be enclosed in square brackets. This will help distinguishing the phases mentioned above.

Lemma 3. *The kb_r rules permutes down every left rule in the CPS fragment.*

Proof. First we note that, in the CPS fragment, $\wedge$, $\vee$, $\forall$, and kb formulas on the left do not have kb modalities as subformulas. We look at the case of kb_l, as the others follow a similar argument.

Since F is not a kb formula, then $F \notin (\Gamma, \mathsf{kb}_{\mathcal{Q}'}F, F) \mid_{\mathcal{Q}}$. Therefore we can conclude that $(\Gamma, \mathsf{kb}_{\mathcal{Q}'}F, F) \mid_{\mathcal{Q}} = (\Gamma, \mathsf{kb}_{\mathcal{Q}'}F) \mid_{\mathcal{Q}}$ and the permutation is:

$$\dfrac{\dfrac{\overset{\varphi}{(\Gamma, \mathsf{kb}_{\mathcal{Q}'}F, F) \mid_{\mathcal{Q}} \longrightarrow G}}{\Gamma, \mathsf{kb}_{\mathcal{Q}'}F, F \longrightarrow \mathsf{kb}_{\mathcal{Q}}G}\ \mathsf{kb}_r}{\Gamma, \mathsf{kb}_{\mathcal{Q}'}F \longrightarrow \mathsf{kb}_{\mathcal{Q}}G}\ \mathsf{kb}_l \quad \rightsquigarrow \quad \dfrac{\overset{\varphi}{(\Gamma, \mathsf{kb}_{\mathcal{Q}'}F) \mid_{\mathcal{Q}} \longrightarrow G}}{\Gamma, \mathsf{kb}_{\mathcal{Q}'}F \longrightarrow \mathsf{kb}_{\mathcal{Q}}G}\ \mathsf{kb}_r$$

The case for $:\!\triangleright_l$ holds vacuously, as it is impossible to have $:\!\triangleright_l$ immediately below kb_r since the former requires the right formula to be of the shape $\mathsf{K} :\!\triangleright$.

The remaining case is $\supset_l$. Observe that in the CPS fragment, the formula F_2 in $F_1 \supset F_2$ is of the form $\mathsf{K} :\!\triangleright \mathsf{kb}_{\mathcal{Q}'}A$. Therefore, $(\Gamma, F_2) \mid_{\mathcal{Q}} = \Gamma \mid_{\mathcal{Q}}$. Also, $(\Gamma, F_1 \supset F_2) \mid_{\mathcal{Q}} = \Gamma \mid_{\mathcal{Q}}$. Thus the permutation is:

$$\dfrac{\Gamma \longrightarrow F_1 \quad \dfrac{\overset{\varphi}{(\Gamma, F_2) \mid_{\mathcal{Q}} \longrightarrow G}}{\Gamma, F_2 \longrightarrow \mathsf{kb}_{\mathcal{Q}}G}\ \mathsf{kb}_r}{\Gamma, F_1 \supset F_2 \longrightarrow \mathsf{kb}_{\mathcal{Q}}G}\ \supset_l \quad \rightsquigarrow \quad \dfrac{\overset{\varphi}{(\Gamma, F_1 \supset F_2) \mid_{\mathcal{Q}} \longrightarrow G}}{\Gamma, F_1 \supset F_2 \longrightarrow \mathsf{kb}_{\mathcal{Q}}G}\ \mathsf{kb}_r$$

□

Notice that it is crucial for attestation formulas to have a $:\!\triangleright$ modality formula on the consequent, otherwise Lemma 3 would not hold. As seen below, this lemma is key to proving completeness of the proof search procedure for CPS.

Theorem 3 (Soundness and completeness of $\mathsf{CL}_{\mathcal{P}}$). *$\Theta; \Lambda; \Delta \longrightarrow [F]$ in $\mathsf{CL}_{\mathcal{P}}$ if and only if $\Theta, \Lambda, \Delta \longrightarrow F$ in CL*

Proof. Soundness is straightforward: a proof in $\mathsf{CL}_{\mathcal{P}}$ can be transformed into a proof in CL by using the same logical rules (possibly expanded – *e.g.* att becomes a sequence of $\forall_l + \supset_l + \wedge_r + \mathsf{kb}_r$) and skipping the phase transition rules $\Rightarrow$ (which only change the syntax of the sequent, but not its content).

Completeness is achieved by reasoning about invertibility and permutability of inference rules in the specific case of CPS. We argue that each phase can be performed in the proposed order.

Goal decomposition The goal formula can be eagerly decomposed until becoming $\mathsf{K} :\!\triangleright \mathsf{kb}_{\mathcal{Q}}A$ before applying other rules because: $\top_r$ and $\wedge_r$ are invertible, and in the absence of $\forall_r$ and $\exists_l$, $\exists_r$ permutes down every rule. Once the right side formula is $\mathsf{K} :\!\triangleright \mathsf{kb}_{\mathcal{Q}}A$, there are two options to continue: (1) change to $:\!\triangleright_l$ application phase, or (2) apply rules $:\!\triangleright_r + \mathsf{kb}_r + \mathsf{kb}_l$ in Figure 1.

The first case is discussed below. In the second case, we need to argue that kb_r may be applied immediately above $:\!\triangleright_r$. Once $:\!\triangleright_r$ is applied, we could choose a formula from the context to continue with. However, kb_r permutes down all left rules for the CPS fragment, as shown in Lemma 3. Therefore any proof that continues with a formula in Θ, Λ, or Δ above $:\!\triangleright_r$ can be transformed into a proof where kb_r is applied immediately above $:\!\triangleright_r$. Since kb_l is invertible, it can be applied to exhaustion safely.

$:\!\triangleright_l$ application After eagerly decomposing the goal, $:\!\triangleright_l$ can be applied to exhaustion since it is an invertible rule (Lemma 2).

Attestation formula decomposition This phase contains only one rule, namely att, which encompasses $\forall_l$, $\supset_l$, $\wedge_r$, and kb_r. The quantifier rule can always be delayed until its subformula is needed, and $\wedge_r$ is an invertible rule, therefore these can be chained together without loss of completeness. Due to Lemma 3, the application of kb_r can be permuted down for the CPS fragment and thus it is safe to apply the rule as soon as possible.

The two top premises of att force the proof search to go back to applying invertible rules, which does not break completeness.

$\mathsf{K} :\rhd A$ **decomposition** Once this state is reached, Θ is left with kb formulas whose subformulas are in first-order logic (i.e., no modalities). In this case, one can either close the proof with an external evidence, or apply $:\rhd_r + \mathsf{kb}_l$ to release the atom on the right side. The eager application of kb_l is justified due to its invertibility. It can also be delayed until this point because it permutes up $\supset_l$ and $:\rhd_r$ in CL, and it permutes up kb_r in the CPS fragment (Lemma 3).

First-order reasoning From this point onwards, there are no modalities in the sequent so it will be proved using only first-order reasoning. □

4 Proof Certificates

Cyberlogic programs may be used to derive facts about attestation (goals), using pure logical reasoning (knowledge bases), principal delegation (attestation formulas), and external evidence. Once a goal is derived, evidence shall be available so that any interested party can verify that the proof is correct. Verifiable evidence means that entities do not need to trust each other's proof producing process, as long as they can check the proofs using their own trusted processes.

Given a cyberlogic program sequent of the shape: $\Theta; \Lambda; \Delta \longrightarrow G$ one could take its full sequent calculus proof in $\mathsf{CL}_{\mathcal{P}}$ as evidence. If the interested parties know the calculus, checking validity of proofs reduces to checking the valid application of each rule. However, these proofs are too fine grained, and contain many uninteresting details that can be easily inferred. Proof certificates elide such details, and keep only the crucial steps for proof reconstruction.

Proof certificates for cyberlogic are defined inspired by λ-terms and *foundational proof certificates* [8,20] (FPC). FPC is a framework for checking proofs in different formalisms using a small trusted kernel. The proposed kernels are the sequent calculus focused systems LKF and LJF [18] for LK and LJ respectively, augmented with predicates for guiding proof search [9]. The definition of proof certificates for a proof system $\mathcal{S}$ relies on two parts: (1) a translation of $\mathcal{S}$'s formulas into LKF or LJF formulas; and (2) a correspondence of $\mathcal{S}$ proofs (or proof steps) to LKF or LJF proof steps. Given these two elements, a proof certificate for a proof of F in $\mathcal{S}$ consists of a predicate which guides a proof of F' s translation in LKF or LJF. The following proof formats can be checked in FPC: resolution, λ-terms, Horn clauses, Frege proofs, matings, tableaux, etc.

Defining LKF or LJF FPCs for cyberlogic is challenging due to the modalities $:\rhd$ and kb, and digital evidences. LKF has been used to check proofs in modal logics [19], but the translation of modal formulas into LK formulas used the

$$\frac{}{\mathsf{top} : \Theta; \Lambda; \Delta \longrightarrow [\top]}\ \top_r \qquad \frac{\Xi : \Theta; \Lambda; \Delta \longrightarrow [G[t/x]]}{\Xi : \Theta; \Lambda; \Delta \longrightarrow [\exists x.G]}\ \exists_r$$

$$\frac{\Xi_1 : \Theta; \Lambda; \Delta \longrightarrow [G_1] \quad \Xi_2 : \Theta; \Lambda; \Delta \longrightarrow [G_2]}{\mathsf{split}(\Xi_1, \Xi_2) : \Theta; \Lambda; \Delta \longrightarrow [G_1 \wedge G_2]}\ \wedge_r$$

$$\frac{\Xi : \Theta; \Lambda; [\Delta] \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A}{\mathsf{toSays_L}(\Xi) : \Theta; \Lambda; \Delta \longrightarrow [\mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A]}\ G \Rightarrow :\rhd_l \qquad \frac{\Psi : \Theta \mid^{\star}_{\mathcal{Q}} \longrightarrow A}{\mathsf{fol}(\Psi) : \Theta; \Lambda; \Delta \longrightarrow [\mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A]}\ :\rhd_r + \mathsf{kb}_r + \mathsf{kb}_l$$

$$\frac{\Xi : \Theta, \mathsf{kb}_{\mathcal{Q}} A; \Lambda; [\Delta] \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}'} A'}{\Xi : \Theta; \Lambda; [\Delta, \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A] \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}'} A'}\ :\rhd_l$$

$$\frac{\Xi : \Theta; [\Lambda]; \Delta^{\dagger} \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A}{\mathsf{toAtt}(\Xi) : \Theta; \Lambda; [\Delta^{\dagger}] \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A}\ :\rhd_l \Rightarrow \mathsf{att} \qquad \frac{\Xi : \Theta; \Lambda; \Delta^{\dagger} \longrightarrow [\mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A]}{\mathsf{toGoal}(\Xi) : \Theta; \Lambda; [\Delta^{\dagger}] \longrightarrow \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A}\ :\rhd_l \Rightarrow G$$

$$\frac{\begin{array}{c}\Xi' : \Theta; \Lambda; \Delta \longrightarrow [G\sigma] \quad \Xi'' : \Theta; \Lambda; [\Delta, \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A\sigma] \longrightarrow \mathsf{K}' :\rhd \mathsf{kb}_{\mathcal{Q}'} A' \\ \Xi_1 : \Theta \mid_{\mathcal{Q}_1}; \cdot; \cdot \longrightarrow [\mathsf{K}_1 :\rhd A_1\sigma] \quad \cdots \quad \Xi_n : \Theta \mid_{\mathcal{Q}_n}; \cdot; \cdot \longrightarrow [\mathsf{K}_n :\rhd A_n\sigma]\end{array}}{\begin{array}{l}\mathsf{att}(i, \sigma, [\Xi_1, ..., \Xi_n], \Xi', \Xi'') : \\ \Theta; [\Lambda, i : \forall \vec{X}.(\mathsf{kb}_{\mathcal{Q}_1}(\mathsf{K}_1 :\rhd A_1) \wedge \cdots \wedge \mathsf{kb}_{\mathcal{Q}_n}(\mathsf{K}_n :\rhd A_n) \wedge G \supset \mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A)]; \Delta \longrightarrow \mathsf{K}' :\rhd \mathsf{kb}_{\mathcal{Q}'} A'\end{array}}\ \mathsf{att}$$

$$\frac{\mathsf{evidence}_{\mathsf{K}}(E, A)}{\mathsf{ext}(E) : \Theta; \cdot; \cdot \longrightarrow [\mathsf{K} :\rhd A]}\ \mathsf{ext} \qquad \frac{\Psi : \Theta^{\star} \longrightarrow A}{\mathsf{fol}(\Psi) : \Theta; \cdot; \cdot \longrightarrow [\mathsf{K} :\rhd A]}\ :\rhd_r + \mathsf{kb}_l$$

Fig. 3. $\mathsf{CL}^a_{\mathcal{P}}$ – $\mathsf{CL}_{\mathcal{P}}$ kernel for verifying $\mathsf{CL}_{\mathcal{P}}$ proof certificates of Cyberlogic programs. $\Delta^{\dagger}$ is such that for all $\mathsf{K}' :\rhd \mathsf{kb}'_{\mathcal{Q}} A' \in \Delta^{\dagger}$, $\mathsf{K}' \neq \mathsf{K}$ and $\Theta^{\star} = \{F \mid \mathsf{kb}_{\mathcal{Q}} F \in \Theta\}$.

modalities' semantic definition. Instead, we propose a modular $\mathsf{CL}_{\mathcal{P}}$ kernel which allows facts derived from knowledge bases or external evidence to be checked by the appropriate engine or entity.

The $\mathsf{CL}_{\mathcal{P}}$ kernel $\mathsf{CL}^a_{\mathcal{P}}$ (Figure 3) is constructed by augmenting sequents with a certificate Ξ (a term indicating how the proof must proceed) and indices for the formulas in Λ. A certificate for a proof of $\Theta; \Lambda; \Delta \longrightarrow G$ is $\Xi : \Theta; \Lambda_I; \Delta \longrightarrow G$, where Ξ is a term built from the predicates used in $\mathsf{CL}^a_{\mathcal{P}}$, and Λ_I is a mapping from indices to formulas in Λ. The indices are used in Ξ. The checking of a cyberlogic sequent $\Theta; \Lambda; \Delta \longrightarrow G$ with certificate Ξ starts from the sequent $\Xi : \Theta; \Lambda_I; \Delta \longrightarrow [G]$. Certificates denoted by the letter Ψ can represent proofs in other formalisms and may be checked by another engine. The predicates in Ξ are used for the following purposes during a derivation in $\mathsf{CL}^a_{\mathcal{P}}$.

First of all, they indicate how the proof should continue when there are multiple choices. For example, if the sequent is of the form $\Theta; \Lambda; \Delta \longrightarrow [\mathsf{K} :\rhd \mathsf{kb}_{\mathcal{Q}} A]$, then Ξ must be one of $\mathsf{toSays_L}(\_)$ or $\mathsf{fol}(\_)$, indicating whether to work on $:\rhd$ modalities on the left, or finish the proof with first-order reasoning, respectively.

Secondly, certificates relay information at the appropriate moment. For example, $\mathsf{split}(\_, \_)$ contains the certificates for each of the branches on a splitting rule, and $\mathsf{ext}(\_)$ includes an external evidence for proposition A. Note that there is no certificate for $\exists_R$ since these can be instantiated with meta-variables, and unification can be verified when the proof is completed.

The certificate for rule att is more interesting. It includes the index i of the attestation formula to be decomposed, the substitution σ for the $\forall$ quantifier, and certificates for each premise. Note that each $\Xi_1, ...\Xi_n$ must be $\mathsf{ext}(\_)$ or $\mathsf{fol}(\_)$.

Example 5. Consider Example 4, and let the indices of the formulas be their labels: **main**, **cont**, and **bankStmt**. The certificate for a proof that alice can get a visa is $\Xi : \mathbf{cont}; \mathbf{main}, \mathbf{bankStmt}; \cdot \longrightarrow \mathsf{cons} :\rhd \mathsf{kb}_{\{\mathsf{cons}\}}\mathsf{issVisa}(\mathsf{alice}, \mathsf{doc}, \mathsf{visa})$. Where Ξ is:

$$\mathsf{att}(\mathbf{main}, \{\mathsf{Id} \mapsto \mathsf{alice}, \mathsf{Doc} \mapsto \mathsf{doc}, \mathsf{V} \mapsto \mathsf{visa}\}, [\mathsf{fol}(\Psi_{\mathsf{visitOk}}), \mathsf{fol}(\Psi_{\mathsf{prepVisa}})], \Xi_G, \Xi_0)$$

The certificates Ψ_{visitOk} and Ψ_{prepVisa} are first-order logic proof certificates from derivations using the consulate's own knowledge base.

Certificate Ξ_0 corresponds to att's premise where the conclusion of **main** is added to the context. This branch can be closed by removing the modalities, so $\Xi_0 = \mathsf{toGoal}(\mathsf{fol}(\mathsf{id}))$, where id is a first-order logic directive to close the proof.

Certificate Ξ_G guides the proof of the new goal:

$$\mathsf{cons} :\rhd \mathsf{kb}_{\{\mathsf{cons}\}}(\mathsf{sufFin}(\mathsf{doc})) \wedge \mathsf{police} :\rhd \mathsf{kb}_{\{\mathsf{police}\}}(\mathsf{noCrimeRec}(\mathsf{alice}))$$

and thus $\Xi_G = \mathsf{split}(\Xi_{\mathsf{fin}}, \Xi_{\mathsf{crime}})$. Ξ_{fin} depends on how cons decides to check for sufficient funds. It could rely on the bank and use the attestation formula **bankStmt**, in which case Ξ_{fin} has the shape

$$\mathsf{toSays}_{\mathsf{L}}(\mathsf{toAtt}(\mathsf{att}(\mathbf{bankStmt}, \_, \_, \_, \_)))$$

Or it could use **cont** from its knowledge base, in which case Ξ_{fin} would be $\mathsf{fol}(\_)$.

5 Related Work

Attestation logics have been proposed for the specification of policies of several distributed systems [14,21,15,5,29,1]. We have been inspired by some of this work in the design of Cyberlogic. Actually, Cyberlogic was proposed some decades ago [29,5], but until now its proof theory had not been carefully investigated. In particular, there were no statements on cut-elimination. Additionally, we have been inspired by the previous works on authorization logics [14,21,15] to extend Cyberlogic with knowledge operators.

The main contribution of our work is the study of proof search and proof certificates for attestation logics with knowledge operators.

In previous work [14] in intuitionistic authorization logic, knowledge was restricted to one principal. As demonstrated in Example 1, allowing for multiple principal knowledge databases ensures collaboration in reasoning.

Proof search for attestation logics is not adequately addressed in the literature. Either the proposed proof systems are Hilbert-style [1,2,17] which do not enjoy the sub-formula property and therefore are not suitable for proof search, or they are sequent calculus proof system, but not focused proof systems [14,21,29,5,16]. [14] only speculates that logic programming languages can be used to carry out proof search for fragments of attestation logic. We confirm this speculation with the definition of Cyberlogic programs.

Our main inspiration for proof certificate is the work on foundational proof certificates [9]. However, the existing work did not consider proof certificates for attestation logics. Closer to our objective is the work of Libal and Volpe [19],

which define proof certificates for modal logics by encoding (the semantics of) these logics in LKF. Our work instead proposes proof certificates directly in Cyberlogic. This means that we are able to capitalize on rules, such as attestation rules, to build more compact certificates. Another difference is that our proof certificates may contain (pointers to) extra-logical evidence.

Cyberlogic has been formalized in Coq [11], encoding evidential transactions for Schengen Visa applications. Our approach is different in that it lays a proof theoretic foundation to Cyberlogic. In particular, proof search is formally justified as well as the representation of Cyberlogic proofs as FPCs.

Logic programming engines, such as ETB [10], have been proposed for programming ETs. However, these engines do not (yet) support attestations, such as $\mathsf{K} :\rhd F$, local knowledge, such as $\mathsf{kb}_{\mathcal{Q}} F$, nor the use of digital certificates. We believe that this work can greatly profit from the foundations laid by this paper.

Finally, works [15,6] propose the use of evidence for authorization. Specifically, [16] show that a fragment of their system is decidable in linear time. It would be interesting to investigate how this fragment relates to Cyberlogic programs, and whether proof certificates as defined in this work can be applied to the decidable fragment. This is left for future work.

6 Conclusions

This paper lays the proof-theoretic foundations for Cyberlogic, an attestation logic for evidential transactions, and refine Cyberlogic with epistemic modalities. We identify a fragment of Cyberlogic, Cyberlogic programs, and propose a proof system similar to focused proof systems for enabling sound and complete proof search. The necessary permutations for completeness rely on the careful interplay between attestation, $:\rhd$, and knowledge modalities, $\mathsf{kb}_{\mathcal{Q}}$. We then propose a concise proof certificate format for proofs of Cyberlogic programs.

This paper is the first step for a framework enabling evidential transactions that we are currently implementing. In particular, we are extending Distributed Datalog engines available in [10] to support Cyberlogic. Moreover, we are integrating such engines with PKI infrastructure, available in, for example, Distributed Ledger Technologies. This means that evidence, both in the form of digital evidence and logical derivations in the form of FPCs, can be stored and audited through the Ledger Technologies.

We are currently investigating extensions to Cyberlogic programs to include other modalities, such as temporal and epistemic [23,12] while still preserving its good proof search properties. We have also started to study conditions for when two attestation rules can be introduced in any order. If two clauses can be introduced in any order, then they can also be introduced in parallel. Therefore, this would provide proof-theoretic justification for proof search optimization. This could be used, for example, for proposing refinements to dependency graphs used for evaluating distributed logic programming [22] which take principals into account. These results will impact the maintenance of evidential transactions,

whose applications can have important consequences to, *e.g.*, certification in automotive and avionics domains.

Acknowledgment: We would like to thank Dian Balta, Natarajan Shankar and Tewodros Beyene for useful discussions and valuable feedback on earlier versions of this paper. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830892 and from BayernCloud 3, AZ: 20-13-3410.I-01A-2017. Nigam is partially supported CNPq grant 303909/2018-8.

A Cut-elimination

Proof. (Sketch) The proof follows the usual Gentzen strategy of reducing the cuts' grade and rank. The interesting cases are rank reduction over kb rules.

In the case of kb_l, contraction of the main formula is needed for the permutation to work. If this was not the case, we could not conclude $\Gamma, A \longrightarrow G$ from $\Gamma, \mathsf{kb}_{\mathcal{Q}} A \longrightarrow G$. The transformations are:

$$\dfrac{\dfrac{\overset{\varphi_1}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A, A \longrightarrow C}}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A \longrightarrow C}\ \mathsf{kb}_l \quad \overset{\varphi_2}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A, C \longrightarrow G}}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A \longrightarrow G}\ \mathsf{cut} \quad \rightsquigarrow \quad \dfrac{\dfrac{\overset{\varphi_1}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A, A \longrightarrow C} \quad \overset{\varphi_2 + \text{weakening}}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A, A, C \longrightarrow G}}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A, A \longrightarrow G}\ \mathsf{cut}}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A, \longrightarrow G}\ \mathsf{kb}_l$$

$$\dfrac{\overset{\varphi_1}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A \longrightarrow C} \quad \dfrac{\overset{\varphi_2}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A, A, C \longrightarrow G}}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A, C \longrightarrow G}\ \mathsf{kb}_l}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A \longrightarrow G}\ \mathsf{cut} \quad \rightsquigarrow \quad \dfrac{\dfrac{\overset{\varphi_1 + \text{weakening}}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A, A \longrightarrow C} \quad \overset{\varphi_2}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A, A, C \longrightarrow G}}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A, A \longrightarrow G}\ \mathsf{cut}}{\Gamma, \mathsf{kb}_{\mathcal{Q}} A \longrightarrow G}\ \mathsf{kb}_l$$

The other interesting case is when we need to permute a cut over a kb_r rule on the right branch:

$$\dfrac{\overset{\varphi_1}{\Gamma \longrightarrow C} \quad \dfrac{\overset{\varphi_2}{(\Gamma, C) \mid_{\mathcal{Q}_i} \longrightarrow G}}{\Gamma, C \longrightarrow \mathsf{kb}_{\mathcal{Q}_i} G}\ \mathsf{kb}_r}{\Gamma \longrightarrow \mathsf{kb}_{\mathcal{Q}_i} G}\ \mathsf{cut}$$

There are two cases to consider:

1. $C \equiv \mathsf{kb}_{\mathcal{Q}_j} C'$ and $\mathcal{Q}_i \preceq \mathcal{Q}_j$: in this case, we can permute the cut over rules on φ_1 (left rules except $:\!\triangleright_L$, which is never applicable) until it is principal. This lemma can be proved by case analysis. At this point, the premise on the left branch will be $\Gamma \mid_{\mathcal{Q}_j} \longrightarrow C'$. Then kb_R can be applied to the end-sequent, resulting in:
$$\dfrac{\overset{\varphi'_1}{\Gamma \mid_{\mathcal{Q}_i} \longrightarrow \mathsf{kb}_{\mathcal{Q}_j} C'} \quad \overset{\varphi'_2}{\Gamma \mid_{\mathcal{Q}_i}, \mathsf{kb}_{\mathcal{Q}_j} C' \longrightarrow G}}{\dfrac{\Gamma \mid_{\mathcal{Q}_i} \longrightarrow G}{\Gamma \longrightarrow \mathsf{kb}_{\mathcal{Q}_i} G}\ \mathsf{kb}_r}\ \mathsf{cut}$$
   The proof φ'_2 is exactly φ_2, since $(\Gamma, \mathsf{kb}_{\mathcal{Q}_j} C') \mid_{\mathcal{Q}_i} \equiv\ \Gamma \mid_{\mathcal{Q}_i}, \mathsf{kb}_{\mathcal{Q}_j} C'$ when $\mathcal{Q}_i \preceq \mathcal{Q}_j$. The proof φ'_1 is obtained from the proof of $\Gamma \mid_{\mathcal{Q}_j} \longrightarrow C'$, since $\Gamma \mid_{\mathcal{Q}_j} \subseteq \Gamma \mid_{\mathcal{Q}_i}$ when $\mathcal{K}_i \preceq \mathcal{Q}_j$.
2. $C \not\equiv \mathsf{kb}_{\mathcal{Q}_j} C'$ or $\mathcal{Q}_i \not\preceq \mathcal{Q}_j$: in this case $C \notin (\Gamma, C) \mid_{\mathcal{Q}_i}$, so kb_r can be applied directly to the end-sequent, and the cut can be removed.

□

References

1. Abadi, M.: Logic in Access Control. In: 18th IEEE Symposium on Logic in Computer Science (LICS) Proceedings. pp. 228–233. IEEE Computer Society (2003). https://doi.org/10.1109/LICS.2003.1210062
2. Abadi, M., Burrows, M., Lampson, B.W., Plotkin, G.D.: A Calculus for Access Control in Distributed Systems. ACM Trans. Program. Lang. Syst. **15**(4), 706–734 (1993). https://doi.org/10.1145/155183.155225
3. Andreoli, J.M.: Logic Programming with Focusing Proofs in Linear Logic. Joural of Logic and Computation **2**(3), 297–347 (1992). https://doi.org/10.1093/logcom/2.3.297
4. Bernat, V.: First-Order Cyberlogic Hereditary Harrop Logic. Tech. rep., SRI International (2006), http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/Bernat-cyberlogic1.ps
5. Bernat, V., Ruess, H., Shankar, N.: First-order Cyberlogic. Technical Report CSL-SRI-04-03, SRI International Computer Science Laboratory (2004)
6. Blass, A., Gurevich, Y., Moskal, M., Neeman, I.: Evidential Authorization. In: Nanz, S. (ed.) The Future of Software Engineering. pp. 73–99. Springer (2010). https://doi.org/10.1007/978-3-642-15187-3_5
7. Chaudhuri, K., Pfenning, F., Price, G.: A Logical Characterization of Forward and Backward Chaining in the Inverse Method. In: Furbach, U., Shankar, N. (eds.) Automated Reasoning, Third International Joint Conference, IJCAR, Proceedings. pp. 97–111. Springer Berlin Heidelberg (2006). https://doi.org/10.1007/11814771_9
8. Chihani, Z., Miller, D., Renaud, F.: Foundational Proof Certificates in First-Order Logic. In: Bonacina, M.P. (ed.) CADE-24 - 24th International Conference on Automated Deduction. Proceedings. Lecture Notes in Computer Science, vol. 7898, pp. 162–177. Springer (2013). https://doi.org/10.1007/978-3-642-38574-2_11
9. Chihani, Z., Miller, D., Renaud, F.: A Semantic Framework for Proof Evidence. J. Autom. Reasoning **59**(3), 287–330 (2017). https://doi.org/10.1007/s10817-016-9380-6
10. Cruanes, S., Hamon, G., Owre, S., Shankar, N.: Tool Integration with the Evidential Tool Bus. In: Giacobazzi, R., Berdine, J., Mastroeni, I. (eds.) Verification, Model Checking, and Abstract Interpretation, 14th International Conference, VMCAI. Proceedings. pp. 275–294. Springer Berlin Heidelberg (2013). https://doi.org/10.1007/978-3-642-35873-9_18
11. Dargaye, Z., Kirchner, F., Tuccı-Piergiovanni, S., Gürcan, O.: Towards Secure and Trusted-by-Design Smart Contracts. In: JFLA (2018)
12. DeYoung, H., Garg, D., Pfenning, F.: An Authorization Logic With Explicit Time. In: Proceedings of the 21st IEEE Computer Security Foundations Symposium, CSF. pp. 133–145. IEEE Computer Society (2008). https://doi.org/10.1109/CSF.2008.15
13. Fairtlough, M., Mendler, M.: Propositional Lax Logic. Inf. Comput. **137**(1), 1–33 (1997). https://doi.org/10.1006/inco.1997.2627
14. Garg, D., Bauer, L., Bowers, K.D., Pfenning, F., Reiter, M.K.: A Linear Logic of Authorization and Knowledge. In: Gollmann, D., Meier, J., Sabelfeld, A. (eds.) Computer Security - ESORICS 2006, 11th European Symposium on Research in Computer Security, Proceedings. pp. 297–312. Springer Berlin Heidelberg (2006). https://doi.org/10.1007/11863908_19

15. Gurevich, Y., Neeman, I.: DKAL: Distributed-Knowledge Authorization Language. Tech. Rep. MSR-TR-2008-09, Microsoft Research (January 2008), https://www.microsoft.com/en-us/research/publication/191tr-dkal-distributed-knowledge-authorization-language/
16. Gurevich, Y., Neeman, I.: DKAL 2 - A Simplified and Improved Authorization Language. Tech. Rep. MSR-TR-2009-11, Microsoft Research (2009), https://www.microsoft.com/en-us/research/publication/200-dkal-2-a-simplified-and-improved-authorization-language/
17. Gurevich, Y., Neeman, I.: Logic of infons: The propositional case. ACM Trans. Comput. Log. **12**(2), 9:1–9:28 (2011). https://doi.org/10.1145/1877714.1877715
18. Liang, C., Miller, D.: Focusing and polarization in linear, intuitionistic, and classical logics. Theor. Comput. Sci. **410**(46), 4747–4768 (2009). https://doi.org/10.1016/j.tcs.2009.07.041
19. Libal, T., Volpe, M.: A general proof certification framework for modal logic. Math. Struct. Comput. Sci. **29**(8), 1344–1378 (2019). https://doi.org/10.1017/S0960129518000440
20. Miller, D.: Foundational Proof Certificates. In: Delahaye, D., Paleo, B.W. (eds.) All about Proofs, Proofs for All, All about Proofs, Proofs for All, vol. Mathematical Logic and Foundations, 55, pp. 150–163. College Publications (2015), https://hal.inria.fr/hal-01239733
21. Nigam, V.: A framework for linear authorization logics. Theor. Comput. Sci. **536**, 21–41 (2014). https://doi.org/10.1016/j.tcs.2014.02.018
22. Nigam, V., Jia, L., Loo, B.T., Scedrov, A.: Maintaining distributed logic programs incrementally. Computer Languages, Systems & Structures **38**(2), 158–180 (2012). https://doi.org/10.1016/j.cl.2012.02.001
23. Nigam, V., Olarte, C., Pimentel, E.: A General Proof System for Modalities in Concurrent Constraint Programming. In: D'Argenio, P.R., Melgratti, H.C. (eds.) CONCUR 2013 - Concurrency Theory - 24th International Conference. Proceedings. Lecture Notes in Computer Science, vol. 8052, pp. 410–424. Springer (2013). https://doi.org/10.1007/978-3-642-40184-8_29
24. Nigam, V., Pimentel, E., Reis, G.: An extended framework for specifying and reasoning about proof systems. J. Log. Comput. **26**(2), 539–576 (2016). https://doi.org/10.1093/logcom/exu029
25. Olarte, C.: L-framework. https://carlosolarte.github.io/L-framework/, accessed on 03-01-2021
26. Olarte, C., Pimentel, E., Rocha, C.: Proving Structural Properties of Sequent Systems in Rewriting Logic. In: Rusu, V. (ed.) Rewriting Logic and Its Applications - 12th International Workshop, WRLA 2018, Held as a Satellite Event of ETAPS, Proceedings. Lecture Notes in Computer Science, vol. 11152, pp. 115–135. Springer (2018). https://doi.org/10.1007/978-3-319-99840-4_7
27. Pfenning, F., Davies, R.: A judgmental reconstruction of modal logic. Mathematical Structures in Computer Science **11**(4), 511–540 (2001). https://doi.org/10.1017/S0960129501003322
28. Reis, G.: Observations about the proof theory of cyberlogic. http://www.gisellereis.com/papers/cyberlogic-report.pdf (2019)
29. Ruess, H., Shankar, N.: Introducing Cyberlogic (2003)
30. Troelstra, A.S., Schwichtenberg, H.: Basic Proof Theory. Cambridge University Press (1996)