Formalized Soundness and Completeness of Epistemic Logic

Asta Halkjær From
DTU Compute
Technical University of Denmark
ahfrom@dtu.dk

Alexander Birch Jensen
DTU Compute
Technical University of Denmark
aleje@dtu.dk

Jørgen Villadsen
DTU Compute
Technical University of Denmark
jovi@dtu.dk

ABSTRACT

Epistemic logic allows reasoning about the knowledge of agents, and deductive proof systems enable this reasoning with a few axioms and inference rules. We strengthen the logical foundations of such a system by formalizing it in the proof assistant Isabelle/HOL. Our definitions are given in the precise language of higher-order logic and every step of our soundness and completeness proofs is mechanically checked.

KEYWORDS

Epistemic Logic, Isabelle/HOL, Formal Proof, Agent Logic

1 INTRODUCTION AND RELATED WORK

Epistemic logic provides a foundation for reasoning about the knowledge of agents, both factual ("I know the sky is blue") and higher-order ("I know that you know that I know the sky is blue"). A deductive proof system enables this reasoning with just a few axioms and inference rules. We formalize epistemic logic with countably many agents in the proof assistant Isabelle/HOL [5, 11]. We include soundness and completeness proofs for the axiom system K_n based on the textbook *Reasoning About Knowledge* by Fagin, Halpern, Moses and Vardi [3]. Our definitions and proofs are specified in the precise language of higher-order logic and every step of our reasoning is mechanically checked. While the results are not new, this level of precision and guarantee, due to formalization in a proof assistant, is. Our formalization can also serve as starting point for similar logics or proof systems.

Our completeness proof does not follow the one by Fagin et al. [3] to the letter but is inspired by Fitting's [4] consistency properties as formalized by Berghofer [1]. We have adapted them from first-order logic to epistemic logic.

It would be interesting to also formalize Dynamic Epistemic Logic [2] which adds dynamics to epistemic logic by considering changes to the knowledge of agents (epistemic events) brought about by events such as public announcements. Some variants also consider events which change the state of the world (ontic events).

In a formalization of a solution to a puzzle [10], the author introduces a logic tailored to the problem that turns out to be very similar to the possible worlds model of epistemic logic.

In [13] the authors present a variant of epistemic logic that adds the notion of secret knowledge as a first-class citizen. The notion of secrets can be defined in terms of the knowledge operator, but a new modality for secrets is introduced. The authors argue that the main principles can be studied this way, for instance when considering a language with an operator for secrets and without the usual knowledge operator. We think it would be interesting to formalize their work in a proof assistant.

An approach using Isabelle/HOL to verify agent programs is considered in [8, 9].

2 SYNTAX AND SEMANTICS

The formal language $\mathcal{L}$ for epistemic logic is a propositional language extended with modal operators $K_1, \ldots, K_n$ for expressing knowledge of agents, for example the formula

$$K_1\varphi \land K_2K_1\varphi \land \neg K_1K_2K_1\varphi$$

states that: (1) agent 1 knows φ, (2) agent 2 knows that agent 1 knows φ, but (3) agent 1 does not know that agent 2 knows (1).

The language is deeply embedded as a datatype in Isabelle/HOL:

```
datatype 'i fm
  = FF (⊥)
  | Pro id
  | Dis ‹'i fm› ‹'i fm› (infixr ∨ 30)
  | Con ‹'i fm› ‹'i fm› (infixr ∧ 35)
  | Imp ‹'i fm› ‹'i fm› (infixr ⟶ 25)
  | K 'i ‹'i fm›
```

The type variable $'i$ is an arbitrary type for agents. In our informal example, we used natural numbers, but we do not commit ourselves to any specific type. Our soundness proof holds for any type while the completeness proof holds for any countable type $'i$.

The semantics of epistemic logic formulas is based on a model of possible worlds as formalized by Kripke structures:

```
datatype ('i, 's) kripke
  = Kripke (π: ‹'s ⇒ id ⇒ bool›) (𝒦: ‹'i ⇒ 's ⇒ 's set›)
```

There are two components: an interpretation π that assigns truth values to propositions for each state (possible world), and a relation $\mathcal{K}$ that given an agent and a state gives a set of states. This set is to be understood as the states the agent considers possible given the information available in the input state. We should mention the type variables $('i,\ 's)$. The type $'i$ is again an arbitrary type for agents while $'s$ is the type of states. Not requiring a specific type of possible worlds ensures that the formalization is generic.

The double turnstile, $M,\ s \models \varphi$, denotes the semantics of a formula $\varphi \in \mathcal{L}$ under a Kripke structure M and state s. We formalize it as the following function:

```
primrec semantics :: ‹('i, 's) kripke ⇒ 's ⇒ 'i fm ⇒ bool›
  (‹_, _ ⊨ _› [50,50] 50) where
  ‹(_, _ ⊨ ⊥) = False›
| ‹(M, s ⊨ Pro i) = π M s i›
| ‹(M, s ⊨ (p ∨ q)) = ((M, s ⊨ p) ∨ (M, s ⊨ q))›
| ‹(M, s ⊨ (p ∧ q)) = ((M, s ⊨ p) ∧ (M, s ⊨ q))›
| ‹(M, s ⊨ (p ⟶ q)) = ((M, s ⊨ p) ⟶ (M, s ⊨ q))›
| ‹(M, s ⊨ K i p) = (∀ t ∈ 𝒦 M i s. M, t ⊨ p)›
```

No combination of model and state satisfies $\perp$. The logical operators are defined by recursively obtaining the semantics of each subformula and combining the Boolean values through the built-in operators in Isabelle/HOL. Two cases remain: the case for a proposition i looks up and returns the truth value of s and i in $\pi\ M$ (the latter gives the π of the Kripke structure M). Lastly, we have the case for a modal operator $K_i\ p$ which requires the semantics of p to be true in every state agent i considers possible (from the current state).

With the semantics in place, we can prove various interesting properties of the modal operator K_i, say, (the proof is omitted in the present paper):

theorem *distribution*: ‹$M, s \models (K\ i\ p \land K\ i\ (p \longrightarrow q) \longrightarrow K\ i\ q)$›

The above states that the operator K_i distributes over implication.

3 AXIOM SYSTEM K_n

The distribution theorem can be recognized in the very compact axiomatic system K_n. We adopt the usual syntax that the provability of a formula $\varphi \in \mathcal{L}$ is denoted by the turnstile symbol: $\vdash \varphi$. The system is inductively defined as follows:

inductive *SystemK* :: ‹$'i\ fm \Rightarrow bool$› ($\vdash$ - [*50*] *50*) **where**
A1: ‹$tautology\ p \Longrightarrow \vdash p$›
| *A2*: ‹$\vdash (K\ i\ p \land K\ i\ (p \longrightarrow q) \longrightarrow K\ i\ q)$›
| *R1*: ‹$\vdash p \Longrightarrow \vdash (p \longrightarrow q) \Longrightarrow \vdash q$›
| *R2*: ‹$\vdash p \Longrightarrow \vdash K\ i\ p$›

$A1$ states that any classical propositional tautology is provable, $A2$ is similar to the distribution theorem, $R1$ is simply modus ponens and $R2$ states that agents also know the provable formulas. The definition *tautology* in $A1$ relies on a semantics that treats modal formulas $K_i\varphi$ as if they were propositional symbols. This is the semantic equivalent of allowing all substitution instances of propositional tautologies, but is simpler to formalize.

4 SOUNDNESS

For the axiom system K to be sound, every formula in $\mathcal{L}$ provable in system K_n must be valid with respect to the semantics:

$$\forall \varphi \in \mathcal{L}.\ \vdash \varphi \longrightarrow (\forall M,\ s.\ M,\ s \models \varphi)$$

That is, no combination of proof rules leads to a formula that is not valid. It does not follow that all valid formulas are provable, however, which is why we also need completeness.

Our formalized proof of soundness requires extra work for the rule $A1$. The following theorem states soundness for this rule:

theorem *tautology*: ‹$tautology\ p \Longrightarrow M, s \models p$›

Note that the quantification $p \in \mathcal{L}$ and $\forall M\, s$ is implicit in Isabelle/HOL. The proof is omitted in the present paper.

Proving soundness for system K_n is now straightforward. The following theorem captures the soundness property for system K_n:

theorem *soundness*: ‹$\vdash p \Longrightarrow M, s \models p$›
by (*induct p arbitrary*: *s rule*: *SystemK.induct*) (*simp-all add*: *tautology*)

The proof strategy is to apply induction over the rules of the system. Once we supply the *tautology* theorem, the simplification proof method in Isabelle/HOL can easily solve each subgoal.

5 COMPLETENESS

We now want to demonstrate that system K_n is not only sound, but also complete, namely that every valid formula in $\mathcal{L}$ is provable:

$$\forall \varphi \in \mathcal{L}.\ (\forall M,\ s.\ M,\ s \models \varphi) \longrightarrow \vdash \varphi$$

The formalized proof follows Hagin et al. [3] and builds on maximal consistent sets of formulas. A formula φ is K_n-consistent if its negation is not provable: $\nvdash \neg\varphi$. A finite set of formulas $\varphi_1, \ldots, \varphi_n$ is K_n-consistent if we cannot prove that they imply a contradiction: $\nvdash \varphi_1 \longrightarrow \ldots \longrightarrow \varphi_n \longrightarrow \bot$. Finally, an infinite set of formulas is K_n-consistent if all its finite subsets are.

Instead of working directly with this definition, we start from Fitting's consistency properties [1], which define the class C of consistent sets S syntactically:

definition *consistency* :: ‹$'i\ fm\ set\ set \Rightarrow bool$› **where**
‹$consistency\ C \equiv \forall S \in C.$
$(\forall p.\ \neg\ (Pro\ p \in S \land (\neg\ Pro\ p) \in S)) \land$
$\bot \notin S \land$
$(\forall Z.\ (\neg\ (\neg\ Z)) \in S \longrightarrow S \cup \{Z\} \in C) \land$
$(\forall A\ B.\ (A \land B) \in S \longrightarrow S \cup \{A, B\} \in C) \land$
$(\forall A\ B.\ (\neg\ (A \lor B)) \in S \longrightarrow S \cup \{\neg\ A, \neg\ B\} \in C) \land$
$(\forall A\ B.\ (A \lor B) \in S \longrightarrow S \cup \{A\} \in C \lor S \cup \{B\} \in C) \land$
$(\forall A\ B.\ (\neg\ (A \land B)) \in S \longrightarrow S \cup \{\neg\ A\} \in C \lor S \cup \{\neg\ B\} \in C) \land$
$(\forall A\ B.\ (A \longrightarrow B) \in S \longrightarrow S \cup \{\neg\ A\} \in C \lor S \cup \{B\} \in C) \land$
$(\forall A\ B.\ (\neg\ (A \longrightarrow B)) \in S \longrightarrow S \cup \{A, \neg\ B\} \in C) \land$
$(\forall A.\ tautology\ A \longrightarrow S \cup \{A\} \in C) \land$
$(\forall A\ i.\ \neg\ (K\ i\ A \in S \land (\neg\ K\ i\ A) \in S))$›

All but the last two conditions are standard and ensure downwards saturation [12] of each set: the satisfiability of any member is guaranteed by conditions on its subformulas, and consistency is ensured at the bottom. The penultimate line ensures that the consistent sets contain all tautologies. This is a technical trick that makes them easier to work with. Similarly, the last condition ensures that no agent both knows and does not know the same formula.

We connect the definition of consistency to provability in system K_n at a later stage through the following theorem:

theorem *K-consistency*: ‹$consistency\ \{set\ G \mid G.\ \neg \vdash imply\ G\ \bot\}$›

The completeness proof follows the usual recipe: (i) assume a valid formula φ has no derivation (ii) then its negation is K_n-consistent and (iii) we can extend the set $\{\neg\varphi\}$ in a standard way to a maximally consistent set [3] which (iv) has a model, contradicting the validity assumption. The model existence rests on four facts outlined by Fagin et al. [3]. Unfortunately we do not have space to cover the formalization here. The completeness theorem is:

theorem *completeness*:
assumes ‹$\forall\ (M :: ('i :: countable, 'i\ fm\ set)\ kripke)\ s.\ M, s \models p$›
shows ‹$\vdash p$›

6 CONCLUDING REMARKS

System K_n provides a concise way of reasoning about the knowledge of agents. To trust such reasoning we need to know that the system is sound and thus only proves valid formulas. Moreover, if we want to use the system in practice, we would like to know that if we cannot prove a formula, then it is not due to a limitation of the proof system but because the formula is incorrect: we want completeness. To prove these properties, we have given precise specifications of the syntax and semantics of an epistemic logic for countably many agents. The proofs are mechanically checked allowing us to fully trust the axiom system. In adapting Fitting's [4] consistency properties from first-order to epistemic logic, we have shown another application of these. More generally, the work is an example of a synthetic completeness proof, a technique we have also used in other formalizations [6, 7].

REFERENCES

[1] Stefan Berghofer. 2007. First-Order Logic According to Fitting. *Archive of Formal Proofs* (2007). https://isa-afp.org/entries/FOL-Fitting.html, Formal proof development.

[2] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. 2008. *Dynamic Epistemic Logic*. Springer.

[3] Ronald Fagin, Joseph Y. Halpern, Moshe Y. Vardi, and Yoram Moses. 1995. *Reasoning about Knowledge*. MIT Press.

[4] Melvin Fitting. 1996. *First-Order Logic and Automated Theorem Proving, Second Edition*. Springer.

[5] Asta Halkjær From. 2018. Epistemic Logic. *Archive of Formal Proofs* (2018). https://isa-afp.org/entries/Epistemic_Logic.html, Formal proof development.

[6] Asta Halkjær From. 2020. Formalizing Henkin-Style Completeness of an Axiomatic System for Propositional Logic. In *Proceedings of the ESSLLI & WeSSLLI Student Session 2020*, Alexandra Pavlova (Ed.).

[7] Asta Halkjær From, Patrick Blackburn, and Jørgen Villadsen. 2020. Formalizing a Seligman-Style Tableau System for Hybrid Logic - (Short Paper). In *Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12166)*, Nicolas Peltier and Viorica Sofronie-Stokkermans (Eds.). Springer, 474–481. https://doi.org/10.1007/978-3-030-51074-9_27

[8] Alexander B. Jensen. 2021. Towards Verifying GOAL Agents in Isabelle/HOL. In *ICAART 2021 – Proceedings of the 13th International Conference on Agents and Artificial Intelligence – Volume 1*. SciTePress, 345–352.

[9] Alexander B. Jensen, Koen V. Hindriks, and Jørgen Villadsen. 2021. On Using Theorem Proving for Cognitive Agent-Oriented Programming. In *ICAART 2021 – Proceedings of the 13th International Conference on Agents and Artificial Intelligence – Volume 1*. SciTePress, 446–453.

[10] Jakub Kądziołka. 2021. Solution to the xkcd Blue Eyes puzzle. *Archive of Formal Proofs* (2021). https://isa-afp.org/entries/Blue_Eyes.html, Formal proof development.

[11] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. 2002. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*. LNCS, Vol. 2283. Springer.

[12] Raymond M. Smullyan. 1968. *First-Order Logic*. Springer-Verlag.

[13] Zuojun Xiong, Thomas Ågotnes, and Yuzhi Zhang. 2020. The Logic of Secrets. In *LAMAS 2020 - 10th Workshop on Logical Aspects of Multi-Agent Systems*.