The Coinductive Formulation of Common Knowledge

Colm Baston[1] and Venanzio Capretta[2]

1 **Functional Programming Lab, School of Computer Science, University of Nottingham, UK**
colm.baston@nottingham.ac.uk

2 **Functional Programming Lab, School of Computer Science, University of Nottingham, UK**
venanzio.capretta@nottingham.ac.uk

Abstract

Common knowledge is a modality in epistemic logic: a group of agents has common knowledge of an event if they all know it and this knowledge is recursively known by everyone. It seems natural to see it as a coinductive operator: common knowledge of an event means that everyone knows it, everyone knows that everyone knows it, everyone knows that everyone knows that everyone knows it, and so on ad infinitum. We define it in a type-theoretic setting, with formalisation in the proof assistants Agda and Coq, and prove that it is equivalent to the traditional characterisation.

In the semantics of epistemic logic, a propositional formula is interpreted as an event, that is, a set of states or possible worlds (those in which the formula holds). Knowledge is represented as an equivalence relation on states: two states are equivalent if an agent can't distinguish them on the basis of individual information. Common knowledge defined in terms of the transitive closure of the union of the individual knowledge relations.

We prove that our approach is equivalent to the traditional one in two ways.

Firstly, our way of talking about knowledge as an operator on events satisfying some properties (essentially a semantic version of the modal logic S5) is equivalent to *frame semantics*, the traditional approach using equivalence relations.

Secondly, we prove that our coinductive definition of common knowledge is equivalent to the one using transitive closure of the equivalence relations for each agent.

These results have been formalised in the two proof assistants Agda and Coq.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases common knowledge, coinductive types, epistemic logic, type theory

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

Here is a version of a famous puzzle that illustrates the notion of common knowledge (traditional versions are known as *the muddy children* or *the cheating husbands* problem [6, 7]).

A group of students of philosophical logic is given a test to see if they deserve to get a degree. A coloured dot, either green or red, is painted on each student's forehead. Then they are gathered together in a room. Each can see everyone else's dot, but not their own.

The researcher tells them: "Your task is to deduce, using pure logic, the colour of the dot on your own forehead. Each of you gets two hats, one red and one green. You will leave the room and go to separate places. There can be no communication, verbal or otherwise, between you outside this room. If you can deduce that the dot on your forehead is green,

wear the green hat. If you deduce that your dot is red, wear the red hat. If it is impossible for you to determine the colour of your dot, don't wear a hat. Then you will all come back to this room and you will be able to see the other students' hat choices. If some students will have made the correct hat choice, they will graduate. Those who make a mistaken choice will be expelled from the school. If you don't put on a hat, then you will be allowed to try again: you can leave the room and have the chance to make a hat choice and come back. By the way, I want you to know that at least one of you has a green dot." Assuming that all the students are skilled in precise logic, what will happen?

Here's what happens if there are n students with a green dot on their forehead: None of the students will wear a hat for the first $n-1$ times when they reenter the room; on the nth time, all n green-dotted students will wear a green hat. (The red-dot students can't draw any conclusion until they see the green-dotted ones wearing their hats.) We show this through a proof by induction on n, which must be at least 1 because of the information that the researcher gave.

Proof. Suppose there is only one student with a green dot, call her Alice for convenience. She can see on the initial gathering that all the others have red dots. Since the researcher told them that at least one of them has a green dot, Alice can deduce that she does and will wear a green hat at the first reentry.

Suppose now that there are two green-dotted students, call them Alice and Bob. Alice can see that Bob has a green dot and everyone else has a red dot. She herself, to her knowledge, may have either a green or a red dot. For the moment she has no sufficient information to make a decision, so she won't wear a hat at the first reentry. Bob will do the same. When she sees that Bob hasn't put on a hat, Alice thinks: "if my dot were red, Bob would be the only student with a green dot, he would have made the reasoning from the first step and would have put on a green hat. If he didn't, then he must see that I have a green dot". So she wears a green hat on the second reentry. Bob makes the same reasoning and does the same.

Now suppose we proved the statement if there are n green-dotted students. If there are $n+1$ of them, one of them being Alice, she can reason in the following way. She can see that there are at least n green-dotted students. So she doesn't wear a hat for the first n reentries: she waits to see what the others do on the nth reentry. The other green-dotted students do the same: none of them wears a hat on the first n reentries. At that point Alice can deduce, seeing that nobody has put on a hat yet, that she must also have a green dot. So she wears a green hat at the $(n+1)$th reentry. ◀

In this puzzle, the students will have been employing some form of epistemic logic, reasoning about the knowledge of their fellows. In particular, when the researcher tells them that at least one of them has a green dot on their forehead, this fact becomes common knowledge. For cases where $n > 1$, it might seem that the researcher's statement gives the students no additional information: after all, they can each see at least one green-dotted student and derive this information for themselves. The additional information in such cases comes in the form of advancing the student's knowledge about the other students' knowledge.

If there are 2 green-dotted students, each knows that there is at least one green-dotted student before the researcher's statement; let us call this first-degree knowledge. Neither student, however, knows that the other knows this until after the statement; let us call this second-degree knowledge. In general, for n green-dotted students, nth-degree knowledge is required for them to deduce the colour of their dots on the nth reentry, but only $(n-1)$th-degree knowledge holds before the researcher's statement [6].

Intuitively, common knowledge of a fact means that everyone knows it, everyone knows that everyone knows it, everyone knows that everyone knows that everyone knows it, and so on ad infinitum. A statement about common knowledge therefore implies an infinite amount of information. Type-theoretical logical systems allow for the direct definition of coinductive types which may contain infinite objects, constructed by guarded corecursion. Interpreted via the Curry-Howard correspondence, coinductive types are propositions whose proofs may be infinite, by coinduction. Naturally then, it would seem that a coinductive type be an ideal mechanism through which we could encode common knowledge.

Common knowledge is traditionally realised within the context of a possible world semantics, however, introduced to epistemic logic by Hintikka [9]. There is said to be a set of possible worlds, or states, in which each propositional variable is assigned a truth value: a proposition may be true in one state, but false in another. An agent's knowledge of a proposition is itself a proposition determined by the state, so one may reason about an agent's knowledge of another agent's knowledge to any arbitrary degree. If an agent knows a proposition to be true, then that proposition must be true in every state that is compatible with their knowledge.

More recent formalisations typically use the relational semantics of a normal modal logic, after Kripke [11]. In such formulations, each agent is equipped with an accessibility relation on states, and a modal operator for an agent's knowledge is defined in terms of that agent's accessibility relation. A modal operator for common knowledge can then be defined in terms of its own relation, formed by combining the accessibility relations for all agents by union and transitive closure.

Capretta, however, formalised common knowledge within a modal logic, embedded in the logic of the Coq proof assistant, as coinductive predicate [5]. He showed that his definition was sufficient to prove Aumann's theorem about common knowledge of rationality in perfect information games [1]. That is, if the players of a perfect information game are rational, each playing to maximise their own payoff, and their rationality is common knowledge, then the game must play out in a backwards induction equilibrium. In this paper, we show that this coinductive formulation of common knowledge is equivalent to the traditional relational formulation.

Previous work on a Coq formalisation of common knowledge in Coq was done by Lescanne [12] using a complementary approach: in that work, Coq is used as a logical framework and epistemic logic is implemented as an object language. Coq can then be used as a metatheory to experiment with the target logic. So Lescanne is using a *deep embedding*, while we use a *shallow embedding*.

In Section 2, we present a shallow embedding of epistemic logic within the logic of a type theory. In Section 3, we define what it means to be a knowledge operator in this setting, and we introduce the coinductive formulation of common knowledge. In Section 4, we show that an operator defined in terms of the traditional relational semantics of epistemic logic satisfies our definition of a knowledge operator. In Section 5, we show how common knowledge is defined in terms of the relational semantics, and prove that it is equivalent to our coinductive definition.

2 Possible Worlds Semantics and Events

The modal logic S5, introduced by Lewis and Langford [13] (modern presentations are in [3, 6]), is a propositional logic enriched with a modal operator $\Box$, often interpreted to mean "it is necessary that", and rules for reasoning with $\Box$. The properties traditionally

associated with knowledge are characterised by S5. In epistemic logic, $\Box$ is written as K and is interpreted epistemically, that is, K is read "it is known that". If we want to reason about the knowledge of multiple agents, we can extend S5 by introducing multiple modal operators, each written K_a for some agent a and read "a knows that". S5 provides an idealised model for knowledge: in short, its properties state that agents are perfect reasoners, can only know things which are actually true, and are aware of what they do and do not know.

Our formalisation of epistemic logic, however, is not axiomatic, but definitional. Instead of postulating a set of axioms for a knowledge modality, we define it using the logic of a proof assistant like Agda or Coq, along the semantic ideas of possible-worlds models. In other words, we work with a shallow embedding of epistemic logic in type theory. (The *deep* and *shallow embedding* approaches to the formalisation of logics and domain-specific languages are well-known and widespread. The first exposition of the concepts, but not the terminology, that we can find is by Reynolds [14]. See, for example, [10] for a clear explanation.)

We postulate a set State of possible worlds, which we call states. A state encodes all relevant information about the world we are modelling. In the semantics of epistemic logic, a proposition is interpreted as a set of states, called an *event*. We shall not talk about propositions in this way, to avoid confusion with propositions as types (the Curry-Howard correspondence) that is standard in type theory. Propositions are identified with the type of their proofs. In Agda, the universe Set of small types is used also for propositions. In Coq, there is a universe Prop of propositional types.

As an example, imagine that we are modelling a coin toss. The statement "the coin landed heads side up" is an event: it might be true in some of the states, but false in others. Events are therefore predicates on states, that is, functions from states to propositions/sets: $\mathsf{Event} = \mathsf{State} \to \mathsf{Set}$.

An event can also be seen extensionally as the set of states in which that event occurs, or is true. It is convenient to define set-like operators to combine events and make logical statements about them. In the following, the variable e ranges over events and the variable w ranges over states. We obtain the truth assignment of an event e in a state w by simply applying the event to the state, written $e\,w$.

$$
\begin{aligned}
&\_ \sqcap \_ : \mathsf{Event} \to \mathsf{Event} \to \mathsf{Event}\\
&e_1 \sqcap e_2 = \lambda w.e_1\,w \wedge e_2\,w\\
&\\
&\_ \sqcup \_ : \mathsf{Event} \to \mathsf{Event} \to \mathsf{Event}\\
&e_1 \sqcup e_2 = \lambda w.e_1\,w \vee e_2\,w\\
&\\
&\_ \sqsubset \_ : \mathsf{Event} \to \mathsf{Event} \to \mathsf{Event}\\
&e_1 \sqsubset e_2 = \lambda w.e_1\,w \to e_2\,w\\
&\\
&\sim \_ : \mathsf{Event} \to \mathsf{Event}\\
&\sim e = \lambda w.\neg(e\,w)
\end{aligned}
\qquad
\begin{aligned}
&\_ \subset \_ : \mathsf{Event} \to \mathsf{Event} \to \mathsf{Set}\\
&e_1 \subset e_2 = \forall w.e_1\,w \to e_2\,w\\
&\\
&\_ \equiv \_ : \mathsf{Event} \to \mathsf{Event} \to \mathsf{Set}\\
&e_1 \equiv e_2 = (e_1 \subset e_2) \wedge (e_2 \subset e_1)\\
&\\
&\Vdash : \mathsf{Event} \to \mathsf{Set}\\
&\Vdash e = \forall w.e\,w
\end{aligned}
$$

The first three operators, $\sqcap$, $\sqcup$, and $\sqsubset$, are binary operations on events: they map two events to the event that is their conjunction, disjunction, and implication, respectively; we can see them set-theoretically as intersection, union, and exponent of sets of states. The fourth, $\sim$, is a unary operator expressing event negation, set theoretically it is the complement.

The next two operators are logical statements between two events. The first of the two, $\subset$, states that the first event logically implies the second, set-theoretically the first is a subset of the second. The next, $\equiv$, states that two events are logically equivalent, their set extensions being equal. Finally, the operator $\Vdash$ expresses the fact that an event is true in all states:

that it is semantically forced to be true. We won't say that it is necessarily true, to avoid confusion with the necessity modality, which in our case is interpreted as knowledge. On the logical system side, it corresponds to a tautology.

As a simple example, imagine a setup in which a coin has been tossed and a six-sided die has been rolled. We have these primitive events:

$$\begin{aligned} C_H &= \text{“the coin landed heads side up”} \\ C_T &= \text{“the coin landed tails side up”} \end{aligned} \qquad \begin{aligned} D_1 &= \text{“the die rolled a 1”} \\ D_2 &= \text{“the die rolled a 2”} \\ &\vdots \\ D_6 &= \text{“the die rolled a 6”} \end{aligned}$$

Then, for example, $D_3 \sqcup D_4$ is the event which is true in those states where the die rolled a 3 or a 4, we might assume $\Box\,(\sim(C_H \sqcap C_T))$ so that there cannot be a state in which the coin landed both heads side up and tails side up, and so on.

Now we come to introducing modal operators for knowledge, so let's introduce two agents in our example, Alice and Bob. Each is given a modal knowledge operator K that maps events to events. We can now also express events such as the following:

$$\begin{aligned} \mathsf{K}_A\, D_1 &= \text{“Alice knows that the die rolled a 1”} \\ \mathsf{K}_B\,(\mathsf{K}_A\, D_2) &= \text{“Bob knows that Alice knows that the die rolled a 2”} \\ \mathsf{K}_A\,(C_H \sqsubset D_6) &= \text{“Alice knows if the coin landed heads side up, then the die rolled a 6”} \\ \sim(\mathsf{K}_B\, C_H \sqcup \mathsf{K}_B\, C_T) &= \text{“Bob doesn't know on which side the coin landed”} \end{aligned}$$

In general, we postulate a non-empty set Agent of agents and a knowledge operator K_a for each agent a:

$$\mathsf{K}_a : \mathsf{Event} \to \mathsf{Event}$$

To be a proper knowledge modality, each operator K_a must satisfy properties corresponding to the axioms of modal operators in the classic logical system S5.

[**Colm:** this section seems to end abruptly]

3 Knowledge Operators and Common Knowledge

The traditional semantics of knowledge modalities is as a relation on states. In our work, we assume directly knowledge operators on events. In this section we fix an operator $\mathsf{K} : \mathsf{Event} \to \mathsf{Event}$ and we specify a set of properties that it must satisfy to be a possible interpretation of the knowledge of an agent.

It must satisfy the semantic correlates of the properties of S5. We discovered that an extra infinitary deduction rule is required to obtain a perfect correspondence with the relational interpretation.

We extend the semantic implication operator $\subset$ to families of events.

▶ **Definition 1.** A *family of events* indexed on a type X is a function $E : X \to \mathsf{Event}$. Given a family E and a single event e, we say that E *semantically entails* e if e is true in every state in which all the members of the family E are true:

$$E \subset e = \forall w.(\forall x : X.E\,x\,w) \to e\,w.$$

Entailment and equivalence are generalised to two families $E_1 : X_1 \to \mathsf{Event}$ and $E_2 : X_2 \to \mathsf{Event}$:

$$E_1 \subset E_2 = \forall x : X_2.E_1 \subset (E_2\,x); \qquad E_1 \equiv E_2 = (E_1 \subset E_2) \wedge (E_2 \subset E_1).$$

We can map K on the whole family by applying it to every member: We write $\mathsf{K}\,E$ for the family $\lambda x.\mathsf{K}\,(E\,x)$.

▶ **Definition 2.** We say that K *preserves semantic entailment* if, for every family E of events and every event e, we have:

$$E \subset e \to \mathsf{K}\,E \subset \mathsf{K}\,e.$$

We require that K has this property and also satisfies the properties of S5. Some of these are derivable from semantic entailment, but we formulate them all in the definition to clarify the relation with traditional epistemic logic.

▶ **Definition 3.** An operator on events, $\mathsf{K} : \mathsf{Event} \to \mathsf{Event}$, is called a *knowledge operator* if it preserves semantic entailment and satisfies the event-based version of the properties of S5:

1. $\Perp\!\!\!\Perp\, e \to \Perp\!\!\!\Perp\, \mathsf{K}\,e$
   This principle is known as *knowledge generalisation* (or *necessitation* in modal logics interpreting $\Box$ to mean "it is necessary that"). It states that all derivable theorems are known, that is, the agent is capable of applying pure logic to derive tautologies. Here we work on the semantics side: instead of logical formulas, the objects of the knowledge operators are events, that is, predicates on states. We understand an event to be a tautology if it is true in every state. The unfolding of the principle is:
   $$(\forall w.e\,w) \to \forall v.\mathsf{K}\,e\,v$$
2. $\mathsf{K}\,(e_1 \sqsubset e_2) \subset (\mathsf{K}\,e_1) \sqsubset (\mathsf{K}\,e_2)$
   Corresponding to *Axiom K*, this states that the knowledge operator distributes over implication: The agent is capable of applying *modus ponens* to what they know. Notice the use of the two operators $\sqsubset$, mapping two events to the event expressing the implication between the two, and $\subset$, stating that the second event is true whenever the first one is (they are related by $e_1 \subset e_2 \leftrightarrow \Perp\!\!\!\Perp\,(e_1 \sqsubset e_2)$). If we unfold the definitions, this states that:
   $$\forall w.\mathsf{K}\,(e_1 \sqsubset e_2)\,w \to \mathsf{K}\,e_1\,w \to \mathsf{K}_a\,e_2\,w$$
   That is, if in a state w the agent knows that e_1 implies e_2 and also knows e_1, then they know e_2.
3. $\mathsf{K}\,e \subset e$
   Corresponding to *Axiom T*, this states that knowledge is true: what distinguishes knowledge from belief or opinion is that when an agent knows an event, that event must actually hold in the present state.
4. $\mathsf{K}\,e \subset \mathsf{K}\,(\mathsf{K}\,e)$
   Corresponding to *Axiom 4*, this is a principle of *self-awareness* of knowledge: agents know when they know something.
5. $\sim\mathsf{K}\,e \subset \mathsf{K}\,(\sim\mathsf{K}\,e)$
   Corresponding to *Axiom 5*, this negative version of the principle of *self-awareness* could be called the *Socratic Principle*: When an agent doesn't know something, they at least know that they don't know it.

▶ **Lemma 4.** *The first two properties in the definition of knowledge operator (knowledge generalisation and Axiom K) are consequences of preservation of semantic entailment.*

Proof. Assume that K preserves semantic entailment.

- Knowledge generalisation is immediate once we see that $\Perp\!\!\!\Perp\, e$ is equivalent to the semantic entailment from the empty family: $\varnothing \subset e$.

- Axiom K follows from applying preservation of the semantic entailment version of modus ponens (using a family with just two elements): $\{e_1 \sqsubset e_2, e_1\} \subset e_2$.

(We have used set notation for the families: they indicate the trivial family indexed on the empty type and a family indexed on the Booleans.) ◀

Let us now return to a setting with a non-empty set Agent, each member of which has their own knowledge operator K_a satisfying Definition 3. Recall that common knowledge of an event intuitively means that everyone knows it, everyone knows that everyone knows it, everyone knows that everyone knows that everyone knows it, and so on ad infinitum.

We define EK to be an operator expressing that "everyone knows" an event:

$$\mathsf{EK} : \mathsf{Event} \to \mathsf{Event}$$
$$\mathsf{EK}\, e = \lambda w.\forall a.\mathsf{K}_a\, e\, w$$

Common knowledge of an event e is then equivalent to the infinite conjunction:

$$\mathsf{EK}\, e \sqcap \mathsf{EK}\,(\mathsf{EK}\, e) \sqcap \mathsf{EK}\,(\mathsf{EK}\,(\mathsf{EK}\, e)) \sqcap \ldots$$

This infinite conjunction can be expressed by a *coinductive definition* saying that common knowledge of e means the conjunction of $\mathsf{EK}\, e$ and, recursively, common knowledge of $\mathsf{EK}\, e$. In Agda or Coq, this can be defined directly by a coinductive operator:

$$\mathsf{CoInductive\ cCK} : \mathsf{Event} \to \mathsf{Event}$$
$$\mathsf{cCK{-}intro} : \forall e.\mathsf{EK}\, e \sqcap \mathsf{cCK}\,(\mathsf{EK}\, e) \subset \mathsf{cCK}\, e$$

This defines common knowledge at a high level without mentioning states at all in the definition. If we unfold the definitions so that we can see the constructor's type in full, it becomes evident that the definition satisfies the positivity condition of (co)inductive types.

$$\mathsf{cCK{-}intro} : \forall e.\forall w.(\mathsf{EK}\, e\, w) \wedge (\mathsf{cCK}\,(\mathsf{EK}\, e)\, w) \to \mathsf{cCK}\, e\, w$$

A proof of $\mathsf{cCK}\, e$ must be constructed from a proof of $\mathsf{EK}\, e \sqcap \mathsf{cCK}\,(\mathsf{EK}\, e)$, so we can derive either conjunct if we have that e is common knowledge. That is, we have the following trivial properties:

▶ **Lemma 5.** *For every event e we have:* $\mathsf{cCK}\, e \subset \mathsf{EK}\, e$.

▶ **Lemma 6.** *For every event e we have:* $\mathsf{cCK}\, e \subset \mathsf{cCK}\,(\mathsf{EK}\, e)$.

Our definition of common knowledge has two convenient aspects. First, it naturally corresponds to the informal recursive notion. Second, it allows us to prove propositions by the method of *guarded corecursion*. (See, for an introduction, Chapter 13 of the Coq book by Bertot and Casteran [2] or the application to general recursion by one of us [4].)

The idea is that when proving that an event e is common knowledge, we must prove $\mathsf{EK}\, e$ without any extra assumption, but we can recursively use the statement that we're proving in the derivation of the common knowledge of $\mathsf{cCK}\,(\mathsf{EK}\, e)$. This seems like circular reasoning, assuming the statement we want to prove in its own proof, but it is validated by the fact that it correctly unfolds into an infinite proof that doesn't need such circular assumption.

We refer the reader to the literature on coinduction for a formal treatment of guardedness in corecursive reasoning. Here, we illustrate it with a first simple example, showing that common knowledge is equivalent to the family of events expressing finite iterations of EK:

$$\mathsf{recEK} : \mathsf{Event} \to \mathbb{N} \to \mathsf{Event}$$
$$\mathsf{recEK}\, e\, 0 = \mathsf{EK}\, e$$
$$\mathsf{recEK}\, e\,(n+1) = \mathsf{EK}\,(\mathsf{recEK}\, e\, n)$$

▶ **Theorem 7.** *For every event e, the family* $\mathsf{recEK}\, e$ *semantically entails* $\mathsf{cCK}\, e$*:*

$$\mathsf{recEK}\, e \subset \mathsf{cCK}\, e.$$

Proof. In a coinductive proof, we are allowed to assume the statement we're proving and use it in a restricted way:

Assumption CH: $\forall e.\mathsf{recEK}\, e \subset \mathsf{cCK}\, e$.

Obviously, we're not allowed to just use Assumption CH to prove the theorem. We must make at least one step in the proof without circularity.

Unfolding the statement, we need to prove that for every state w we have:

$$(\forall n : \mathbb{N}.\mathsf{recEK}\, e\, n\, w) \to \mathsf{cCK}\, e\, w.$$

So let's assume that for every natural number n, $\mathsf{recEK}\, e\, n\, w$ holds.

We must now prove $\mathsf{cCK}\, e\, w$, which can be derived using the constructor $\mathsf{cCK-intro}$ from $\mathsf{EK}\, e\, w$ and $\mathsf{cCK}\, (\mathsf{EK}\, e)\, w$.

- $\mathsf{EK}\, e\, w$ is just $\mathsf{recEK}\, e\, 0\, w$, which is true by assumption;
- To prove $\mathsf{cCK}\, (\mathsf{EK}\, e)\, w$, we invoke Assumption CH, instantiated for the event $\mathsf{EK}\, e$:

  $$\mathsf{recEK}\, (\mathsf{EK}\, e) \subset \mathsf{cCK}\, (\mathsf{EK}\, e)$$

  That is:

  $$\forall w.(\forall n.\mathsf{recEK}\, (\mathsf{EK}\, e)\, n\, w) \to \mathsf{EK}\, e\, w$$

  So we need to prove that for every n, $\mathsf{recEK}\, (\mathsf{EK}\, e)\, n\, w$. This is trivially equivalent to $\mathsf{recEK}\, e\, (n+1)\, w$, which is true by assumption. Therefore, Assumption CH allows us to conclude $\mathsf{EK}\, e\, w$, as desired.

◀

Let's observe the structure of this proof. We allowed ourselves to assume the statement of the theorem as an hypothesis. But it can only be used in a limited way. We used it immediately after applying the constructor $\mathsf{cCK-intro}$, to prove the recursive branch of it. This is the typical way in which *guarded corecursion* works: we can make a circular call to the object we're defining immediately under the application of the constructor.

The implication in the other direction is simply repeated unfolding of the definition of common knowledge.

▶ **Theorem 8.** *For every event e and $n : \mathbb{N}$:*

$$\mathsf{cCK}\, e \subset \mathsf{recEK}\, e\, n.$$

Proof. By induction on n:

- $\mathsf{recEK}\, e\, 0 = \mathsf{EK}\, e$ and $\mathsf{cCK}\, e \subset \mathsf{EK}\, e$ by Lemma 5.
- Assume, by inductive hypothesis, that $\forall e.\mathsf{cCK}\, e \subset \mathsf{recEK}\, e\, n$. We want to prove that $\forall e.\mathsf{cCK}\, e \subset \mathsf{recEK}\, e\, (n+1)$.

  We can instantiate the induction hypothesis for $\mathsf{EK}\, e$: $\mathsf{cCK}\, (\mathsf{EK}\, e) \subset \mathsf{recEK}\, (\mathsf{EK}\, e)\, n$.

  By Lemma 6, $\mathsf{cCK}\, e \subset \mathsf{cCK}\, (\mathsf{EK}\, e)$. From this and the induction hypothesis, by transitivity of $\subset$, we get that $\mathsf{cCK}\, e \subset \mathsf{recEK}\, (\mathsf{EK}\, e)\, n$. By definition $\mathsf{recEK}\, e\, (n+1) = \mathsf{recEK}\, (\mathsf{EK}\, e)\, n$, so we obtain $\mathsf{cCK}\, e \subset \mathsf{recEK}\, e\, (n+1)$ as desired.

◀

The equivalence of common knowledge with the family recEK gives an immediate proof of the following useful property corresponding to Axiom 4 of S5.

▶ **Lemma 9.** *For every event e, we have* $\mathsf{cCK}\, e \subset \mathsf{cCK}\,(\mathsf{cCK}\, e)$.

Finally, the coinductive definition of common knowledge satisfies the properties of knowledge operators. We must prove all the S5 properties and preservation of semantic entailment for cCK.

▶ **Theorem 10.** *Common knowledge,* cCK*, is itself a knowledge operator.*

Proof. Lemma 9 shows that Axiom 4 holds. Proofs of all other S5 properties and of preservation of semantic entailment are in the Coq formalisation. ◀

4 The Relational Semantics

In the relational semantics of epistemic logic, the agents' knowledge operators are not introduced directly, but through *accessibility relations* in the following way.

Two states may differ by a number of events, each being true in one of the states, but false in the other. If an agent has no knowledge of any of these events, only knowing events which are common to both states, then those states are indistinguishable as far as the agent is aware. We say that these states are *epistemically accessible* from one another: if the world were in one of those states, the agent would consider either state to be plausible, not having sufficient knowledge to inform them precisely in which state the world is actually in.

To say that an agent has knowledge of an event in a particular state is then to say that the event holds in all states that the agent finds epistemically accessible from that state. We formalise this notion by defining a transformation from relations on states to unary operators on events.

$$\mathsf{K}_{[\,]} : (\mathsf{State} \to \mathsf{State} \to \mathsf{Set}) \to (\mathsf{Event} \to \mathsf{Event})$$
$$\mathsf{K}_{[R]} = \lambda e.\lambda w.\forall v.w\, R\, v \to e\, v$$

Care must be taken to distinguish this notation from the notation of earlier sections where each agent a had a knowledge operator K_a directly postulated. When talking in terms of the relational semantics, we don't take these operators as primitive. Here, $\mathsf{K}_{[R]}$ refers to the operator associated with some relation $R : \mathsf{State} \to \mathsf{State} \to \mathsf{Set}$.

It is a well known result in modal logic that applying this transformation to an equivalence relation yields a knowledge operator satisfying the properties of S5 (see [8] for an extensive listing of which relation properties entail which axioms). To complete a proof that $\mathsf{K}_{[R]}$ is a knowledge operator, we have to show in addition that it preserves semantic entailment.

▶ **Lemma 11.** *For every family $E : X \to \mathsf{Event}$ and every event e, we have:*

$$E \subset e \to \mathsf{K}_{[R]}\, E \subset \mathsf{K}_{[R]}\, e.$$

Proof. Assume that $E \subset e$. To prove that $\mathsf{K}_{[R]}\, E \subset \mathsf{K}_{[R]}\, e$, we must show that for every state w in which all the members of the family $\mathsf{K}_{[R]}\, E$ are true, also $\mathsf{K}_{[R]}\, e\, w$ holds. So we assume that $\forall x.\mathsf{K}_{[R]}\,(E\, x)\, w$.

We want to prove $\mathsf{K}_{[R]}\, e\, w$, that is, by definition, $\forall v.w\, R\, v \to e\, v$.

Assume that $w\, R\, v$, we must prove $e\, v$.

By the assumption $\forall x.\mathsf{K}_{[R]}\,(E\, x)\, w$, we have that $E\, x\, v$ holds for every x. Therefore, by the assumption $\mathsf{K}_{[R]}\, E \subset \mathsf{K}_{[R]}\, e$, we conclude that $e\, v$ is true, as desired. ◀

The following lemma establishes each of the properties corresponding to the properties of S5, assuming only the needed properties of R. Proofs can be adapted from standard expositions (we have completely formalised them in Coq and Agda).

▶ **Lemma 12.** *If R is a relation on states, then the operator $\mathsf{K}_{[R]}$ has the following properties.*

- $\mathsf{K}_{[R]}$ *satisfies knowledge generalisation:* $\forall e.\Perp\, e \to \Perp\, \mathsf{K}_{[R]}\, e$
- $\mathsf{K}_{[R]}$ *satisfies Axiom K:* $\forall e_1.\forall e_2.\mathsf{K}_{[R]}\,(e_1 \sqsubset e_2) \subset \mathsf{K}_{[R]}\, e_1 \sqsubset \mathsf{K}_{[R]}\, e_2$
- *If R is reflexive, then* $\mathsf{K}_{[R]}$ *satisfies Axiom T:* $\forall e.\mathsf{K}_{[R]}\, e \subset e$
- *If R is transitive, then* $\mathsf{K}_{[R]}$ *satisfies Axiom 4:* $\forall e.\mathsf{K}_{[R]}\, e \subset \mathsf{K}_{[R]}\,(\mathsf{K}_{[R]}\, e)$
- *If R symmetric and transitive, then* $\mathsf{K}_{[R]}$ *satisfies Axiom 5:* $\forall e. \sim \mathsf{K}_{[R]}\, e \subset \mathsf{K}_{[R]}\,(\sim \mathsf{K}_{[R]}\, e)$

Finally, we can put the two lemmas together to satisfy Definition 3.

▶ **Theorem 13.** *If R is an equivalence relation on states, then $\mathsf{K}_{[R]}$ is a knowledge operator.*

The inverse transformation, taking a modal operator on events and returning a relation on states is:

$$R_{[\,]} : (\mathsf{Event} \to \mathsf{Event}) \to (\mathsf{State} \to \mathsf{State} \to \mathsf{Set})$$
$$R_{[\mathsf{K}]} = \lambda w.\lambda v.\forall e.\mathsf{K}\, e\, w \leftrightarrow \mathsf{K}\, e\, v$$

This transformation always results in an equivalence relation, as $\leftrightarrow$ is itself an equivalence relation. In fact, if we admit classical reasoning, one direction of the implication is sufficient.

▶ **Lemma 14.** *For every pair of states w and v, if $\forall e.\mathsf{K}\, e\, w \to \mathsf{K}\, e\, v$, then $\forall e.\mathsf{K}\, e\, v \to \mathsf{K}\, e\, w$.*

Proof. Assume that $\forall e.\mathsf{K}\, e\, w \to \mathsf{K}\, e\, v$ and, for some event e, $\mathsf{K}\, e\, v$. We want to prove that $\mathsf{K}\, e\, w$.

Suppose, towards a contradiction, that $\neg\mathsf{K}\, e\, w$. By Axiom 5, which K satisfies, we have $\mathsf{K}\,(\neg\mathsf{K}\, e)\, w$. By instantiating the first assumption with $\neg\mathsf{K}\, e\, w$, we deduce that $\mathsf{K}\,(\neg\mathsf{K}\, e)\, v$. By Axiom T, this implies $\neg\mathsf{K}\, e\, v$, which contradicts the second assumption. We have reached a contradiction, so our supposition $\neg\mathsf{K}\, e\, w$ must be false. We conclude, by excluded middle, that $\mathsf{K}\, e\, w$ is true, as desired. ◀

Working with knowledge operators is equivalent to working with equivalence relations. It is known that equivalence relations give a complete semantics to S5. But to obtain an isomorphism between equivalence relations and knowledge operators, we had to add the assumption of preservation of semantic entailment.

We will show that the mapping of knowledge operators to equivalence relations and vice versa are inverse of each other. The proofs are mostly a straightforward application of equivalence and the properties of S5, except one direction, which needs semantic entailment. We give the proof of this.

In order to do this we first characterise the definition of $R_{[\mathsf{K}]}$ using event families generated by K on a fixed state w Choose as index set the set of events that are known in w: $X = \{e \mid \mathsf{K}\, e\, w\}$ (in Coq or Agda, we use the Σ dependent sum type constructor); the family itself is just application of K. Formally:

$$\mathsf{KFam}^w : (\Sigma e : \mathsf{Event}.\mathsf{K}\, e\, w) \to \mathsf{Event}$$
$$\mathsf{KFam}^w\, \langle e, h\rangle = \mathsf{K}\, e$$

Intuitively, KFam^w is the set of all the knowledge information in state w. Set-theoretically it is $\{\mathsf{K}\, e \mid e : \mathsf{Event}, \mathsf{K}\, e\, w\}$. The relation generated by K can be characterised using knowledge families.

▶ **Lemma 15.**
- *For all states w and v, $R_{[\mathsf{K}]}\, w\, v$ is equivalent to $\mathsf{KFam}^w \equiv \mathsf{KFam}^v$.*
- *For every event e and state w, $\mathsf{K}_{[R_{[\mathsf{K}]}]}\, e\, w$ is equivalent to $\mathsf{KFam}^w \subset e$.*

Proof. Just unfold the definitions. ◀

▶ **Lemma 16.** *For every knowledge operator* K *and every event* e*, we have that:*

$$\mathsf{K}_{[R_{[\mathsf{K}]}]}\, e \subset \mathsf{K}\, e.$$

Proof. Assume, for some state w, that $\mathsf{K}_{[R_{[\mathsf{K}]}]}\, e\, w$. We must prove $\mathsf{K}\, e\, w$.

By Lemma 15, the assumption is equivalent to $\mathsf{KFam}^w \subset e$. Since K preserves semantic entailment, we also have $\mathsf{K}\,\mathsf{KFam}^w \subset \mathsf{K}\, e$.

We just need to prove that all elements of the family $\mathsf{K}\,\mathsf{KFam}^w$ are true in state w, to deduce that $\mathsf{K}\, e\, w$ holds, as desired. But in fact, given and index $\langle e, h\rangle$ for the family, with h a proof of $\mathsf{K}\, e, w$, we have that $(\mathsf{K}\,\mathsf{KFam}^w)\,\langle e, h\rangle = \mathsf{K}\, e$ and this event trivially holds in w by h. ◀

The other three directions of the isomorphism are straightforward applications of the properties of knowledge operators and equivalence relations.

▶ **Theorem 17.** *For every knowledge operator* K*, every event* e *and every state* w*, we have*

$$\mathsf{K}_{[R_{[\mathsf{K}]}]}\, e\, w \leftrightarrow \mathsf{K}\, e\, w.$$

For every equivalence relation R *and every pair of states* w *and* v*, we have*

$$R_{[\mathsf{K}_{[R]}]}\, w\, v \leftrightarrow R\, w\, v.$$

5 Relational Common Knowledge and Equivalence

The isomorphism of Theorem 17 assures that the relational characterisation of knowledge is completely equivalent to the use of knowledge operators. Traditionally, common knowledge has been treated using equivalence relations. Therefore, we shall work in terms of the relational semantics from this point on.

We first equip our agents with individual knowledge operators by postulating an equivalence relation $\simeq_a : \mathsf{State} \to \mathsf{State} \to \mathsf{Set}$ for each agent a as their epistemic accessibility relation. The knowledge operator for an agent a is then $\mathsf{K}_{[\simeq_a]}$, which we shall write K_a as shorthand.

Our formulation of the "everyone knows" operator, EK, and the coinductive common knowledge operator, cCK, are as they appear in Section 3. The only difference is in the underlying definition of K_a, which had previously been taken as primitive and assumed to satisfy the knowledge operator properties outlined in Definition 3. The relations $\simeq_a$ are equivalence relations, so we can conclude that this new formulation of K_a also satisfies these properties by Theorem 13.

The relational definition of the common knowledge operator is given by its own relation: the transitive closure of the union of all accessibility relations $\simeq_a$. We write this relation as $\propto$. It is defined inductively as follows:

$$\begin{aligned}&\mathsf{Inductive}\ \_ \propto \_ : \mathsf{State} \to \mathsf{State} \to \mathsf{Set}\\ &\quad \propto\!\!-\mathsf{base} : \forall a.\forall w.\forall v.w \simeq_a v \to w \propto v\\ &\quad \propto\!\!-\mathsf{trans} : \forall w.\forall v.\forall u.w \propto v \to v \propto u \to w \propto u\end{aligned}$$

The relation $\propto$ is an equivalence relation: its transitivity is by definition, while its reflexivity and symmetry are by virtue of the reflexivity and symmetry of the underlying

accessibility relations (for reflexivity it is essential that there is at least one agent). We can intuitively grasp how it gets us to common knowledge in the following way.

If an agent a's accessibility relation $\simeq_a$ were empty, each state alone in its own equivalence class, then a would be omniscient, able to perfectly distinguish each state from all others. If a were to forget an event, then all of those states which differ only by that event would collapse into an equivalence class together. In general, the fewer equivalence classes of relation $\simeq_a$, the fewer events a will know.

The relation $\propto$ is essentially the accessibility relation characterising the union of all agents' ignorance. It is as if there were a virtual, maximally-ignorant agent whose accessibility relation is $\propto$, knowing only those events which are common knowledge among all agents and nothing more. With this in mind, we can define the relational common knowledge operator, rCK, in the same way that we defined each agent's knowledge operator and conclude that it also satisfies the knowledge operator properties:

$$\mathsf{rCK} : \mathsf{Event} \to \mathsf{Event}$$
$$\mathsf{rCK} = \mathsf{K}_{[\propto]}$$

We can verify relational common knowledge has properties corresponding to the two trivial properties of coinductive common knowledge, Lemmas 5 and 6.

▶ **Lemma 18.** *For every event e we have:* $\mathsf{rCK}\, e \subset \mathsf{EK}\, e$.

Proof. Unfolding the statement, we need to prove that for every state w we have: $(\forall v.w \propto v \to e\, v) \to \forall a.\forall u.w \simeq_a u \to e\, u$. So we assume that if we have a state v such that $w \propto v$, we can conclude that e holds in v, and we also assume we have an agent a and state u such that $w \simeq_a u$. We are left to prove that e holds in u.

By the definition of constructor $\propto\!-\mathsf{base}$, given $w \simeq_a u$, we can derive that $w \propto u$, and then by our first assumption, we have that e does hold in u. ◀

▶ **Lemma 19.** *For every event e we have:* $\mathsf{rCK}\, e \subset \mathsf{rCK}\, (\mathsf{EK}\, e)$.

Proof. Unfolding the statement, we need to prove that for every state w we have:

$$(\forall v.w \propto v \to e\, v) \to \forall u.w \propto u \to \forall a.\forall t.u \simeq_a t \to e\, t$$

As in the previous proof, we have the assumption that for any state v such that $w \propto v$, e holds in v, so to reach our conclusion $e\, t$ we can prove that $w \propto t$. We have the additional assumptions $w \propto u$ and $u \simeq_a t$.

From the latter, by $\propto\!-\mathsf{base}$ we derive $u \propto t$. Then by the transitive property of $\propto$, constructor $\propto\!-\mathsf{trans}$, we conclude $w \propto t$. ◀

With these results, we are now able to prove the first direction of the equivalence of rCK and cCK.

▶ **Lemma 20.** *For every event e we have:* $\mathsf{rCK}\, e \subset \mathsf{cCK}\, e$.

Proof. The conclusion of this theorem is an application of the coinductive predicate cCK, so we may proceed by coinduction, assuming the statement as our coinductive hypothesis. Assumption CH : $\forall e.\mathsf{rCK}\, e \subset \mathsf{cCK}\, e$

Unfolding only the application of $\subset$, we need to prove that for every state w:

$$\mathsf{rCK}\, e\, w \to \mathsf{cCK}\, e\, w$$

So we assume $\mathsf{rCK}\, e\, w$, and use constructor $\mathsf{cCK}-\mathsf{intro}$ to derive the conclusion $\mathsf{cCK}\, e\, w$, generating the proof obligations $\mathsf{EK}\, e\, w$ and $\mathsf{cCK}\, (\mathsf{EK}\, e)\, w$.

- $\mathsf{EK}\, e\, w$ comes from Lemma 18 applied to assumption $\mathsf{rCK}\, e\, w$.
- To prove $\mathsf{cCK}\, (\mathsf{EK}\, e)\, w$ we invoke assumption CH, instantiating it with event $\mathsf{EK}\, e$, leaving us to prove $\mathsf{rCK}\, (\mathsf{EK}\, e)\, w$. This is the conclusion of Lemma 19, which we can apply to assumption $\mathsf{rCK}\, e\, w$ to complete the proof.

◀

We need an additional property of cCK before we are able to complete the other direction of the equivalence proof. The property is related to the Axiom 4 property of knowledge operators, for example, for an agent a's knowledge operator K_a:

$$\forall e.\mathsf{K}_a\, e \subset \mathsf{K}_a\, (\mathsf{K}_a\, e)$$

Unfolding the $\subset$ and the outermost application of K_a in the conclusion yields the following principle.

$$\forall e.\forall w.\mathsf{K}_a\, e\, w \to \forall v.w \simeq_a v \to \mathsf{K}_a\, e\, v$$

That is, if we have that $\mathsf{K}_a\, e$ at some state w, and we also have that $w \simeq_a v$ for some state v, then we can conclude that $\mathsf{K}_a\, e$ holds at state v too. We call this *transporting* the agent's knowledge across the relation $\simeq_a$.

As rCK is defined in the same way as K_a, just for a different equivalence relation, this transportation principle applies to it too: relational common knowledge of an event can be transported from one state to another provided that those states are bridged by $\propto$. The additional property of cCK that we are to prove is that it too can be transported across $\propto$.

▶ **Lemma 21.** *For every pair of states, w and v, and event e we have:*

$$\mathsf{cCK}\, e\, w \to w \propto v \to \mathsf{cCK}\, e\, v$$

Proof. We assume $\mathsf{cCK}\, e\, w$ and proceed by induction on $w \propto v$:

- If $w \propto v$ is constructed by $\propto\!-\mathsf{base}$, then there is some agent a for whom $w \simeq_a v$ holds. We can apply Lemma 9 to our assumption $\mathsf{cCK}\, e\, w$ to obtain $\mathsf{cCK}\, (\mathsf{cCK}\, e)\, w$, and then, by Lemma 5, it follows that $\mathsf{EK}\, (\mathsf{cCK}\, e)\, w$. Since everyone knows this, it follows that a must know it: $\mathsf{K}_a\, (\mathsf{cCK}\, e)\, w$.
  We will use the transportation principle of K_a to transport $\mathsf{K}_a\, (\mathsf{cCK}\, e)$ from state w to state v as these states are bridged by $w \simeq_a v$. Then, as a knows $\mathsf{cCK}\, e$ at state v, by Axiom T it must actually hold in state v.
- If $w \propto v$ is constructed by $\propto\!-\mathsf{trans}$, then there is some state u for which $w \propto u$ and $u \propto v$ hold. We also have the induction hypotheses $\mathsf{cCK}\, e\, w \to \mathsf{cCK}\, e\, u$ and $\mathsf{cCK}\, e\, u \to \mathsf{cCK}\, e\, v$. We can simply apply the first and second induction hypotheses in turn to the assumption $\mathsf{cCK}\, e\, w$ to reach our goal.

◀

▶ **Lemma 22.** *For every event e we have:* $\mathsf{cCK}\, e \subset \mathsf{rCK}\, e$.

Proof. Unfolding the statement we are to prove, for every event e and state w:

$$\mathsf{cCK}\, e\, w \to \forall v.w \propto v \to e\, v$$

So we assume $\mathsf{cCK}\, e\, w$ and $w \propto v$. By Lemma 21 and these assumptions, we can then transport $\mathsf{cCK}\, e$ from state w to v:$\mathsf{cCK}\, e\, v$. From this, we can derive $\mathsf{EK}\, e\, v$ by Theorem 5. Since everyone knows e at state v, and our set of agents is non-empty, there must be some agent who knows e at v. Finally then, by Theorem 13, Axiom T, e must actually hold at v. ◀

In conjunction, Lemmas 20 and 22 form the full equivalence proof.

▶ **Theorem 23.** *For all events e, we have:*

$$\mathsf{rCK}\, e \equiv \mathsf{cCK}\, e \qquad \textit{that is} \qquad \mathsf{K}_{[\propto]}\, e \equiv \mathsf{cCK}\, e.$$

6 Conclusion

We presented a type-theoretic formalisation of epistemic logic and a coinductive implementation of the common knowledge operator. This was done through a shallow embedding: we formulated knowledge operators as functions on events, which are predicates on a set of possible worlds or states.

The coinductive version of common knowledge has some advantages with respect to the traditional relational version.

- It is a straightforward formulation of the intuitive definition: common knowledge of an event means that everyone knows it and the fact that everyone knows it is itself common knowledge.
- It can be formulated at a higher level, using only the knowledge operators of each agent and the connectives of epistemic logic: the coinductive definition of cCK does not mention states.
- It gives us a new reasoning tool in the form of guarded corecursion. We demonstrated its power in several proofs in this paper and in the previous work on Aumann's Theorem [5].

We proved that our coinductive formulation is equivalent to two other versions:

- The traditional one as transitive closure of the union of the accessibility relations of all agents;
- The recursive family of iterations of the "everyone knows" operator.

In the process of investigating this subject we discovered that knowledge operators obtained from equivalence relations satisfy a (to our knowledge) previously unknown property of *preservation of semantic entailment* in addition to the properties of S5. We proved that this fully characterises knowledge operators and gives an isomorphism between them and equivalence relations.

References

1. Robert J. Aumann. Backward induction and common knowledge of rationality. *Games and Economic Behavior*, 8(1):6–19, 1995. doi:http://dx.doi.org/10.1016/S0899-8256(05)80015-6.
2. Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions*. Springer, 2004.
3. Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, New York, NY, USA, 2001.
4. Venanzio Capretta. General recursion via coinductive types. *Logical Methods in Computer Science*, 1(2):1–18, 2005. doi:10.2168/LMCS-1(2:1)2005.
5. Venanzio Capretta. Common knowledge as a coinductive modality. In Erik Barendsen, Herman Geuvers, Venanzio Capretta, and Milad Niqui, editors, *Reflections on Type Theory, Lambda Calculus, and the Mind*, pages 51–61. ICIS, Faculty of Science, Radbout University Nijmegen, 2007. Essays Dedicated to Henk Barendregt on the Occasion of his 60th Birthday.
6. Ronald Fagin, Joseph Y. Halpern, Moshe Y. Vardi, and Yoram Moses. *Reasoning About Knowledge*. MIT Press, Cambridge, MA, USA, 1995.

7 George Gamow and Marvin Stern. *Puzzle Math*. Viking Press, New York, 1958.

8 James Garson. Modal logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2016 edition, 2016.

9 Jaakko Hintikka. *Knowledge and Belief*. Ithaca: Cornell University Press, 1962.

10 Chantal Keller and Benjamin Werner. Importing HOL light into coq. In Matt Kaufmann and Lawrence C. Paulson, editors, *Interactive Theorem Proving, First International Conference, ITP 2010, Edinburgh, UK, July 11-14, 2010. Proceedings*, volume 6172 of *Lecture Notes in Computer Science*, pages 307–322. Springer, 2010. doi:10.1007/978-3-642-14052-5_22.

11 Saul A. Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24(1):1–14, 03 1959.

12 Pierre Lescanne. Common knowledge logic in a higher order proof assistant. In Andrei Voronkov and Christoph Weidenbach, editors, *Programming Logics - Essays in Memory of Harald Ganzinger*, volume 7797 of *Lecture Notes in Computer Science*, pages 271–284. Springer, 2013. doi:10.1007/978-3-642-37651-1.

13 Clarence Irving Lewis and Cooper Harold Langford. *Symbolic Logic*. The Century Co., New York, 1932.

14 John C. Reynolds. *User-Defined Types and Procedural Data Structures as Complementary Approaches to Data Abstraction*, pages 309–317. Springer New York, New York, NY, 1978. doi:10.1007/978-1-4612-6315-9_22.