Knowledge and the Logic of Local Propositions

(Extended abstract)

Kai Engelhardt
Ron van der Meyden

Computing Sciences,
University of Technology, Sydney,
PO Box 123, Broadway NSW 2007
Australia
{ke,ron}@socs.uts.edu.au

Yoram Moses

9 Alexander Yanai
Tel Aviv 62382
Israel
yoram@cs.weizmann.ac.il

Abstract

An agent's limited view of the state of a distributed system may render globally different situations indistinguishable. A proposition is local for this agent whenever his view suffices to decide this proposition. Motivated by a framework for the development of distributed programs from knowledge-based specifications, we introduce a modal logic of local propositions, in which it is possible to quantify over such propositions. We show that this logic is able to represent a rich set of epistemic notions. Under the usual strong semantics, this logic is not recursively axiomatizable, however. We show that by weakening the semantics of quantification, it is possible to obtain a logic that is axiomatizable and is still able to express interesting epistemic notions.

1 Introduction

Formal approaches to modelling knowledge have received considerable attention in the second half of the Twentieth century. In the philosophical literature, numerous models and axiomatizations were suggested and attacked, mainly in the 1960's and 1970's [vW51, Hin62, Get63, Len78]. The modal logic S5, widely accepted as an appropriate characterization of necessity, has been attacked in many ways and is no longer considered by philosophers to be a good candidate for capturing knowledge. Since the 1970's formal treatments of knowledge and belief have been pursued in other fields. In economics [Aum76] and in the study of distributed computing systems [HM90, FHMV95], explicit concrete definitions for knowledge have been given and used in a variety of applications. Interestingly, the definitions in both cases are essentially equivalent, and define a notion of knowledge that satisfies S5.

Very roughly, the essence of the definition used in economics and distributed systems can be described as follows. In every world, each agent is associated with a *local state*, a specific piece of information that completely determines the agent's knowledge: In world w, the world v is possible for agent i exactly if i has the same local state in both. The underlying structure immediately yields for each agent an equivalence relation (or partition) over the set of all worlds, and the properties of S5 follow. The properties of S5 are not the goal of the definition in this case, however. Rather, the definition is independently motivated, and a notion satisfying S5 is the outcome.

Clearly, some of the properties of S5 are as objectionable in economics or distributed systems as they are for philosophers. The success of the information-based S5 definition has been mainly in applications where the problematic aspects of S5 play a negligible role. This is the case, for example, in a communication protocol where a process receiving an acknowledgement knows that the original message it sent has been received [HZ92]. In applications where computing relevant information is very complex, the assumption that knowledge is closed under deduction is clearly unreasonable. This happens in cryptographic protocols [DH76], whose correctness depends crucially on the inability of agents to compute their knowledge within reasonable time bounds.

For knowledge of facts concerning "nature" or the global state of the system, the S5 interpretation can be thought of as capturing the knowledge of an *idealized* agent. However, once we accept that agents can apply only bounded reasoning abilities, this interpretation is no longer convincing for nested knowledge statements, such as "Alice knows that Bob knows p." In a case in which p follows from Bob's information but Bob has not been able to detect this fact, it is problematic to consider even a very competent Alice as "knowing" that Bob knows p.

Our goal in this paper is to consider frameworks that facilitate the use of epistemic notions that are weaker than S5 knowledge, while retaining its information theoretic basis. We are interested in semantic notions that will satisfy the so-called *Knowledge Axiom:*

$$K_i\varphi \rightarrow \varphi,$$

without committing us to unreasonable additional properties. Specifically, we introduce and study a *logic of local propositions*, in which it is possible to quantify over propositions and over local propositions, and where there is an operator $\Box$ standing for truth in all worlds of a Kripke structure. Two variants are considered. With the standard semantics for the quantifiers, the traditional S5 notions of knowledge are expressible, and weaker notions of knowledge can be treated as well. The resulting logic is not even recursively axiomatizable, however. We next consider a weak semantics, in which bounded, rather than arbitrary, quantification is used. In this case S5 knowledge is not expressible, but once we add it we obtain a natural and simple complete axiomatization.

This work is motivated by a project we are involved in, concerning the development of a framework for top-down design of distributed protocols. In this framework, we would like to allow the designer to think in terms of knowledge, and to have flexibility in determining how sophisticated she wishes the implementations of these tests for knowledge to be. This is closely related to the notion of *knowledge-based programs* [FHMV95]. These are computer programs with statements containing explicit tests for knowledge, such as

$$\textbf{if } K_i\varphi \textbf{ then } x := 0.$$

Knowledge-based programs have been used successfully for the design and analysis of optimal protocols for a variety of problems [DM90, MT88, HMW90, NB92]. As we discuss in Section 4, however, the semantics of such programs insists on using the most sophisticated tests for knowledge. This seems suitable mainly for the design of optimal protocols. The setup we develop in this paper should, we hope, ultimately lead to a more broadly applicable notion of programs with test for knowledge. The approach we are pursuing is consistent with that promoted by Sanders, who suggested to model knowledge in terms of "sound local predicates" in a critique of S5-based approaches to knowledge-based descriptions of protocols [San91].

This paper is organised as follows. In Section 2 we introduce and study the logic of local propositions. Section 3 considers a variant of this logic with weak semantics using bounded quantification over propositions. Section 4 provides an example illustrating a sense in which the semantics of knowledge-based programs given in [FHMV95] are overly restrictive, and how the logic of local propositions can help. Finally, Section 5 considers some connections to related work and provides concluding remarks.

2 The Logic of Local Propositions

Let *Prop* be a set of propositional variables. An *n-agent Kripke structure* is a tuple $M = \langle W, R_1, \ldots, R_n, \pi \rangle$ where W is a set of worlds, the R_i are equivalence relations on W, and $\pi : Prop \to \mathcal{P}(W)$ assigns a set of worlds to every propositional variable. If W is the set of worlds of an n-agent Kripke structure, we call a subset U of W a *proposition* on M. If p is a propositional variable, a structure M' is a *p-variant* of a structure M, denoted $M' \simeq_p M$, whenever it differs from M at most in the proposition assigned (by π') to p. We say that a proposition U on $M = \langle W, R_1, \ldots, R_n, \pi \rangle$ is *i-local* if for all $u, v \in W$ such that uR_iv, we have $u \in U$ iff $v \in U$. Intuitively, a proposition is i-local if agent i is able to determine its truth value using only locally available information. Put another way, a proposition U is i-local if it is a union of equivalence classes from R_i (and since R_i defines i's partition, U is a union of cells of i's partition).

All the languages we deal with contain the propositional operators $\wedge$, $\neg$ and the monadic modal operator $\Box$, which refers to truth in all worlds (of W). In addition, we will consider a number of different propositional quantifiers: $\forall$ and, for each agent i, the operator $\forall_i$, such that if p is a propositional variable and φ is a formula, then $\forall p(\varphi)$ and $\forall_i p(\varphi)$ are formulae. Generally, we shall write $\mathcal{L}_{(o_1,\ldots,o_m)}$ for the language generated from a set *Prop* of propositional variables by $\wedge$, $\neg$, and operators o_i. For instance, we write $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$ for the language generated from *Prop* by $\wedge$, $\neg$, $\Box$, $\forall$ and the $\forall_i$. The interesting cases of the semantics of this language are as follows: If $M = \langle W, R_1, \ldots, R_n, \pi \rangle$ and $w \in W$

- $M, w \models p$ if $w \in \pi(p)$, where $p \in Prop$.
- $M, w \models \Box\varphi$ if $M, u \models \varphi$ for all $u \in W$.
- $M, w \models \forall p(\varphi)$ if $M', w \models \varphi$ for all structures $M' \simeq_p M$.
- $M, w \models \forall_i p(\varphi)$ if $M', w \models \varphi$ for all structures $M' \simeq_p M$ such that $\pi'(p)$ is an *i-local* proposition.

This definition provides a standard, sometimes called *strong*, semantics to the propositional quantifiers, in that quantification is over the set of all propositions (all subsets of W). It is possible to weaken this semantics, along the usual lines for weak second order logic [Hen50]. This is pursued in Section 3.

The language $\mathcal{L}_{(\forall,\Box)}$ has been studied in the past, under a variety of semantics, including that just presented and the weak semantics we consider later [Kri59, Bul69, Kap70, Fin70]. The novelty of our proposal is the consideration of the multi-agent context, and the introduction of the local quantifiers $\forall_i$. Using local quantifiers, we can express a variety of epistemic notions. First of all, we now show that the standard S5 notions of knowledge, distributed knowledge, and common knowledge can be expressed in $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$. Recall that (see, e.g. [FHMV95]) the semantics of the knowledge operators K_i, where i is an agent, the distributed knowledge operators D_G, and the common knowledge operators C_G, where G is a set of agents, are defined by

- $M, w \models K_i\varphi$ if $M, u \models \varphi$ for all u with wR_iu,
- $M, w \models D_G\varphi$ if $M, u \models \varphi$ for all u with wR^D_Gu, where $R^D_G = \bigcap_{i\in G} R_i$, and
- $M, w \models C_G\varphi$ if $M, u \models \varphi$ for all u with wR^C_Gu, where R^C_G is the smallest equivalence relation containing $\bigcup_{i\in G} R_i$.

Each one of these three operators is expressible in $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$, as the following proposition shows.

Proposition 1 Let $i \leq n$ and $G = \{1, \ldots, m\}$ where $m \leq n$ and assume that $q, q_1, \ldots, q_m$ are propositional variables not occurring in φ. The following formulae are valid for n-agent Kripke structures.[1]

(1) $K_i\varphi \equiv \exists_i q(q \wedge \Box[q \to \varphi])$,

(2) $D_G\varphi \equiv \exists q(q \wedge \Box[q \to \varphi] \wedge \exists_1 q_1 \ldots \exists_m q_m \Box[q \equiv \bigwedge_{1\leq j\leq m} q_j])$, and

(3) $C_G\varphi \equiv \exists q(q \wedge \Box[q \to \varphi] \wedge \bigwedge_{1\leq j\leq m} \exists_j q_j \Box[q \equiv q_j])$.

Proof: For this abstract we show just point (1). Let $M = \langle W, R_1, \ldots, R_n, \pi\rangle$ and assume that $M, w \models K_i\varphi$. We prove $M, w \models \exists_i q(q \wedge \Box[q \to \varphi])$. Define M' by $M' \simeq_q M$ and $\pi'(q) = \{\, u \in W \;:\; wR_iu \,\}$. Since $w \in \pi'(q)$ we have $M', w \models q$. Using the fact that q does not occur in φ, a simple inductive argument shows that $M', u \models \varphi$ iff $M, u \models \varphi$ holds for all $u \in W$. Because $M, w \models K_i\varphi$, it follows that $M', u \models \varphi$ for all $u \in q$, and thus $M', w \models \Box[q \to \varphi]$. This shows that $M, w \models \exists_i q(q \wedge \Box[q \to \varphi])$, as desired.

Conversely, suppose $M, w \models \exists_i q(q \wedge \Box[q \to \varphi])$, and let $M' \simeq_q M$ such that $M', w \models q \wedge \Box[q \to \varphi]$. Let $U = \{\, u \in W \;:\; wR_iu \,\}$ be the equivalence class of R_i containing w. Then U is the smallest i-local proposition satisfying $w \in U$. Because $M', w \models q$, and $\pi'(q)$ is i-local, we have $U \subseteq \pi'(q)$. Thus, it follows from $M', w \models \Box[q \to \varphi]$ that $M', u \models \varphi$ for all $u \in U$. By the same consideration as above, we obtain that $M, u \models \varphi$ for all $u \in U$, i.e. that $M, w \models K_i\varphi$. ■

It is interesting to observe that the expressions given above for the three operators have the structure $\exists q(q \wedge \Box[q \to \varphi] \wedge \psi)$ for some ψ.

It is also possible to express in our language epistemic operators related to the notion of *only knowing* [HM84, Lak93, Lev90]. Several semantics are considered in these works for a modal operator O_i, such that the formula $O_i\varphi$ is intended to capture the intuitive notion that agent i's knowledge is completely characterized by the formula φ. The formula

$$\exists_i q(q \wedge \Box[q \equiv \varphi] \wedge \forall_i p(p \to \Box[q \to p]))$$

[1] The global quantifier $\exists q$ in (2) and (3) can be eliminated at the cost of a slightly more cumbersome or less symmetric form, i.e., all three operators are already expressible in $\mathcal{L}_{(\forall_1,\ldots,\forall_n,\Box)}$.

captures one sense of this intuition: it asserts that the equivalence class of agent i containing the current world consists precisely of the worlds at which φ is true. The notion expressed by this formula is distinct from the other notions considered in the literature for several reasons. For example, Levesque's definition of only knowing [Lev90] (which assumes a single agent) is based on the logic K45 rather than S5. Moreover, while we allow arbitrary Kripke structures, work in this area typically assumes a particular structure, containing a representative for all worlds that could occur in *any* Kripke structure. In a single agent setting this means that the set of worlds amounts to the set of all possible assignments of truth values to the propositions of interest. One could express this constraint using the operator '$\Box$' in the case of a single agent and a finite number of propositions, but there are added complexities in the multi-agent case. As these issues are rather subtle, we will not pursue them here.

It is interesting to note that we could use the notion of locality of propositions to provide an alternate basis for the language $\mathcal{L}_{(\forall_1,\ldots,\forall_n,\Box)}$. For each agent $i = 1,\ldots,n$, introduce an operator L_i, such that $L_i\varphi$ expresses that φ is an i-local formula. The semantics of these operators is defined by:

$$M, w \models L_i\varphi \text{ if } \{\ u\ :\ M, u \models \varphi\ \} \text{ is an } i\text{-local proposition.}$$

The operators L_i are expressible in $\mathcal{L}_{(\forall_1,\ldots,\forall_n,\Box)}$, since the following is valid:

$$L_i(p) \equiv \exists_i q \Box[p \equiv q] \qquad \mathbf{L_iD}$$

Conversely, the operators $\forall_i$ can be defined in $\mathcal{L}_{(\forall,L_1,\ldots,L_n)}$, using the validity of

$$\forall_i p(\varphi) \equiv \forall p(L_i(p) \rightarrow \varphi) \qquad \mathbf{\forall_iD}$$

It is also worth noting that there are some interesting interactions between the knowledge operators and the local quantifiers. Observe that the formula

$$\forall q(L_i(K_i q) \wedge L_i(\neg K_i q))$$

is valid. In particular, this implies that all knowledge formulae express local propositions. Using this fact, we may see that the formula

$$\forall_i q(K_i q \equiv q)$$

which is also valid, is a generalization of the positive and negative introspection axioms of S5.

For our strong semantics, the logic of $\mathcal{L}_{(\forall,\Box)}$ is known to have a decidable satisfiability problem. Moreover, the following axioms

all S5 axioms for $\Box$.	$\mathbf{S5\Box}$
$\forall p(\varphi(p)) \rightarrow \varphi(\psi)$, where ψ is a formula free for p in φ.	$\mathbf{\forall 1}$
$\forall p(\varphi \rightarrow \psi) \rightarrow (\forall p(\varphi) \rightarrow \forall p(\psi))$.	$\mathbf{\forall D}$
$\varphi \rightarrow \forall p(\varphi)$, where p is not free in φ.	$\mathbf{\forall N}$
$\exists q(q \wedge \forall p(p \rightarrow \Box[q \rightarrow p]))$.	$\mathbf{AT}$

together with the rules of inference

If $\vdash \varphi$ and $\vdash \varphi \rightarrow \psi$ then $\vdash \psi$. **MP**

If $\vdash \varphi$ then $\vdash \Box\varphi$. **□G**

If $\vdash \varphi$ then $\vdash \forall p(\varphi)$. **UG**

provide a sound and complete axiomatization for $\mathcal{L}_{(\forall,\Box)}$ [Bul69, Kap70, Fin70]. Most of these axioms and rules are to be expected, being simply derived from axiomatizations of the logics for the operators $\forall$ and $\Box$ individually. The one interesting case is **AT** which expresses that there exists a minimal proposition true at the current world. (For each $w \in W$, the proposition $q = \{w\}$ satisfies **AT** at w, since $\{w\}$ is the minimal proposition true at w.)

Is the logic of the richer language $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$ also axiomatizable and decidable? Using the fact that the $\forall_i$ are restricted versions of $\forall$, as expressed by $\forall_i$**D**, it is straightforward to derive some valid formulae and rules for $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$:

(1) $(\forall_i p(\varphi(p)) \wedge \exists_i q\Box[\psi \equiv q]) \rightarrow \varphi(\psi)$, where ψ is a formula free for p in φ.

(2) $(\forall_i p(\varphi) \;\wedge\; \forall_i p(\varphi \rightarrow \psi)) \;\rightarrow\; \forall_i p(\psi)$.

(3) $\varphi \rightarrow \forall_i p(\varphi)$, where p is not free in φ.

(4) $\exists_i q(q \wedge \forall_i p(p \rightarrow \Box[q \rightarrow p]))$.

(5) If $\vdash \exists_i q(\Box[p \equiv q]) \rightarrow \varphi$ then $\vdash \forall_i p(\varphi)$.

In particular, note that (4) reflects the fact that every world is in a unique minimal i-local proposition, namely the R_i equivalence class containing that world.

However, the language $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$ also has some valid formulae that do not correspond directly to those listed above for $\mathcal{L}_{(\forall,\Box)}$. For example, common knowledge is known to satisfy the following *induction principle*:

$$\Box[\varphi \rightarrow \textstyle\bigwedge_{i\in G} K_i(\psi \wedge \varphi)] \;\rightarrow\; \Box[\varphi \rightarrow C_G\psi]$$

Using Proposition 1, this may be translated to a valid formula of $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$, one that does not appear to follow from the abovementioned rules and axioms. The fact that it is possible to formulate such induction principles suggests that $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$ may be a considerably more expressive language than $\mathcal{L}_{(\forall,\Box)}$. Indeed, this proves to be the case, in a very strong sense.

Theorem 2 *There exists an interpretation of second order predicate logic in $\mathcal{L}_{(\forall_1,\ldots,\forall_n,\Box)}$, provided $n \geq 2$. Consequently, $\mathcal{L}_{(\forall_1,\ldots,\forall_n,\Box)}$ is not recursively axiomatizable.*

This result stands in marked contrast to the fact that $\mathcal{L}_{(\forall,\Box)}$ is decidable.[2] This indicates that, in one sense, the strong semantics we have considered in this section is *too* strong. We consider an alternative in the next section.

[2]We remark that if the logic of the modality $\Box$ is S4.2 or weaker, or is S4.3, then $\mathcal{L}_{(\forall,\Box)}$ is again equivalent to second order predicate logic in expressive power [KT96].

3 A Weak Semantics for Local Propositions

It is possible to weaken the semantics of logics of local propositions given in the previous section along the usual lines for weak second order logic [Hen50]. We could obtain this by having the Kripke structure specify a set of propositions that the quantifiers range over. That is, a *weak n-agent Kripke structure* is a tuple of the form $M = \langle W, P, R_1, \ldots, R_n, \pi \rangle$, where W is a set of worlds, P is a non-empty subset of $\mathcal{P}(W)$, the R_i are equivalence relations on W and $\pi : \mathit{Prop} \to P$ is now required to assign a proposition in P to each propositional variable. The semantics is exactly as before, except that in the cases dealing with $\forall p(\varphi)$ and $\forall_i p(\varphi)$, the structures $M' \simeq_p M$ are required to be weak n-agent Kripke structures that differ from M only on the value of $\pi(p)$, and this value in M' is now required to be an element of the set of propositions P. The quantifiers thus range over the propositions in P, rather than over $\mathcal{P}(W)$ as before.

A variety of constraints could be placed on the allowable sets of propositions P. Previous work [Bul69, Kap70] has considered the case of the language $\mathcal{L}_{(\forall,\Box)}$ when P is either (1) closed under Boolean operators or (2) closed under formulae, i.e., satisfies the constraint that the set of worlds satisfying a formula of $\mathcal{L}_{(\forall,\Box)}$ is a proposition in P. These cases are known to be axiomatizable. For (2), the axiomatization consists of axioms **S5**$\Box$, $\forall$**D**, $\forall$**N**, $\forall$**1** and rules **MP**, $\Box$**G**, and **UG**. For (1), one replaces $\forall$**1** by the variant in which ψ is required to be a formula of *propositional logic*, free for p in φ. Note that both of these axiomatizations exclude the axiom **AT**: this principle is not valid under conditions (1) or (2).

Conditions (1) and (2), however, are just two cases from a much larger set of plausible candidates for constraints on P. Another restriction of interest would be to require P to be the set of propositions computable within some complexity bound (polynomial time, for example), or the set of propositions computable by an agent able to perform a specified set of operations. Assumptions such as these would be plausible candidates for applications of the logic to specification of cryptographic protocols.

It is interesting in this context to examine the interpretation with respect to the weak semantics of the formula we found in Proposition 1 to be equivalent to S5 knowledge. Denote by $K_i^l\varphi$ the formula $\exists_i p(p \wedge \Box[p \to \varphi])$ from Proposition 1(1). Interpreted with respect to the weak semantics, $K_i^l\varphi$ is no longer equivalent to the information theoretic notion of knowledge captured by the formula $K_i\varphi$, so Proposition 1 does not hold under this interpretation. Indeed, it now appears that *no* formula of $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$ is able to express knowledge. However, we are still able to express interesting knowledge-like notions, and $K_i^l\varphi$ is one such notion. We will take up this topic in the following section, and concentrate here on the question of some of the properties of the weak semantics, including axiomatizability. For the purposes of this abstract, we focus on the case where the only constraint on the set of propositions is that it be non-empty.

Besides losing the ability to express knowledge, it also appears that the ability to express locality is lost under the weak semantics: in particular, the formula L_i**D** is no longer sound. As a consequence, we are no longer able to relate local and global quantification through a version of the formula $\forall_i$**D**.

One expects that, in our context, the move from the strong to the weak semantics corresponds to the move from a non-axiomatizable logic to an axiomatizable one, just as it does for second order predicate logic [Hen50]. We will show below that this is indeed the case.

However, the loss of expressiveness just noted appears to create difficulties for axiomatization, and it seems that a complicated set of axioms will be required to capture the weak logic of $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$. In the full version of the paper, we will expand upon the nature of the difficulties and provide some examples of the complex axioms one encounters. Here, we focus on one way of avoiding the complexities, which is to work with a richer language that recaptures some of the lost expressive power.

In particular, it turns out to be sufficient to work with a language that includes the knowledge operators K_i. Let us begin by noting that these are sufficient to express locality, for

$$L_i\varphi \equiv \Box(K_i\varphi \vee K_i\neg\varphi) \qquad \mathbf{L}_i\mathbf{K}_i$$

is valid (under both strong and weak semantics). Thus, using axiom $\forall_i\mathbf{D}$, which continues to hold under the weak semantics, the languages $\mathcal{L}_{(\forall,K_i,\ldots,K_n,\Box)}$ and $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,K_i,\ldots,K_n,\Box)}$ have equivalent expressive power. We focus on providing an axiomatization based on the operators in the former.

Observe that Axiom $\forall\mathbf{1}$ is unsound under the weak semantics because not every formula is equivalent to some proposition in the range P of quantification. However, because the weak semantics requires that every propositional constant refers to a proposition in P, a special case is valid:

$$\forall p(\varphi(p)) \rightarrow \varphi(q) \text{ , where } q \in \mathit{Prop} \text{ is free for } p \text{ in } \varphi. \qquad \forall\mathbf{1'}$$

Define the proof system $\mathbf{W}$ for weak $\mathcal{L}_{(\forall,K_1,\ldots,K_n,\Box)}$ to consist of all S5 axioms and rules for each K_i, the axioms $\mathbf{S5}\Box$, $\forall\mathbf{1'}$, $\forall\mathbf{D}$, $\forall\mathbf{N}$, the axiom

$$\Box\varphi \rightarrow K_i\Box\varphi \ , \qquad \Box\mathbf{K}_i$$

and the rules **MP**, $\Box$**G**, and **UG**.

Theorem 3 *System* $\mathbf{W}$ *is a sound and complete axiomatization of weak* $\mathcal{L}_{(\forall,K_1,\ldots,K_n,\Box)}$.

Proof: Soundness of $\mathbf{W}$ is straightforward. The completeness proof uses a variant of well-known Henkin-style completeness techniques for modal predicate logic. See, e.g., [HC96]. ■

From this result and the fact that weak $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$ is expressible in weak $\mathcal{L}_{(\forall,K_1,\ldots,K_n,\Box)}$, it follows that the former is recursively enumerable. (As we have remarked above, finding an axiomatization based on these operators alone may be complex, however.)

Theorem 4 *For* $n \geq 2$, *weak* $\mathcal{L}_{(\forall_1,\ldots,\forall_n,\Box)}$ *and weak* $\mathcal{L}_{(\forall,K_1,\ldots,K_n,\Box)}$ *are not decidable.*

The proof of Theorem 4 amounts to noting that the proof of Theorem 2 also provides an interpretation of *weak* second order predicate logic in *weak* $\mathcal{L}_{(\forall_1,\ldots,\forall_n,\Box)}$ for $n \geq 2$.

4 Knowledge-based programs: An application

We mentioned in the introduction that this work has been motivated by a project whose goal is to facilitate the development of computer programs. Specifically, we are interested in providing adequate tools for the use of epistemic notions in such development. In this section we wish to describe some of the problematic aspects of the current approach to using knowledge in programming, and illustrate how logics of local propositions can help. A similar issue is treated in [HM98], where a modification of knowledge-based programs based on counterfactuals is suggested.

A knowledge-based program is best viewed as a specification. Intuitively, in the definitions of [FHMV95], a standard program is said to *implement* (or represent) a knowledge-based program if its actions are always consistent with those dictated by the knowledge-based program.[3] The tests for knowledge are evaluated in terms of the information-based S5 notion discussed in the introduction. Moreover, in the Kripke structure used to evaluate these tests, both the context and the (behavior of the) standard program are common knowledge.

Let us consider an example in which this semantics leads to an undesirable outcome. Imagine a very simple context, in which there is a single agent a. There is one variable, x, and the world can be in two possible states called s_0 and s_1, in which $x = 0$ and $x = 1$, respectively. Any program we design is guaranteed to start executing in the state s_0. The agent can perform one of two actions: $x := 1$, which has the effect of assigning the value 1 to the variable x, and skip, which leaves x with its current value. Finally, let "$\texttt{next}\, x = 1$" be a formula that holds if $x = 1$ is true at the next state of the computation. Imagine a situation in which there is a cost associated with performing the assignment $x := 1$. Let Pg be the program

$$\textbf{if } K_a(\texttt{next}\, x = 1) \textbf{ then } \mathsf{skip} \textbf{ else } x := 1.$$

This program has a very natural interpretation. Intuitively, it can be viewed as specifying the desirable behavior of an agent whose goal is to ensure that $x = 1$ ultimately holds. When the agent is assured that $\texttt{next}\, x = 1$, it is not required to act. If, however, the agent is not sure that the goal will be satisfied, it should take an action ($x := 1$) that will guarantee that the goal is attained. While these intuitions seem natural enough, there is a problem.

Proposition 5 The knowledge-based program Pg is unimplementable in the context described.

Sketch of proof: A careful analysis shows that there are only two candidate programs to check, one consisting of the single action skip, and the other consisting of the action $x := 1$. In the [FHMV95] definition, the agent's knowledge is based both on the fact that the initial state can only be s_0 and on the program being followed. Hence, when following the program skip the agent knows that $\neg\,\texttt{next}\, x = 1$, for which Pg specifies the agent perform $x := 1$. Similarly, when following the program $x := 1$, the agent knows that $\texttt{next}\, x = 1$, for which Pg specifies the action skip. As a result, each of the candidate programs fails to implement Pg. ∎

It is worth noting that there is nothing inherently "wrong" or inconsistent with the program Pg. In a slightly modified context in which both s_0 and s_1 are possible initial states (and the agent's local state cannot distinguish the two), Pg is implementable. Indeed, the program consisting of the action $x := 1$ implements Pg in that setting. This example showed that the current definition of implementation is very restrictive: the only tests for knowledge allowed are ones that make full use of all information about the implementation and the context. This focuses

[3] For a short summary of the [FHMV95] definitions underlying knowledge-based programs, which is beyond the scope of this paper, the reader is refered to [HM98] in these proceedings.

attention on "optimal" solutions. While in some cases, this is exactly what is desired [DM90], in other cases, such as in the case of Pg, this results in a sensible looking program being unimplementable. Our point of view is that sub-optimal tests should be considered acceptable, as long as they are sound: a positive answer for a test for $K_i\varphi$ should imply that φ holds, but a negative answer may be acceptable even if the agent's information determines that φ holds.

We can capture this idea in $\mathcal{L}_{(\forall,\forall_1,\ldots,\forall_n,\Box)}$. Consider the formula $\exists_i p(\Box[p \equiv q]) \wedge \Box[q \rightarrow \varphi]$, which for brevity we denote $S_i(q,\varphi)$. Roughly speaking, $S_i(q,\varphi)$ says that q is a sound local witness for φ.[4] It is easy to verify that $(q \wedge S_i(q,\varphi)) \rightarrow K_i\varphi$ is valid. Hence, we can also interpret $S_i(q,\varphi)$ as saying that q is a sound witness for $K_i\varphi$. This is of interest, for example, if we want to apply logics of local propositions as a tool for relaxing the semantics of knowledge based programs. (Very) roughly speaking, we can now accept as an implementation of a test for $K_i\varphi$ in a knowledge-based program any local condition q satisfying $S_i(q,\varphi)$. In the specific case of our program Pg, let us consider two such conditions. The first is $q = \mathit{false}$. It trivially satisfies $S_a(q,\varphi)$ for every φ. The standard program Pg_0 resulting from substituting *false* for the test for knowledge in Pg is equivalent to the desirable program $x := 1$, which is a good solution that obtains the goal. A second condition q' satisfying $S_a(q',\varphi)$ is $x = 1$. Notice that when the only actions possible are skip and $x := 1$, then $x = 1 \rightarrow \ \mathtt{next}\, x = 1$ is valid. If we assume that the value of x is testable by the agent, the test is sound. Indeed, in our context, where $x = 0$ is guaranteed to hold initially, the two programs Pg_0 and Pg_1 generate the same behaviors. It is interesting to note that both programs are correct even if we are not guaranteed that $x = 0$ holds initially. In such a case (say x may initially have either value 0 or value 1), however, the program Pg_1 will perform the assignment $x := 1$ (and incur the associated costs) only when this is necessary, while Pg_0 will always do so. We thus view both programs as being correct, while Pg_1 is in some cases more sophisticated than Pg_0 is.

We remark that there are other reasons that make desirable a change in the definition of implementation for knowledge-based programs along the lines proposed here. Intuitively, the original definition of implementation of knowledge-based programs assumes that the final standard program is common knowledge, and knowledge is evaluated with respect to this information. As a result, it is difficult to implement a knowledge-based program in a modular way, by implementing pieces of the program and combining the implementations. By going to sound tests, such an approach is facilitated. This is related to a proposal by Sanders [San91] to use sound predicates for knowledge for implementing tests in knowledge-based programs.

In summary, the original definition of implementation for knowledge based programs requires tests for knowledge to be replaced by tests that are both sound and complete. Our analysis here illustrates that it is possible to relax the requirement by allowing tests to be sound but not necessarily complete. The example shows that this is sometimes necessary: The original definition introduces a circularity that in some cases makes a reasonable program such as Pg unimplementable.

We remark that [HM98] addresses very similar issues of unimplementability of knowledge-based programs. They use a somewhat different approach based on counterfactuals. They argue that in our program Pg we should replace the test for knowledge of $\mathtt{next}\, x = 1$. Roughly speaking, they suggest we replace it by a test for knowledge of the counterfactual statement: *if*

[4] Similar issues arise in the definition of *algorithmic* knowledge [FHMV95], where there are algorithms that serve to test the local state of the agent for $K_i\varphi$, and they are often required only to be sound, in the sense that a YES answer implies that $K_i\varphi$ holds, but a "?" answer is allowed even if $K_i\varphi$ holds.

the agent were to perform skip *then* next $x = 1$.

5 Conclusion

We have introduced two logics of local propositions, one with a standard (strong) notion of quantification, and the other with a weak one. What have we gained? The resulting logics provide us with a framework in which a wide range of epistemic notions can be defined. In all cases, these definitions are made on top of the natural informational structure, which itself induces equivalence relations (partitions) on the possible worlds. First of all, we have obtained considerable expressive power. In particular, we can conclude from Proposition 1 that the standard information-based S5 logic of knowledge and common knowledge can be translated directly into (strong) logics of local propositions. Unfortunately, strong logics of local propositions are so expressive, that they are also intractable. They are not even recursively axiomatizable. Weak logics of local propositions are somewhat more tractable. While standard S5 knowledge is not definable in this framework, a related notion $K_i^! \varphi$ can be defined by the same formula $\exists_i p(p \wedge \Box[p \to \varphi])$, characterizing knowledge under the strong semantics, but this time interpreted under the weak semantics. The meaning of $K_i^! \varphi$ in this case depends crucially on the structure of P. In general, $K_i^!$ is not closed under deduction in this case: the Distribution axiom fails in general. A slightly weaker form of logical omniscience does hold, however. It is a notion called *closure under logical implication* in [FHMV95]: If both $K_i^! \varphi$ and $\Box[\varphi \to \psi]$ hold, then $K_i^! \psi$ holds. In weak n-agent Kripke structures $M = \langle W, P, R_1, \ldots, R_n, \pi \rangle$, we can view the set P as inducing a notion of awareness [FH88]. The formula $K_i^! \varphi$ will hold only if some i-local $q \in P$ exists with the desirable properties. Thus, i is "aware" only of what follows from the i-local propositions in P. An interesting example is when P corresponds to a complexity class. E.g., roughly speaking, consider P to be the set of all PTIME (polynomial-time) computable properties. Then $K_i^! \varphi$ holds if agent i can detect the truth of φ via some PTIME computable property. (This does not imply, by the way, that i can efficiently determine which PTIME property to use in a given world.) The resulting notion $K_i^! \varphi$ obtained this way is essentially the notion of resource-bounded knowledge from [Mos88]. We elaborate on this and other connections in the full paper.

The crucial point, of course, is that strong and weak logics of local propositions allow us to express much more than just $K_i^! \varphi$. Many of these more general statements are clearly beyond the scope of other systems that have been proposed to overcome the logical omniscience problem, such as Montague-Scott [Mon68, Sco70] or awareness logics. Nevertheless, they can involve local and epistemic aspects. For example, the formula $S_i(q, \varphi)$ discussed in Section 4 has a distinct epistemic flavor. Moreover, as we have seen, it can serve the basis of a modified definition of implementation for knowledge-based programs. With such a modification, the anomalies discussed in Section 4 can be overcome.

We believe that the logic of local propositions can play a useful role in the development of computer programs in a more general setting. Indeed, in developing a framework for knowledge-based development of distributed protocols, we are finding the greater flexibility in defining implementations using this approach to be a useful and promising tool. Details of that will be the subject of future papers.

References

[Aum76] R. J. Aumann. Agreeing to disagree. *Annals of Statistics*, 4(6):1236–1239, 1976.

[Bul69] R. Bull. On modal logic with propositional quantifiers. *Journal of Symbolic Logic*, 34:257–263, 1969.

[DH76] W. Diffie and M. E. Hellman. Multiuser cryptographic techniques. In *Proc. AFIPS 1976 National Computer Conference*, pages 109–112, Montvale, NJ, 1976.

[DM90] C. Dwork and Y. Moses. Knowledge and common knowledge in a Byzantine environment: crash failures. *Information and Computation*, 88(2):156–186, 1990.

[FH88] R. Fagin and J. Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988.

[FHMV95] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, MA, 1995.

[Fin70] K. Fine. Propositional quantifiers in modal logic. *Theoria*, 36:336–346, 1970.

[Get63] E. Gettier. Is justified true belief knowledge? *Analysis*, 23:121–123, 1963.

[HC96] G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic*. Routledge, London, New York, 1996.

[Hen50] L. A. Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15:81–91, 1950.

[Hin62] J. Hintikka. *Knowledge and Belief*. Cornell University Press, Ithaca, NY, 1962.

[HM84] J. Y. Halpern and Y. Moses. Towards a theory of knowledge and ignorance. In *Proc. AAAI Workshop on Non-monotonic Logic*, pages 125–143, 1984. Reprinted in K. R. Apt (Ed.), *Logics and Models of Concurrent Systems*, Springer-Verlag, Berlin, New York, pp. 459–476, 1985.

[HM90] J. Y. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, 1990. A preliminary version appeared in *Proc. 3rd ACM Symp. on Principles of Distributed Computing*, 1984.

[HM98] J. Y. Halpern and Y. Moses. Using counterfactuals in knowledge-based programs. In Y. Gilboa, editor, *Theoretical Aspects of Rationality and Knowledge*, 1998. (These proceedings).

[HMW90] J. Y. Halpern, Y. Moses, and O. Waarts. A characterization of eventual Byzantine agreement. In *Proc. 9th ACM Symp. on Principles of Distributed Computing*, pages 333–346, 1990.

[HZ92] J. Y. Halpern and L. D. Zuck. A little knowledge goes a long way: knowledge-based derivations and correctness proofs for a family of protocols. *Journal of the ACM*, 39(3):449–478, 1992.

[Kap70] D. Kaplan. S5 with quantifiable propositional variables. *Journal of Symbolic Logic*, 35:355, 1970.

[Kri59] S. Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24:1–14, 1959.

[KT96] M. Kaminski and M. Tiomkin. The expressive power of second-order propositional modal logic. *Notre Dame Journal of Formal Logic*, 37(1):35–43, 1996.

[Lak93] G. Lakemeyer. All they know: a study in multi-agent autoepestemic reasoning. In *Proc. 13th International Joint Conference on Artificial Intelligence (IJCAI '93)*, pages 376–381, 1993.

[Len78] W. Lenzen. Recent work in epistemic logic. *Acta Philosophica Fennica*, 30:1–219, 1978.

[Lev90] H. J. Levesque. All I know: a study in autoepistemic logic. *Artificial Intelligence*, 42(3):263–309, 1990.

[Mon68] R. Montague. Pragmatics. In R. Kalibansky, editor, *Contemporary Philosophy*, pages 101–121. La Nuova Italia Editrice, Florence, Italy, 1968.

[Mos88] Y. Moses. Resource bounded knowledge. In M. Y. Vardi, editor, *Reasoning about Knowledge, Proc. 2nd Conf.* Morgan Kaufmann, 1988.

[MT88] Y. Moses and M. R. Tuttle. Programming simultaneous actions using common knowledge. *Algorithmica*, 3:121–169, 1988.

[NB92] G. Neiger and R. Bazzi. Using knowledge to optimally achieve coordination in distributed systems. In Y. Moses, editor, *Proc. of the 4th Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 43–59. Morgan Kaufmann, 1992.

[San91] B. Sanders. A predicate transformer approach to knowledge and knowledge-based protocols. In *Proc. 10th ACM Symp. on Principles of Distributed Computing*, pages 217–230, 1991. A revised report appears as ETH Informatik Technical Report 181, 1992.

[Sco70] D. Scott. Advice on modal logic. In K. Lambert, editor, *Philosophical Problems in Logic*, pages 143–173. Reidel, Dordrecht, Netherlands, 1970.

[vW51] G. H. von Wright. *An Essay in Modal Logic.* North-Holland, Amsterdam, 1951.