An Epistemic Logic of Situations (Extended Abstract)

Paul F. Syverson
Code 5543
Naval Research Laboratory
Washington, DC 20375
(syverson@itd.nrl.navy.mil)

In this paper we present a first order epistemic logic that incorporates the essentially finite character of what is actually known by any knower. Our logic and language allows us to represent familiarity with individuals including individual situations. It is also a logic of limited awareness in the manner of [FH88]. It is adequate for the syntactic characterization of the shared-situation account of common knowledge [Bar89]. Finally, it is sound and complete with respect to the presented semantics.

1 Common Knowledge

Common knowledge has been given a number of logical characterizations. (Cf. [Bar89] and [HM90].) We have no space to review even the basics of all of them here; however, we set out the shared-situation account since capturing this is one of the important features of our logic.[1]

On this account, A and B have common knowledge that φ just in case there is a situation s such that:

- $s \models \varphi$
- $s \models A$ knows s.
- $s \models B$ knows s.

Barwise develops much of the model theory of situations with respect to common knowledge in [Bar89]. Below we will give our own characterization of semantic notions, such as what it means for a sentence to be true in a situation. For the present we simply think of situations intuitively as partial possible worlds.[2] So, $s \models$ (A knows s.) can be viewed as saying that 'A knows s.' is true at s. That is, in situation s, A knows that s is the situation.

[1] The first version of this account was given in [Lew69].

[2] In this paper, no significance is attached to the sometimes important distinction between (possible) worlds and situations, and we will use the corresponding expressions interchangeably below.

Notice that 's' is used both in the object language and metalanguage to refer to the same situation. Barwise introduces a language that allows such dual usage, both of names of situations and of the satisfaction symbol '$\models$'. Our language presented below has this feature. We also present a logic that is sound and complete with respect to our semantics.

2 Three Grades of Epistemic Involvement

Epistemic logic in the tradition of [Hin62] generally fails to be cognitive in the sense of accounting for the finite memory and inferential speed of any actual, nonideal reasoner. In particular, the rule of epistemic generalization runs contrary to a basic feature of knowledge that we would like to capture.

2.1 Logical Omniscience

All normal systems of modal logic contain a rule of the form[3]

$$\text{From} \vdash \varphi \text{ infer} \vdash \Box\varphi.$$

If the operator '$\Box$' is understood as *alethic* (i.e., pertaining to necessity), this rule is usually called necessitation. If the operator is understood as *epistemic* (i.e., pertaining to knowledge), as in our case, it is usually called epistemic generalization. Unfortunately, in the epistemic context this leads to the so called logical omniscience problem, viz: each agent knows all logical truths. One feature of our logic that addresses this issue is our limitation on familiarity with individuals. Like many modal logics ours has a possible world semantics where what is known in a situation w is affected by what other situations are accessible to an agent. Our representation of familiarity will fit naturally into this context. Here is an example. If a world w' is accessible from world w for Addie, then at world w, she cannot tell them apart. From the perspective of world w, Addie finds both w and w' equally possible ways things might be. Now, suppose some individual, Bob, is present at one of these worlds but not at the other. (What 'present' means will be made precise once the model theory is spelled out below.) Then Addie cannot tell the difference between a world where Bob is present and one where he is not. So, she must not really be aware of Bob, *know* Bob, if she can't tell whether he's there or not. We have elsewhere described logics that were devoted specifically to this issue ([Syv90] and [Syv93], Chap. 4). Here it is just one type of cognitive limitation. Various attempts have been made to solve logical omniscience and related problems by restricting the logic in one way or another. (Cf. e.g., [Ebe74], [FH88], [Lev85], [Lev90].) Other research has been done on analyzing the complexity issues in reasoning about knowledge and belief. (Cf. e.g., [GMR85], [HV86], [Var89], [HM92].)

While perhaps a useful simplifying abstraction in some contexts, epistemic generalization is too strong a rule for a realistic representation of actual knowledge; positing the ideal rationality implicit in such a rule is patently unrealistic. Not only does knowledge of logical truths vary from individual to individual but also from situation to situation for the same individual. Therefore, it seems far more reasonable to simply abandon such an idealization

[3]For an explication of 'normal' cf. [Che80], pp. 113–ff.

and to set out our assumptions about the known formulae in specific situations just as, in the manner just described, we might set out the known individual terms in specific situations. One way to pursue this line would be to limit awareness at a world to some subset of the primitive propositions. In a sense we have already done this in the logics of familiarity cited above: any formula containing an undefined term is itself undefined. But, this does not preclude unrealistic knowledge of very complicated formulae. For, we can still construct arbitrarily large formulae out of the known ones. This could be avoided by restricting logical closure in some way so that even though, e.g., an agent is aware of φ and ψ in a situation, he is not aware of $\varphi \wedge \psi$ in that situation. This is a feature of the approach taken in [FH88], and of the approach taken in this paper.

2.2 Epistemic Situations

We will present a possible world or situation semantics. Unlike other approaches, our situations are meant to capture different possible conceptions of the world. Any knower, any cognitive agent, might potentially know infinitely many things. But, any knower that we are interested in representing, be it a person or a computer process or whatever, actually knows only finitely many things in any given situation. And, our basic world structure will reflect this. Trying to reflect this while remaining as close as we can to traditional possible worlds might lead us to determine a world by the (finite) set of formulae that is true there. But, we need something more. A conceptual world should reflect not only what is true (and false) but also what is understood. For instance, while I might believe that it is raining outside my window right now, I understand the statement that it is not raining outside my window right now. This understanding is not determined by the truth values I attach to statements (except for some special cases to be discussed below). Truth values, however, are affected by understanding since, in a conceptual world, a formula cannot be true or false if it is not understood. This is not quite as trivial as it would be for alethic worlds. Because of the limited cognitive nature of our worlds, outside of certain constraints, the true formulae do not necessarily determine the false formulae (or the domain of quantification).

It is necessary to grasp a fundamental difference in the way possible worlds are to be thought of in our semantics. Ordinarily, possible worlds are construed as different ways the world might be. These are metaphysical possibilities and the modality that they naturally engender is alethic. Nonetheless, these same possible worlds are ordinarily used to underly epistemic logic. Knowledge is represented as the ability to discriminate amongst all the ontologically given possible worlds. But, if our logic is to be one of epistemic rather than alethic modality, then this should be reflected in the worlds that engender that modality. Thus, our worlds are construed as different ways the world might be conceived.

We will treat worlds atomically. In this sense they will serve essentially as indices to determine the terms and formulae that are understood at the given world. Thus, in the definition of a model we will need an awareness function h to determine the understood formulae. This is somewhat like the approach taken in [FH88] except that their awareness function depends on both worlds and individuals. In their semantics different individuals can be aware of different formulae in the same situation. Once we allow this, we have lost the intuitive semantic motivation that we started with, that of having worlds that are themselves epistemically rather than metaphysically based. From our perspective, it will be important to maintain an awareness function that is dependent on situations only.

3 The Language and Semantics

The language contains a denumerable supply of individual constant symbols: a, b, c, ... These serve as names for individuals and may or may not have subscripts. We also have individual variables: x, y, z, ..., also with or without subscripts. These are the atomic terms. They can be combined via function symbols to build up compound terms in the usual recursive manner. Amongst the individual constant symbols: $a, b, c, \ldots$ we distinguish a set of situation constants referring to situations (possible worlds): $w_1, w_2, w_3, \ldots$ These may stand in all the places where ordinary constants go. We also distinguish amongst the individual variables a set of variables that can only be instantiated with situation constants: z_1, z_2, z_3, ... Our language also contains denumerably many predicate constants: P, Q, R, ..., each of finite arity and taking tuples of individual constants as arguments. We generally do not indicate the arity of predicate constants. If necessary this is done via superscripts. Predicate symbols also may or may not be subscripted. Of the predicate constants, we call particular attention to a set of unary epistemic predicate constants: C_1, C_2, ..., C_n. We will also write these with individual names as subscripts. Intuitively, $C_b(a)$ should be taken to mean that b knows a, i.e., b can recognize a. (Recall the example with Addie and Bob above: in that example $\neg C_{Addie}(Bob)$ is true. This will be given a precise semantics below.) We also have the identity symbol, $=$.

The language is first order. The basic (open) sentences of the language are expressions of the form $P(x_1, \ldots, x_j)$ or $P(a_1, \ldots, a_j)$. Closed sentences are those containing no free variables. Sentences (open or closed) may be assembled into (finite) complex sentences according to ordinary recursive formation rules using the usual connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$ with the usual scope assumptions and using the usual delimiters (parentheses and brackets). We also have a finite set of propositional knowledge operators: S_1, S_2, ..., S_n. These are standard epistemic operators in the tradition of [Hin62]. Intuitively $S_a\varphi$ should be taken to mean that a knows the proposition expressed by φ.[4] (φ is a metalinguistic variable ranging over arbitrary sentences.) $S_a\varphi$ is a sentence, provided that φ is a sentence. Thus these operators may be iterated. Finally, we add '$\models$' to the object language as follows. If 'w' is a situation constant and 'φ' is any formula, then '$w \models \varphi$' is a formula.

3.1 Models

A model is a tuple $\langle D, W, R_1, \ldots R_n, h, \iota, \{\iota_w\}_{w \in W}, a\rangle$ where W is a set of nonempty possible worlds; R_1, ..., R_n are reflexive binary accessibility relations between members of W; and D is a domain of objects for all possible worlds. Since worlds themselves are objects, $W \subseteq D$.

The function h goes from members of W to finite sets of formulae in the language; it gives the terms and formulae understood in different situations. We represent that a

[4]The choice of symbols for knowledge derives from the French words '*connâitre*' and '*savoir*'. For example, in French, you *connais* a person but you *sais* that it's raining. English does not distinguish these senses and translates both words as 'to know'. (Actually, especially in Scottish dialectical English the word 'ken' is used essentially like *connâitre*. Unfortunately, this does not help us make the notational distinction we need.)

term t is understood at a world by having $t = t \in h(w)$. We require that understood equations be composed only of understood terms. Thus, $f(t_1, \ldots, t_k) = t \in h(w)$ only if $(t_1 = t_1), \ldots, (t_k = t_k) \in h(w)$, and $(t = t) \in h(w)$. Similarly, understood formulae should be composed of understood formulae containing understood terms. Thus, $\varphi \in h(w)$ only if all hereditary subformulae of φ are in $h(w)$ and $t = t \in h(w)$ for any constant term t occurring in φ. Finally, we require that h be closed under the general properties of equality. In other words, we require the following: $s = t \in h(w)$ and $\varphi \in h(w)$ imply $\varphi' \in h(w)$ and $\varphi'' \in h(w)$ (where φ' is the result of replacing some or all occurrences of t in φ with s, and φ'' is the result of replacing some or all occurrences of s in φ with t). These restrictions amount to our minimal closure conditions on rational awareness.

The function ι is a one-one function from terms in the language to members of the domain. Note that this means that every distinct name denotes a distinct object. It might seem that this would preclude one's knowing, for example, that Mark Twain is Samuel Clemens. Indeed, this would seem to preclude the possibility that Mark Twain be Samuel Clemens. That is not the case. Objects that are distinct in the domain may fail to be distinguished in particular worlds. The assignment function set out below requires that, in w, Robbie knows that Mark Twain is Samuel Clemens precisely when these names pick out the same thing at all worlds accessible from w for him. What ι being one-one allows us to express in our framework is the possibility that Robbie does not know that Mark Twain is Samuel Clemens. Recall that we have constant terms for denoting worlds in our language. The family of functions $\{\iota_w\}_{w \in W}$ determines the denotation of terms at each world. $\iota_w(t)$ is a set of individuals $\iota(s)$ such that $s = t \in h(w)$. In particular, ι_w partitions the set $\{\iota(s) : s = s \in h(w)\}$. This guarantees that all and only understood terms denote in our (epistemic) worlds. So, each ι_w gives rise to an equivalence relation on terms that determines which terms denote the same individuals at each world. Continuing our example from above, at all worlds w' accessible for Robbie from w, $\iota_{w'}$('Samuel Clemens') $=$ $\iota_{w'}$('Mark Twain') $=$ an equivalence class containing both ι('Samuel Clemens') and ι('Mark Twain'). Note that an individual in a world is a set of individuals in D: individuals in D are distinguished in the language but might not be in a particular world. If $t = t \notin h(w)$, then $\iota_w(t) = \emptyset$, which we understand as saying that t denotes nothing at that world. While technically t is assigned $\emptyset$ at w rather than nothing, this is a reasonably clear abuse of terminology. Each ι_w assigns values to tuples of terms such that $\iota_w(\langle t_1, \ldots, t_n \rangle) = \langle \iota_w(t_1), \ldots, \iota_w(t_n) \rangle$. Each ι_w also assigns a value to each predicate constant in the language. For an n-ary predicate constant P, $\iota_w(P) = \{\iota_w(\langle t_{1_1}, \ldots, t_{1_n} \rangle), \ldots, \iota_w(\langle t_{k_1}, \ldots, t_{k_n} \rangle)\}$, where $\iota_w(t_{j_m}) \neq \emptyset$. Note that while no member of a tuple in $\iota_w(P)$ may be nondenoting, it is possible that $\iota_w(P) = \emptyset$, in other words that P is nondenoting at w. ('Nondenoting' is understood for predicate constants in the same sense that it is understood for terms.) There will be a further restriction on the ι_w determined by the assignment function, which we will introduce at the appropriate place.

The function a assigns truth values to formulae in our language in the manner given below. We want to allow a to be undefined sometimes; thus we use the standard trick of adding a value '$*$' to represent being undefined. Also, a is entirely relative. That is, a only takes values at situations and thus is a function of two arguments, a situation and a formula. For convenience we generally write things in terms of the naturally induced family of situation projection functions associated with a. In other words, $a(w, \varphi) = a_w(\varphi)$ for all $w \in W$ and φ in the language.

We make one more constraint on rational awareness that relates h and a. Specifically, we want to require that any understood formula receives a determinate truth value. It might seem that a sentence can be understood without a decision as to its truth value. I might understand the claim that Bill Clinton is in Washington right now without knowing whether it is true or false. But, this is really only a reflection of our limited terminology. The understanding in this case is that of an individual not that of a situation. To say that I understand the claim but don't know whether it is true or false is to say that I accept as possible both a situation where it is true and one where it is false. But, each of these situations does attach a determinate truth value to the claim. Saying that formulae in the image of $h(w)$ are understood at w may thus be a little deceptive. I am open to suggestions for better terminology. Nothing in the individual cases for types of formulae below forces or precludes this constraint. Therefore, we stipulate up front that $\varphi \in h(w)$ implies either $a_w(\varphi) = \text{T}$ or $a_w(\varphi) = \text{F}$. (Note that the cases for formula types will imply that in the latter case $\neg\varphi \in h(w)$ as well.)

3.1.1 The Assignment Function

The a_w assign truth values to formulae as follows:

$$a_w(s = t) = \begin{cases} \text{T} & \text{if } s = t \in h(w) \text{ and } \iota_w(s) = \iota_w(t) \\ \text{F} & \text{if } s \neq t \in h(w) \text{ and } \iota_w(s) \neq \iota_w(t) \\ * & \text{otherwise} \end{cases}$$

$$a_w(C_i(t)) = \begin{cases} \text{T} & \text{if } \iota_{w'}(t) \neq \emptyset \text{ for all } w' \text{ s.t. } wR_iw' \\ \text{F} & \text{if } \iota_w(t) \neq \emptyset \text{ and } \iota_{w'}(t) = \emptyset \text{ for some } w' \text{ s.t. } wR_iw' \\ * & \text{otherwise} \end{cases}$$

For k-ary predicate letters P, other than equality and C_i (for $i = 1, \ldots, n$), we have

$$a_w(P(t_1, \ldots, t_k)) = \begin{cases} \text{T} & \text{if } \iota_w(\langle t_1, \ldots, t_k \rangle) \in \iota_w(P) \text{ and} \\ & P(t_1, \ldots, t_k) \in h(w) \\ \text{F} & \text{if } \iota_w(\langle t_1, \ldots, t_k \rangle) \notin \iota_w(P) \text{ and} \\ & \neg P(t_1, \ldots, t_k) \in h(w) \\ * & \text{otherwise} \end{cases}$$

For an arbitrary sentence φ,

$$a_w(S_i\varphi) = \begin{cases} \text{T} & \text{if } S_i\varphi \in h(w) \text{ and} \\ & \quad a_{w'}(\varphi) = \text{T, for all } w' \text{ s.t. } wR_iw' \\ \text{F} & \text{if } \neg(S_i\varphi) \in h(w) \text{, and} \\ & \quad a_{w'}(\varphi) \text{ is defined for all } w' \text{ s.t. } wR_iw', \\ & \text{and } a_{w'}(\varphi) = \text{F for some } w' \text{ s.t. } wR_iw' \\ * & \text{otherwise} \end{cases}$$

$$a_w(\forall x\varphi) = \begin{cases} \text{T} & \text{if } \forall x\varphi \in h(w) \text{ and} \\ & \quad a_w(\varphi[t/x]) = \text{T, for all } t \text{ s.t. } \iota_w(t) \neq \emptyset \\ \text{F} & \text{if } \neg\forall x\varphi \in h(w) \text{ and} \\ & \quad a_w(\varphi[t/x]) = \text{F, for some } t \text{ s.t. } \iota_w(t) \neq \emptyset \\ * & \text{otherwise} \end{cases}$$

where $\varphi[t/x]$ is the same sentence as φ except that all free occurrences of x in φ are replaced by t

$$a_w(\varphi \wedge \psi) = \begin{cases} \text{T} & \text{if } (\varphi \wedge \psi) \in h(w) \text{ and} \\ & \quad a_w(\varphi) = \text{T and } a_w(\psi) = \text{T} \\ \text{F} & \text{if } \neg(\varphi \wedge \psi) \in h(w) \text{ and} \\ & \quad a_w(\varphi) = \text{F or } a_w(\psi) = \text{F} \\ & \text{and both } a_w(\varphi) \text{ and } a_w(\psi) \text{ are defined} \\ * & \text{otherwise} \end{cases}$$

$$a_w(\varphi \vee \psi) = \begin{cases} \text{T} & \text{if } (\varphi \vee \psi) \in h(w) \text{ and} \\ & \quad a_w(\varphi) = \text{T or } a_w(\psi) = \text{T} \\ & \text{and both } a_w(\varphi) \text{ and } a_w(\psi) \text{ are defined} \\ \text{F} & \text{if } \neg(\varphi \vee \psi) \in h(w) \text{ and} \\ & \quad a_w(\varphi) = \text{F and } a_w(\psi) = \text{F} \\ * & \text{otherwise} \end{cases}$$

$$a_w(\varphi \rightarrow \psi) = \begin{cases} \text{T} & \text{if } (\varphi \rightarrow \psi) \in h(w) \text{ and} \\ & \quad a_w(\varphi) = \text{F or } a_w(\psi) = \text{T} \\ & \text{and both } a_w(\varphi) \text{ and } a_w(\psi) \text{ are defined} \\ \text{F} & \text{if } \neg(\varphi \rightarrow \psi) \in h(w) \text{ and} \\ & \quad a_w(\varphi) = \text{T and } a_w(\psi) = \text{F} \\ * & \text{otherwise} \end{cases}$$

$$a_w(\neg\varphi) = \begin{cases} \text{T} & \text{if } \neg\varphi \in h(w) \text{ and } a_w(\varphi) = \text{F} \\ \text{F} & \text{if } \neg\neg\varphi \in h(w) \text{ and } a_w(\varphi) = \text{T} \\ * & \text{otherwise} \end{cases}$$

The assignment function for formulae of the form $w \models \varphi$ can now be properly expressed using the h function and the ι function as,

$$a_w(w' \models \varphi) = \begin{cases} \text{T} & \text{if } \iota_w(w') \neq \emptyset,\ (w'' \models \varphi) \in h(w), \text{ and} \\ & \quad a_{w''}(\varphi) = \text{T for all } w'' \in \iota_w(w') \\ \text{F} & \text{if } \iota_w(w') \neq \emptyset,\ \neg(w'' \models \varphi) \in h(w), \text{ and} \\ & \quad \text{either } a_{w''}(\varphi) = \text{F or } a_{w''}(\varphi) = * \\ & \quad \text{for all } w'' \in \iota_w(w') \\ * & \text{otherwise} \end{cases}$$

What is the relationship between the understood terms of w, $\{t : t = t \in h(w)\}$, and the terms in the domain of w, $d(w) = \{\iota_w(t) : \iota_w(t) \neq \emptyset\}$? Under the constraints set out above, $(t = t) \in h(w)$ if and only if $\iota_w(t) \in d(w)$. But, this need not be the case. We could constrain h and d such that $t = t \in h(w)$ only if $\iota_w(t) \in d(w)$ while allowing that the converse may fail to hold. Or, we might require only the converse. Space limitations preclude a detailed discussion of these issues. We simply note that the options might allow us to pursue motivations that are rather different from ours, those of free logic. In particular, free logics are concerned with the ability to talk meaningfully about that which does not exist, and this is intimately tied to their use of existential quantification. Given that our interests and our use of domains are different from those of free logics, handling existence in the sense that distinguishes the fictional from the real is trivial for us. We could simply introduce an ordinary unary predicate letter, say 'E', to pick out those things that exist from those things that don't. For us 'E' is not tied in any way to existential quantification, even though our account of quantification derives largely from some approaches to free logic. (Cf. [Ben86] and [Gar84].) For ease of expression we do define a predicate given by 'M' that is tied to existential quantification (by definition), viz: $M(t) =_{\mathrm{df}} \exists x(x = t)$, where x is a variable distinct from t. This will appear in some of the axioms and rules below. In free logic, a similar definition is a way of getting at what exists; thus, they might use 'E' rather than 'M'. But, we are trying to get at what is understood, known—in the sense of the C-predicates. Therefore, we have chosen 'M' as a mnemonic for terms that are *meaningful* in an epistemic situation.[5]

4 The Logic

Axioms are the universal closures of the following.

1. $\varphi \to (\psi \to \varphi)$
2. $(\varphi \to (\psi \to \gamma)) \to ((\varphi \to \psi) \to (\varphi \to \gamma))$
3. $(\neg\psi \to \neg\varphi) \to (\neg\psi \to (\varphi \to \psi))$
4. $S_i(\varphi) \wedge S_i(\varphi \to \psi) \to S_i(\psi)$
5. $S_i(\varphi) \to \varphi$
6. $\forall x(x = x)$
7. $\forall x \forall y((x = y) \to (\varphi \to \varphi'))$

   (where φ' is the result of replacing no, some or all occurrences of 'x' in φ with 'y')
8. $S_i\varphi \to C_i t$

   (where φ is an arbitrary sentence and t is any term occurring freely in φ)
9. $(w \models (\varphi \wedge \psi)) \to ((w \models \varphi) \wedge (w \models \psi))$
10. $(w \models (\varphi \vee \psi)) \to ((w \models \varphi) \vee (w \models \psi))$
11. $(w \models \neg\varphi) \to \neg(w \models \varphi)$
12. $(w \models \varphi) \to ((w \models \psi) \vee (w \models \neg\psi))$

    (where ψ is any hereditary subformula of φ)
13. $(w \models (\varphi \to \psi)) \to (w \models \varphi) \to (w \models \psi)$
14. $(w \models \varphi) \to (w \models (t = t))$

[5]This should in no way be confused with the use of 'M' for a possibility operator as in, e.g., [HC68].

(where φ is an arbitrary sentence and t is any term occurring freely in φ)

15. $(w \models S_i\varphi) \rightarrow (w \models \varphi)$

16. $((w \models S_i\varphi) \wedge (w \models S_i(\varphi \rightarrow \psi)) \rightarrow (w \models S_i\psi)$

17. $(w \models \forall x\varphi) \rightarrow (w \models t = t) \rightarrow (w \models \varphi[t/x])$

18. $(w \models C_i t) \rightarrow (w \models S_i\forall x\varphi) \rightarrow (w \models S_i\varphi[t/x])$

19. $(w \models v \models \varphi) \rightarrow (v \models \varphi)$

20. $(w \models \varphi) \rightarrow (w \models \psi)$

    (where ψ is any axiom that is a hereditary subformula of φ)

21. $(w \models \varphi) \rightarrow \psi$

    (where ψ is any axiom that is a hereditary subformula of φ)

22. $(w \models \psi \rightarrow (Mt \rightarrow \varphi)) \wedge ((w \models \forall x\varphi[x/t]) \vee (w \models \neg\forall x\varphi[x/t])) \rightarrow$
    $(w \models \psi \rightarrow \forall x\varphi[x/t])$

    (where t is any term that does not occur freely in ψ)

Many of the 'new' axioms are present to insure that logical behavior of the satisfaction relation respects standard intuitions. The one that may seem least intuitive is axiom 19. Roughly, it says that if a situation w says that φ is true in another situation v, then φ is true in v. It might seem that this is too high a price to pay for expressibility. Why should what is true in one world force what is true in another?

There are a few sufficient answers to this. In the interest of brevity we give only one. This axiom simply insures that we don't use the same name to refer to distinct situations u and v in distinct situations w and w'. We don't account for the possibility that two situations mistakenly refer to different situations as the same. Situations are referred to in both the object language and the metalanguage. Thus, they must respect the metalinguistic truths of some theoretical perspective. And, it is not possible to consistently respect two conflicting perspectives for the naming of situations.

The logical rules are as follows:

R1. (Modus Ponens) From φ and $\varphi \rightarrow \psi$ infer ψ.

R2. (Universal Instantiation) From $\forall x\varphi \wedge Mt$ infer $\varphi[t/x]$.

(for any term t, where φ is a sentence in the language, and $\varphi[t/x]$ is the same sentence as φ except that all free occurrences of x in φ are replaced by t)

R3. (Universal Generalization) From $\psi \rightarrow (Mt \rightarrow \varphi)$ and $\forall x\varphi[x/t] \vee \neg\forall x\varphi[x/t]$ infer $\psi \rightarrow \forall x\varphi[x/t]$.

(where t is any term that does not occur freely in ψ or in any assumption on which $\psi \rightarrow (Mt \rightarrow \varphi)$ depends)

5 Metalogic

To describe our metalogical results we need to adopt the following notational conventions. Let Γ stand for a finite set of sentences and φ, ψ, etc. stand for arbitrary sentences as before. '$\Gamma \vdash \varphi$' means that φ is derivable from Γ and the axioms using the inference rules of the logic. '$\Gamma \mid\models \varphi$' means that, in all models, φ is true at all the worlds at which all the members of Γ are true. The more standard notation corresponding to our '$\mid\models$' is '$\models$'. But, this notation is usually assigned double duty. It is used in the way we have used '$\mid\models$', and it is used to mean that a formula is true at a particular world or in a particular model. The reason for this is unclear; however, part of the justification may stem from the following. To represent that a formula φ is true at all worlds in all models we usually write '$\models \varphi$'. But, a logic and semantics are generally designed so that axioms are true at all worlds in all models. (After all, these formulae are *axioms*.) Thus, if Γ is the set of axioms, it is somewhat redundant to write '$\Gamma \models \varphi$' since the axioms are true in all the models. Thus, the two uses coincide for logical truths. This confusion of notation is thus somewhat forgivable. But, in this paper we consider possibilities in which some individuals may not know all of the axioms. This means that some of them may not be true at all of the possible worlds. To keep this clear we have adopted the above usage of '$\mid\models$'.

Our soundness theorem is slightly different from the usual one because it is possible for axioms to fail to be true in a situation when they are not understood there. Therefore, it is possible to derive a result using formulae that are not understood at a world, hence not true, and thus to arrive at a conclusion that is also not understood at that world. Nonetheless, every axiom φ is *weakly valid* in the sense that, for all worlds w in all models, $\varphi \in h(w)$ implies $a_w(\varphi) = \mathrm{T}$. Also, throughout this paper it is assumed that we are dealing only with finite sets of formulae. (Thus, compactness is simply a non-issue.) Syntactic derivation is generally taken as a finite phenomenon. But, given the cognitive nature of our situations, it is only natural to view semantic consequence as finite also.

Theorem 5.1 (Soundness) If $\Gamma \vdash \varphi$ and $\varphi \in h(w)$ for all worlds w such that $\Gamma \subseteq h(w)$, then $\Gamma \mid\models \varphi$.

Proof: Cf. [Syv93]. □

Theorem 5.2 (Completeness) If $\Gamma \mid\models \varphi$, then $\Gamma \vdash \varphi$.

Proof: Cf. [Syv93]. □

6 Adding Knowledge of Situations to the Logic

A major goal of this paper was to introduce a logic that is adequate to express and reason about common knowledge. The astute reader will have noticed that the language set out above is not expressive enough to do the job. We have yet to introduce a means to represent knowing that a situation obtains. Actually, it is possible to represent such knowledge with an eliminable syntactic definition. If we represent that agent i knows that v is the situation by $S_i v$, then we can define this as follows:

$$S_i v \leftrightarrow \bigwedge_{\varphi \in h(v)} ((v \models \varphi) \rightarrow (S_i\varphi \wedge S_i(v \models \varphi)))$$

Since $h(v)$ is finite for all v and since the formulae on the right are all well formed, $S_i v$ can always be completely eliminated from any formula (or finite set of formulae) and replaced by the longer expression. The semantics for such formulae are thus determined by the above definition as follows.

$$a_w(S_i v) = \begin{cases} \text{T} & \text{if } S_i v \in h(w), \text{ and for all } \varphi \text{ such that } a_v(\varphi) = \text{T} \\ & a_w(S_i\varphi) = \text{T and } a_w(S_i(v \models \varphi)) = \text{T} \\ \text{F} & \text{if } \neg S_i v \in h(w), \text{ and for some } \varphi \text{ such that } a_v(\varphi) = \text{T}, \\ & a_w(S_i\varphi) = \text{F or } a_w(S_i(v \models \varphi)) = \text{F} \\ * & \text{otherwise} \end{cases}$$

Letting '$S_{ij}\varphi$' represent common knowledge between i and j of φ, we can then give a syntactic definition of common knowledge corresponding to Barwise's shared-situation account in [Bar89], viz:

$$S_{ij}\varphi \leftrightarrow \exists w(S_i w \wedge S_j w \wedge w \models S_i w \wedge w \models S_j w \wedge w \models \varphi)$$

It may thus seem that, with the definition of formulae of the form $S_i w$ in place, we can claim to have a sound and complete logic that can represent common knowledge. But that is too hasty. If we only use these formulae as abbreviations, we do have a sound and complete logic that can represent knowledge of situations. Once we allow these formulae as proper formulae of the language we are in a position where truth assignments may be nonwellfounded. For determination of truth value depends on the truth values of all the understood formulae in a situation. If $S_i v \in h(v)$, then to determine if $S_i v$ is true at v we must first know if all the formulae true at v are known, including $S_i v$ itself.

The problem is that there are different partitions of $h(v)$ into true and false formulae (at v and possibly elsewhere) that will satisfy the definition of the assignment. The solution is to stipulate that we always choose the one that yields the largest set of true formulae.[6] This is not to say that there is a unique largest set any more than it would for wellfounded truth assignments. Given a formula $(\varphi \vee \psi) \wedge \neg(\varphi \wedge \psi)$ with no other relevant constraints on φ and ψ there are two possible assignments making as many formulae as possible true amongs the hereditary subformulae of $(\varphi \vee \psi) \wedge \neg(\varphi \wedge \psi)$. But, for example, even if we must assign truth values to $S_i v$ and $S_i w$ subject to the constraint $\neg(S_i v \wedge S_i w)$, it is still possible to consistently make both of them false or only one. Our understanding of the assignment function is to make formulae of the form $S_i v$ true when it is consistent with our other decisions to do so.

We can capture this via a slightly more complicated assignment function as follows.

$$a_w(S_i v) = \begin{cases} \text{T} & \text{if } S_i v \in h(w), \text{ and for all } \varphi \text{ such that } a_v(\varphi) = \text{T} \\ & \quad a_w(S_i\varphi) \neq \text{F and } a_w(S_i(v \models \varphi)) \neq \text{F}, \\ & \text{and there exists a } w' \text{ such that } a_w(S_i w') = \text{T} \\ & \quad \text{and for all } \varphi \text{ such that } a_v(\varphi) = \text{T}, \ a_{w'}(\varphi) = \text{T} \\ \text{F} & \text{if } \neg S_i v \in h(w), \text{ and for all } w' \text{ such that } a_w(S_i w') = \text{T} \\ & \text{there exists a } \varphi \text{ such that } a_v(\varphi) = \text{T and } a_{w'}(\varphi) = \text{F} \\ * & \text{otherwise} \end{cases}$$

[6] Our approach has many similarities to that taken by Barwise in [Bar89] when he gives his coinductive characterization of satisfaction.

We can now add the above (no longer eliminable) definition as an axiom.

23. $S_i v \leftrightarrow \bigwedge_{\varphi \in h(v)} ((v \models \varphi) \rightarrow (S_i\varphi \wedge S_i(v \models \varphi)))$

The resulting logic is sound and complete with respect to the same semantics, with the addition to the assignment function just given. (Cf. [Syv93] for details.)

7 Conclusion

In this paper we have presented an epistemic logic with some interesting features, including: the ability to represent familiarity with individuals, the reflection of finite cognitive capacity, and the ability to syntactically capture the shared-situation account of common knowledge. This logic is also sound and complete with respect to the presented semantics. Given our unusual worlds, domains, etc. we have maintained a fairly traditional epistemic modality and corresponding accessibility relation. An interesting open question is to look at other modalities that seem to be naturally engendered by our approach. In particular, it should be interesting to look at a modality of understanding where a formula φ is understood at w by Eleri precisely when it is in $h(w')$ for all w' that are accessible for her from w.

References

[Bar89] Jon Barwise. On the Model Theory of Common Knowledge. In *The Situation in Logic*, volume 17 of *CSLI Lecture Notes*, chapter 9, pages 201–220. CSLI Publications, Stanford, 1989. A preliminary version of this appeared as "Three Views of Common Knowledge" in *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, 1988, pp. 365–379.

[Ben86] Ermanno Bencivenga. Free Logics. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, vol. III: Alternatives to Classical Logic*, volume 166 of *Synthese Library*, chapter III.6, pages 373–426. D. Reidel, 1986.

[Che80] Brian F. Chellas. *Modal Logic: An Introduction.* Cambridge University Press, Cambridge, 1980.

[Ebe74] Rolf Eberle. A Logic of Believing, Knowing, and Inferring. *Synthese*, 26:356–382, 1974.

[FH85] Ronald Fagin and Joseph Y. Halpern. Belief Awareness and Limited Reasoning: Preliminary Report. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence, vol. 1*, pages 491–501. William Kaufmann Inc., Los Altos, California, 1985.

[FH88] Ronald Fagin and Joseph Y. Halpern. Belief Awareness and Limited Reasoning. *Artificial Intelligence*, 34:39–76, 1988. This issue actually came out in December 1987, though the article is labelled with '1988'. A preliminary version of this article was published as [FH85].

[Gar84] James W. Garson. Quantification in Modal Logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, vol. II: Extensions of Classical Logic*, volume 165 of *Synthese Library*, chapter II.5, pages 249–307. D. Reidel, 1984.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In *Proceedings of the Seventeenth Annual ACM Symposium on the Theory of Computing*, pages 291–304. ACM Press, New York, 1985.

[HC68] G.E. Hughes and M.J. Cresswell. *An Introduction to Modal Logic.* Methuen and Co., London, 1968.

[Hin62] Jaakko Hintikka. *Knowledge and Belief: An Introduction to the Logic of Two Notions.* Cornell University Press, Ithaca, N.Y., 1962.

[HM90] Joseph Y. Halpern and Yoram Moses. Knowledge and Common Knowledge in a Distributed Environment. *Journal of the ACM*, 37(3):549–587, July 1990.

[HM92] Joseph Y. Halpern and Yoram Moses. A guide to the completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, April 1992.

[HV86] Joseph Y. Halpern and Moshe Y. Vardi. The Complexity of Reasoning about Knowledge and Time: Extended Abstract. In *Proceedings of the Eighteenth Annual ACM Symposium on the Theory of Computing*, pages 304–315. ACM Press, New York, 1986.

[Lev85] Hector J. Levesque. A Logic of Implicit and Explicit Belief. In *AAAI-84, Proceedings of the Conference on Artificial Intelligence*, pages 198–202. William Kaufmann Inc., Los Altos, California, 1985.

[Lev90] Hector J. Levesque. All I Know: A Study in Artificial Intelligence. *Artificial Intelligence*, 42:263–309, 1990.

[Lew69] David Lewis. *Convention: A Philosophical Study.* Harvard University Press, Cambridge, Mass., 1969.

[Syv90] Paul F. Syverson. A Logic for the Analysis of Cryptographic Protocols. Formal Report 9305, NRL, December 1990.

[Syv93] Paul F. Syverson. *Logic, Convention, and Common Knowledge.* PhD thesis, Indiana University, 1993.

[Var89] Moshe Y. Vardi. On the Complexity of Epistemic Reasoning. In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science*, pages 243–252. IEEE Computer Society Press, Washington, D.C., 1989.