Justification Logics, Logics of Knowledge, and Conservativity*

Melvin Fitting
Lehman College, CUNY, 250 Bedford Park Boulevard West, Bronx, NY 10468-1589
CUNY Graduate Center, 365 Fifth Avenue, New York, NY 10016
melvin.fitting@lehman.cuny.edu

October 23, 2008

Abstract

Several *justification logics* have been created, starting with the logic LP, [1]. These can be thought of as explicit versions of modal logics, or of logics of knowledge or belief, in which the unanalyzed necessity (knowledge, belief) operator has been replaced with a family of explicit justification terms. We begin by sketching the basics of justification logics and their relations with modal logics. Then we move to new material. Modal logics come in various strengths. For their corresponding justification logics, differing strength is reflected in different vocabularies. What we show here is that for justification logics corresponding to modal logics extending T, various familiar extensions are actually conservative with respect to each other. Our method of proof is very simple, and general enough to handle several justification logics not directly corresponding to distinct modal logics. Our methods do not, however, allow us to prove comparable results for justification logics corresponding to modal logics that do not extend T. That is, we are able to handle explicit logics of knowledge, but not explicit logics of belief. This remains open.

1 Introduction

This paper is partly an advertisement for justification logics, and partly a presentation of some technical results of interest concerning them. Justification logics are a relatively new area within the general field of logics of knowledge and modal logics, so some advertising is appropriate. They provide a natural finer-grained analysis than Hintikka-style logics do, yet relate to them in precise and useful ways. They are becoming appropriate tools for several purposes, and there is much current investigation of them. As to the technical results, we will show conservativity theorems relating the modal logics T, S4, and S5. (We actually show something more general, but this will do for now.) It might be asked what sense conservativity issues make for these logics since they have the same vocabulary. It will be seen that behind the modal operator $\Box$, or K as it will generally be written here, there is hidden, explicit machinery, supplied by justification logics. When this is taken into account, conservativity results can be easily stated, and demonstrated.

*This paper is an extended version of [10]

2 Logics of Knowledge

Reasoning about knowledge is often done using the machinery of modal logic, following the lead of [13]. Multiple agent situations make use of multi-modal logic, but here we only consider the mono-modal case. With only one agent, it is standard to use K for the necessity operator, reading KX as *it is known (by the agent) that* X. We take $\supset$ and $\bot$ as primitive, with other connectives defined in the usual way. Then the simplest logic of knowledge is axiomatized by the following.

Axiom Schemes

K1 All instances of classical tautologies

K2 $K(X \supset Y) \supset (KX \supset KY)$

K3 $KX \supset X$

Rules of Inference

Modus Ponens $\dfrac{X \quad X \supset Y}{Y}$

Necessitation $\dfrac{X}{KX}$

Axiom **K2** says agents are capable of reasoning, and can apply *modus ponens*. Axiom **K3** says that it is knowledge we are considering—what is known is so. If this is dropped we have a logic of belief, not of knowledge. The **Necessitation** rule embodies the idea that all logical truths are known.

Commonly, additional axioms are added to these basic ones.

Axiom Schemes

Positive Introspection $KX \supset KKX$

Negative Introspection $\neg KX \supset K\neg KX$

Positive Introspection says that if we know something, we know we know it. Negative Introspection says if we don't know something, we know we don't know it. It is common in the literature for both of these to be assumed, though many applications do not actually need this full strength. The basic system is the familiar modal logic T; with positive introspection added it is $\mathsf{S4}$, and with negative introspection also added it is $\mathsf{S5}$.

Semantics is the familiar Kripke/Hintikka possible worlds version. A model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \Vdash \rangle$ consists of a collection $\mathcal{G}$ of *states of knowledge*, an accessibility relation $\mathcal{R}$ on them, and a notion of truth at a state, symbolized here as $\mathcal{M}, \Gamma \Vdash X$, where $\mathcal{M}$ is a model, Γ is a state, and X is a formula. On propositional connectives $\Vdash$ is truth-functional at each world, and we have the standard condition for the modal operator.

$$\mathcal{M}, \Gamma \Vdash KX \iff \mathcal{M}, \Delta \Vdash X \text{ for all } \Delta \in \mathcal{G} \text{ with } \Gamma \mathcal{R} \Delta \tag{1}$$

This is usually understood as saying that an agent knows X at state Γ if X is the case at all states the agent cannot distinguish from Γ.

If only a reflexivity condition is placed on $\mathcal{R}$, the semantics corresponds to the basic axiom system given above, the modal logic T. Positive Introspection, $\mathsf{S4}$, requires transitivity, and positive plus negative introspection, $\mathsf{S5}$, also requires symmetry. All this is widely familiar machinery.

This Hintikka approach has been successful but it is not without problems, many of them noted by Hintikka himself. What is it that an agent has knowledge of, sentences or propositions? Hintikka's systems work well as logics of propositions—in them if $X \equiv Y$ is provable, so is $KX \equiv KY$. But two sentences might be equivalent and so express the same proposition, while that equivalence might not be easy to see. What we communicate directly is sentences, and propositions only indirectly. Much important mathematics amounts to proving that two quite different sentences are equivalent—in some way they express the same proposition. In the Hintikka approach, sentences and propositions are in a kind of uneasy tension.

In addition we run into the well-known problems of *logical omniscience*. The first of these originates with the **Necessitation Rule**. According to this an agent knows all tautologies, even those with as many symbols as there are atoms in the universe! This is not a problem if we think of knowledge logics as being about propositions, but it certainly is if we think of them as being about sentences. The second omniscience problem comes from axiom **K2**. It follows from this that an agent would know the consequences of what it knows. We know the axioms of set theory, so we know all the theorems as well! (Properly speaking, this assumes a quantified logic and not a propositional one, but the underlying principle is the same. Quantified versions of justification logics are currently under development.)

Commonly one says we are not actually dealing with a logic of knowledge, but with a logic of *potential* knowledge. KX really means that X is *knowable*, not that it is actually known. In fact a true logic of knowledge for sentences may be unattainable. We might not know something simply because we haven't thought about it, and it's very hard to see a pattern in this that could be captured in a formal logic. *Awareness logic*, [6], attempts to do just that, but essentially it simply adds a new predicate A, like K, to the language where AX asserts that X is something we have thought about—we are aware of X. This works, of course, but the construction of models is rather arbitrary. Constraints come from the outside, and do not emerge directly from any logical structure. We will not discuss the awareness approach further here.

3 Justification Logics Axiomatically

Justification Logics are modal-like logics with a family of *justification terms*, intended to represent explicit reasons. They originated with work of Artemov on the arithmetic semantics of propositional intuitionistic logic, [1], and since then the field has broadened considerably. See [12] for a brief history of the subject. As with logics of knowledge, they constitute a range of logics of differing strengths. Here we sketch the standard terminology and other basics.

As just noted, the language of justification logics makes use of a family of *justification terms*. (In earlier work these were usually called *proof terms*.) Justification terms are built up from *variables*: $x_1, x_2, \ldots$; and *constant symbols*: $c_1, c_2, \ldots$. They are built up using the *operation symbols*: $+$ and $\cdot$, both binary, and ! and ?, both unary. These are used as infix and prefix, respectively. Not all operators may be present in a particular justification logic. The operation $\cdot$ is an application operation. The intention is, if t is a justification of $X \supset Y$ and u is a justification of X then $t \cdot u$ is a justification of Y. The operation $+$ combines justifications, $t + u$ justifies whatever t justifies and also whatever u justifies. ! is a positive verifier, if t justifies X then $!t$ justifies the fact that t justifies X. And ? is a negative verifier, if t does not justify X then $?t$ justifies that fact.

Formulas are built up from *propositional letters*: $P_1, P_2, \ldots$, and a *falsehood constant*, $\bot$, using $\supset$ in the usual way, together with an additional rule of formation, $t{:}X$ is a formula provided t is a justification term and X is a formula. Other connectives may be defined as usual. We assume a tight binding for the colon operator, and omit parentheses when possible. Thus, for example,

$t{:}X \supset X$ should be understood as $(t{:}X) \supset X$.

Here is a list of axiom schemes (though we use the term 'axiom' for short) from which we will pick and choose.

Classical Axioms	all tautologies
Truth Axioms	$t{:}X \supset X$
	$(X \supset Y) \supset (t{:}X \supset Y)$
	$t{:}(X \supset Y) \supset (u{:}X \supset Y)$
+ Axioms	$t{:}X \supset (t+u){:}X$
	$u{:}X \supset (t+u){:}X$
· Axiom	$t{:}(X \supset Y) \supset (u{:}X \supset (t \cdot u){:}Y)$
! Axiom	$t{:}X \supset !t{:}t{:}X$
? Axiom	$\neg t{:}X \supset ?t{:}\neg t{:}X$

The Truth Axioms include two that are not standard. The last two can be shown to be easy consequences of the first Truth Axiom using classical logic. Likewise as Classical Axioms we assume all tautologies, though a finite set is possible. Both of these peculiarities arise for the same reason, and have to do with the role of constants in justification logics. Assuming these extra axioms simplifies some parts of proofs later on. Eliminating the extra axioms will be discussed more fully in Section 9.

For rules, of course we have the standard one.

$$\textbf{Modus Ponens} \quad \frac{X \qquad X \supset Y}{Y}$$

Finally there is a version of the modal necessitation rule, and here there is some non-uniformity. Constant symbols serve as justifications for truths that we do not further analyze, but our ability to analyze depends on available machinery and so we have three different versions, each stronger than the one before.

Definition 3.1 The following are versions of a *Constant Necessitation* rule.

Axiom Necessitation If X is an axiom and c is a constant, then $c{:}X$ is a theorem.

Iterated Axiom Necessitation If X is an axiom and $c_1, c_2, \ldots, c_n$ are constants, then $c_1{:}c_2{:} \ldots c_n{:}X$ is theorem.

Theorem Necessitation If X is a theorem and $c_1, c_2, \ldots, c_n$ are constants, then $c_1{:}c_2{:} \ldots c_n{:}X$ is theorem.

If we assume as axioms all except those for ! and ?, and as rules **Modus Ponens** and **Iterated Axiom Necessitation**, we have the justification logic called JT, an analog of T. If we add the axiom for !, and replace iterated axiom necessitation by **Axiom Necessitation** we have what could be called JS4, but which is called LP instead, for historical reasons. LP stands for "logic of proofs," reflecting its original application as a logic of explicit proofs in formal arithmetic. This logic is an analog of S4. Finally if we also add the axiom for ? we have JS5, an analog of S5.

If X is a theorem of any of the implicit logics of knowledge T, S4, or S5, so is KX. In fact this is almost always taken as a rule of derivation in axiomatic formulations, as we have done here. Corresponding to this is an important feature of justification logics called *internalization.* If X is a theorem of any of JT, LP, or JS5, then there is a closed justification term t such that $t{:}X$ is also a theorem. In fact, t can be produced constructively from any axiomatic proof of X. In a very real sense, t *internalizes* the axiomatic proof. The Internalization result is due to Artemov, [1]. Justification logics internalize their own proof structure, but the construction of an internalizing term t requires a certain minimal amount of machinery. Axioms themselves are never the result of elaborate proofs—we simply assume them. This is embodied in the **Axiom Necessitation** rule above, the weakest of the three rule versions. Suppose we have this rule and we are working with a justification logic with ! available, that is, we have the ! axiom. Then if X is an axiom it has a constant justification, so we have $c{:}X$, this in turn has a justification, $!c{:}c{:}X$, this has its justification, $!!c : !c : c : X$, and so on. But if ! is not part of the machinery we cannot take this route, and so **Iterated Axiom Necessitation** is assumed instead, otherwise Internalization cannot be established. Finally if we have a really weak justification logic not containing $\cdot$, which is weaker than any we have seen so far, we lack machinery to analyze anything complex, and the **Theorem Necessitation** version will be assumed—everything provable has a justification about which nothing very interesting can be said.

Justification logics were formulated axiomatically above. Sequent calculus formulations also exist—one is given in [1] for instance—and hence also tableau formulations. Unfortunately, all current versions include a rule that does not obey the subformula principle, and this somewhat limits their uses.

It might be noted that particular combinators can be introduced as constants. For instance, the K combinator can be represented by a constant symbol k, where $k{:}(A \supset (B \supset A))$ is taken to be a theorem introduced via a Constant Necessitation rule. Then the $\cdot$ operation corresponds to application of combinators, and thus combinatory logic embeds into a fragment of justification logic. This was pointed out by Artemov, in [1].

Finally, note that justification terms can be used to limit the problem of logical omniscience. We might restrict ourselves to using formulas that do not involve justification terms that are 'too complex.' This could be measured in terms of number of symbols, or nesting depth of terms, or some other way. The point is, simply, that justification terms provide us with natural machinery for the measurement of complexity.

4 Justification Logic Semantics

The first semantics for justification logics is from [15]. In [8] a possible world semantics appeared, and this is now the most common semantics—the semantics of [15] can be seen as a single-world version of it. A model is $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{A}, \Vdash \rangle$, where $\mathcal{G}$ and $\mathcal{R}$ are a state set and an accessibility relation, as usual, with $\mathcal{R}$ reflexive, and possibly also transitive or symmetric depending on the particular justification logic being considered. The relation $\Vdash$ behaves on propositional connectives in the standard way—its modal behavior will be discussed shortly. The new item is $\mathcal{A}$, which is an *admissible justification* function. The idea is, $\mathcal{A}$ assigns to each justification term t and each formula X a set of worlds—those worlds in which t is considered to be relevant evidence for X. Relevant evidence is not to be understood as conclusive evidence. The date of publication listed on the title page of a book is relevent evidence for the date on which the book was published, though it could be in error (this happened to me once in the first, scrapped, printing of one of my books). The color of the binding of a book would not be relevant evidence for date of publication. There are a variety of conditions that are imposed on admissible justification functions, depending on the particular logic in question. Here is the full list.

Monotonicity $\Gamma\mathcal{R}\Delta$ and $\Gamma \in \mathcal{A}(t, X)$ implies $\Delta \in \mathcal{A}(t, X)$

Application $\mathcal{A}(t, X \supset Y) \cap \mathcal{A}(u, X) \subseteq \mathcal{A}(t \cdot u, Y)$

Weakening $\mathcal{A}(s, X) \cup \mathcal{A}(t, X) \subseteq \mathcal{A}(s + t, X)$

Postive Checking $\mathcal{A}(t, X) \subseteq \mathcal{A}(!t, t{:}X)$

Negative Checking $\overline{\mathcal{A}(t, X)} \subseteq \mathcal{A}(?t, t{:}X)$

For JT the Application and Weakening conditions are required. For LP, the Monotonicity and Positive Checking conditions are added, and for JS5 the Negative Checking condition is also added.

The modal condition on $\Vdash$ concerns the behavior of justification terms. It is the counterpart of (1) for standard logics of knowledge.

$$\begin{aligned} \mathcal{M}, \Gamma \Vdash t{:}X \Longleftrightarrow \mathcal{M}, \Delta \Vdash X \text{ for all } \Delta \in \mathcal{G} \text{ with } \Gamma\mathcal{R}\Delta \\ \text{and } \Gamma \in \mathcal{A}(t, X) \end{aligned} \tag{2}$$

In short, we have $t{:}X$ at Γ if X is knowable at Γ in the Hintikka sense, and t is an admissible justification for X at Γ. If we think of Hintikka semantics as capturing the idea of *true belief*, then what this machinery captures is *justified* true belief.

There is also a stronger version of the semantics. A model $\mathcal{M}$ is said to be *fully explanatory* provided, if $\mathcal{M}, \Delta \Vdash X$ for all $\Delta \in \mathcal{G}$ with $\Gamma\mathcal{R}\Delta$ then there is some justification t such that $\mathcal{M}, \Gamma \Vdash t{:}X$. More informally, $\mathcal{M}$ is fully explanatory provided knowability of X at Γ (in the Hintikka sense) implies there is a justification for X at Γ. Under simple and reasonable conditions, designed to ensure that constants behave in corresponding ways semantically and proof theoretically, provability agrees with truth at all worlds of all models, and this agrees with truth at all worlds of all fully explanatory models, [8], and [18] for JS5.

5 Explicit and Implicit Knowledge

We have said that the logics T and JT are analogous, and similarly for S4 and LP, and S5 and JS5. It is time to make this more precise. We can think of the modal/Hintikka versions as those in which the representation of knowledge is an implicit one, while the justification versions make this representation explicit. The relationship between implicit and explicit logics of knowledge can be seen as involving information hiding. We can think of the operator K as a kind of existential quantifier—KX says that there is a justification for X, but details are not being revealed.

Suppose we take a formula of one of the justification logics and replace each occurrence of a justification term with the modal operator K. More specifically, define a translation f as follows.

$$\begin{aligned} f(A) &= A \text{ for } A \text{ atomic} \\ f(X \supset Y) &= f(X) \supset f(Y) \\ f(t{:}X) &= Kf(X) \end{aligned}$$

The mapping f is called the *forgetful functor*. This mapping turns formulas in the justification language into formulas in conventional knowledge logic. It is easy to see that each axiom of JT maps to an axiom of T, and similarly for LP and S4, and for JS5 and S5. It is also easy to see that the forgetful functor maps applications of modus ponens and constant necessitation to applications of modus ponens and (ordinary) necessitation. Consequently we have the following, which is an observation originating in [1].

Proposition 5.1 *The forgetful functor maps theorems of* JT *to theorems of T. Likewise it maps theorems of LP to theorems of S4, and theorems of* JS5 *to theorems of S5.*

In fact, every theorem of one of our implicit logics is the image of the forgetful functor applied to a theorem of an explicit logic. This is a deep theorem in the subject, known as the *Realization Theorem*. To state it properly, we need the following.

Definition 5.2 (Realization) Let X be a formula in the implicit language, involving the operator K. A *realization* of X is a formula in the explicit language, that results by replacing each occurrence of K with some explicit justification, t. A realization is *normal* if negative occurrences of K are replaced with distinct variables (which are always part of the language of explicit justification logics).

Theorem 5.3 (Realization Theorem) *If X is a theorem of one of T, S4, or S5, there is some normal realization of X that is a theorem of* JT*, LP, or* JS5 *respectively.*

This result is due to Artemov, [1]. The original proof connecting LP and S4 was constructive, extracting a realization from a cut-free sequent calculus proof. Since then there have been other constructive proofs, all building on sequent calculi, and there is a non-constructive proof making use of the semantics from Section 4.

Here is an example, taken from [1]. The following is a theorem of S4:

$$(KA \vee KB) \supset K(KA \vee KB) \tag{3}$$

And here is a normal realization of (3), provable in LP.

$$(x{:}A \vee y{:}B) \supset (a\cdot!x + b\cdot!y){:}(x{:}A \vee y{:}B) \tag{4}$$

In (4) the role of the constants is as follows. Constant a is an unanalyzed justification of the classical axiom $x : A \supset (x : A \vee y : B)$ and b is an unanalyzed justification of the classical axiom $y{:}B \supset (x{:}A \vee y{:}B)$.

In effect, the Realization Theorem tells us we can reason in implicit logics of knowledge, which are simpler, then extract explicit reasons from the final result. Behind standard modal operators lies hidden machinery, which can be brought into the light as needed.

6 Transition

At the beginning we said this paper would be both an advertisement and a presentation of new results. We are reaching the end of the advertising part, and a brief summary of current research is appropriate—there is a sizable community at work on justification logics. The justification logics considered above were, in a sense, pure. There have been mixed versions, with both a Hintikka-style knowledge operator and explicit justification terms, and there have been multi-agent versions of such logics as well, [2, 4, 3, 7], with the justification terms shared by all—in effect they become common knowledge terms. Realization theorems for these logics are still open questions. Issues of communication in such logics are under investigation, [17]. There have been justification logics in which one can quantify over justifications, [9, 11] while under current development are justification logics in which one can quantify over things, as in standard first-order logics, [20].

Much has been accomplished, but there are still curious gaps in our knowledge about justification logics. So far there is no fully satisfactory multi-agent version in which each agent has its own set of justification terms, though there has been some work on this, [19]. This is perhaps related to the issue of what a reasonable set of justification terms and operations should be—are there ways of combining and modifying justifications used in the 'real' world that should be added to the formal machinery?

Perhaps the most frustrating gap so far is the connection between *modus ponens* and realization. Suppose both X and $X \supset Y$ are theorems of a standard logic of knowledge, $\mathsf{S4}$ say. And suppose we have realizations $X' \supset Y'$ of $X \supset Y$, and X'' of X, both provable in LP. By using *modus ponens* in $\mathsf{S4}$ we know that Y is provable, and hence it has a realization in LP. But so far, nobody knows how to extract a realization for Y from $X' \supset Y'$ and X''. Since Y is provable in $\mathsf{S4}$ it has a cut-free sequent calculus proof, and a realization can be extracted from that, but all information contained in the realizations for X and for $X \supset Y$ has been discarded in the process. Since *modus ponens* is so central to our ordinary reasoning, this is especially important. In some way the problem seems to be related to algorithms for cut-elimination in sequent calculi, but little is yet known despite much effort.

Now we move to the other side of the transition—new results. Let us begin with the obvious. In the sequence of modal logics T, $\mathsf{S4}$, $\mathsf{S5}$, each is stronger than the one before. They have the same vocabulary, so it does not make sense to ask if each is conservative over its predecessor. But as we have seen, each of these logics has an *explicit* counterpart, JT, LP, and $\mathsf{JS5}$. These actually have different vocabularies, and so conservativity questions can be meaningfully raised. In fact, each of these is a conservative extension of its predecessor, as we will show. Thus the machinery of justification logics helps make precise in what sense modal operators have been strengthened. In at least these cases, strengthening amounts to the addition of more 'hidden' machinery, but it is machinery that can be made explicit.

More generally, we will prove a fairly broad family of conservativity results, with the ones just mentioned among them. In order to state this family of results conveniently, we introduce some special terminology in the next section, then state and prove the results, and conclude with some open problems.

7 Alternate Terminology

The technical results to be shown below concern the logics JT, LP, and $\mathsf{JS5}$, and also several others that have not been looked at to date. In order to be able to state our results most simply, we now introduce some non-standard nomenclature. We do not intend this to become standard—it is merely a convenience for our present purposes.

Definition 7.1 Let S be a subset of $\{+, \cdot, !, ?\}$. For each such S let $L(S)$ be that part of the justification language described in Section 3, but with all justification operations restricted to the set S. We specify two justification logics whose language is $L(S)$. They are denoted $K(S)$ (with K for knowledge) and $B(S)$ (with B for belief). These have axioms and rules specified as follows.

1. For axioms, both $K(S)$ and $B(S)$ have the Classical Axioms. $K(S)$ assumes the Truth Axioms, while $B(S)$ does not. Finally, both assume the $+$ axiom if $+$ is in S, and similarly for $\cdot$, $!$, and $?$.

2. For rules, both have *Modus Ponens*. If both $\cdot$ and $!$ are in S, $K(S)$ and $B(S)$ have the *Axiom Necessitation* rule. If $\cdot$ is in S but $!$ is not, both have the *Iterated Axiom Necessitation* rule. Finally, if $\cdot$ is not in S, both have the *Theorem Necessitation* rule.

The primary utility of the notation introduced here is that it makes it very easy to state our results compactly. Since our nomenclature is not standard, here are some correspondences with the literature. Besides these logics, there are others that have been considered in the literature, and there are also systems that can be characterized in present terms, for example $K(\{!\})$, that have not been considered in the literature. (It's probably not very interesting.)

Standard Name	Name Used Here	Origin	Modal Version
LP(K)	$B(\{+,\cdot\})$	[5]	K
LP⁻(K)	$B(\{\cdot\})$	[8]	
JT, LP(T)	$K(\{+,\cdot\})$	[5]	T
LP⁻(T)	$K(\{\cdot\})$	[8]	
LP(K4)	$B(\{+,\cdot,!\})$	[5]	K4
LP⁻(K4)	$B(\cdot,!\})$	[8]	
LP	$K(\{+,\cdot,!\})$	[1]	S4
LP⁻	$K(\{\cdot,!\})$	[8]	
JS5, LP(S5)	$K(\{+,\cdot,!,?\})$	[16, 18]	S5

All the logics considered here have two fundamental properties common to justification logics. Since these will be needed in Section 9, they are stated now for the record.

Proposition 7.2 (Substitution Closure) *For every $S \subseteq \{+,\cdot,!,?\}$, both $K(S)$ and $B(S)$ are closed under substitution. That is, if X is a theorem of one of these logics, and X' is the result of replacing all occurrences of a variable x with a justification term t, then X' is also a theorem.*

The proof for this is standard. It is true for axioms, since they are specified by axiom schemes. Then one shows it is true for each line of a proof by induction on proof length. Generally the association of constants with axioms shifts when moving from a theorem to a substitution instance of it, but that does not matter for present purposes.

Proposition 7.3 (Internalization) *For every operator set $S \subseteq \{+,\cdot,!,?\}$, both $K(S)$ and $B(S)$ have the internalization property: if X is a theorem so is $t{:}X$ for some ground (that is, variable free) justification term t.*

If S contains $\cdot$, this proposition has a proof due to Artemov, [1]. If S does not contain $\cdot$, the proposition simply defaults to the Theorem Necessitation rule.

8 Results

Theorem 8.1 *Let $S_1, S_2 \subseteq \{+,\cdot,!,?\}$ and suppose $S_1 \subsetneq S_2$. Then $K(S_2)$ is a conservative extension of $K(S_1)$.*

The proof for this Theorem shows how to convert proofs from logic extensions back into proofs in the logic being extended. It does this by eliminating operator symbols. The rest of the section is devoted to giving the argument.

Definition 8.2 Let o be one of $+$, $\cdot$, $!$, or $?$. If X is a formula of $L(\{+,\cdot,!,?\})$, by X^o we mean the result of eliminating all justification terms containing o. More precisely, we have the following recursive characterization. For propositional letters $P^o = P$, and also $\bot^o = \bot$. Of course $(X \supset Y)^o = (X^o \supset Y^o)$. And finally:

$$(t{:}X)^o = \begin{cases} X^o & \text{if } o \text{ occurs in } t \\ t{:}X^o & \text{if } o \text{ does not occur in } t \end{cases}$$

The central part of the proof of Theorem 8.1 is contained in the following Proposition. Note that its proof is constructive (and simple).

Proposition 8.3 (Operator Elimination) *Assume $S \subseteq \{+, \cdot, !, ?\}$ and let o be one of the operation symbols in S. If Z is one of the axioms of $K(S)$, then Z^o is an axiom of $K(S - \{o\})$.*

Proof There are several cases and subcases, depending on choice of axiom and choice of operation symbol. The argument in each case is straightforward. It might be simpler to construct your own argument rather than reading mine. Here are the cases.

Classical Axiom: If Z is a tautology, so is Z^o.

Truth Axiom: Z **is** $t{:}X \supset X$ **or** $(X \supset Y) \supset (t{:}X \supset Y)$ There are two simple subcases

- o does not occur in t. Then Z^o is again a Truth Axiom, of the same kind.
- o occurs in t. Then Z^o is a Classical Axiom.

Truth Axiom: $Z = t{:}(X \supset Y) \supset (u{:}X \supset Y)$ Again there are simple subcases.

- o does not occur in t or in u. Then Z^o is again a Truth Axiom, of the same kind.
- o occurs in u but not in t. Then Z^o is $t{:}(X^o \supset Y^o) \supset (X^o \supset Y^o)$, a different kind of Truth Axiom.
- o occurs in t but not in u. Then Z^o is $(X^o \supset Y^o) \supset (u{:}X^o \supset Y^o)$, again a different kind of Truth Axiom.
- o occurs in both t and u. Then Z^o is $(X^o \supset Y^o) \supset (X^o \supset Y^o)$, a Classical Axiom.

$+$ **Axiom:** $Z = t{:}X \supset (t + u){:}X$ The other $+$ axiom is similar so only this one is considered.

- o occurs in t. Z^o is $X^o \supset X^o$, a Classical Axiom.
- o occurs in u but not in t. Z^o is $t{:}X^o \supset X^o$, a Truth Axiom.
- o occurs in neither t nor u, and o is not $+$. Z^o is $t{:}X^o \supset (t + u){:}X^o$, another $+$ axiom.
- o occurs in neither t nor u, and o is $+$. Z^o is $t{:}X^o \supset X^o$, a Truth Axiom.

$\cdot$ **Axiom:** $Z = (t{:}(X \supset Y) \supset (u{:}X \supset (t \cdot u){:}Y))$ The subcases are as follows.

- o occurs in both t and u. In this case Z^o is $(X^o \supset Y^o) \supset (X^o \supset Y^o)$, a Classical Axiom.
- o occurs in u but not in t. Then Z^o is $t{:}(X^o \supset Y^o) \supset (X^o \supset Y^o)$, an instance of the first Truth Axiom.
- o occurs in t but not in u. Then Z^o is $(X^o \supset Y^o) \supset (u{:}X^o \supset Y^o)$, an instance of the second Truth Axiom.
- o occurs in neither t nor u, and o is not $\cdot$. Then Z^o is $t{:}(X^o \supset Y^o) \supset (u{:}X^o \supset (t \cdot u){:}Y^o)$, an instance of the $\cdot$ Axiom.

o occurs in neither t nor u, and o is $\cdot$. Then Z^o is $t{:}(X^o \supset Y^o) \supset (u{:}X^o \supset Y^o)$, an instance of the third Truth Axiom.

! **Axiom:** $Z = t{:}X \supset !t{:}t{:}X$ The cases are as follows.

o occurs in t. Z^o is $X^o \supset X^o$, a Classical Axiom.

o does not occur in t, and o is not !. Z^o is $t{:}X^o \supset !t{:}t{:}X^o$, a ! Axiom.

o does not occur in t, and o is !. Z^o is $t{:}X^o \supset t{:}X^o$, a Classical Axiom.

? **Axiom:** $Z = \neg t{:}X \supset ?t{:}\neg t{:}X$ This case is similar to the ! case.

■

Finally there is very little left to do.

Proof of Theorem 8.1 Suppose $S_1 \subsetneq S_2$, where both are subsets of $\{+, \cdot, !, ?\}$. Assume S_2 contains a single operation symbol o that is missing from S_1. (The case of multiple operation symbols is handled by iterating the single operator case.) Let X be a theorem of $K(S_2)$, where X does not contain any occurrence of o. We show X is a theorem of $K(S_1)$.

Consider a proof of X in $K(S_2)$. Replace each line, Z, of that proof with Z^o. Each axiom of $K(S_2)$ is replaced with an axiom of $K(S_1)$, by Proposition 8.3. Applications of *modus ponens* turn into other applications of *modus ponens*. Also, applications of Constant Necessitation in $K(S_2)$ turn into applications of Constant Necessitation in $K(S_1)$, because $K(S_2)$ axioms turn into $K(S_1)$ axioms. Thus the entire proof converts to one in $K(S_1)$. Finally, since X did not contain o, it is still the last line of the proof, hence X is provable in $K(S_1)$. ■

9 Embedding and Equivalence

The role of constants in justification logics imposes certain peculiar complications. For instance consider LP, or $K(\{+, \cdot, !\})$ in present terminology. There is much flexibility possible in its axiomatization. For one thing we need an underpinning of classical logic, but that could be axiomatized in several ways—infinitely many different ways, in fact. But a choice of axiomatization affects applications of the *Constant Necessitation* rule. If, say, $X \supset X$ is an axiom, we can conclude $c{:}(X \supset X)$ for a constant c. If we have a different axiomatization of classical logic in which $X \supset X$ is not an axiom, nonetheless it will be a theorem, but then *Constant Necessitation* does not apply to it. We do, however, have the internalization feature to appeal to, Proposition 7.3: for some justification term t, $t{:}(X \supset X)$ will be a theorem. In some sense the difference between c and t shouldn't matter very much—what is basic in one axiomatization (and so has a constant justification) is subject to proof in the other (and so has a more complex justification). In this section we address the issue. Our treatment is not as general as might be desired. It assumes $\cdot$ and ! are present, so the version of Constant Necessitation used is Axiom Necessitation. And it assumes an *injective* constant specification (defined below) is used. Some of this can be relaxed, but the technical details become more complex. What is given here is enough to 'justify' the presence of three Truth Axiom schema in Section 3, instead of the customary single one.

Recall that in this section we are assuming Axiom Necessitation is the version of Constant Necessitation we use. A *constant specification* $\mathcal{C}$ is an assignment of axioms to constants. A proof meets constant specification $\mathcal{C}$ provided that whenever $c{:}X$ is introduced using the Axiom Necessitation rule, then X is a formula that $\mathcal{C}$ assigns to constant c. A constant specification can be given ahead of time, or created during the course of a proof. A constant specification is *injective* if at most one formula is associated with each justification term.

Definition 9.1 We say one justification logic, J_1, *embeds in another*, J_2, provided there is a mapping from constants of J_1 to justification terms of J_2 that converts each theorem of J_1 into a theorem of J_2.

We say two justification logics are *equivalent* if each embeds in the other.

Here is a basic result concerning these items. As noted earlier, this is not as general as it might be.

Theorem 9.2 (Embedding) *Let J_1 and J_2 be two justification logics in the same language $L(S)$, where $\{\cdot, !\} \subseteq S \subseteq \{+, \cdot, !, ?\}$. We assume the rules of infererence for J_1 and J_2 are* modus ponens *and Axiom Necessitation, as given in Section 3, but the choice of axioms may be entirely different. Suppose the following conditions are met.*

1. *An injective constant specification $\mathcal{C}$ is used for proofs in J_1.*
2. *J_1 is axiomatized using axiom schemes.*
3. *J_2 satisfies Substitution Closure (see Proposition 7.2).*
4. *J_2 satisfies Internalization (see Proposition 7.3).*
5. *Every axiom of J_1 is a theorem of J_2.*

Then J_1 embeds in J_2.

Proof We must create a mapping from constants of $L(S)$ to terms. If c is a constant that the constant specification $\mathcal{C}$ does not assign any J_1 axiom to, we simply map c to itself. Now suppose $\mathcal{C}$ does assign J_1 axiom A to c; we specify which term t the constant c maps to. Since $\mathcal{C}$ is injective, the axiom A is uniquely determined by c. Still, complications can arise due to the fact that A may contain an occurrence of c itself. If this happens, c is said to be *self-referential*, and it was shown in [14] that such self-referentiality is essential. (Thanks to Sergei Artemov for suggestions on how to handle this.) Suppose $c_1 (= c)$, c_2, $\ldots$, c_n are all the constants occurring in A (in some standard order). For clarity we write $A(c, c_2, \ldots, c_n)$ for A. Let x_1, x_2, $\ldots$, x_n be distinct variables not occurring in A (again in some standard order). Since A is an axiom of J_1, which is axiomatized using axiom schemes, then $A(x_1, x_2, \ldots, x_n)$ will also be an axiom. Then by hypothesis, $A(x_1, x_2, \ldots, x_n)$ is a theorem of J_2. Since J_2 satisfies Internalization, there is some ground justification term t such that $t{:}A(x_1, x_2, \ldots, x_n)$ is a theorem of J_2, if there is more than one such term, say we choose the first in some standard enumeration. Now, we map the constant c to the justification term t.

For each constant c, let c' be the term that was assigned to c above. For each formula Z in the language $L(S)$, let Z' be the formula that results when each constant c is replaced by the justification term c'.

Suppose Z_1, Z_2, $\ldots$, Z_n is a proof in the logic J_1, meeting constant specification $\mathcal{C}$. The sequence Z'_1, Z'_2, $\ldots$, Z'_n is not, itself, a proof in J_2, but each item in it is a theorem of J_2. This has a straightforward proof by induction. If Z_i is an axiom of J_1, since J_1 is axiomatized by schemes, Z'_i will also be an axiom, and hence a theorem of J_2 by hypothesis 5. If Z_i follows from earlier terms Z_j and $Z_j \supset Z_i$ by *modus ponens*, Z'_i also follows from Z'_j and $(Z_j \supset Z_i)' = (Z'_j \supset Z'_i)$ by *modus ponens*. Finally we have the Axiom Necessitation case. Suppose Z_i is $c{:}A$ where A is a J_1 axiom. Say $A = A(c, c_2, \ldots, c_n)$, where all the constants of A are explicitly displayed. If x_1, x_2, $\ldots$, x_n are variables not occurring in A, as above, there is a ground term $t = c'$ such that $t{:}A(x_1, x_2, \ldots, x_n)$

is a theorem of J_2. Since J_2 satisfies Internalization, $t{:}A(c', c'_2, \ldots, c'_n)$ is also a theorem, and this is $c'{:}A' = [c{:}A]' = Z'_i$.

It now follows that if X is any theorem of J_1 then X' will be a theorem of J_2. ■

Usually in the literature, the Truth Axiom is given by a single schema: $t{:}X \supset X$. We assumed two additional schemas: $(X \supset Y) \supset (t{:}X \supset Y)$ and $t{:}(X \supset Y) \supset (u{:}X \supset Y)$. It is an easy consequence of Theorem 9.2 that, for any S with $\{\cdot, !\} \subseteq S$, if we had axiomatized $K(S)$ with the usual single Truth schema instead of the way we did, the resulting logic would have been equivalent to the version we used, in the sense of Definition 9.1. Similarly we could have used "enough" tautologies instead of taking all of them as Classical Axioms, and that would have given equivalent logics as well. We made the choices we did because then the manipulations involved in the proof of Proposition 8.3 always turned axioms into axioms, and hence the behavior of Constant Necessitation was simple to describe.

10 Conclusion

The main thing left undone is quite obvious: there is no analog of Theorem 8.1 for logics of belief instead of knowledge—of the form $B(S)$ instead of $K(S)$. The methods of proof used here clearly do not extend to explicit logics of belief. Many of the cases involved in the proof of Proposition 8.3 yield an instance of a Truth Axiom. Without the Truth Axioms, present methods cannot succeed. Nonetheless, either a belief analog of Theorem 8.1 holds, or it does not. A result either way would be of interest. The desirable conjecture is that it holds, and some partial (unpublished) results have been obtained by Artemov.

One other item was left unfinished, but it is of lesser importance. Theorem 9.2 needed the presence of both $\cdot$ and !. Producing a version not needing ! is straightforward, but a bit messy to state. A version without $\cdot$ probably has little intrinsic interest. Also, giving an appropriate version (if there is one) not requiring an *injective* constant specification is still a problem. But this too is left to others.

References

[1] S. Artemov. Explicit provability and constructive semantics. *The Bulletin for Symbolic Logic*, 7(1):1–36, 2001.

[2] S. Artemov. Evidence-based common knowledge. Technical report, Technical Report TR-2004018, CUNY Ph. D. Program in Computer Science, 2004, 2004.

[3] S. Artemov. Justified common knowledge. *Theoretical Computer Science*, 357(1-3):4–22, July 2006.

[4] S. Artemov and E. Nogina. Introducing Justification into Epistemic Logic. *Journal of Logic and Computation*, 15(6):1059–1073, 2005.

[5] V. Brezhnev. Explicit counterparts of modal logic (in Russian). Master's thesis, Lomonosov Moscow State University, 1999.

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

[7] M. C. Fitting. Semantics and tableaus for LPS4. Technical Report TR-2004016, CUNY Ph.D. Program in Computer Science, 2004. `http://www.cs.gc.cuny.edu/tr/`.

[8] M. C. Fitting. The logic of proofs, semantically. *Annals of Pure and Applied Logic*, 132:1–25, 2005.

[9] M. C. Fitting. A quantified logic of evidence (short version). In R. de Queiroz, A. Macintyre, and G. Bittencourt, editors, *WoLLIC 2005 Proceedings*, Electronic Notes in Theoretical Computer Science, pages 59 – 70. Elsevier, 2005.

[10] M. C. Fitting. Explicit logics of knowledge and conservativity. In *Proceedings, Tenth International Symposium on Artificial Intelligence and Mathematics*, 2008. Electronic version at http://isaim2008.unl.edu/PAPERS/SS1-AI+Logic/MFitting-ss1.pdf.

[11] M. C. Fitting. A quantified logic of evidence. *Annals of Pure and Applied Logic*, 152(1-3):67–83, March 2008.

[12] M. C. Fitting. Reasoning with justifications. Forthcoming in *Towards Mathematical Philosophy*, in the *Trends in Logic* series, 2008.

[13] J. Hintikka. *Knowledge and Belief.* Cornell University Press, 1962.

[14] R. Kuznets. On self-referentiality in modal logic. *The Bulletin of Symbolic Logic*, 12(3):510, 2006.

[15] A. Mkrtychev. Models for the logic of proofs. In S. I. Adian and A. Nerode, editors, *Logical Foundations of Computer Science*, number 1234 in Lecture Notes in Computer Science, pages 266–275. Springer, 1997.

[16] E. Pacuit. A note on some explicit modal logics. In C. Dimitracopoulos, editor, *Proceedings of the Fifth Panhellenic Logic Symposium*, pages 117–125, 2005.

[17] B. Renne. Bisimulation and public announcements in logics of explicit knowledge. In S. Artemov and R. Parikh, editors, *Proceedings of the Workshop on Rationality and Knowledge, 18th European Summer School in Logic, Language, and Information (ESSLLI)*, pages 112–123, Málaga, Spain, 2006.

[18] N. Rubtsova. Evidence reconstruction of epistemic modal logic S5. In D. Grigoriev, J. Harrison, and E. A. Hirsch, editors, *Computer Science — Theory and Applications*, Lecture Notes in Computer Science, vol 3967, pages 313–321. Springer-Verlag, 2006.

[19] T. Yavorskaya (Sidon). Logic of proofs with two proof predicates. In D. Grivoriev, J. Harrison, and E. Hirsch, editors, *Computer Science — Theory and Applications*, Lecture Notes in Computer Science, vol 3967. Springer, 2006.

[20] R. Yavorsky. Provability logics with quantifiers on proofs. *Annals of Pure and Applied Logic*, 113(1-3):373–387, 2002.