Melvin Fitting

# Reasoning With Justifications

**Abstract.** This is an expository paper in which the basic ideas of a family of *Justification Logics* are presented. Justification Logics evolved from a logic called LP, introduced by Sergei Artemov [1, 3], which formed the central part of a project to provide an arithmetic semantics for propositional intuitionistic logic. The project was successful, but there was a considerable bonus: LP came to be understood as a logic of knowledge with explicit justifications and, as such, was capable of addressing in a natural way long-standing problems of logical omniscience. Since then, LP has become one member of a family of related logics, all logics of knowledge with explicit knowledge terms. In this paper the original problem of intuitionistic foundations is discussed only briefly. We concentrate entirely on issues of reasoning about knowledge.

## 1. Introduction

This is an expository paper in which the basic ideas of a family of *Justification Logics* are presented. Justification Logics evolved from a logic called LP, introduced by Sergei Artemov [1, 3], which formed the central part of a project to provide an arithmetic semantics for propositional intuitionistic logic. The project was successful, but there was a considerable bonus: LP came to be understood as a logic of knowledge with explicit justifications and, as such, was capable of addressing in a natural way long-standing problems of logical omniscience, [7]. Since then, LP has become one member of a family of related logics, all logics of knowledge with explicit knowledge terms. In this paper the original problem of intuitionistic foundations is discussed only briefly. We concentrate entirely on issues of reasoning about knowledge.

## 2. Hintikka's Logics of Knowledge

In [21] Hintikka developed an approach to logics of knowledge that has become the basis for much that followed. While the central ideas are generally familiar, a sketch of them will be useful. A logic with multiple agents is the natural one but for the time being we will confine things to a single agent, and discuss widening the setting towards the end of the paper.

---

Presented by **Name of Editor**; *Received* December 1, 2005

 

A propositional modal logic is constructed. It is customary to denote the necessity operator by $K$, standing for *it is known that*. We take $\supset$ and $\bot$ as basic, with other connectives defined in the usual way. Then a minimal logic of knowledge can be formulated as follows.

**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 **K3** can be seen as capturing part of the classic characterization of knowledge as justified, true belief: it says that what is known must be true. Without such an axiom we are capturing belief, not knowledge. Axiom **K2** is familiar from normal modal logics, but is somewhat problematic here. It says knowledge is closed under modus ponens—briefly, we know the consequences of what we know. This will be discussed further below. The **Necessitation** rule is also familiar from normal modal logics, and is also problematic here. It says all logical truths are known, and it too will be discussed further below.

These minimal axioms are generally extended with one or both of the following.

**Axiom Schemes**

**Positive Introspection** $KX \supset KKX$

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

The first says that if we know something, we know we know it. The second says if we don't know something, we know we don't know it. While these are increasingly strong, and increasingly doubtful assumptions, adding them both seems to have been the most common approach in the literature.

Historically, the justified knowledge approach sketched in this paper began with an analog of Hintikka's axioms including the one for Positive Introspection but not Negative Introspection. It was straightforward to provide an analog for the system without either Introspection axiom, and more recently an analog of the system with both Introspection axioms has appeared.

To keep things relatively simple, we will follow the historical development here, and assume only Positive Introspection.

The semantics Hintikka introduced is a possible world one. 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 that is reflexive and transitive (since we have positive introspection), and a notion of truth at a state, which we write as $\mathcal{M}, \Gamma \Vdash X$, where $\mathcal{M}$ is a model, $\Gamma$ is a state, and $X$ is a formula. On propositional connectives $\Vdash$ is truth-functional at each world, and the usual Kripke condition is met,

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

where this is usually read informally as: the agent knows $X$ at state $\Gamma$ if $X$ is the case at all states the agent cannot distinguish from $\Gamma$. For a single agent this logic of knowledge is simply the well-known modal logic S4—things become more complex when multiple agents are involved.

Hintikka's approach has been successfully applied to many well-known puzzles and problems, but it is not the end of the matter. What is it that an agent has knowledge of, sentences or propositions? Hintikka's logic is quite unproblematic when taken to be a logic of propositions—in it, 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 is not at all easy to see—we may not be aware of the equivalence. What we communicate directly is sentences, and propositions only indirectly. Wittgenstein argued that all mathematics is, essentially, a single tautology. In this sense, if we know the proposition that $2 + 2 = 4$, we know all mathematics—it's just a single proposition, the truth. But here the distinction between sentences and propositions is fundamental—mathematicians work with sentences directly, and propositions quite indirectly. Thought of as a logic of sentences, Hintikka's approach suffers from a fundamental difficulty usually referred to as *logical omniscience*. An agent turns out to know too much. This problem really breaks into two separate pieces, which we now discuss.

The first omniscience problem arises from the **Necessitation Rule**. According to this, an agent must know all tautologies. But a tautology could have as many symbols as there are atoms in the universe, and it is unlikely an agent actually would know the truth of such a formula. The second comes from axiom **K2**. It follows from this scheme that an agent would know the consequences of what it knows. This, too, seems unlikely in practice.

The usual sentence-based solution is to say that we are not really dealing with a logic of knowledge, but a logic of *potential* knowledge. $KX$ informally

means that $X$ is *knowable*, rather than actually known. This has its negative uses—if $\neg KX$ is established then $X$ is not knowable, so it is certainly not known, whatever we might mean by that. But still, a true logic of knowledge for sentences, not just knowability, would be a nice thing to have.

## 3. Awareness Logic

One of the reasons we might not know something that is knowable is that we haven't thought about it. In [11], Fagin and Halpern give this simple idea a formal treatment, producing a family of *awareness logics*. In these there is an explicit representation of the things one has thought about, so to speak.

Semantically, an awareness model is $\langle \mathcal{G}, \mathcal{R}, \mathcal{A}, \Vdash \rangle$, where $\mathcal{G}$, $\mathcal{R}$, and $\Vdash$ are as before, and $\mathcal{A}$ is a mapping assigning to each member of $\mathcal{G}$ a set of formulas. The members of $\mathcal{A}(\Gamma)$ are the formulas we are *aware* of at $\Gamma$. No special conditions are placed on this function; in particular, $\mathcal{A}(\Gamma)$ need not be complete, or consistent, or closed under subformulas. Syntactically, an operator $A$ is added to the language, and $AX$ is taken to be true at a state $\Gamma$ just in case $X \in \mathcal{A}(\Gamma)$.

With this machinery added, 'actual' knowledge can be represented as a conjunction $KX \wedge AX$. We explicitly know those formulas that are knowable, and that we have thought about. This now allows us to avoid logical omniscience problems—we easily have models in which formulas that might be problems are not because we are not aware of them.

The authors of [11] consider various natural conditions one might place on $\mathcal{A}$ such as closure under subformulas, or preservation on passing to accessible states (monotonicity). Likewise one might want to say one is only aware of formulas that are not too complicated, or formulas whose possible justifications are not too long. The framework is, in fact, very general. It is more of a toolbox, able to contain many things useful or not, than a tool in itself.

## 4. Explicit Justifications

Now we start on the main subject matter of this survey—logics with explicit justification terms. Instead of $KX$, that is, "$X$ is known," consider $t{:}X$, that is, "$X$ is known for the explicit reason $t$." Of course these explicit reasons, or justification terms, should have some internal structure. We introduce the basic machinery.

### 4.1. Syntax and Axiom System

A justification for $X \supset Y$ applied to a justification for $X$ should produce a justification for $Y$. The symbol $\cdot$, called *application*, is used and the basic principle is this.

1. $s{:}(X \supset Y) \supset (t{:}X \supset (s \cdot t){:}Y)$

Adding extra (perhaps useless) material to a justification still gives a justification, though a weaker one. The symbol $+$ is used for this.

2. $s{:}X \supset (s + t){:}X$
3. $t{:}X \supset (s + t){:}X$

A justification can be verified for correctness. It has it's own justification. The symbol ! is used here. This is referred to as *checking* or *verification*.

4. $t{:}X \supset !t{:}t{:}X$

A logical truth has a justification and no further analysis is needed. Constant symbols are used for this, that is, if $X$ is a 'basic' truth we can conclude $c{:}X$, where $c$ undergoes no further analysis. Of course, $c$ can be assigned a weight of some kind, reflecting the complexity of $X$, but this will not be done here.

In addition to logical truths, there are also facts of the world. These are, in a sense, inputs from outside the structure. They are represented by variables, thus $x{:}X$.

Gathering this together, we have the following formal language. First we have *justifications*, or *terms*, sometimes called *proof polynomials* when used in a mathematical setting.

***Variables:*** $x$, $y$, $z$, ..., are justifications.

***Constants:*** $c$, $d$, $e$, ..., are justifications.

***Application:*** If $s$ and $t$ are justifications, so is $(s \cdot t)$.

***Weakening:*** If $s$ and $t$ are justifications, so is $(s + t)$.

***Checking:*** If $t$ is a justification, so is $!t$.

Next, the definition of formulas.

***Propositional Letters:*** $P$, $Q$, ..., are formulas.

***Falsehood:*** $\bot$ is a formula.

***Implication:*** If $X$ and $Y$ are formulas, so is $(X \supset Y)$.

***Justification Formulas:*** If $X$ is a formula and $t$ is a justification then $t{:}X$ is a formula.

Axioms were, mostly, given above. Here they are in full. All formulas of the following form are axioms.

**J0** A sufficient set of classical tautologies

**J1** $s{:}(X \supset Y) \supset (t{:}X \supset (s \cdot t){:}Y)$

**J2** $s{:}X \supset (s + t){:}X$

**J3** $t{:}X \supset (s + t){:}X$

**J4** $t{:}X \supset !t{:}t{:}X$

**J5** $t{:}X \supset X$

Finally the rules of inference.

***Modus Ponens:*** From $X$ and $X \supset Y$ infer $Y$.

***Axiom Necessitation:*** If $X$ is an axiom, infer $c{:}X$, where $c$ is a constant symbol.

The logic just described is called LP, standing for *logic of proofs*, [1, 3]. The name comes from the fact that it was originally created to represent arithmetic proofs, which are certainly justifications of a very special kind. Variations on LP will be discussed in Section 9.

### 4.2. Semantics

The standard epistemic semantics for LP comes from [13], and amounts to a blending of an earlier semantics from [22] with the usual Hintikka style semantics for logics of knowledge. One can see it as being in the tradition of Awareness Logics, but with the awareness function supplied with an additional structure of justifications. A model is $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{E}, \Vdash \rangle$, where $\mathcal{G}$ and $\mathcal{R}$ are as usual, with $\mathcal{R}$ reflexive and transitive, and with $\Vdash$ behaving on propositional connectives in the usual way. The new item is $\mathcal{E}$, which is an *evidence* function. The idea is, $\mathcal{E}$ assigns to each possible world $\Gamma$ and to each justification $t$ a set of formulas—those formulas that $t$ is relevant to, or that $t$ can serve as possible evidence for, at $\Gamma$. Evidence functions must meet certain conditions.

***Monotonicity*** $\Gamma \mathcal{R} \Delta$ implies $\mathcal{E}(\Gamma, t) \subseteq \mathcal{E}(\Delta, t)$

***Application*** $X \supset Y \in \mathcal{E}(\Gamma, s)$ and $X \in \mathcal{E}(\Gamma, t)$ imply $Y \in \mathcal{E}(\Gamma, s \cdot t)$

***Weakening*** $\mathcal{E}(\Gamma, s) \cup \mathcal{E}(\Gamma, t) \subseteq \mathcal{E}(\Gamma, s + t)$

***Checking*** $X \in \mathcal{E}(\Gamma, t)$ implies $t{:}X \in \mathcal{E}(\Gamma, !t)$

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

$$\mathcal{M}, \Gamma \Vdash t{:}X \iff \mathcal{M}, \Delta \Vdash X \text{ for all } \Delta \in \mathcal{G} \text{ with } \Gamma \mathcal{R} \Delta \text{ and } X \in \mathcal{E}(\Gamma, t) \tag{2}$$

In short, we have $t{:}X$ at $\Gamma$ if $X$ is knowable at $\Gamma$ in the Hintikka sense, and $t$ is relevant evidence for $X$ at $\Gamma$. If we think of Hintikka semantics as capturing the idea of *true belief*, then what the present 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 $\Gamma$ (in the Hintikka approach) implies there is a justification for $X$ at $\Gamma$. Under simple, 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, [13].

### 4.3. Awareness Logics Again

LP can be seen as an extension of awareness logic with the awareness function made more explicit. One can extract a variety of awareness logics from LP directly, as well. For instance, we might define $\#(t)$ to be the number of operation symbols in term $t$; then $\mathcal{A}(\Gamma) = \{X \mid X \in \mathcal{E}(\Gamma, t) \text{ and } \#(t) < n\}$ is a natural awareness function, for each choice of $n$—we are aware of formulas with possible justifications that are not too complicated. Or again, we might set $\mathcal{A}(\Gamma) = \{X \mid X \in \mathcal{E}(\Gamma, t) \text{ and } t \in S\}$, where $S$ is a fixed set of justifications. Both these are quite plausible awareness functions, and others are easy to come by as well.

## 5. Internalization

In logics of knowledge, and more generally in normal modal logics, one has a *necessitation* rule: if $X$ is provable, so is $\Box X$. In LP this takes a much stronger form, and is a constructively provable theorem rather than a basic

rule. It is called *internalization* and is due to Artemov. It says, given a proof of $X$, then $t{:}X$ is provable for some closed term $t$, where $t$ embodies the given proof of $X$. Here is a very simple example. First we have a proof in LP, of $t{:}P \supset (c \cdot t){:}(P \vee Q)$, where $c$ is a proof constant for $P \supset (P \vee Q)$, and $t$ is some arbitrary proof term.

1. $c{:}(P \supset (P \vee Q))$
2. $c{:}(P \supset (P \vee Q)) \supset (t{:}P \supset (c \cdot t){:}(P \vee Q))$
3. $t{:}P \supset (c \cdot t){:}(P \vee Q)$

Line 1 is by axiom necessitation, 2 is an instance of axiom scheme 1, and 3 is from 1 and 2 by modus ponens. Internalizing this proof, the following is also LP provable, where $d$ is a proof constant for $c{:}(P \supset (P \vee Q)) \supset (t{:}P \supset (c \cdot t){:}(P \vee Q))$.

$$(d{\cdot}!c){:}(t{:}P \supset (c \cdot t){:}(P \vee Q))$$

Notice that $!c$ justifies line 1 of the proof above, $d$ justifies line 2, and $(d{\cdot}!c)$ justifies line 3 by representing the application of modus ponens. We omit the proof of this formula in LP.

## 6. Information Hiding and Recovery

In LP justifications are explicit, while in Hintikka-style logics of knowledge they are hiden. The knowledge operator of a Hintikka logic is a kind of existential quantifier asserting the existence of a justification without saying what it is. Explicit justifications can easily be hidden behind such quantifier-like operators. Remarkably, explicit justifications can also be recovered. This is the content of a fundamental theorem in the subject of justification logics.

First, the easy direction. As noted above, one can think of $K$ as a kind of existential quantifier: $KX$ is read as "there is a reason for $X$". Then if $X$ is a theorem of LP, with explicit justifications, and we replace each justification with $K$, we get a theorem of the Hintikka knowledge logic S4. This is easy to see. Each LP axiom scheme instance turns into an axiom of S4, and applications of LP rules of inference turn into applications of S4 rules of inference. Consequently an entire LP axiomatic proof converts into a proof in S4, and hence theorems convert as well.

A translation in the opposite direction is also possible, but much more difficult, and goes under the name *Realization Theorem*. It is due to Artemov, as is its first proof. Loosely it says, if $X$ is a theorem of S4, there is some way of replacing occurrences of $K$ with explicit justifications to produce a theorem of LP. But one can do better yet. In making the replacement

of $K$ symbols with justifications, negative occurrences of $K$ can always be replaced with distinct variables, and positive occurrences with justifications that may be computed from those variables. Thus S4 theorems have a hidden input-output structure that is exposed by a realization in LP.

Here is an example whose verification is left to the reader. $(KP \vee KQ) \supset K(KP \vee KQ)$ is a theorem of S4. And here is a realization of it, provable in LP, and with negative occurrences of $K$ replaced with distinct variables.

$$(x{:}P \vee y{:}Q) \supset (c{\cdot}!x + d{\cdot}!y){:}(x{:}P \vee y{:}Q)$$

In this, $c$ and $d$ are constant symbols introduced using the Axiom Necessitation rule, with $c$ introduced for the tautology $x{:}P \supset (x{:}P \vee y{:}Q)$ and $d$ for the tautology $y{:}Q \supset (x{:}P \vee y{:}Q)$.

Artemov's original proof of the Realization Theorem was entirely constructive, [3]. It extracted a provable realized version of an S4 theorem from a *cut-free sequent calculus* proof of the S4 theorem. Since then variations on the construction have been developed by several people, and the algorithm has become more efficient. In [13] a non-constructive proof was given, using the semantics described in Section 4.2. While not algorithmic, it goes more deeply into the role of the + operator. More recently a constructive proof along somewhat different lines, but still using a cut-free sequent calculus, was given in [15]—more about this in Section 8.

The Realization Theorem, and its associated algorithms, is central to understanding the significance of a logic like LP. It says we can reason Hintikka style and then, on demand, produce a conclusion with full justifications present. This holds great potential which is being explored by a number of researchers.

## 7. Original Intent

The logic LP is the first in a family of epistemic logics with explicit justifications; others will be discussed in section 9. But the original reason for its creation was quite different, and of considerable significance. It was part of a project to produce a constructive foundation for intuitionistic propositional logic. This project was successful. In this section we sketch the basic ideas.

There is a well-known BHK interpretation of the intuitionistic connectives (Brouwer, Heyting, Kolmogorov). Loosely, one thinks of intuitionistic truth as being rather like provability. This is often used informally to motivate intuitionistic logic. There have been various attempts to make the idea into a proper mathematical construct, Kleene's notion of realizability, for instance.

Gödel gave an axiomatization of the intuitive notion of provability, in [19]. He wrote *Bew* for the modal operator—we will use $\Box$ in this section. In his short paper he observed three fundamental things: 1) his axiomatization was equivalent to the Lewis logic S4; 2) intuitionistic logic embedded into it by inserting his 'provable' operator before each subformula; and 3) his axiomatization of provability did not correspond to the formal notion of provability in Peano arithmetic, because under an arithmetic interpretation $\Box X \supset X$ amounted to a consistency assertion. Thus the attempt was only partially successful, though tremendously influential. It was eventually realized that the logic of provability in Peano arithmetic was not S4, but GL, in which $\Box X \supset X$ is replaced with the Löb axiom, $\Box(\Box X \supset X) \supset \Box X$, but GL is not a logic into which one can embed intuitionistic logic, Gödel style.

Gödel made another suggestion, in [20], which remained largely unknown until the publication of his collected works. One might interpret S4 arithmetically, not as the logic of *provability*, but as the logic of *explicit proofs*. Sergei Artemov independently conceived the same idea, and carried it through to a successful conclusion, thus completing Gödel's project. The chain of construction goes as follows. First, as Gödel noted, propositional intuitionistic logic embeds in S4, with intuitionistic connectives being translated as 'provable' versions of their classical counterparts. Second, via the Realization theorem, S4 embeds in LP. And third, LP does embed into Peano arithmetic, with LP terms mapping to Gödel numbers of explicit arithmetic proofs—this is the Artemov Arithmetic Completeness theorem. All this combines to provide the desired arithmetic semantics for intuitionistic logic.

It became clear as early as 1998 that logics with explicit proof terms could also be seen as logics of explicit justifications in a more general sense, [2, 6] for instance. The introduction of a Kripke-style semantics for LP, [13], provided a significant technical and conceptual tool for epistemic applications. Today work on understanding and applying LP and its relatives to epistemic problems proceeds at an increasing pace. But from an *epistemic* point of view, and taking various generalizations of LP into account, arithmetic completeness is not central in the way it was for the intuitionistic logic project. The Realization theorem, however, remains fundamental. This accounts for the minimal mention the arithmetic result gets here—it is important, but for something other than the subject of our immediate concern.

## 8. Realizations As First-Class Objects

Originally a realization was simply a tool for extracting the explicit content of an S4 theorem. More recently realizations have become objects for in-

vestigation in their own right. In [17, 16, 15] they are functions mapping *occurrences* of necessity operators to justification terms (the use of $\Box$ instead of $K$ will be continued). Occurrences themselves are formally distinguished by breaking $\Box$ up into infinitely many copies, $\Box_1$, $\Box_2$, ..., with each having at most one occurrence in a formula. Further, positive and negative occurrences are distinguished, with even indexed operators in negative positions and odd indexed ones in positive positions. A formula with such subscripted modal operators is called *properly annotated*. Every modal formula can be properly annotated, and all properly annotated versions of the same formula are effectively interchangeable for our purposes. Then a realization is simply a function from positive integers to justification terms that maps even integers to distinct variables. If $r$ is a realization function and $X$ is an annotated formula, $r(X)$ is the result of replacing each subformula $\Box_i Y$ with $r(i){:}r(Y)$.

We also need the standard notion of substitution—recall that justification terms can contain variables. A substitution is a mapping, $\sigma$, from variables to justification terms. The result of applying a substitution $\sigma$ throughout a formula $X$ is denoted $X\sigma$. It is not hard to show that if $X$ is a theorem of $\mathsf{LP}$, so is $X\sigma$ for any substitution (though the role of constants may shift).

The direct use of realization functions, and of substitutions, has made it possible to state some algorithmic results concerning $\mathsf{LP}$ in a relatively coherent way—we do this in the rest of the section.

### 8.1. The Replacement Theorem

In $\mathsf{S4}$, as in every normal modal logic, one has a Replacement Theorem. If $A \equiv B$ is provable, then so is $X \equiv Y$, where $Y$ is like $X$ except that an occurrence of $A$ as a subformula has been replaced with $B$. (Multiple replacements can be handled sequentially.) To state this more easily, we use the following notation. Suppose $X(P)$ is a formula with at most one occurrence of the propositional letter $P$. Then we write $X(Z)$ for the result of substituting $Z$ for $P$ in $X(P)$. Now the usual Replacement Theorem can be stated as follows. If $\vdash_{\mathsf{S4}} A \equiv B$ then $\vdash_{\mathsf{S4}} X(A) \equiv X(B)$.

We saw in the previous section that, in $\mathsf{LP}$, positive and negative occurrences of subformulas sometimes play different roles. One problem with developing a Replacement Theorem for $\mathsf{LP}$ is that when $A \equiv B$ is expanded using $\wedge$, $\vee$, and $\neg$, one sees that $A$ has both a positive and a negative occurrence. Fortunately this difficulty can be addressed, because there is a 'polarity preserving' version of Replacement. Suppose $P$ has at most one *positive* occurrence in $X(P)$. Then if $\vdash_{\mathsf{S4}} A \supset B$ then $\vdash_{\mathsf{S4}} X(A) \supset X(B)$.

Here is this result again, for purposes of comparison with the corresponding LP result given below.

$$\frac{A \supset B}{X(A) \supset X(B)} \tag{3}$$

We might have a hope, then, that a polarity preserving version of Replacement will transfer to LP. But the obvious transfer does not work, and a moment's thought suggests why. Suppose we have $\vdash_{\mathsf{LP}} A \supset B$, and $P$ has at most one positive occurrence in $X(P)$. In $X(A)$ the subformula $A$ might occur within the scope of a justification term, and when we replace $A$ with $B$ to produce $X(B)$ we should expect that justification term will need updating to incorporate its original reasoning, plus a reason accounting for the passage from $A$ to $B$. That justification term itself may occur within the scope of another one, which will need updating, and so on up. In short, wherever appropriate, reasons must be modified to reflect the fact that $A$ implies $B$.

With realization and substitution machinery available a proper, algorithmic, version of Replacement for LP can be given. Suppose $X(P)$, $A$ and $B$ are properly annotated modal formulas (not LP formulas), where $P$ has a single positive occurrence in $X(P)$. Suppose also that $r_0$ is a realization function such that $\vdash_{\mathsf{LP}} r_0(A) \supset r_0(B)$. Then there is a pair, $\langle r, \sigma \rangle$ where $r$ is a realization function and $\sigma$ is a substitution, such that $\vdash_{\mathsf{LP}} r_0(X(A))\sigma \supset r(X(B))$. Schematically in LP we have the following, instead of (3).

$$\frac{r_0(A) \supset r_0(B)}{r_0(X(A))\sigma \supset r(X(B))} \tag{4}$$

In this, the substitution $\sigma$ and the new realization function $r$ take care of the 'justification adjustment' discussed above.

There are some conditions on the result above that must be stated. First, neither $A$ nor $B$ should share an index with $X(P)$. This is rather minor since annotations can always be reassigned in $X(P)$. Second and more serious, $r_0$ must be what is called *non self-referential on variables* over $X(A)$, that is, if $\Box_{2n}Z$ is a subformula of $X(P)$, and $r_0(2n)$ is the variable $x$, then $x$ does not occur in $r_0(Z)$. This is especially important since it was shown in [10] that self-referential constant symbols are essential for completeness.

There are also stronger versions of the Replacement Theorem than we stated. There are, for instance, restrictions on the behavior of the substitution $\sigma$. But most importantly, the pair $\langle r, \sigma \rangle$ carries out a replacement of $A$ by $B$ not only in $X(P)$, but in subformulas as well. (With negative subformulas the implication in the conclusion is reversed.) Thus it is a kind of *uniform* replacement.

The proof of the LP Replacement Theorem is entirely algorithmic, with the algorithm depending on the complexity of $X(P)$.

### 8.2. Realization Merging

Suppose we have two different realization functions; is there some way of merging them into a single one? As with the Replacement Theorem, there is an algorithmic solution to this problem too, and the result takes the following form. If $r_1$ and $r_2$ are realization functions, and $X$ is a properly annotated modal formula. Then there is a pair $\langle r, \sigma \rangle$, where $r$ is a realization function, $\sigma$ is a substitution, so that we have both $\vdash_{\mathsf{LP}} r_1(X)\sigma \supset r(X)$ and $\vdash_{\mathsf{LP}} r_2(X)\sigma \supset r(X)$. Actually, the full result says $\langle r, \sigma \rangle$ will merge not only $X$, but subformulas as well, but we can avoid the complexities in this survey paper.

Here is a simple example to show the utility of this. Suppose $A \supset C$ and $B \supset C$ are properly annotated modal formulas, and we have separate realization functions $r_1$ and $r_2$ such that $\vdash_{\mathsf{LP}} r_1(A \supset C)$ and $\vdash_{\mathsf{LP}} r_2(B \supset C)$. If we apply the algorithm referred to above, using $(A \vee B) \supset C$ for the formula $X$, a pair $\langle r, \sigma \rangle$ is produced, and it is not hard to show that $\vdash_{\mathsf{LP}} r((A \vee B) \supset C)$.

### 8.3. The Realization Theorem, Again

Both the Replacement result of section 8.1 and the Merging result of section 8.2 are special cases of a more general result which will not be stated here. Replacement, Merging, and one more consequence of the general theorem, together yield yet another algorithm for producing an LP provable realization of an S4 theorem, [15]. Like the original Artemov proof, it makes use of a cut-free S4 proof to create the provable realization. The relationship of this algorithm to the original one has not yet been determined.

### 8.4. What's Missing

A central open problem in this area concerns the familiar rule *modus ponens*. Suppose $X$ and $X \supset Y$ are modal formulas, and we have LP provable realizations for both. Then there will be a provable realization for $Y$ as well, by the following indirect argument. Since each of these has a provable realization, then $X$ and $X \supset Y$ themselves are provable in S4. But then so is $Y$, and by the Realization Theorem, it will have an LP provable realization. The problem is, the only algorithmic ways currently known for producing a provable realization of $Y$ is to begin with a cut-free proof of $Y$.

This is expensive, and we wind up discarding any information contained in the provable realizations for $X$ and $X \supset Y$ that we started with.

Is there a direct way of calculating a provable realization for $Y$, given provable realizations for $X$ and $X \supset Y$? Nobody knows how to handle this. The various merging and replacement algorithms discussed above do not apply. All of them pay careful attention to polarity of subformulas, but notice that the occurrence of $X$ in $X \supset Y$ is in a negative position, while in $X$ itself it is positive. Until this has been dealt with, we do not have a fully satisfactory calculus of realization functions.

## 9. Generalizations

LP is a version of S4 with explicit justifications. S4 is just one single agent logic of knowledge. What about others; what about multiple agents; what about communication of justifications?

### 9.1. Single Agent Logics

Logics weaker than S4 were investigated early on. T works fine. One simply removes the obvious axioms from LP. Actually, the axiom necessitation rule needs modification too—it now reads: if $X$ is an axiom, so is $c : X$ for a constant $c$, and now the same rule can be reapplied. This change is to compensate for the removal of the ! operator. K4 and K can be thought of as logics of belief rather than of knowledge, but again justification versions of them are straightforward to develop. For all these logics a Realization Theorem holds, [9], and completeness relative to a possible world semantics, as in section 4.2, can be shown.

In the other direction, S5 has also been given its justification version, independently in [25] and [23]. This requires the introduction of an additional operator, ?, dual to !. Whereas ! is designed for an explicit version of positive introspection, ? is intended to deal with negative introspection. Once again, Realization and semantical completeness have been established.

Several other single-knower variations have been considered, but this should be enough to give the general picture.

### 9.2. Multiple Agent Logics

There has already been work on justification versions of multiple agent Hintikka logics of knowledge. The idea has been to treat shared justifications as a kind of explicit common knowledge [4, 8, 5]. In fact, explicit justifications

can be shown to satisfy the fixpoint condition usually imposed on common knowledge.

More recent work, still in progress, concerns multiple agent logics in which each agent has its own set of reasons, [26]. This is a natural setting for considering the results of communication, and some work has been done on the justification version of public announcements [24]. Among other things, this involves putting a syntactic counterpart of the semantic evidence function into the language in a way reminiscent of the treatment of the awareness function, discussed in section 3.

## 10. The Goal

What we are after is logics in which we can reason, not just about facts, but about reasons for facts, and in which we can reason about these reasons conveniently and efficiently. Work is very much in progress. We conclude with a quick summary of the current state of things.

For single agent logics of knowledge, both implicit (Hintikka style) versions and explicit (justification style) versions exist, and there is effective machinery to translate between them. How to handle *modus ponens* is still an open problem, but some progress is being made on this. In the meantime, cut-free sequent formulations serve as tools, though expensively. There are versions in which both implicit and explicit knowledge can be expressed. These are natural, and have a well-understood proof theory and semantics. What is missing for these is a version of the Realization Theorem, which now would take the form of 'self-realization.' Here the treatment of *modus ponens* seems to be even more of a central issue.

Multiple agent logics of knowledge with explicit justifications are not as well developed yet, partly because of the additional richness available. So far the most successful versions have had implicit knowledge individually, while explicit justifications were shared machinery. The two central issues currently being explored are: allowing a separate family of justifications for each agent, and permitting communication of justifications. This is perhaps the most active current area of development.

One might consider a move to a first-order logic of knowledge, implicit and explicit. One might have quantifiers over things or even quantifiers over justifications. There has been some work on this, [27, 18, 14], but the situation is complex and not yet well-understood.

Formalizing the reasoning of knowledge with justifications that are explicitly present has turned out to be a rich source of results and techniques. Already what has been achieved is significant, and progress remains steady.

The range of logical systems that has resulted presents an exciting mix of expressiveness and succinctness, with strong proof-theoretic and semantic aspects. It is to be hoped that this work will lead to a better understanding of reasoning, its explanation and communication.

## References


[1] S. Artemov. Operational modal logic. Technical Report MSI 95-29, Cornell University, December 1995.

[2] S. Artemov. Logic of proofs: a unified semantics for modality and lambda-terms. Technical Report CFIS 98-17, Cornell University, 1998.

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

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

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

[6] S. Artemov, E. Kazakov, and D. Shapiro. On logic of knowledge with justifications. Technical Report CFIS 99-12, Cornell University, 1999.

[7] S. Artemov and R. Kuznets. Logical omniscience via proof complexity. In *Computer Science Logic 2006*, Lecture Notes in Computer Science, Vol 4207, pages 135–149. Lecture Notes in Computer Science, Vol 4207, Springer, 2006.

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

[9] V. Brezhnev. On the logic of proofs. In K. Striegnitz, editor, *Proceedings of the Sixth ESSLLI Student Session*, pages 35–46, Helsinki, 2001.

[10] V. Brezhnev and R. Kuznets. Making knowledge explicit: How hard it is. *Theoretical Computer Science*, 357:23–34, 2006.

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

[12] S. Feferman, editor. *Kurt Gödel Collected Works*. Oxford, 1986-2003. Five volumes.

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

[14] 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.

[15] M. C. Fitting. Realizations and LP. Available at `http://comet.lehman.cuny.edu/fitting/`, 2006.

[16] M. C. Fitting. Realizing substitution instances of modal theorems. Available at `http://comet.lehman.cuny.edu/fitting/`, 2006.

[17] M. C. Fitting. A replacement theorem for LP. Technical report, CUNY Ph.D. Program in Computer Science, 2006. `http://www.cs.gc.cuny.edu/tr/`.

[18] M. C. Fitting. A quantified logic of evidence. *Annals of Pure and Applied Logic*, 2007. Forthcoming.

[19] K. Gödel. Eine Interpretation des intuistionistischen Aussagenkalkuls. *Ergebnisse eines mathematischen Kolloquiums*, 4:39–40, 1933. Translated as *An interpretation of the intuitionistic propositional calculus* in [12] I, 296-301.

[20] K. Gödel. Vortrag bei Zilsel. Translated as *Lecture at Zilsel's* in [12] III, 62-113, 1938.

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

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

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